Ministry of Higher Education and Scientific Research
General Directorate of Technological Studies
Higher Institute of Technological Studies of Bizerte
Information Technology Department

Reference	**Dep.**	IT
	AY	2026
	No.	DSI2625

Report of

END-OF-STUDIES PROJECT

In partial fulfillment for:

Licence en Informatique

Development of an Internal Platform for Managing Requests, Projects, and Tasks at ArabSoft

Prepared by:

Abderrahmen Ben Hattab

Supervised by:

Ms. Anissa CHALOUAH (ISET)

Mr. Mouadh Zidi (COMPANY)

Conducted at:

Company: ArabSoft

- Address: Montplaisir, Tunis
- Tel: 71 951 248
- Email: arabsoft@arabsoft.com.tn

Academic Year: 2025/2026

Dedication

To all those whose presence illuminated every step of this journey…

To my father, Adel,

You taught me that hard work and perseverance
are the keys to every success.
Your constant support and trust in me
have been my anchor throughout this journey.
This work is the result of your precious guidance.

To my mother, Amira,

An endless source of love and encouragement,
your caring presence has always carried me.
Every effort I made contains a part of you,
because you have been my greatest strength.

To my sister, Sarra,

For your joy, your support, and your generosity.
With one smile, you always gave me courage
in the most difficult moments.
Your presence is a precious gift.

To all my friends and colleagues, and to everyone who contributed,
directly or indirectly, to the completion of this work.

Abderrahmen Ben Hattab

Acknowledgements

First, I thank **Allah** for granting me the health, determination, and courage required to complete this work.

I express my sincere gratitude to my academic supervisor, **Miss Anissa CHALOUAH**, lecturer at ISET Bizerte, for her valuable guidance, insightful advice, and continuous availability throughout this project.

I also warmly thank my company supervisor, **Mr. Mouadh Zidi**, from **ArabSoft**, for his welcome, professional support, and the knowledge he shared with me.

My thanks also go to the entire **ArabSoft** team for their collaboration, availability, and the positive work environment they provided during the internship period.

I am equally grateful to all faculty and administrative staff of the **Information Technology Department** at ISET Bizerte for the quality of education and support provided throughout my university journey.

Finally, I thank everyone who contributed, directly or indirectly, to the completion of this end-of-studies project.

Table of Contents

General Introduction **1**

Chapter 1: Project Context Presentation **2**

I. Host organization presentation 2

I.1 General presentation of ArabSoft 2

I.2 ArabSoft in Tunisia 2

II. Project presentation 3

II.1 Current-state study 3

II.2 Current-state critique 3

II.3 Proposed solution 4

III. Adopted methodologies and formalism 5

III.1 Modeling and design methodology 5

III.2 Working methodology 5

IV. Development environment 7

IV.1 Hardware environment 7

IV.2 Software environment 8

IV.3 Technologies used 9

V. Application architecture 11

V.1 Global architecture 11

V.2 Frontend part: MVVM architecture with React 11

V.3 Backend part: layered architecture with Spring Boot 12

V.4 Why these architectures? 13

V.5 Global architecture illustration 14

Conclusion 15

Chapter 2: Product Backlog Planning **16**

Introduction 16

I. Actor Identification 16

I.1 Employee 16

I.2 Team Leader 16

I.3 HR Manager 16

I.4 New User 16
II. Functional Requirements 16
II.1 Authentication and access management 16
II.2 Administrative request management 17
II.3 Team and employee management 17
II.4 Project and task management 17
II.5 Notification and reporting 17
II.6 AI assistance and decision support 17
III. Non-Functional Requirements 17
III.1 Security 17
III.2 Performance 17
III.3 Usability 17
III.4 Maintainability 18
III.5 Scalability 18
IV. User Stories 18
IV.1 Employee User Stories 18
IV.2 Team Leader User Stories 19
IV.3 HR Manager User Stories 20
V. Product Backlog 21
VI. Release Organization 25
Conclusion 25
Chapter 3: Release 1 - Foundations 26
Introduction 26
I. Release Organization 26
II. Sprint 1: Access and Account Management 27
II.1 Sprint Goal 27
II.2 Visual Sprint Journal 27
II.3 Sprint Backlog 27
II.4 Sprint Implementation 29
II.5 Sprint Review 62
II.6 Sprint Retrospective 63
III. Sprint 2: Leave Management 64
III.1 Sprint Goal 64
III.2 Visual Sprint Journal 64

III.3 Sprint Backlog 64
III.4 Sprint Implementation 66
III.5 Sprint Review 88
III.6 Sprint Retrospective 88
Conclusion 89
Chapter 4: Release 2 - Workflow Management 90
Introduction 90
I. Release Organization 90
II. Sprint 3: Administrative Requests 91
II.1 Sprint Goal 91
II.2 Visual Sprint Journal 91
II.3 Sprint Backlog 91
II.4 Sprint Implementation 93
II.5 Sprint Review 127
II.6 Sprint Retrospective 127
III. Sprint 4: HR Management and Calendar 128
III.1 Sprint Goal 128
III.2 Visual Sprint Journal 128
III.3 Sprint Backlog 128
III.4 Sprint Implementation 130
III.5 Sprint Review 153
III.6 Sprint Retrospective 153
Conclusion 154
Chapter 5: Release 3 - Advanced Support and Intelligence 155
Introduction 155
I. Release Organization 155
II. Sprint 5: Projects, Tasks and Reports 156
II.1 Sprint Goal 156
II.2 Visual Sprint Journal 156
II.3 Sprint Backlog 157
II.4 Sprint Implementation 158
II.5 Sprint Review 175
II.6 Sprint Retrospective 176
III. Sprint 6: AI Assistant, Local Deployment, and Stabilization 176

III.1 Sprint Goal 176
III.2 Visual Sprint Journal 176
III.3 Sprint Backlog 176
III.4 Requirements Analysis 178
III.5 Sprint Review 187
III.6 Sprint Retrospective 187

General Conclusion **189**

Bibliographie et Nétographie **190**

List of Figures

Figure 1: ArabSoft logo 2
Figure 2: UML logo 5
Figure 3: Scrum team roles 7
Figure 4: SCRUM method 7
Figure 5: PC specifications used for development 8
Figure 6: MVVM frontend architecture with React 12
Figure 7: Layered architecture with Spring Boot 13
Figure 8: Global application architecture 14
Figure 9: Organization of Releases 25
Figure 10: Release 1 Organization 26
Figure 11: Use Case Diagram – Sprint 1 Access and Account Management 29
Figure 12: System Sequence Diagram – User Registration 30
Figure 13: System Sequence Diagram – Secure Login 31
Figure 14: System Sequence Diagram – Logout 32
Figure 15: Use Case Diagram – Personal Account Management 32
Figure 16: System Sequence Diagram – Password Reset 33
Figure 17: System Sequence Diagram – User Information Update 34
Figure 18: System Sequence Diagram – Authenticated Password Change 35
Figure 19: Use Case Diagram – HR User Account Management 36
Figure 20: System Sequence Diagram – HR User Account Setup 36
Figure 21: System Sequence Diagram – Account Activation and Deactivation 37
Figure 22: Domain Model – Sprint 1 Access and Account Management 38
Figure 23: Participant Class Diagram – Authentication and Access Control 39
Figure 24: Participant Class Diagram – Password Reset and Password Change 39
Figure 25: Participant Class Diagram – HR Account Management 40
Figure 26: State Diagram – User Account Lifecycle 41
Figure 27: Design Class Diagram – Registration and Pending Access 42
Figure 28: Design Class Diagram – Security, Keycloak, and Access Control 42

Figure 29: Design Class Diagram – HR Account Setup and Notification 43
Figure 30: Conception Sequence Diagram – User Registration Frontend 44
Figure 31: Conception Sequence Diagram – User Registration Backend 45
Figure 32: Conception Sequence Diagram – Secure Login Frontend 46
Figure 33: Conception Sequence Diagram – Secure Login Backend 47
Figure 34: Conception Sequence Diagram – Logout Frontend 48
Figure 35: Conception Sequence Diagram – Logout Backend 49
Figure 36: Conception Sequence Diagram – Password Reset Frontend..... 50
Figure 37: Conception Sequence Diagram – Password Reset Backend 51
Figure 38: Conception Sequence Diagram – User Information Update Frontend ... 52
Figure 39: Conception Sequence Diagram – User Information Update Backend ... 53
Figure 40: Conception Sequence Diagram – Authenticated Password Change Frontend 54
Figure 41: Conception Sequence Diagram – Authenticated Password Change Backend 55
Figure 42: Conception Sequence Diagram – HR User Account Setup Frontend 56
Figure 43: Conception Sequence Diagram – HR User Account Setup Backend 57
Figure 44: Conception Sequence Diagram – Account Activation and Deactivation Frontend 58
Figure 45: Conception Sequence Diagram – Account Activation and Deactivation Backend 59
Figure 46: Secure Login Interface 60
Figure 47: Pending Account Access After Registration 60
Figure 48: Password Reset Interface 60
Figure 49: User Profile Update Interface 61
Figure 50: Authenticated Password Change Interface 61
Figure 51: HR User Account Setup Interface 62
Figure 52: Account Activation and Deactivation Interface 62
Figure 53: Use Case Diagram – Sprint 2 Leave Management 66
Figure 54: Use Case Diagram – Employee Leave Request Management 67
Figure 55: System Sequence Diagram – Submit Leave Request 68
Figure 56: System Sequence Diagram – Edit or Cancel Pending Leave 68
Figure 57: Use Case Diagram – Leave Approval Workflow 69

Figure 58: System Sequence Diagram – Team Leader Leave Decision 70
Figure 59: System Sequence Diagram – HR Final Leave Decision 71
Figure 60: Domain Model – Sprint 2 Leave Management 72
Figure 61: Participant Class Diagram – Employee Leave Request Management 73
Figure 62: Participant Class Diagram – Team Leader Leave Review 73
Figure 63: Participant Class Diagram – HR Final Leave Decision 74
Figure 64: State Diagram – Leave Request Lifecycle 75
Figure 65: Design Class Diagram – Leave Request Creation and Validation 76
Figure 66: Design Class Diagram – Team Leader Leave Review 76
Figure 67: Design Class Diagram – HR Final Leave Decision 77
Figure 68: Design Class Diagram – Working Day, Holiday, and Balance Management 77
Figure 69: Conception Sequence Diagram – Submit Leave Request Frontend 78
Figure 70: Conception Sequence Diagram – Submit Leave Request Backend 79
Figure 71: Conception Sequence Diagram – Edit or Cancel Pending Leave Frontend 80
Figure 72: Conception Sequence Diagram – Edit or Cancel Pending Leave Backend 81
Figure 73: Conception Sequence Diagram – Team Leader Review Frontend 82
Figure 74: Conception Sequence Diagram – Team Leader Review Backend 83
Figure 75: Conception Sequence Diagram – HR Final Review Frontend 84
Figure 76: Conception Sequence Diagram – HR Final Review Backend 85
Figure 77: Employee Leave Requests Interface 86
Figure 78: Leave Request Details Interface 86
Figure 79: Team Leader Leave Review Interface 87
Figure 80: HR Final Leave Decision Interface 87
Figure 81: Leave Calendar Interface 88
Figure 82: Release 2 Organization 90
Figure 83: Use Case Diagram – Sprint 3 Administrative Requests 94
Figure 84: Use Case Diagram – HR Administrative Request Processing 94
Figure 85: System Sequence Diagram – Submit Document Request 95
Figure 86: System Sequence Diagram – Process Document Request 96
Figure 87: System Sequence Diagram – Submit Loan Request 96
Figure 88: System Sequence Diagram – Schedule Loan Meeting 97

Figure 89: System Sequence Diagram – Submit Authorization Request 98
Figure 90: System Sequence Diagram – HR Administrative Request Decision 99
Figure 91: Domain Model – Sprint 3 Administrative Requests 100
Figure 92: Participant Class Diagram – Document Request Management 101
Figure 93: Participant Class Diagram – Loan Request Management 101
Figure 94: Participant Class Diagram – Authorization Request Management 102
Figure 95: Participant Class Diagram – HR Request Processing 102
Figure 96: State Diagram – Document Request Lifecycle 103
Figure 97: State Diagram – Loan Request Lifecycle 104
Figure 98: State Diagram – Authorization Request Lifecycle 105
Figure 99: Design Class Diagram – Document Request Final File Management ... 106
Figure 100: Design Class Diagram – Loan Request Review Management 107
Figure 101: Design Class Diagram – Authorization Request Management 108
Figure 102: Design Class Diagram – HR Request Processing 108
Figure 103: Conception Sequence Diagram – Document Request HR Decision Frontend 109
Figure 104: Conception Sequence Diagram – Document Request Decision Backend 110
Figure 105: Conception Sequence Diagram – Document Request Final File Frontend 111
Figure 106: Conception Sequence Diagram – Document Request Final File Backend 112
Figure 107: Conception Sequence Diagram – Loan Request Submission Frontend .. 113
Figure 108: Conception Sequence Diagram – Loan Request Submission Backend .. 114
Figure 109: Conception Sequence Diagram – Loan Request Meeting Scheduling Frontend 115
Figure 110: Conception Sequence Diagram – Loan Meeting Scheduling Backend .. 116
Figure 111: Conception Sequence Diagram – Loan Request Decision Frontend 117
Figure 112: Conception Sequence Diagram – Loan Request Decision Backend 118
Figure 113: Conception Sequence Diagram – Loan Final File Delivery Frontend 119
Figure 114: Conception Sequence Diagram – Loan Final File Backend 120
Figure 115: Conception Sequence Diagram – Authorization Request Submission Frontend 121
Figure 116: Conception Sequence Diagram – Authorization Request Submission Backend 122

Figure 117: Conception Sequence Diagram – Authorization Request Decision Frontend 123
Figure 118: Conception Sequence Diagram – Authorization Request Decision Backend 124
Figure 119: Employee Documents Interface 125
Figure 120: Employee Loans and Advances Interface 125
Figure 121: Employee Authorizations Interface 126
Figure 122: HR Administrative Requests Interface 126
Figure 123: Request Detail and History Interface 127
Figure 124: Use Case Diagram – Sprint 4 HR Management and Calendar 130
Figure 125: Use Case Diagram – HR Employee and Account Supervision 131
Figure 126: Use Case Diagram – Organization and Team Management 131
Figure 127: Use Case Diagram – Calendar Supervision 132
Figure 128: System Sequence Diagram – Consult HR Dashboard 133
Figure 129: System Sequence Diagram – Supervise Employee Account 133
Figure 130: System Sequence Diagram – Manage Organization Reference Data 134
Figure 131: System Sequence Diagram – Manage Team Structure 134
Figure 132: System Sequence Diagram – Consult HR Leave Calendar 135
Figure 133: System Sequence Diagram – Consult Team Leave Calendar 135
Figure 134: Domain Model – Sprint 4 HR Management and Calendar 136
Figure 135: Participant Class Diagram – HR Employee and Account Supervision... 137
Figure 136: Participant Class Diagram – Organization Reference Data Management 137
Figure 137: Participant Class Diagram – Team Structure Management 138
Figure 138: Participant Class Diagram – HR Leave Calendar Consultation 138
Figure 139: Participant Class Diagram – Team Leave Calendar Consultation 139
Figure 140: Design Class Diagram – HR User Management 139
Figure 141: Design Class Diagram – HR Organization Setup 140
Figure 142: Design Class Diagram – HR Team Management 140
Figure 143: Design Class Diagram – HR Leave Calendar 141
Figure 144: Design Class Diagram – Team Leave Calendar 141
Figure 145: Conception Sequence Diagram – HR User Management Frontend 142
Figure 146: Conception Sequence Diagram – HR User Management Backend 143

Figure 147: Conception Sequence Diagram – HR Organization Setup Frontend 144
Figure 148: Conception Sequence Diagram – HR Organization Setup Backend 145
Figure 149: Conception Sequence Diagram – HR Team Management Frontend 146
Figure 150: Conception Sequence Diagram – HR Team Management Backend 147
Figure 151: Conception Sequence Diagram – HR Leave Calendar Frontend 148
Figure 152: Conception Sequence Diagram – HR Leave Calendar Backend 148
Figure 153: Conception Sequence Diagram – Team Leave Calendar Frontend 149
Figure 154: Conception Sequence Diagram – Team Leave Decision Backend 150
Figure 155: HR Dashboard Interface 151
Figure 156: HR User Management Interface 151
Figure 157: Team Management Interface 151
Figure 158: Organization Reference Data Interface 152
Figure 159: HR Leave Calendar Interface 152
Figure 160: Release 3 Organization 156
Figure 161: Use Case Diagram – Sprint 5 Projects, Tasks and Reports 158
Figure 162: Use Case Diagram – Team Leader Project and Task Management 159
Figure 163: System Sequence Diagram – Create Project 159
Figure 164: System Sequence Diagram – Create and Assign Task 160
Figure 165: System Sequence Diagram – Update Task Status 160
Figure 166: System Sequence Diagram – Consult HR Reports 161
Figure 167: Domain Model – Sprint 5 Projects, Tasks and Reports 162
Figure 168: State Diagram – Task Lifecycle 163
Figure 169: Participant Class Diagram – Team Leader Project and Task Management 164
Figure 170: Participant Class Diagram – Task Status Update 164
Figure 171: Participant Class Diagram – HR Reports and Export 165
Figure 172: Design Class Diagram – Team Leader Project and Task Management .. 165
Figure 173: Design Class Diagram – Task Status Update 166
Figure 174: Design Class Diagram – HR Reports and Excel Export 166
Figure 175: Conception Sequence Diagram – Project Management Frontend 167
Figure 176: Conception Sequence Diagram – Project Management Backend 168
Figure 177: Conception Sequence Diagram – Task Assignment Frontend 169
Figure 178: Conception Sequence Diagram – Task Assignment Backend 170

Figure 179: Conception Sequence Diagram – Task Status Update Frontend 171
Figure 180: Conception Sequence Diagram – Task Status Update Backend 171
Figure 181: Team Projects and Tasks Interface 172
Figure 182: Task Assignment Interface 173
Figure 183: Employee Task Tracking Interface 174
Figure 184: HR Request Reports Interface 174
Figure 185: Excel Report Export Result 175
Figure 186: Use Case Diagram – Sprint 6 AI Assistant and Decision Support 178
Figure 187: System Sequence Diagram – Assistant Platform Question 179
Figure 188: System Sequence Diagram – Assistant Request Drafting 180
Figure 189: System Sequence Diagram – Team Leader Decision Support 180
Figure 190: Domain Model – Sprint 6 AI Assistant Context Data 181
Figure 191: Participant Class Diagram – AI Assistant Gateway and Safe Context ... 182
Figure 192: Design Class Diagram – Assistant Gateway and Safe Context 182
Figure 193: Conception Sequence Diagram – Assistant Platform Question Frontend 183
Figure 194: Conception Sequence Diagram – Assistant Platform Question Backend 183
Figure 195: Conception Sequence Diagram – Assistant Request Drafting Frontend . 184
Figure 196: Conception Sequence Diagram – Assistant Request Drafting Backend . 184
Figure 197: Conception Sequence Diagram – Team Leader Decision Support Frontend 185
Figure 198: Conception Sequence Diagram – Team Leader Decision Support Backend 185
Figure 199: Assistant Platform Guidance Interface 186
Figure 200: Assistant Leave Draft Preview Interface 186
Figure 201: Team Leader Decision Support Interface 187

List of Tables

Table 1: Product Backlog 21
Table 2: Sprint Backlog – Sprint 1 28
Table 3: Sprint Retrospective – Sprint 1 63
Table 4: Sprint Backlog – Sprint 2 65
Table 5: Sprint Retrospective – Sprint 2 88
Table 6: Sprint Backlog – Sprint 3 92
Table 7: Sprint Retrospective – Sprint 3 127
Table 8: Sprint Backlog – Sprint 4 129
Table 9: Sprint Retrospective – Sprint 4 153
Table 10: Sprint Retrospective – Sprint 5 176
Table 11: Sprint Retrospective – Sprint 6 188

General Introduction

Digital transformation has become a major strategic driver for companies seeking to improve organizational efficiency and ensure smooth management of internal processes. The centralization of administrative requests, project tracking, and internal communication represents significant challenges when operations are handled manually or in a fragmented manner. The growing complexity of internal workflows, together with the need for effective coordination between departments, makes the adoption of modern and structured digital tools essential.

In this context, my work focuses on the **development of an internal platform for managing requests, projects, and tasks at ArabSoft**. The objective is to design a practical and reliable solution that centralizes administrative requests, monitors project progress, and improves communication between employees, team leaders, and HR managers. By providing a global and real-time view of organizational activities, this platform aims to strengthen collaboration, improve process efficiency, and support better operational decision-making.

The ambition of this project is to provide a tool that addresses current organizational needs while remaining adaptable to future evolution. Whether for handling internal requests, tracking project progress, assigning tasks, or improving information flow between stakeholders, the platform is designed as a complete and intuitive solution that integrates smoothly into existing work practices.

This report presents the different phases of project realization. From requirements analysis to deployment, including design, development, and testing phases, I explain how this platform was built to meet user expectations and deliver concrete added value to the company. Particular attention is given to the methodological approach, technical choices, and implementation strategy adopted throughout the project.

Through this work, I aim to contribute to the modernization of internal management practices and to the continuous improvement of organizational performance. This project represents an opportunity to rethink how administrative and project activities are managed by making them more structured, collaborative, and efficient.

Chapter 1

Project Context Presentation

I. Host organization presentation

1. General presentation of ArabSoft

ArabSoft is a company specialized in software engineering and digital solutions for organizations. It operates in several areas such as business process management, enterprise application development, systems integration, and digital transformation support. Through its experience in information systems, the company helps institutions modernize operations and improve service quality.

ArabSoft combines **technological expertise** and **business understanding** to deliver reliable and scalable solutions adapted to client needs. With a structured project approach and multidisciplinary teams, the company supports organizations in optimizing internal workflows, improving traceability, and increasing overall operational efficiency.

Figure 1: ArabSoft logo

2. ArabSoft in Tunisia

In Tunisia, **ArabSoft** supports both public and private organizations in the digitalization of their internal processes. Its teams work on the design, development, deployment, and maintenance of business-oriented platforms, with a focus on security, scalability, and long-term

maintainability. Key missions include process automation, workflow management, data centralization, and decision support through structured reporting.

II. Project presentation

1. Current-state study

At present, internal process management at ArabSoft relies on a combination of office tools and partially connected workflows. Daily practices include Excel files, email exchanges, messaging channels, and manual follow-up sheets to handle administrative requests and monitor project execution.

Although these tools support day-to-day operations, information remains distributed across multiple sources and teams. This fragmentation limits process visibility, slows coordination, and increases the risk of inconsistencies between request status, project progress, and assigned tasks.

The current operational flow mainly covers:

- **Administrative request handling**, to receive, validate, and process internal service requests.
- **Project and task follow-up**, to plan activities, assign responsibilities, and track execution progress.
- **Operational coordination**, to synchronize information between employees, team leaders, and HR managers and ensure timely communication.

2. Current-state critique

Despite the use of multiple tools, several difficulties remain and reduce operational efficiency. Main issues include:

- **Data fragmentation:** Request, project, and task information is spread across multiple files and channels, which makes a coherent global view difficult.
- **Delayed follow-up:** Manual tracking slows the handling of requests and makes it harder to monitor priorities, deadlines, and responsibilities.

- **Inefficient coordination:** Employees, team leaders, and HR managers do not share a single operational workspace, which increases duplication and communication gaps.
- **Limited access to updated information:** Stakeholders do not always have real-time access to the latest status, which reduces responsiveness.

These limitations show that the current tools and processes no longer fully meet ArabSoft's needs for traceability, responsiveness, and coordination in the management of internal requests, projects, and tasks.

3. Proposed solution

To address these challenges, I propose the development of an **internal platform for managing requests, projects, and tasks at ArabSoft**. The platform is designed to centralize operational data, structure workflow execution, and improve coordination across departments through a practical and intuitive environment. Key features include:

- **Data centralization:** All requests, projects, and tasks are stored in a unified database accessible in real time.
- **Workflow management:** Structured request lifecycle with clear statuses, validation steps, and processing history.
- **Project and task tracking:** Planning, assignment, prioritization, and progress monitoring through dedicated views.
- **Role-based access:** Secure access control according to user profiles and organizational responsibilities for employees, team leaders, and HR managers.
- **Notifications and integrations:** Alerts and automated exchanges with external services when needed to support authentication, messaging, and reporting.
- **Dashboards and reporting:** Operational indicators for monitoring performance, identifying bottlenecks, and supporting decision-making.

This solution provides a global real-time view of internal activities and improves efficiency, responsiveness, and traceability across the organization.

Modular, secure, and scalable, the platform is designed to adapt to ArabSoft's evolving needs while enabling progressive adoption and continuous process improvement.

III. Adopted methodologies and formalism

1. Modeling and design methodology

For project modeling and design, I chose **UML (Unified Modeling Language)**. UML is a standardized visual language used to represent both system structure and behavior. [1]

Using UML in this project allows me to:

- **Visualize requirements:** Use-case diagrams capture expected functionalities and actor-system interactions.
- **Structure the system:** Class and sequence diagrams define module architecture and data/interaction flows.
- **Facilitate communication:** UML provides a common language for developers, project managers, and clients.

UML was selected for its flexibility, standardization, and ability to fit complex project needs.

Figure 2: UML logo

2. Working methodology

For project management, I adopted the **Agile** methodology, specifically the **Scrum** framework. This iterative and incremental approach supports requirement changes and continuous user feedback. [2]

2.1. Methodological choice (Scrum)

Scrum was chosen for several reasons:

- **Flexibility:** Scrum adapts quickly to change, which is critical when requirements evolve.
- **Transparency:** Regular meetings such as daily stand-ups and sprint reviews ensure clear communication.
- **Continuous delivery:** Sprint cycles provide continuous validation of implemented features.

2.2. Why Scrum?

Scrum is well suited to this project because it supports multidisciplinary teams and collaboration. It also helps to:

- **Maximize value:** Focus on high-priority features aligned with project objectives.
- **Improve quality:** Frequent feedback and regular testing enable rapid issue detection and correction.

2.3. Scrum team roles

In this project, Scrum roles are distributed as follows:

- **Product Owner:** Defines priorities and manages the backlog to ensure delivered features meet user needs. In this project, the Product Owner is **Mr. Mouadh Zidi**, my company supervisor.
- **Scrum Master:** Ensures Scrum principles are followed and helps remove obstacles. In this project, the Scrum Master is **Ms. Anissa CHALOUAH**, my ISET supervisor.
- **Development Team:** Responsible for design, development, and testing of platform features. In this project, the development team is composed of **Abderrahmen Ben Hattab** only.

This role distribution enables efficient collaboration and clear responsibilities, contributing directly to project success.

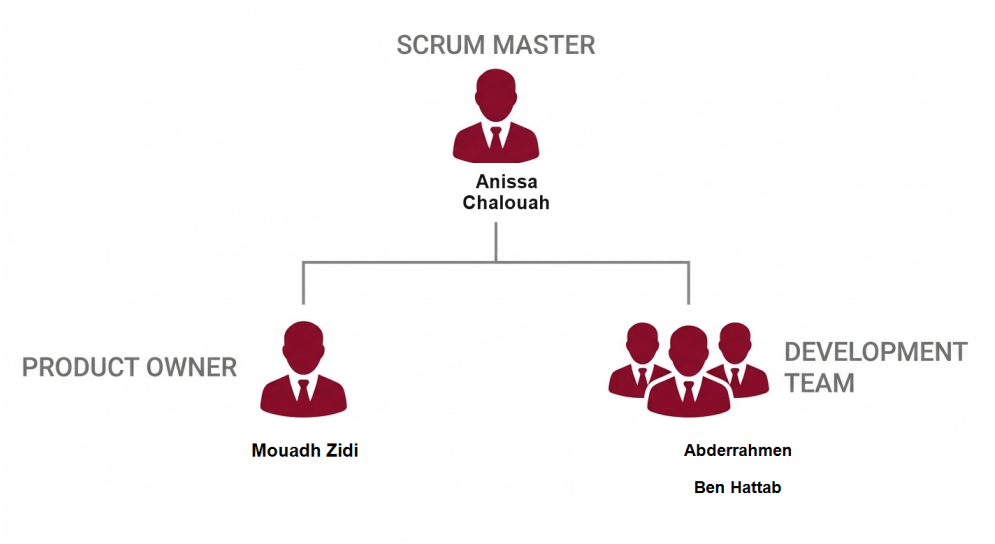


Figure 3: Scrum team roles

IV. Development environment

To complete this project, an appropriate development environment was set up. It includes hardware resources, software tools, and modern technologies to ensure effective design, development, and deployment.

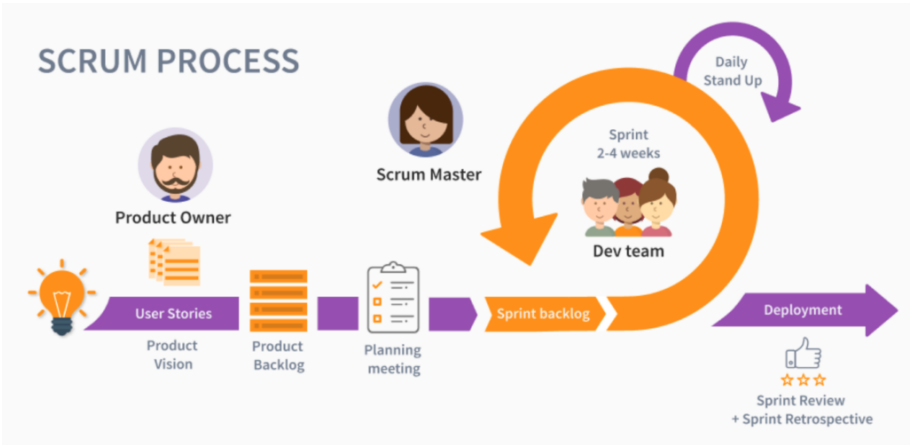
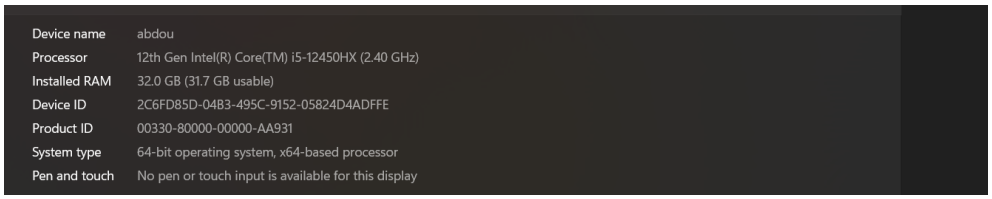


Figure 4: SCRUM method

1. Hardware environment

This project was developed on a laptop with the following specifications:



Device name	abdou
Processor	12th Gen Intel(R) Core(TM) i5-12450HX (2.40 GHz)
Installed RAM	32.0 GB (31.7 GB usable)
Device ID	2C6FD85D-04B3-495C-9152-05824D4ADFFE
Product ID	00330-80000-00000-AA931
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Figure 5: PC specifications used for development

2. Software environment

For developing the internal request, project, and task management platform, several software tools were used. Each tool was selected for its specific features and its fit with project requirements.

- **Visual Studio Code :**

Main code editor used for **frontend** development with **React** and **Tailwind CSS**. Its lightweight design and extension ecosystem provide a productive environment for UI implementation. [3]

- **IntelliJ IDEA :**

Integrated development environment used for **backend** development with **Spring Boot** and **PostgreSQL**. IntelliJ IDEA provides advanced autocompletion, debugging, refactoring, and native integration with Maven and Spring. [4]

- **Postman :**

Used to test and validate REST APIs developed with Spring Boot through an intuitive interface and strong debugging and documentation features. [5]

- **Draw.io :**

Used to design UML diagrams such as use-case, class, and sequence diagrams, as well as functional mockups, thanks to its ease of use and free access. [6]

- **Git and GitHub :**

Used for version control and source-code management, with reliable workflows and easy integration in Visual Studio Code and IntelliJ IDEA.[7, 8]

- **Jira :**

Used for task management and Scrum sprint tracking. It supports user-story planning, sprint monitoring, and progress reporting. [9]

- **LaTeX :**

LaTeX was selected for report writing to ensure a professional layout, accurate references, and typographic consistency throughout the document. [10]

3. Technologies used

The following technologies were chosen for their fit with project requirements and their ability to deliver a modern, high-performance solution.

- **React :**

Main frontend library used in the View layer of the platform interface. React provides reusable components, efficient rendering, and flexible state management for scalable user interfaces. [11]

- **Tailwind CSS :**

Utility-first CSS framework used for interface styling and layout. Tailwind CSS enables fast, consistent, and responsive UI development while supporting design customization. [12]

- **Spring Boot :**

Java framework used for **backend** development based on a **layered architecture**. Spring Boot simplifies secure and high-performance REST API development through auto-configuration and a rich ecosystem. [13]

- **PostgreSQL :**

Relational database management system used for platform data storage and management. PostgreSQL was selected for reliability, performance, and strong support for complex queries and data integrity. [14]

- **Keycloak :**

Identity and access management solution used to handle secure authentication, role-based authorization, and token-based security for the platform. [15]

- **Docker :**

Containerization platform used to package and run application services consistently across development and deployment environments. [16]

- **Kafka :**

Distributed event-streaming platform used for asynchronous communication, especially for notifications and background event processing in the system. [17]

V. Application architecture

1. Global architecture

The application follows a web-based architecture composed of two complementary parts: a **React frontend structured with MVVM** and a **layered** backend. This organization fits an internal platform used by employees, team leaders, and HR managers to manage requests, projects, tasks, notifications, and dashboards. It ensures clear separation of responsibilities, smooth integration through REST APIs, and maintainable system evolution. External services such as Keycloak, Kafka, Apache POI for Excel export, and email notifications are integrated when needed to support authentication, event handling, reporting, and communication. [11, 13, 15, 17, 19]

2. Frontend part: MVVM architecture with React

The frontend part is implemented with **React** and **Tailwind CSS** and organized according to the **MVVM** pattern. This choice is appropriate for a role-aware internal platform where employees, team leaders, and HR managers access different views over the same operational data. React provides the View layer, view models coordinate presentation logic and state, and service classes handle communication with the Spring Boot REST API. This organization ensures a clear separation between interface rendering, presentation behavior, and service communication, which improves maintainability and functional consistency. [11, 12]

- **View layer:** Contains React pages and reusable UI components used to display

request forms, task boards, project dashboards, notification panels, and role-specific interfaces.

- **ViewModel layer:** Handles presentation logic, state coordination, form behavior, interaction flows, and the transformation of backend data into structures adapted to the interface.
- **Model / Service layer:** Groups frontend data models, DTO-shaped structures when needed, and service classes or functions responsible for calling the backend API and managing request and response data.

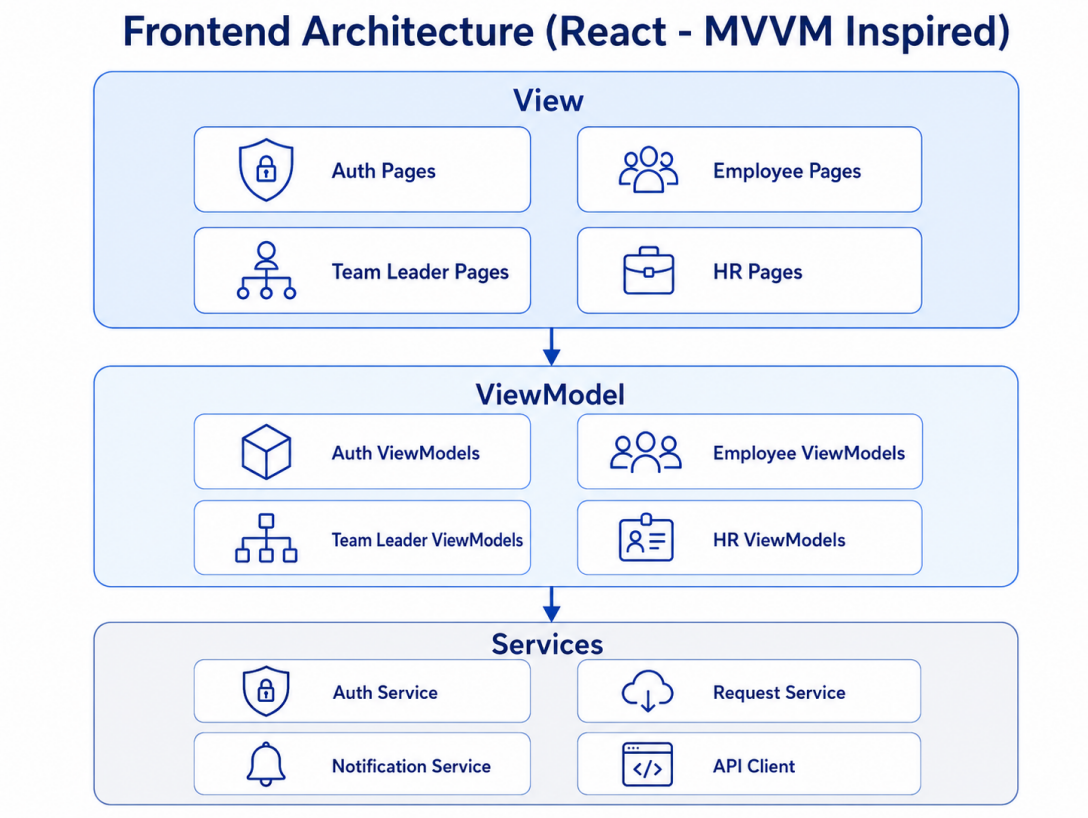


Figure 6: MVVM frontend architecture with React

3. Backend part: layered architecture with Spring Boot

The backend is based on a **layered architecture**, implemented with **Spring Boot** and **PostgreSQL**. This architecture structures the backend into clear functional layers, improving maintainability, testability, and scalability. [13, 14]

- **Controller layer:** Spring MVC REST controllers expose API endpoints and handle

HTTP requests from the React frontend.

- **Service layer:** Contains business logic, validation rules, and orchestration of application use cases.
- **Repository layer:** Spring Data JPA repositories handle persistence and data access in **PostgreSQL**.

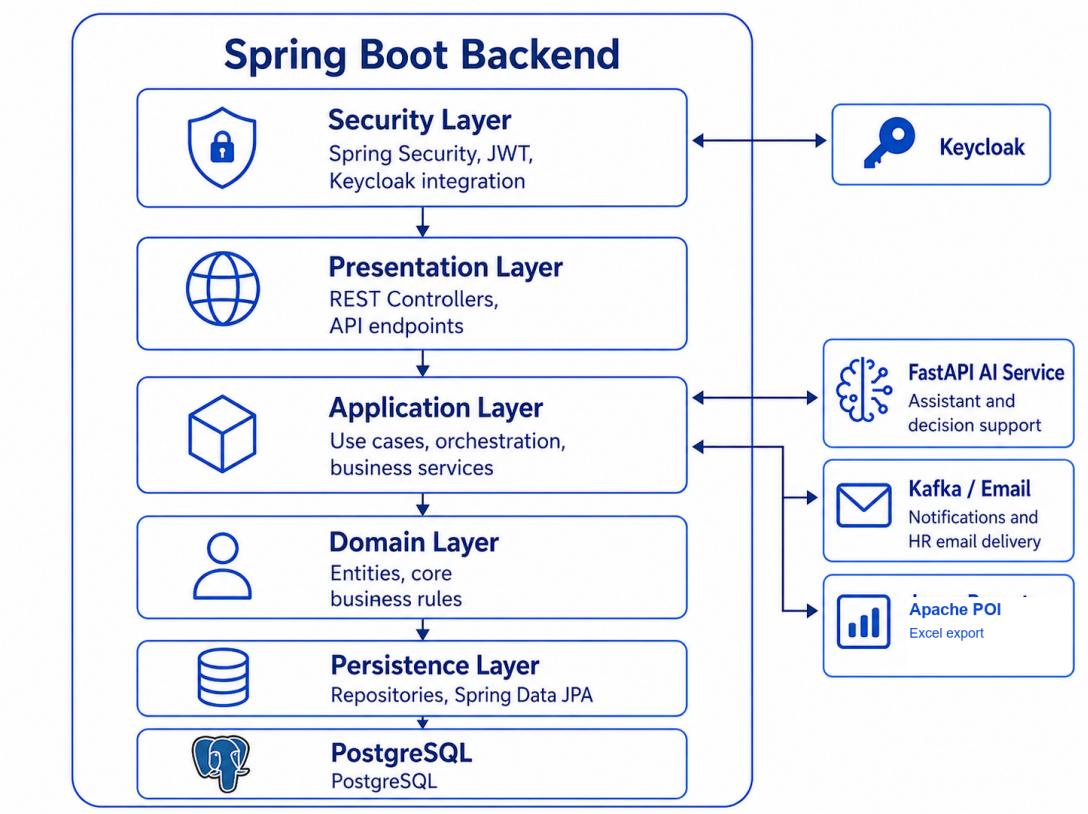


Figure 7: Layered architecture with Spring Boot

4. Why these architectures?

These architectures were selected for several reasons:

- **Separation of concerns:** MVVM separates the interface, presentation logic, and service access in the frontend, while the backend keeps controllers, services, and repositories clearly isolated.
- **Maintainability and testability:** View models can be tested and evolved independently from React pages, and backend service logic remains independent from persistence concerns.

- **Reusability and consistency:** A structured frontend makes it easier to reuse UI components and shared presentation logic across requests, projects, tasks, dashboards, and notifications.
- **Scalability and integration:** React communicates efficiently with the Spring Boot REST API, which supports progressive extension of the platform without compromising organization.

5. Global architecture illustration

The figure below presents the global application architecture in a cleaner and more report-ready form. It shows the React frontend structured with MVVM, the Spring Boot backend organized in layers, PostgreSQL as the persistent storage, and the external services integrated to support authentication, asynchronous communication, reporting, and notifications. The diagram also highlights the role-based usage of the platform by employees, team leaders, and HR managers.

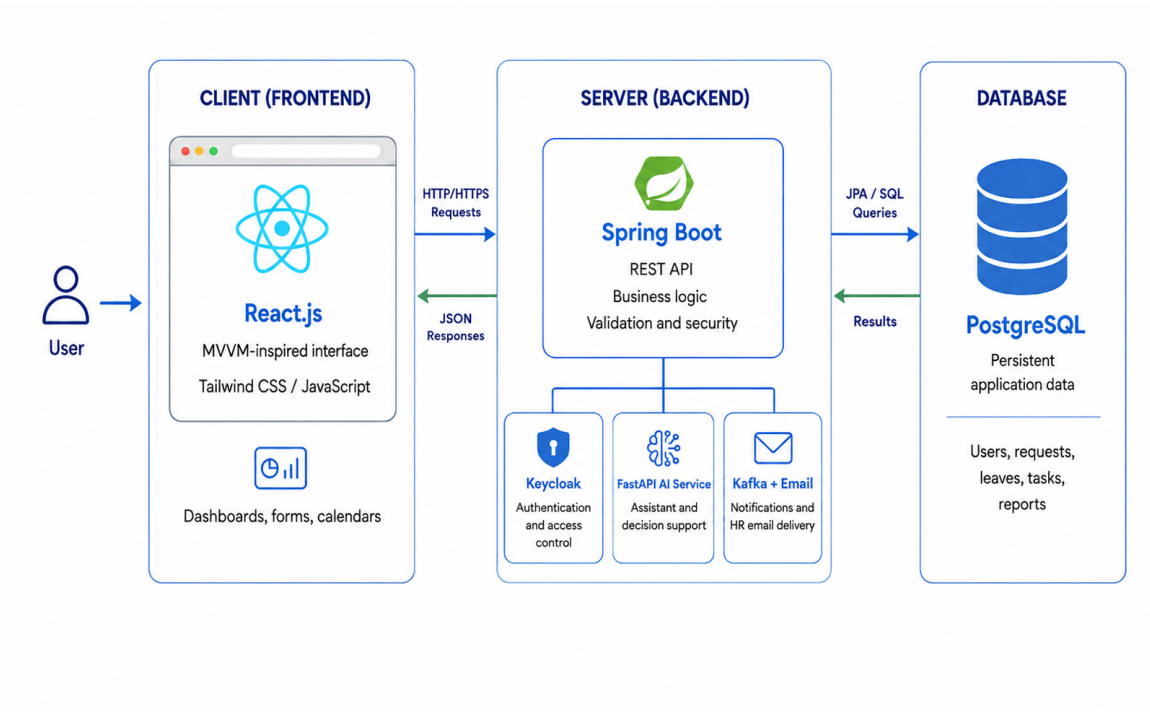


Figure 8: Global application architecture

Conclusion

This chapter presented the general project context: the host organization ArabSoft, the current internal management challenges related to requests, projects, and tasks, the proposed solution, and the adopted methodologies and development environment. These elements provide the foundation for the rest of the work. The next chapter focuses on product backlog planning and functional requirement identification.

Chapter 2

Product Backlog Planning

Introduction

This chapter defines the product backlog planning for the proposed internal platform at ArabSoft. It identifies the actors, structures the functional and non-functional requirements, and formalizes user stories and backlog priorities that will guide sprint planning and implementation.

I. Actor Identification

1. Employee

The employee submits administrative requests, tracks processing status, accesses assigned tasks, and receives notifications related to approvals, rejections, and project activities.

2. Team Leader

The team leader supervises team activities, validates requests requiring managerial approval, assigns and follows tasks, and monitors project execution indicators.

3. HR Manager

The HR manager manages employee records and roles, performs final validation of administrative requests, and monitors compliance through dashboards and reports.

4. New User

A new user is registered in the platform but remains in a restricted state until HR validation and role assignment are completed.

II. Functional Requirements

1. Authentication and access management

The system shall support registration, secure authentication, logout, account activation by HR, and role-based access control according to user profile.

2. Administrative request management

The platform shall manage submission, validation, rejection, and history tracking for leave requests, document requests, loan requests, and authorization requests.

3. Team and employee management

The system shall provide profile management, team assignment, leader designation, and maintenance of employee administrative information.

4. Project and task management

The platform shall support project creation, task assignment, workload organization, progress tracking, and status visibility for all authorized stakeholders.

5. Notification and reporting

The system shall send operational notifications and provide reports and dashboards to support monitoring, traceability, and decision-making.

6. AI assistance and decision support

The platform shall provide AI-assisted support to help employees prepare administrative requests, assist team leaders in reviewing requests and workload constraints, and support HR managers with recommendation-oriented decision assistance while preserving human validation, traceability, and auditability.

III. Non-Functional Requirements

1. Security

The application must guarantee secure authentication, protection of sensitive data, strict role-based permissions, and complete audit traceability.

2. Performance

The platform must maintain acceptable response times under expected internal usage and support concurrent request processing without significant delays.

3. Usability

Interfaces must be clear and intuitive to reduce training effort and facilitate adoption by employees, team leaders, and HR managers.

4. Maintainability

The system architecture must allow efficient correction, enhancement, and extension through modular and well-documented components.

5. Scalability

The platform should support future growth in users, workflows, and modules without requiring major redesign.

IV. User Stories

1. Employee User Stories

Employee User Stories

As an employee, I want to sign in securely so that I can access the platform according to my role.	As an employee, I want to sign out securely so that I can protect my account after use.	As an employee, I want to manage my profile information so that my personal data stays accurate.
As an employee, I want to change my password securely so that I can keep my account protected.	As an employee, I want to access my personal dashboard so that I can quickly view the most relevant information.	As an employee, I want to submit and cancel a leave request before decision so that my leave workflow stays accurate.
As an employee, I want to submit a document request so that I can obtain official HR documents.	As an employee, I want to submit a loan request so that I can ask for financial support through the platform.	As an employee, I want to submit an authorization request so that I can formalize exceptional permissions.

As an employee, I want to track request status and history so that I can follow leave and administrative workflows.

As an employee, I want to view my assigned tasks and project calendar so that I can organize my work efficiently.

As an employee, I want to filter tasks by project so that I can focus on the work context concerned.

As an employee, I want to receive notifications and emails about role, request, leave, and task updates so that I stay informed.

2. Team Leader User Stories

Team Leader User Stories

As a team leader, I want to sign in securely so that I can access the platform according to my role.

As a team leader, I want to sign out securely so that I can protect my account after use.

As a team leader, I want to manage my profile information so that my personal data stays accurate.

As a team leader, I want to change my password securely so that I can keep my account protected.

As a team leader, I want to access my dashboard so that I can quickly monitor team activity.

As a team leader, I want to review team leave requests so that I can process them according to team constraints.

As a team leader, I want to approve or reject leave requests with a reason so that decisions remain clear and traceable.

As a team leader, I want to monitor team workload so that I can balance responsibilities effectively.

As a team leader, I want to assign tasks to team members so that project work is distributed properly.

As a team leader, I want to follow task progress and receive completion notifications so that I can monitor delivery.

As a team leader, I want to view the team leave calendar within my scope so that I can anticipate availability issues.

As a team leader, I want to use employee-side leave and request actions for my own needs so that personal workflows remain consistent.

3. HR Manager User Stories

HR Manager User Stories

As an HR manager, I want to sign in securely so that I can access the platform according to my role.

As an HR manager, I want to sign out securely so that I can protect my account after use.

As an HR manager, I want to manage my profile information so that my personal data stays accurate.

As an HR manager, I want to change my password securely so that I can keep my account protected.

As an HR manager, I want to access my dashboard so that I can monitor operational activity.

As an HR manager, I want to assign roles and trigger user notification so that access changes are communicated clearly.

As an HR manager, I want to activate or deactivate accounts so that access remains controlled.

As an HR manager, I want to manage employee administrative information so that records stay complete and accurate.

As an HR manager, I want to make final decisions on leave and administrative requests so that workflows are completed properly.

As an HR manager, I want to manage teams and team leaders so that the organizational structure remains consistent.

As an HR manager, I want broad leave calendar visibility and filters so that I can supervise teams, employees, dates, and statuses.

As an HR manager, I want to review pending user accounts so that I can complete their access setup.

As an HR manager, I want to upload official request files and notify users so that final HR outputs are communicated.

As an HR manager, I want to trace request history and cancellations before decision so that workflow follow-up remains clear.

V. Product Backlog

The product backlog is organized into six functional sprints. Each sprint groups a coherent set of user stories and technical tasks according to business value, implementation dependencies, and the progressive delivery of the platform. The backlog was tracked through Jira and structured around access management, leave workflows, administrative requests, HR operations, collaboration features, reporting, notifications, and AI-assisted support.

Table 1: Product Backlog

Sprint	Num	User Story / Task	Description	Complexity
Sprint 1 Access and Account Management	**US 01**	**User registration**	As a new user, I want to create an account so that I can request access to the platform.	Medium
	US 02	**Secure authentication**	As a user, I want to sign in securely so that I can access features according to my role.	Medium
	US 03	**Pending access state**	As a new user, I want to see a pending-access page until HR assigns my role.	Low
	US 04	**Password reset**	As a user, I want to reset my password so that I can recover access to my account.	Medium
	US 05	**Profile update**	As a user, I want to update my profile information so that my personal data remains accurate.	Low
	US 06	**Role assignment**	As an HR manager, I want to assign roles so that users receive the correct access rights.	Medium
	US 07	**Account control**	As an HR manager, I want to activate or deactivate accounts so that platform access remains controlled.	Medium
	US 08	**Pending user review**	As an HR manager, I want to review pending user accounts so that I can complete their access setup.	Medium
	TS 01	**Security configuration**	Configure Keycloak, PostgreSQL, and Spring Security for secure role-based access.	High

Sprint	Num	User Story / Task	Description	Complexity
	TS 02	**Role notification**	Send notification and email after role assignment.	Low
Sprint 2 Leave Management	**US 09**	**Leave submission**	As an employee, I want to submit a leave request so that I can ask for time off.	Medium
	US 10	**Leave balance and history**	As an employee, I want to view my leave balance and leave history.	Medium
	US 11	**Working-day validation**	As an employee, I want the system to validate leave dates using working days and public holidays.	High
	US 12	**Overlap prevention**	As an employee, I want the system to prevent overlapping leave requests.	High
	US 13	**Edit and cancellation**	As an employee, I want to edit or cancel a leave request before final decision.	Medium
	US 14	**Team leader decision**	As a team leader, I want to approve or reject team leave requests.	Medium
	US 15	**HR final decision**	As an HR manager, I want to make the final decision on leave requests.	High
	US 16	**Leave calendar**	As a user, I want to view leave requests in a calendar.	Medium
	US 17	**Leave filters**	As an HR manager, I want to filter leave records by employee, status, and date.	Low
	TS 03	**Leave notifications**	Send leave decision notifications and emails.	Low
	TS 04	**Leave reports**	Generate leave approval reports.	Medium
Sprint 3 Administrative Requests	**US 18**	**Document request**	As an employee, I want to submit a document request so that I can obtain official HR documents.	Medium
	US 19	**Document processing**	As an HR manager, I want to process document requests and upload final files.	Medium
	US 20	**Loan request**	As an employee, I want to submit a loan request so that I can request financial support.	Medium
	US 21	**Loan eligibility**	As an HR manager, I want to check loan eligibility before making a decision.	High
	US 22	**Loan meeting**	As an HR manager, I want to schedule a loan meeting with valid time slots.	High

Sprint	Num	User Story / Task	Description	Complexity
	US 23	**Authorization request**	As an employee, I want to submit an authorization request for exceptional permission.	Medium
	US 24	**HR request decision**	As an HR manager, I want to approve or reject administrative requests.	High
	US 25	**Request tracking**	As a user, I want to track the status and history of my administrative requests.	Low
	US 26	**Request filters**	As an HR manager, I want to search and filter administrative requests.	Low
	TS 05	**Request notifications**	Send request status notifications and emails.	Low
	TS 06	**Final files**	Manage final request files and downloads.	Medium
Sprint 4 HR Management and Calendar	**US 27**	**Employee records**	As an HR manager, I want to manage employee records.	Medium
	US 28	**Department management**	As an HR manager, I want to manage departments.	Medium
	US 29	**Job title management**	As an HR manager, I want to manage job titles.	Medium
	US 30	**Team management**	As an HR manager, I want to create teams and assign team leaders.	High
	US 31	**Employee movement**	As an HR manager, I want to move employees between teams.	Medium
	US 32	**User search and filters**	As an HR manager, I want to search and filter users.	Low
	US 33	**HR dashboard**	As an HR manager, I want to view HR dashboard indicators.	Medium
	US 34	**HR calendar**	As an HR manager, I want to supervise leave planning through the HR calendar.	Medium
	US 35	**Team calendar**	As a team leader, I want to view my team calendar to understand availability.	Medium
	TS 07	**Calendar filters**	Add calendar filters by employee, status, and date.	Low
	TS 08	**HR approval views**	Improve HR request and approval views.	Medium
	US 36	**Project management**	As a team leader, I want to create and manage projects.	Medium

Sprint	Num	User Story / Task	Description	Complexity
Sprint 5 Projects, Tasks, Reports	**US 37**	**Task assignment**	As a team leader, I want to create and assign tasks to team members.	Medium
	US 38	**Task status update**	As an assignee, I want to update task status so that progress remains visible.	Low
	US 39	**Task filters**	As a user, I want to filter tasks by project and status.	Low
	US 40	**Availability check**	As a team leader, I want to check availability before assigning work.	High
	US 41	**Excel report export**	As an HR manager, I want to export filtered reports to Excel so that reporting data can be reviewed and archived outside the platform.	Medium
	US 42	**Reports and statistics**	As an HR manager, I want to monitor report indicators and statistics so that I can follow operational activity.	Medium
	TS 09	**Report data normalization**	Normalize project, task, user, team, and request data for consistent report filtering and aggregation.	High
	TS 10	**Excel export backend**	Implement the backend report export service and generate structured Excel workbooks.	High
Sprint 6 AI, Deployment and Stabilization	**US 43**	**Platform questions**	As a user, I want to ask platform questions using the assistant.	Medium
	US 44	**Role-aware answers**	As a user, I want the assistant to answer according to my role.	High
	US 45	**Leave draft assistance**	As an employee, I want the assistant to help draft a leave request.	High
	US 46	**Draft validation**	As an employee, I want the system to validate the leave draft before submission.	High
	US 47	**Explicit confirmation**	As an employee, I want leave submission to require explicit confirmation.	Medium
	US 48	**Unsafe-action refusal**	As a user, I want the assistant to refuse unsafe actions.	Medium
	TS 11	**React assistant integration**	Integrate the React chatbot with the Spring Boot assistant gateway.	High
	TS 12	**FastAPI integration**	Connect the Spring Boot assistant gateway with the FastAPI AI service.	High

Sprint	Num	User Story / Task	Description	Complexity
	TS 13	**Demo deployment preparation**	Prepare local demonstration deployment configuration.	Medium
	TS 14	**Final build checks**	Run final frontend, backend, and AI service build checks.	Medium
	TS 15	**Final stabilization**	Stabilize the platform and correct final integration issues.	High

VI. Release Organization

The six planned sprints are grouped into three major releases to ensure progressive and coherent delivery of the platform.

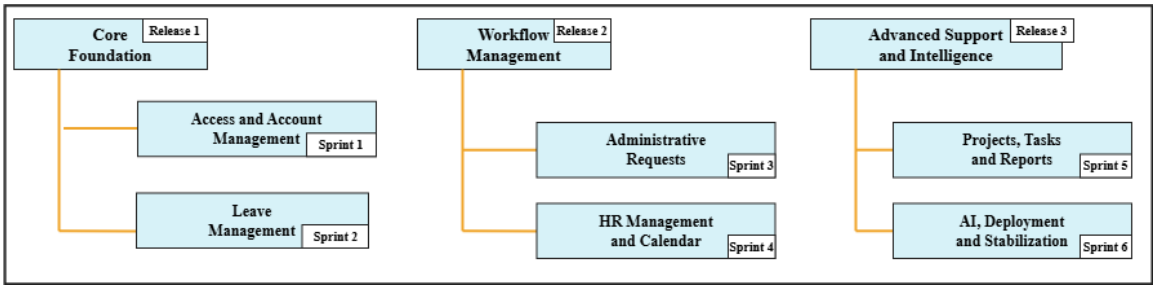


Figure 9: Organization of Releases

Conclusion

This chapter established the backlog-planning foundation of the project through actor identification, requirement structuring, user stories, and prioritized backlog definition. It provides the planning baseline for sprint execution and the progressive realization of the platform at ArabSoft.

Chapter 3

Release 1 – Foundations

Introduction

This chapter focuses on Release 1, which forms the functional and technical foundation of the ArabSoft internal platform. Through its use case diagrams, system sequence diagrams, state diagrams, class diagrams, and interface illustrations, this chapter presents how the first core functionalities of the platform are structured. It highlights the implementation logic of secure access management and the first complete administrative workflow. More specifically, Release 1 covers user registration, authentication, role assignment, account control, and leave request management. Together, these elements establish the first operational base of the platform for employees, team leaders, and HR managers.

I. Release Organization

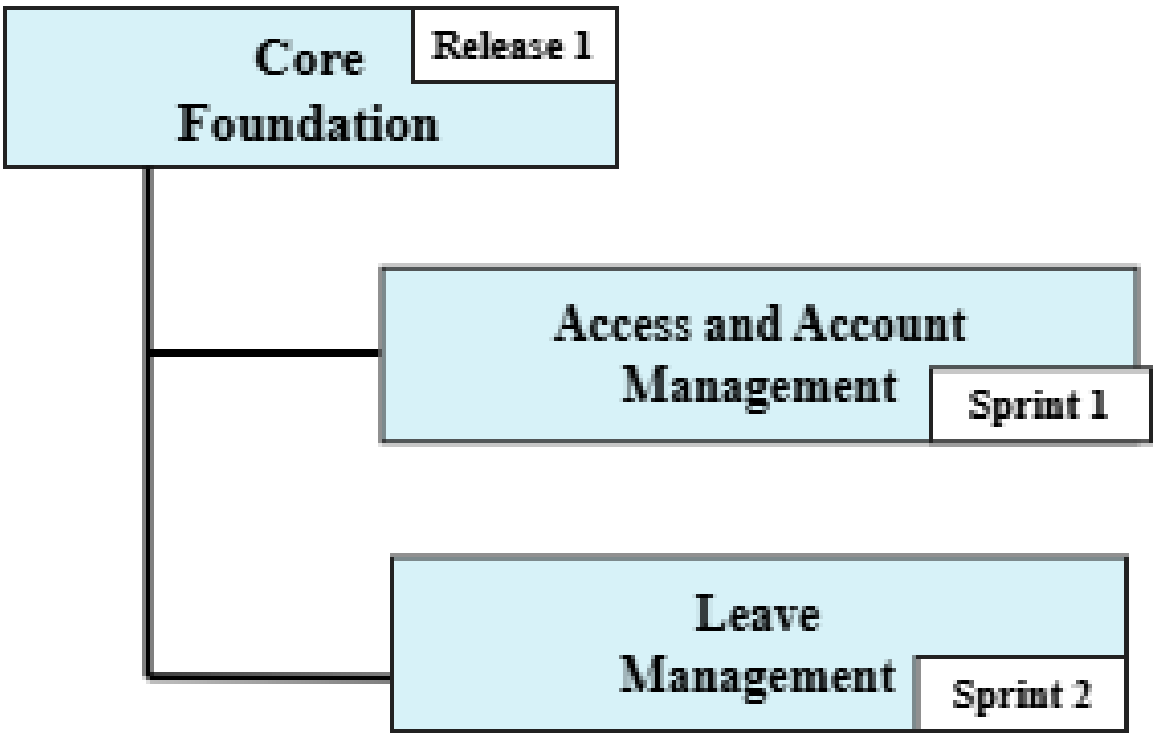


Figure 10: Release 1 Organization

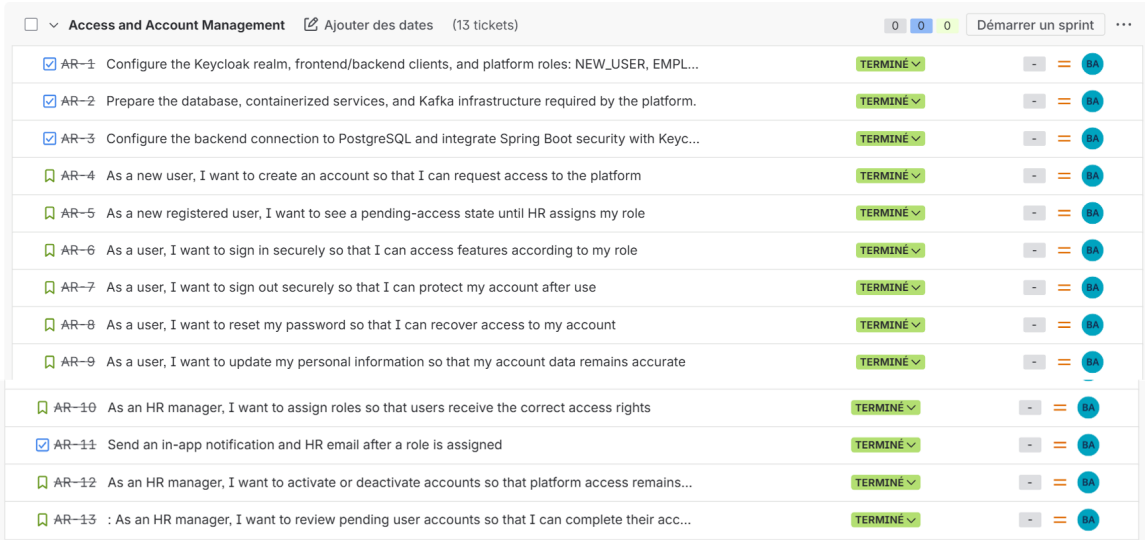
II. Sprint 1: Access and Account Management

1. Sprint Goal

This first sprint aims to establish the technical and functional access foundation of the platform. It implements the core access features: user registration, secure authentication, password reset, HR user account setup, role assignment with notification and email, account activation and deactivation, profile update, and access control. It also introduces the restricted **NEW_USER** state to control onboarding before HR validation. The sprint is planned as a two-week iteration. The objective is to obtain an operational base for the next functional developments of the platform.

2. Visual Sprint Journal

The visual sprint journal below presents the selected user stories of Sprint 1 and their completion status in Jira.



Access and Account Management	Ajouter des dates	(13 tickets)	0 0 0	Démarrer un sprint
AR-1	Configure the Keycloak realm, frontend/backend clients, and platform roles: NEW_USER, EMPL...	TERMINÉ	-	BA
AR-2	Prepare the database, containerized services, and Kafka infrastructure required by the platform.	TERMINÉ	-	BA
AR-3	Configure the backend connection to PostgreSQL and integrate Spring Boot security with Keyc...	TERMINÉ	-	BA
AR-4	As a new user, I want to create an account so that I can request access to the platform	TERMINÉ	-	BA
AR-5	As a new registered user, I want to see a pending-access state until HR assigns my role	TERMINÉ	-	BA
AR-6	As a user, I want to sign in securely so that I can access features according to my role	TERMINÉ	-	BA
AR-7	As a user, I want to sign out securely so that I can protect my account after use	TERMINÉ	-	BA
AR-8	As a user, I want to reset my password so that I can recover access to my account	TERMINÉ	-	BA
AR-9	As a user, I want to update my personal information so that my account data remains accurate	TERMINÉ	-	BA
AR-10	As an HR manager, I want to assign roles so that users receive the correct access rights	TERMINÉ	-	BA
AR-11	Send an in-app notification and HR email after a role is assigned	TERMINÉ	-	BA
AR-12	As an HR manager, I want to activate or deactivate accounts so that platform access remains...	TERMINÉ	-	BA
AR-13	: As an HR manager, I want to review pending user accounts so that I can complete their acc...	TERMINÉ	-	BA

Visual Sprint Journal – Sprint 1

3. Sprint Backlog

The table below presents the backlog of the first sprint, planned as a two-week iteration.

Sprint	Stories	Tasks	Duration
Sprint 1	**Configure security and infrastructure**	Configure Keycloak realm, client, and roles	2d
		Set up PostgreSQL, Docker, and Kafka	
		Connect Spring Boot to PostgreSQL and Keycloak	
	Registration and access control	Implement user registration	3d
		Enforce the restricted **NEW_USER** state	
		Implement secure login	
		Implement frontend logout	
	Password recovery and profile management	Implement password reset	2d
		Implement profile update	
	HR account governance	Implement HR role assignment	2d
		Trigger role assignment notification and email	
		Implement account activation and deactivation	
	HR account setup validation	Validate pending users and complete their access setup	1d

Table 2: Sprint Backlog – Sprint 1

4. Sprint Implementation

Requirements Analysis

Sprint 1 covers the access and account management foundation, including registration, authentication, password recovery, profile management, HR account setup, and account status control.

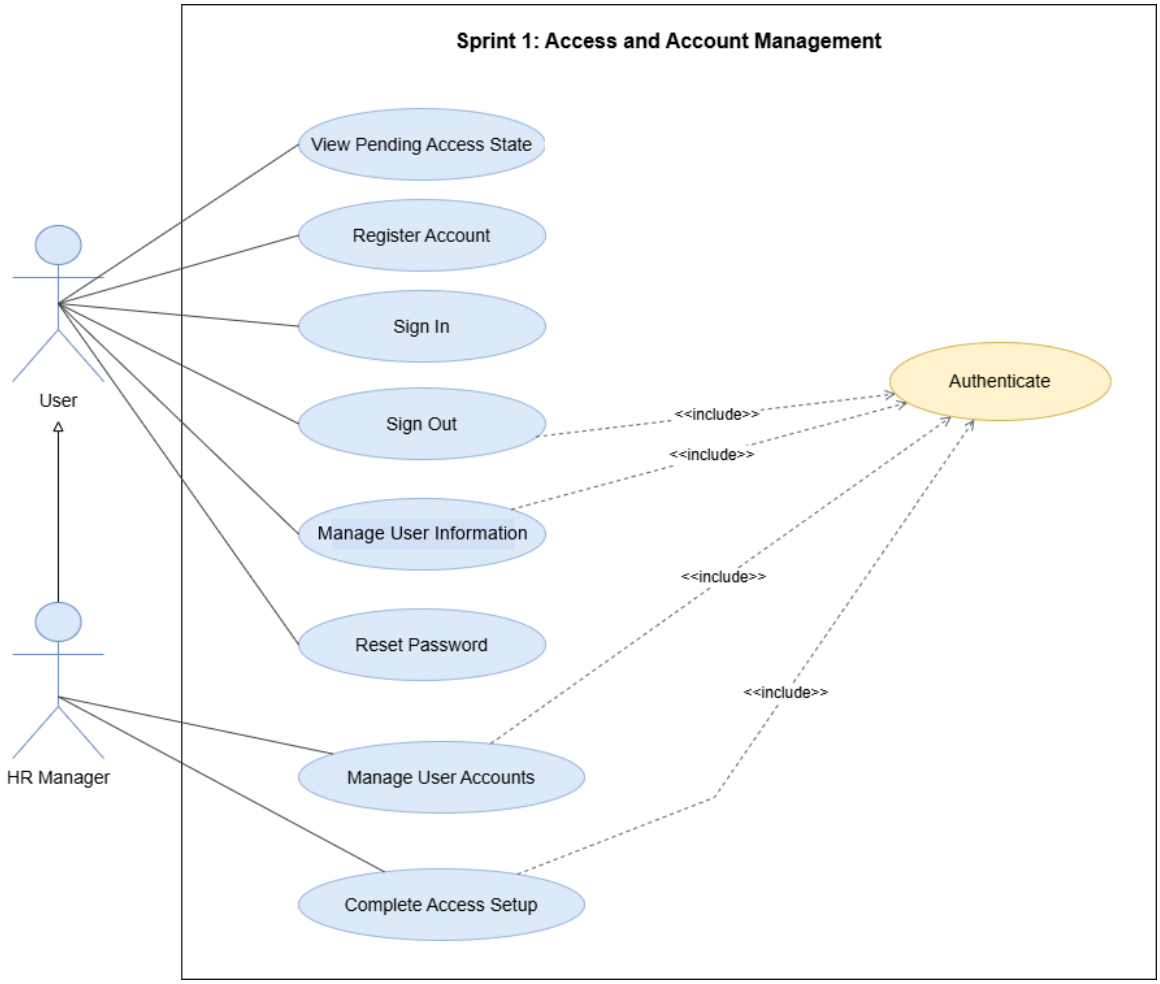


Figure 11: Use Case Diagram – Sprint 1 Access and Account Management

System Sequence Diagrams

These diagrams summarize the main user-system exchanges for Sprint 1 access scenarios.

User Registration

This diagram shows how a visitor registers and remains pending until HR validation.

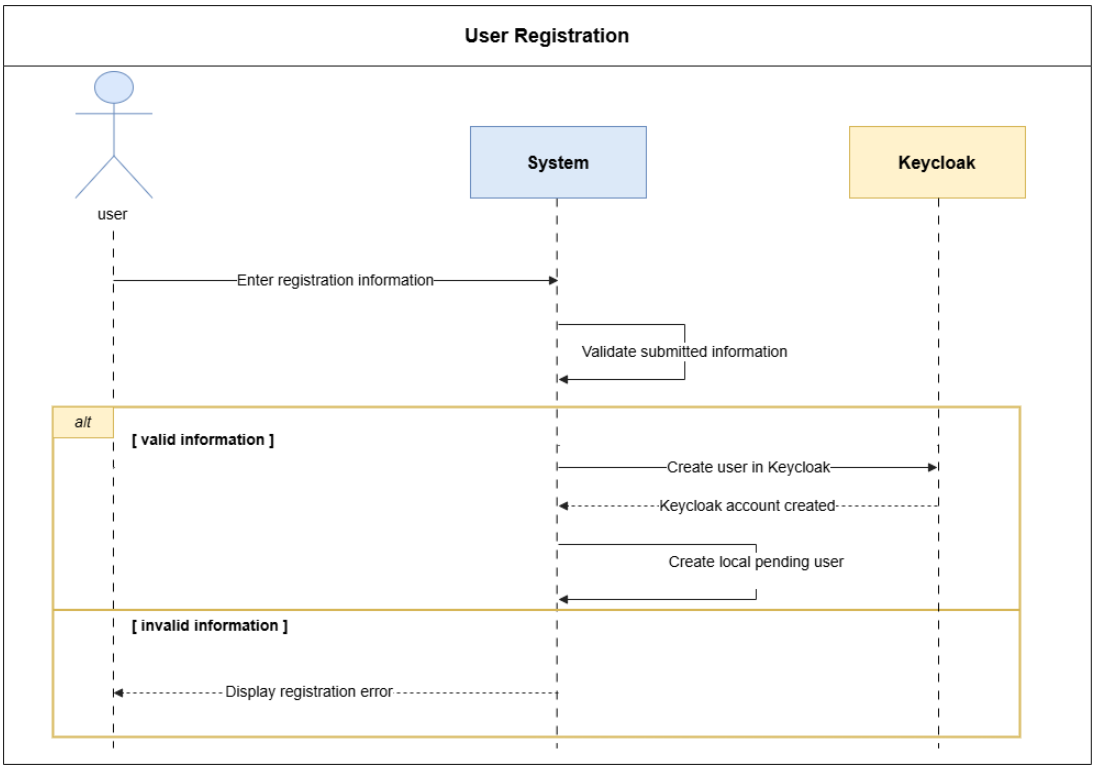


Figure 12: System Sequence Diagram – User Registration

Secure Login

This diagram shows credential validation through Keycloak and role-based redirection.

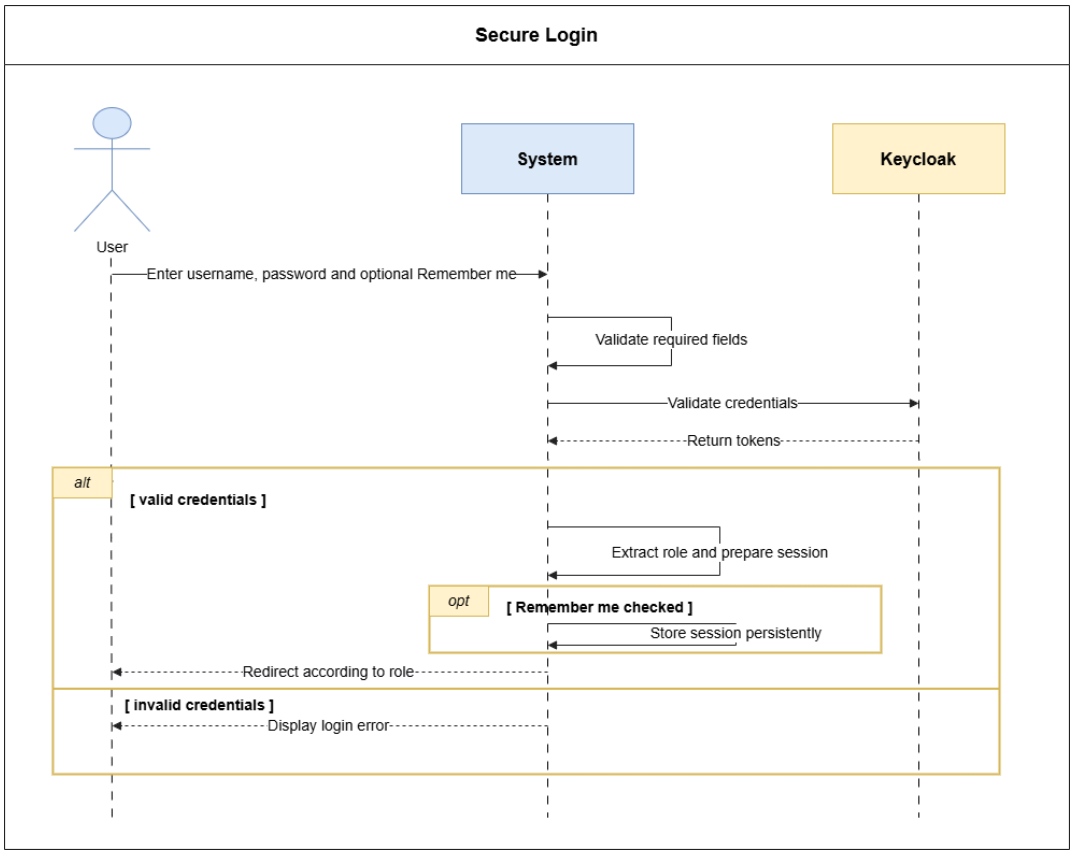


Figure 13: System Sequence Diagram – Secure Login

Logout

This diagram shows session cleanup, optional Keycloak logout, and redirection to the login page.

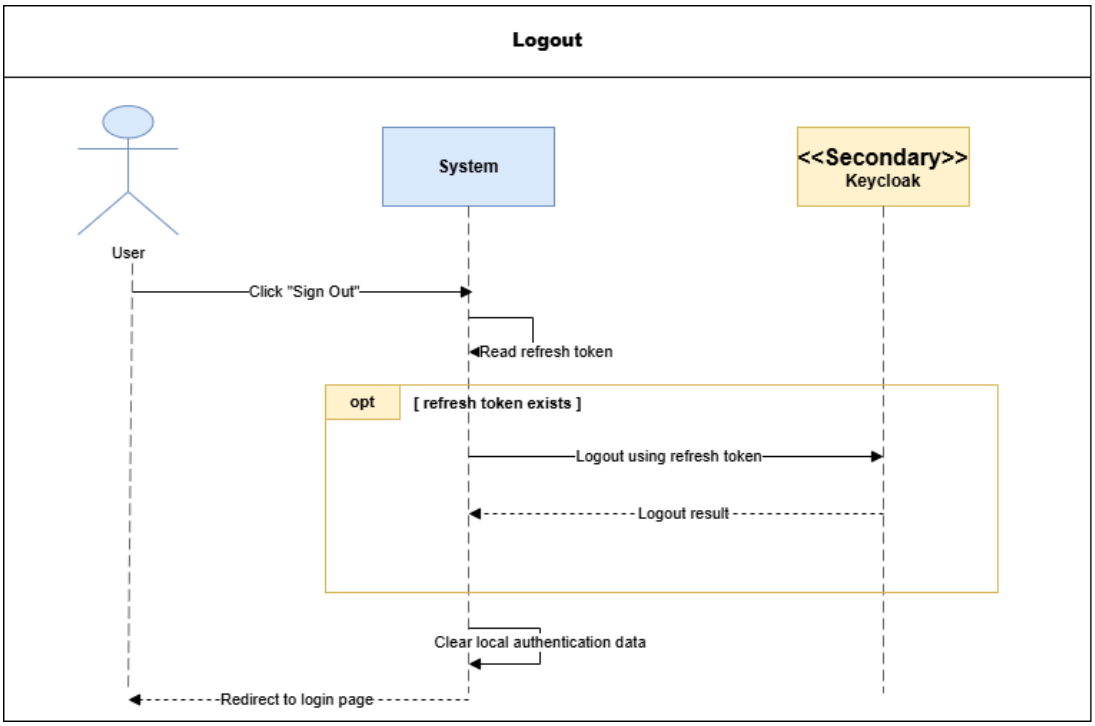


Figure 14: System Sequence Diagram – Logout

Use Case Diagram – Personal Account Management

This diagram groups the personal account actions available to an authenticated user.

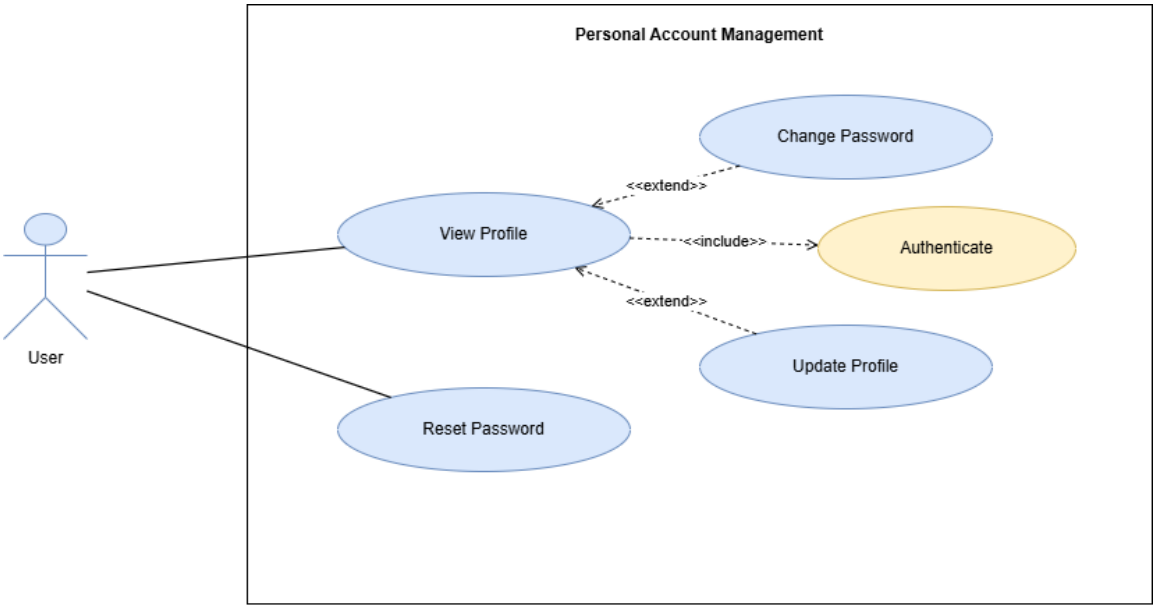


Figure 15: Use Case Diagram – Personal Account Management

Password Reset

This diagram shows how a user receives a reset code and defines a new password.

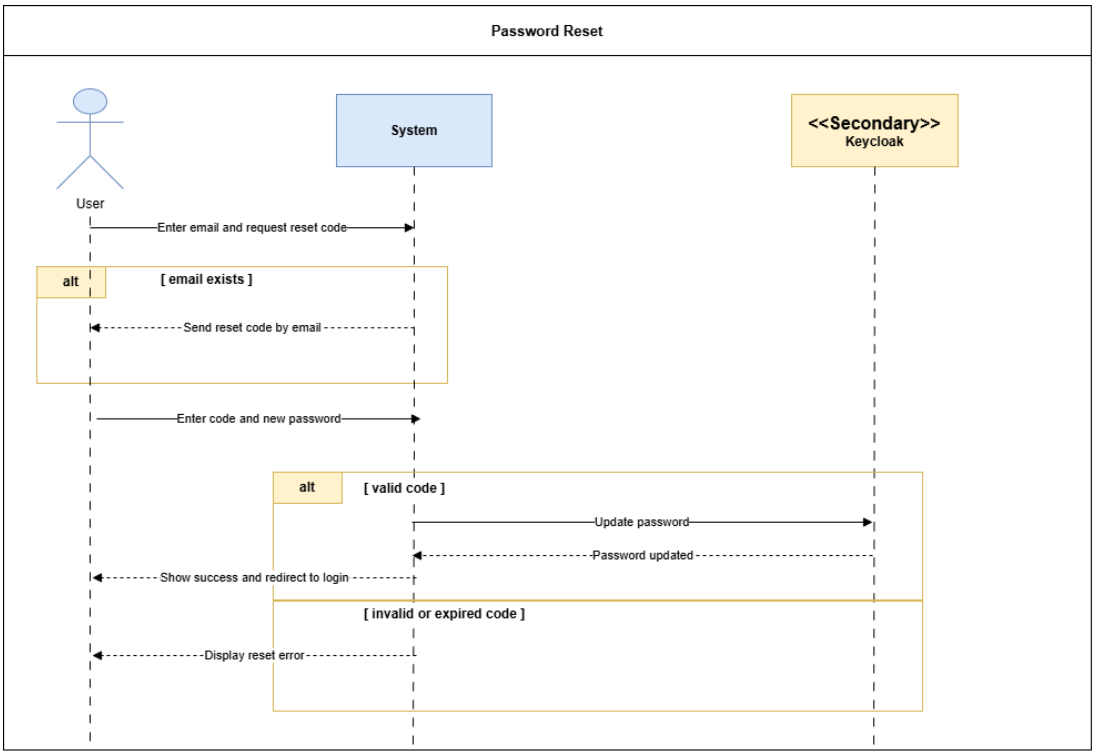


Figure 16: System Sequence Diagram – Password Reset

Profile Update

This diagram shows how an authenticated user updates allowed profile information.

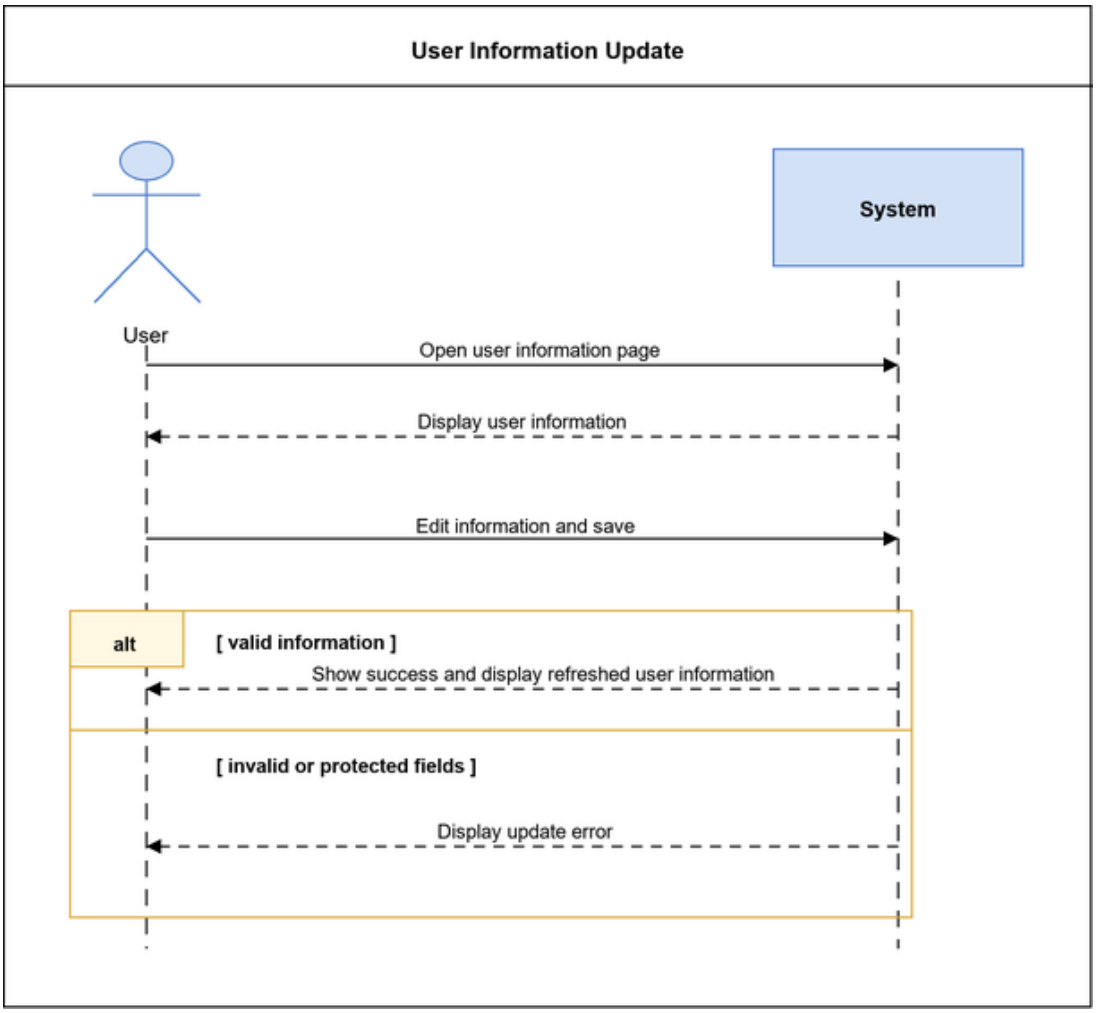


Figure 17: System Sequence Diagram – User Information Update

Authenticated Password Change

This diagram shows password change using the current password or OTP verification.

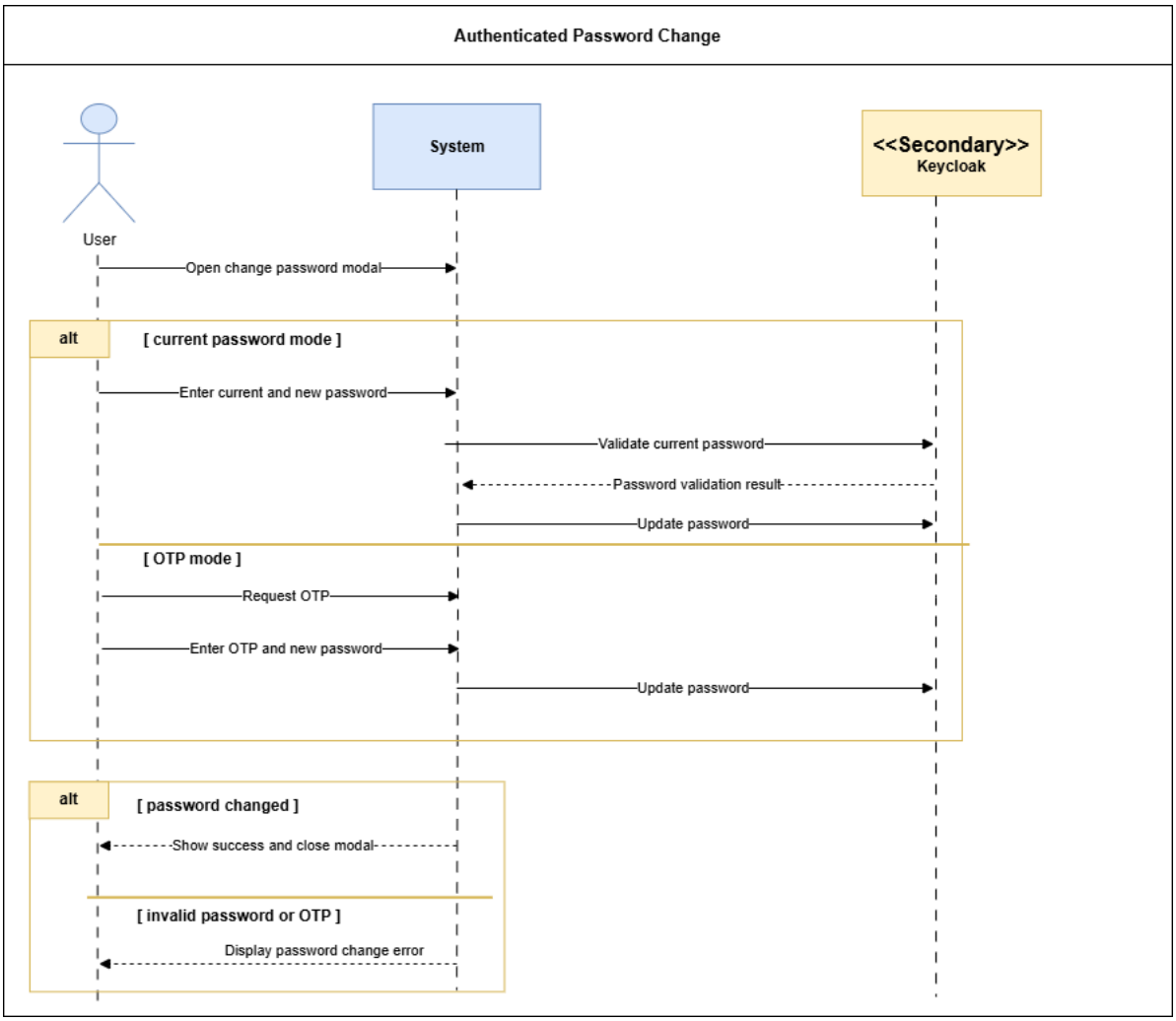


Figure 18: System Sequence Diagram – Authenticated Password Change

Use Case Diagram – HR User Account Management

This diagram groups the HR actions used to manage user accounts.

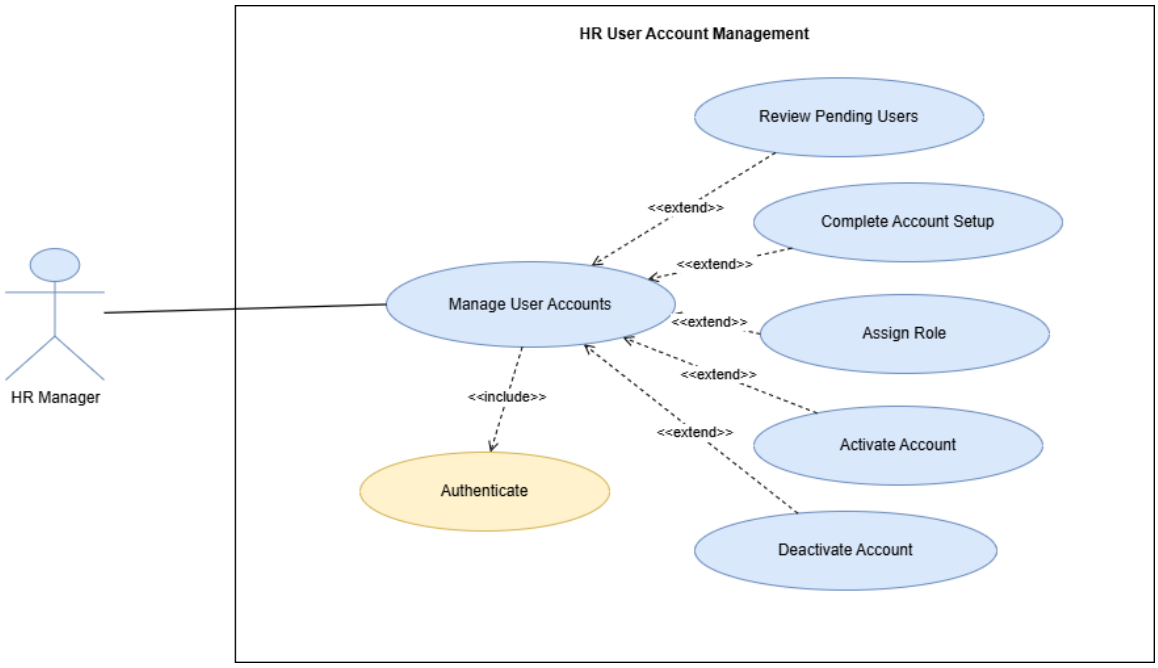


Figure 19: Use Case Diagram – HR User Account Management

HR User Account Setup

This diagram shows how HR completes a pending account setup and activates access.

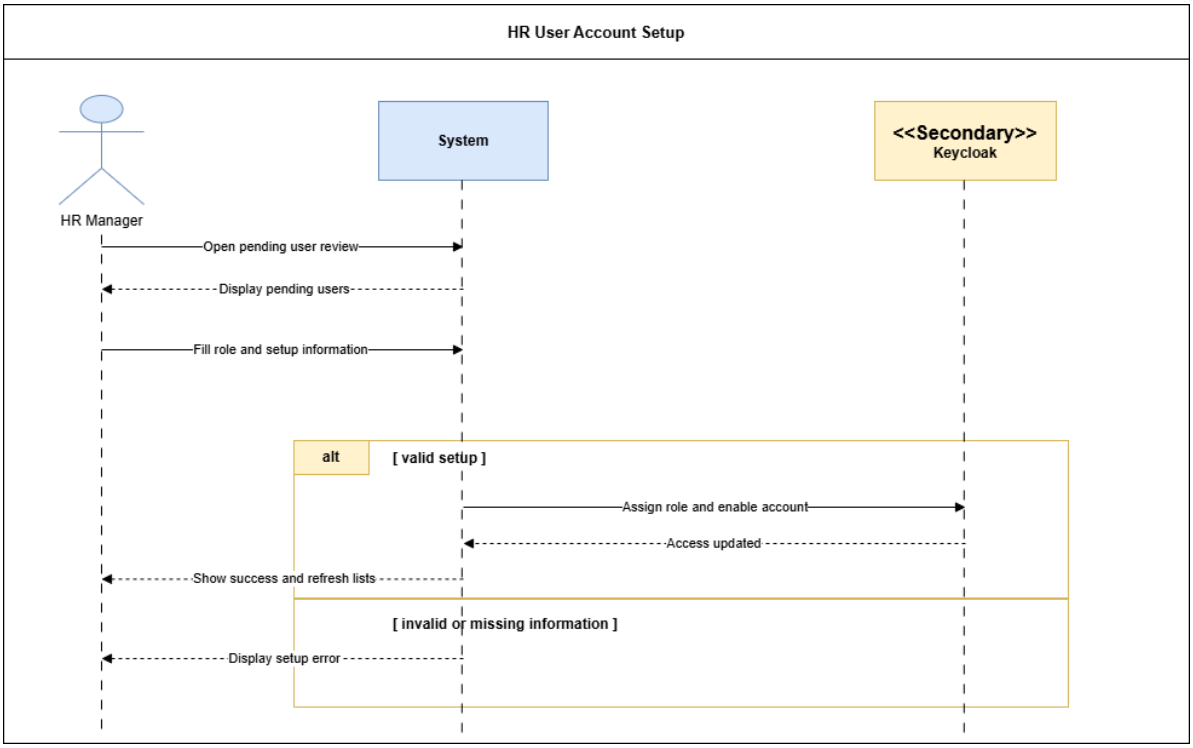


Figure 20: System Sequence Diagram – HR User Account Setup

Account Activation and Deactivation

This diagram shows HR-controlled account activation and deactivation.

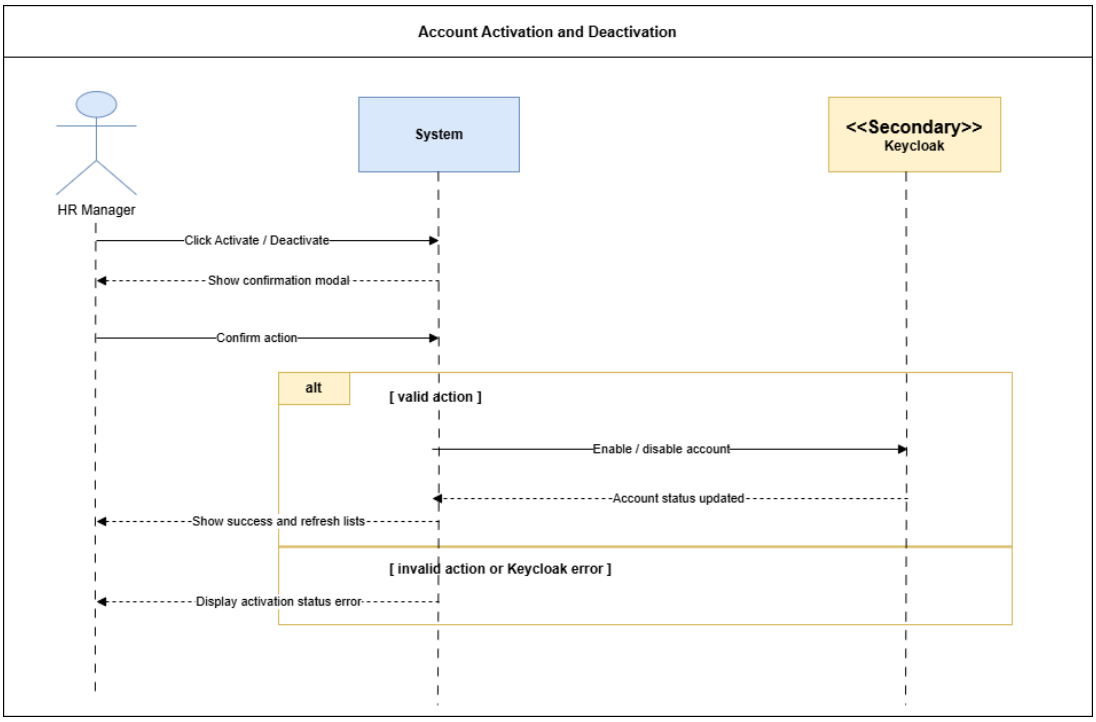


Figure 21: System Sequence Diagram – Account Activation and Deactivation

Analysis

This section summarizes the account lifecycle and the main business concepts used in Sprint 1.

Domain Model – Sprint 1 Access and Account Management

This diagram shows the main business concepts used for Sprint 1 account and access management.

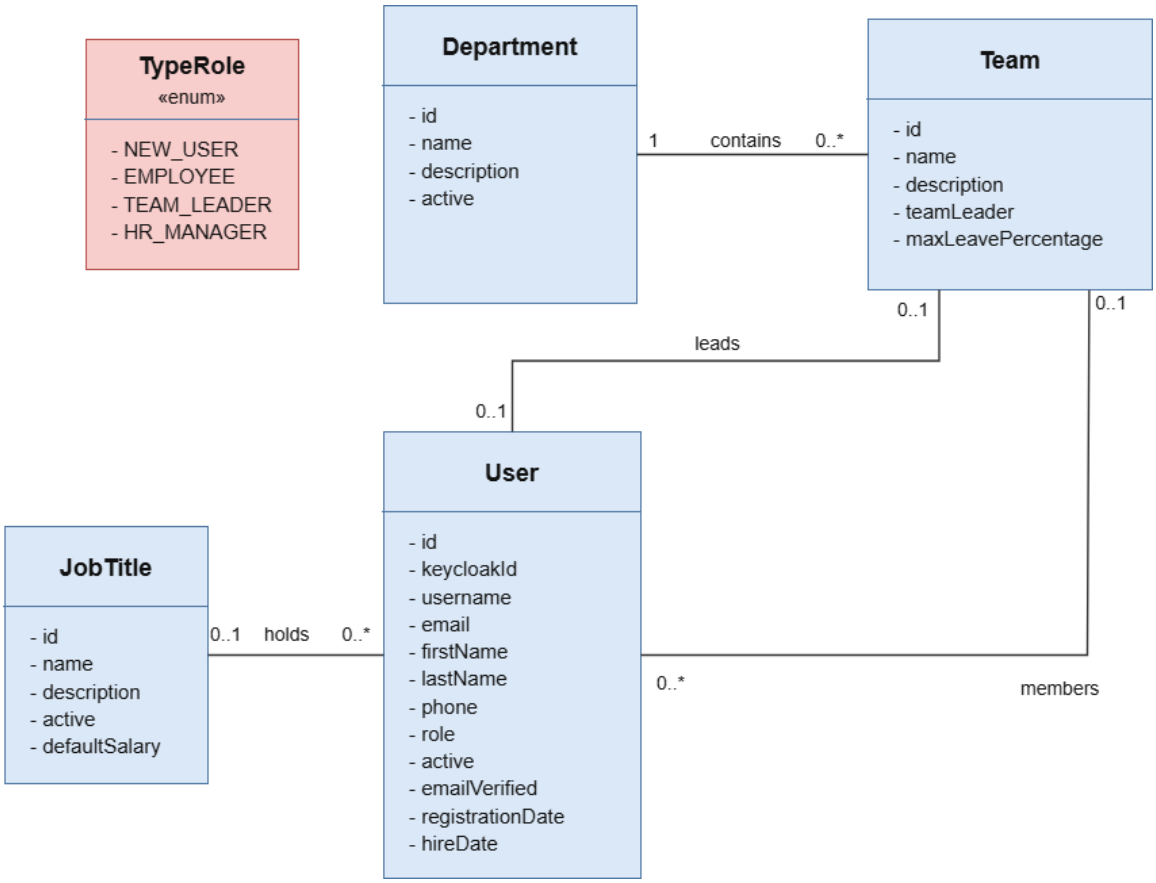


Figure 22: Domain Model – Sprint 1 Access and Account Management

Participant Class Diagrams

These diagrams identify the main frontend and backend participants involved in Sprint 1 scenarios.

Participant Class Diagram – Authentication and Access Control

This diagram shows the participants involved in authentication and role-based access control.

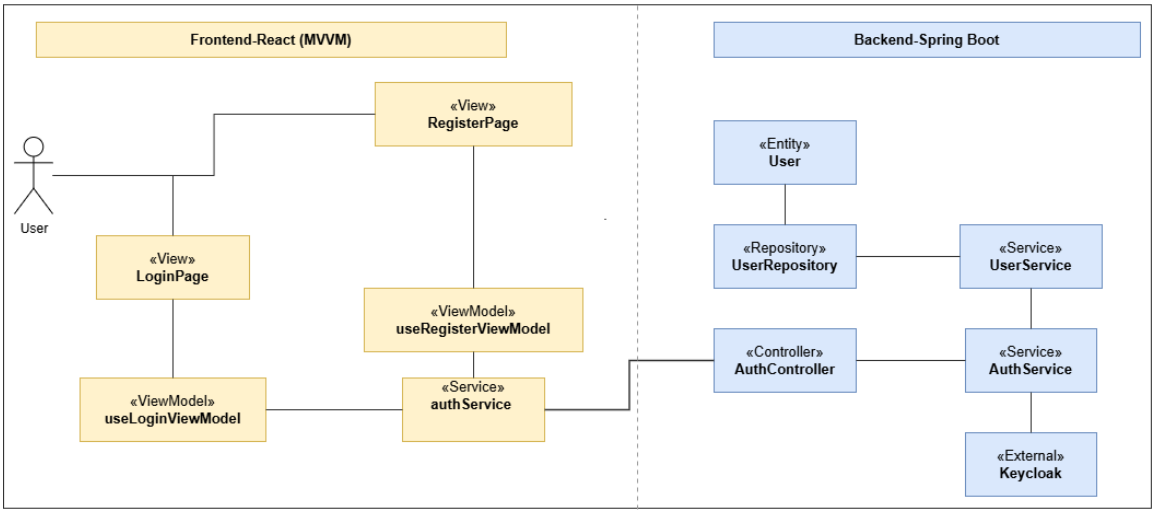


Figure 23: Participant Class Diagram – Authentication and Access Control

Participant Class Diagram – Password Reset and Password Change

This diagram shows the participants involved in password recovery and authenticated password change.

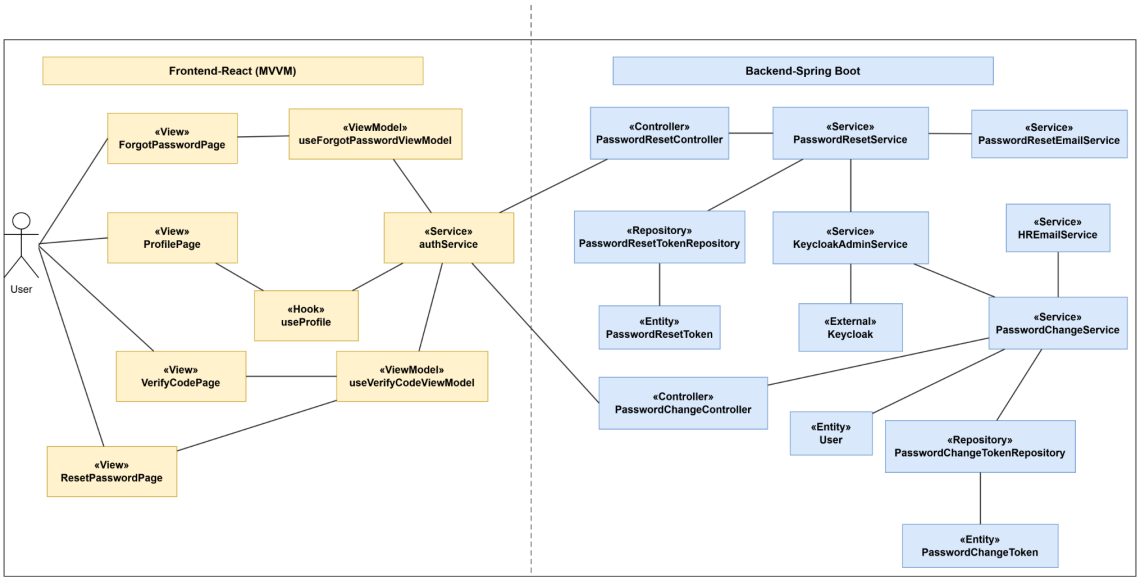


Figure 24: Participant Class Diagram – Password Reset and Password Change

Participant Class Diagram – HR Account Management

This diagram shows the participants involved in HR account setup, role assignment, and account status control.

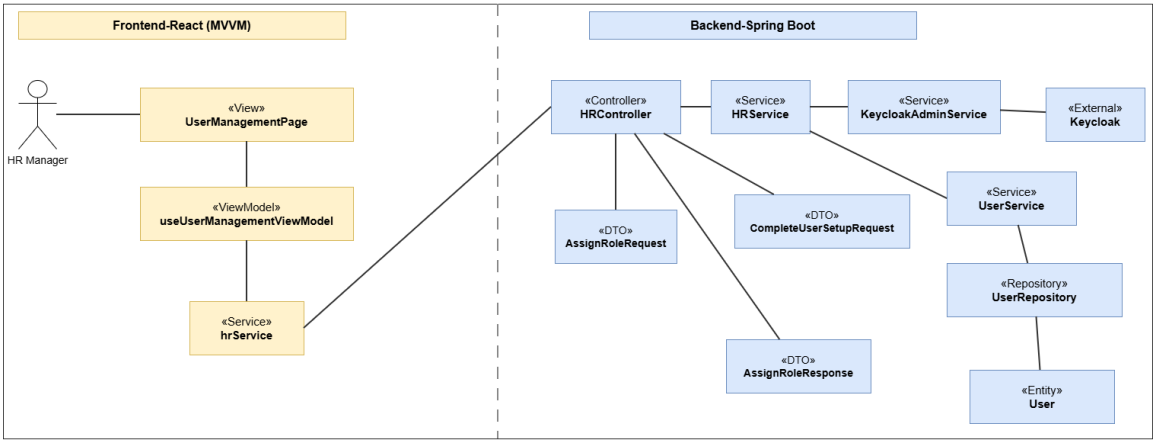


Figure 25: Participant Class Diagram – HR Account Management

State Diagrams

This diagram presents the lifecycle of a user account during the access-management workflow.

State Diagram – User Account Lifecycle

This diagram shows the transition from a pending account to an active or deactivated account.

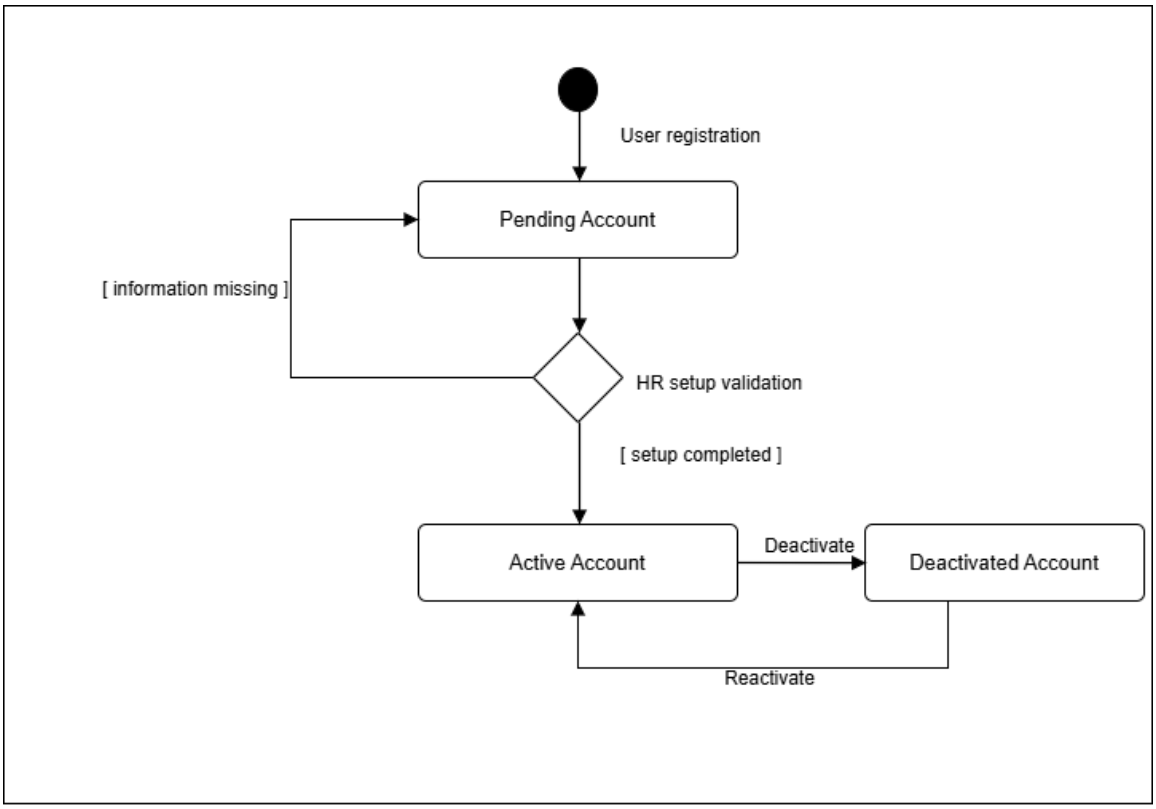


Figure 26: State Diagram – User Account Lifecycle

Design Class Diagrams

These diagrams present the main classes supporting Sprint 1 access-management features.

Design Class Diagram – Registration and Pending Access

This diagram illustrates the main classes used to register a new user and keep the account pending until HR validation.

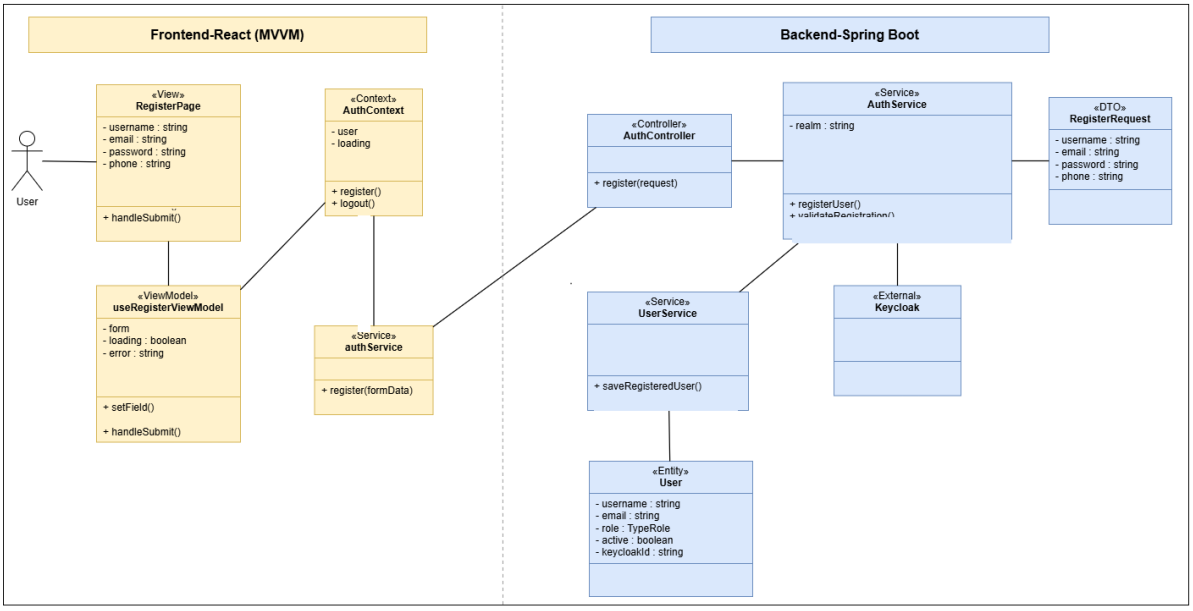


Figure 27: Design Class Diagram – Registration and Pending Access

Design Class Diagram – Security, Keycloak, and Access Control

This diagram shows the classes involved in secure authentication, Keycloak integration, and role-based access control.

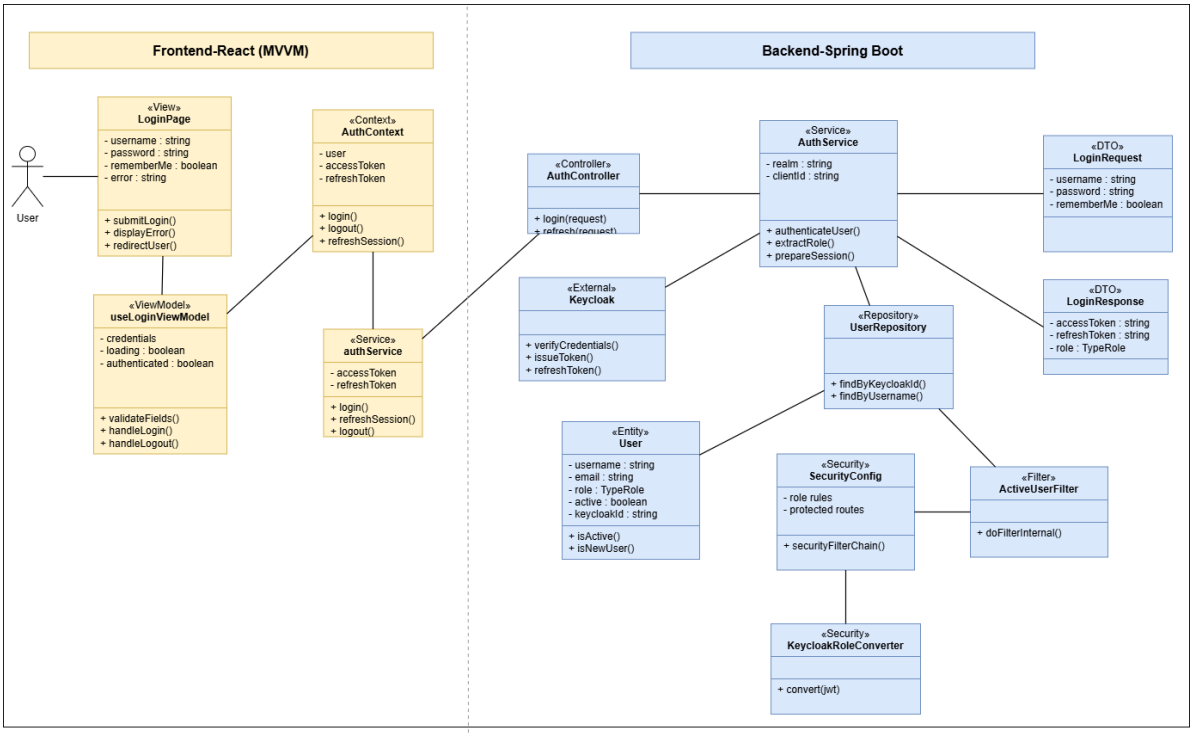


Figure 28: Design Class Diagram – Security, Keycloak, and Access Control

Design Class Diagram – HR Account Setup and Notification

This diagram presents the classes used by HR to complete account setup, update access rights, and trigger notifications.

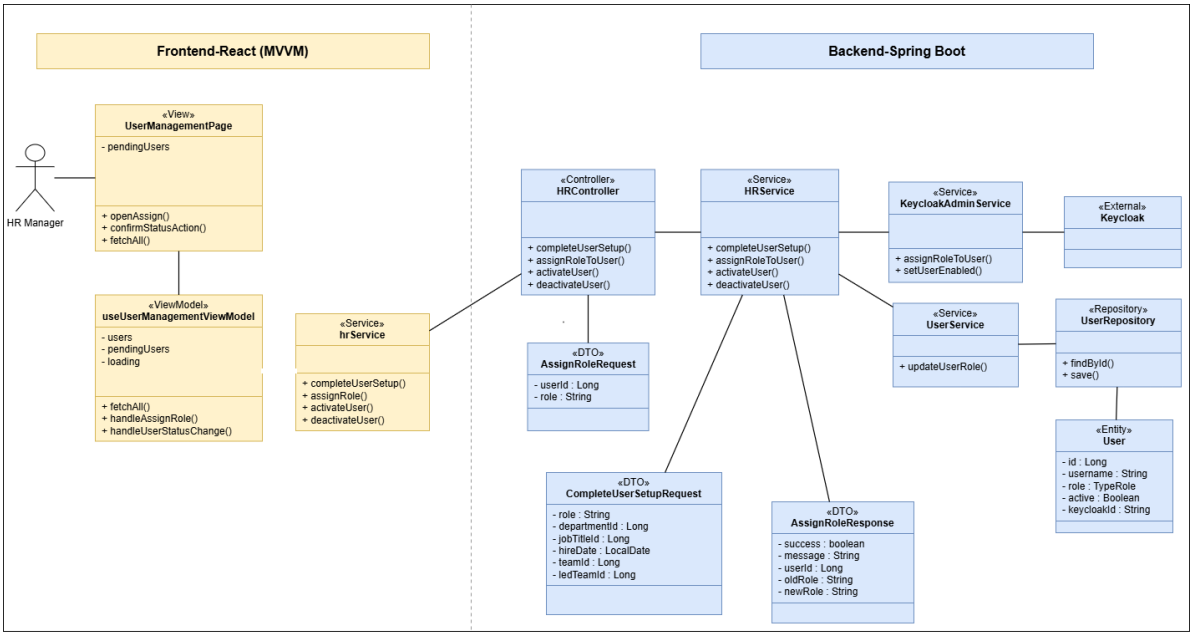


Figure 29: Design Class Diagram – HR Account Setup and Notification

Conception Sequence Diagrams

These diagrams detail the internal frontend and backend interactions behind the main Sprint 1 scenarios.

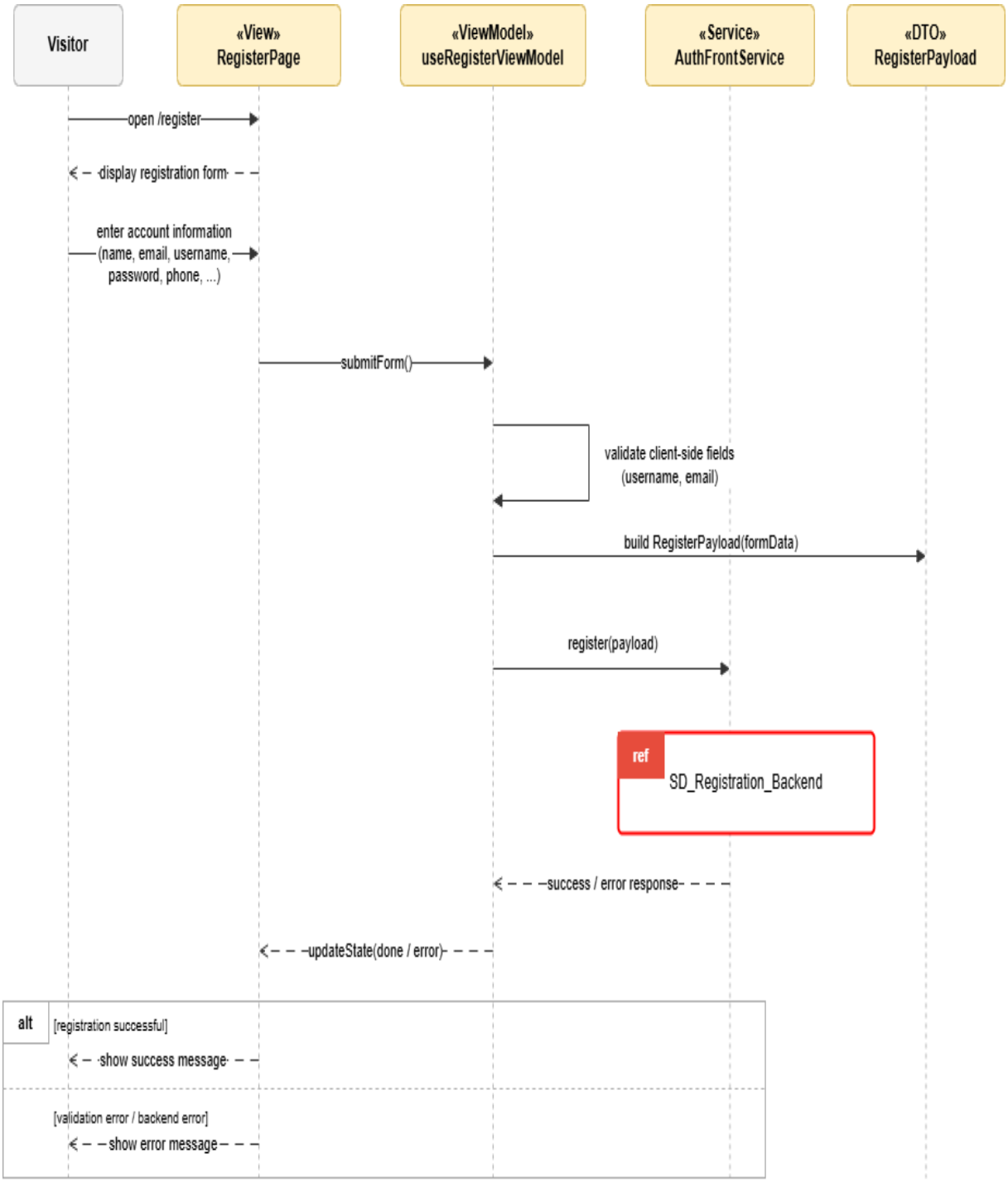


Figure 30: Conception Sequence Diagram – User Registration Frontend

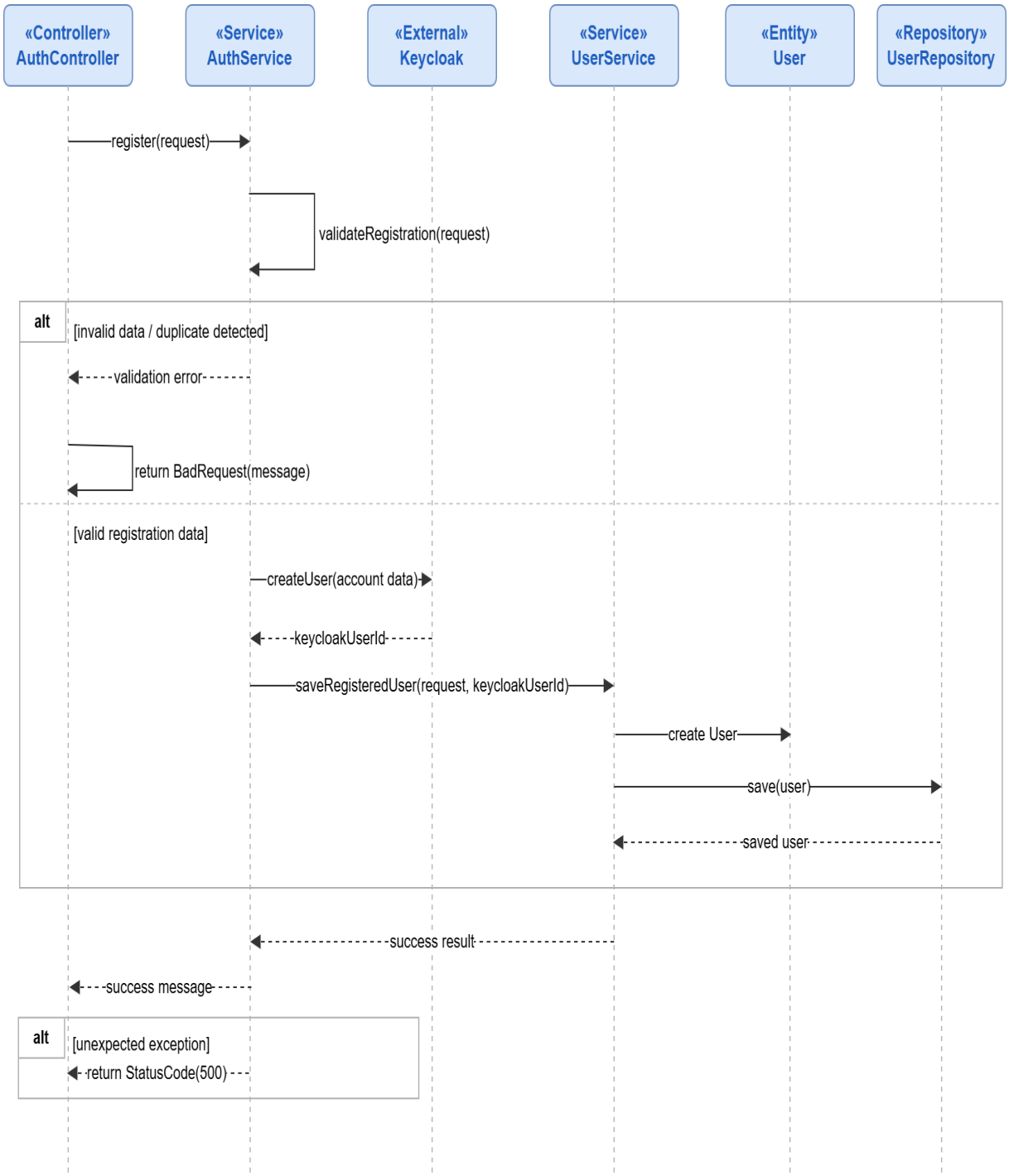


Figure 31: Conception Sequence Diagram – User Registration Backend

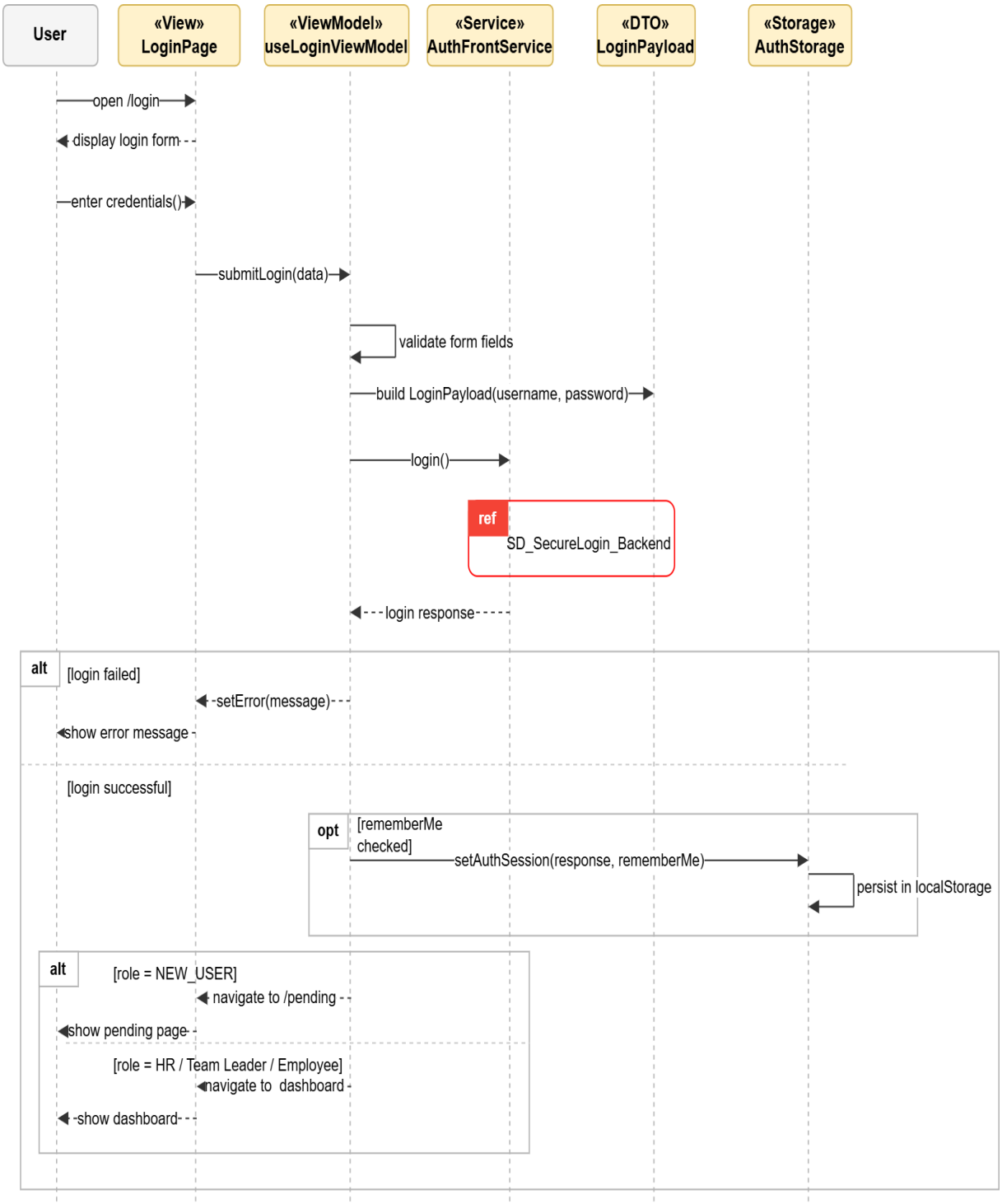


Figure 32: Conception Sequence Diagram – Secure Login Frontend

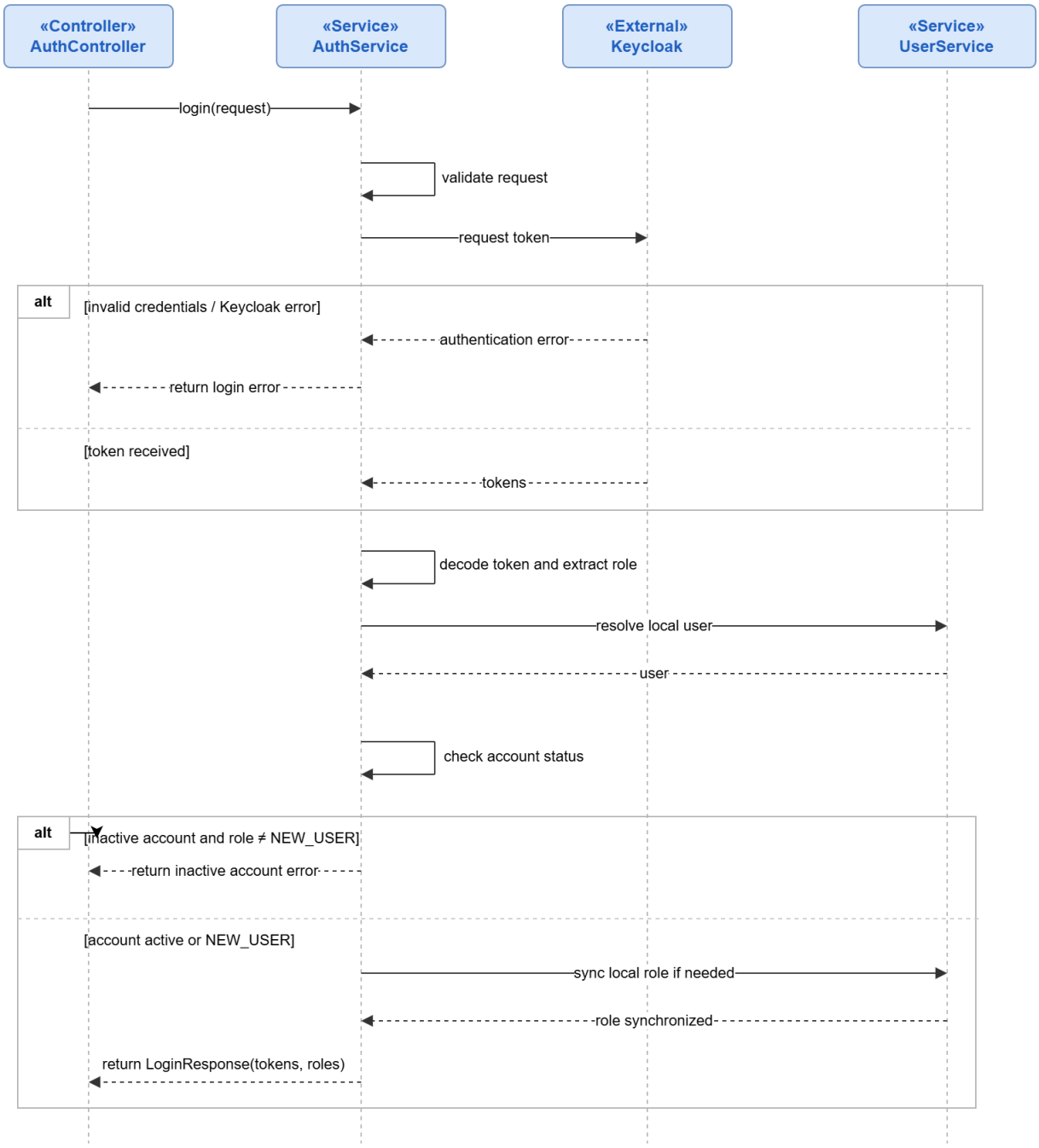


Figure 33: Conception Sequence Diagram – Secure Login Backend

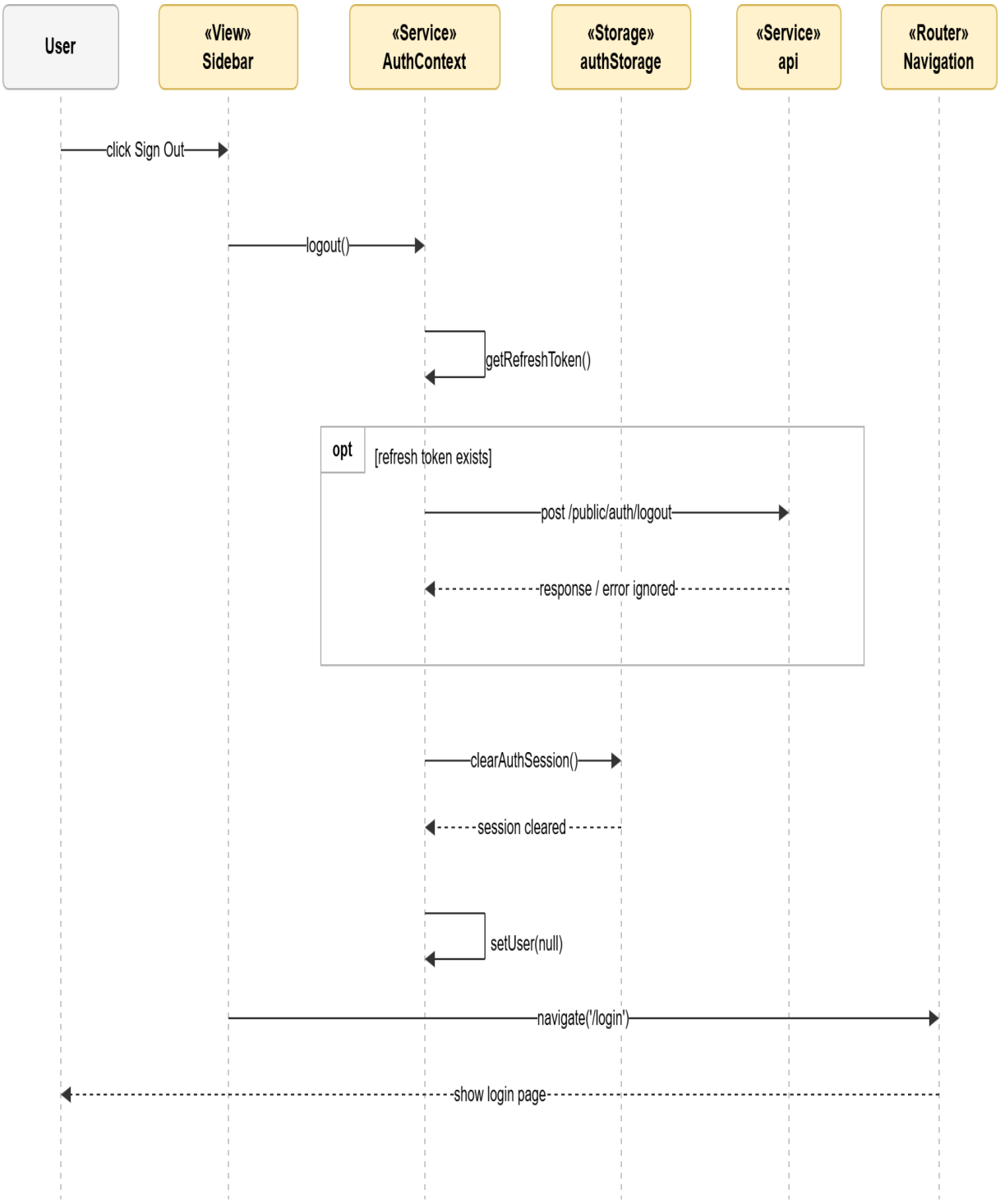


Figure 34: Conception Sequence Diagram – Logout Frontend

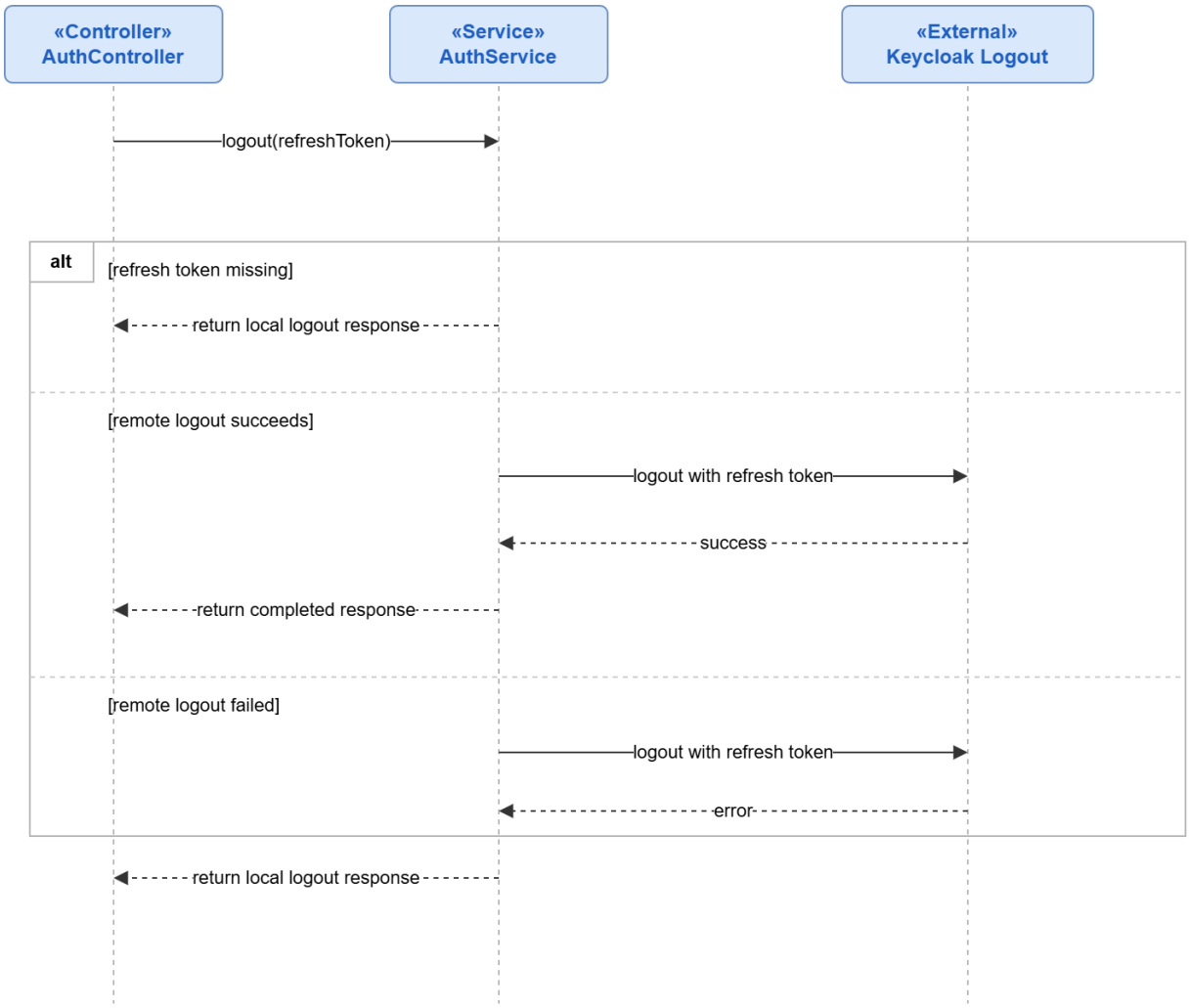


Figure 35: Conception Sequence Diagram – Logout Backend

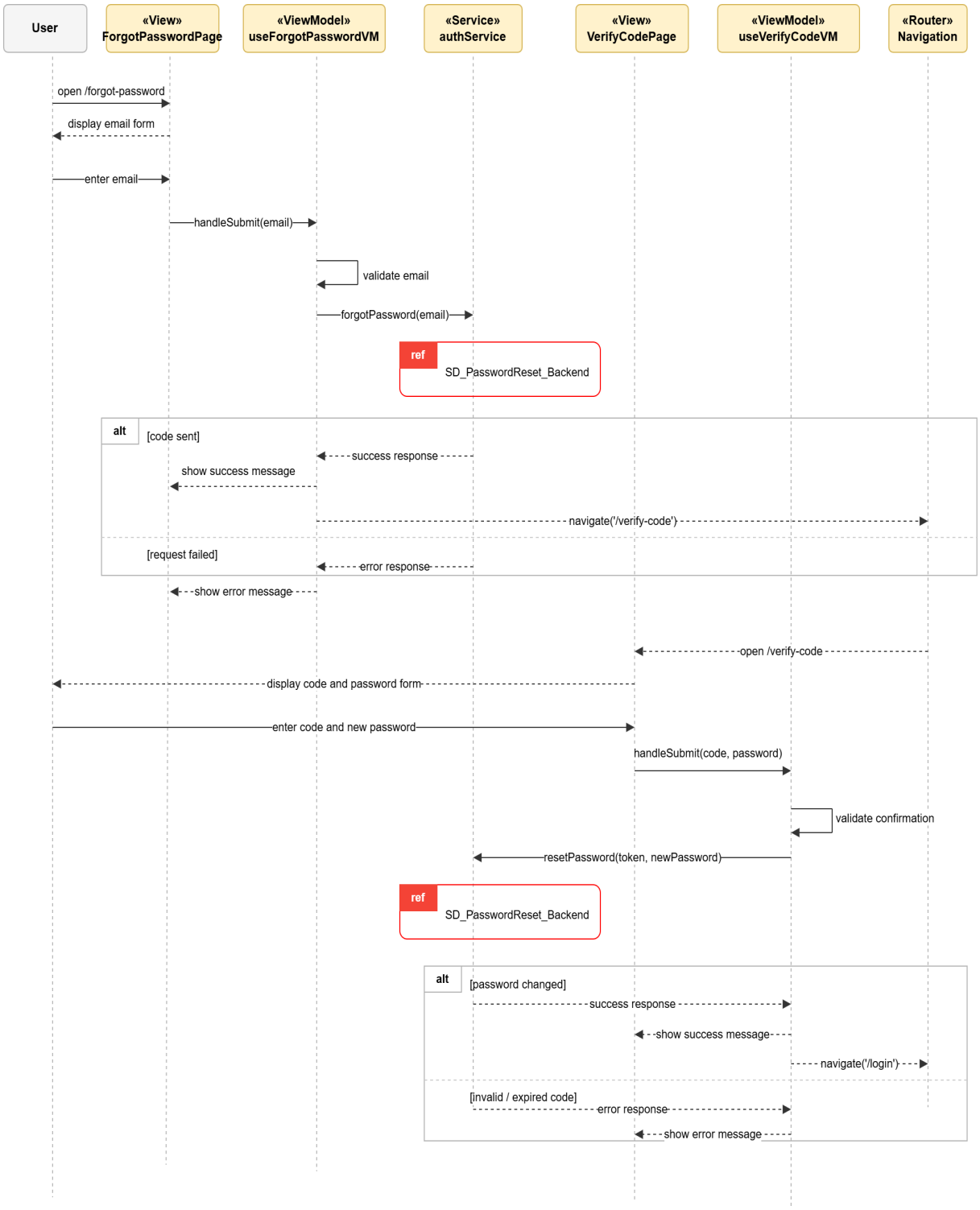


Figure 36: Conception Sequence Diagram – Password Reset Frontend

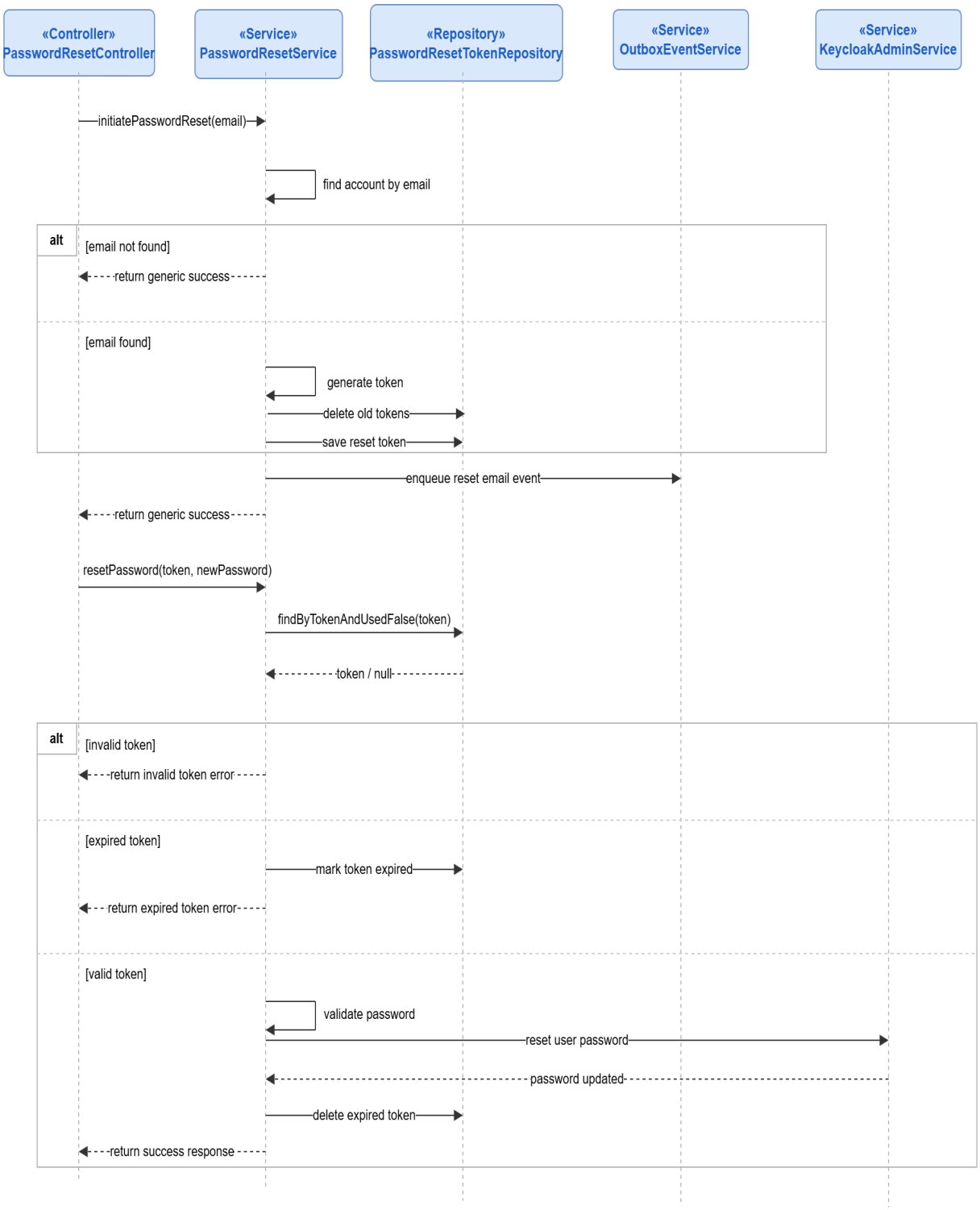


Figure 37: Conception Sequence Diagram – Password Reset Backend

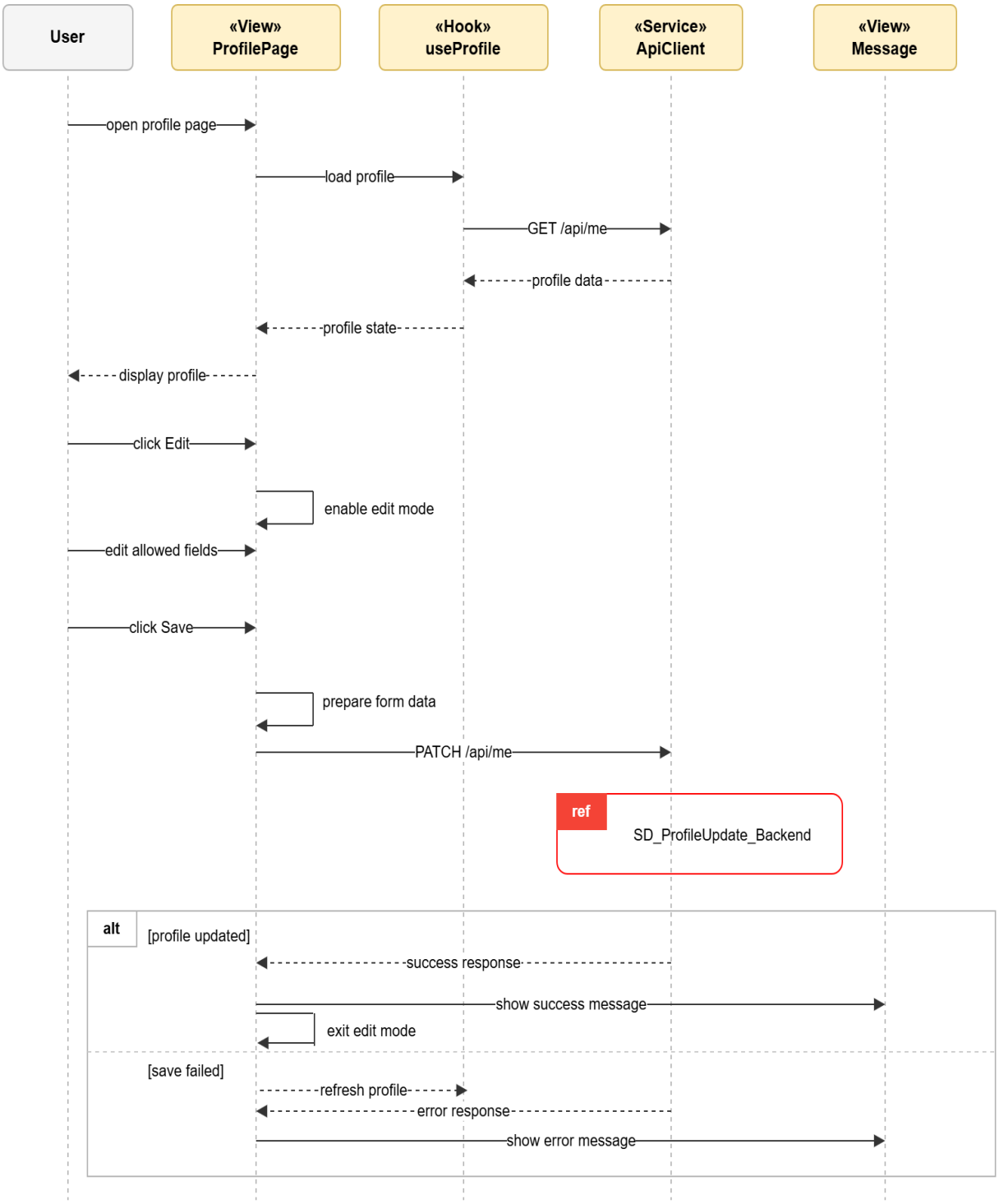


Figure 38: Conception Sequence Diagram – User Information Update Frontend

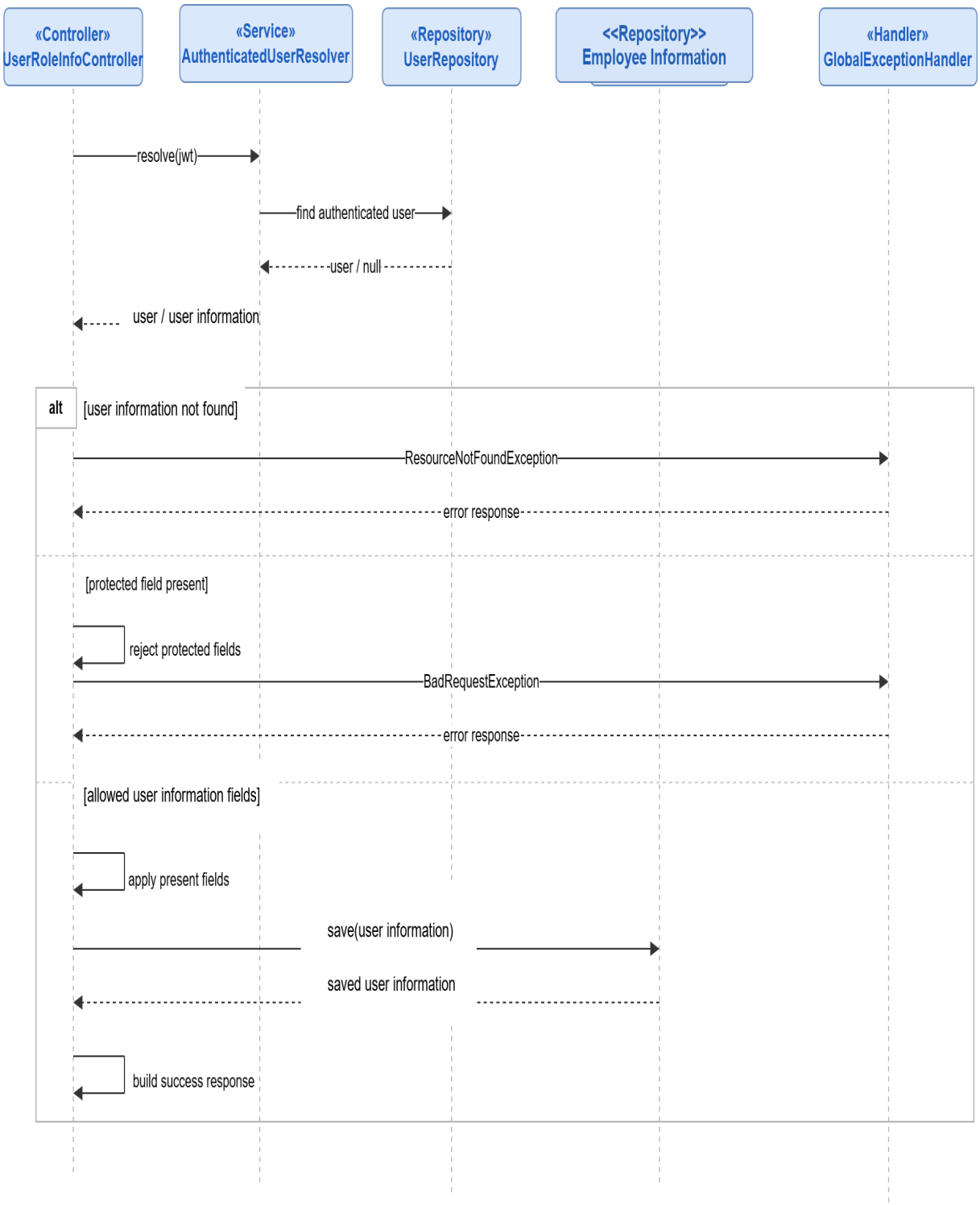


Figure 39: Conception Sequence Diagram – User Information Update Backend

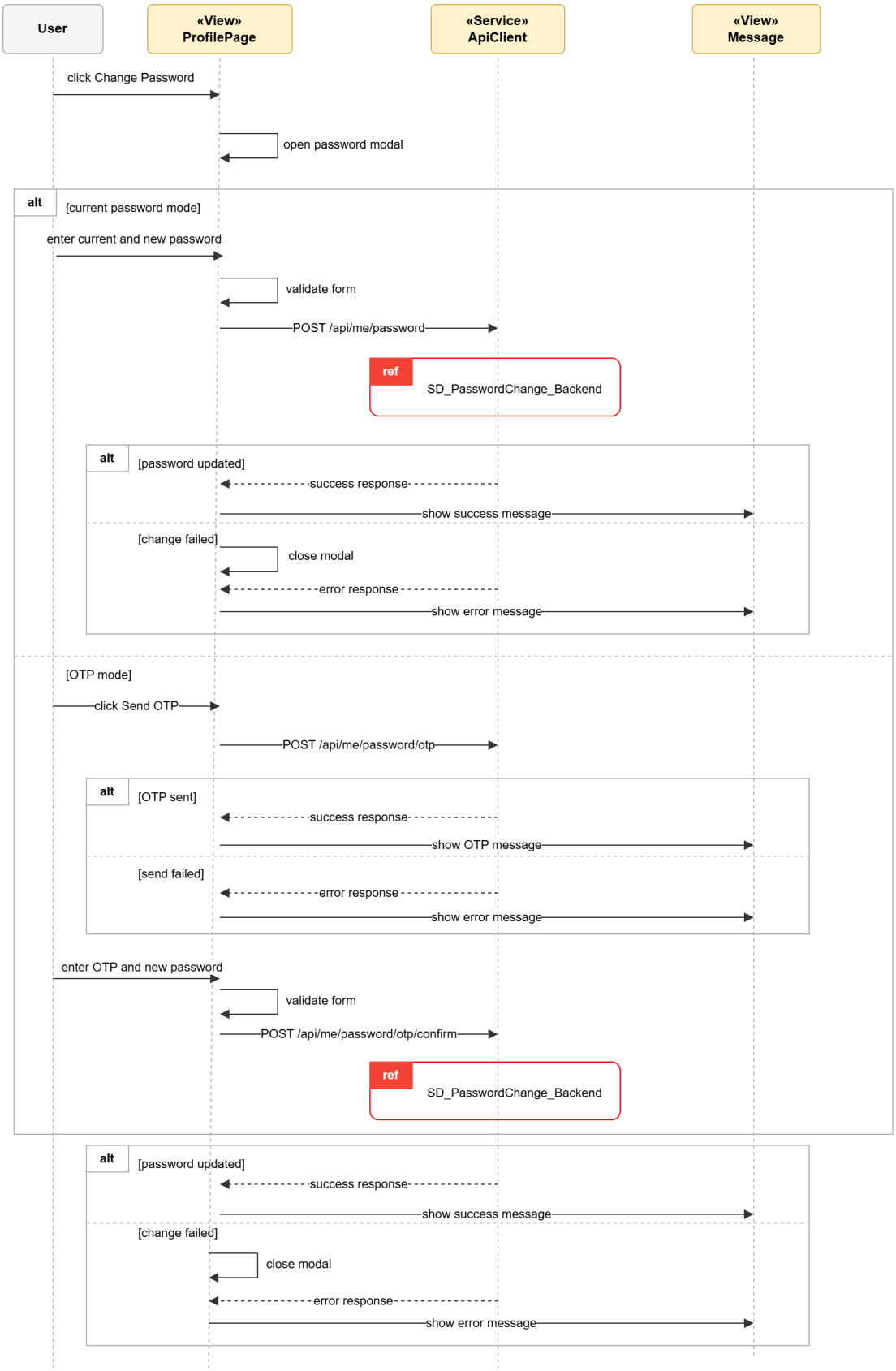


Figure 40: Conception Sequence Diagram – Authenticated Password Change Frontend

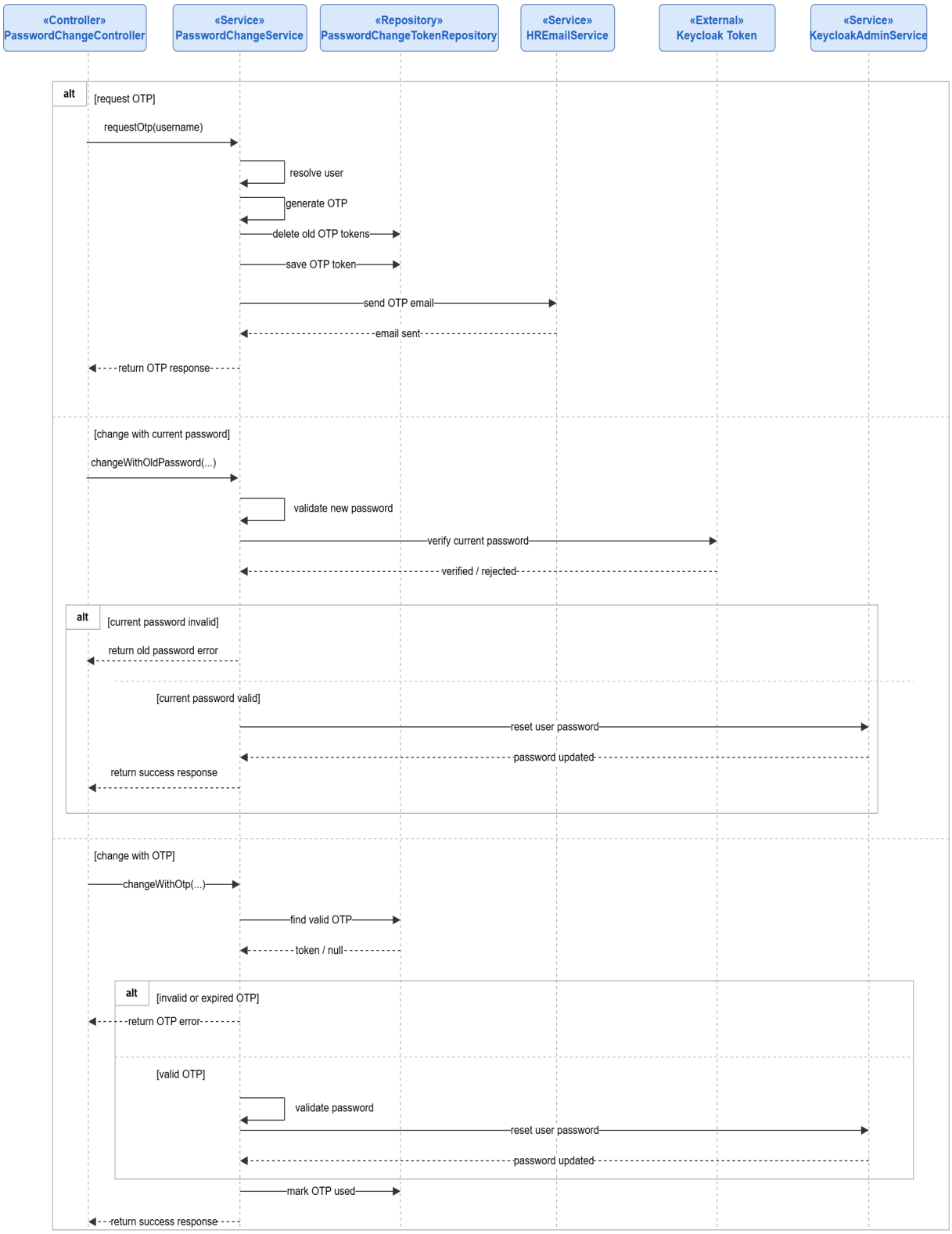


Figure 41: Conception Sequence Diagram – Authenticated Password Change Backend

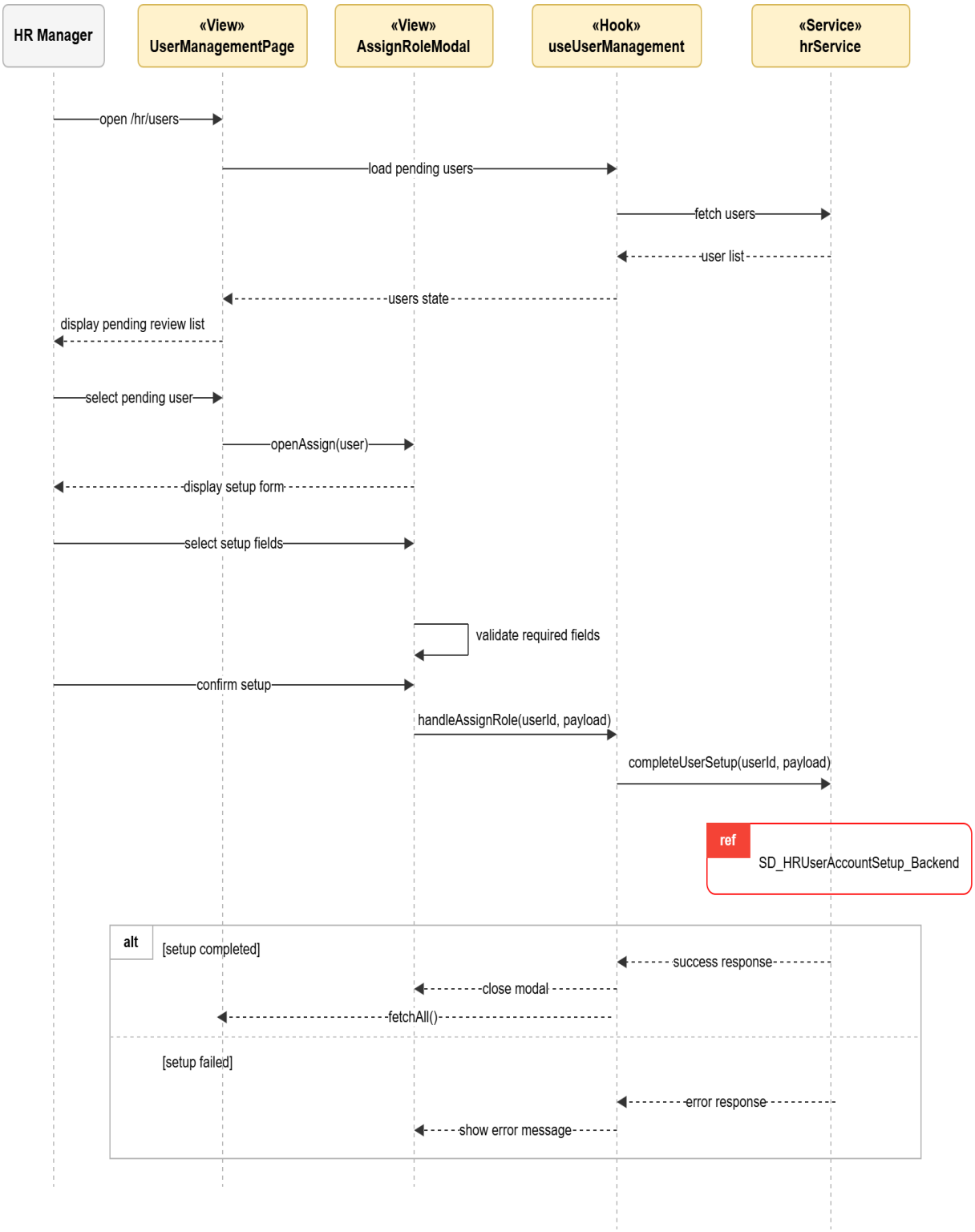


Figure 42: Conception Sequence Diagram – HR User Account Setup Frontend

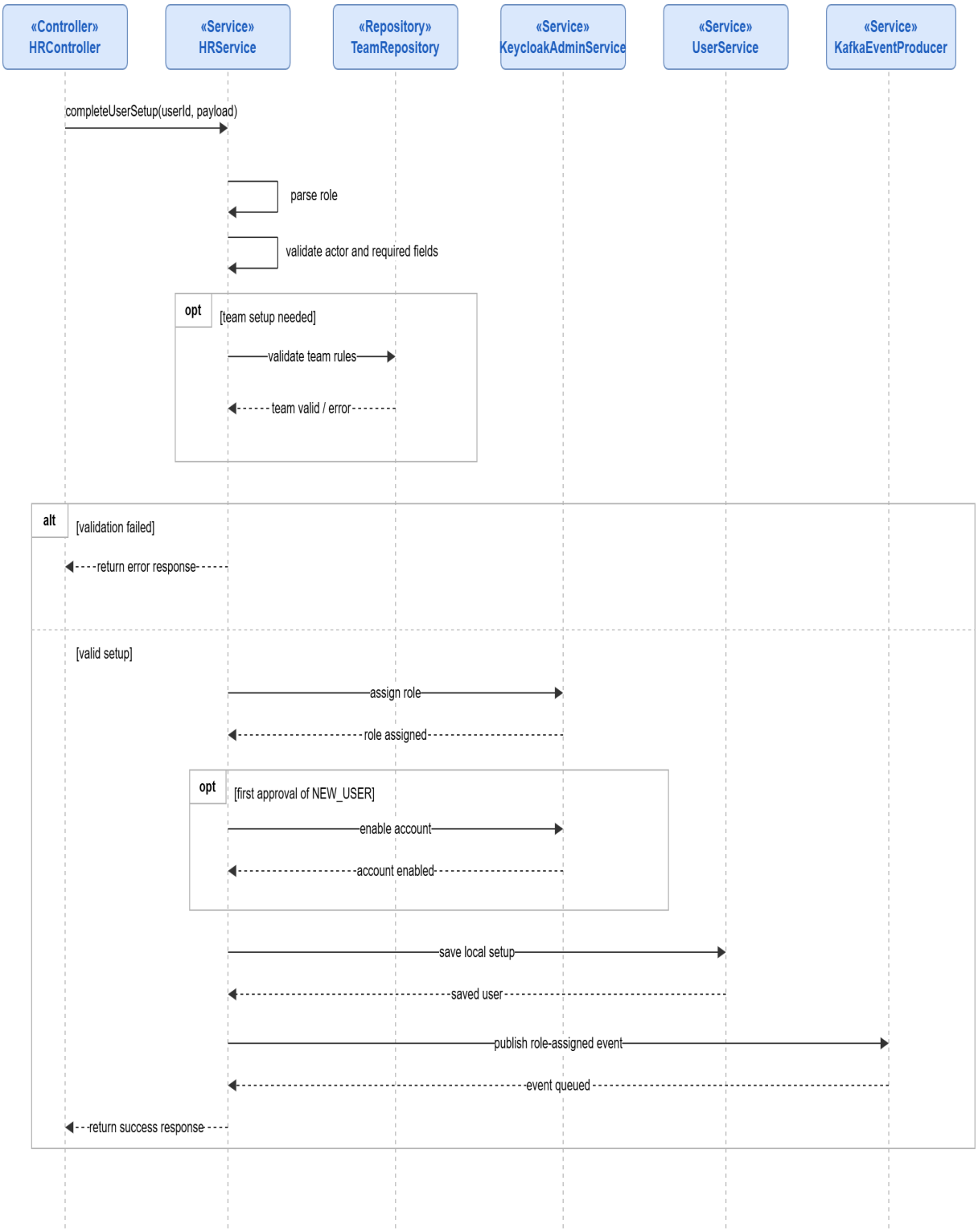


Figure 43: Conception Sequence Diagram – HR User Account Setup Backend

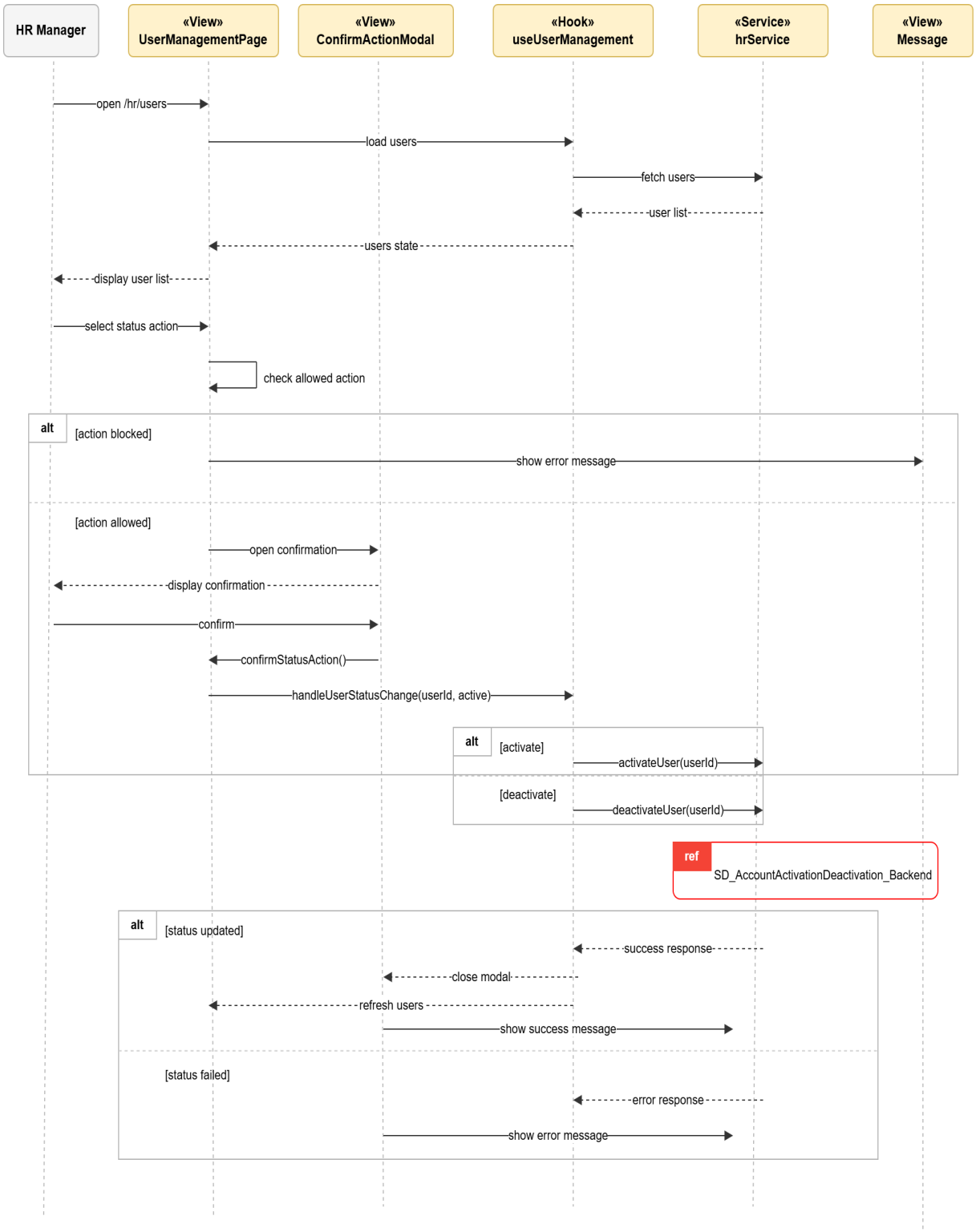


Figure 44: Conception Sequence Diagram – Account Activation and Deactivation Frontend

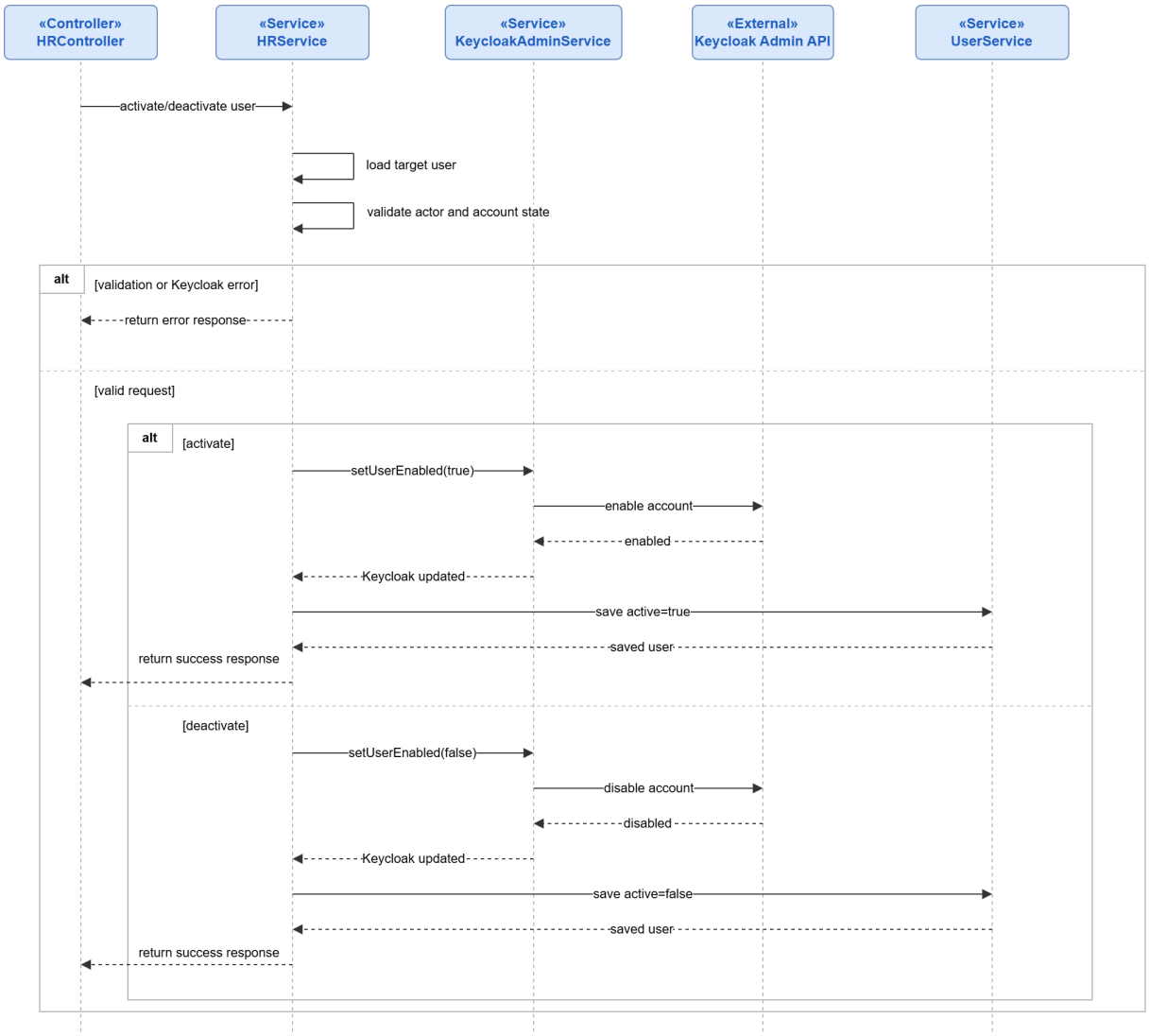


Figure 45: Conception Sequence Diagram – Account Activation and Deactivation Backend

Realization

This section presents the main user interfaces implemented during Sprint 1.

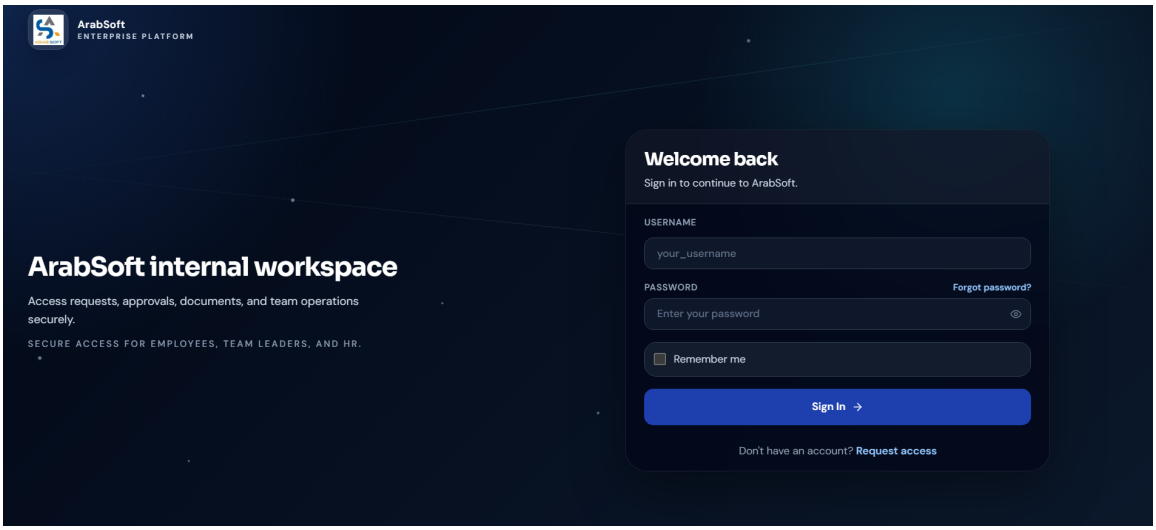


Figure 46: Secure Login Interface

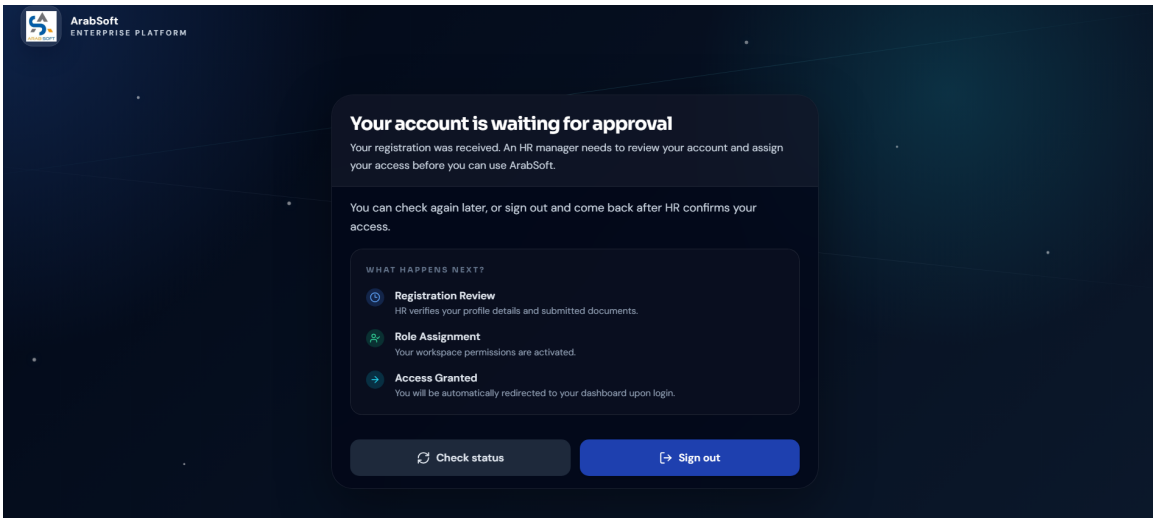


Figure 47: Pending Account Access After Registration

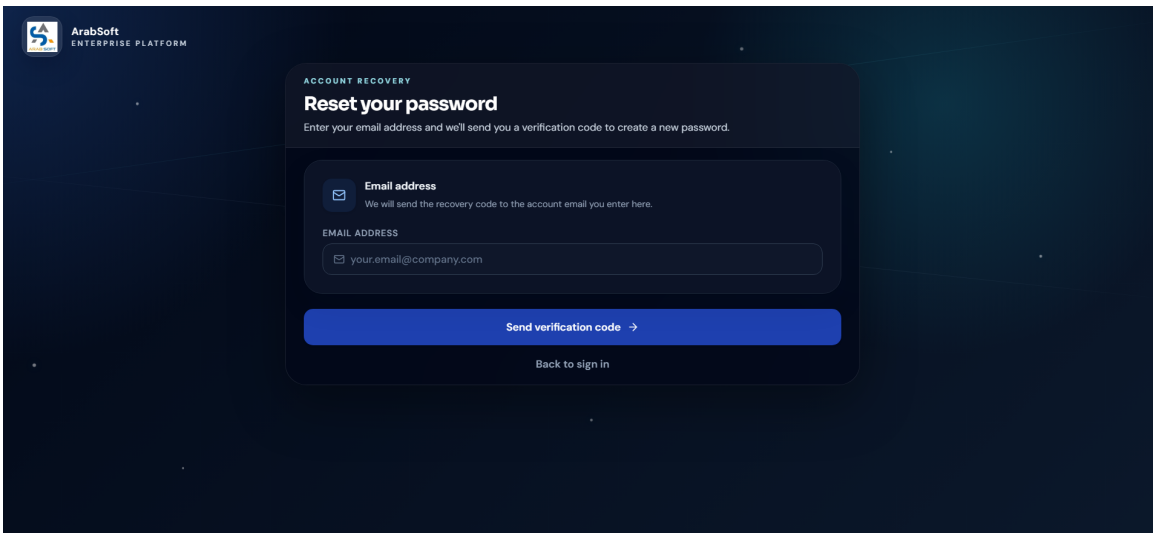


Figure 48: Password Reset Interface

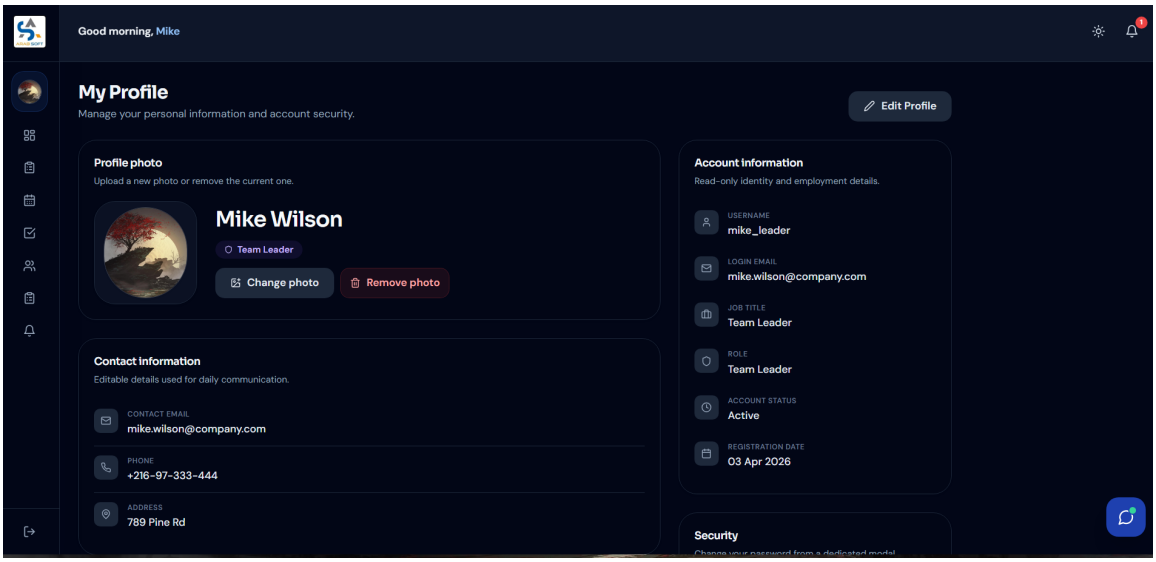


Figure 49: User Profile Update Interface

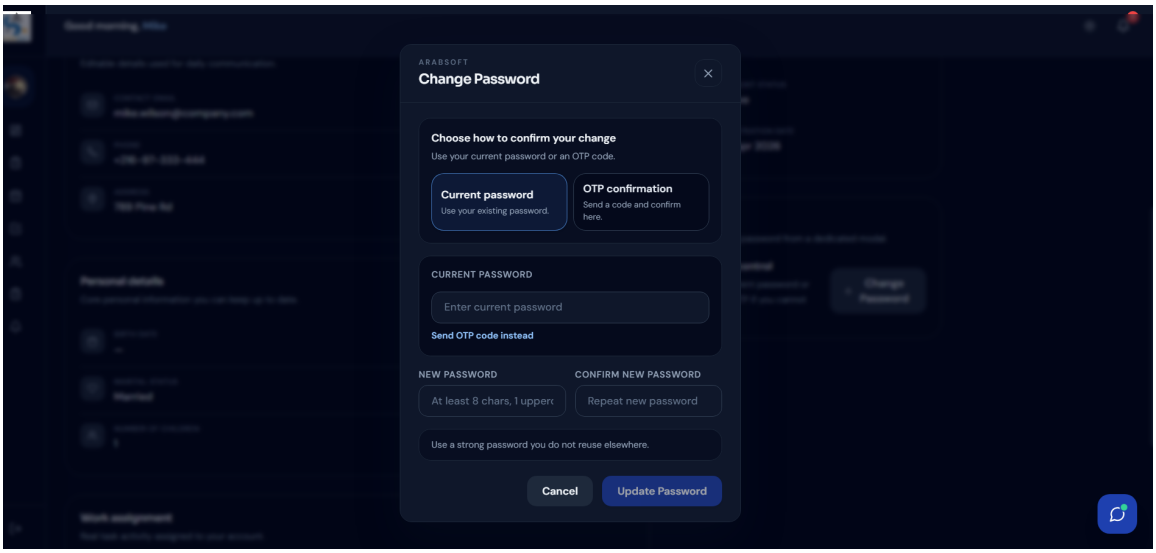


Figure 50: Authenticated Password Change Interface

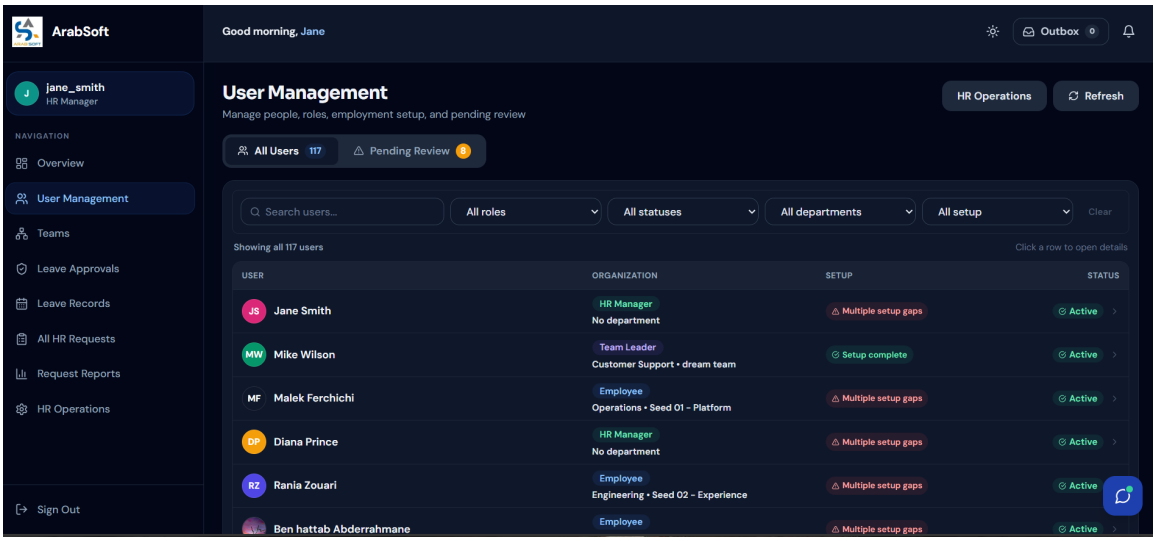


Figure 51: HR User Account Setup Interface

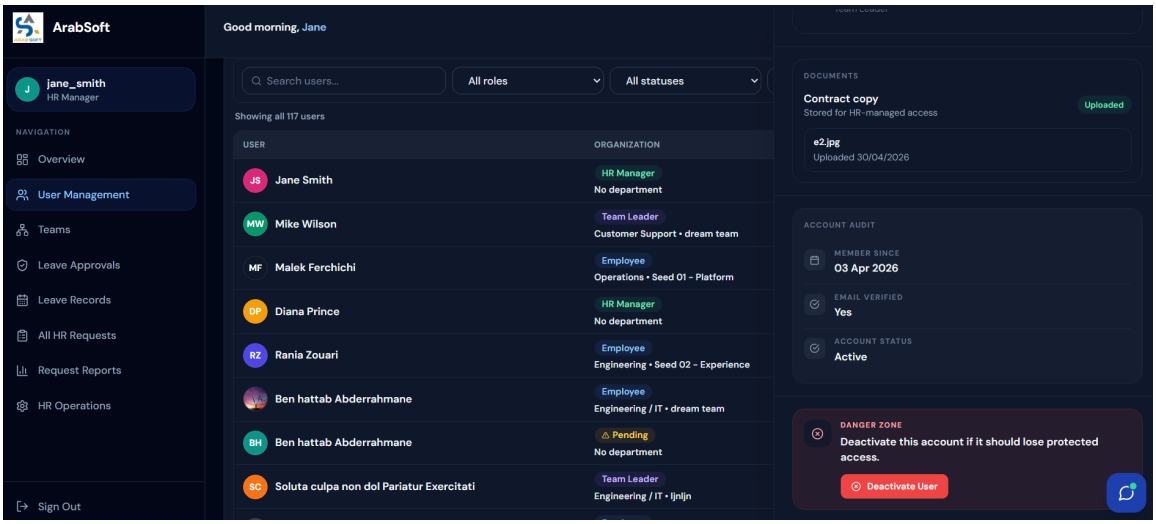


Figure 52: Account Activation and Deactivation Interface

5. Sprint Review

At the end of Sprint 1, the main access and account-management features were reviewed and validated. The sprint provided a secure foundation for the platform by implementing registration, authentication, logout, password recovery, profile update, HR account setup, role assignment, and account activation or deactivation. It also introduced the restricted **NEW_USER** state, which ensures that newly registered users cannot access the platform before HR completes their account setup.

6. Sprint Retrospective

What went well	What needs improvement
All planned Sprint 1 tasks were completed and delivered.	Future modules should keep frontend feedback, backend validation, Keycloak roles, and local user records synchronized from the beginning.

Table 3: Sprint Retrospective – Sprint 1

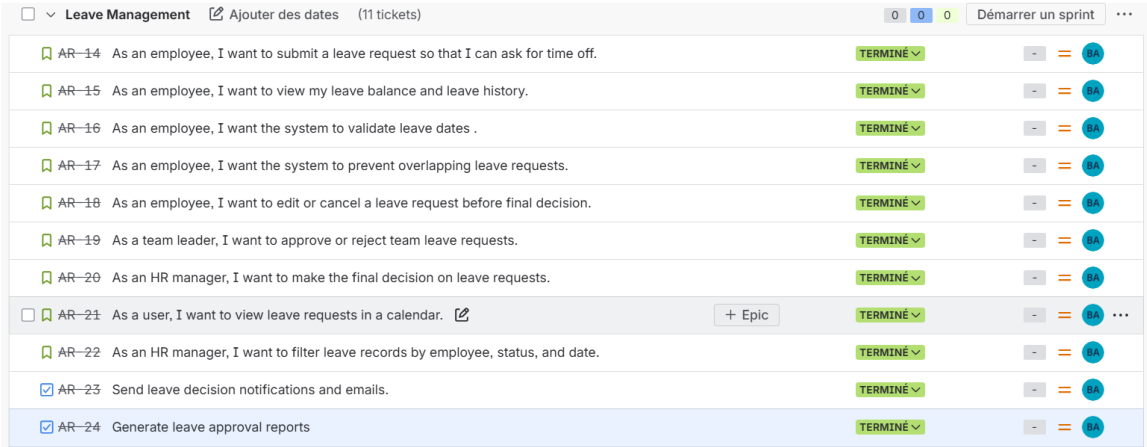
III. Sprint 2: Leave Management

1. Sprint Goal

This second sprint aims to implement the leave-management workflow of the ArabSoft platform after the access foundation established in Sprint 1. It allows employees to submit leave requests, consult their leave balance and history, update or cancel requests while they are still pending team leader review, and follow the progress of their leave workflow. The sprint also introduces working-day validation, weekend and public-holiday exclusion, overlap prevention, team leader approval or rejection, HR final decision after team leader approval, calendar visualization, filters, workflow notifications, and approved leave PDF download. The objective is to deliver the first complete administrative workflow involving employees, team leaders, and HR managers.

2. Visual Sprint Journal

The visual sprint journal below presents the selected user stories of Sprint 2 and their completion status in Jira.



Leave Management	Ajouter des dates (11 tickets)			Démarrer un sprint
AR-14	As an employee, I want to submit a leave request so that I can ask for time off.		TERMINÉ	BA
AR-15	As an employee, I want to view my leave balance and leave history.		TERMINÉ	BA
AR-16	As an employee, I want the system to validate leave dates .		TERMINÉ	BA
AR-17	As an employee, I want the system to prevent overlapping leave requests.		TERMINÉ	BA
AR-18	As an employee, I want to edit or cancel a leave request before final decision.		TERMINÉ	BA
AR-19	As a team leader, I want to approve or reject team leave requests.		TERMINÉ	BA
AR-20	As an HR manager, I want to make the final decision on leave requests.		TERMINÉ	BA
AR-21	As a user, I want to view leave requests in a calendar.	+ Epic	TERMINÉ	BA
AR-22	As an HR manager, I want to filter leave records by employee, status, and date.		TERMINÉ	BA
AR-23	Send leave decision notifications and emails.		TERMINÉ	BA
AR-24	Generate leave approval reports		TERMINÉ	BA

Visual Sprint Journal – Sprint 2

3. Sprint Backlog

The table below presents the backlog of the second sprint, planned as a two-week iteration dedicated to leave management.

Sprint	Stories	Tasks	Duration
Sprint 2	**Leave request submission and balance consultation**	Implement leave request creation	2d
		Display leave balance and leave history	
		Connect employee leave actions with backend APIs	
	Working-day, holiday, and overlap validation	Validate leave dates using working days	2d
		Exclude weekends and public holidays from deducted days	
		Prevent overlapping leave requests	
		Display clear validation feedback to the user	
	Leave update and cancellation	Allow request update while pending team leader review	1d
		Allow request cancellation while pending team leader review	
	Team leader leave decision	Implement team leader approval	2d
		Implement team leader rejection with reason	
	HR final leave decision	Implement HR final approval	2d
		Implement HR final rejection and decision tracking	
	Leave calendar, filters, notifications, and approved PDF	Display leave requests in role-aware calendar views	1d
		Add leave filters by employee, status, date, department, and team	
		Send workflow events and leave notifications	
		Enable approved leave PDF download	

Table 4: Sprint Backlog – Sprint 2

4. Sprint Implementation

Requirements Analysis

Sprint 2 covers the leave-management workflow, including employee leave request management, team leader review, HR final decision after team leader approval, leave calendar consultation, filtering, workflow notifications, and approved leave PDF download. The following use-case diagram presents the global scope of the sprint.

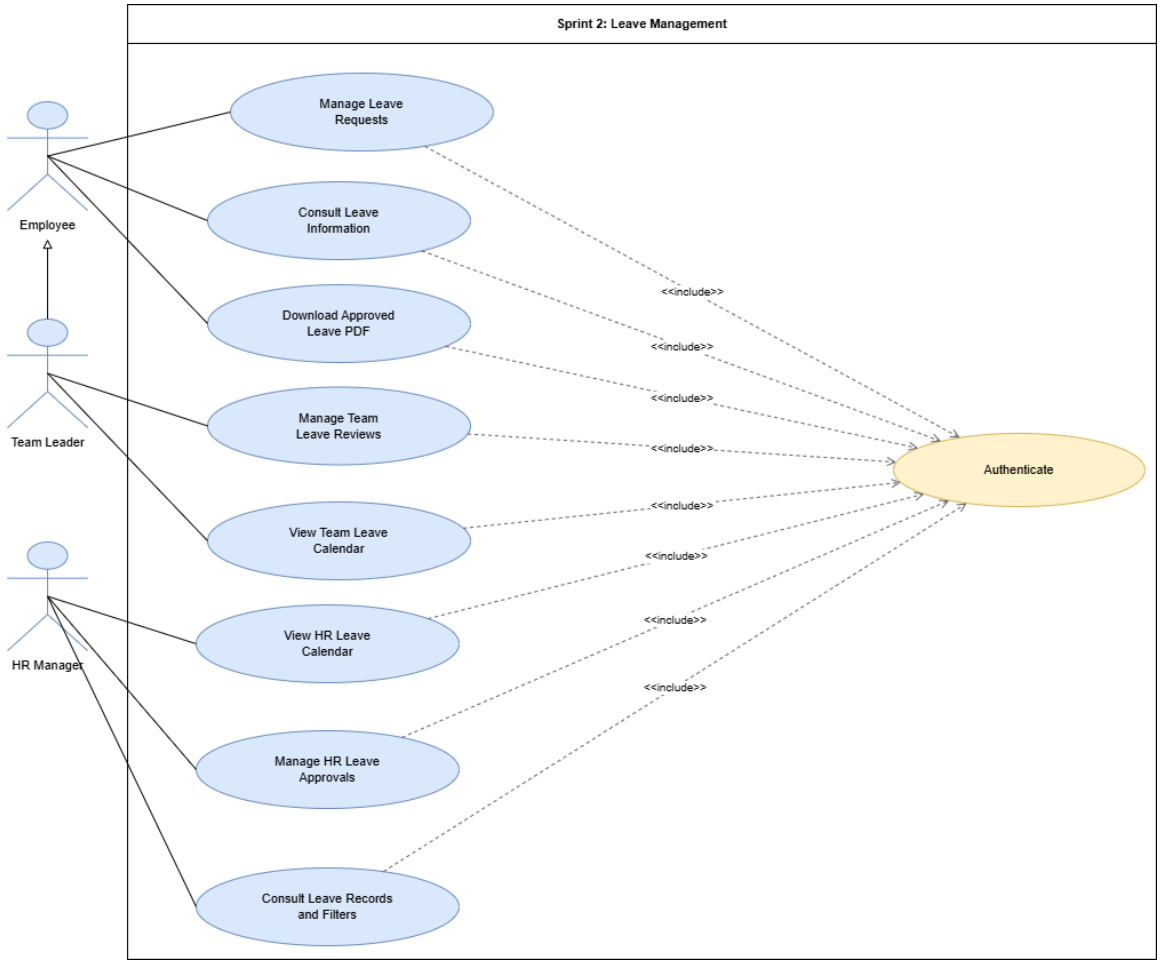


Figure 53: Use Case Diagram – Sprint 2 Leave Management

System Sequence Diagrams

These diagrams summarize the main user-system exchanges for Sprint 2 leave-management scenarios. The section also introduces focused use-case views before the employee request-management scenarios and the approval workflow scenarios.

Use Case Diagram – Employee Leave Request Management

This diagram groups the leave actions available to an employee, including request submission, balance consultation, history tracking, update or cancellation while pending team leader review, and approved leave PDF download.

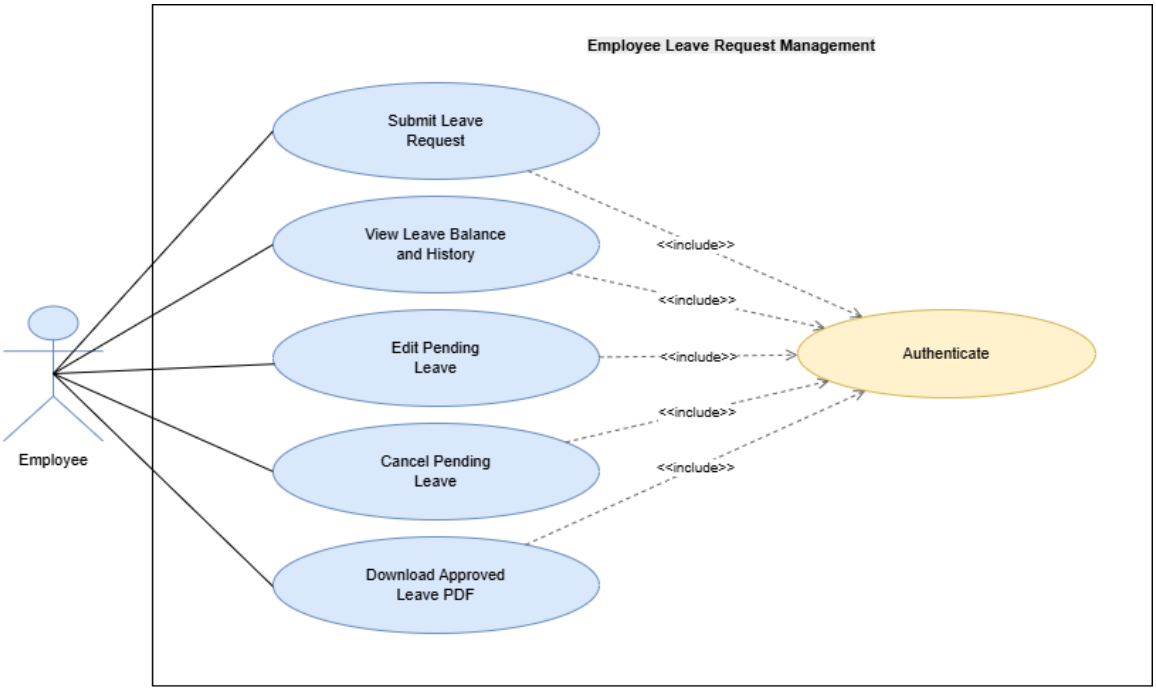


Figure 54: Use Case Diagram – Employee Leave Request Management

Submit Leave Request

This diagram shows how an employee submits a leave request and receives validation feedback from the system.

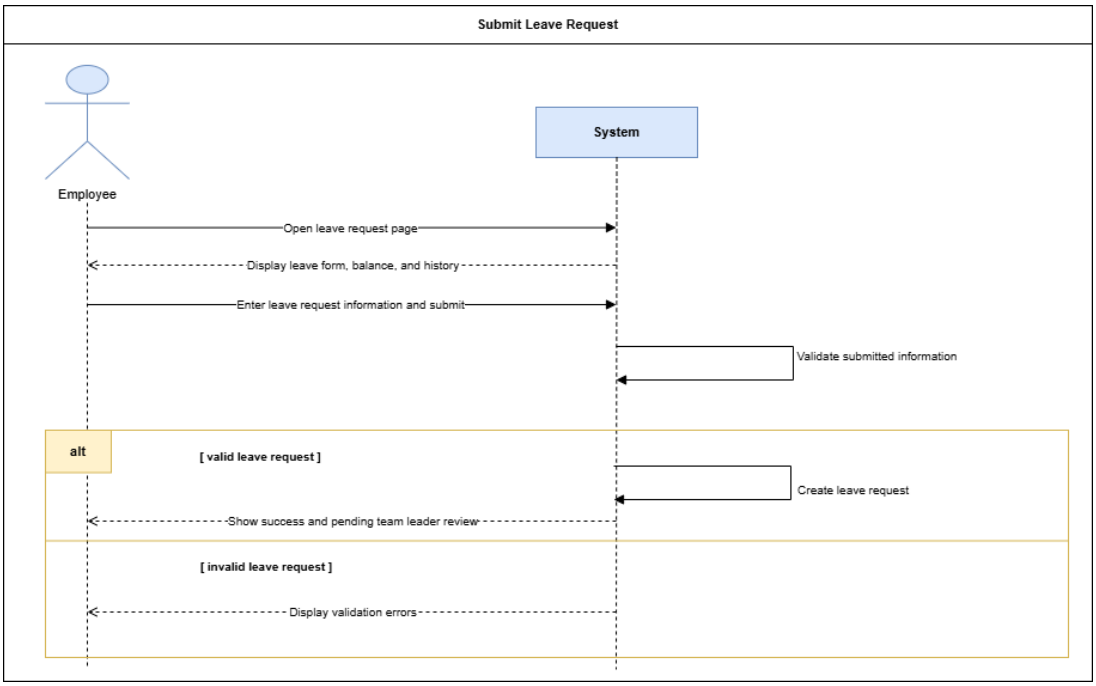


Figure 55: System Sequence Diagram – Submit Leave Request

Edit or Cancel Pending Leave

This diagram shows how an employee updates or cancels a leave request while it is still pending team leader review.

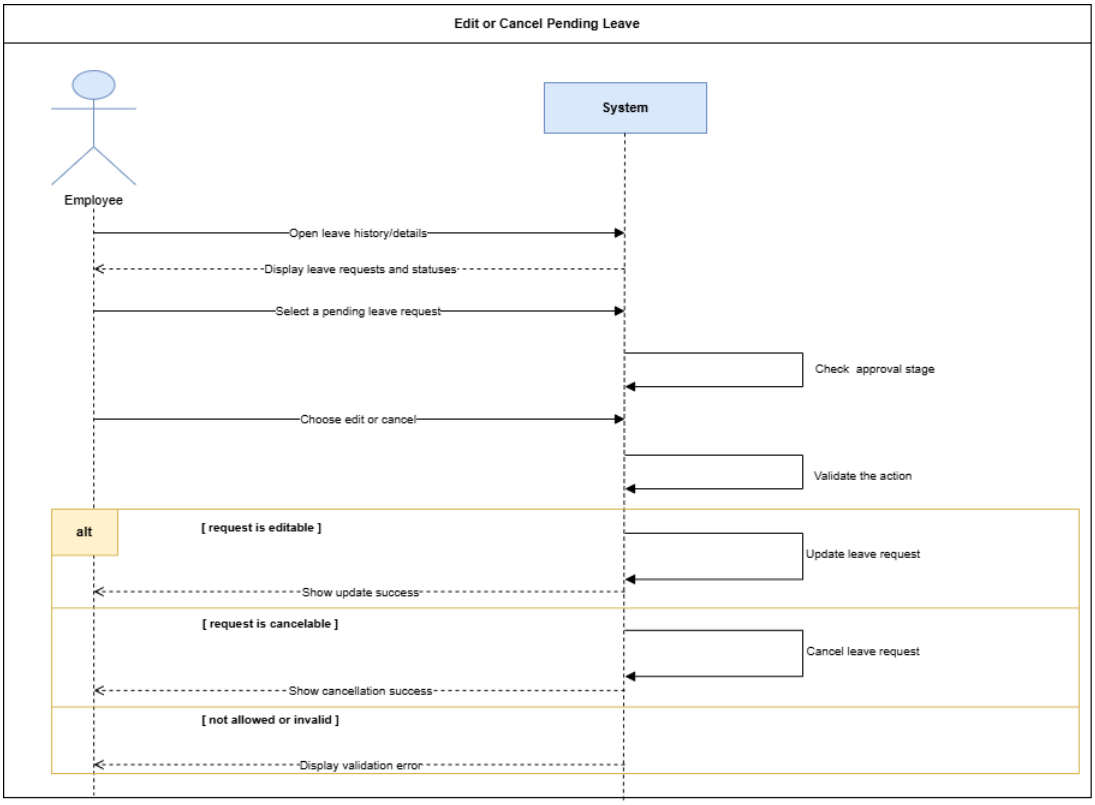


Figure 56: System Sequence Diagram – Edit or Cancel Pending Leave

Use Case Diagram – Leave Approval Workflow

This diagram presents the approval responsibilities of the team leader and HR manager, including team review, approval, rejection, HR final decision, leave records, and calendar consultation.

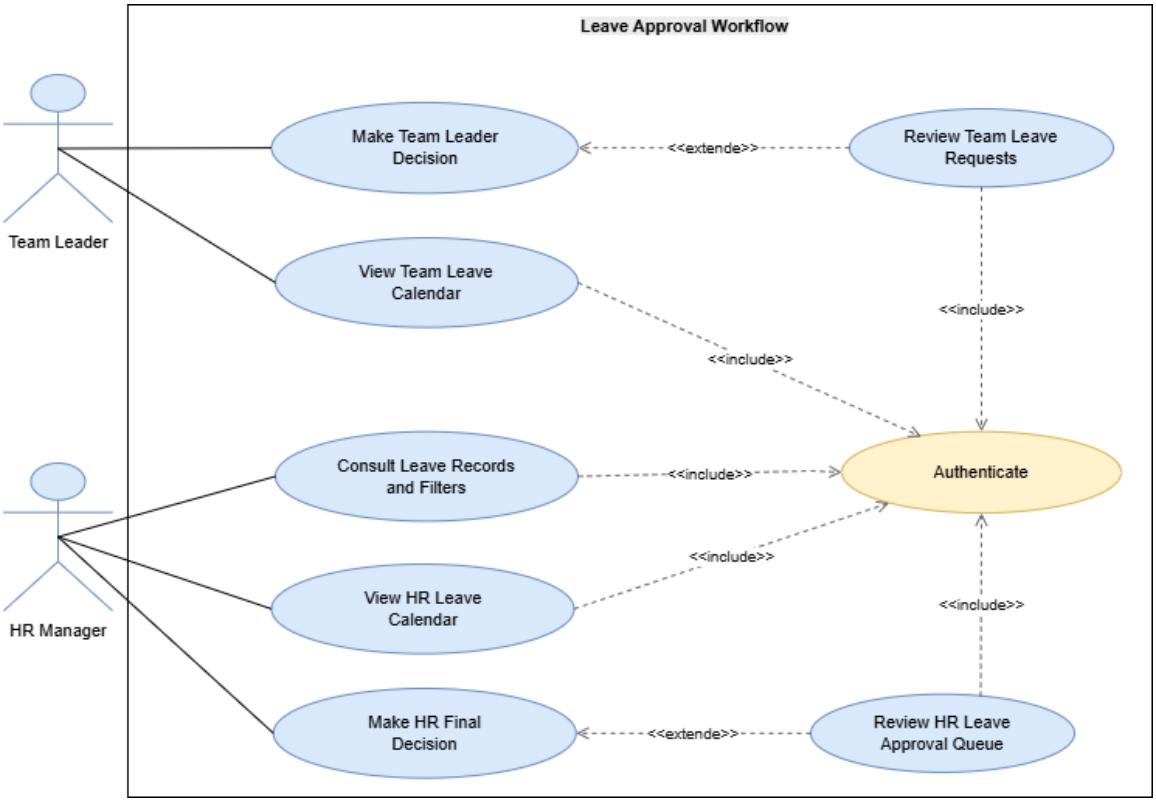


Figure 57: Use Case Diagram – Leave Approval Workflow

Team Leader Leave Decision

This diagram shows how a team leader approves or rejects a team member's leave request with a traceable decision.

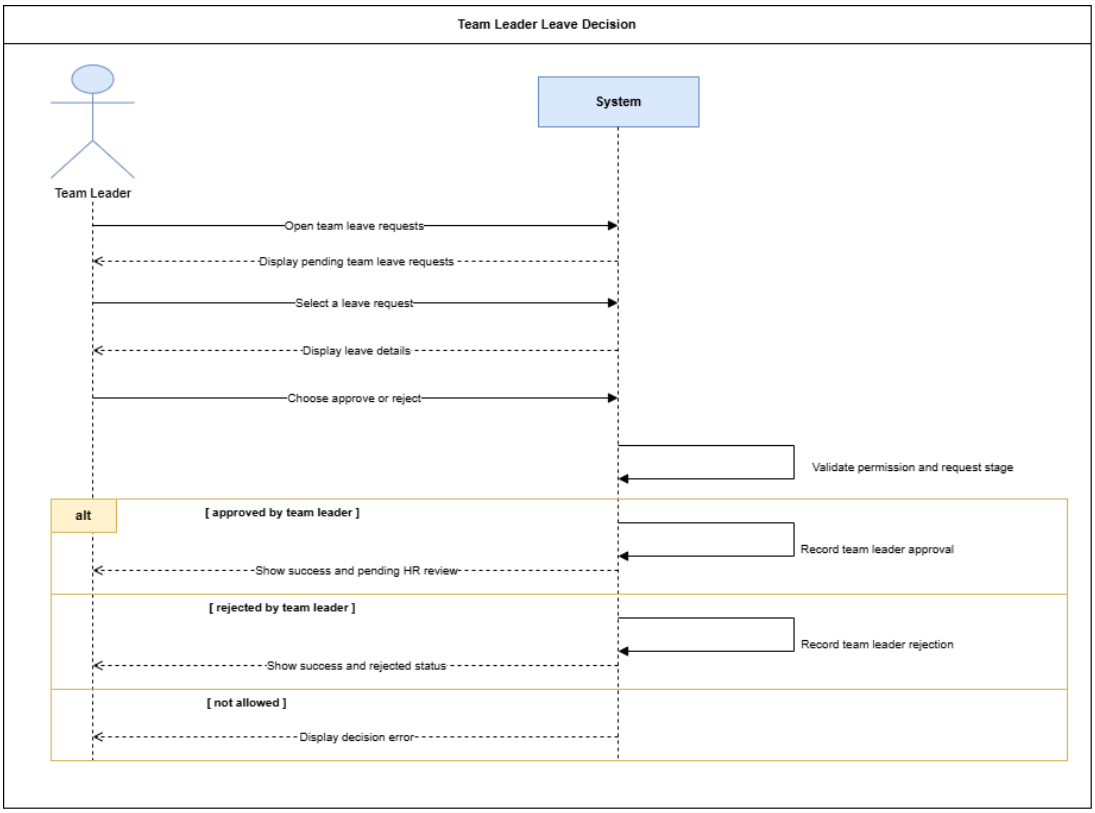


Figure 58: System Sequence Diagram – Team Leader Leave Decision

HR Final Leave Decision

This diagram shows how HR makes the final decision after team leader approval and updates the leave workflow accordingly.

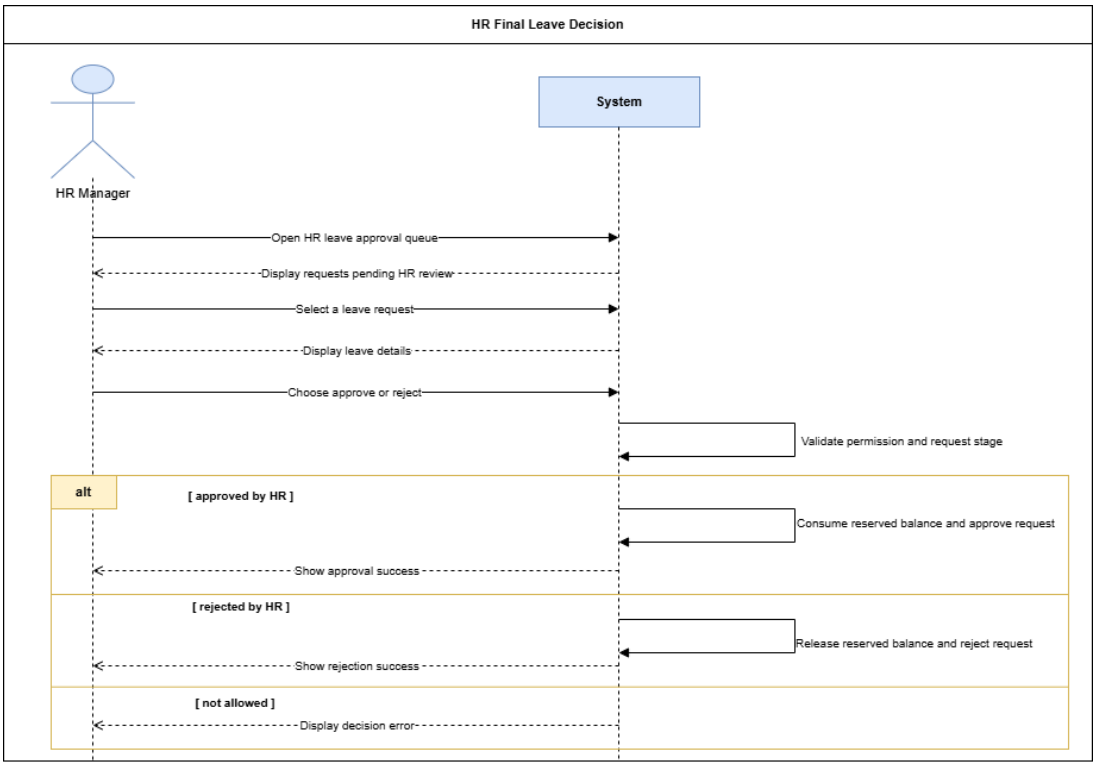


Figure 59: System Sequence Diagram – HR Final Leave Decision

Analysis

This section summarizes the main business concepts and lifecycle rules involved in the leave-management workflow.

Domain Model – Sprint 2 Leave Management

This diagram shows the main business concepts used for leave request management, including users, teams, leave requests, leave types, balances, public holidays, request history, and approval decisions.

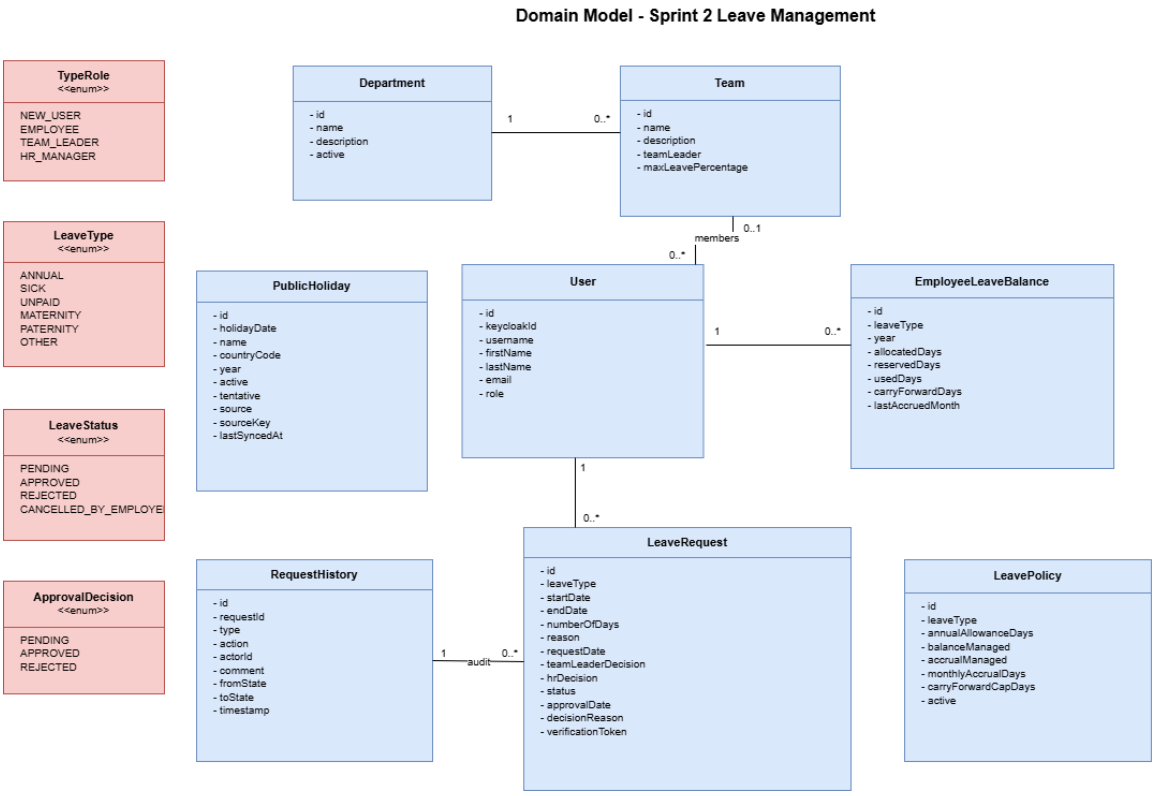


Figure 60: Domain Model – Sprint 2 Leave Management

Participant Class Diagrams

These diagrams identify the main frontend and backend participants involved in the leave-management workflow, following the main responsibility stages of the sprint: employee leave request management, team leader review, and HR final decision.

Participant Class Diagram – Employee Leave Request Management

This diagram identifies the main frontend and backend participants involved in employee leave submission, validation, balance consultation, and pending request update or cancellation.

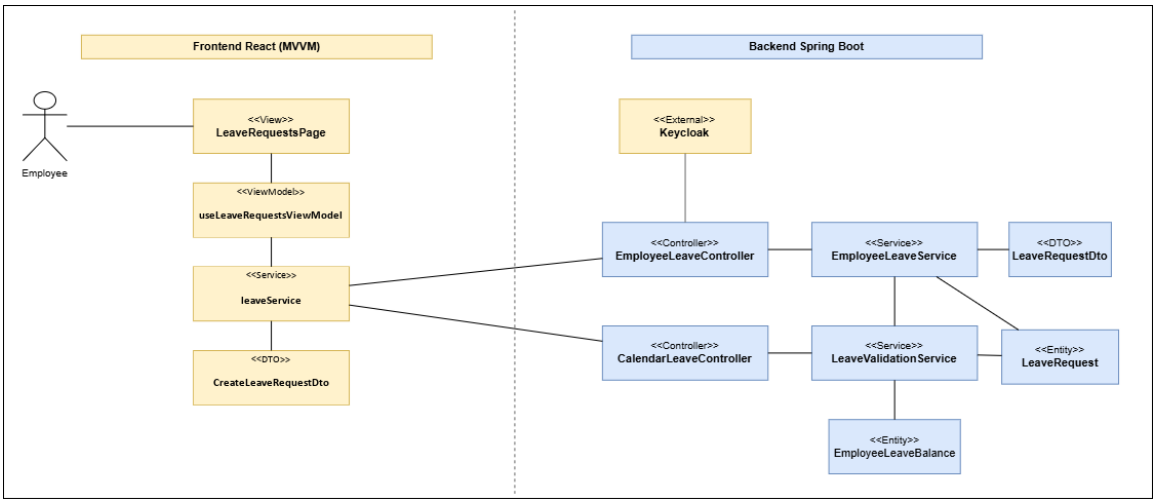


Figure 61: Participant Class Diagram – Employee Leave Request Management

Participant Class Diagram – Team Leader Leave Review

This diagram presents the participants involved in team leader leave review, including team request consultation, approval or rejection, and decision traceability.

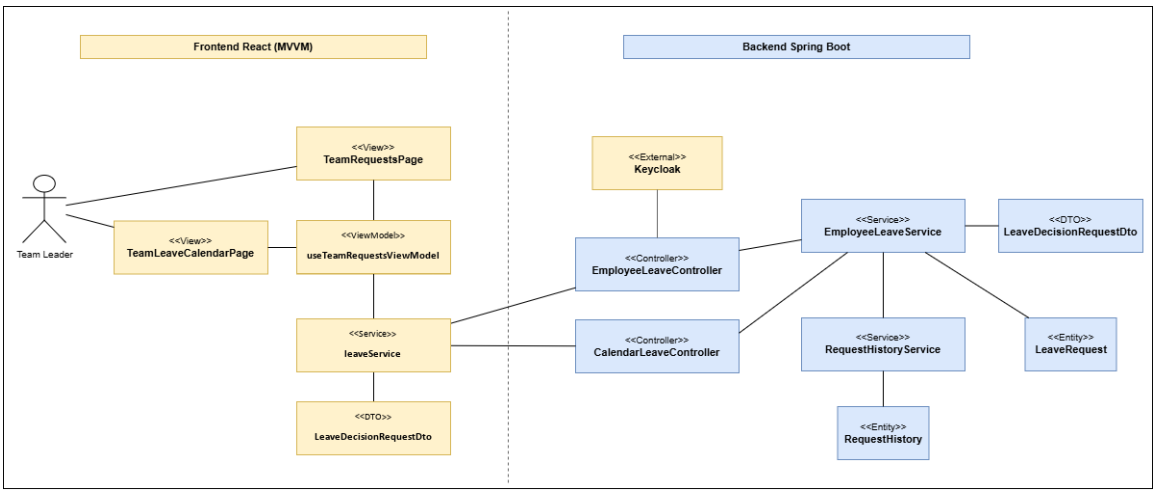


Figure 62: Participant Class Diagram – Team Leader Leave Review

Participant Class Diagram – HR Final Leave Decision

This diagram shows the participants involved in the HR final decision workflow, including approval queue consultation, final approval or rejection, and balance update.

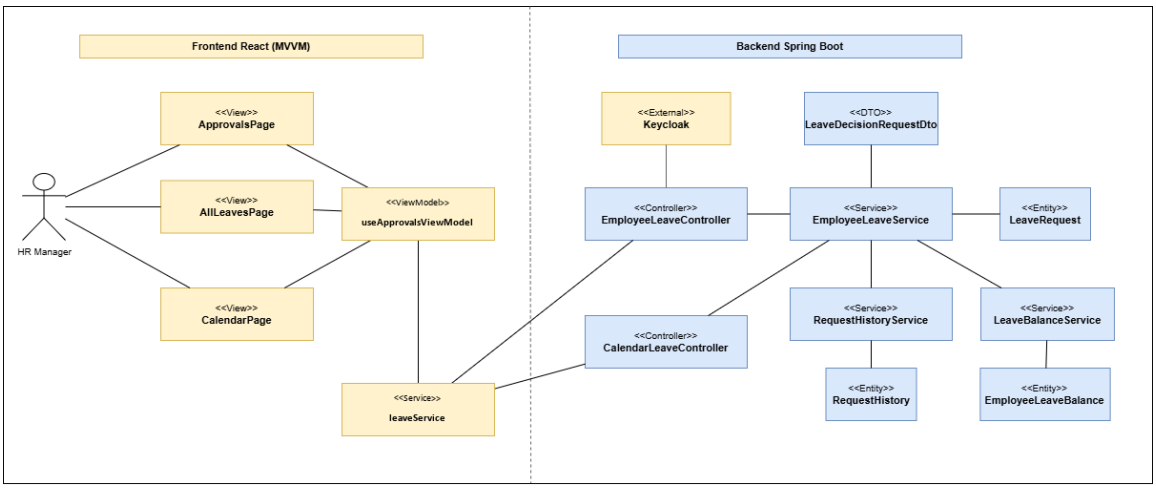


Figure 63: Participant Class Diagram – HR Final Leave Decision

State Diagrams

This section presents the lifecycle of a leave request using the real workflow states and stages of Sprint 2.

State Diagram – Leave Request Lifecycle

This diagram shows the leave request lifecycle from submission to team leader review, HR final decision, approval, rejection, or employee cancellation.

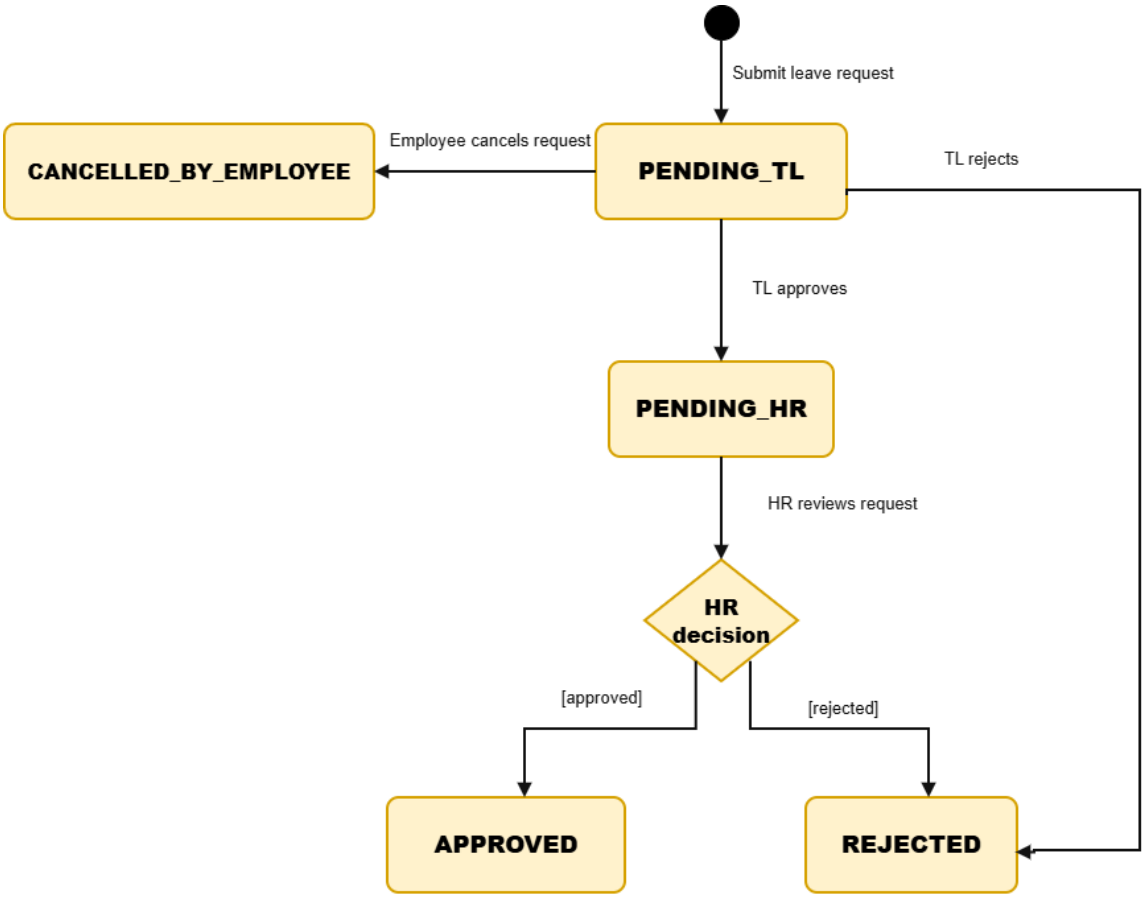


Figure 64: State Diagram – Leave Request Lifecycle

Design Class Diagrams

These diagrams present the main classes supporting Sprint 2 leave-management features.

Design Class Diagram – Leave Request Creation and Validation

This diagram illustrates the main classes used to create leave requests and apply validation rules before persistence.

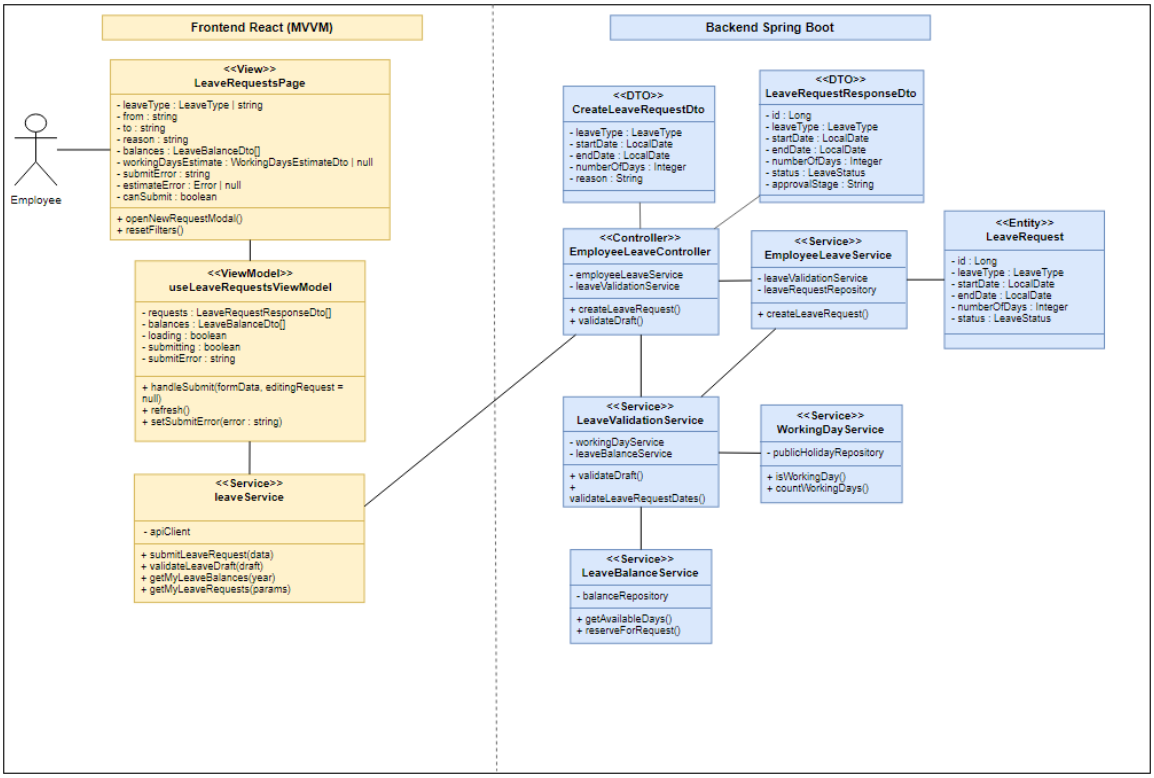


Figure 65: Design Class Diagram – Leave Request Creation and Validation

Design Class Diagram – Team Leader Leave Review

This diagram presents the main frontend and backend classes used for team leader leave consultation, approval, rejection, history recording, and balance release when needed.

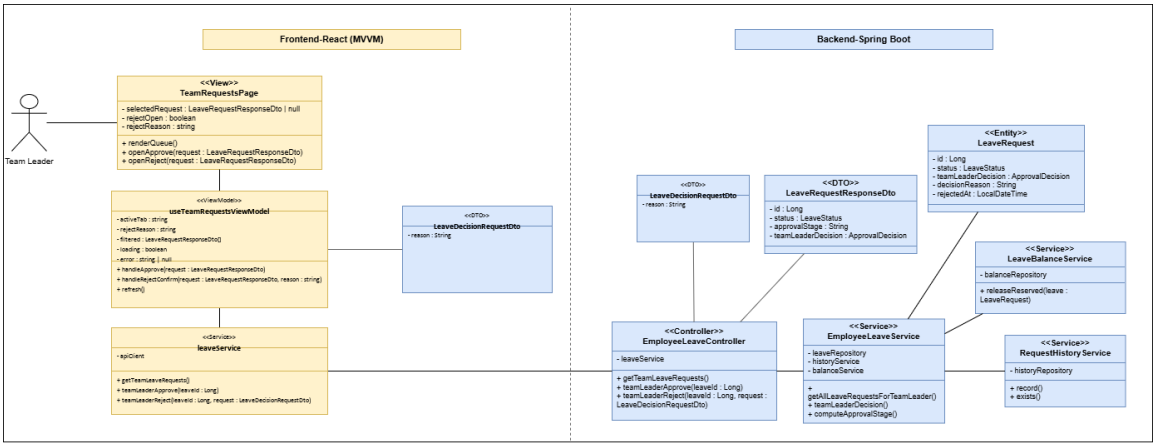


Figure 66: Design Class Diagram – Team Leader Leave Review

Design Class Diagram – HR Final Leave Decision

This diagram shows the classes used by HR to make the final leave decision, update the request status, record history, and consume or release reserved balance.

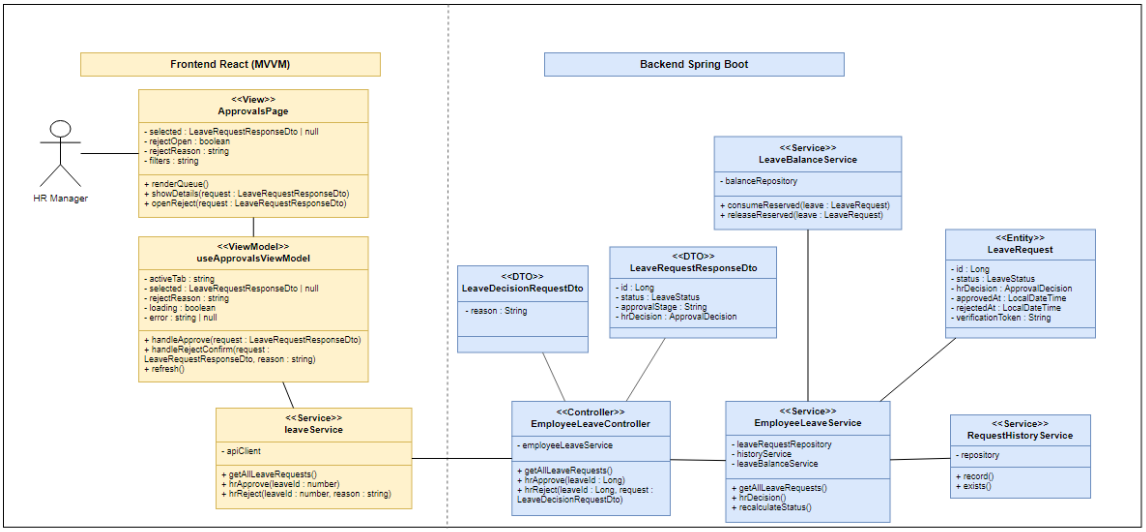


Figure 67: Design Class Diagram – HR Final Leave Decision

Design Class Diagram – Working Day, Holiday, and Balance Management

This diagram presents the classes used to calculate deducted working days, exclude weekends and public holidays, manage leave policy, and check available balance.

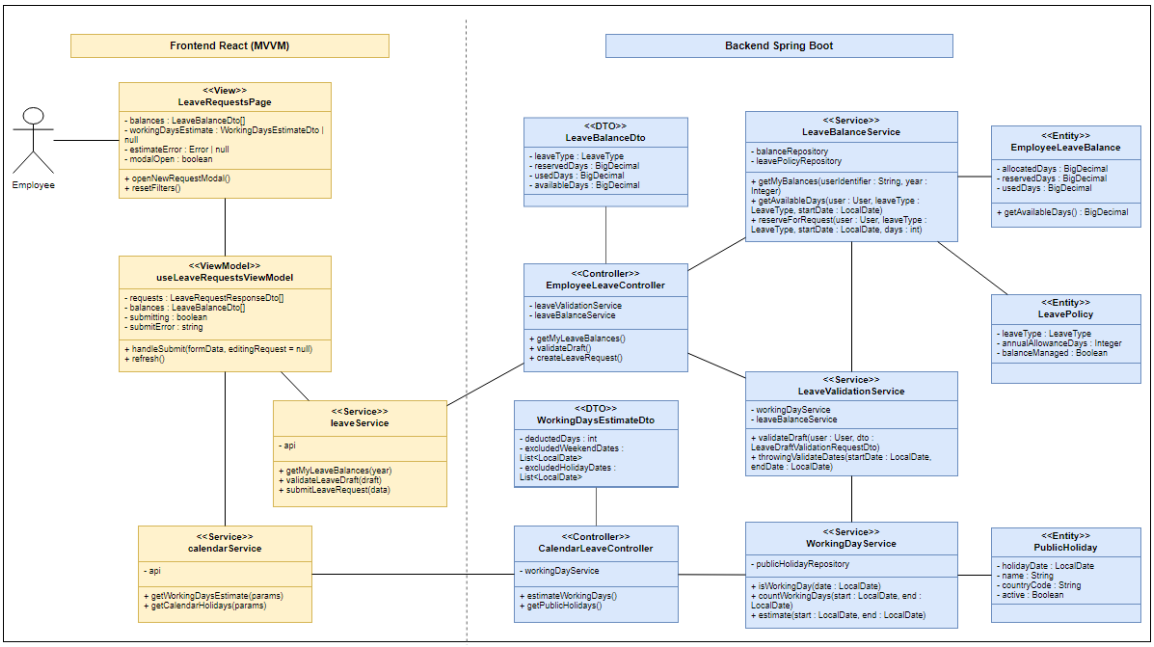


Figure 68: Design Class Diagram – Working Day, Holiday, and Balance Management

Conception Sequence Diagrams

These diagrams detail the internal frontend and backend interactions behind the main Sprint 2 scenarios: leave submission, pending leave update or cancellation, team leader review, and HR final review.

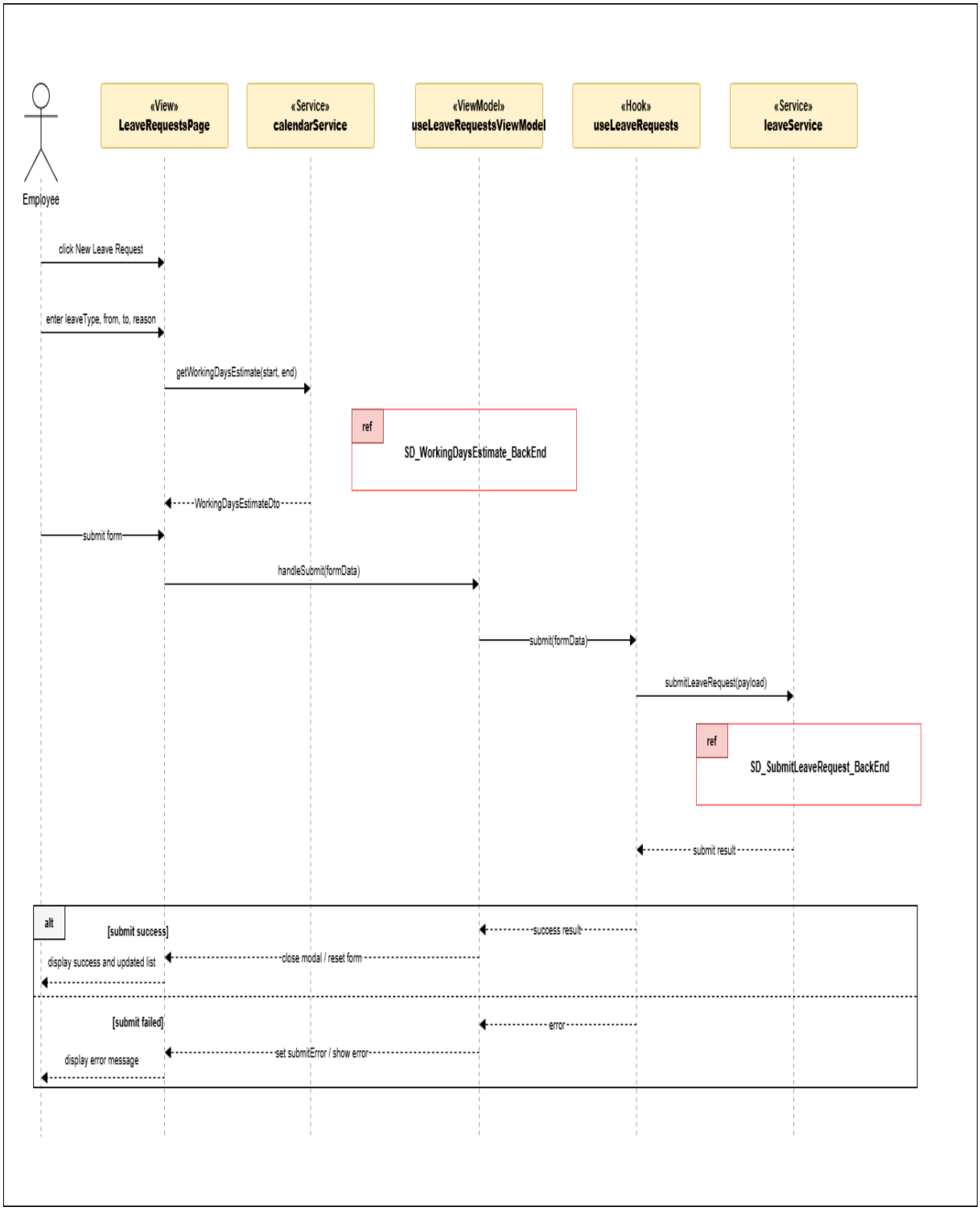


Figure 69: Conception Sequence Diagram – Submit Leave Request Frontend

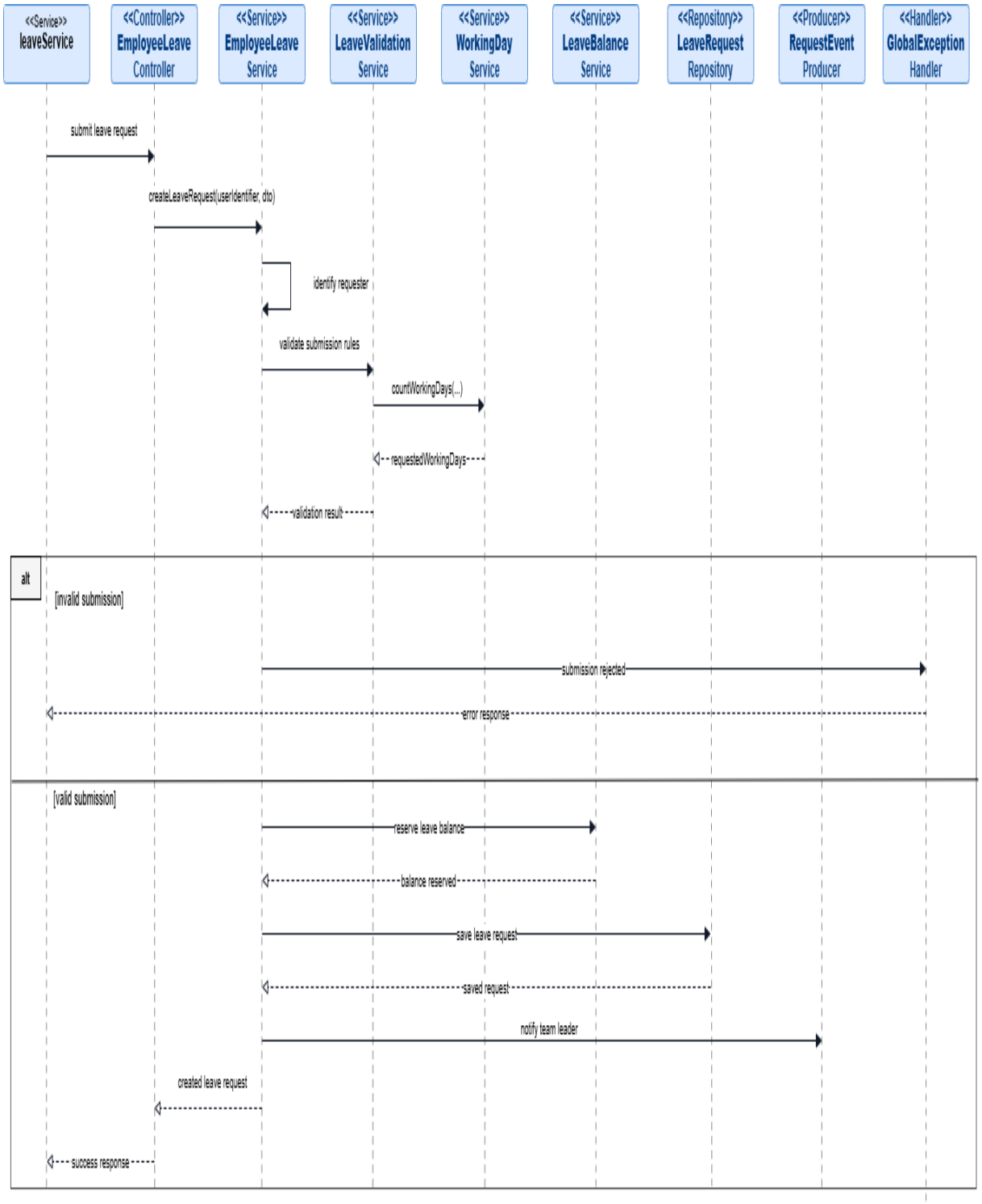


Figure 70: Conception Sequence Diagram – Submit Leave Request Backend

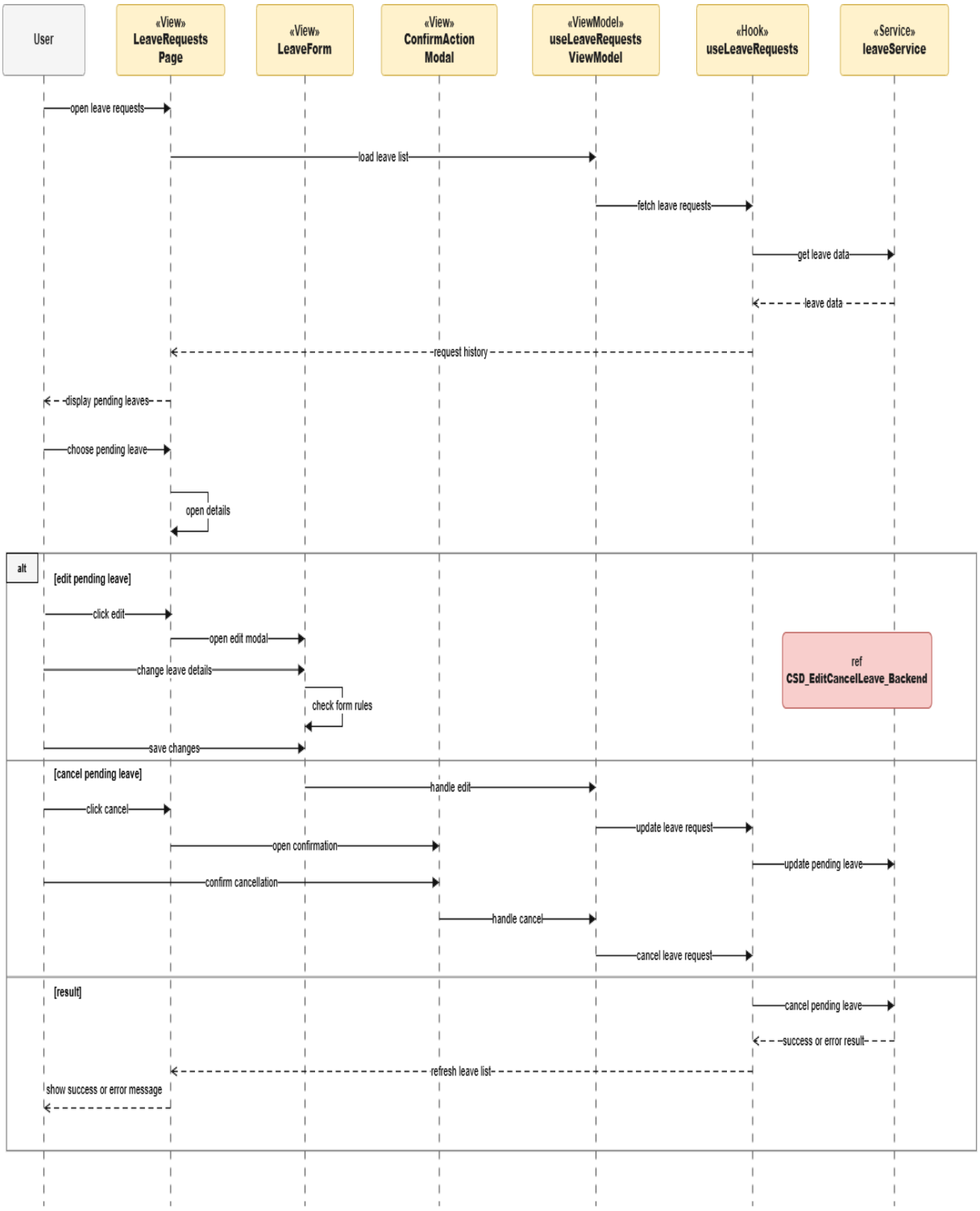


Figure 71: Conception Sequence Diagram – Edit or Cancel Pending Leave Frontend

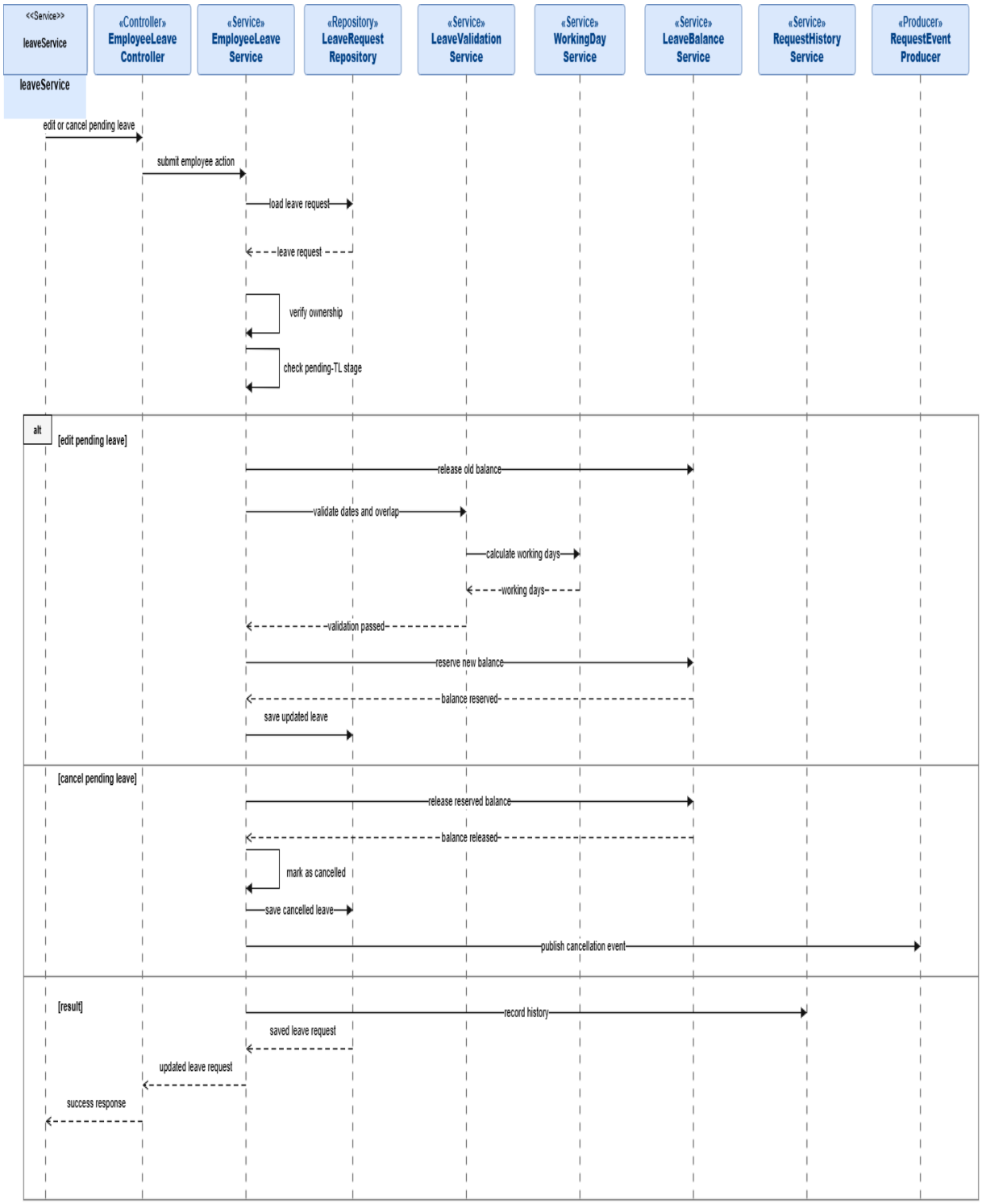


Figure 72: Conception Sequence Diagram – Edit or Cancel Pending Leave Backend

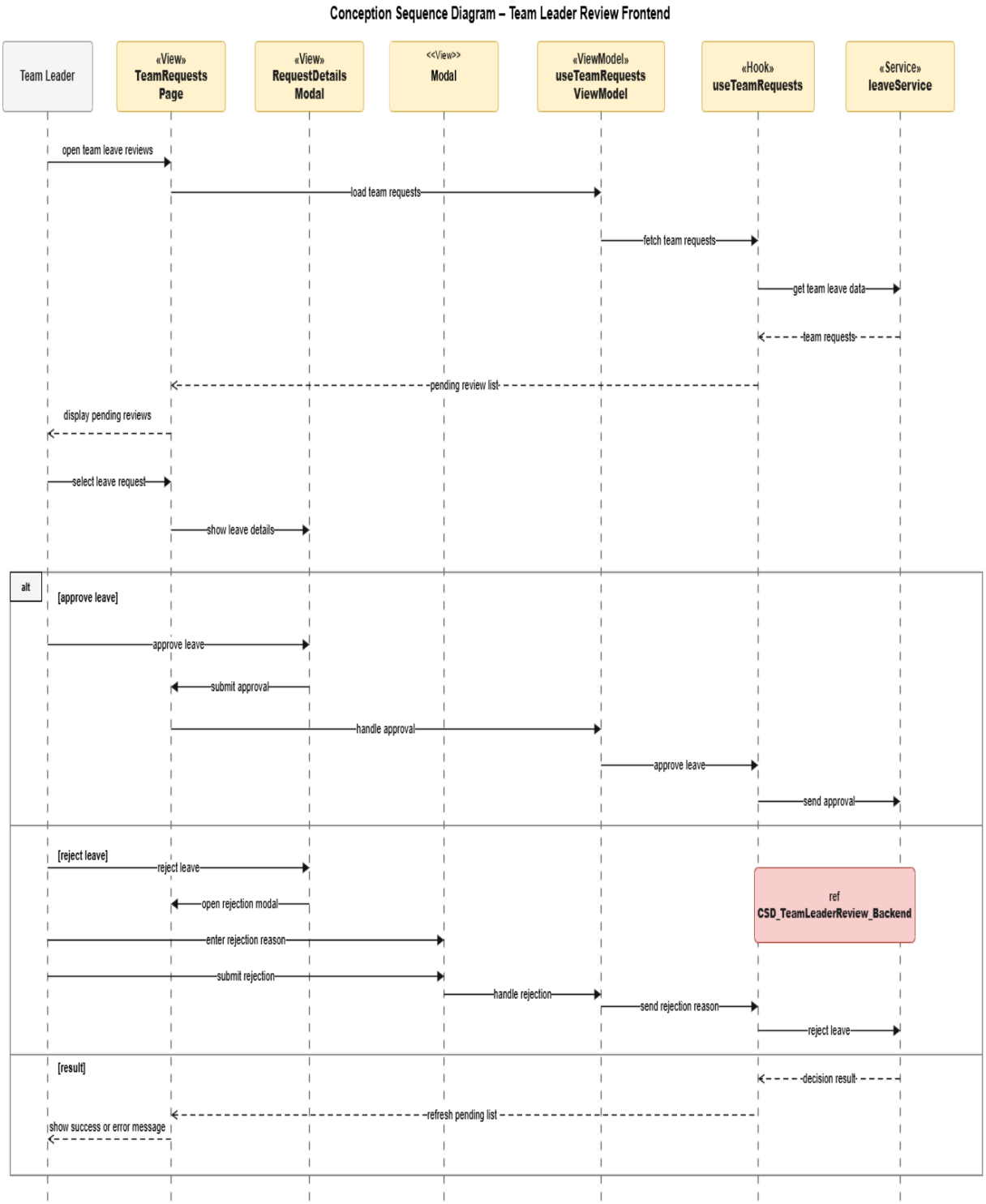


Figure 73: Conception Sequence Diagram – Team Leader Review Frontend

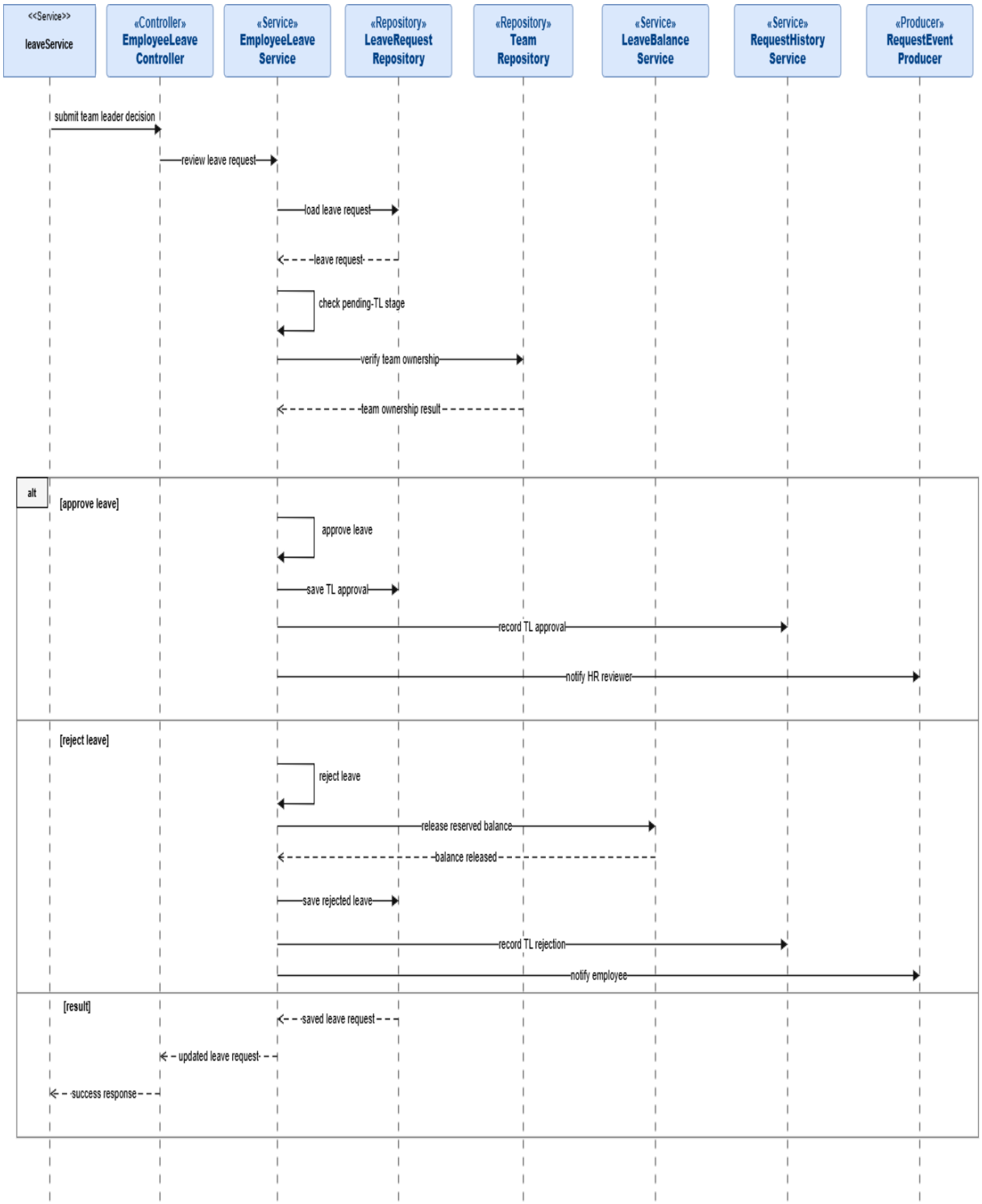


Figure 74: Conception Sequence Diagram – Team Leader Review Backend

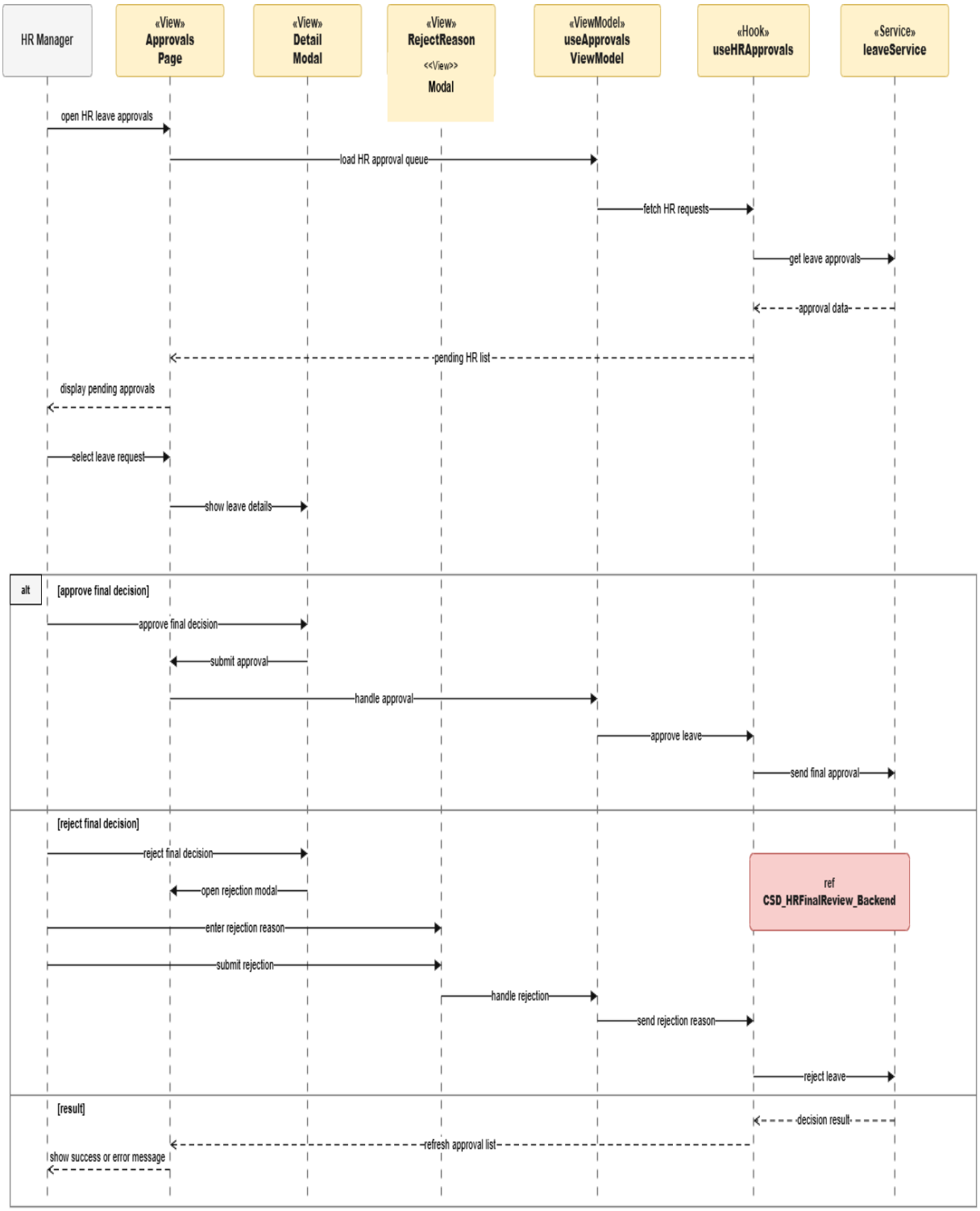


Figure 75: Conception Sequence Diagram – HR Final Review Frontend

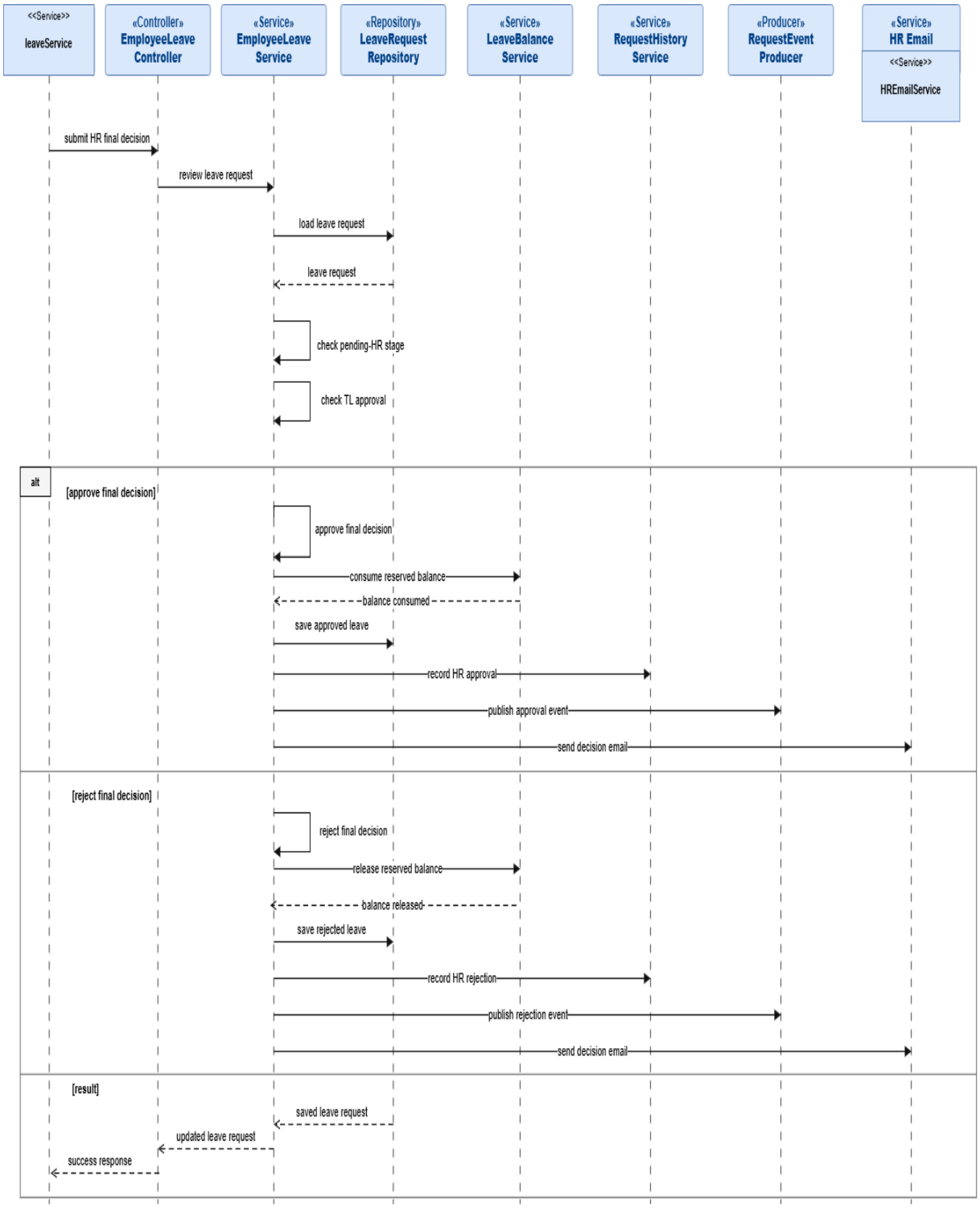


Figure 76: Conception Sequence Diagram – HR Final Review Backend

Realization

This section presents the main user interfaces implemented during Sprint 2.

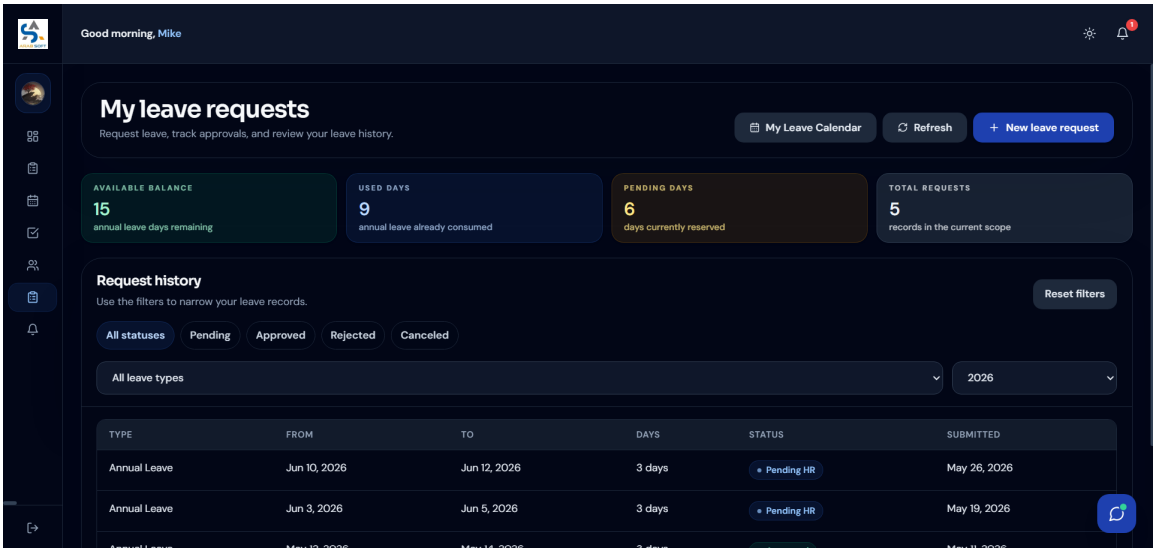


Figure 77: Employee Leave Requests Interface

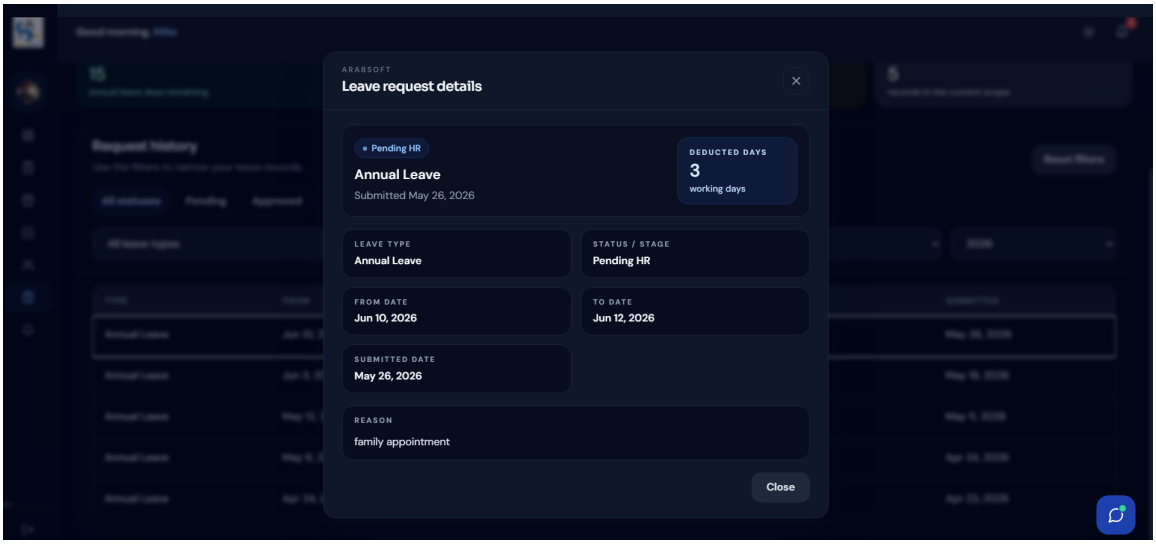


Figure 78: Leave Request Details Interface

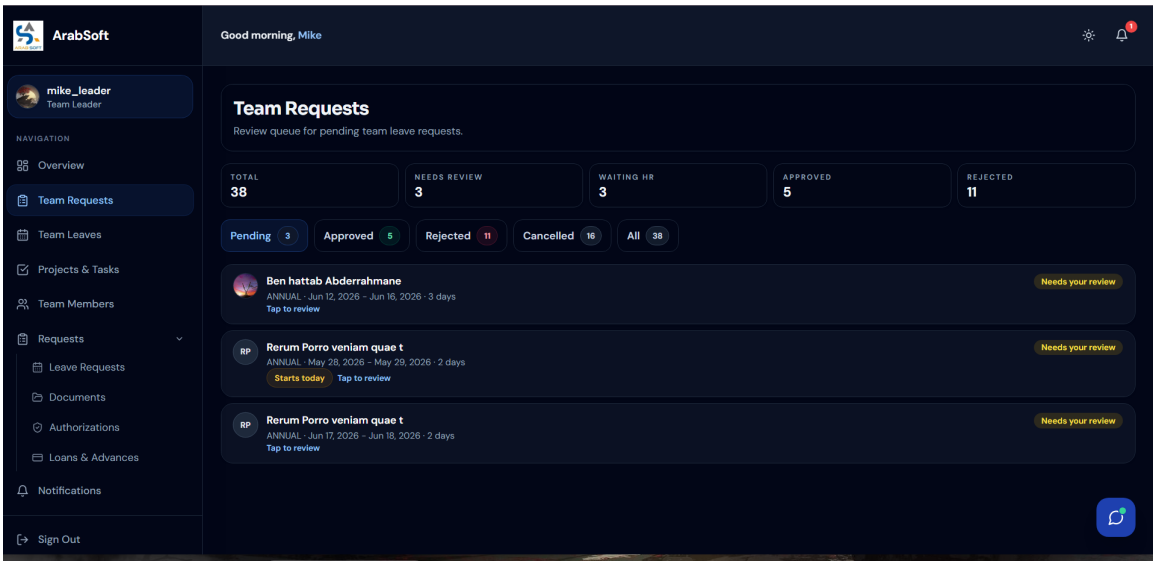


Figure 79: Team Leader Leave Review Interface

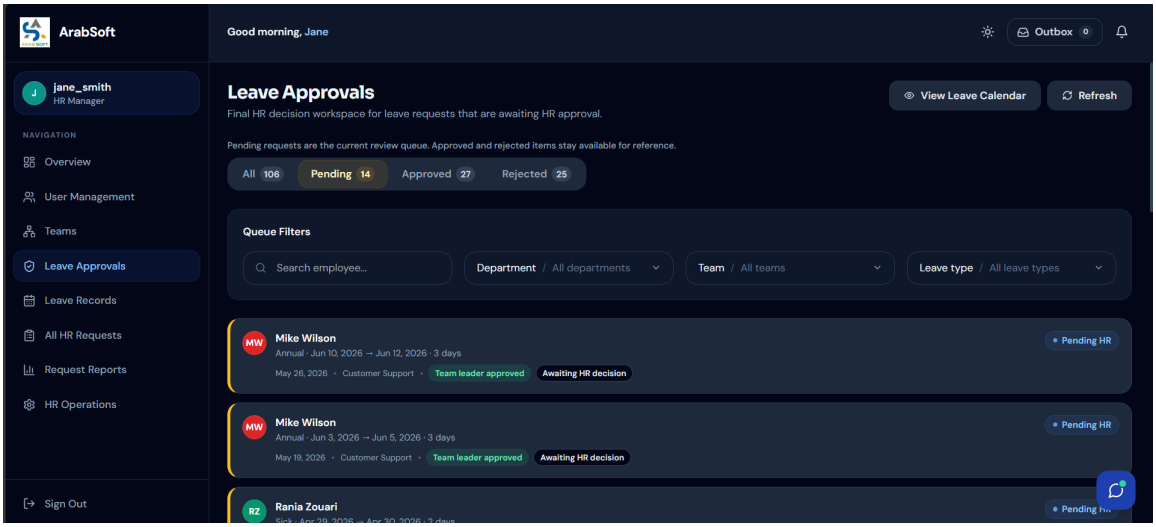


Figure 80: HR Final Leave Decision Interface

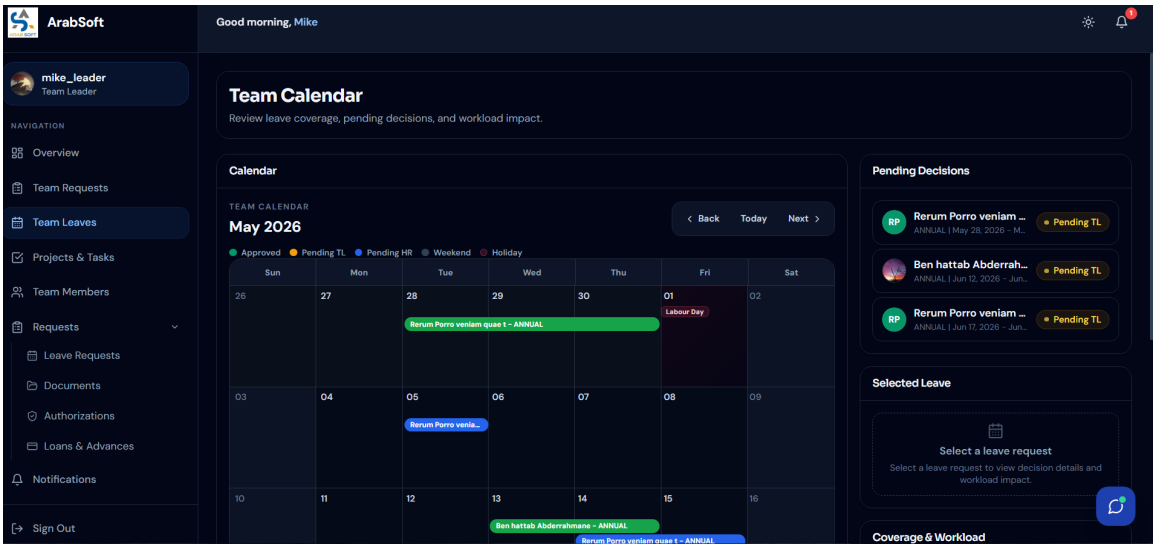


Figure 81: Leave Calendar Interface

5. Sprint Review

At the end of Sprint 2, the leave-management workflow was reviewed as the first complete administrative process of the platform. The sprint enabled employees to submit, update, cancel, and track leave requests while applying working-day validation, public-holiday exclusion, overlap prevention, and leave-balance control. It also introduced the two-level approval workflow, where team leaders review requests first and HR managers make the final decision. In addition, role-aware calendars, filters, notifications, and approved leave PDF generation improved visibility and traceability across the workflow.

6. Sprint Retrospective

What went well	What needs improvement
The leave workflow became fully operational, from employee submission to team leader review and HR final decision, with balance control, status tracking, and approved leave PDF generation.	The main improvement point is to simplify complex leave feedback when several rules apply at the same time, especially working-day validation, public holidays, overlaps, and calendar visibility.

Table 5: Sprint Retrospective – Sprint 2

Conclusion

This chapter covered Release 1 of the ArabSoft internal platform, structured into two complementary sprints. Sprint 1 established the secure access foundation of the system through registration, authentication, password recovery, profile management, HR account setup, role assignment, and account activation control. Sprint 2 delivered the leave-management module, including leave submission, update and cancellation, working-day validation, leave balance control, team leader review, HR final decision, calendars, notifications, and approved leave reporting. These two sprints provided the first operational version of the platform, combining secure user access with a complete administrative workflow. The next chapter will present Release 2, which extends the platform with additional administrative requests and HR management features.

Chapter 4

Release 2 – Workflow Management

Introduction

This chapter presents Release 2 of the ArabSoft internal platform. It focuses on workflow management through two main sprints: Sprint 3, dedicated to administrative requests, and Sprint 4, dedicated to HR management and calendar supervision. The chapter follows the same structure as the previous release by presenting the sprint goals, backlog, UML diagrams, implementation views, review, and retrospective.

I. Release Organization

Release 2 is organized around workflow management. It first covers administrative request processing through document, loan, and authorization workflows, then extends the platform with HR user supervision, organization setup, team management, and leave calendar consultation.

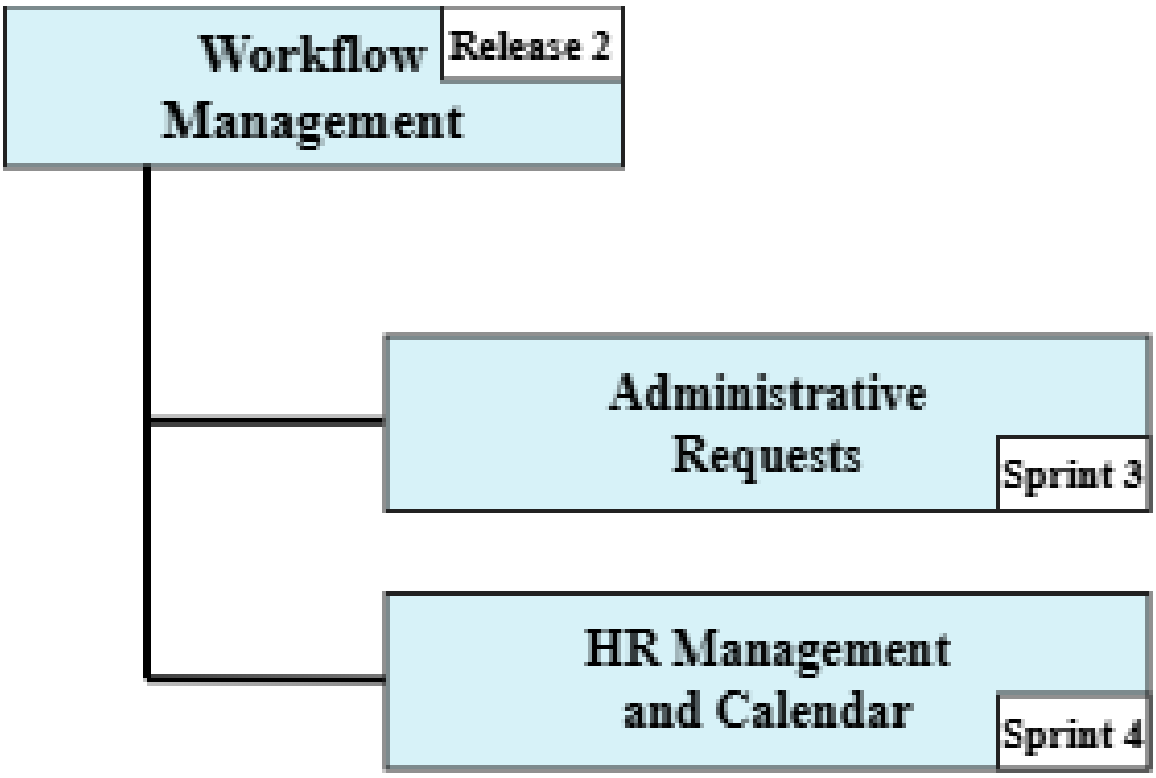


Figure 82: Release 2 Organization

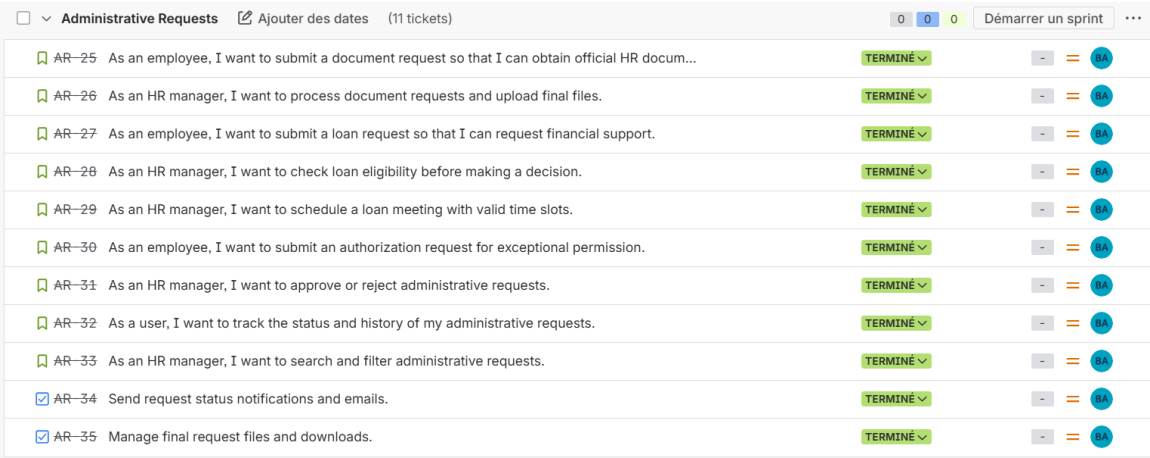
II. Sprint 3: Administrative Requests

1. Sprint Goal

This sprint covers the administrative-request workflow of Release 2. It focuses on document requests, loan requests, authorization requests, request tracking, and the related HR decisions.

2. Visual Sprint Journal

The sprint journal will capture the tracked user stories and their validation status in Jira.



Administrative Requests	Ajouter des dates (11 tickets)	0 0 0	Démarrer un sprint
~~AR-25~~	As an employee, I want to submit a document request so that I can obtain official HR docum...	TERMINÉ	
~~AR-26~~	As an HR manager, I want to process document requests and upload final files.	TERMINÉ	
~~AR-27~~	As an employee, I want to submit a loan request so that I can request financial support.	TERMINÉ	
~~AR-28~~	As an HR manager, I want to check loan eligibility before making a decision.	TERMINÉ	
~~AR-29~~	As an HR manager, I want to schedule a loan meeting with valid time slots.	TERMINÉ	
~~AR-30~~	As an employee, I want to submit an authorization request for exceptional permission.	TERMINÉ	
~~AR-31~~	As an HR manager, I want to approve or reject administrative requests.	TERMINÉ	
~~AR-32~~	As a user, I want to track the status and history of my administrative requests.	TERMINÉ	
~~AR-33~~	As an HR manager, I want to search and filter administrative requests.	TERMINÉ	
~~AR-34~~	Send request status notifications and emails.	TERMINÉ	
~~AR-35~~	Manage final request files and downloads.	TERMINÉ	

Visual Sprint Journal – Sprint 3

3. Sprint Backlog

The table below summarizes the Sprint 3 work packages prepared from the product backlog defined in Chapter 2.

Sprint	Stories	Tasks	Duration
Sprint 3	**Document request and processing**	Implement employee document request submission	2d
		Implement HR document request processing	
		Manage final document file upload and download	
	Loan request and eligibility	Implement employee loan request submission	2d
		Add HR loan eligibility verification	
		Display clear loan request status and feedback	
	Loan meeting scheduling	Implement loan meeting scheduling by HR	2d
		Validate meeting date and available time slots	
		Prevent scheduling conflicts when applicable	
	Authorization request management	Implement authorization request submission	2d
		Support short absence and equipment request workflows	
		Connect HR decision handling for authorizations	
	Request tracking, filters, notifications, and final files	Display administrative request history and status	2d
		Add HR search and filtering for requests	
		Send request status notifications and emails	
		Manage final files for completed requests	

Table 6: Sprint Backlog – Sprint 3

4. Sprint Implementation

Requirements Analysis

The following analysis describes the business rules that govern each request type handled in Sprint 3, covering document requests, loan requests and meetings, authorization requests, and the HR processing and tracking logic behind each one.

Business Rules

Employees can request official HR documents such as employment certificates, salary certificates, work reference letters, experience certificates, custom administrative letters, and leave balance statements. The leave balance statement is generated automatically after HR approval, while the other employee-requestable documents require HR to upload a final file before download. Contract copies are not submitted through the employee document request flow; they remain HR-managed through stored employee documents. Pending document requests may be approved or rejected by HR, and rejected requests require a reason.

Loan requests require a type, amount, and reason. Eligibility depends on salary, seniority, and the absence of an existing active loan request. HR validation also includes meeting scheduling rules, and each decision is stored in the request history with notifications or emails.

Authorization requests in the active workflow are limited to time permission and equipment request types. Time permission must respect working hours and avoid leave overlap, while equipment requests require a type, reason, and start date. HR can approve or reject each pending request with a required rejection reason.

Each request is traceable through its detail view, history records, status updates, and notification events. Employee pages show personal request progress, and HR pages support search and filtering.

Use Case Diagrams

These diagrams present the global administrative-request scope of Sprint 3 and the HR processing viewpoint for document, loan, and authorization requests.

Use Case Diagram – Sprint 3 Administrative Requests

This diagram presents the overall administrative-request scope of Sprint 3.

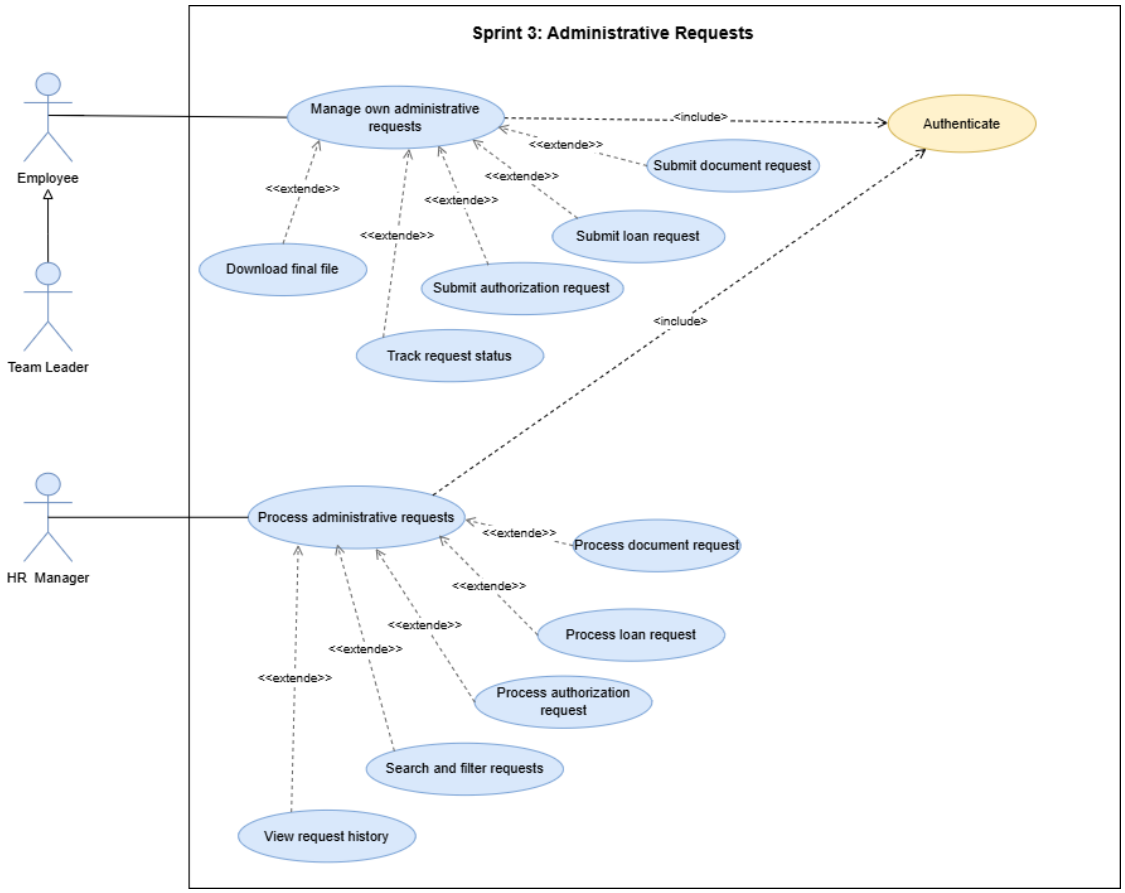


Figure 83: Use Case Diagram – Sprint 3 Administrative Requests

Use Case Diagram – HR Administrative Request Processing

This diagram summarizes HR processing, validation, and final decision responsibilities.

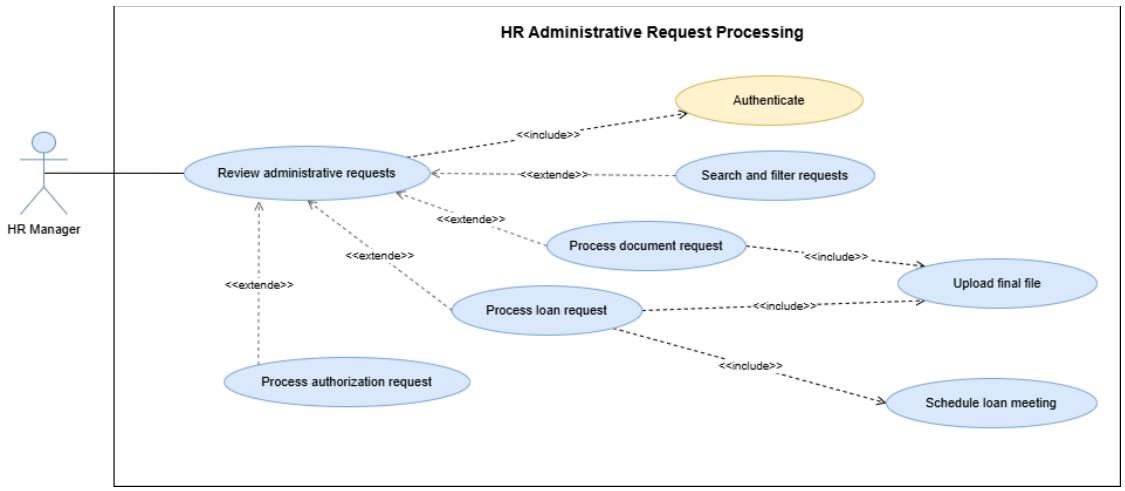


Figure 84: Use Case Diagram – HR Administrative Request Processing

System Sequence Diagrams

These diagrams describe the main request submissions, HR processing steps, and scheduling interactions in Sprint 3.

System Sequence Diagram – Submit Document Request

This diagram shows how a user, either an employee or a team leader using employee-side request actions, submits a document request and receives validation feedback.

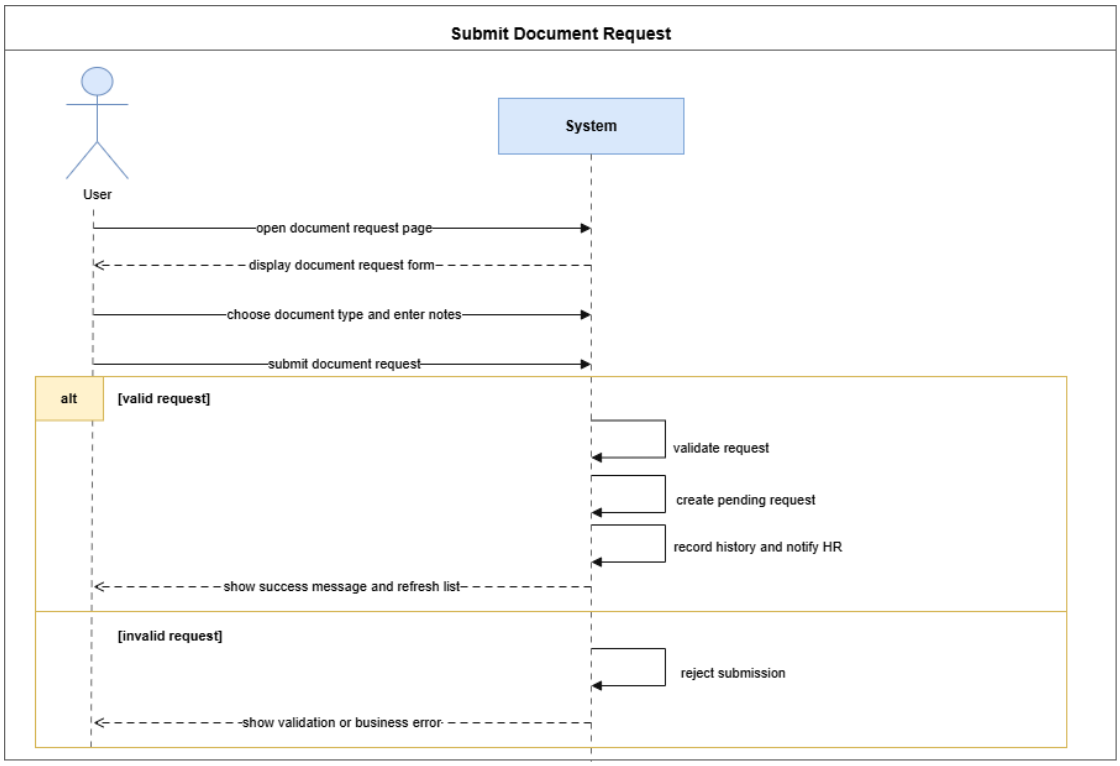


Figure 85: System Sequence Diagram – Submit Document Request

System Sequence Diagram – Process Document Request

This diagram shows how HR processes a pending document request and manages the final file.

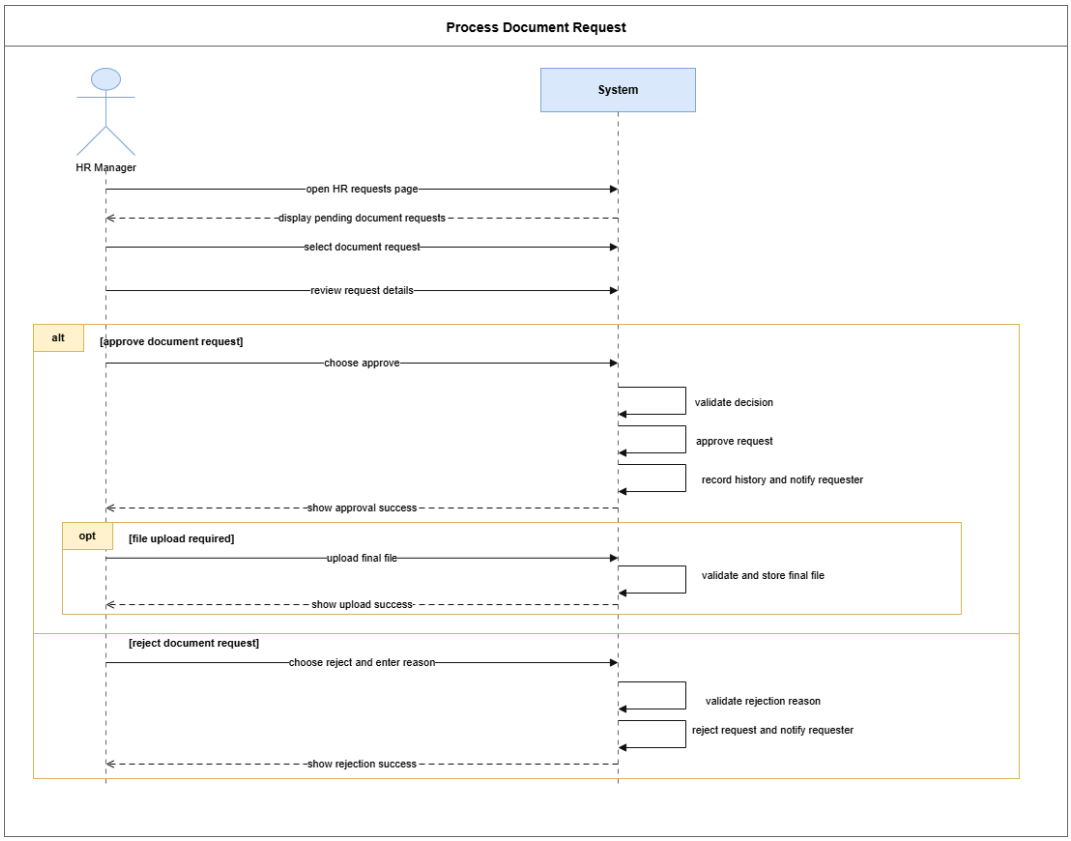


Figure 86: System Sequence Diagram – Process Document Request

System Sequence Diagram – Submit Loan Request

This diagram shows how a user, either an employee or a team leader using employee-side request actions, submits a loan request with the required request data.

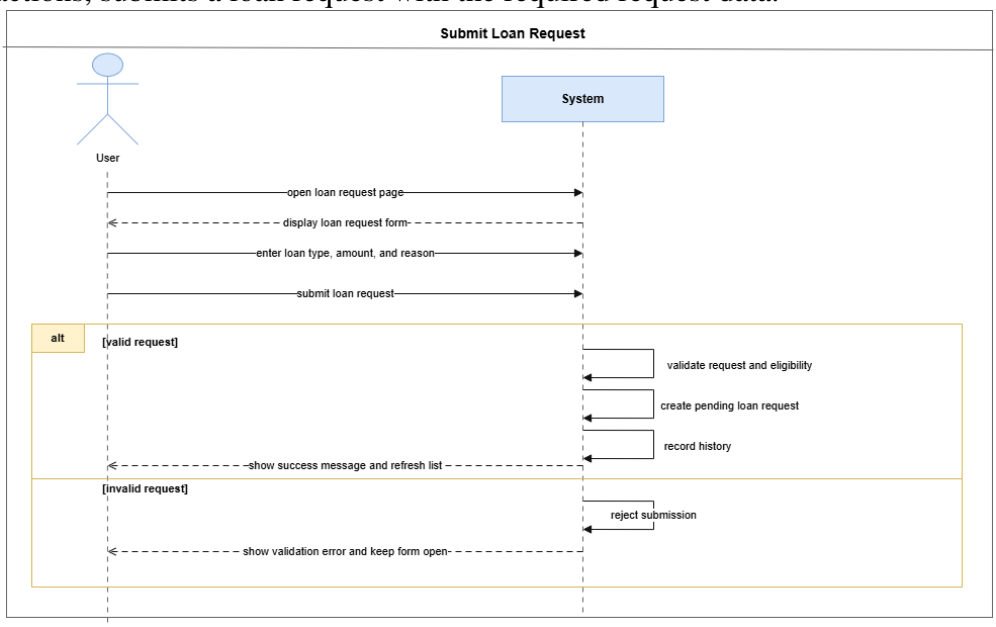


Figure 87: System Sequence Diagram – Submit Loan Request

System Sequence Diagram – Schedule Loan Meeting

This diagram shows how HR schedules a loan meeting under the workflow constraints.

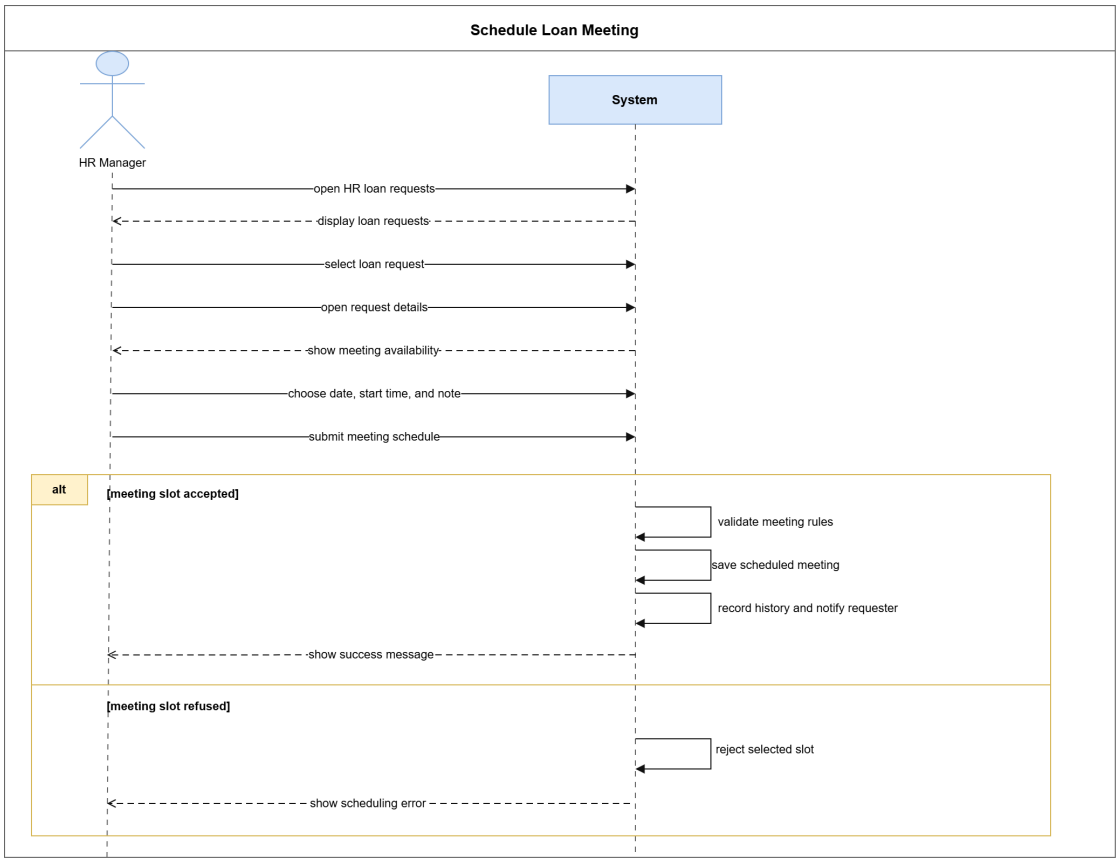


Figure 88: System Sequence Diagram – Schedule Loan Meeting

System Sequence Diagram – Submit Authorization Request

This diagram shows how a user, either an employee or a team leader using employee-side request actions, submits a time permission or equipment request.

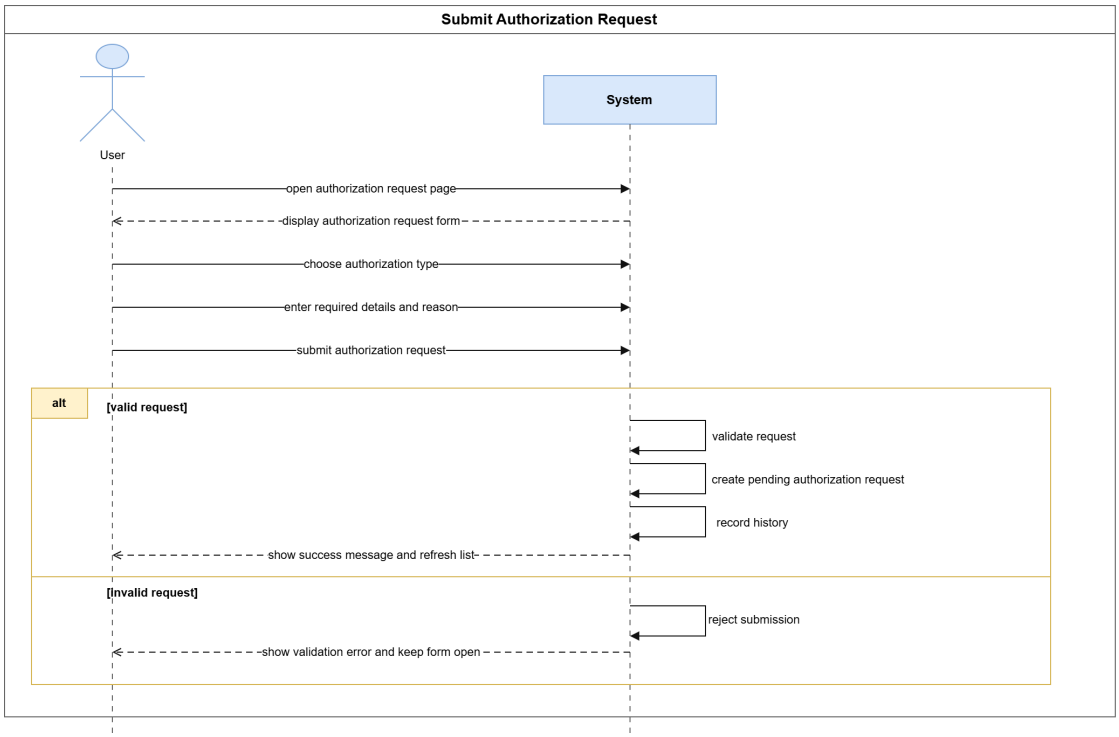


Figure 89: System Sequence Diagram – Submit Authorization Request

System Sequence Diagram – HR Administrative Request Decision

This diagram shows how HR approves or rejects a pending administrative request.

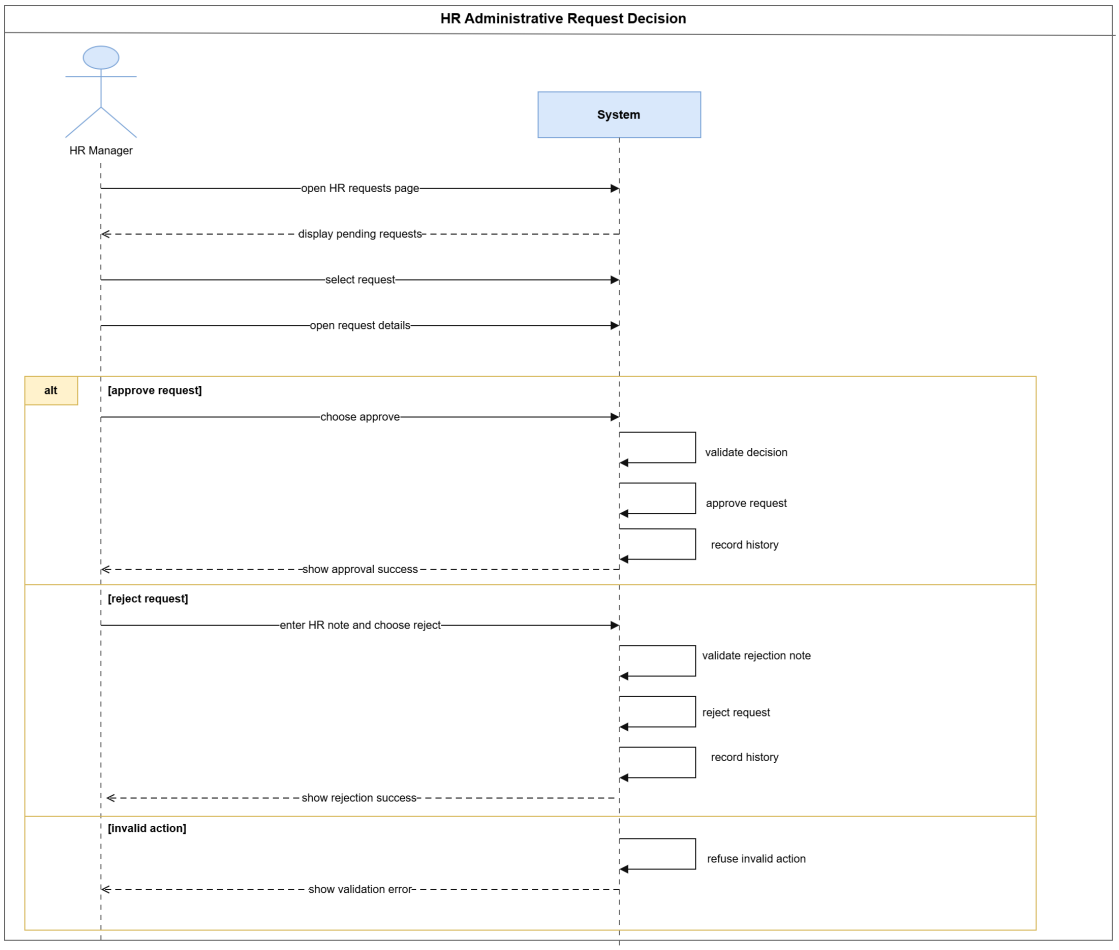


Figure 90: System Sequence Diagram – HR Administrative Request Decision

Analysis

The analysis covers the main domain concepts in Sprint 3: the request entities, decision traces, and final-file handling across the three request types.

Domain Model – Sprint 3 Administrative Requests

This diagram presents the main domain concepts used to manage Sprint 3 administrative requests, including document requests, loan requests, authorization requests, request history, stored employee documents, and the related status and type values.

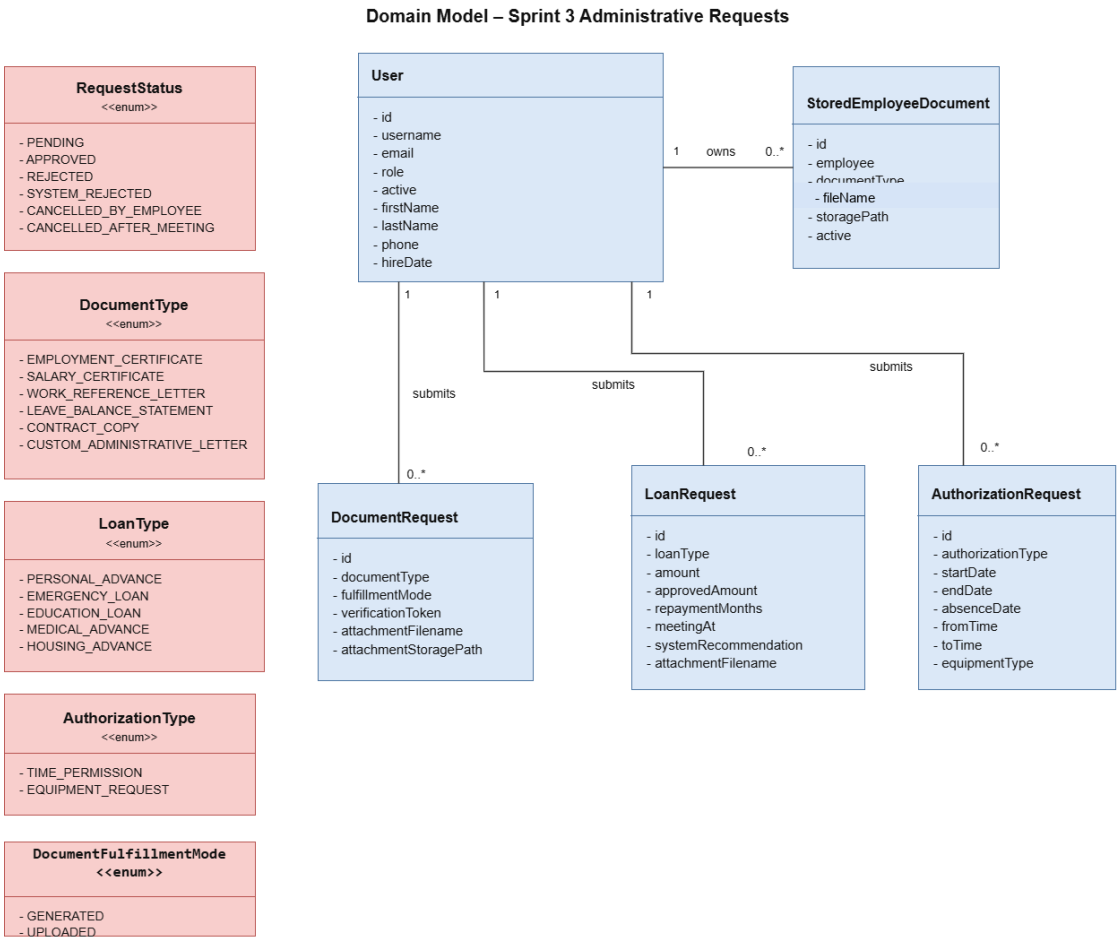


Figure 91: Domain Model – Sprint 3 Administrative Requests

Participant Class Diagrams

These diagrams identify the main frontend and backend participants involved in employee administrative request submission and HR request processing during Sprint 3.

Participant Class Diagram – Document Request Management

This diagram shows the participants involved in employee document request submission, draft validation, status tracking, and request history management.

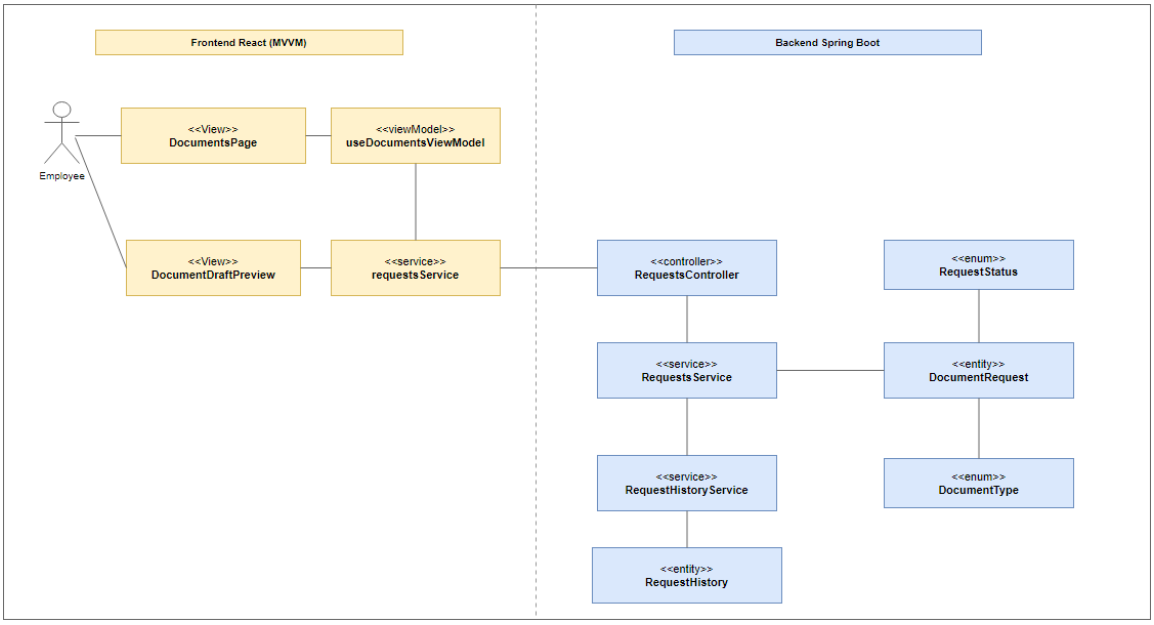


Figure 92: Participant Class Diagram – Document Request Management

Participant Class Diagram – Loan Request Management

This diagram shows the participants involved in employee loan request submission, draft validation, eligibility evaluation, status tracking, and request history management.

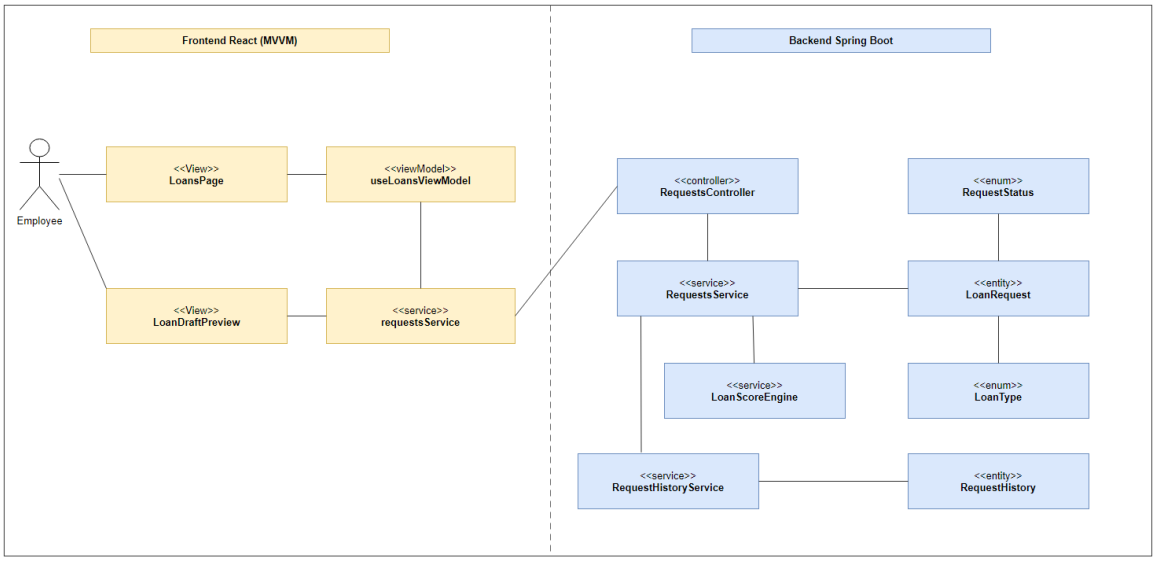


Figure 93: Participant Class Diagram – Loan Request Management

Participant Class Diagram – Authorization Request Management

This diagram shows the participants involved in employee authorization request submission, draft validation, status tracking, and request history management.

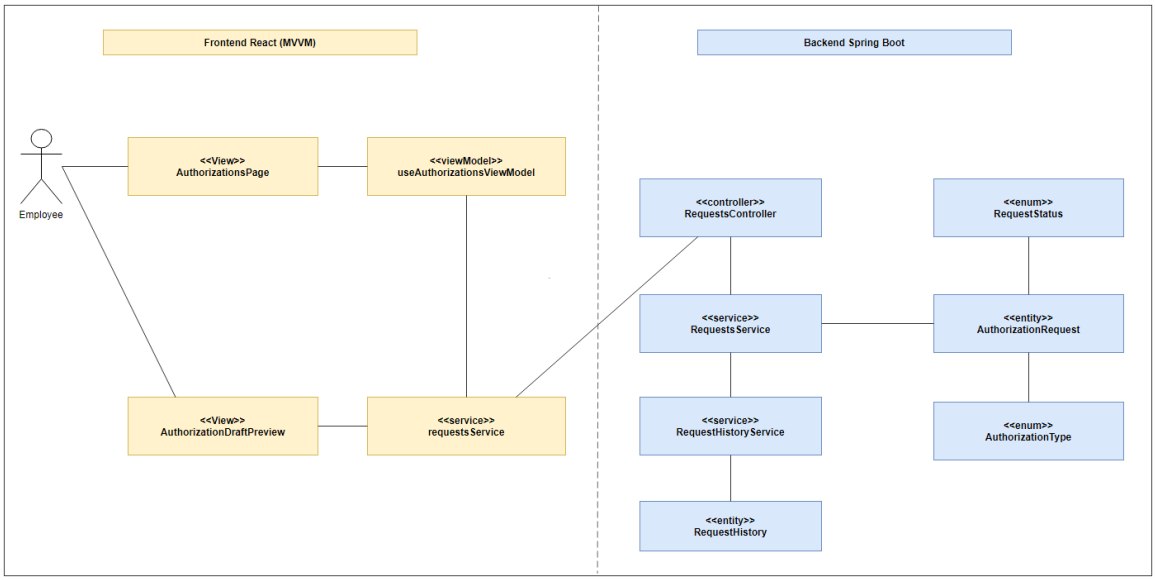


Figure 94: Participant Class Diagram – Authorization Request Management

Participant Class Diagram – HR Request Processing

This diagram shows the participants involved in HR processing of document, loan, and authorization requests, including request review, status updates, and history tracking.

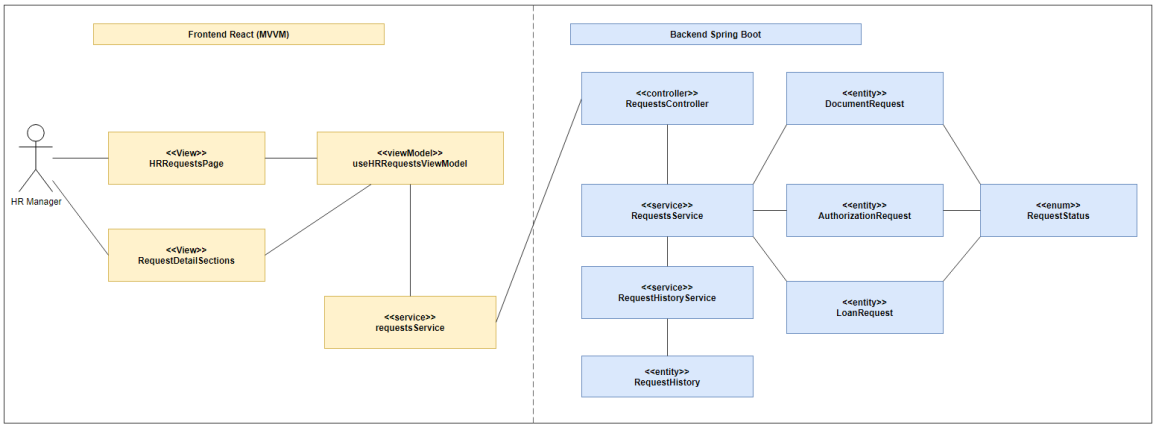


Figure 95: Participant Class Diagram – HR Request Processing

State Diagrams

These diagrams summarize the lifecycle states and transitions used by document, loan, and authorization requests during Sprint 3. They distinguish real backend request statuses from derived display outcomes such as document readiness or waiting for an HR-uploaded file.

State Diagram – Document Request Lifecycle

This diagram shows the lifecycle of a document request from submission to HR decision, including the derived fulfillment outcomes used when a leave balance statement is immediately ready or another document is waiting for an HR-uploaded file.

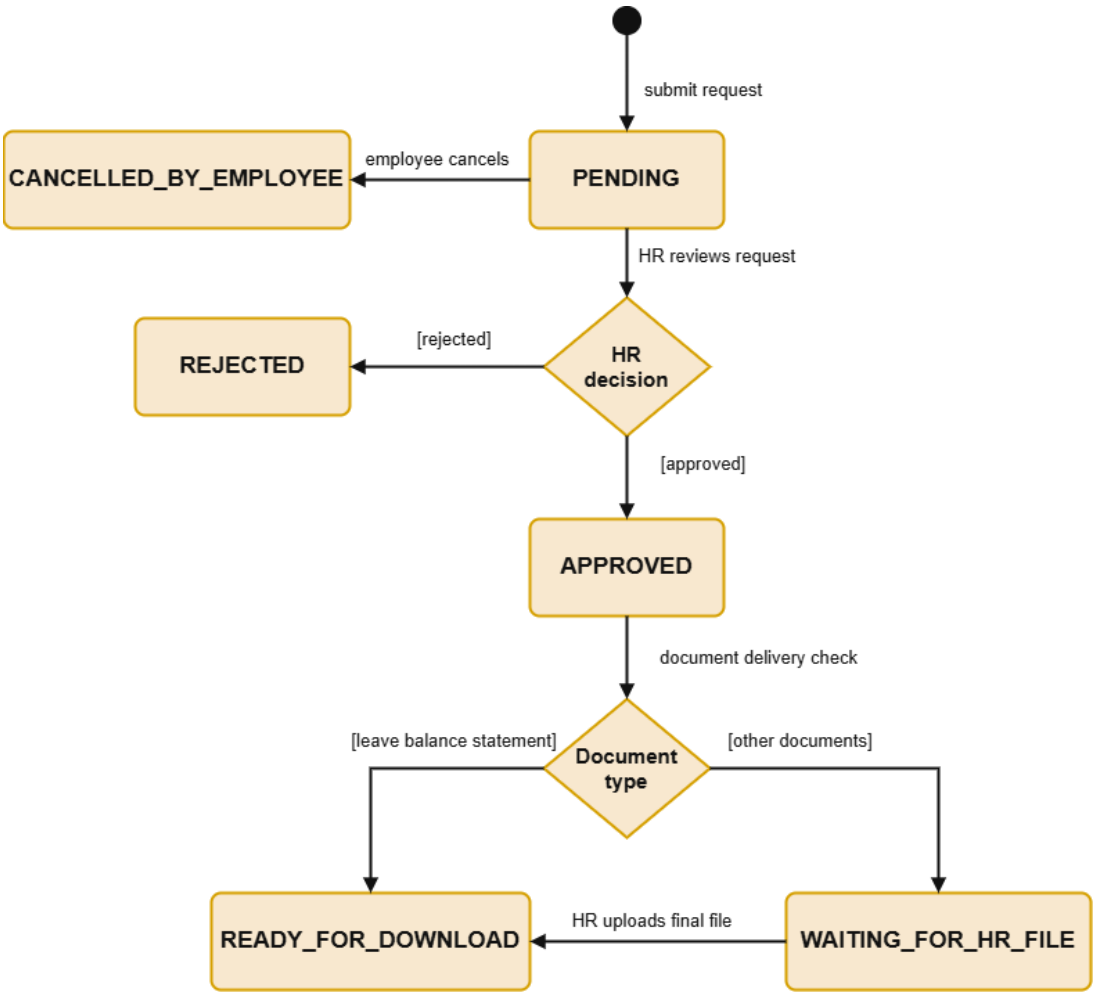


Figure 96: State Diagram – Document Request Lifecycle

State Diagram – Loan Request Lifecycle

This diagram shows the lifecycle of a loan request, including meeting and final decision states.

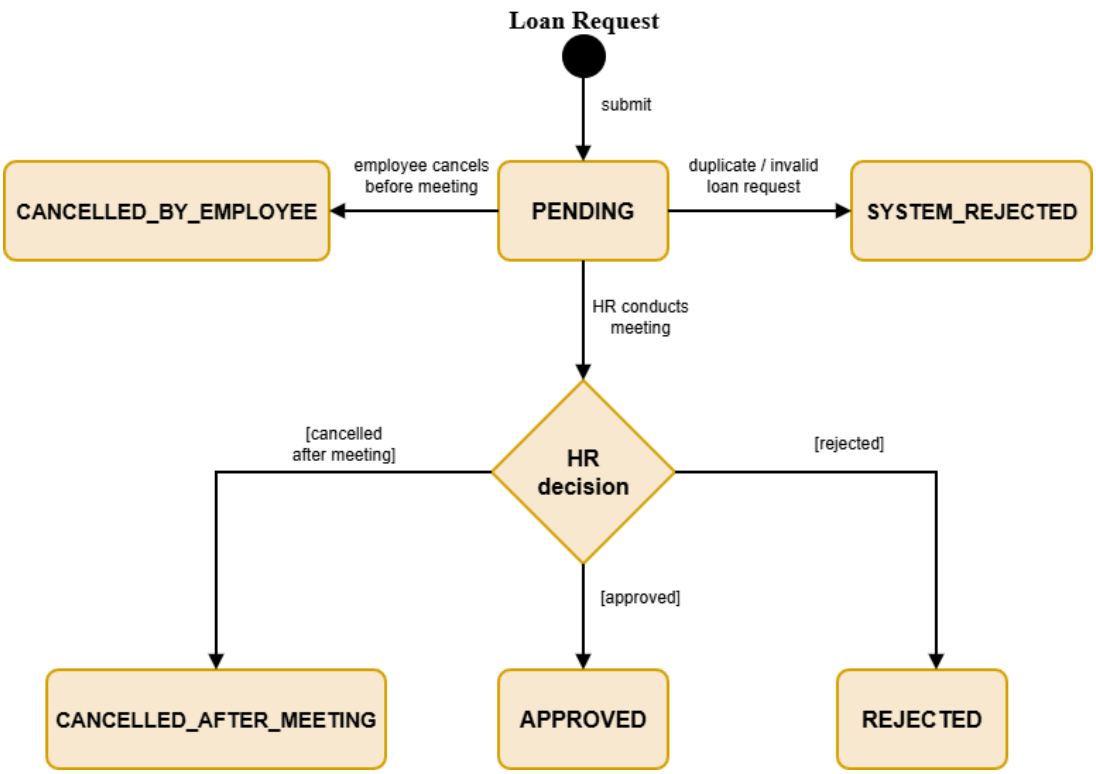


Figure 97: State Diagram – Loan Request Lifecycle

State Diagram – Authorization Request Lifecycle

This diagram shows the lifecycle of an authorization request from submission to HR decision.

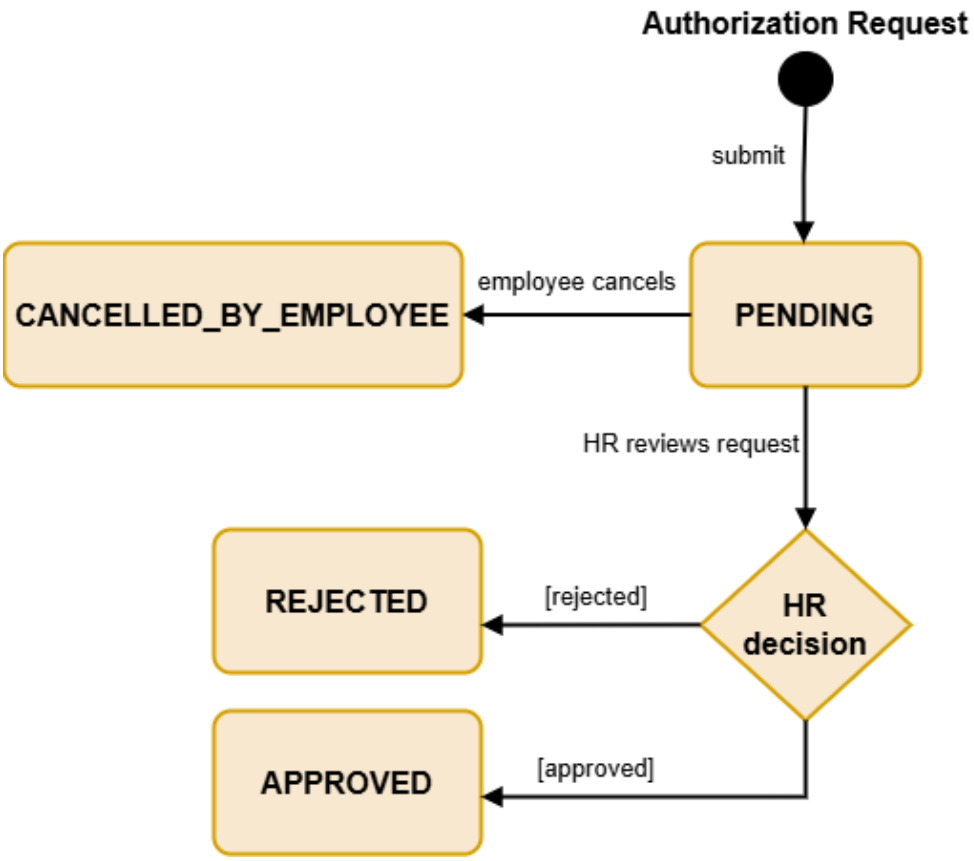


Figure 98: State Diagram – Authorization Request Lifecycle

Design Class Diagrams

These diagrams present the main design classes that support document handling, loan eligibility, authorization decisions, and history tracking.

Design Class Diagram – Document Request Final File Management

This diagram shows the classes used to manage generated documents and HR-uploaded final files.

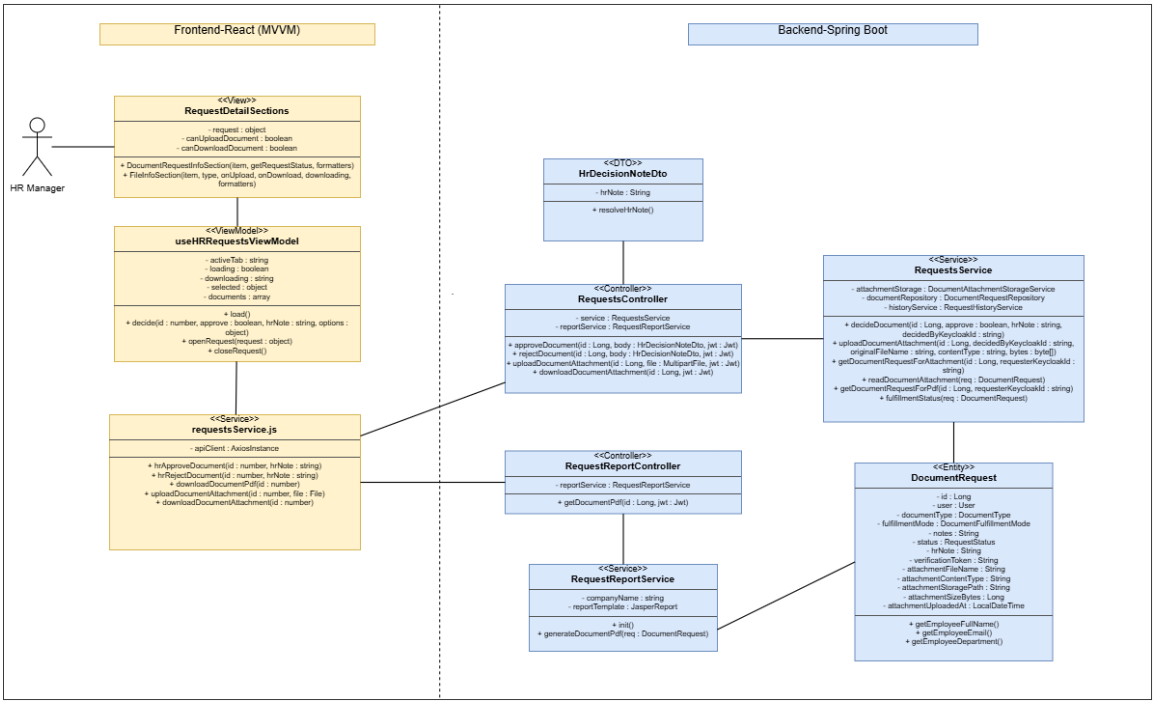


Figure 99: Design Class Diagram – Document Request Final File Management

Design Class Diagram – Loan Eligibility and Meeting Management

This diagram shows the classes used to verify loan eligibility and manage loan meetings.

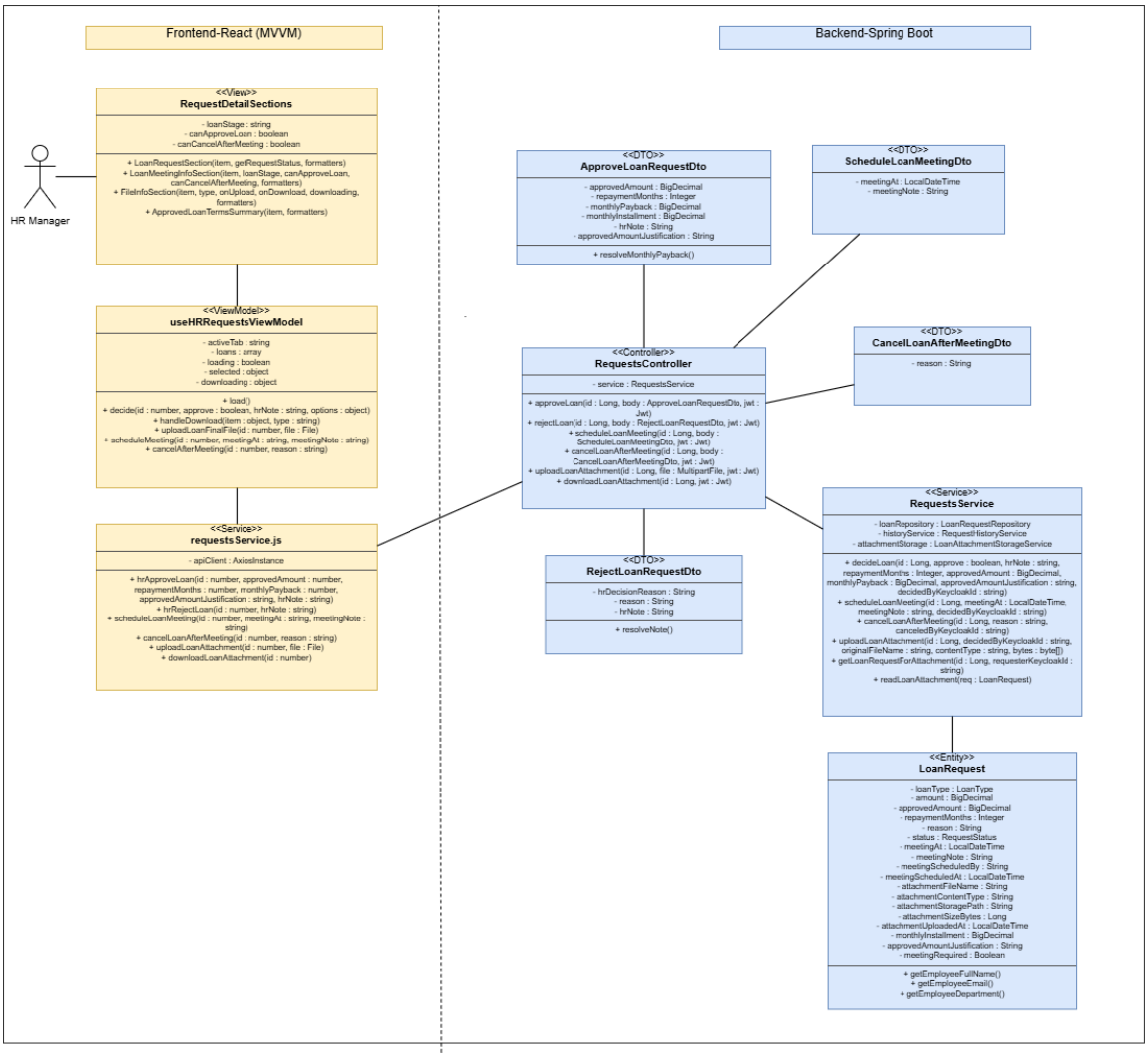


Figure 100: Design Class Diagram – Loan Request Review Management

Design Class Diagram – Authorization Request Decision Management

This diagram shows the classes used to process authorization decisions and rejection reasons.

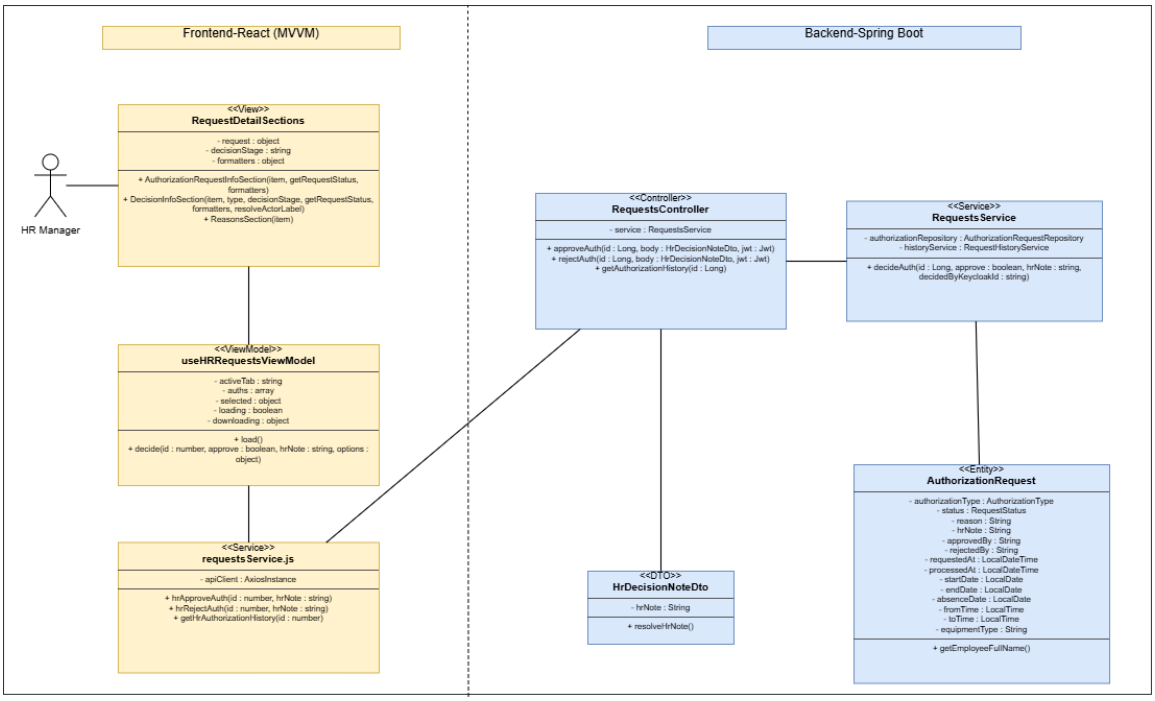


Figure 101: Design Class Diagram – Authorization Request Management

Design Class Diagram – Request History and Notification Management

This diagram shows the classes used to trace history and emit request events and notifications.

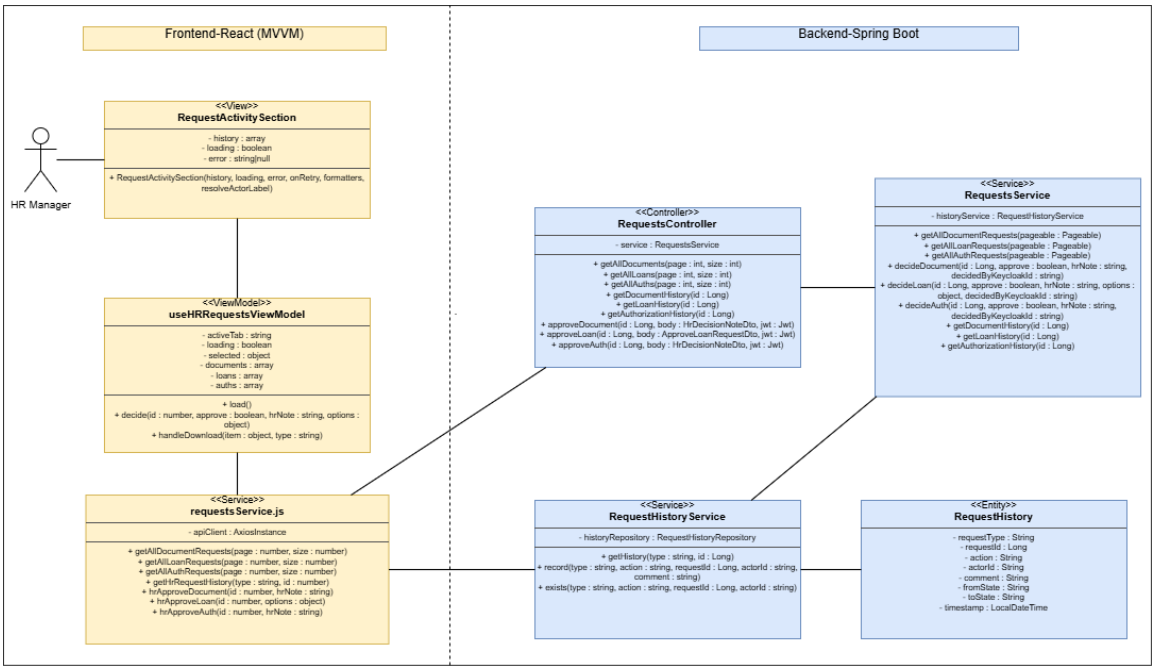


Figure 102: Design Class Diagram – HR Request Processing

Conception Sequence Diagrams

These diagrams describe the internal frontend and backend interactions behind the ad-

ministrative request workflows of Sprint 3. They are organized by request category: document requests, loan requests, and authorization requests. Each workflow is split into focused frontend and backend views to keep the diagrams readable and to separate user-interface behavior from backend validation, persistence, history, and notification logic.

Conception Sequence Diagram – Document Request HR Decision Frontend

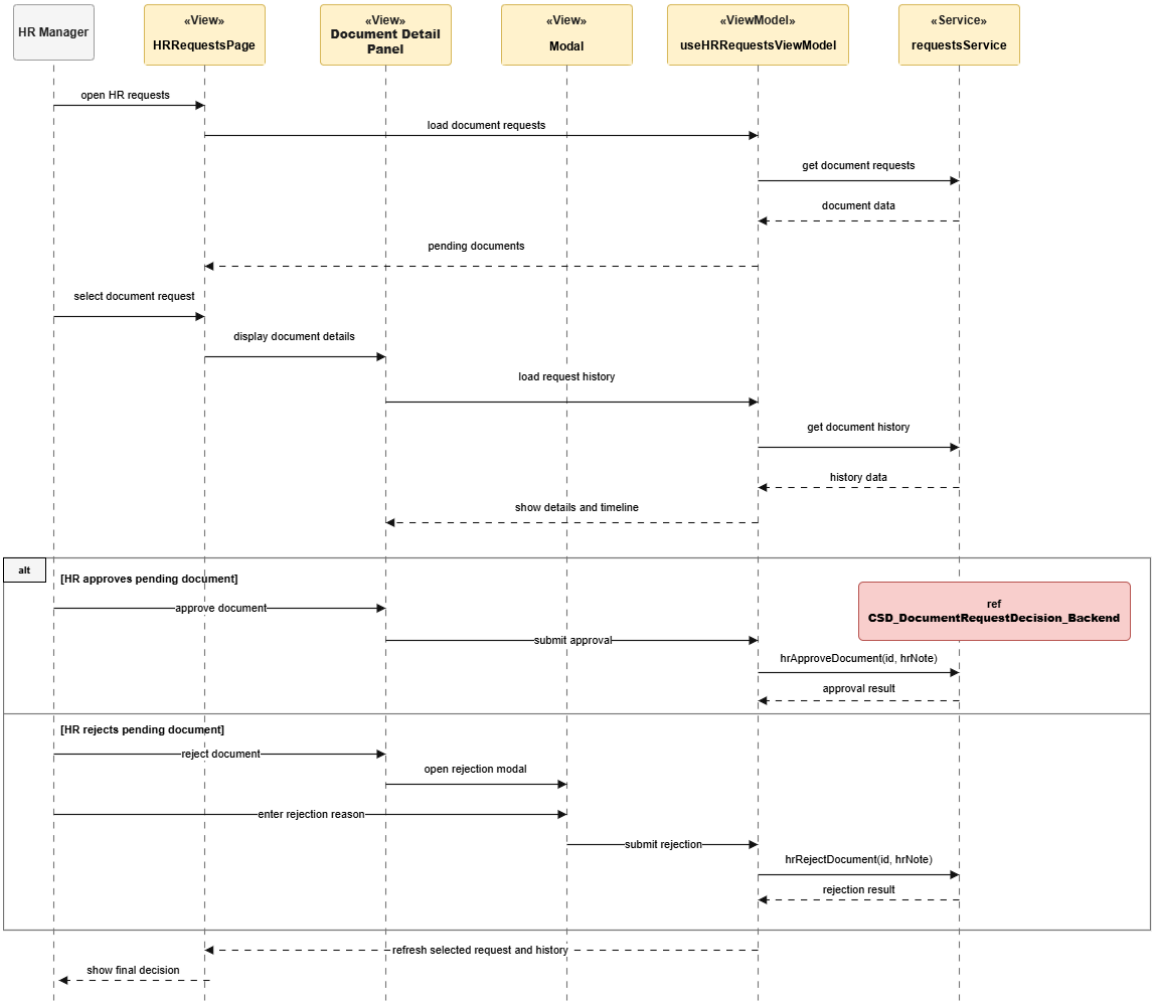


Figure 103: Conception Sequence Diagram – Document Request HR Decision Frontend

Conception Sequence Diagram – Document Request Decision Backend

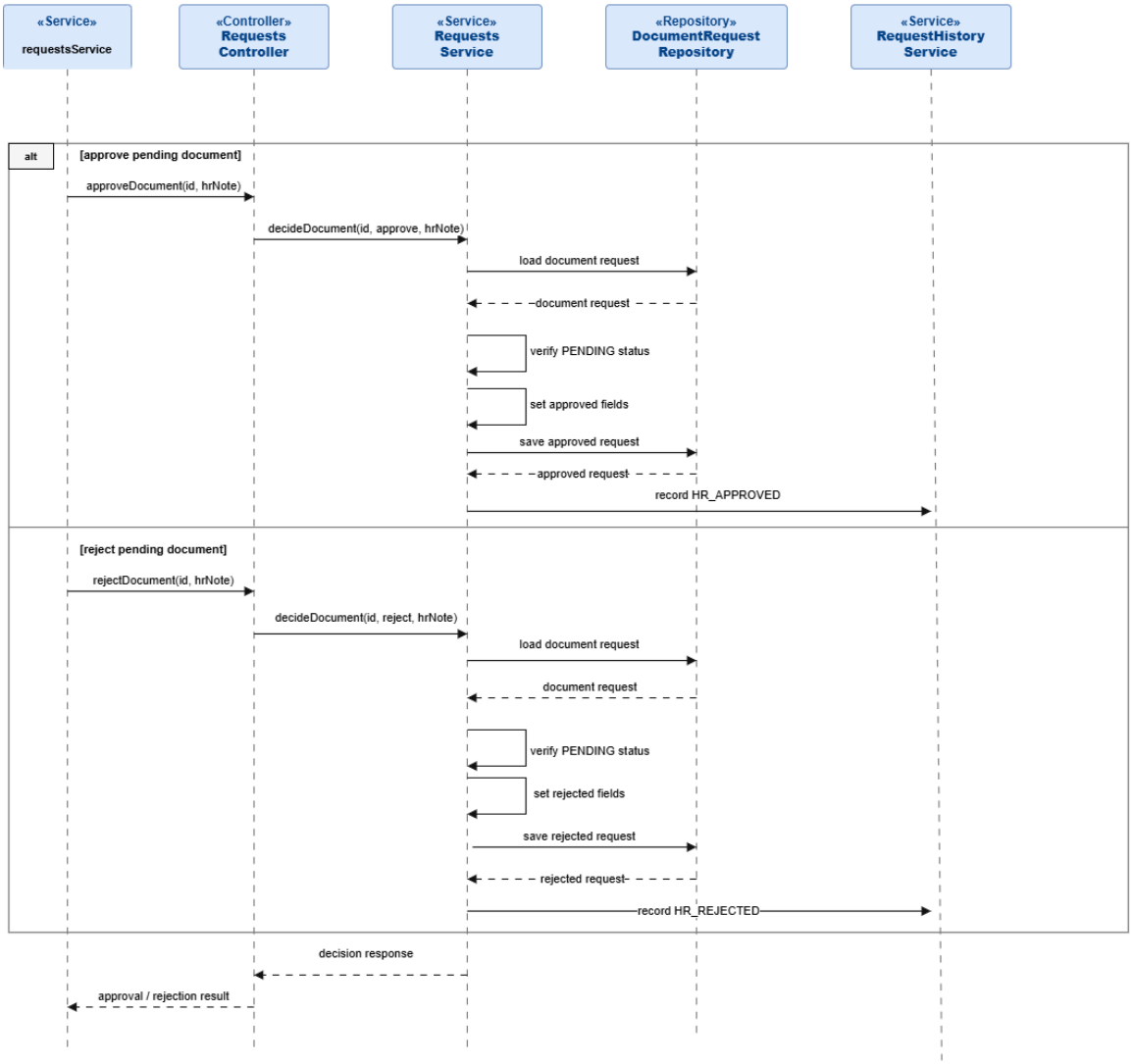


Figure 104: Conception Sequence Diagram – Document Request Decision Backend

Conception Sequence Diagram – Document Request Final File Frontend

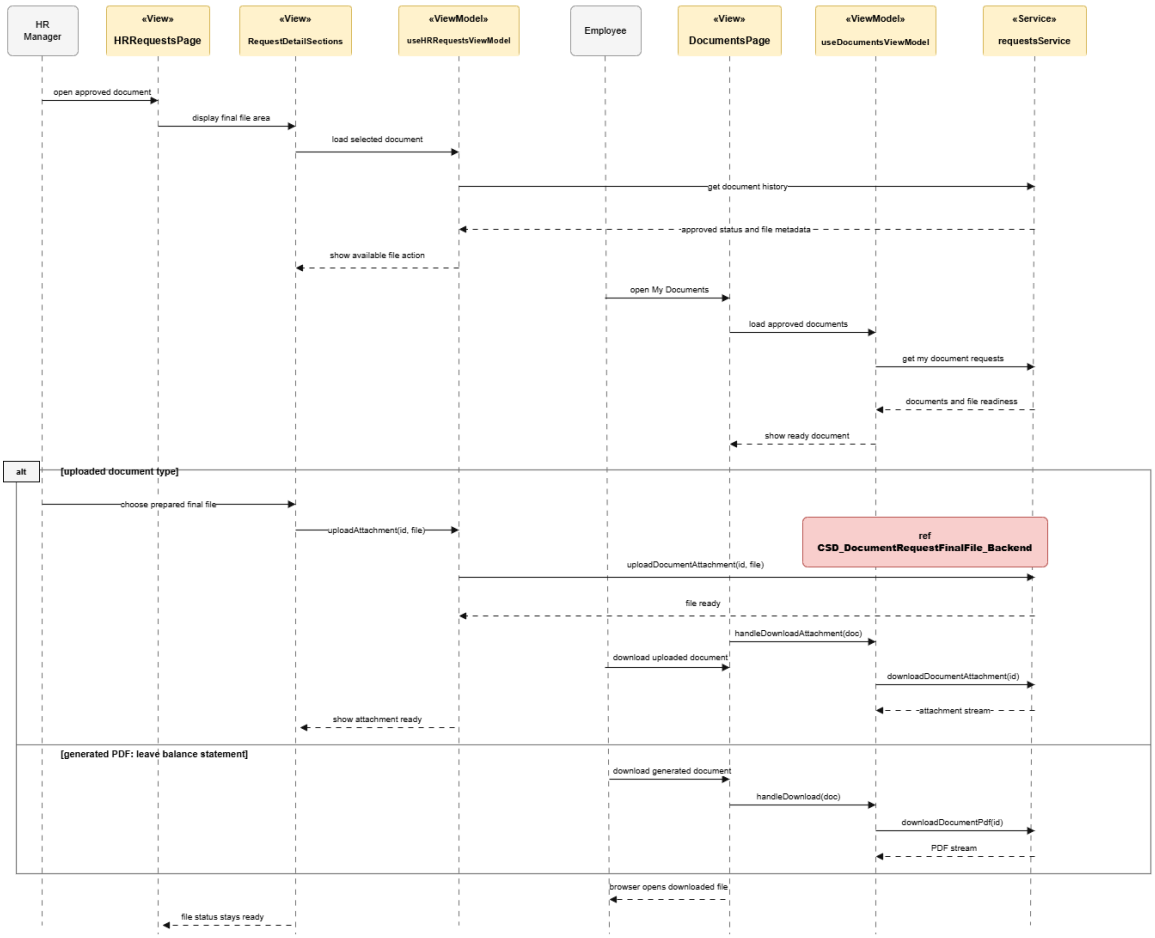


Figure 105: Conception Sequence Diagram – Document Request Final File Frontend

Conception Sequence Diagram – Document Request Final File Backend

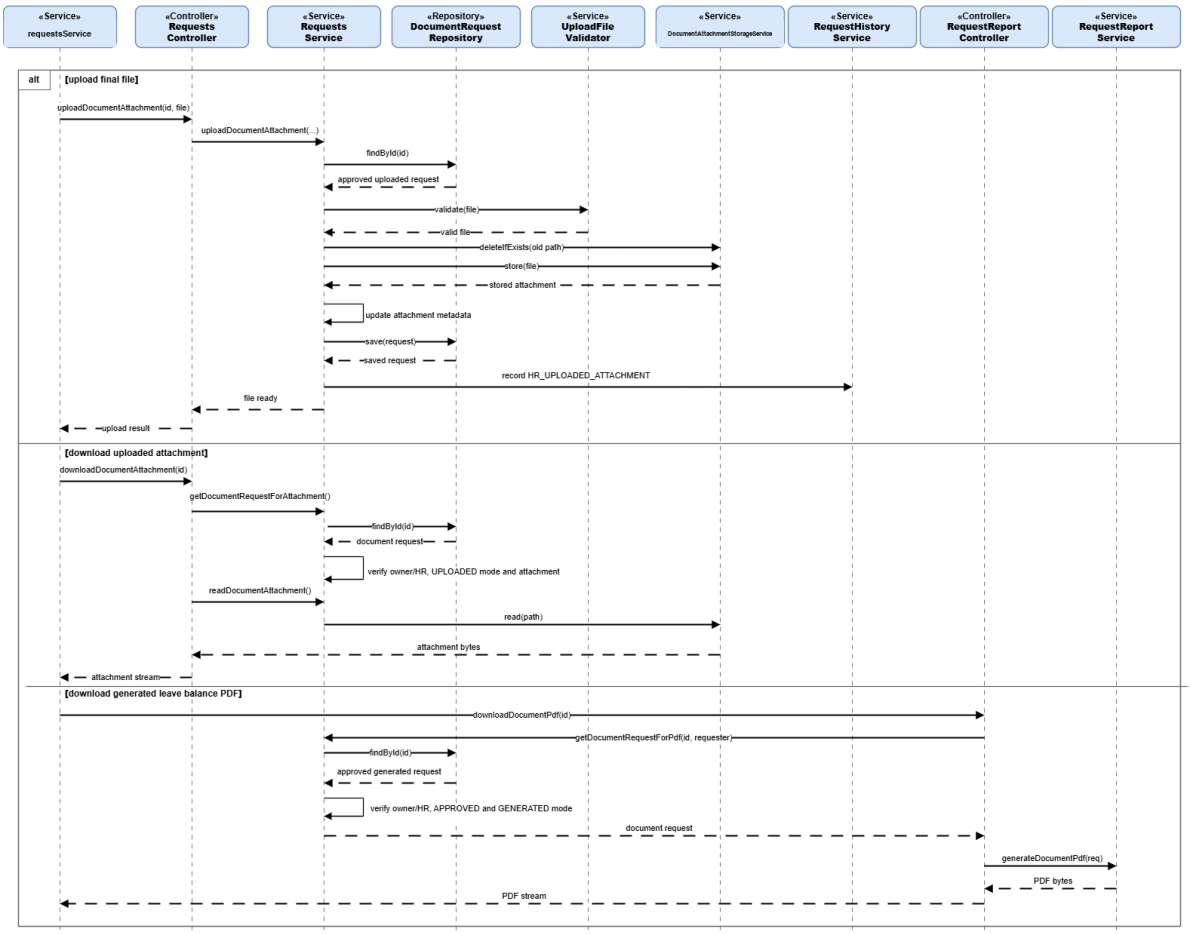


Figure 106: Conception Sequence Diagram – Document Request Final File Backend

Conception Sequence Diagram – Loan Request Submission Frontend

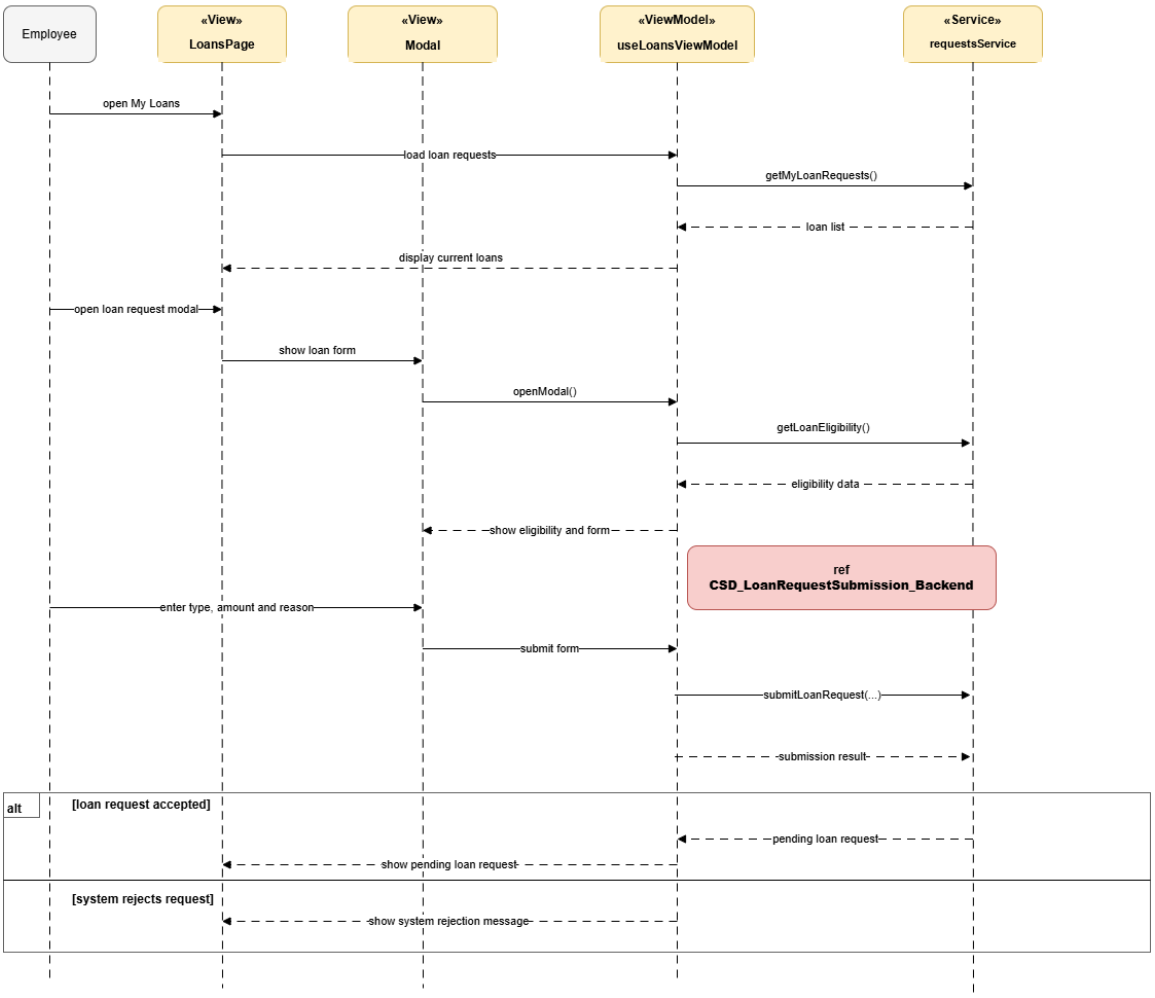


Figure 107: Conception Sequence Diagram – Loan Request Submission Frontend

Conception Sequence Diagram – Loan Request Submission Backend

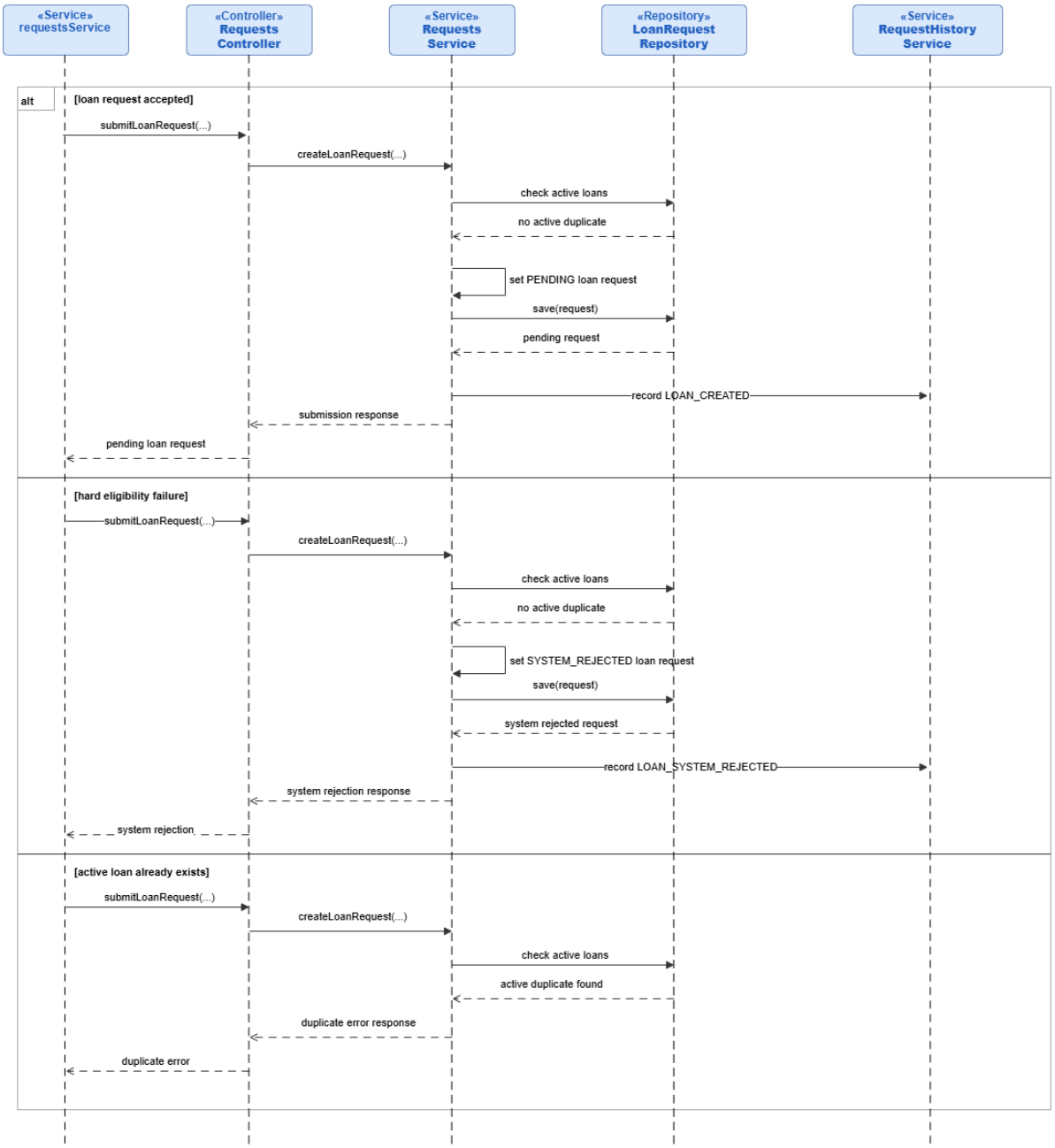


Figure 108: Conception Sequence Diagram – Loan Request Submission Backend

Conception Sequence Diagram – Loan Request Meeting Scheduling Frontend

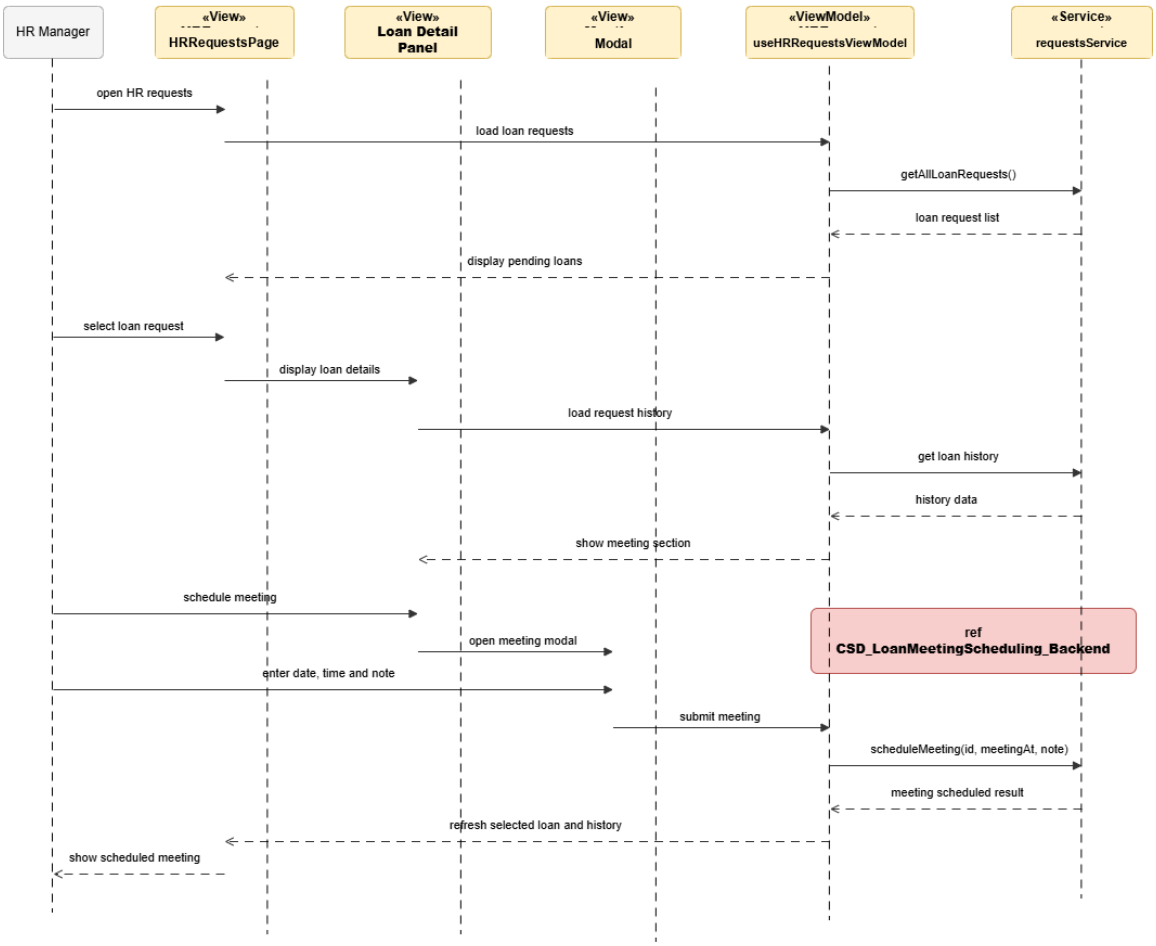


Figure 109: Conception Sequence Diagram – Loan Request Meeting Scheduling Frontend

Conception Sequence Diagram – Loan Meeting Scheduling Backend

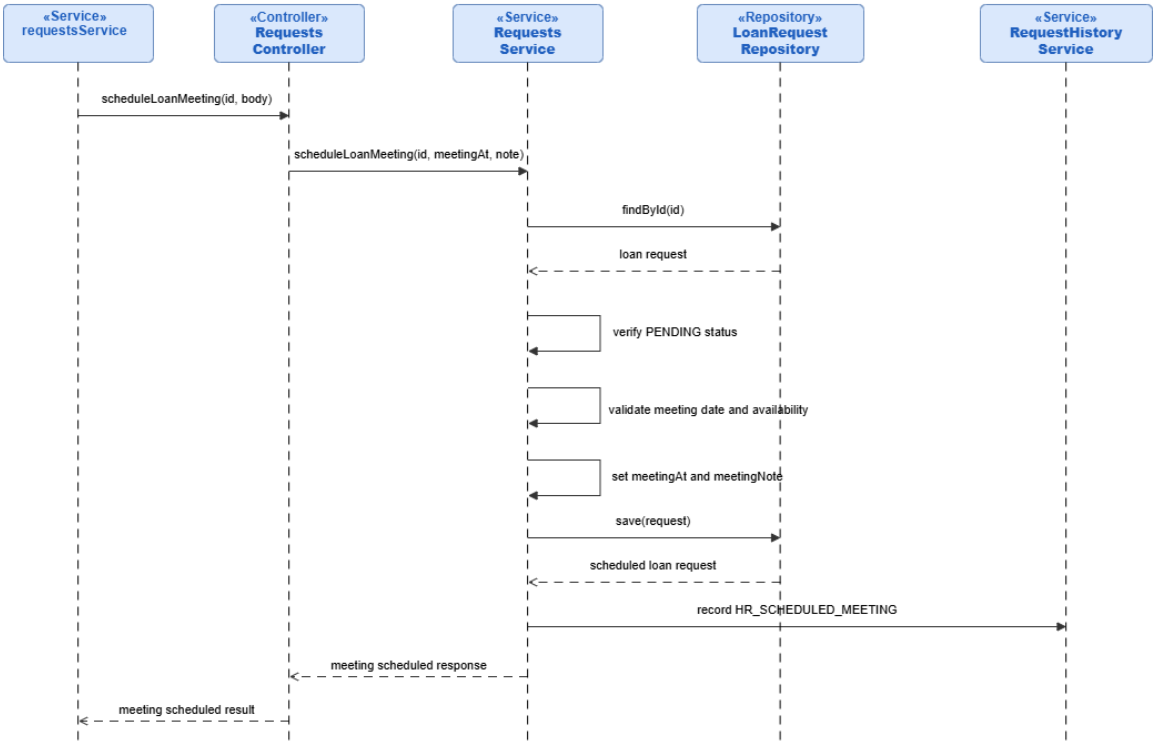


Figure 110: Conception Sequence Diagram – Loan Meeting Scheduling Backend

Conception Sequence Diagram – Loan Request Decision Frontend

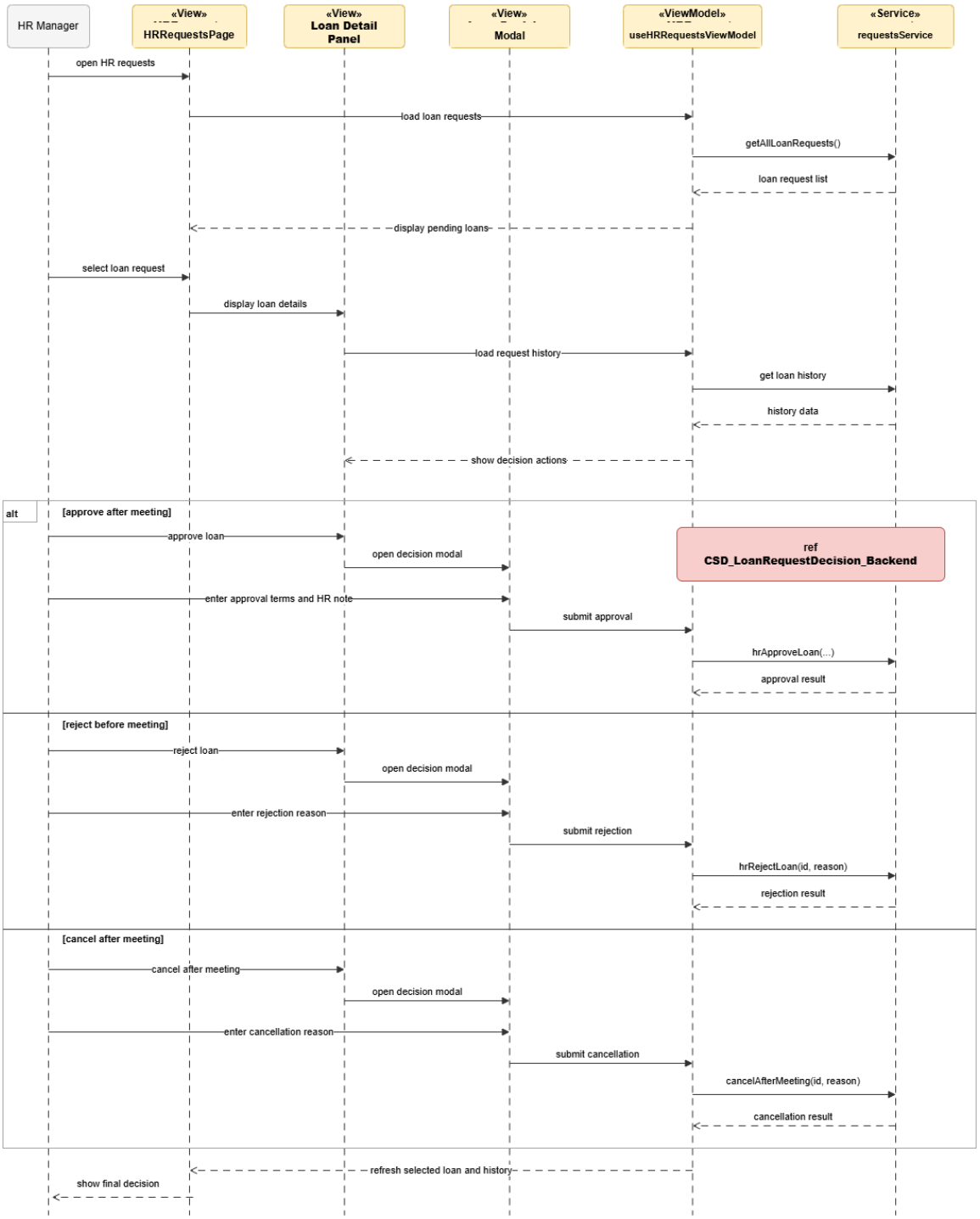


Figure 111: Conception Sequence Diagram – Loan Request Decision Frontend

Conception Sequence Diagram – Loan Request Decision Backend

Figure 112: Conception Sequence Diagram – Loan Request Decision Backend

Conception Sequence Diagram – Loan Final File Delivery Frontend

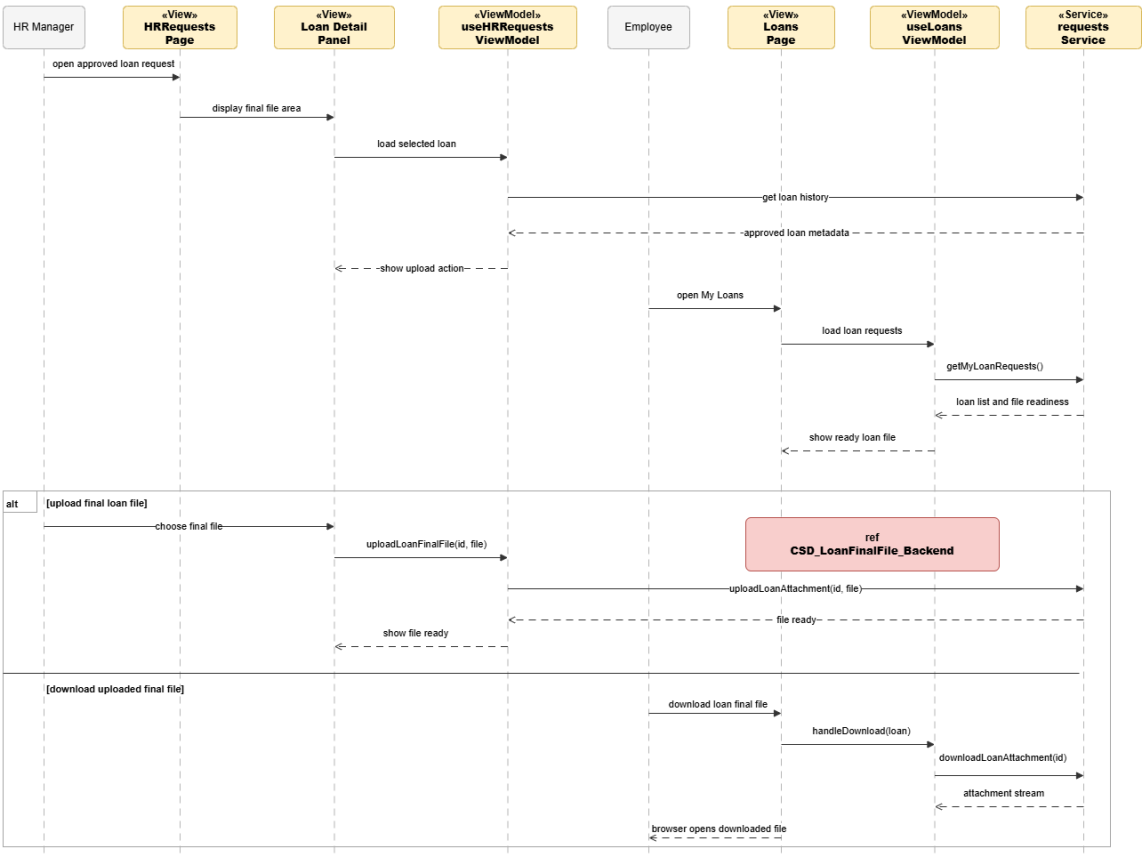


Figure 113: Conception Sequence Diagram – Loan Final File Delivery Frontend

Conception Sequence Diagram – Loan Final File Backend

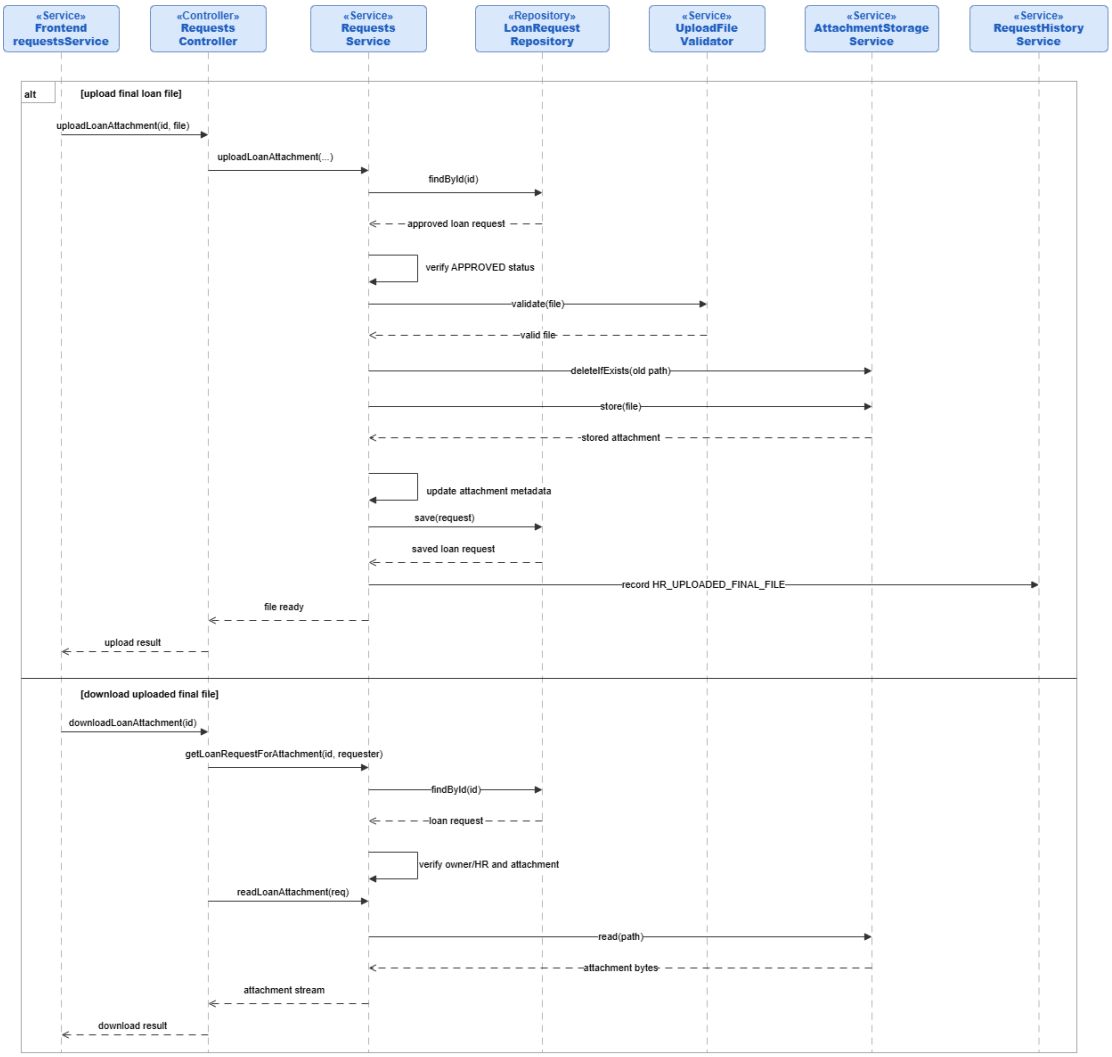


Figure 114: Conception Sequence Diagram – Loan Final File Backend

Conception Sequence Diagram – Authorization Request Submission Frontend

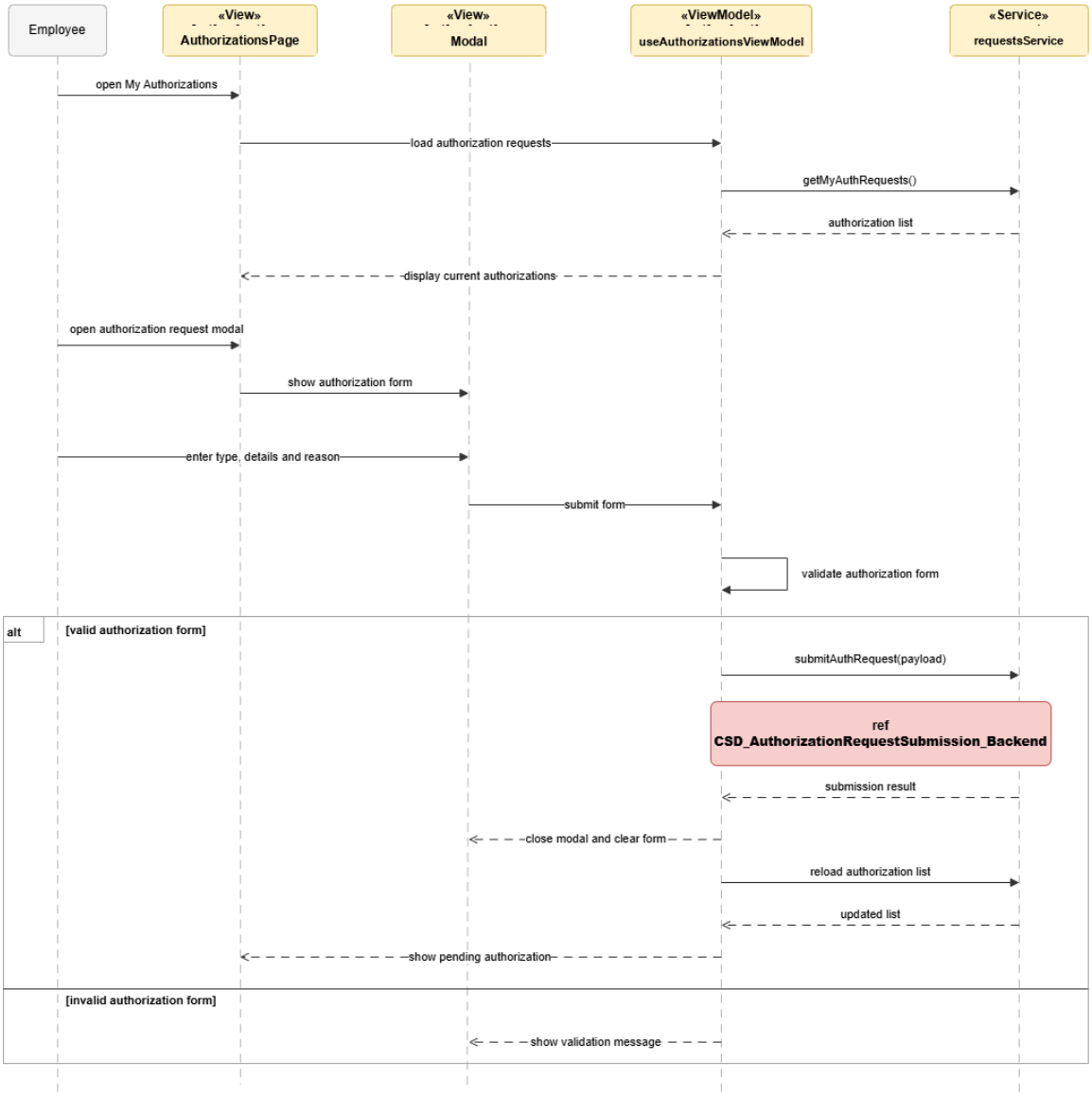


Figure 115: Conception Sequence Diagram – Authorization Request Submission Frontend

Conception Sequence Diagram – Authorization Request Submission Backend

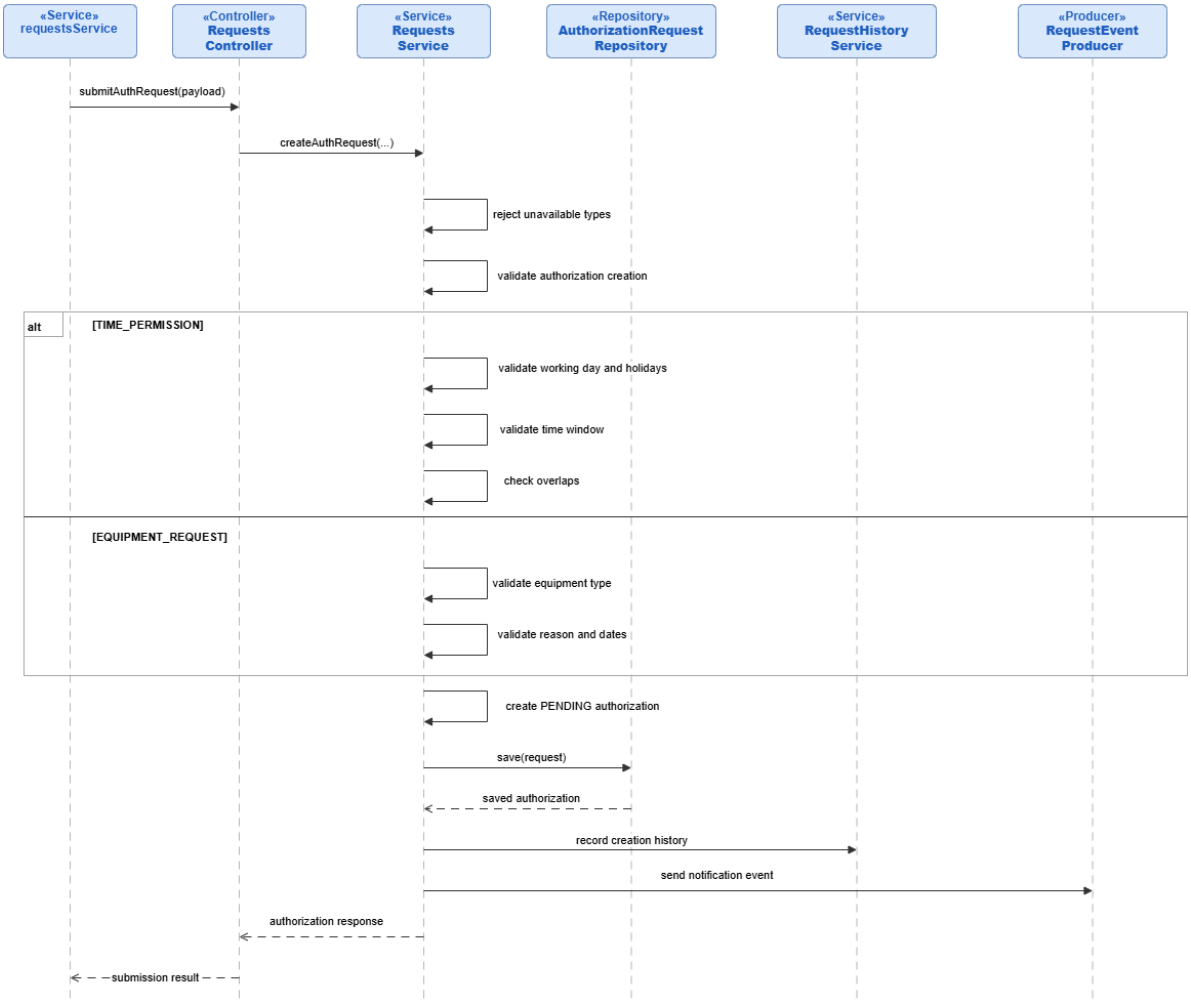


Figure 116: Conception Sequence Diagram – Authorization Request Submission Backend

Conception Sequence Diagram – Authorization Request Decision Frontend

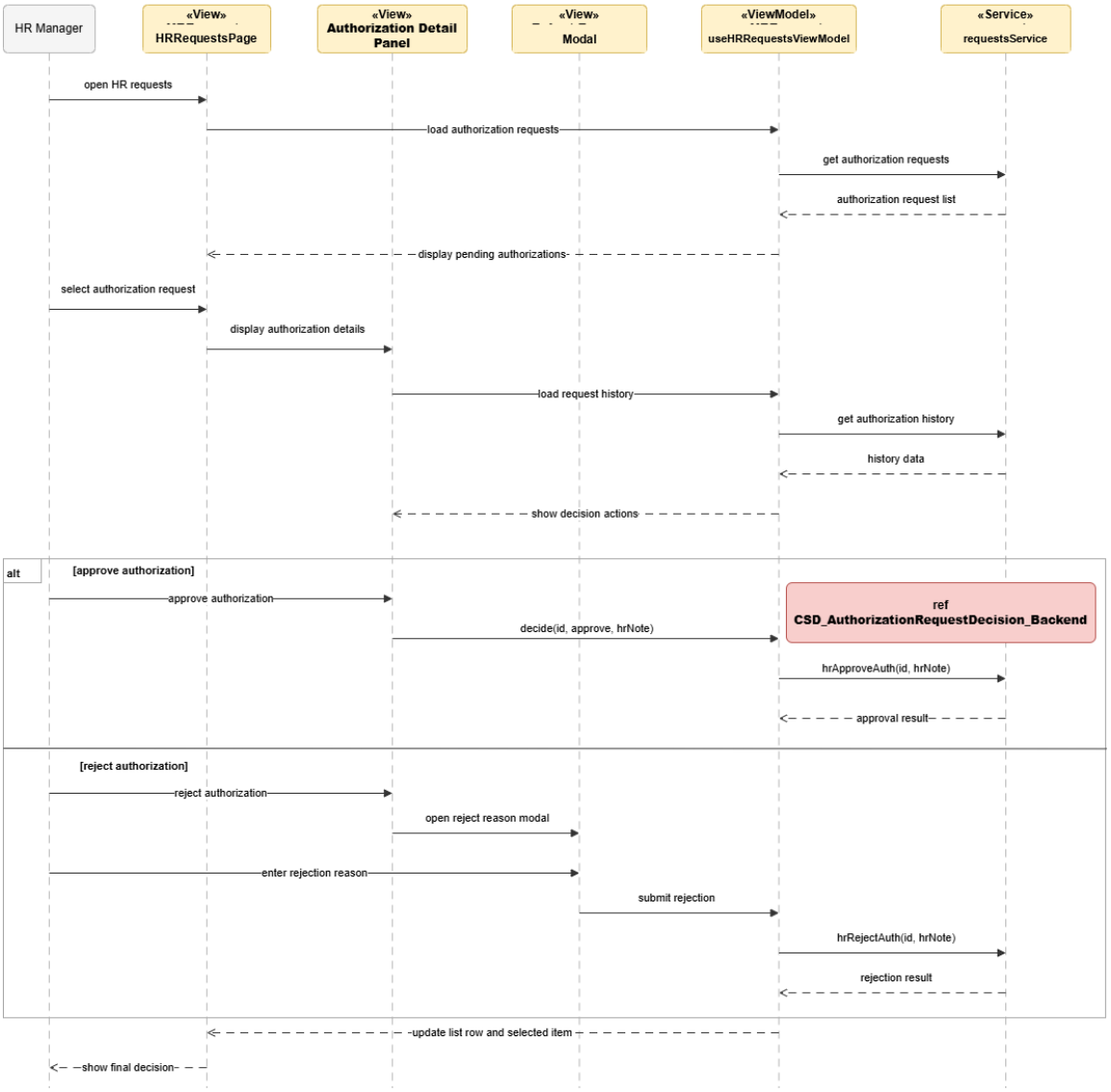


Figure 117: Conception Sequence Diagram – Authorization Request Decision Frontend

Conception Sequence Diagram – Authorization Request Decision Backend

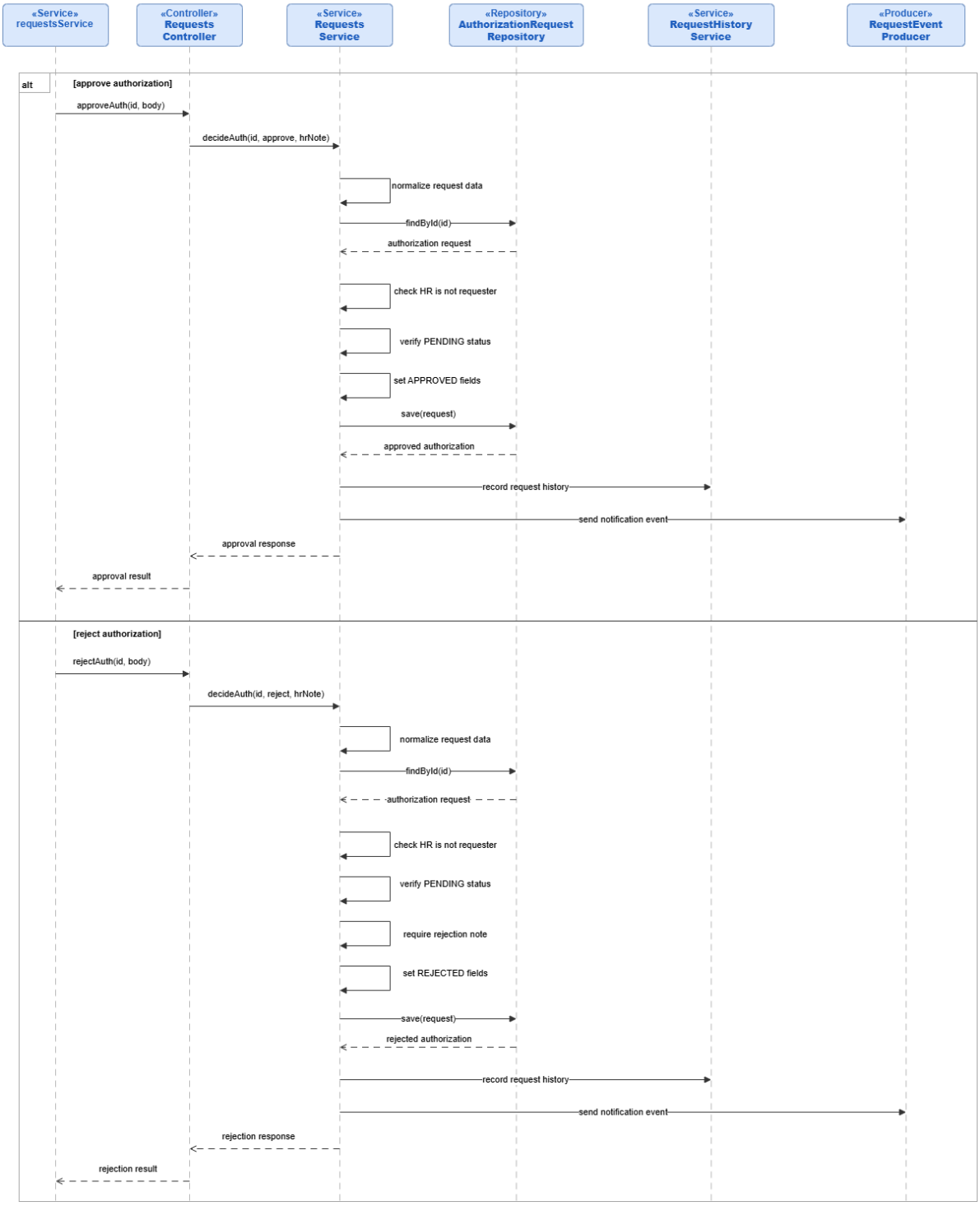


Figure 118: Conception Sequence Diagram – Authorization Request Decision Backend

Realization

The realization section presents the validated user interfaces for Sprint 3.

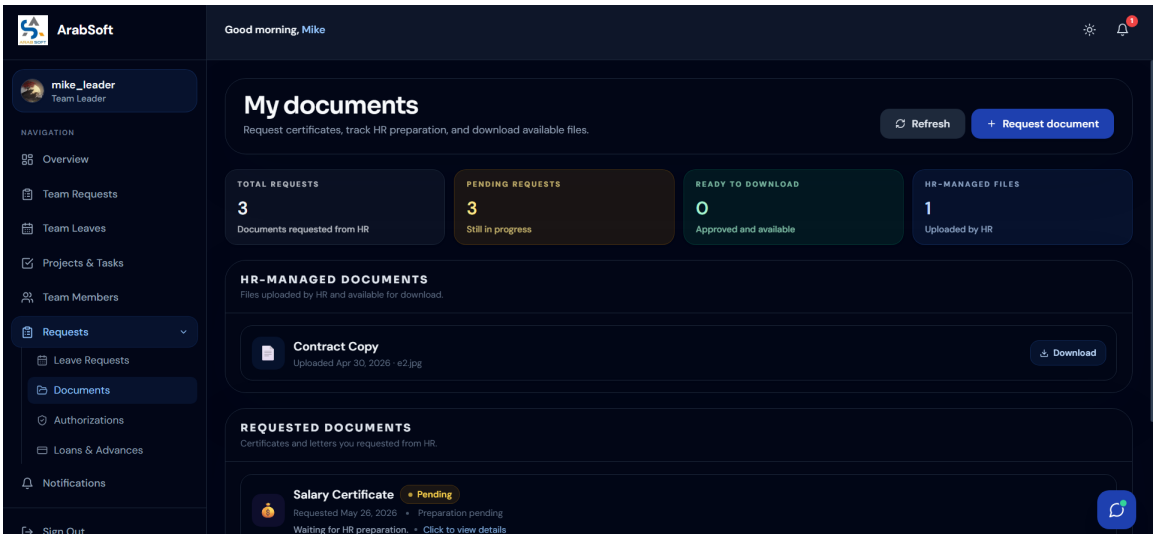


Figure 119: Employee Documents Interface

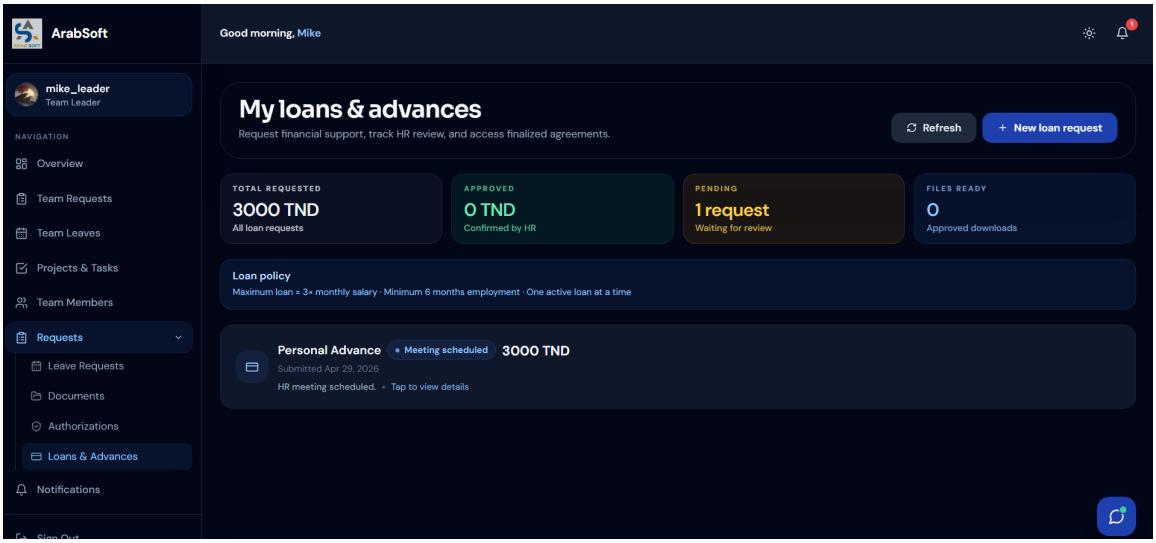


Figure 120: Employee Loans and Advances Interface

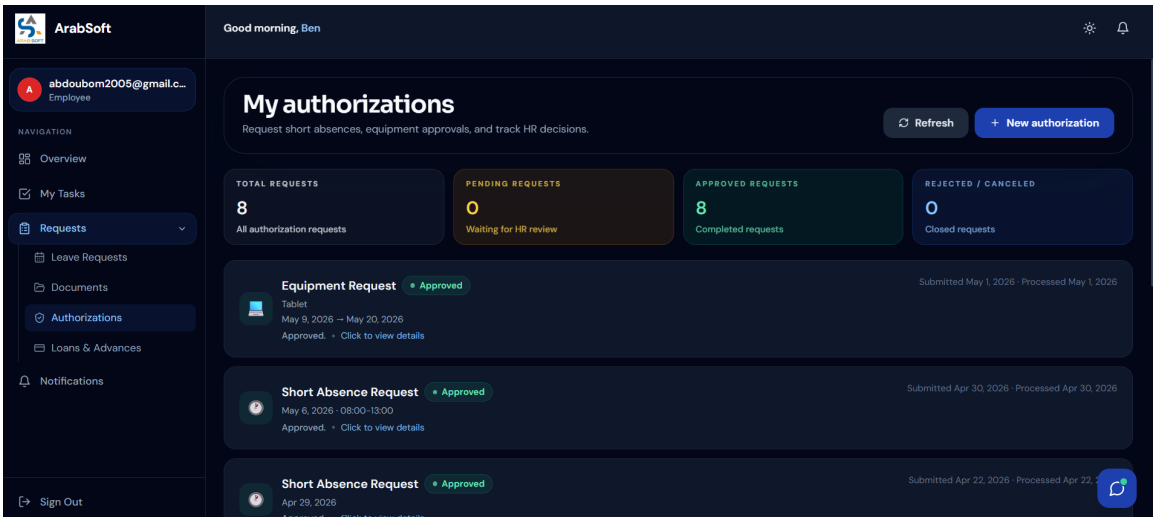


Figure 121: Employee Authorizations Interface

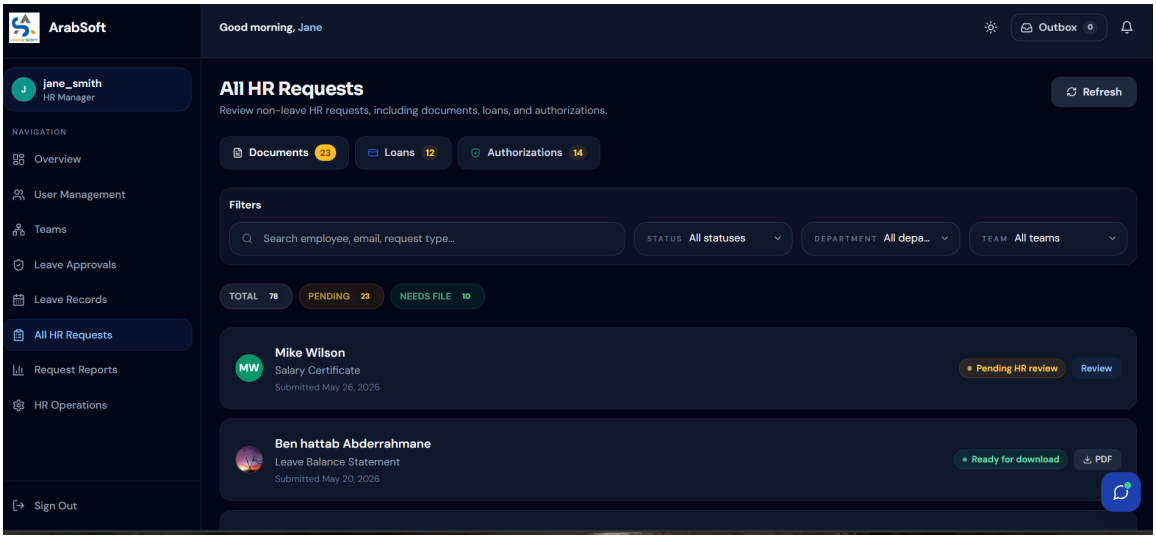


Figure 122: HR Administrative Requests Interface

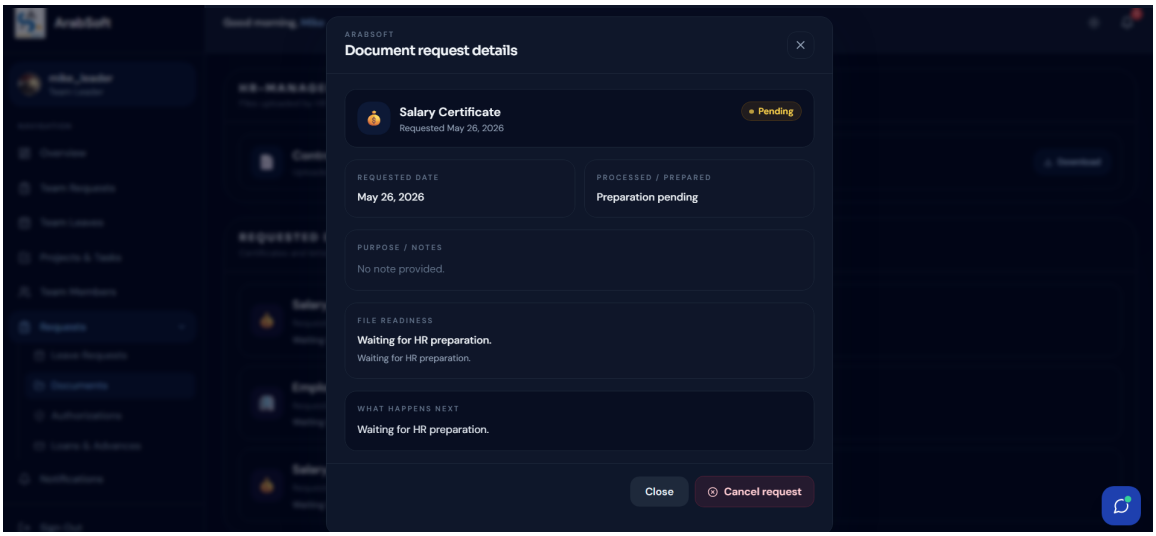


Figure 123: Request Detail and History Interface

5. Sprint Review

Sprint 3 covered the administrative request workflow of the platform. Employees can submit and track document, loan, and authorization requests. HR managers can review each request, approve or reject it with a decision note, schedule loan meetings, upload final files where required, and consult the full request history. Request statuses, history records, decision notes, and notification events keep each request traceable from submission through to final decision. Final-file handling for document and loan requests was also built into the workflow.

6. Sprint Retrospective

What went well	What needs improvement
The sprint delivered the main administrative request workflows for documents, loans, and authorizations.	**Workflow complexity required tighter validation rules and a clearer HR decision flow across request states.**
Request history, decision notes, statuses, and notifications improved workflow traceability.	**Final-file handling and edge-case paths still need clearer validation and traceability.**
Splitting frontend and backend diagrams made the technical design easier to explain.	**UI validation and user testing still need to cover feedback messages and request-state transitions.**

Table 7: Sprint Retrospective – Sprint 3

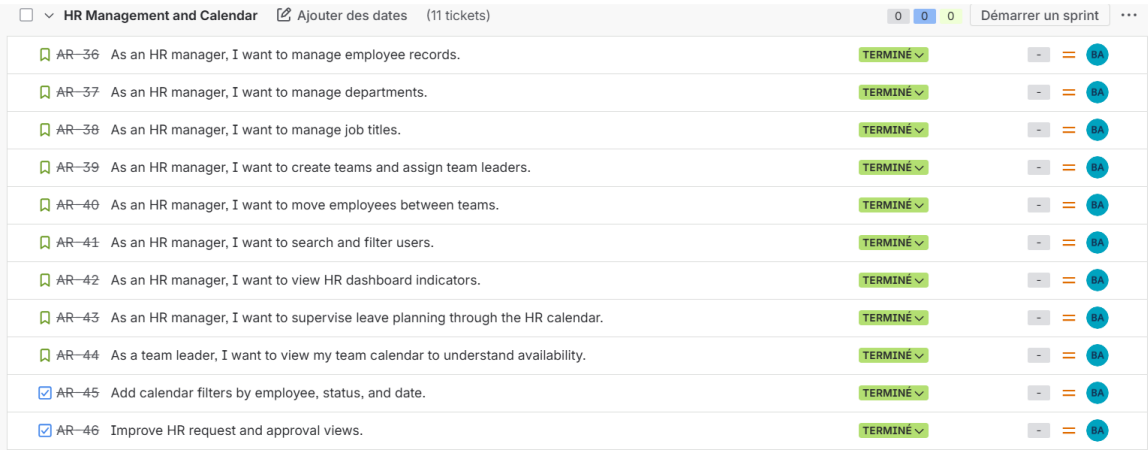
III. Sprint 4: HR Management and Calendar

1. Sprint Goal

This sprint covers HR management and calendar supervision. It handles HR user administration, employee records, team setup, role assignment, account activation and deactivation, and leave-availability views for employees and teams.

2. Visual Sprint Journal

The sprint journal captures the tracked HR and calendar user stories and their validation status in Jira.



HR Management and Calendar	Ajouter des dates	(11 tickets)	0 0 0	Démarrer un sprint
~~AR-36~~	As an HR manager, I want to manage employee records.	TERMINÉ	-	BA
~~AR-37~~	As an HR manager, I want to manage departments.	TERMINÉ	-	BA
~~AR-38~~	As an HR manager, I want to manage job titles.	TERMINÉ	-	BA
~~AR-39~~	As an HR manager, I want to create teams and assign team leaders.	TERMINÉ	-	BA
~~AR-40~~	As an HR manager, I want to move employees between teams.	TERMINÉ	-	BA
~~AR-41~~	As an HR manager, I want to search and filter users.	TERMINÉ	-	BA
~~AR-42~~	As an HR manager, I want to view HR dashboard indicators.	TERMINÉ	-	BA
~~AR-43~~	As an HR manager, I want to supervise leave planning through the HR calendar.	TERMINÉ	-	BA
~~AR-44~~	As a team leader, I want to view my team calendar to understand availability.	TERMINÉ	-	BA
~~AR-45~~	Add calendar filters by employee, status, and date.	TERMINÉ	-	BA
~~AR-46~~	Improve HR request and approval views.	TERMINÉ	-	BA

Visual Sprint Journal – Sprint 4

3. Sprint Backlog

The table below summarizes the Sprint 4 work packages prepared from the product backlog defined in Chapter 2.

Sprint	Stories	Tasks	Duration
Sprint 4	**Employee records and user search**	Implement employee record consultation and update	2d
		Add user search and filtering for HR	
		Improve employee profile visibility for HR	
	Department and job title management	Implement department management	2d
		Implement job title management	
		Connect departments and job titles with employee records	
	Team management and leader assignment	Implement team creation and update	2d
		Assign or change team leaders	
		Handle teams without assigned leaders when needed	
	Employee movement and HR approval views	Move employees between teams	1d
		Improve HR request and approval views	
		Keep employee assignment data consistent	
	HR dashboard and HR calendar	Display HR dashboard indicators	2d
		Implement HR leave calendar supervision	
		Add calendar visibility for employees, teams, and statuses	
	Team calendar and calendar filters	Implement team calendar consultation	1d
		Add calendar filters by employee, status, and date	
		Improve calendar readability for planning	

Table 8: Sprint Backlog – Sprint 4

4. Sprint Implementation

Requirements Analysis

Sprint 4 covers HR management and calendar supervision, including the HR dashboard, employee account supervision, organization setup, team structure management, and leave calendar views.

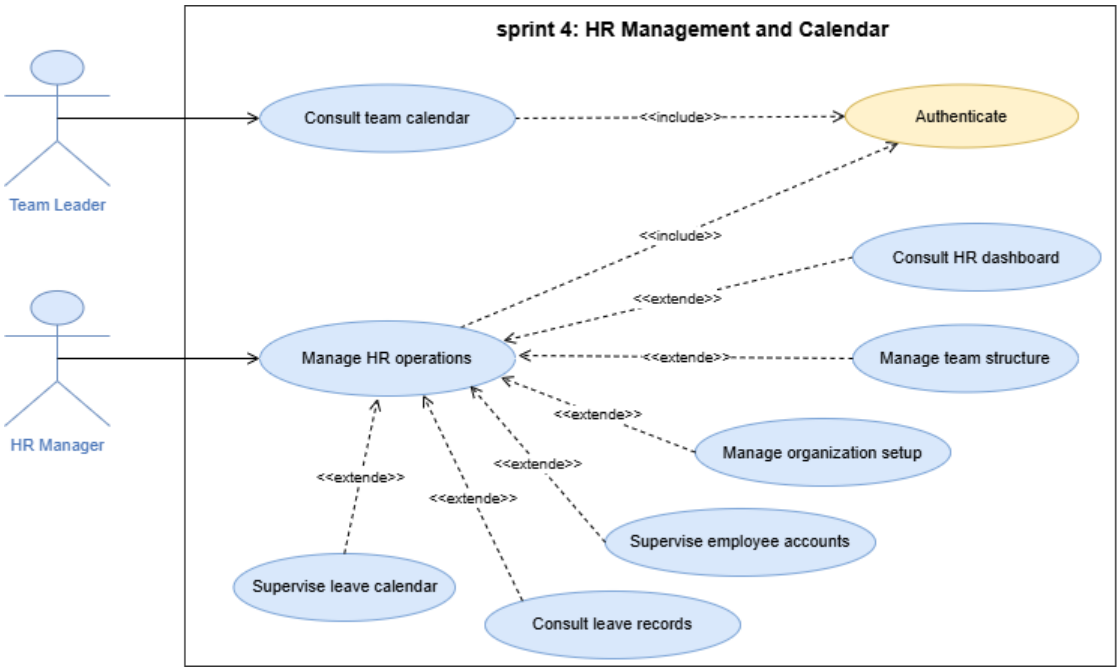


Figure 124: Use Case Diagram – Sprint 4 HR Management and Calendar

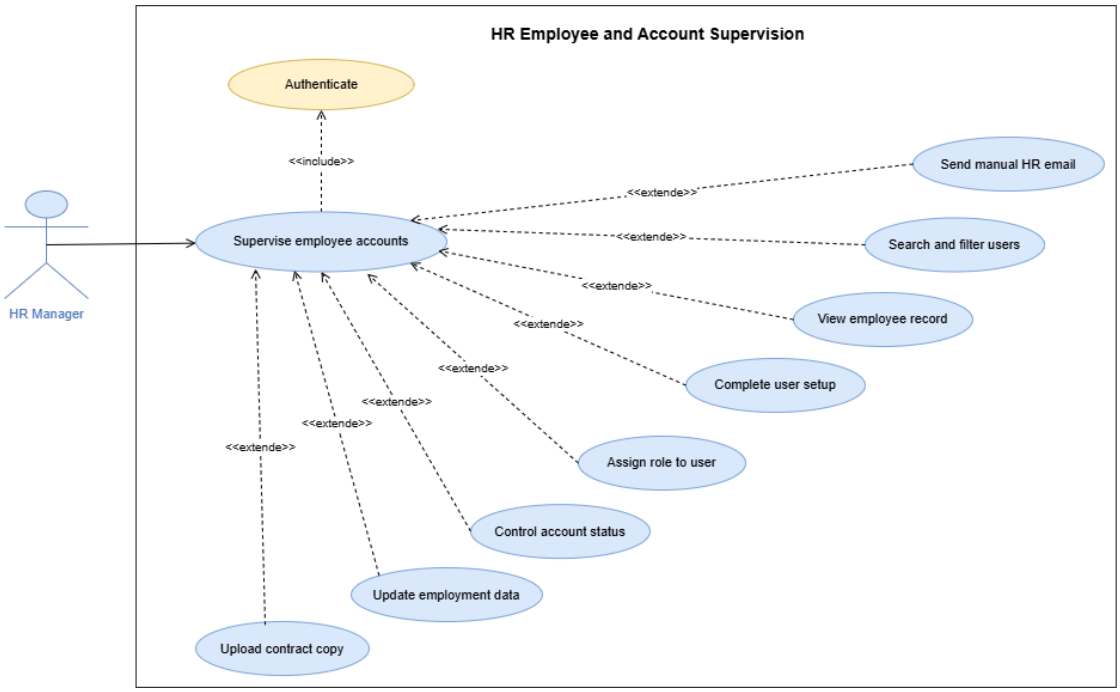


Figure 125: Use Case Diagram – HR Employee and Account Supervision

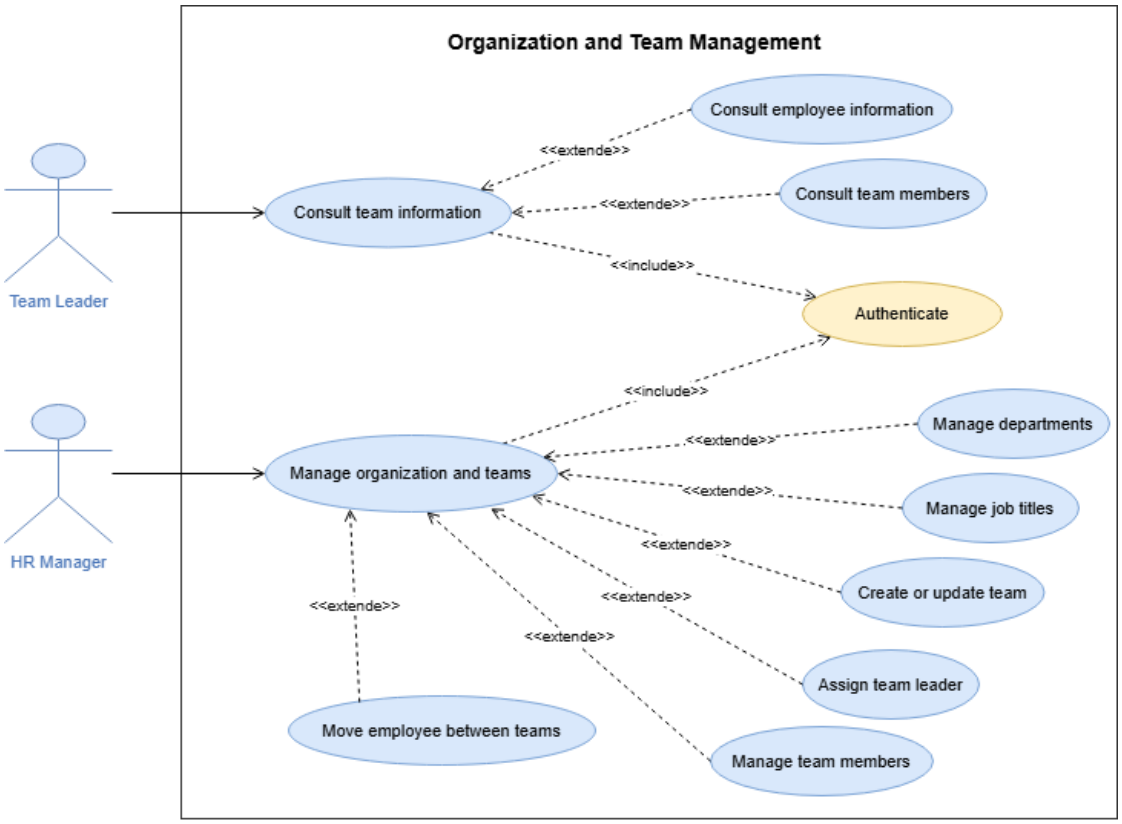


Figure 126: Use Case Diagram – Organization and Team Management

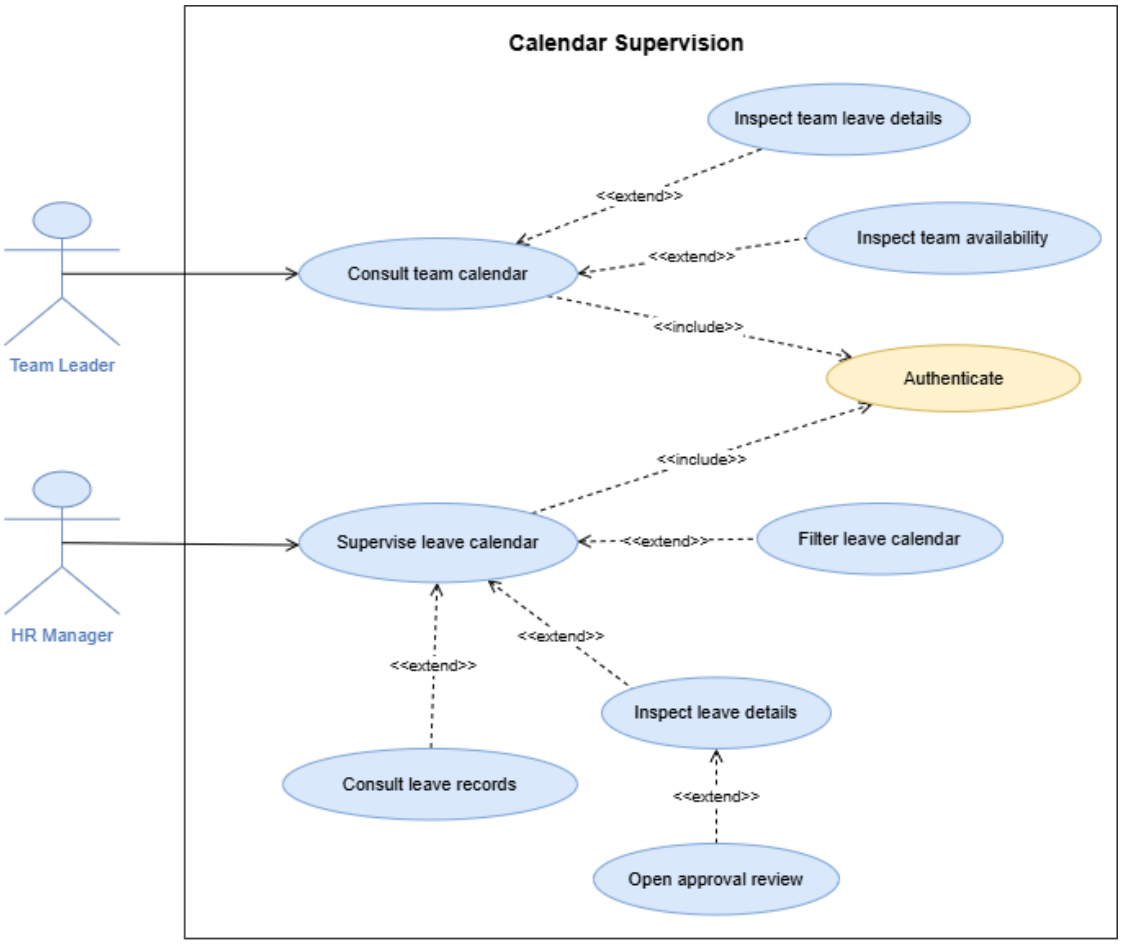


Figure 127: Use Case Diagram – Calendar Supervision

System Sequence Diagrams

These diagrams show the main user–system exchanges for Sprint 4 HR management and calendar workflows.

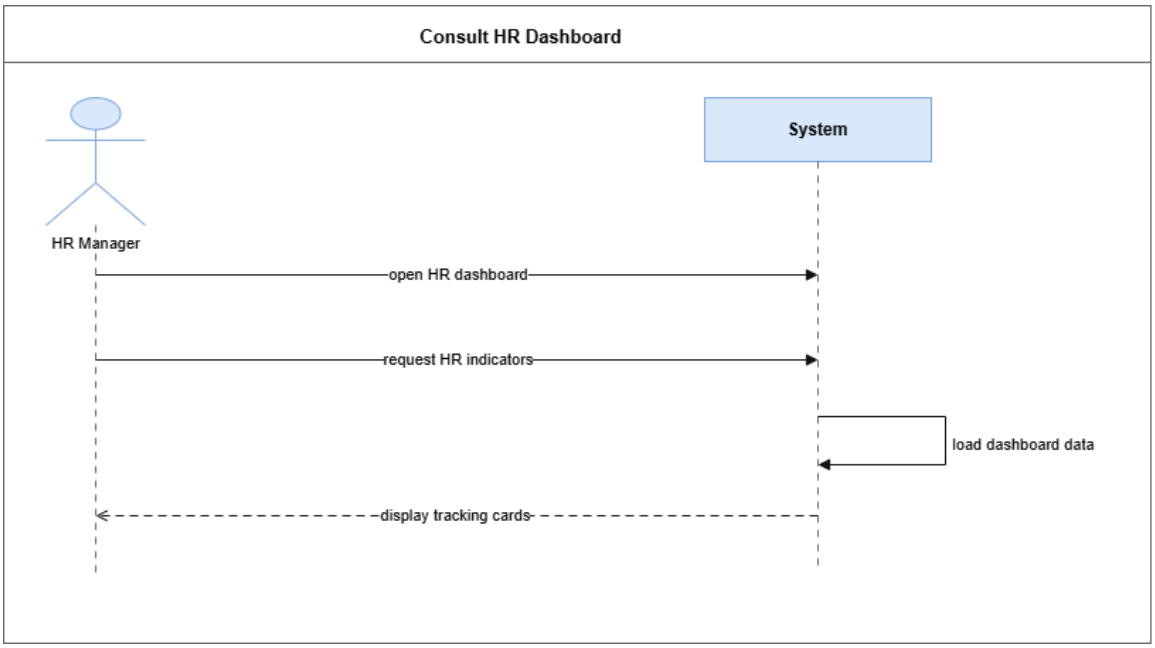


Figure 128: System Sequence Diagram – Consult HR Dashboard

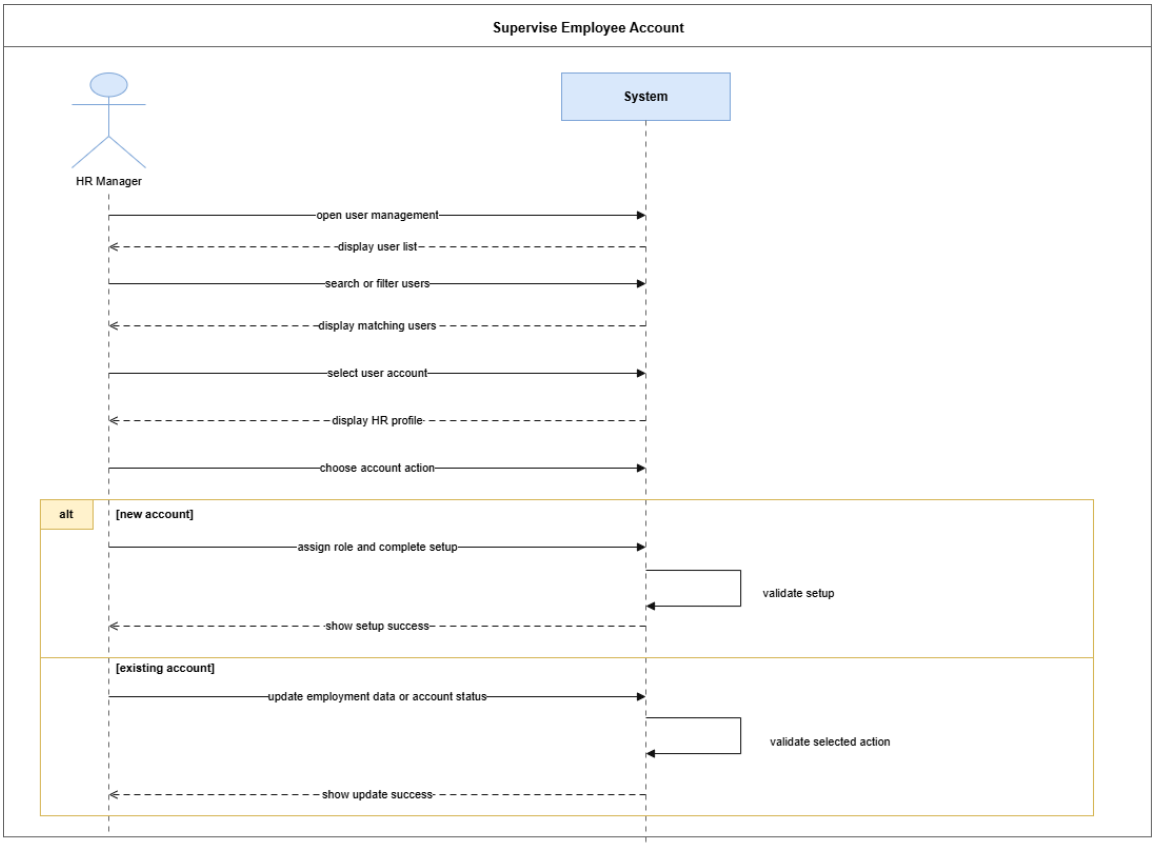


Figure 129: System Sequence Diagram – Supervise Employee Account

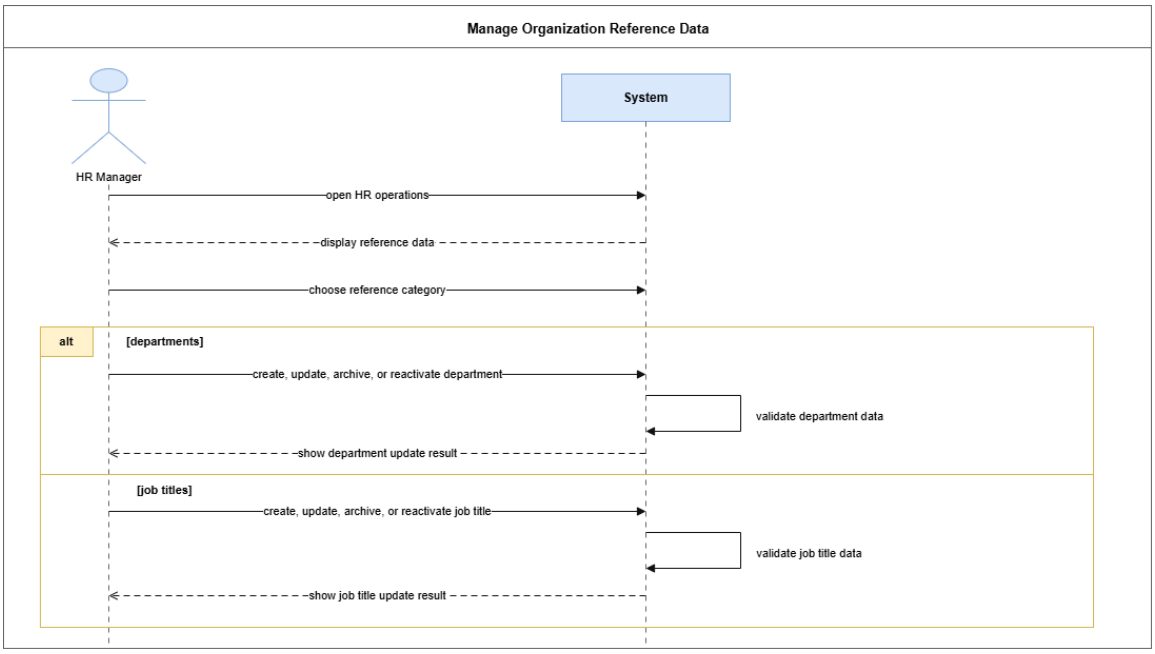


Figure 130: System Sequence Diagram – Manage Organization Reference Data

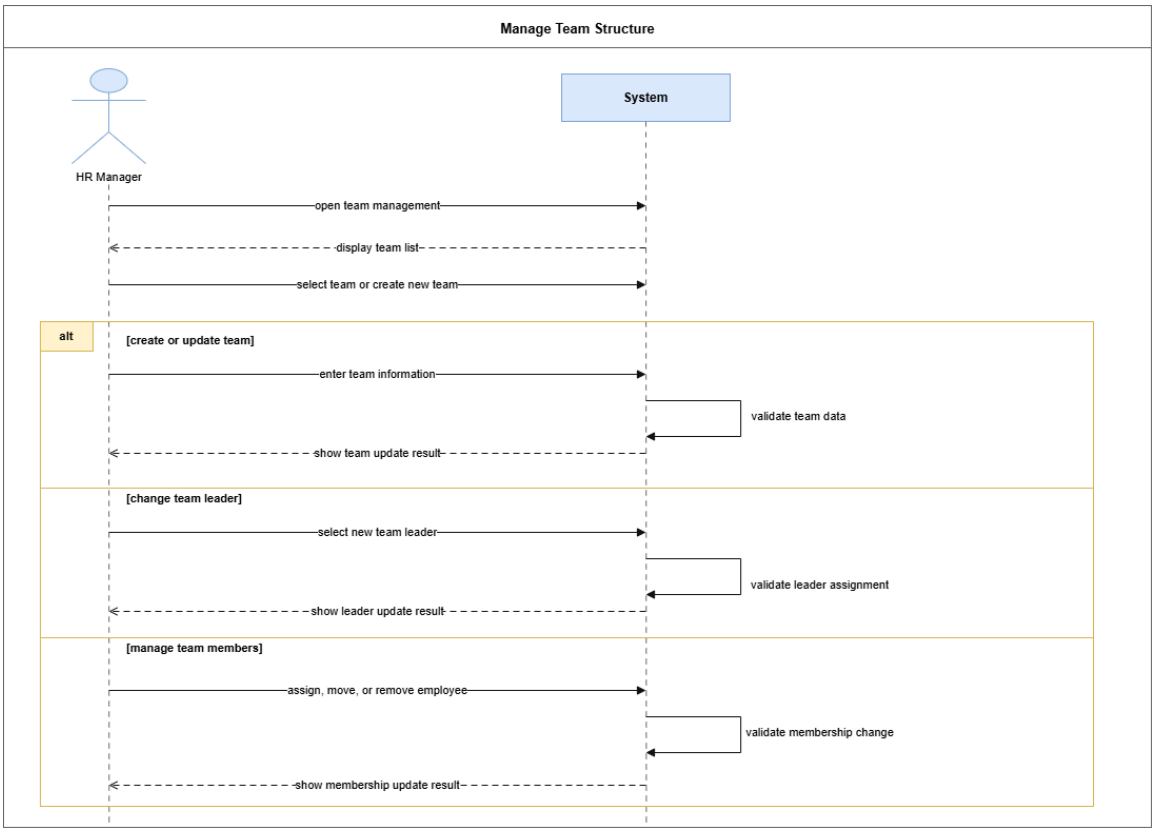


Figure 131: System Sequence Diagram – Manage Team Structure

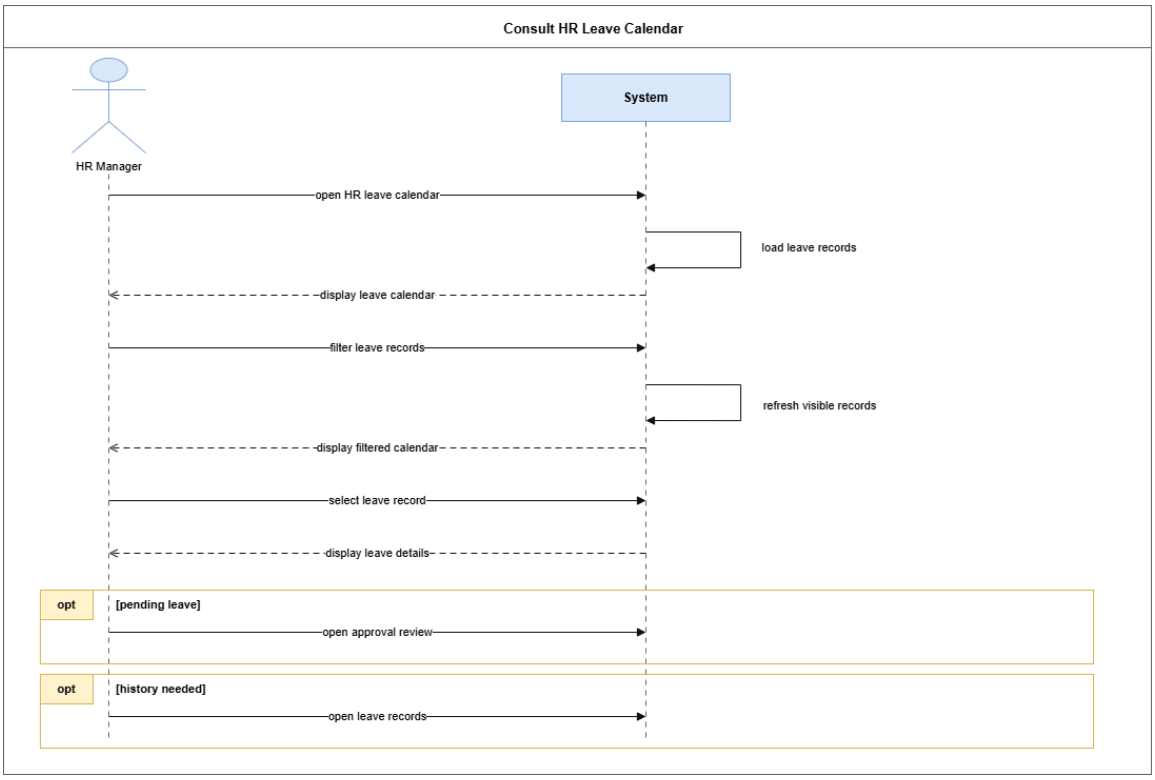


Figure 132: System Sequence Diagram – Consult HR Leave Calendar

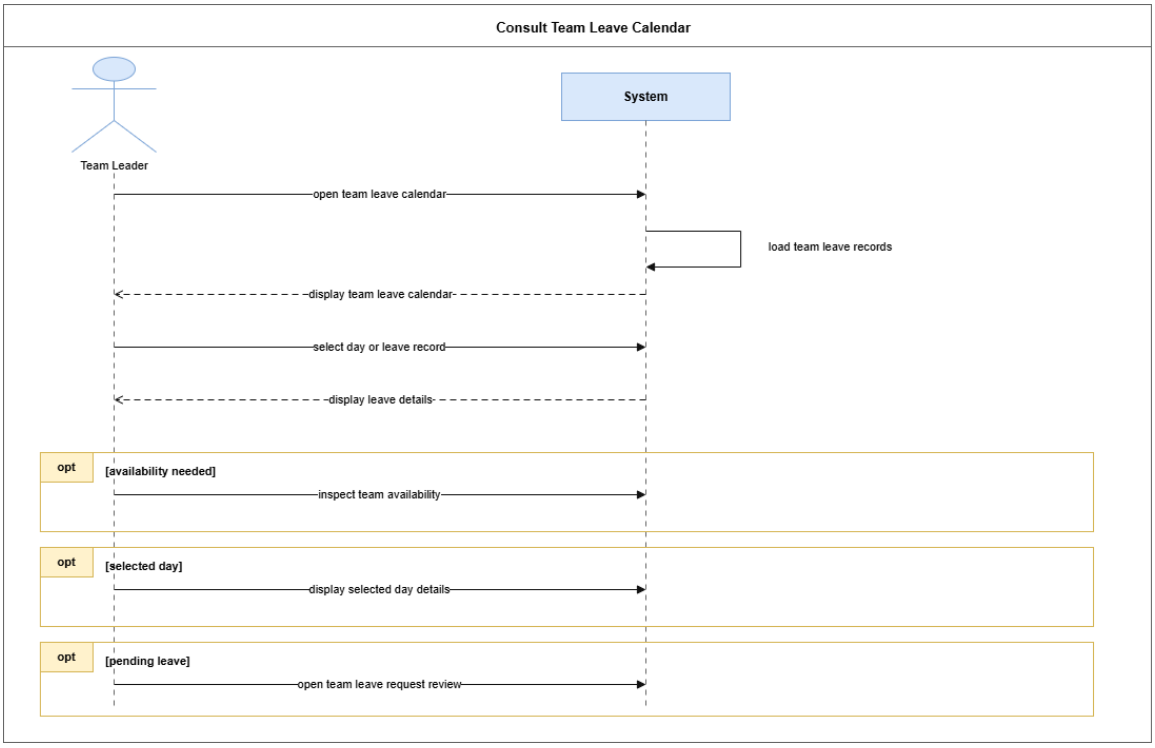


Figure 133: System Sequence Diagram – Consult Team Leave Calendar

Analysis

The Sprint 4 domain covers User accounts, Departments, Job Titles, Teams, team membership, and the leave requests used in the calendar views.

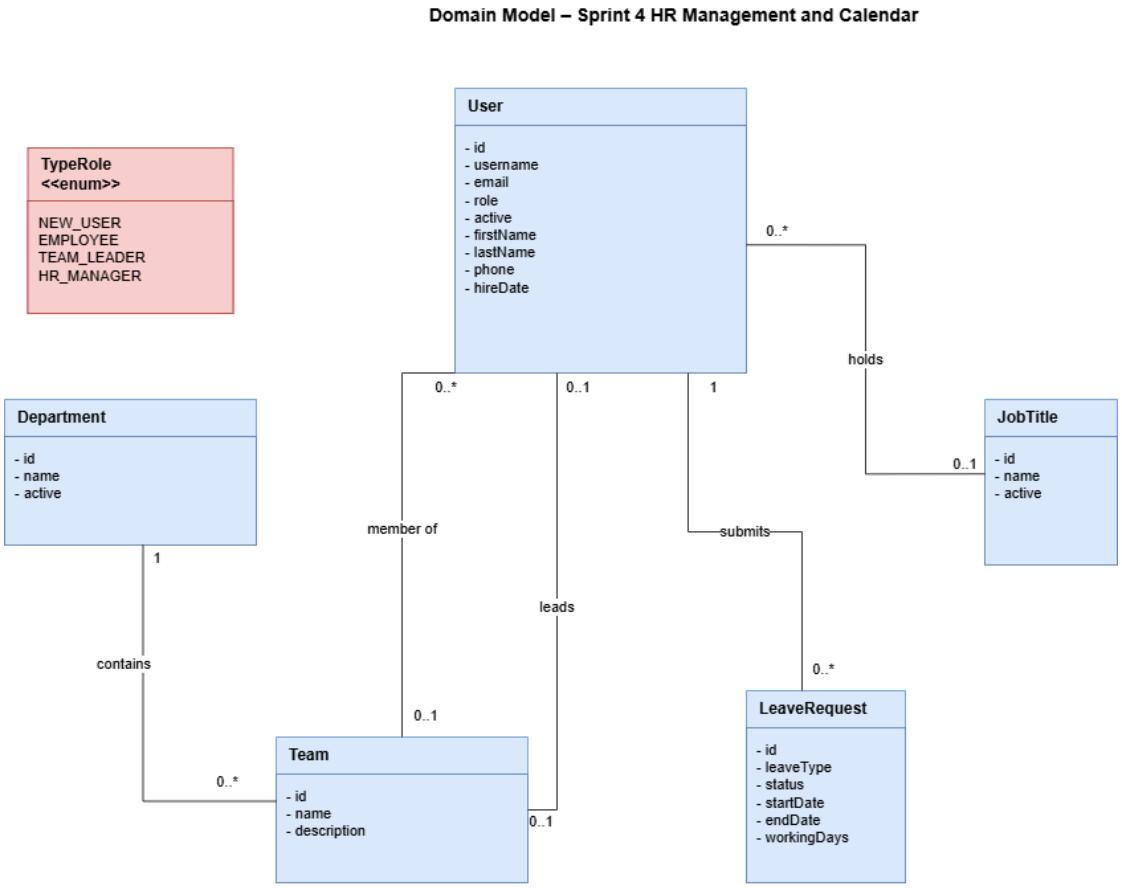


Figure 134: Domain Model – Sprint 4 HR Management and Calendar

Participant Class Diagrams

These diagrams show the main frontend and backend participants for HR employee account supervision, organization reference data management, team structure management, and leave calendar consultation.

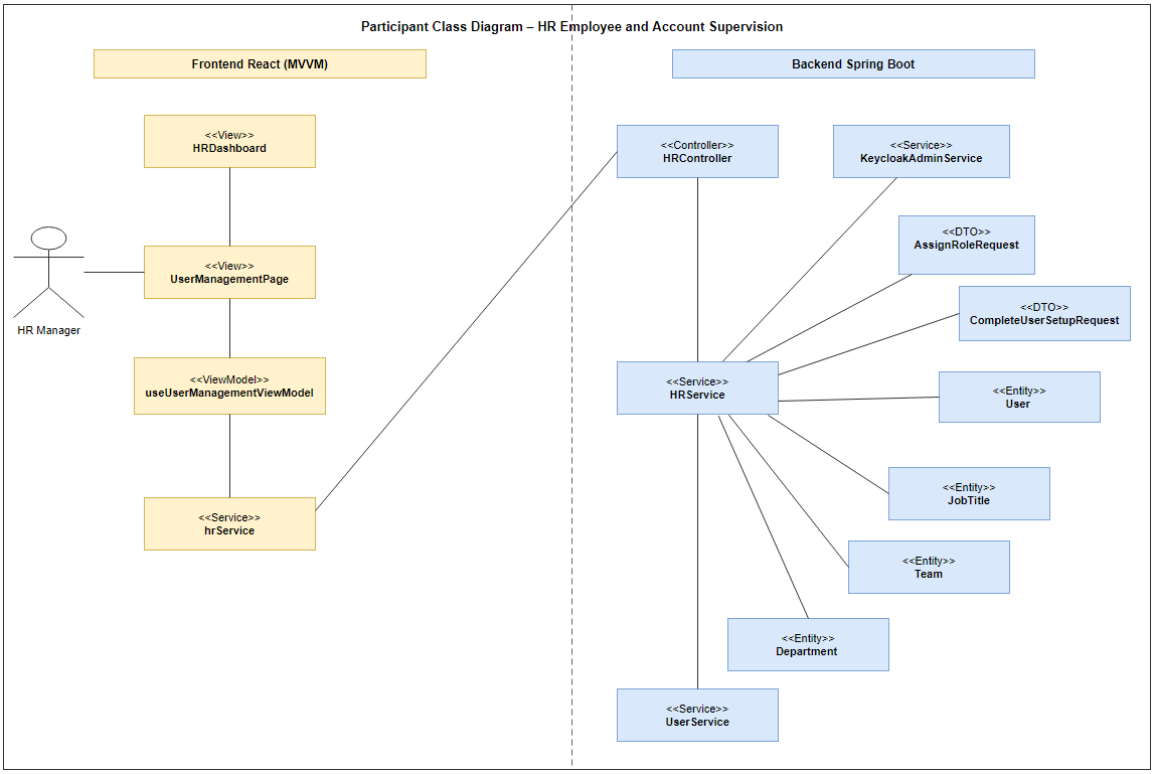


Figure 135: Participant Class Diagram – HR Employee and Account Supervision

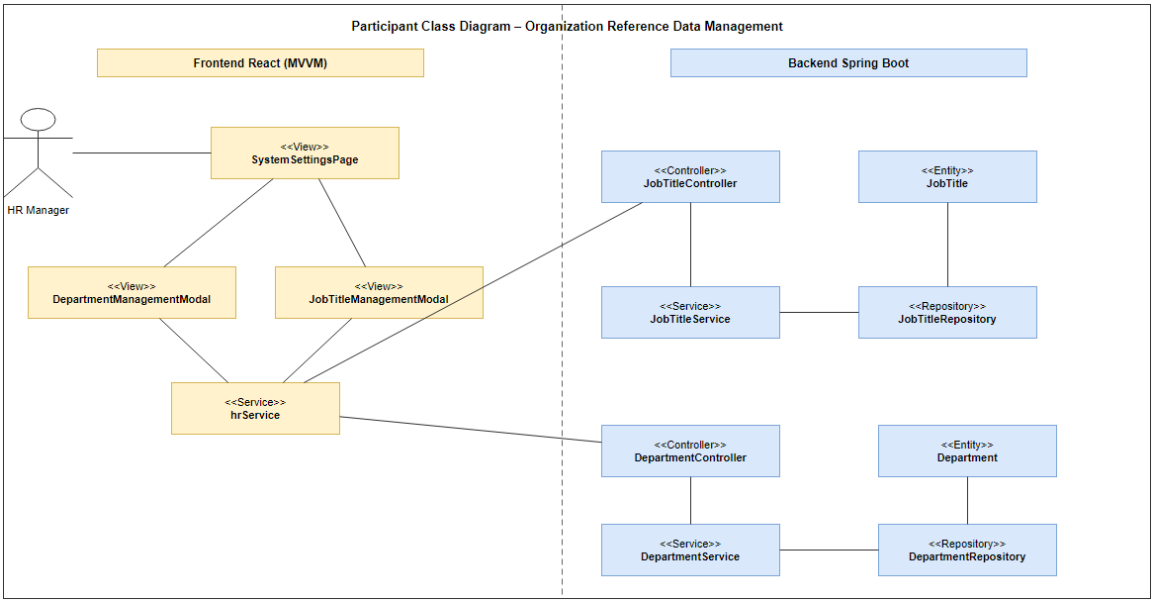


Figure 136: Participant Class Diagram – Organization Reference Data Management

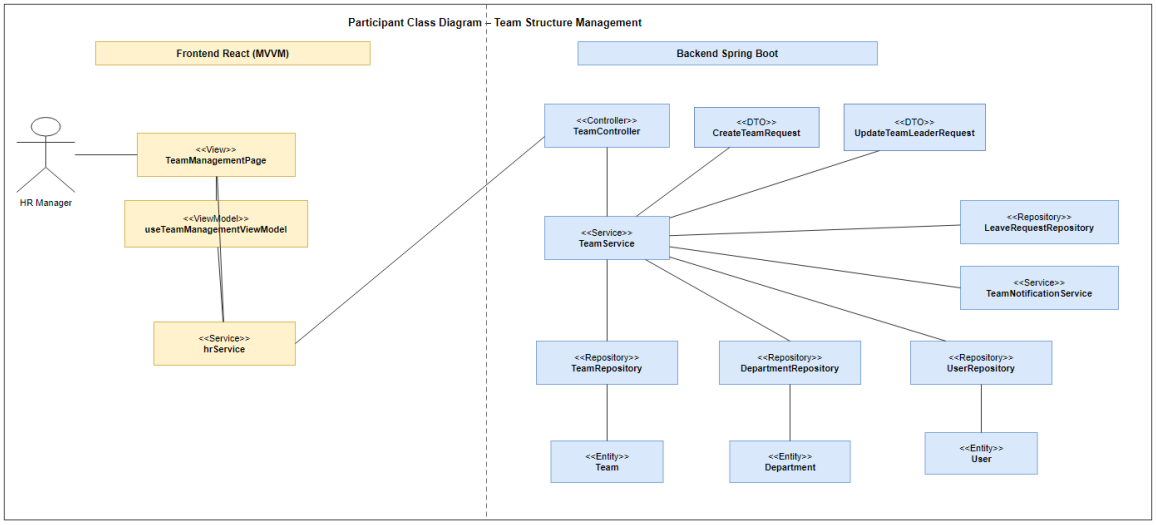


Figure 137: Participant Class Diagram – Team Structure Management

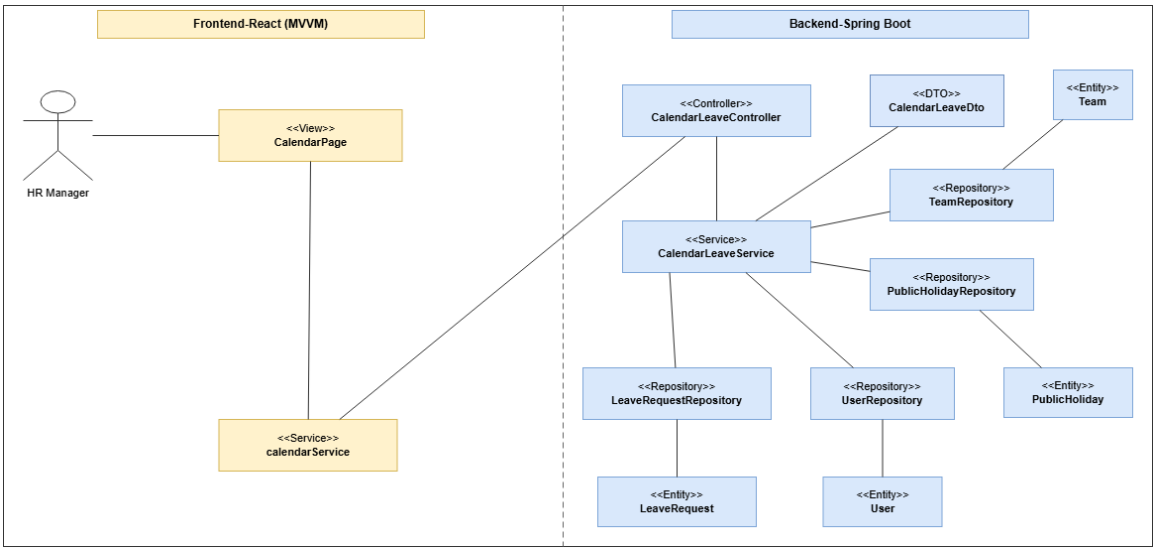


Figure 138: Participant Class Diagram – HR Leave Calendar Consultation

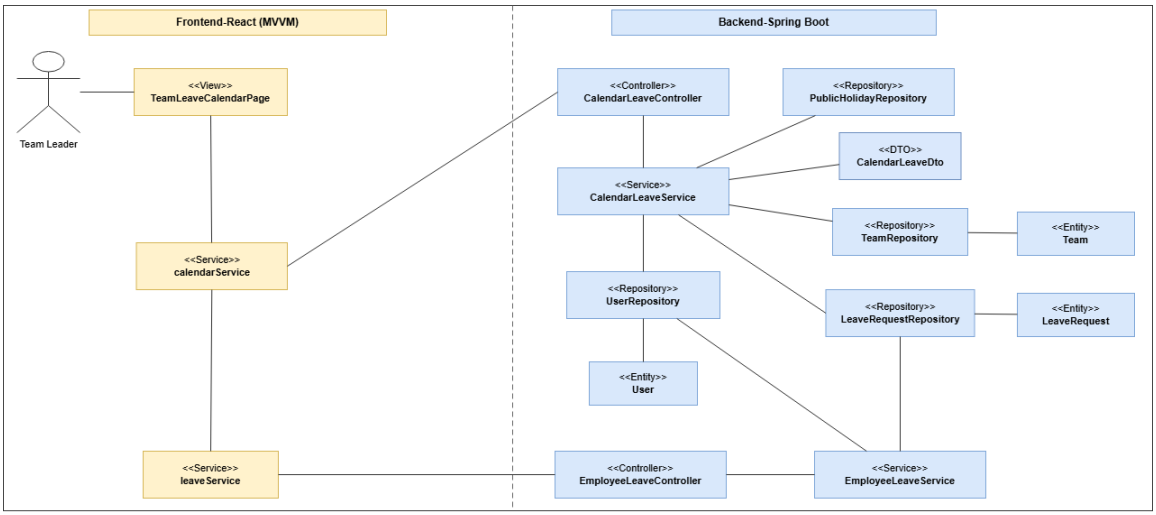


Figure 139: Participant Class Diagram – Team Leave Calendar Consultation

Design Class Diagrams

These diagrams show the frontend MVVM classes and backend Spring Boot classes used for HR user management, organization setup, team administration, and leave calendar consultation.

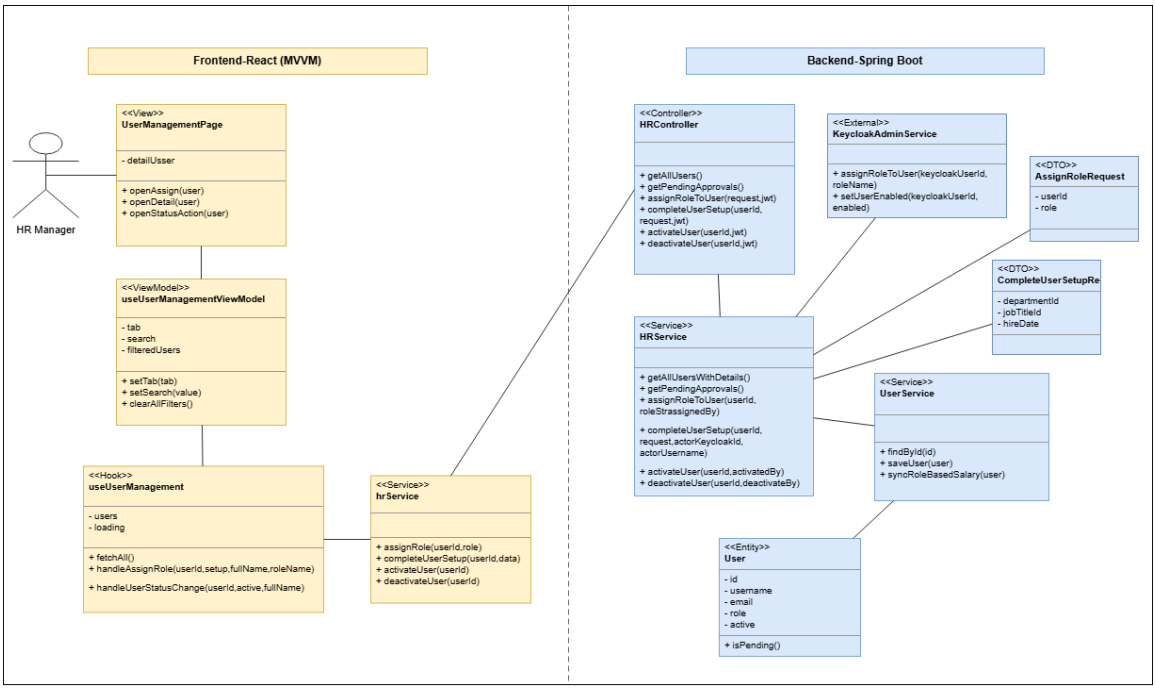


Figure 140: Design Class Diagram – HR User Management

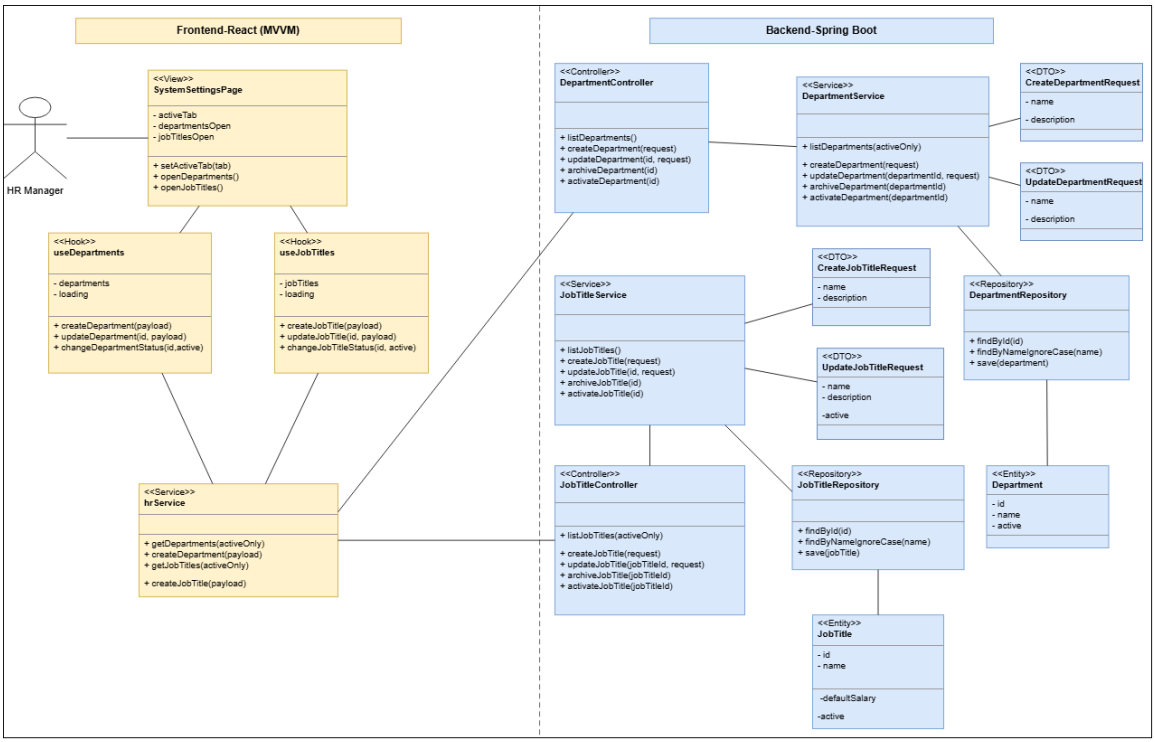


Figure 141: Design Class Diagram – HR Organization Setup

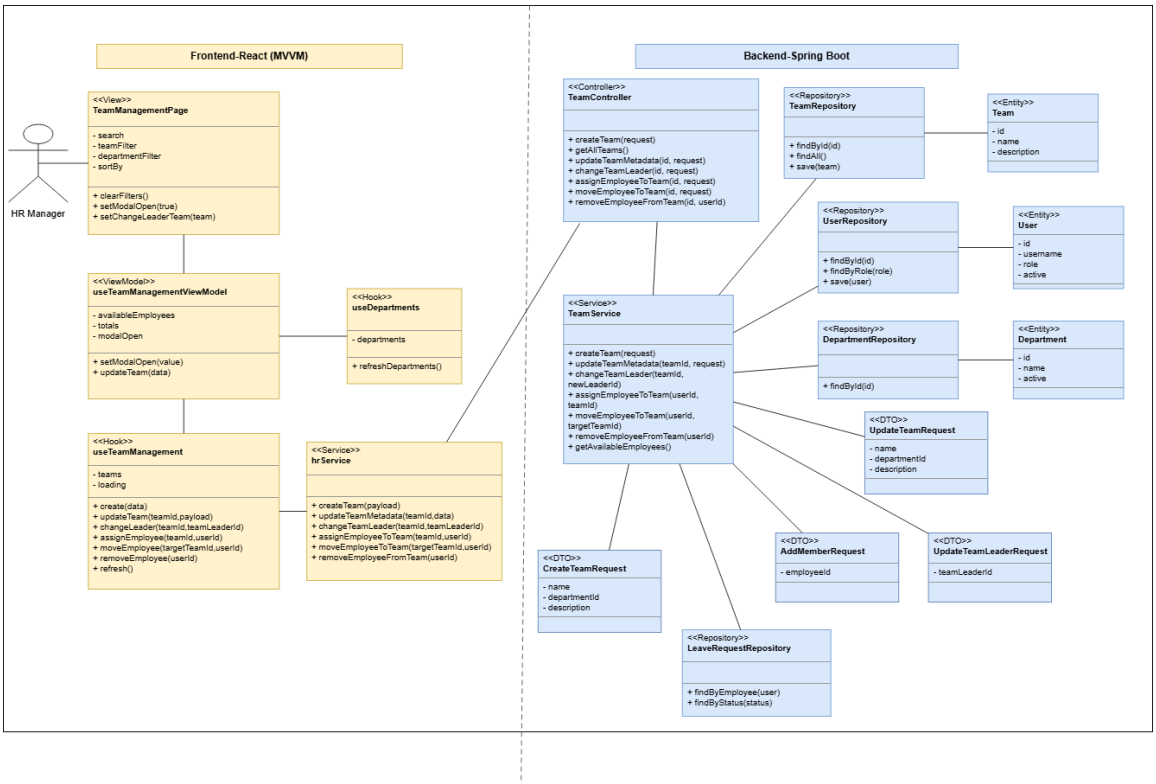


Figure 142: Design Class Diagram – HR Team Management

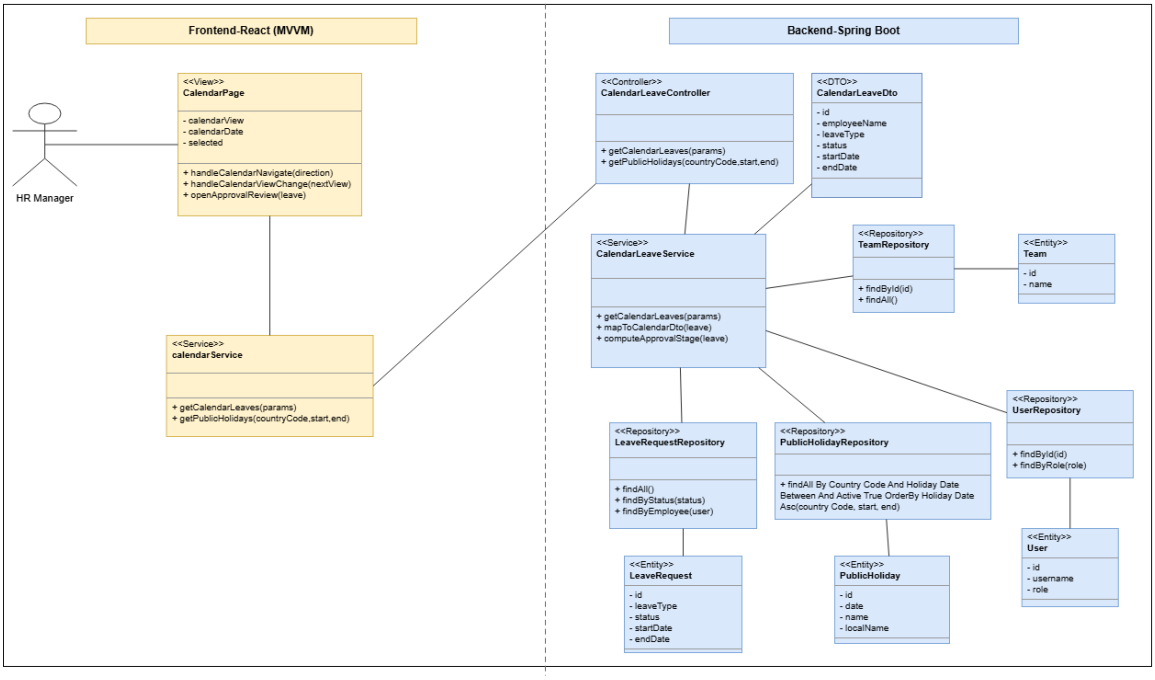


Figure 143: Design Class Diagram – HR Leave Calendar

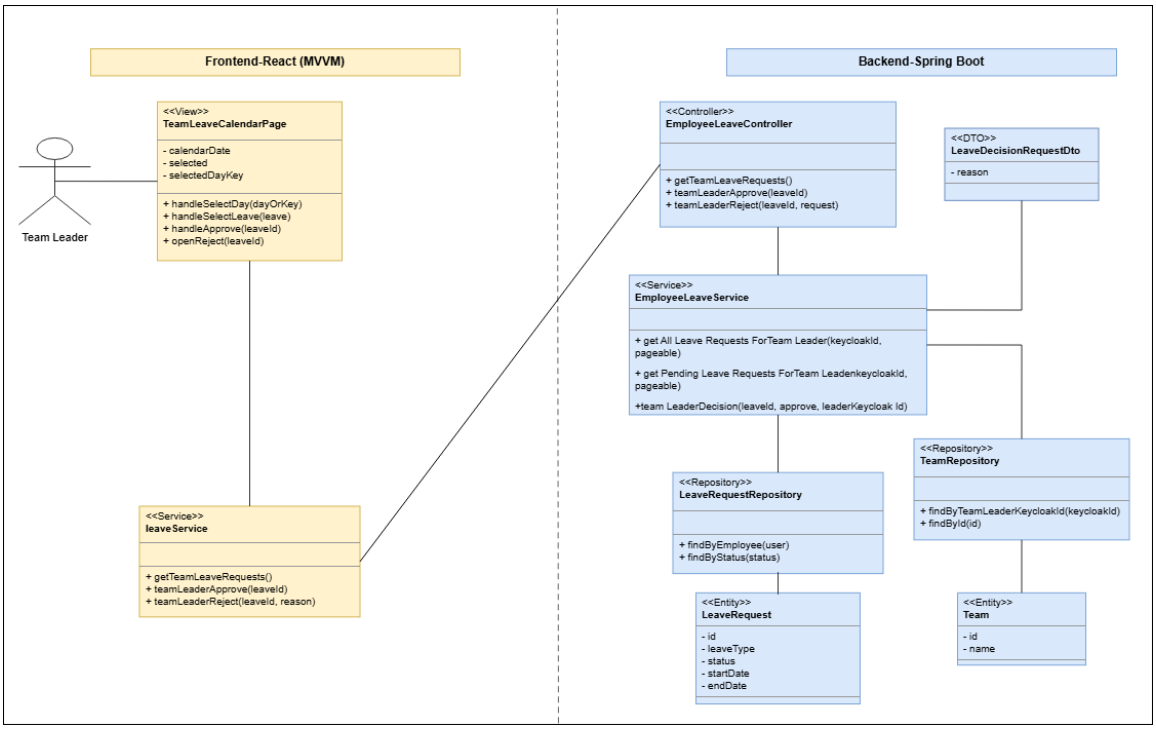


Figure 144: Design Class Diagram – Team Leave Calendar

Conception Sequence Diagrams

These diagrams trace the interactions between the React frontend and the Spring Boot backend for HR account supervision, organization setup, team management, and leave calendar

consultation.

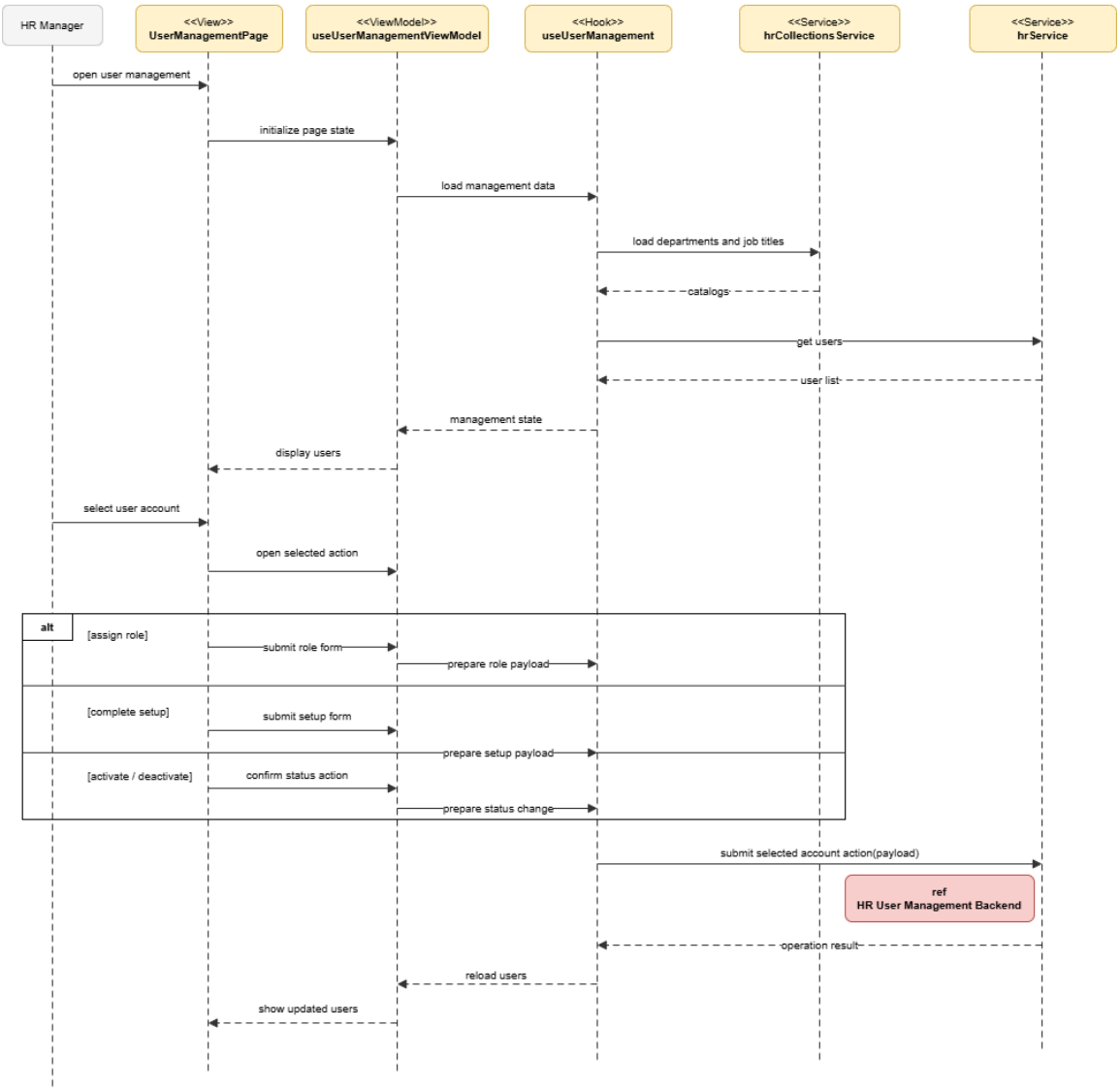


Figure 145: Conception Sequence Diagram – HR User Management Frontend

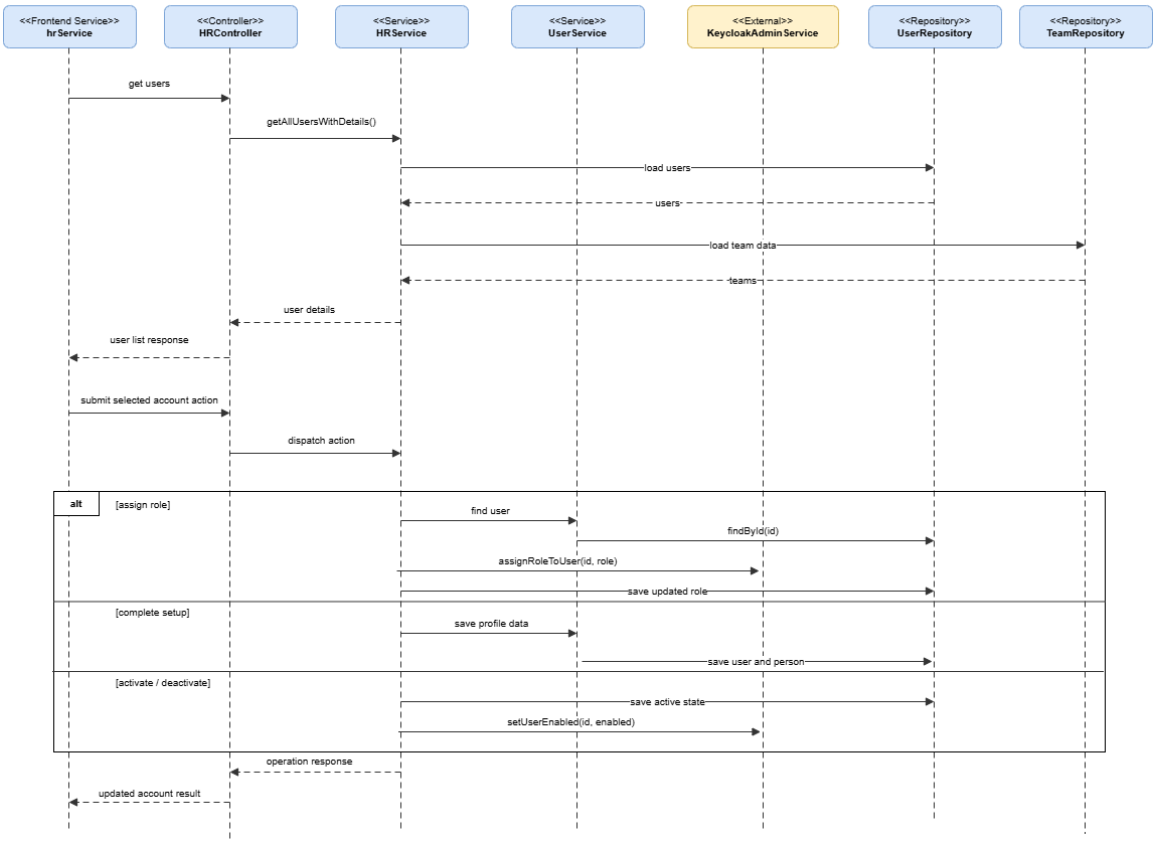


Figure 146: Conception Sequence Diagram – HR User Management Backend

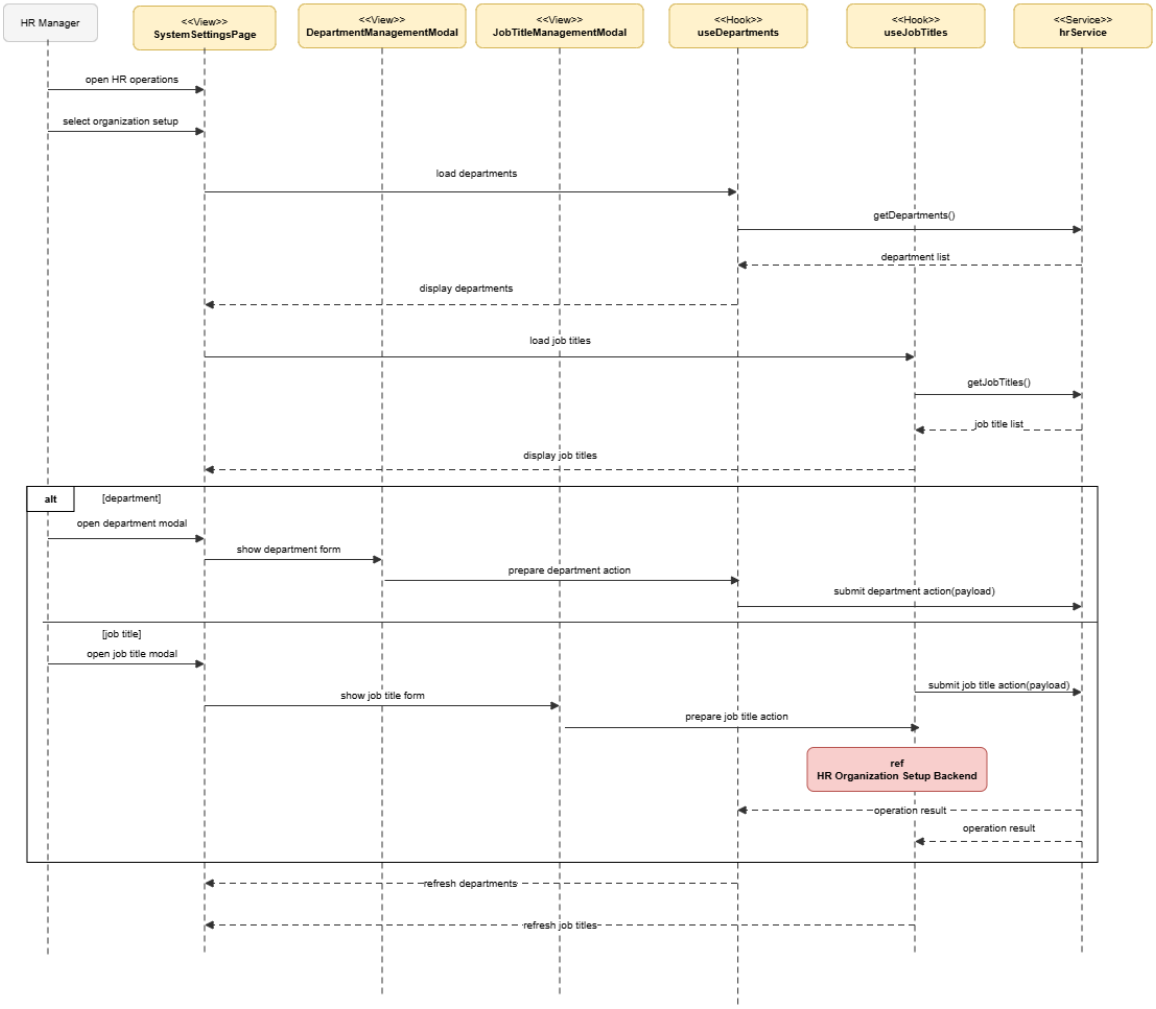


Figure 147: Conception Sequence Diagram – HR Organization Setup Frontend

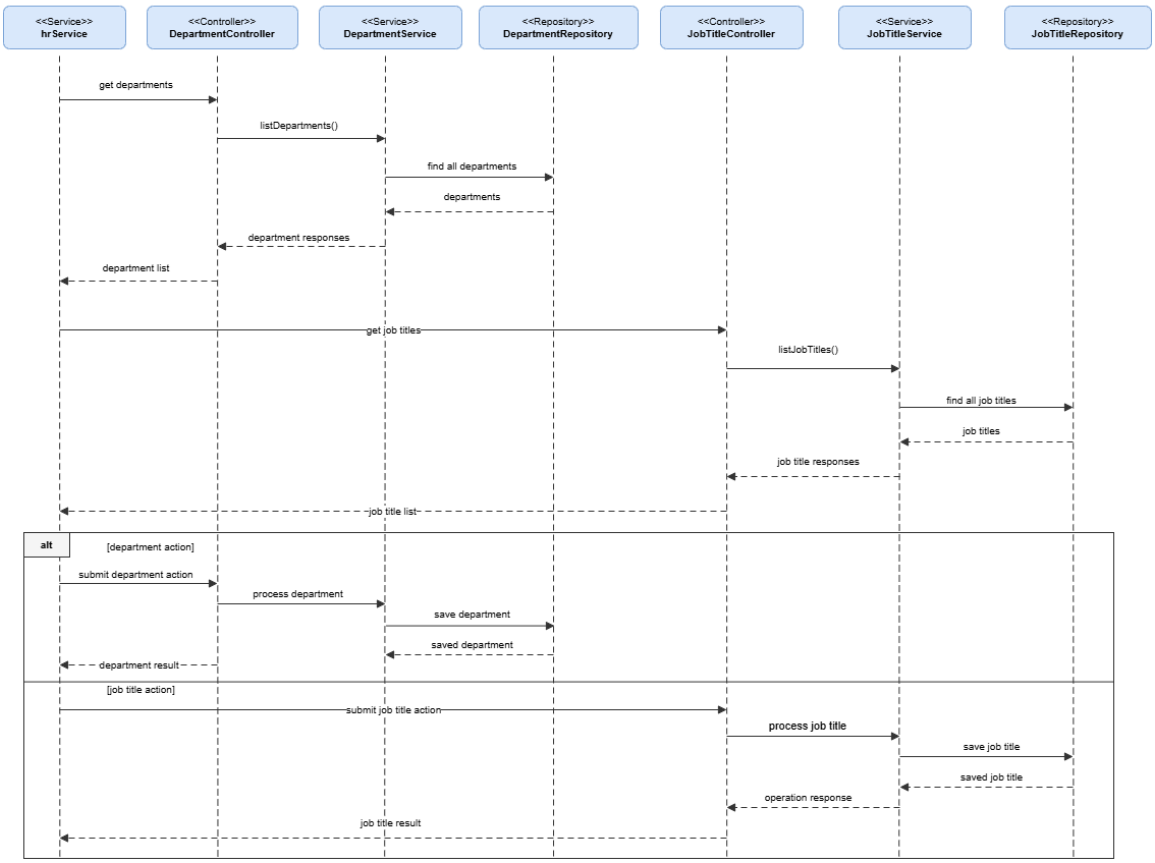


Figure 148: Conception Sequence Diagram – HR Organization Setup Backend

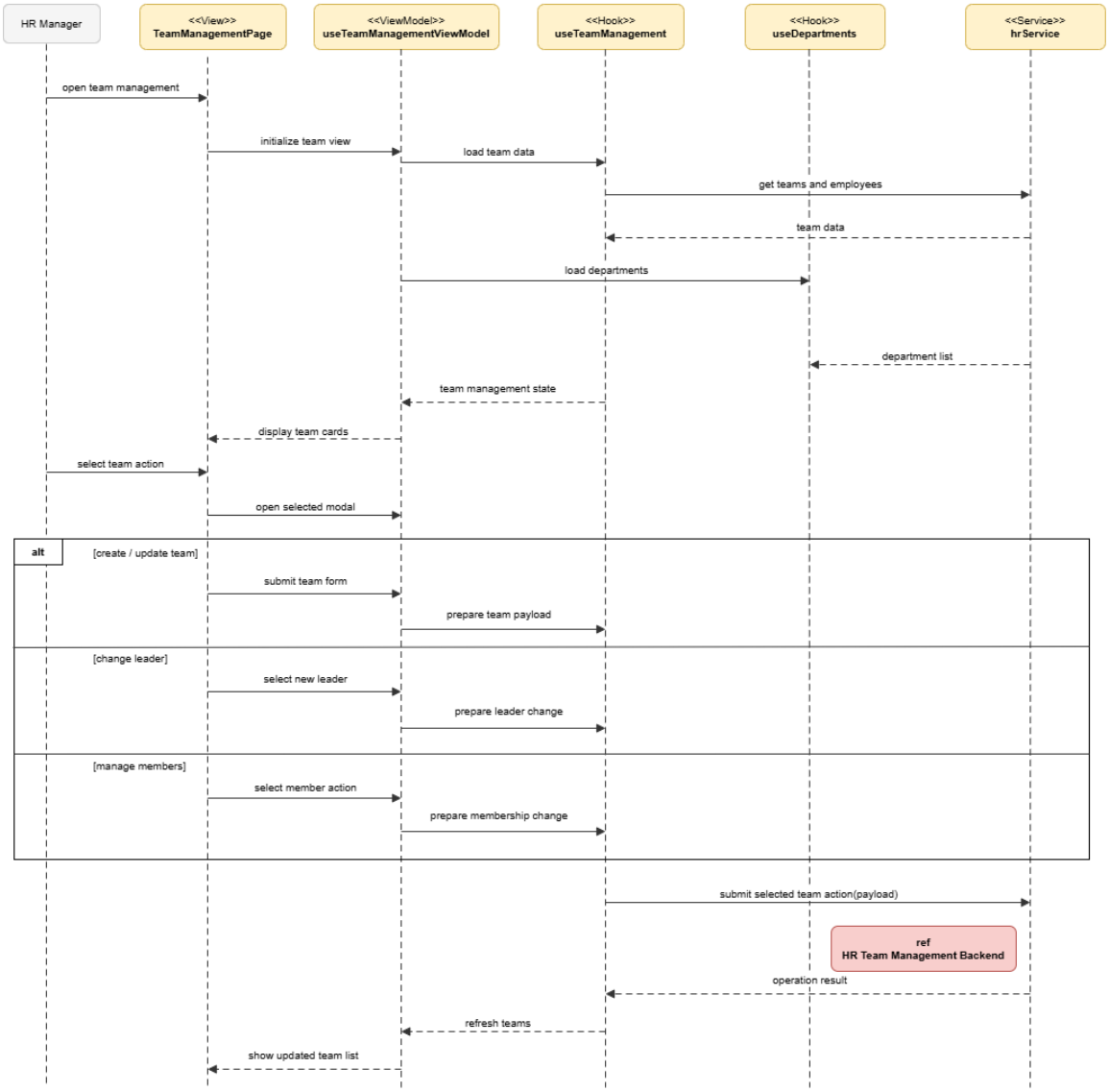


Figure 149: Conception Sequence Diagram – HR Team Management Frontend

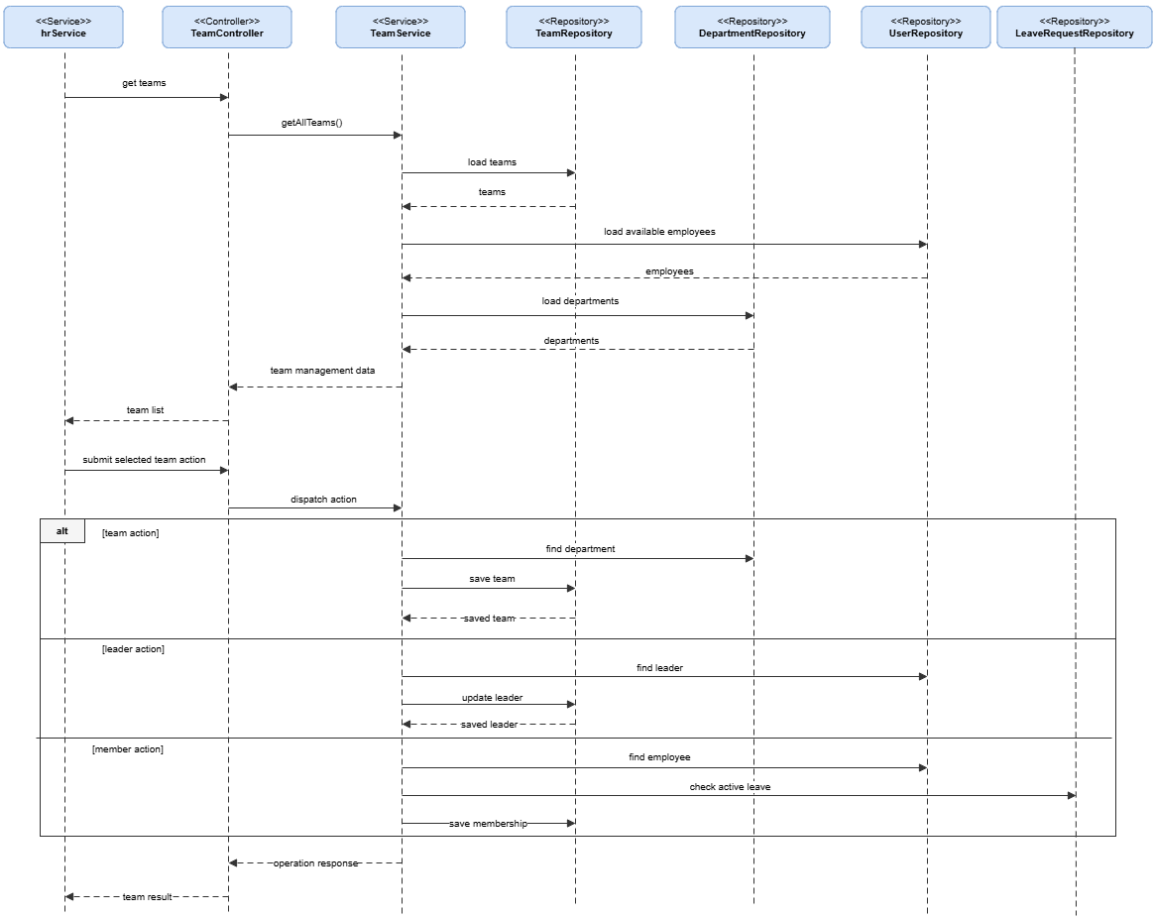


Figure 150: Conception Sequence Diagram – HR Team Management Backend

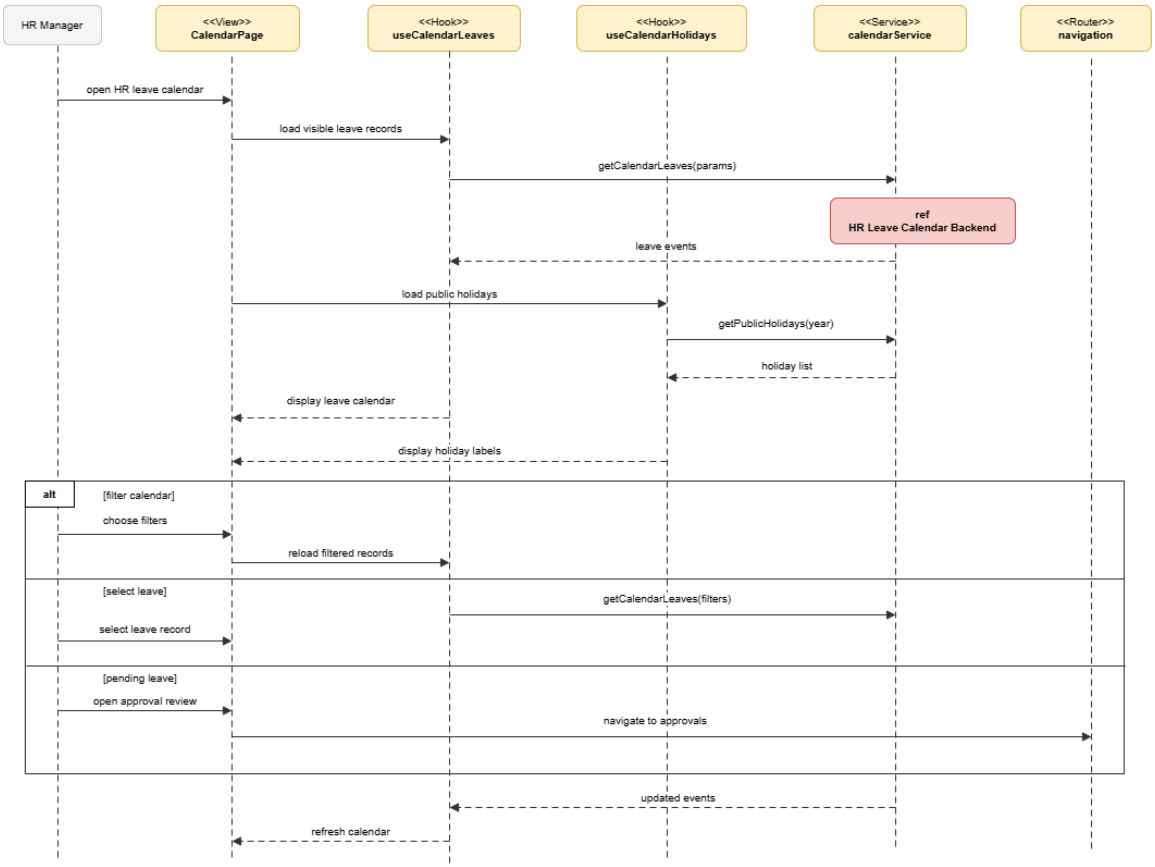


Figure 151: Conception Sequence Diagram – HR Leave Calendar Frontend

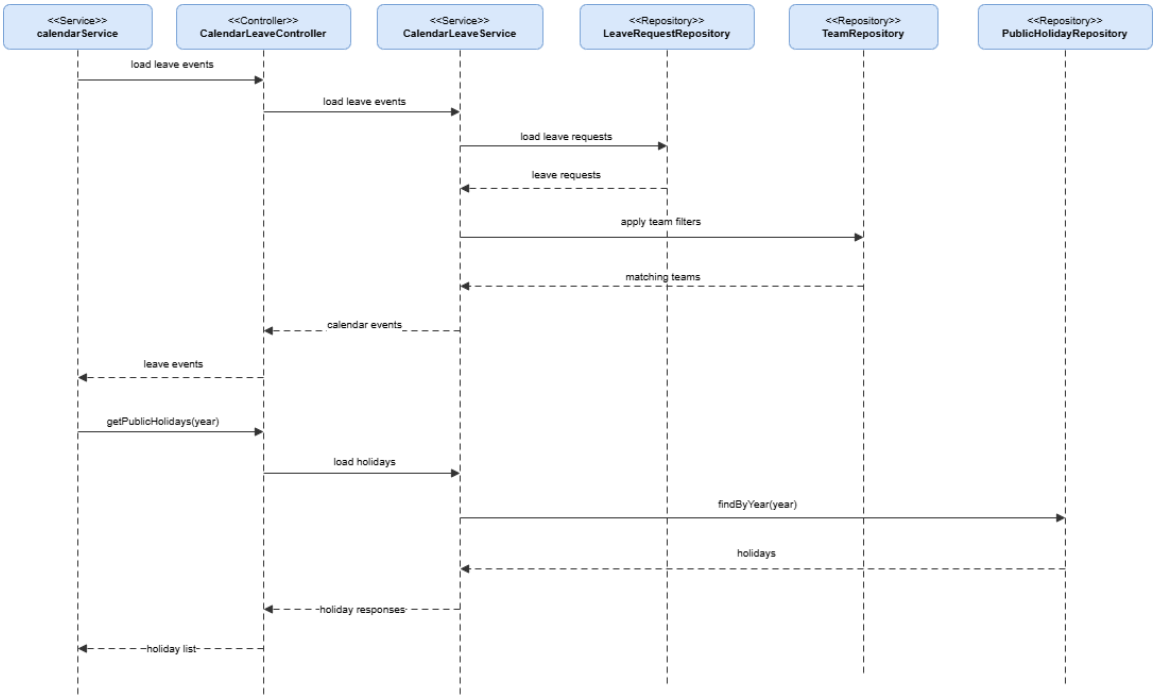


Figure 152: Conception Sequence Diagram – HR Leave Calendar Backend

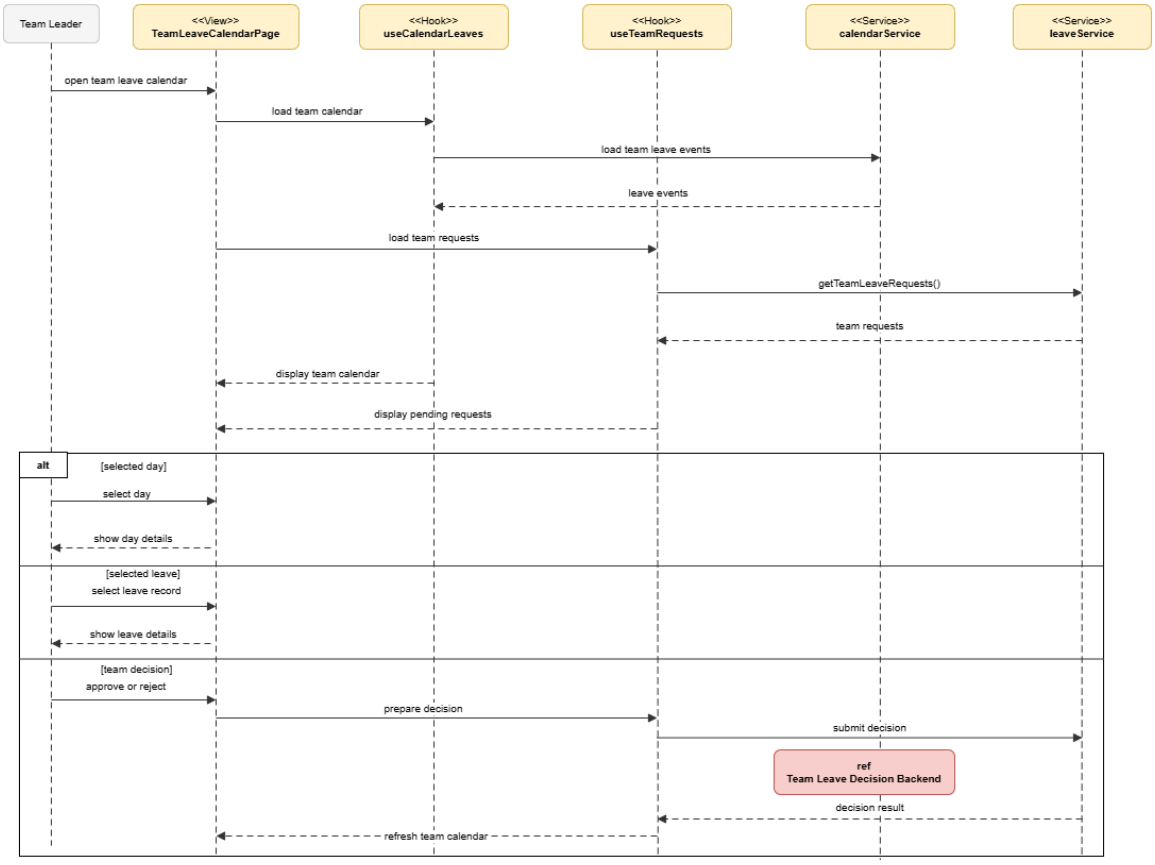


Figure 153: Conception Sequence Diagram – Team Leave Calendar Frontend

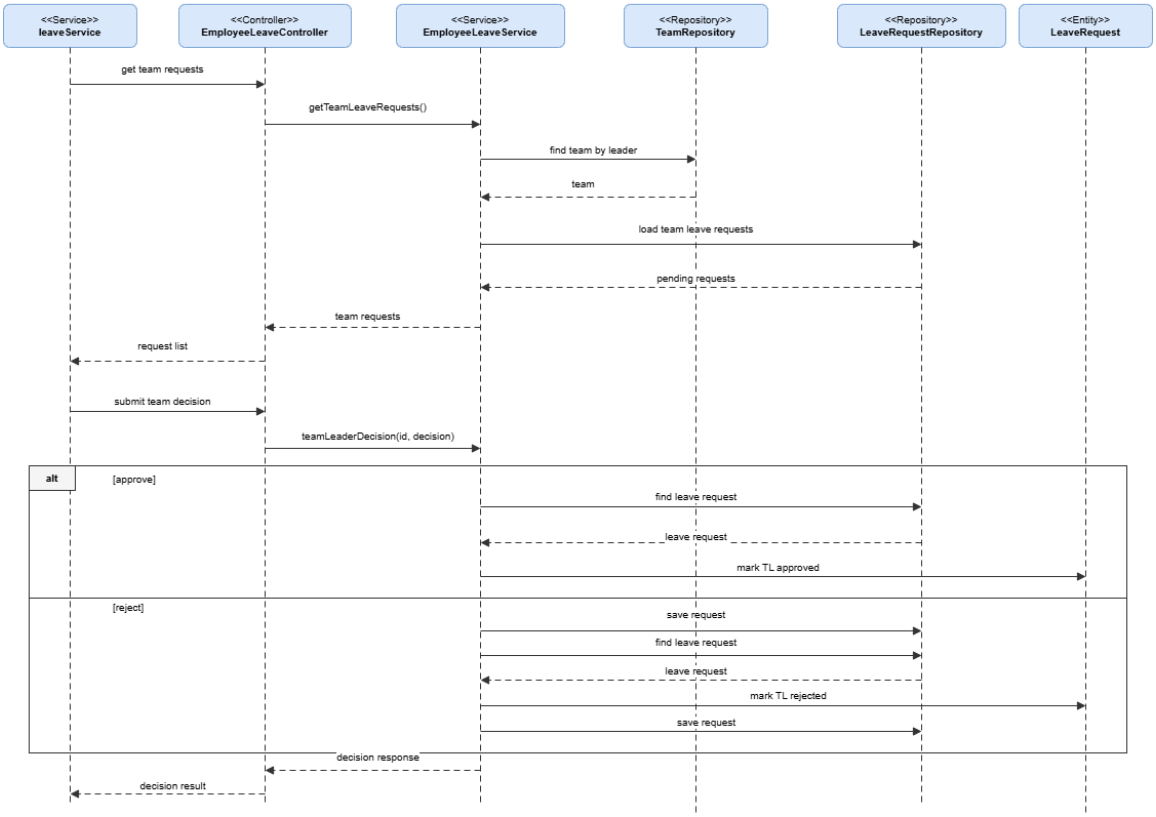


Figure 154: Conception Sequence Diagram – Team Leave Decision Backend

Realization

This subsection shows the interface screens for HR supervision, employee records, team administration, and calendar views.

The screenshots below cover the main implemented flows in the application.

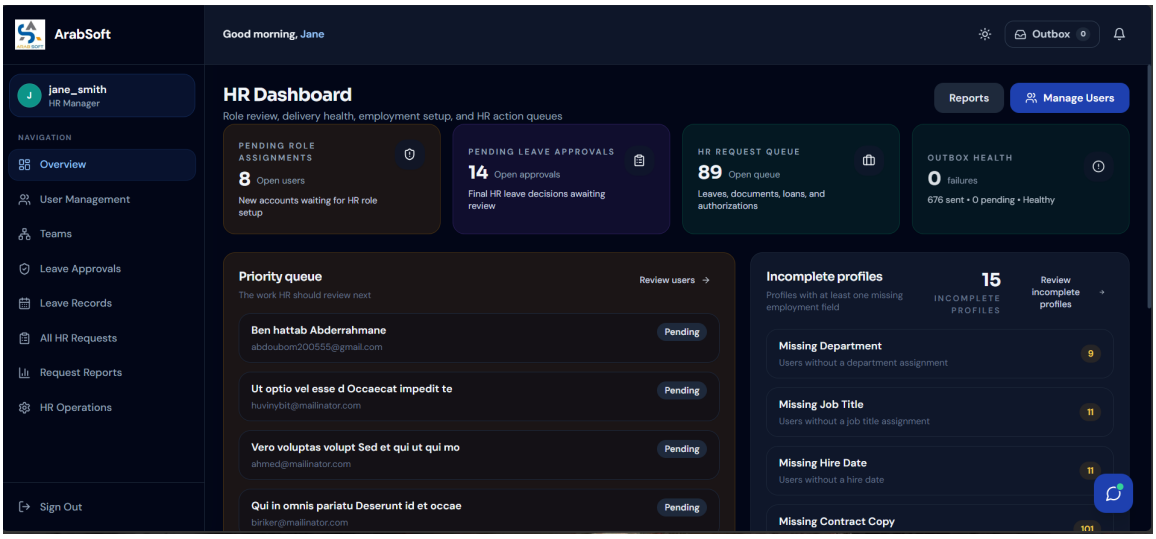


Figure 155: HR Dashboard Interface

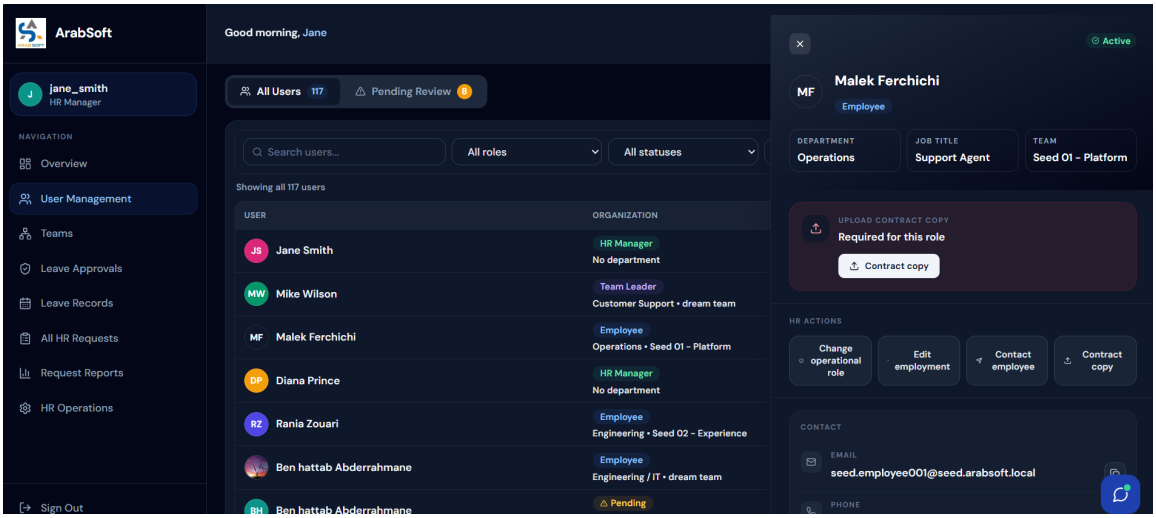


Figure 156: HR User Management Interface

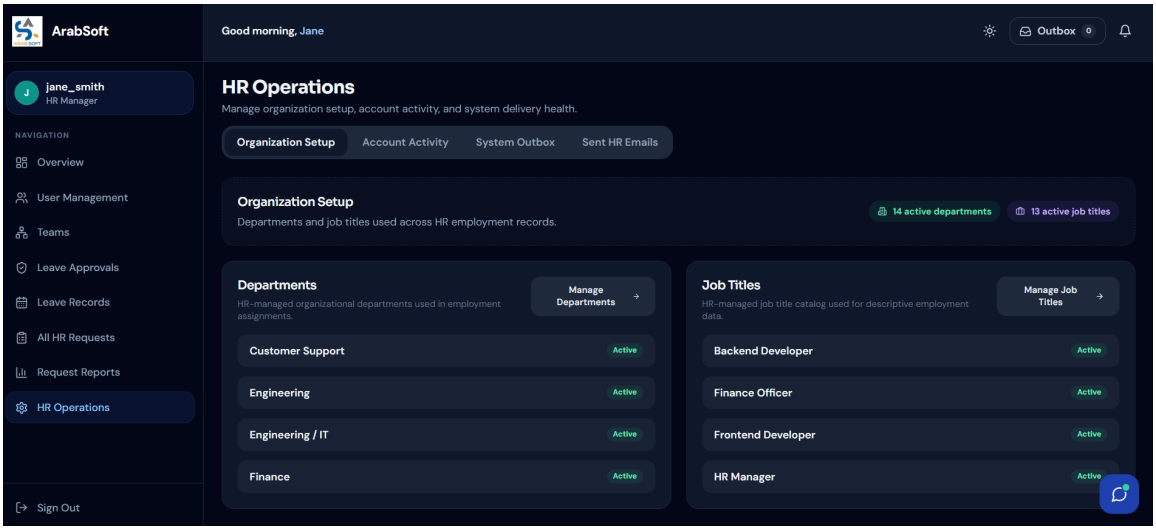


Figure 157: Team Management Interface

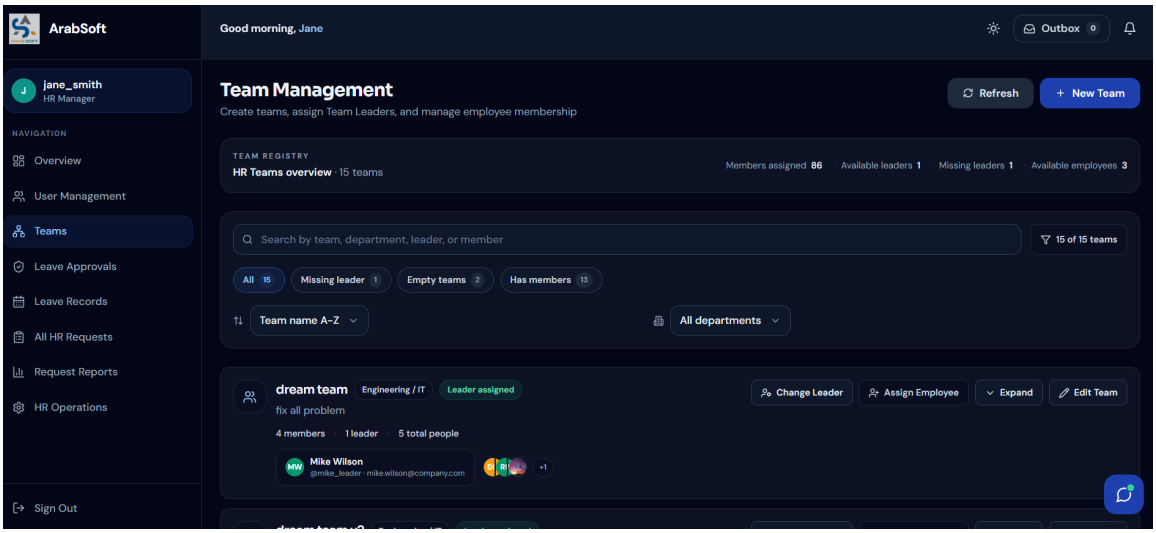


Figure 158: Organization Reference Data Interface

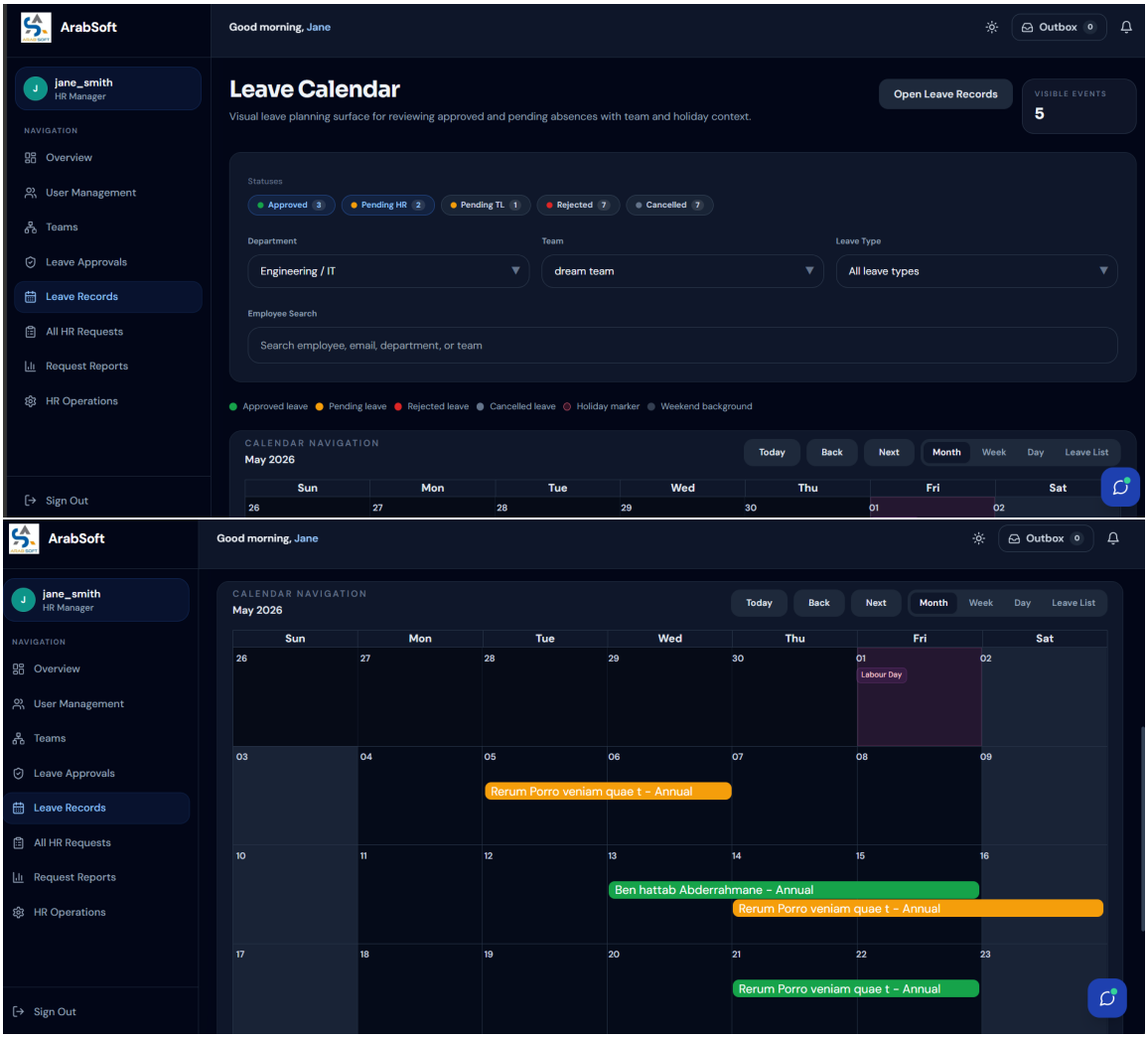


Figure 159: HR Leave Calendar Interface

5. Sprint Review

Sprint 4 covered HR management and calendar supervision. The sprint included HR user management, employee records, role assignment, account activation and deactivation, department and job title management, team creation, team leader assignment, team membership changes, and leave calendar consultation. Team leaders also received a dedicated team calendar and leave-decision support within their team scope.

6. Sprint Retrospective

What went well	What needs improvement
Sprint 4 features were completed and integrated successfully.	**Some code still needs final cleanup and consistency checks.**
HR user, team, and calendar workflows now work in the platform.	**A few screens need small frontend adjustments after final testing.**

Table 9: Sprint Retrospective – Sprint 4

Conclusion

This chapter presented Release 2 of the ArabSoft platform. Sprint 3 introduced the administrative request workflows for documents, loans, and authorizations, including HR decisions, request history, notifications, meetings, and final-file handling. Sprint 4 extended the platform with HR management and calendar supervision by covering user records, roles, account status, organization reference data, teams, team membership, and role-aware leave calendar views.

Together, these two sprints added request workflows, HR supervision, team management, and leave calendar access to the platform. The next release covers project tracking, task management, report export, and AI-assisted support features.

Chapter 5

Release 3 – Advanced Support and Intelligence

Introduction

This chapter presents Release 3 of the ArabSoft platform. It covers the final advanced scope of the project through two sprints. Sprint 5 focuses on project management, task assignment, task progress tracking, and HR report export. Sprint 6 is reserved for AI assistance, deployment preparation, and final stabilization. Together, these elements complete the platform by adding collaboration, reporting, and intelligent support features on top of the workflows developed in the previous releases.

I. Release Organization

Release 3 is organized into two final sprints. Sprint 5 covers Projects, Tasks and Reports, while Sprint 6 covers AI, Deployment and Stabilization. This organization separates collaboration and reporting features from the assistant and stabilization work, making the final release easier to document and validate.

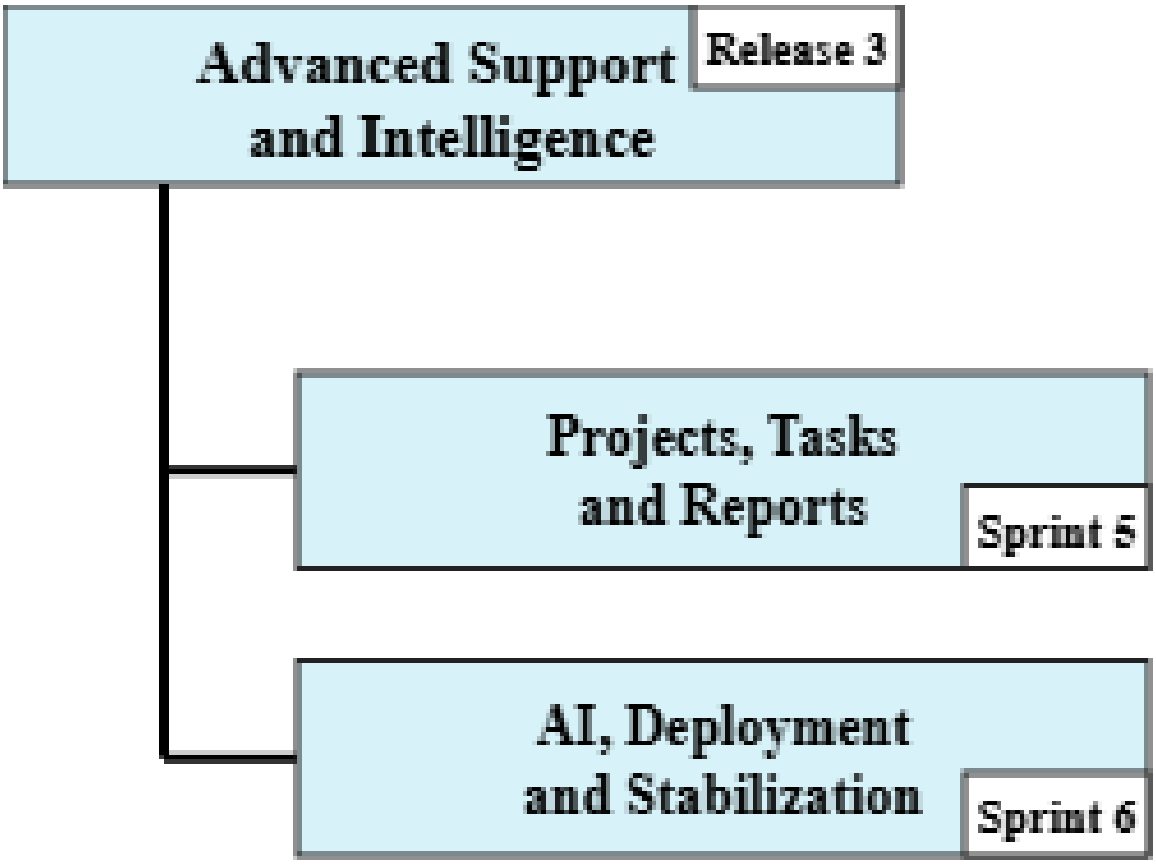


Figure 160: Release 3 Organization

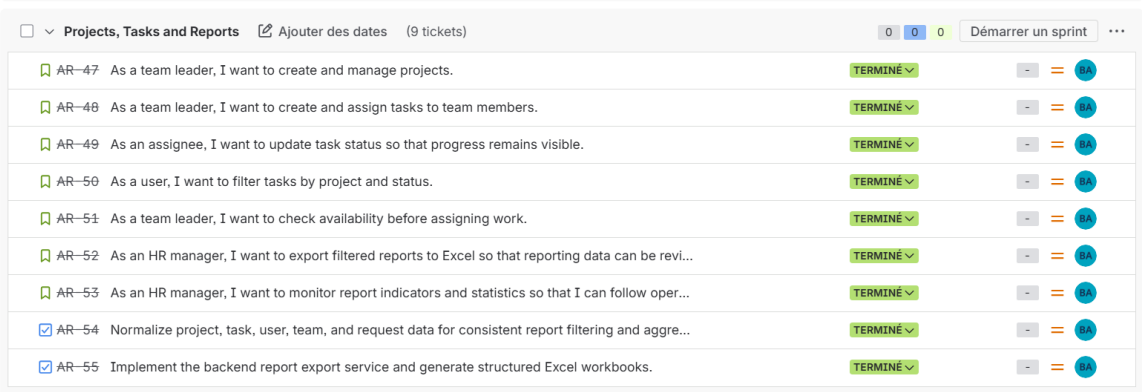
II. Sprint 5: Projects, Tasks and Reports

1. Sprint Goal

This sprint focuses on project and task management for team leaders, assignee-side task progress tracking, and HR reporting through consultation and Excel export. It connects operational planning, task execution, and reporting features while keeping each workflow aligned with the implemented role permissions.

2. Visual Sprint Journal

The visual sprint journal below presents the selected user stories of Sprint 5 and their completion status in Jira.



Visual Sprint Journal – Sprint 5

3. Sprint Backlog

The table below summarizes the Sprint 5 work packages prepared from the product backlog defined in Chapter 2.

Sprint	Stories	Tasks	Duration
Sprint 5	**Project management**	Create, update, delete, and view team-scoped projects	2d
		Keep project consultation limited to the connected team scope	
	Task creation and assignment	Create tasks and assign them using ONE, SELECTED, or ALL modes	2d
		Preview member availability before assigning tasks	
		Keep task assignment aligned with team membership and assignment mode rules	
	Task status update	Allow assigned users to update the task lifecycle from TODO to IN_PROGRESS to DONE	1d
		Track assignee-side progress through the defined lifecycle states	
	HR reports consultation and Excel export	Consult leave, document, loan, and authorization report data	2d
		Filter report results before export	
		Export filtered HR reports to Excel	

Sprint Backlog – Sprint 5

4. Sprint Implementation

The Sprint 5 implementation is presented through the following UML diagrams, covering project management, task assignment, task status update, and HR report export.

Requirements Analysis

The use case diagrams below show the main actor interactions for Sprint 5.

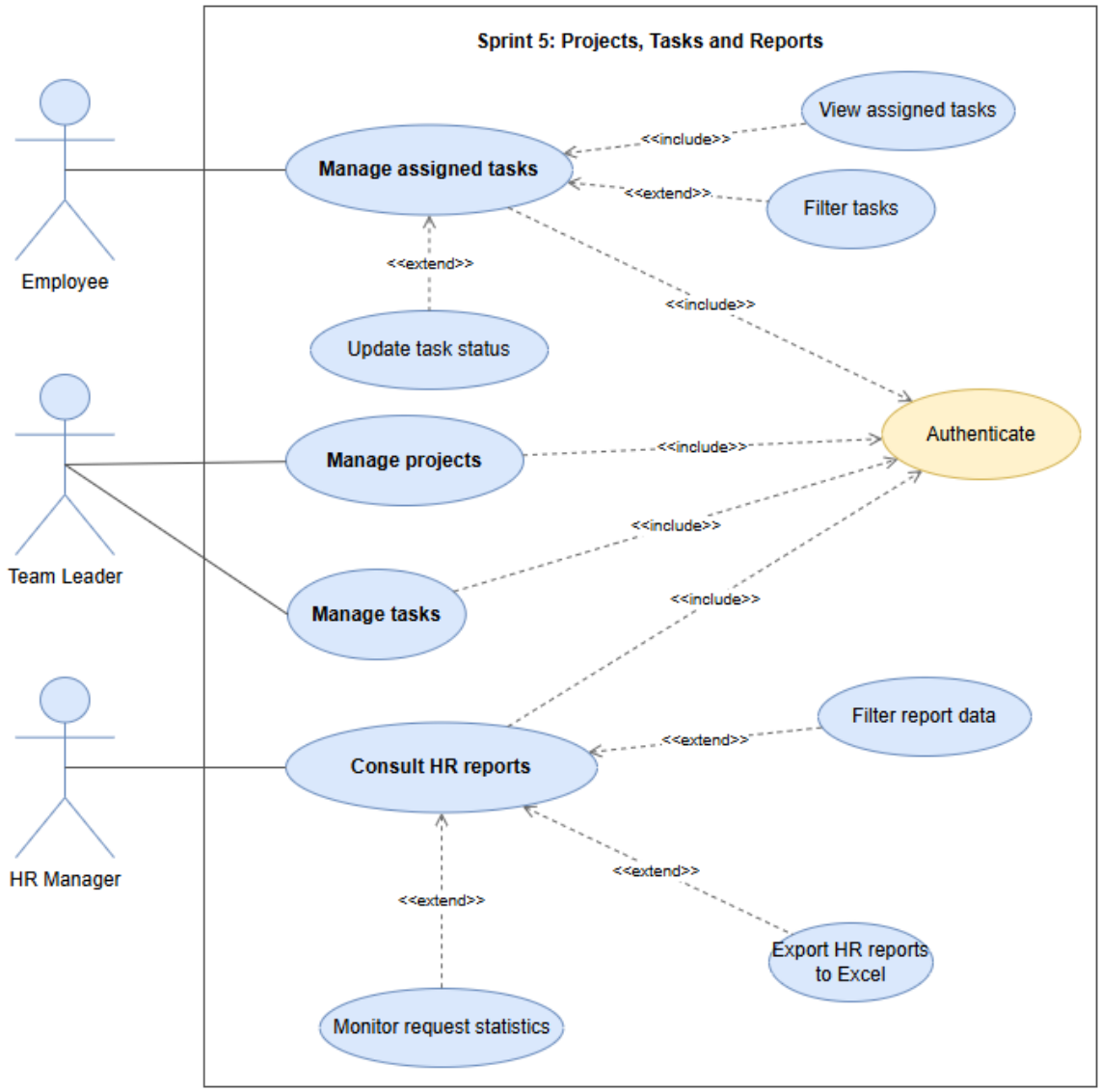


Figure 161: Use Case Diagram – Sprint 5 Projects, Tasks and Reports

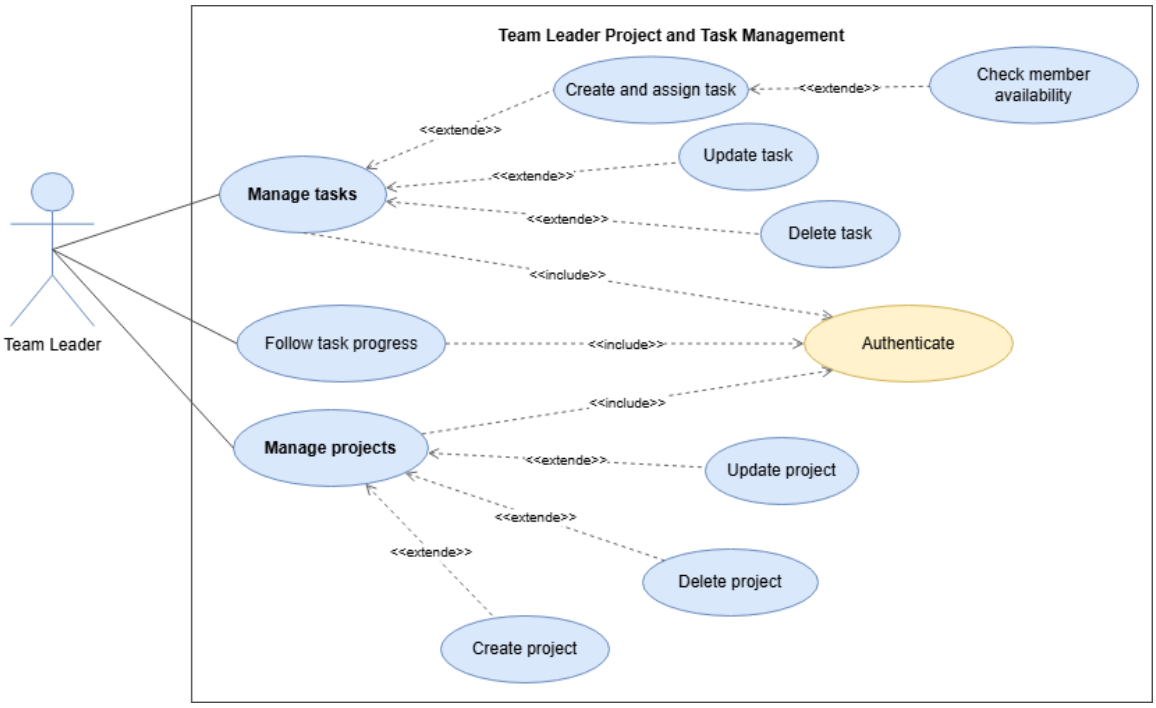


Figure 162: Use Case Diagram – Team Leader Project and Task Management

System Sequence Diagrams

These diagrams show how the user and system interact for the main Sprint 5 operations.

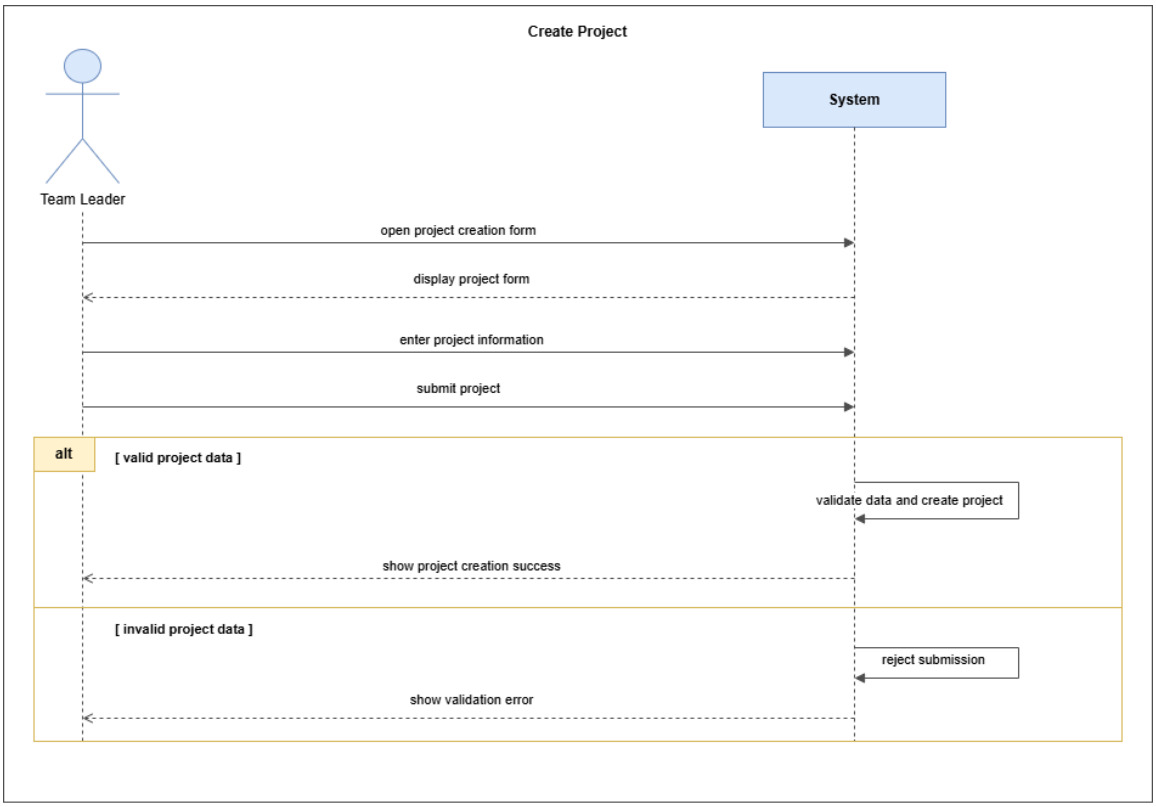


Figure 163: System Sequence Diagram – Create Project

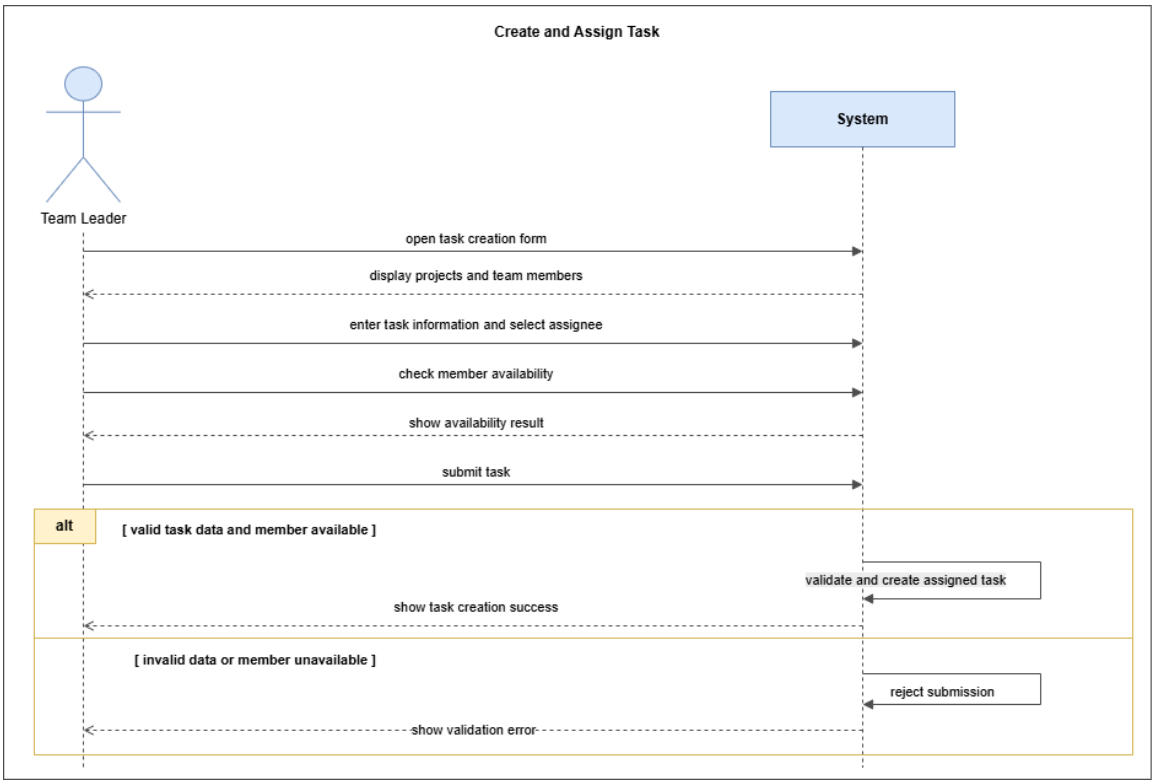


Figure 164: System Sequence Diagram – Create and Assign Task

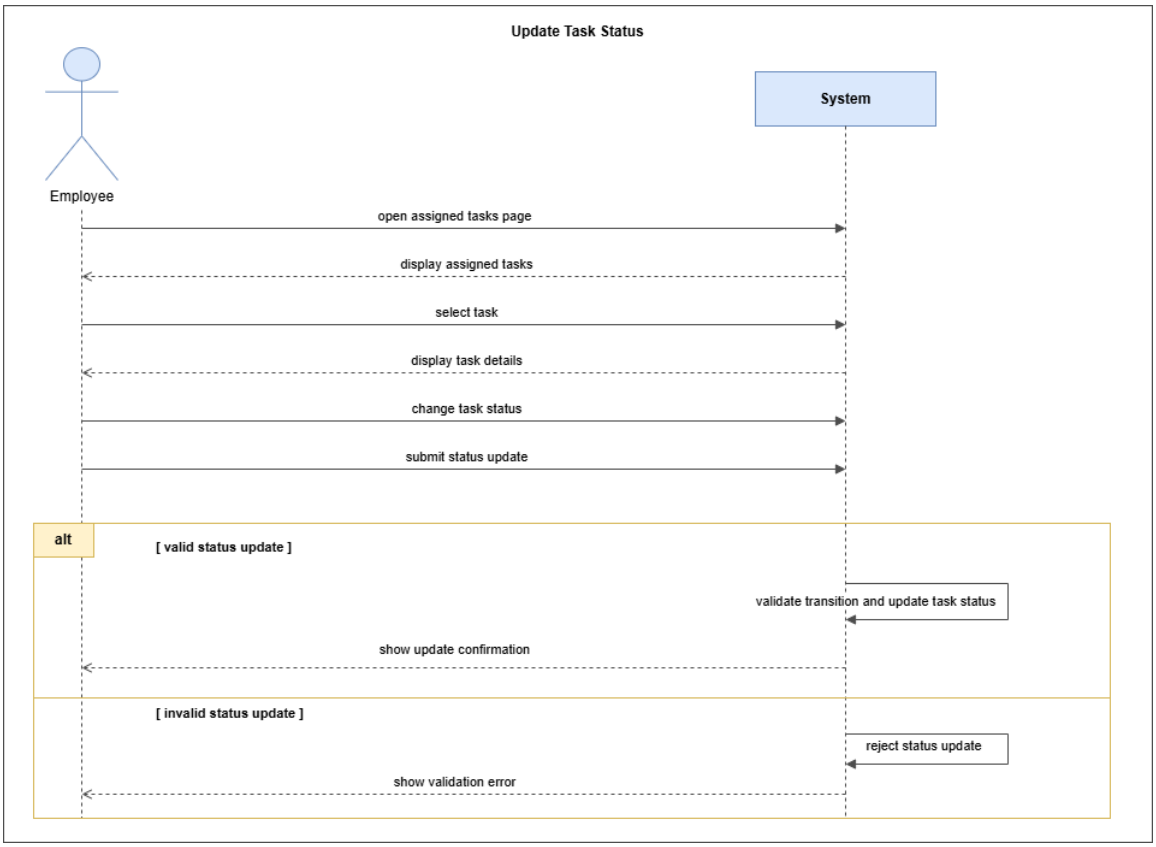


Figure 165: System Sequence Diagram – Update Task Status

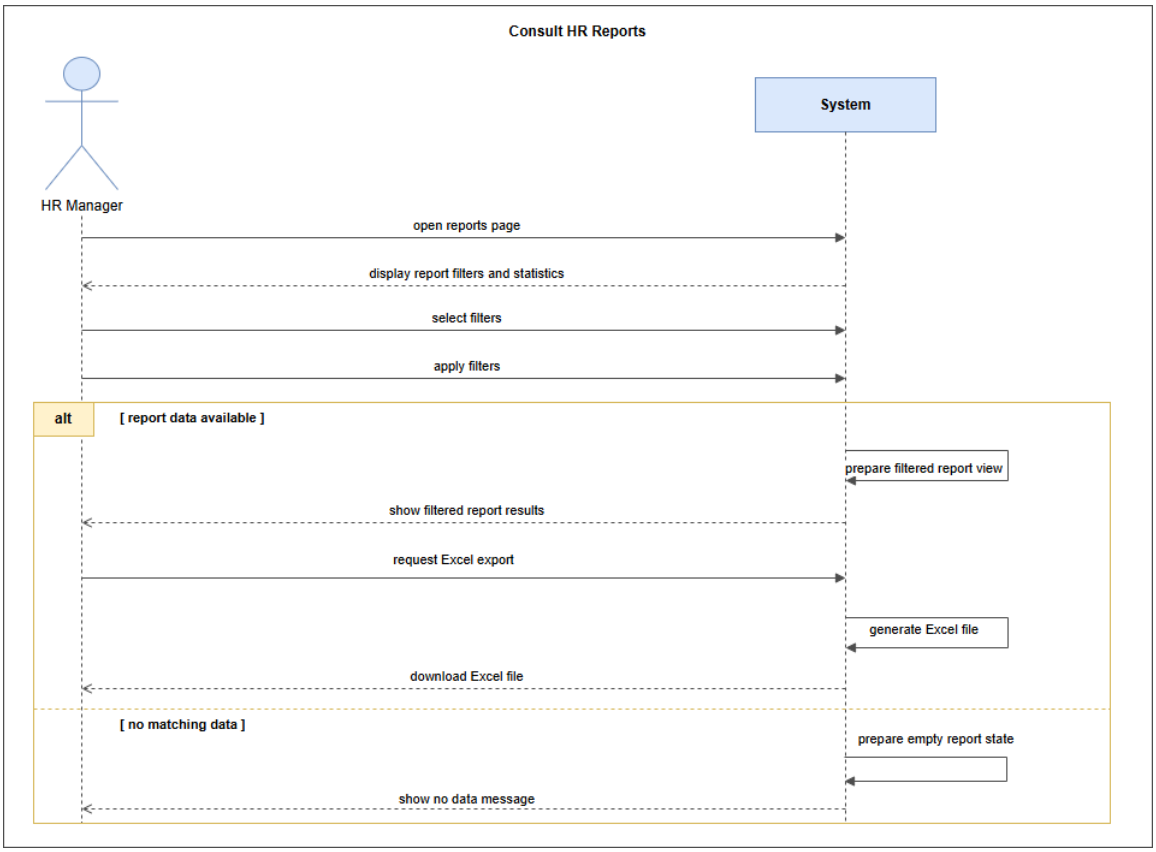


Figure 166: System Sequence Diagram – Consult HR Reports

Analysis

This diagram shows the main domain concepts behind Sprint 5's project management, task assignment, and reporting features.

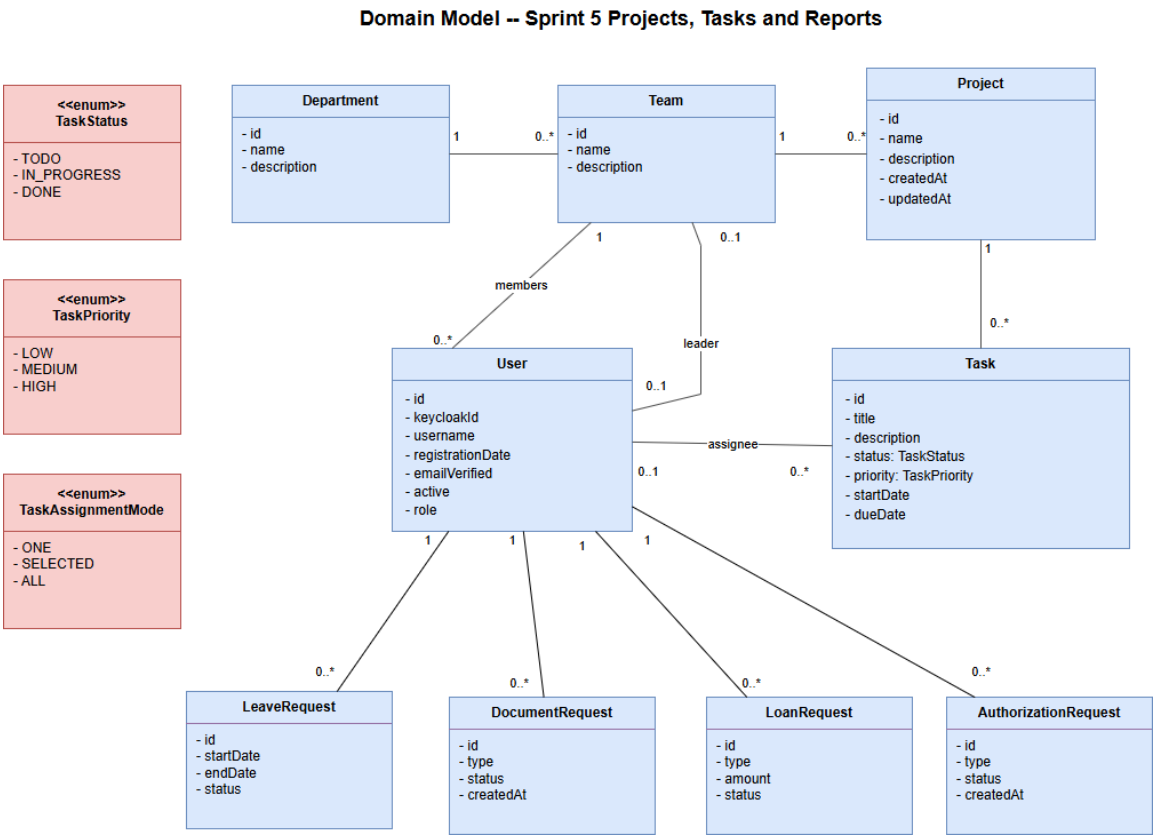


Figure 167: Domain Model – Sprint 5 Projects, Tasks and Reports

State Diagrams

The state diagram below shows the task lifecycle for Sprint 5. A task starts as TODO, moves to IN_PROGRESS when the assignee picks it up, and ends as DONE.

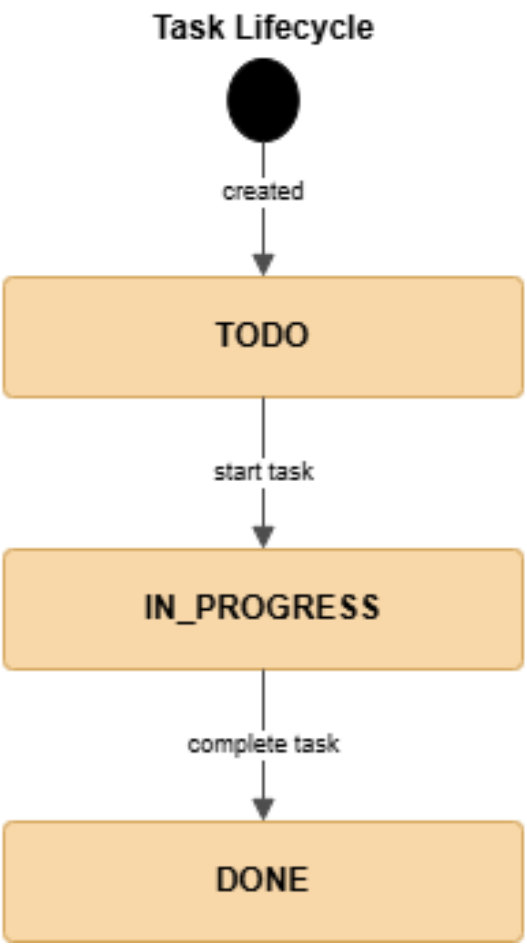


Figure 168: State Diagram – Task Lifecycle

Participant Class Diagrams

The participant class diagrams below show the frontend and backend classes involved in the main Sprint 5 workflows.

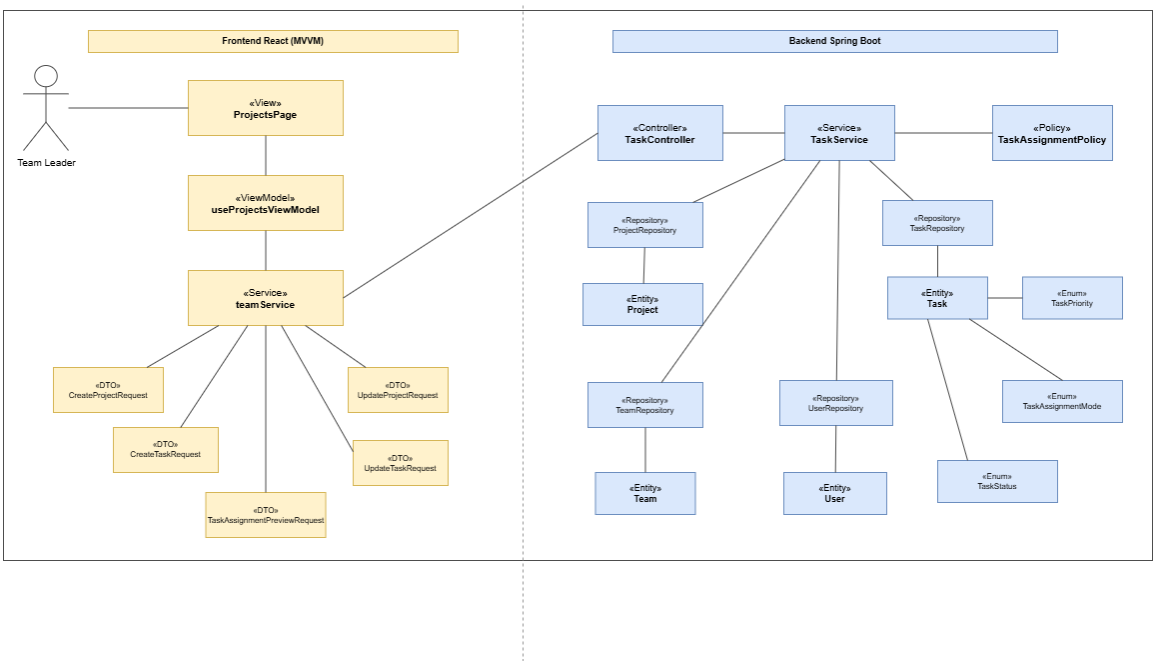


Figure 169: Participant Class Diagram – Team Leader Project and Task Management

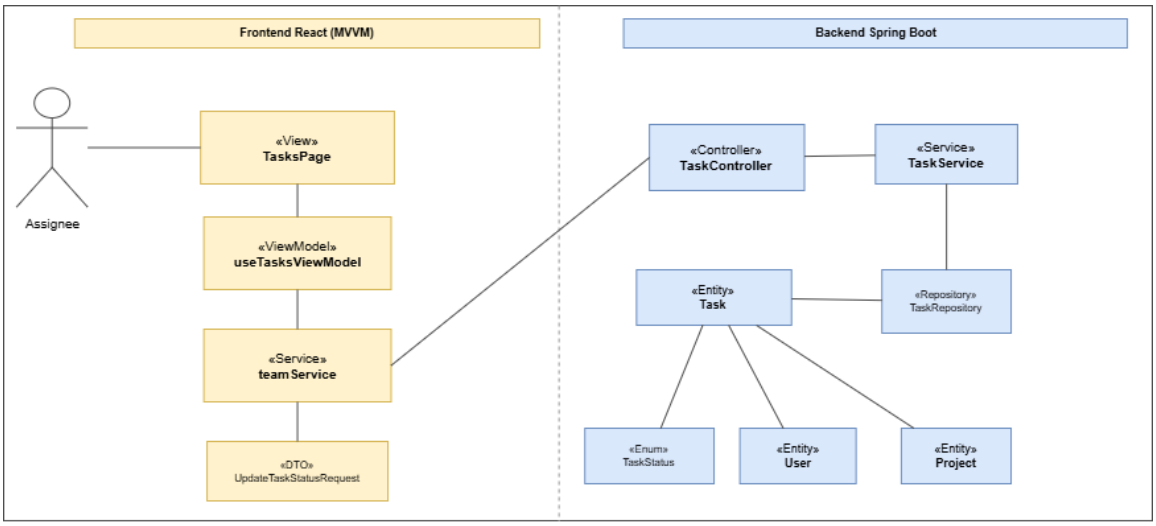


Figure 170: Participant Class Diagram – Task Status Update

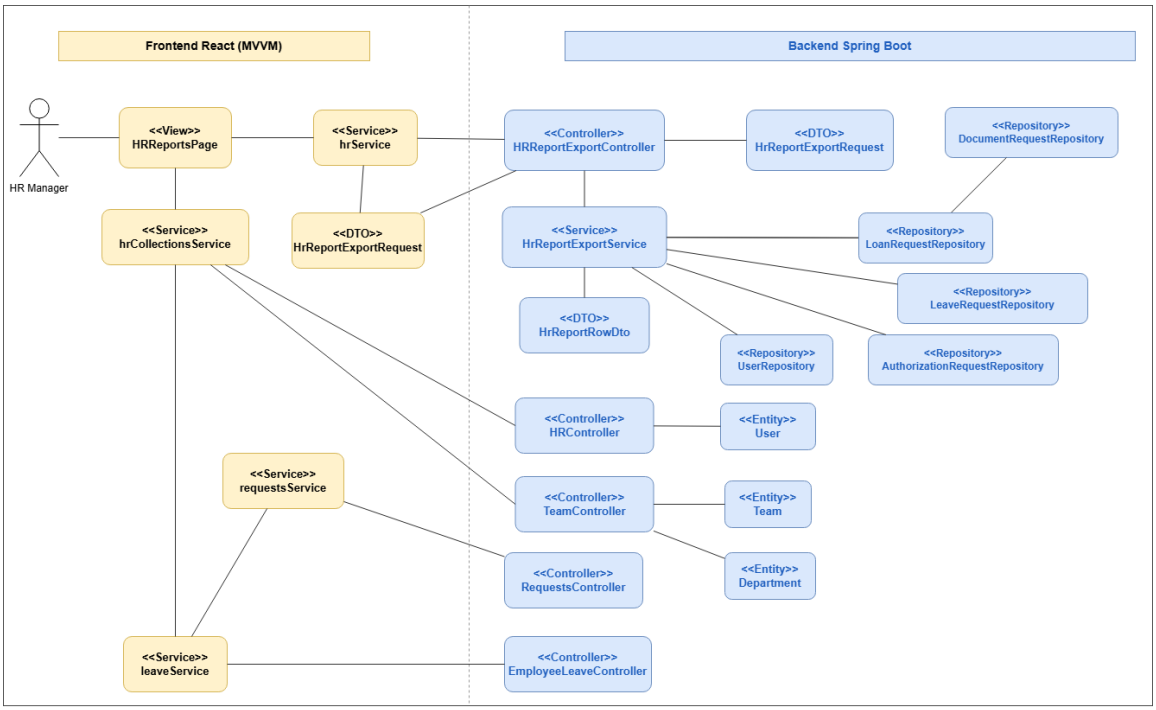


Figure 171: Participant Class Diagram – HR Reports and Export

Design Class Diagrams

The design class diagrams below show the class structure for Sprint 5 project management, task handling, and HR report export.

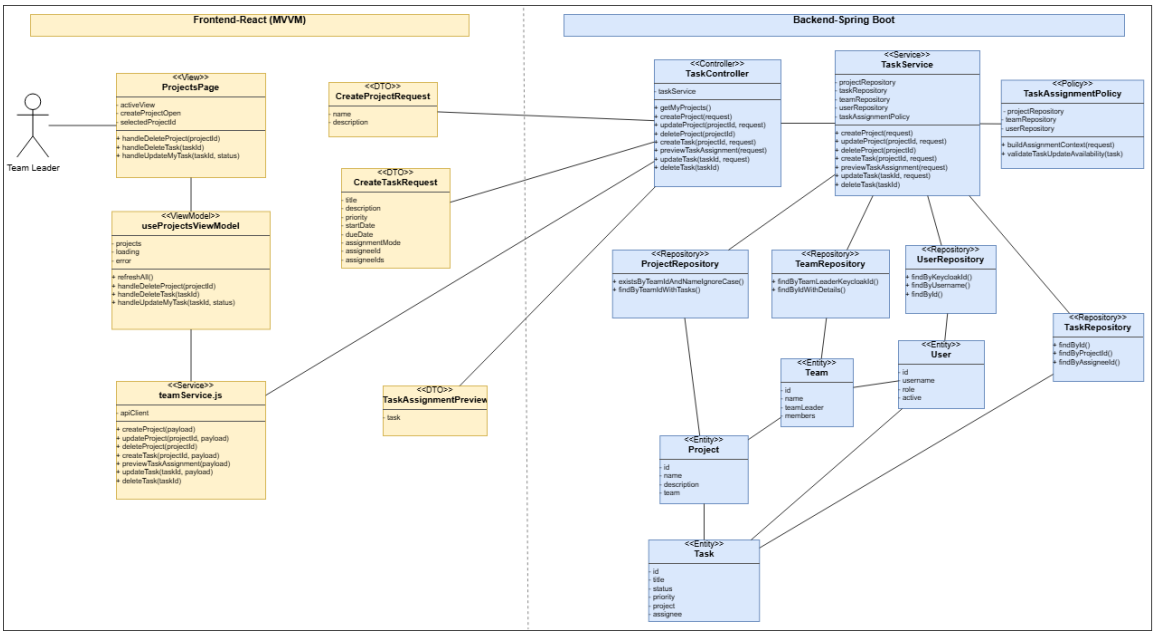


Figure 172: Design Class Diagram – Team Leader Project and Task Management

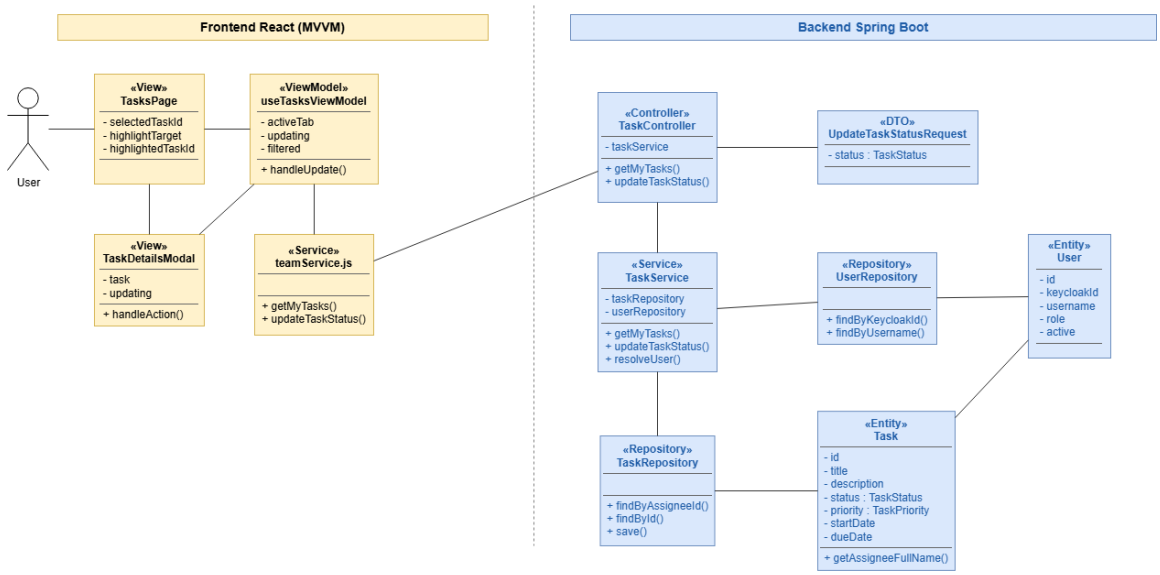


Figure 173: Design Class Diagram – Task Status Update

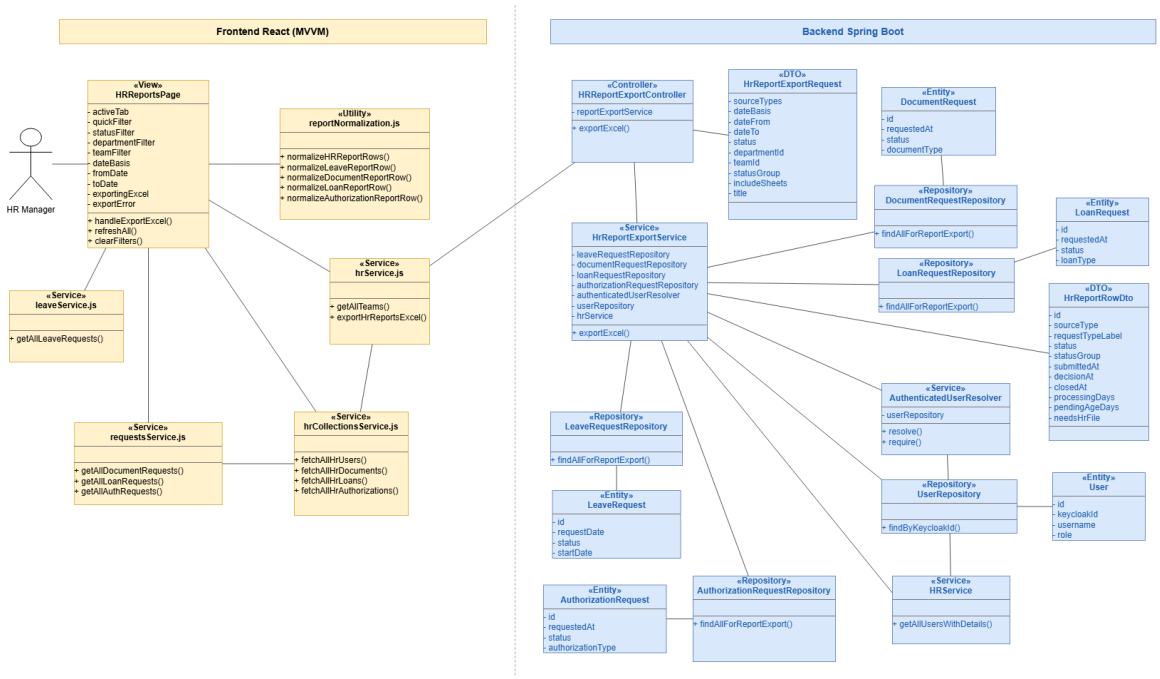


Figure 174: Design Class Diagram – HR Reports and Excel Export

Conception Sequence Diagrams

The conception sequence diagrams below show the frontend and backend interactions for project management, task assignment, and status update.

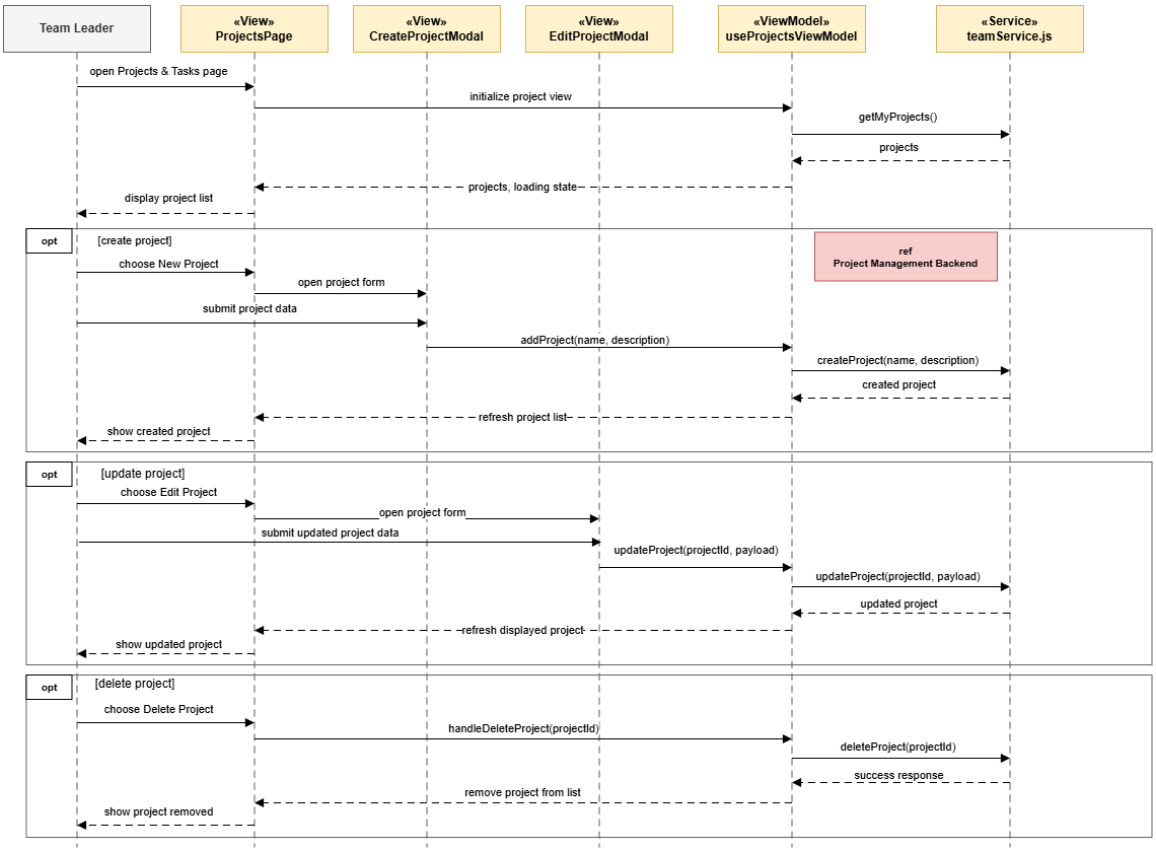


Figure 175: Conception Sequence Diagram – Project Management Frontend

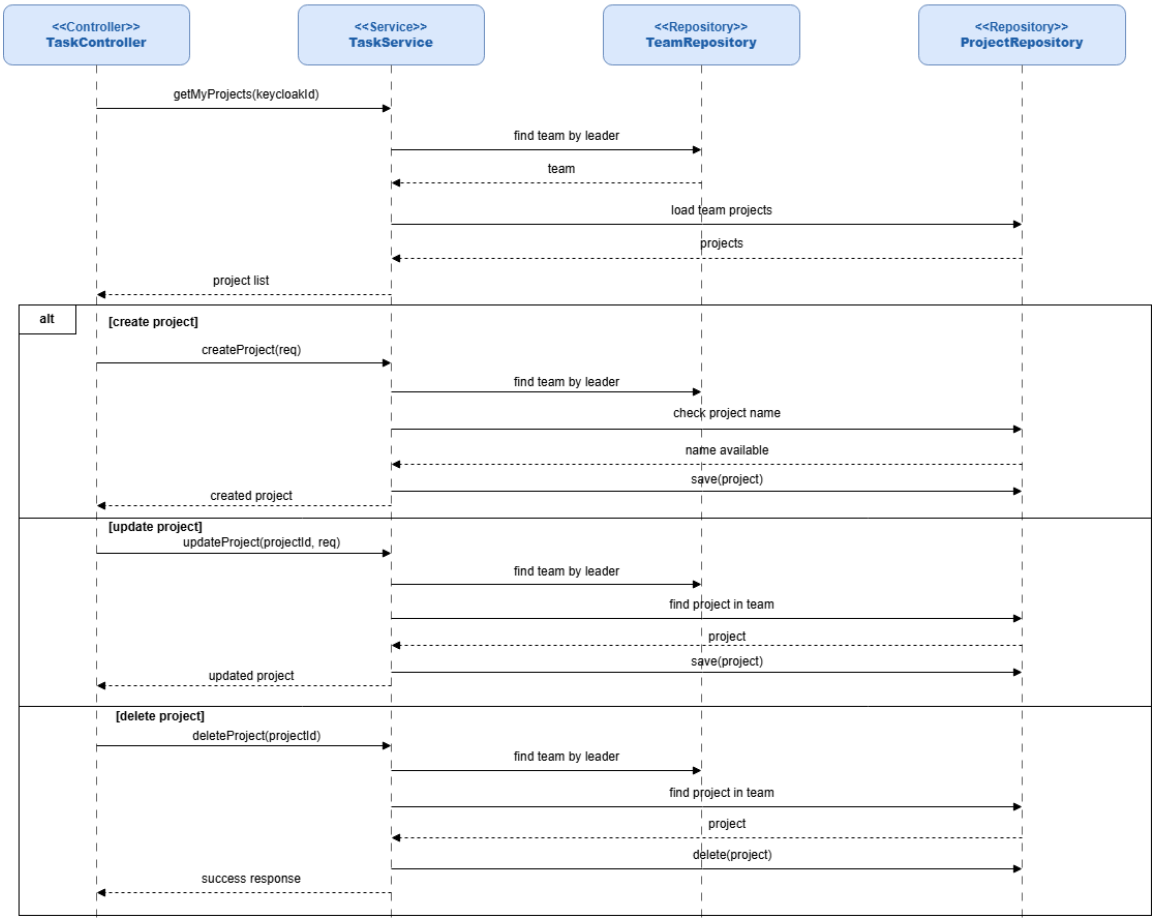


Figure 176: Conception Sequence Diagram – Project Management Backend

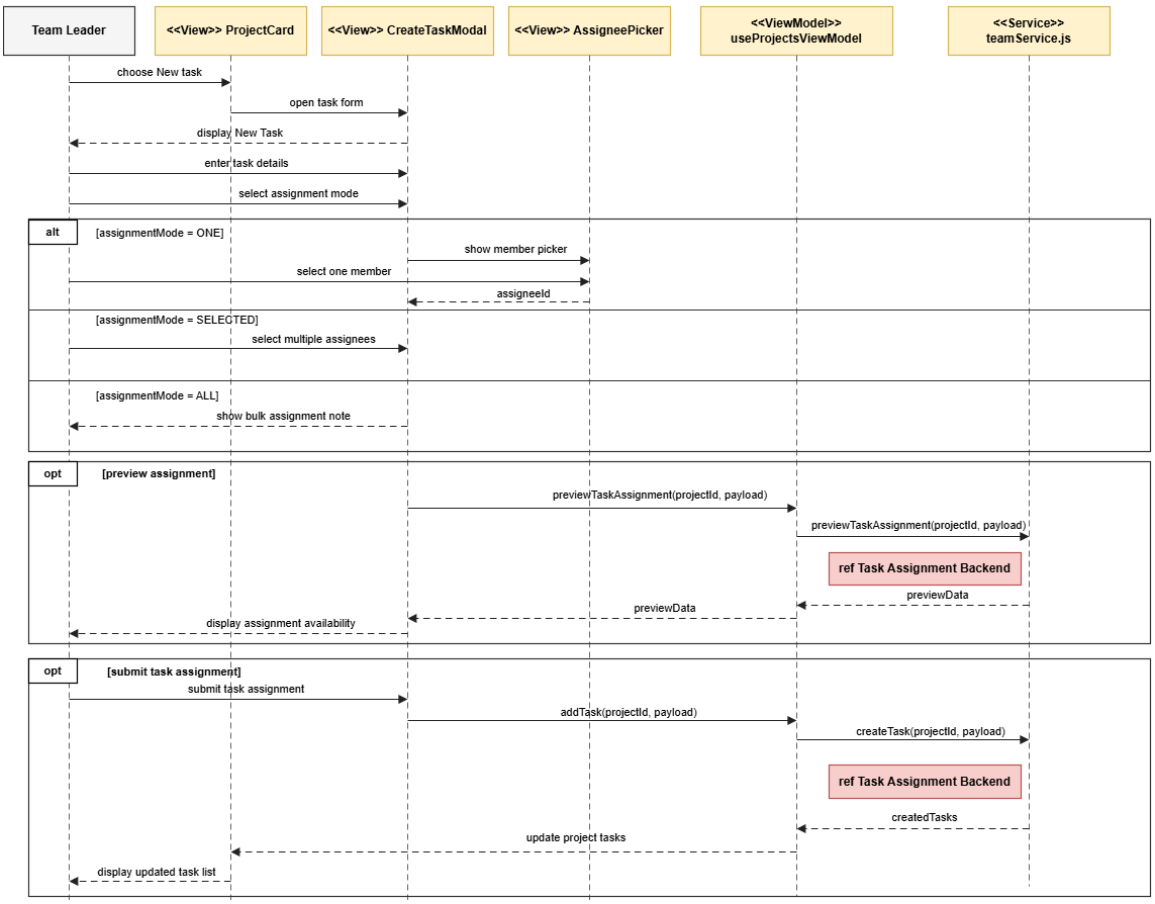


Figure 177: Conception Sequence Diagram – Task Assignment Frontend

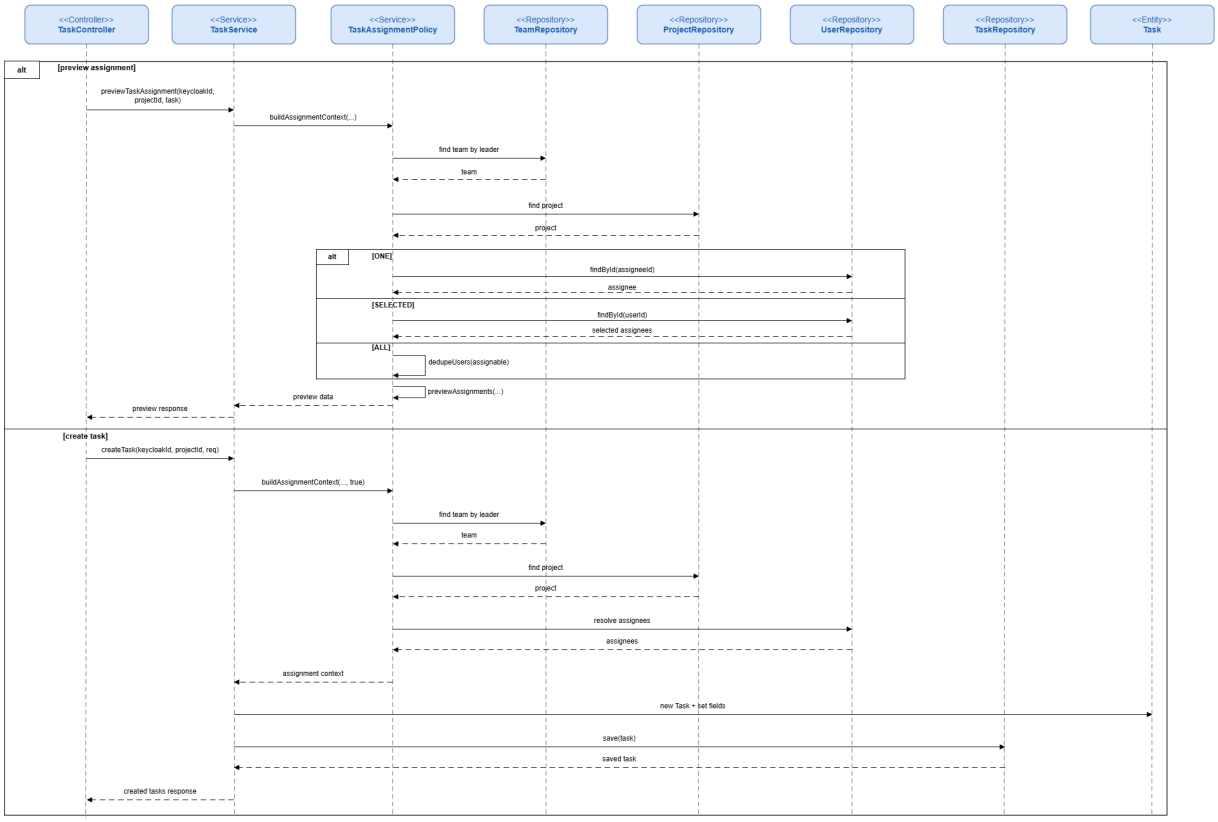


Figure 178: Conception Sequence Diagram – Task Assignment Backend

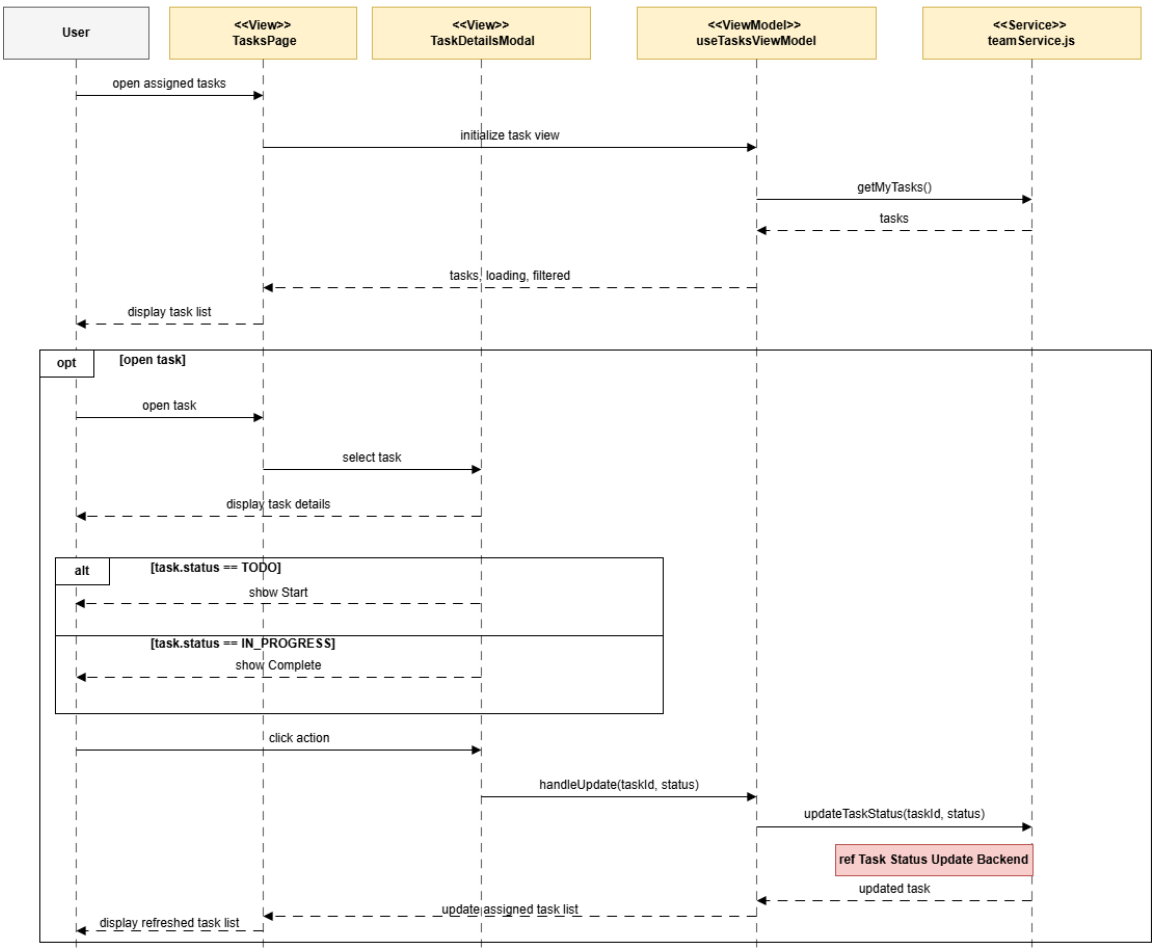


Figure 179: Conception Sequence Diagram – Task Status Update Frontend

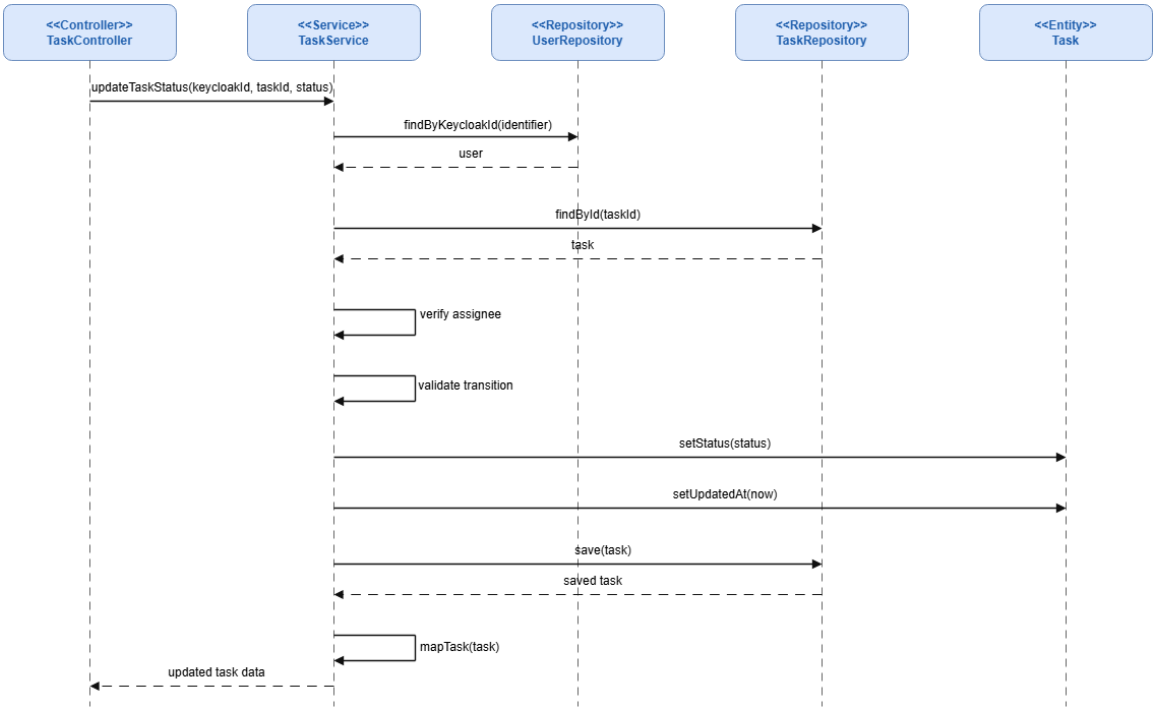


Figure 180: Conception Sequence Diagram – Task Status Update Backend

Interface Screenshots

The following screenshots document the Sprint 5 user interfaces for projects, tasks, reporting, and Excel export.

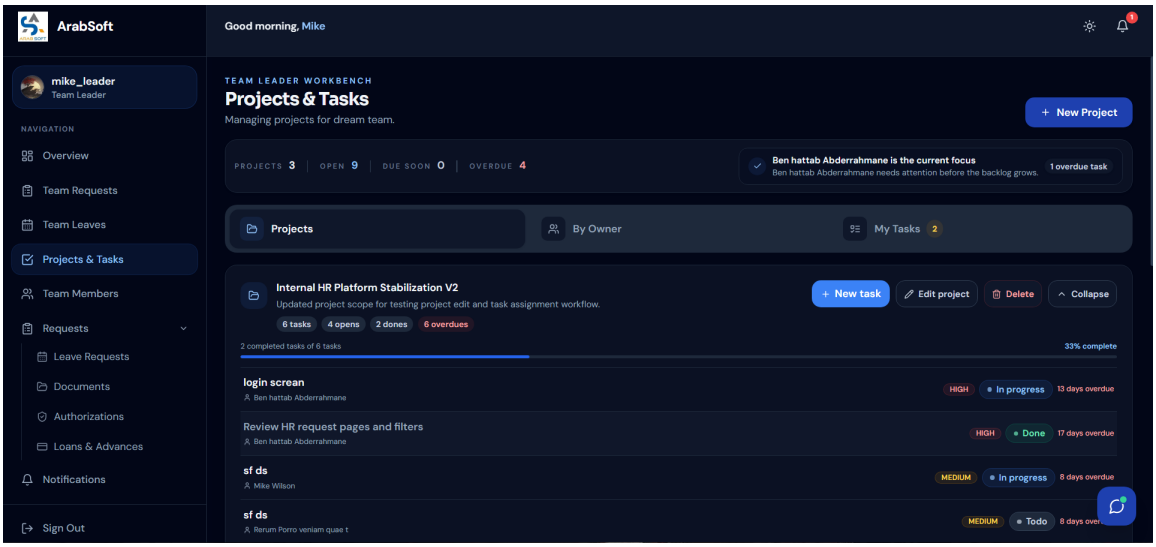


Figure 181: Team Projects and Tasks Interface

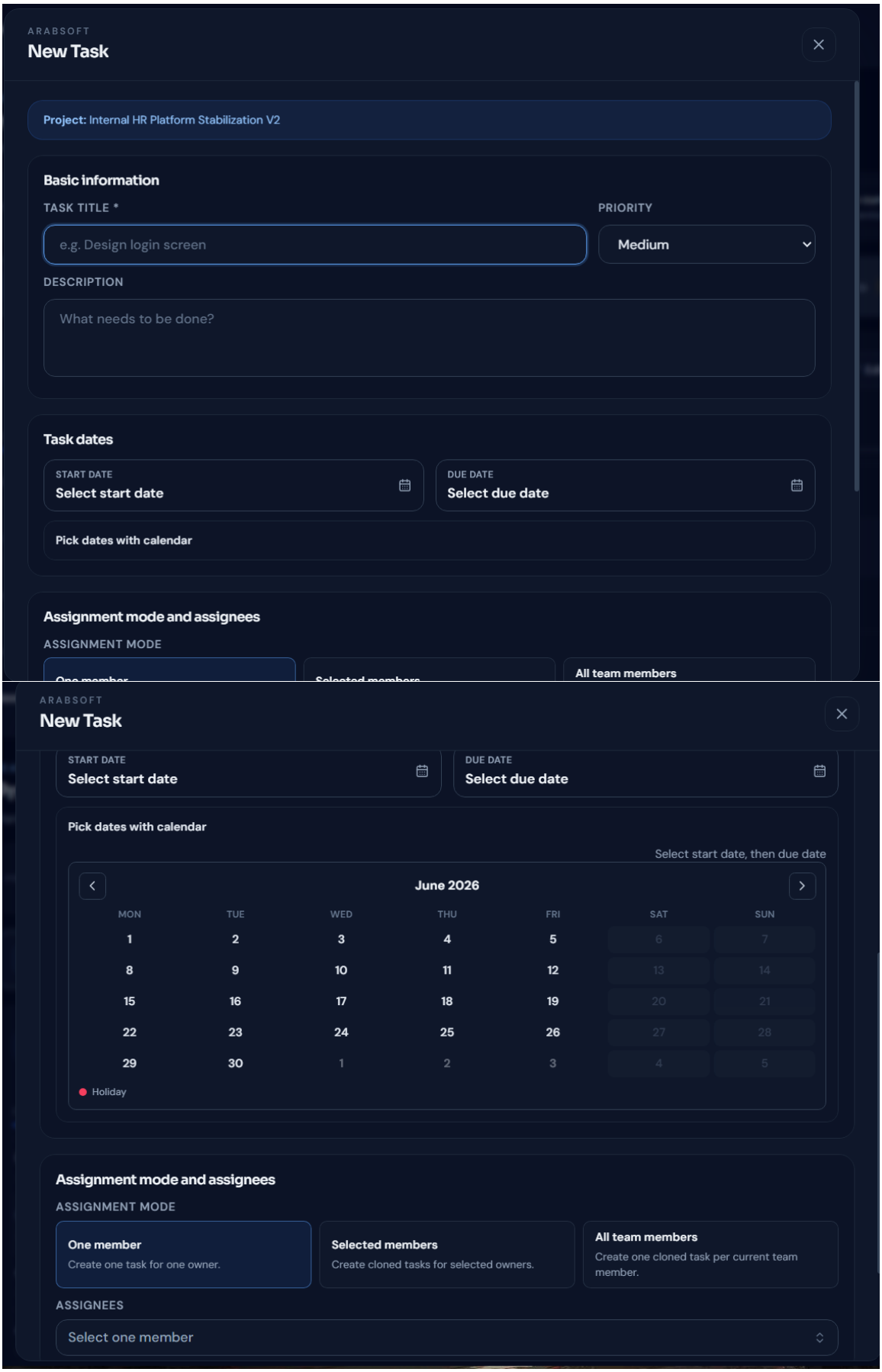


Figure 182: Task Assignment Interface

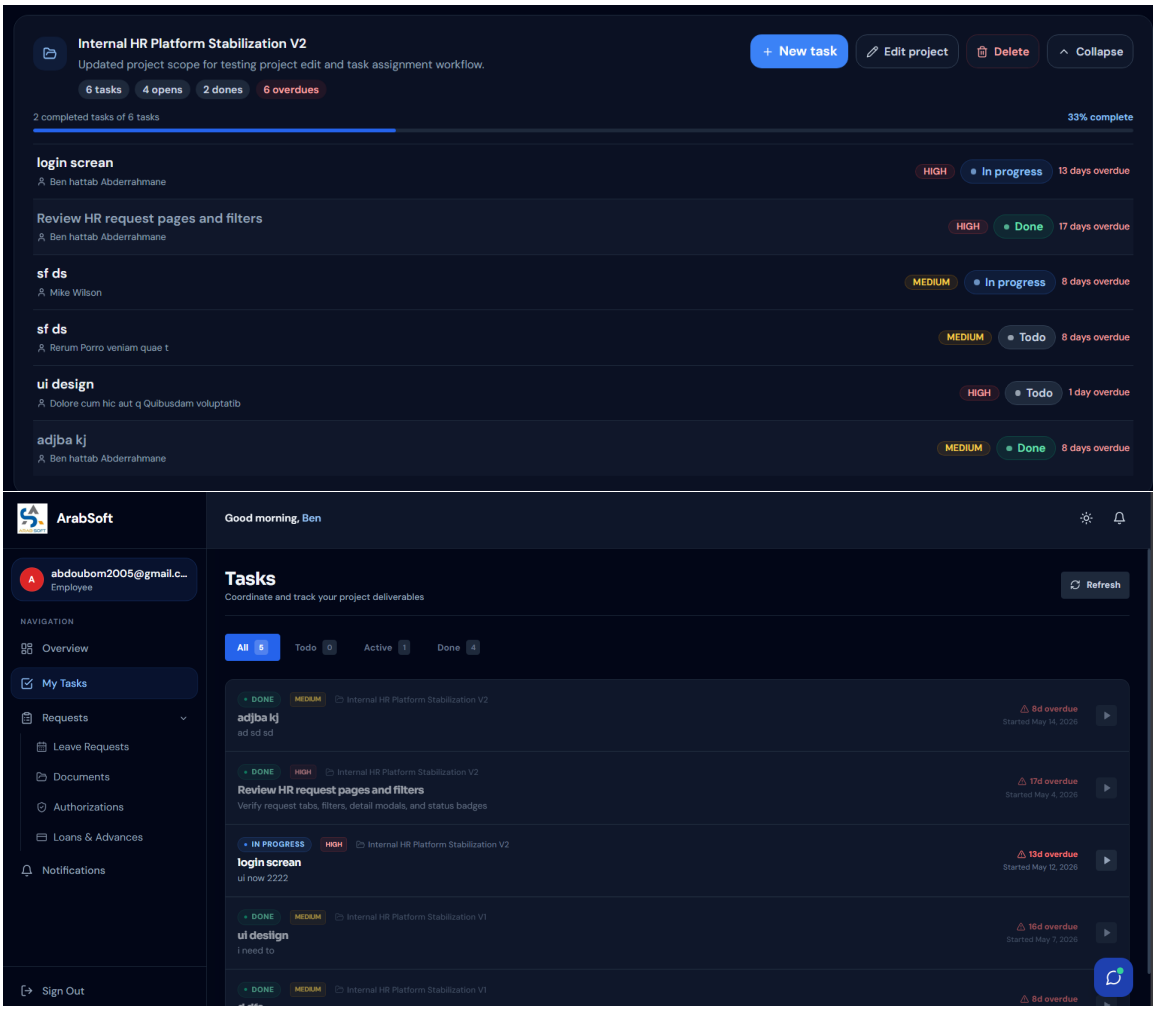


Figure 183: Employee Task Tracking Interface

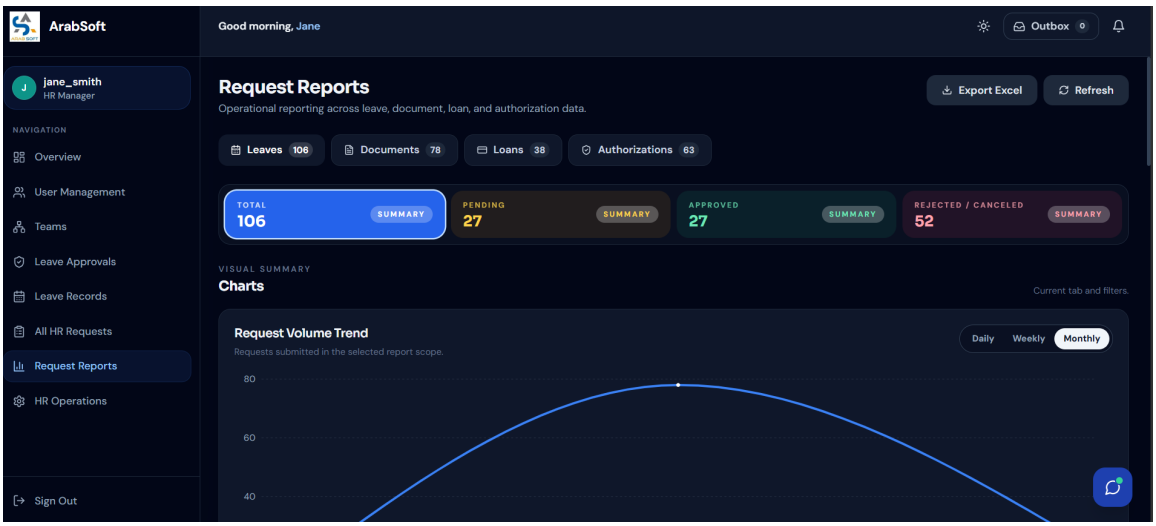


Figure 184: HR Request Reports Interface

ArabSoft HR Reports - Leaves

Detailed employee/request rows are available in the Leaves/Documents/Loans/Authorizations sheets.

Generated at	28/05/2026 00:02

Report scope

Label	Value
Source	Leaves

Applied filters

Label	Value
Date basis	Submitted date
Period	All dates
Status	All statuses
Department	All departments
Team	All teams

Key metrics

Metric	Value
Total requests	106
Pending	27
Approved	27
Rejected	25
Cancelled	27
Stale pending	24
Needs HR file	0
Average processing time	7.0 days
Average pending age	35.1 days

Source type breakdown

Source type	Count
Leaves	106
Documents	0
Loans	0
Authorizations	0

Department workload top 10

Department	Total	Pending	Approved	Rejected/Cancelled
Engineering / IT	40	6	6	28
No department	11	0	8	3
Operations	8	3	2	3
Finance	7	3	2	2
Support	7	2	1	4
Engineering	7	3	1	3
QA	7	2	2	3
Product	7	3	1	3
HR	7	3	1	3
Customer Support	5	2	3	0

Pending aging

Bucket	Count
0-2 days	1
3-7 days	0
8-14 days	2
15+ days	24

Attention items / stale pending preview

Employee name	Request type	Department	Team	Pending age	Submitted date
Omar Ayari	Annual	Engineering	Seed 07 - Delivery	60 days	29/03/2026 16:34
Ines Cherif	Other	Operations	Seed 06 - Care	59 days	30/03/2026 16:34
Nour Mejri	Sick	Product	Seed 02 - Experience	55 days	03/04/2026 16:34
Karim Jaziri	Annual	QA	Seed 01 - Platform	54 days	04/04/2026 16:34
Adam Dridi	Unpaid	Finance	Seed 07 - Delivery	50 days	08/04/2026 16:34
Salma Amri	Sick	HR	Seed 06 - Care	49 days	09/04/2026 16:34
Rania Zouari	Maternity	Operations	Seed 02 - Experience	45 days	13/04/2026 16:34
Malek Ferchichi	Unpaid	Support	Seed 01 - Platform	44 days	14/04/2026 16:34
Omar Ayari	Paternity	QA	Seed 07 - Delivery	40 days	18/04/2026 16:34
Ines Cherif	Maternity	Engineering	Seed 06 - Care	39 days	19/04/2026 16:34

Figure 185: Excel Report Export Result

5. Sprint Review

Sprint 5 delivered team-scoped project management, task assignment with availability checks, task status tracking by assignees, and HR report consultation with Excel export. Team leaders can manage projects and tasks in one place, and HR managers can filter and export administrative request data.

6. Sprint Retrospective

What went well	What needs improvement
Having project and task management scoped to the team made the workflows easier to follow and validate.	**Task assignment needed careful validation to keep availability checks, team membership, and assignment modes consistent.**
HR managers could filter administrative request data and export it to Excel directly.	**Future changes to the report section should keep the export limited to administrative request data and not include project or task data unless a dedicated backend source exists.**

Table 10: Sprint Retrospective – Sprint 5

III. Sprint 6: AI Assistant, Local Deployment, and Stabilization

1. Sprint Goal

This sprint focuses on the ArabSoft AI assistant, local deployment packaging, and final stabilization. The assistant is guidance-only, Spring Boot remains the secure gateway, and FastAPI receives only safe context prepared by the backend. [18] Gemini is optional and the deployment scope remains limited to local Docker Compose.

2. Visual Sprint Journal

The Sprint 6 visual journal reflects the validated assistant and deployment work completed during the sprint.

3. Sprint Backlog

The table below summarizes the Sprint 6 work packages prepared from the validated implementation scope.

Sprint	Stories	Tasks	Duration
Sprint 6	**AI assistant platform help and Q&A**	Route assistant questions through Spring Boot, use role-aware platform guidance, and return safe navigation/help responses.	2d
	AI request drafting assistant	Support leave, document, authorization, and loan drafts with preview, validation, and explicit submission.	3d
	Team Leader decision support	Provide read-only decision support for one selected leave request using safe team and workload context.	2d
	Assistant safety and role restrictions	Enforce refusal rules, sanitize routes, protect against stale confirmations, and keep the assistant guidance-only.	2d
	Assistant UI rendering	Render assistant replies, draft previews, and decision-support cards in the frontend.	2d
	Local Docker deployment	Containerize the frontend, backend, AI service, PostgreSQL, Keycloak, Kafka, Zookeeper, and PgAdmin for local deployment and testing.	2d
	Stabilization and regression coverage	Keep fallback paths, safety checks, and assistant regression tests aligned with the implemented behavior.	1d

Sprint Backlog – Sprint 6

4. Requirements Analysis

Sprint 6 adds an assistant scope centered on guidance, request drafting, and decision support. The assistant helps users ask platform questions, prepare drafts, review validation feedback, and access suggested pages. Final workflow decisions remain with the user and the authorized role.

Assistant Scope and Business Rules

Figure 186 summarizes the user-facing assistant scope for Sprint 6. It covers platform questions, request drafting, validation preview, suggested pages, and Team Leader decision support. The assistant remains guidance-only and does not take workflow decisions.

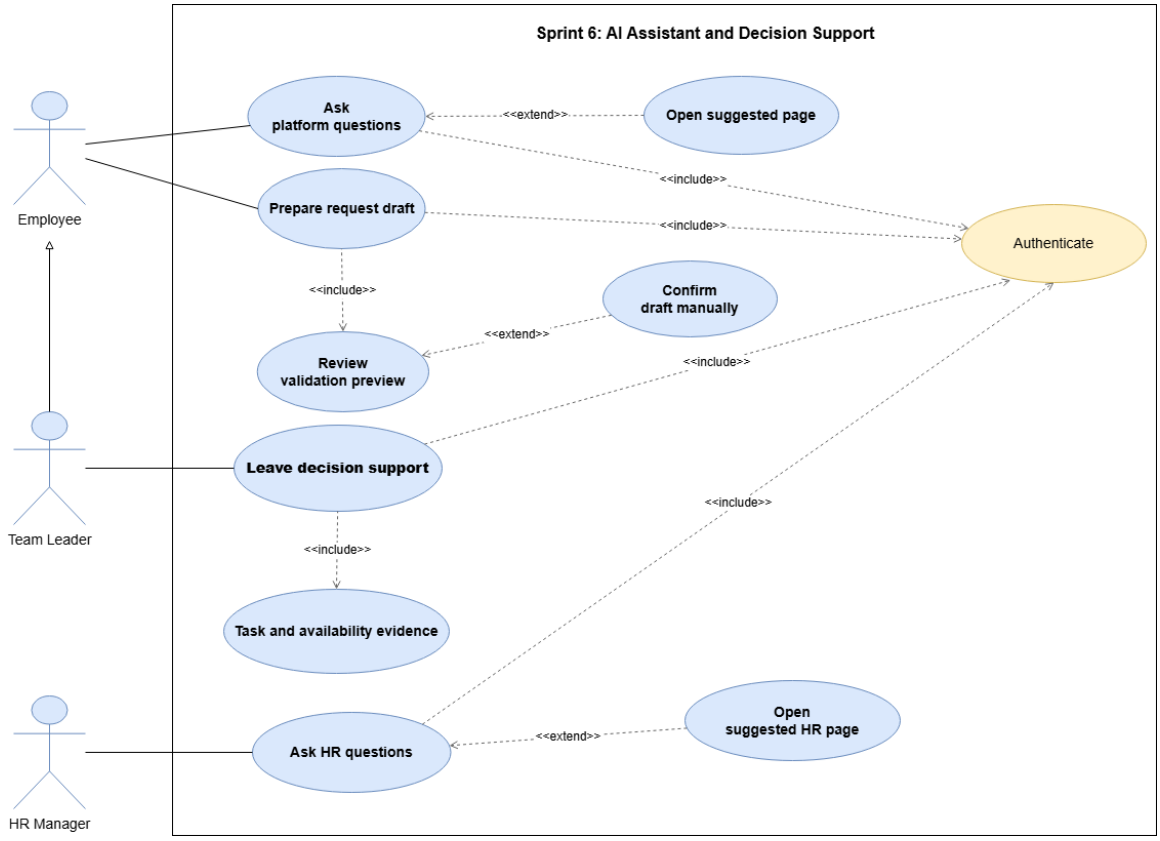


Figure 186: Use Case Diagram – Sprint 6 AI Assistant and Decision Support

The assistant remains focused on safe guidance and manual control. It does not approve, reject, submit, or change roles on behalf of the user.

System Sequence Diagrams

The Sprint 6 system sequence diagrams describe the user-system interaction for assistant access, request drafting, and Team Leader decision support. The assistant remains guidance-only and does not take workflow decisions.

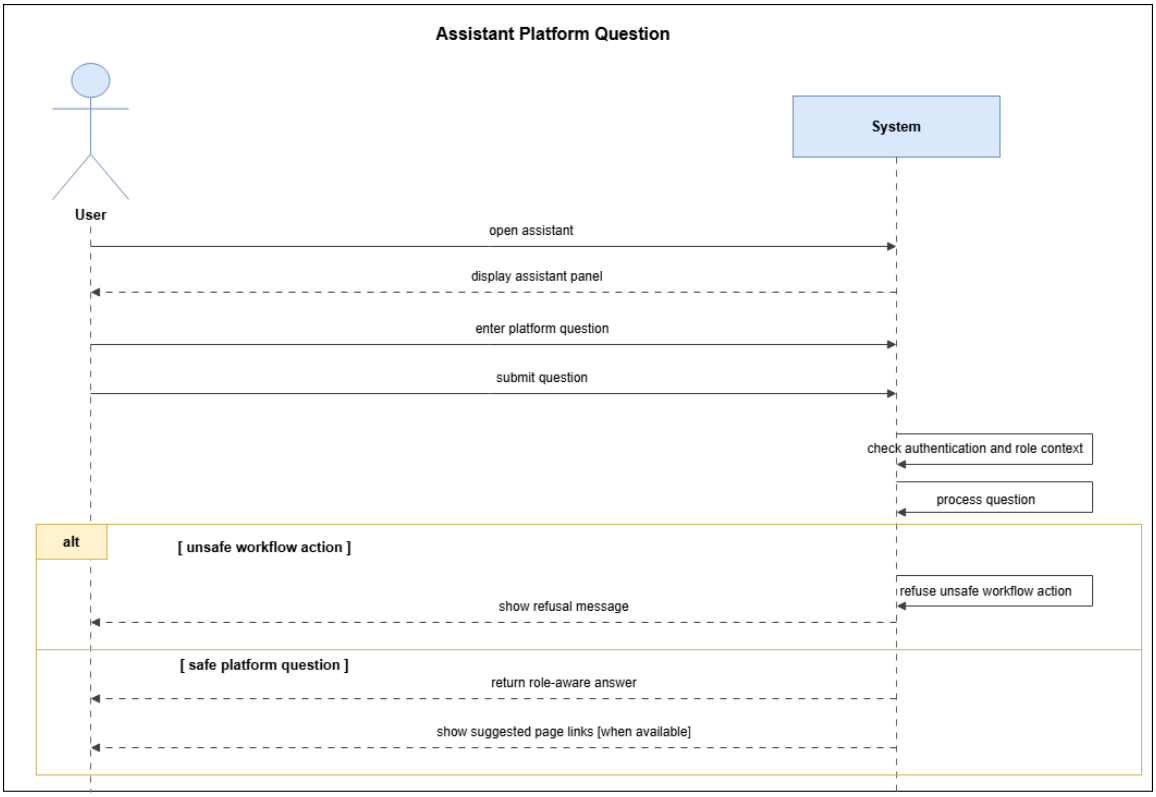


Figure 187: System Sequence Diagram – Assistant Platform Question

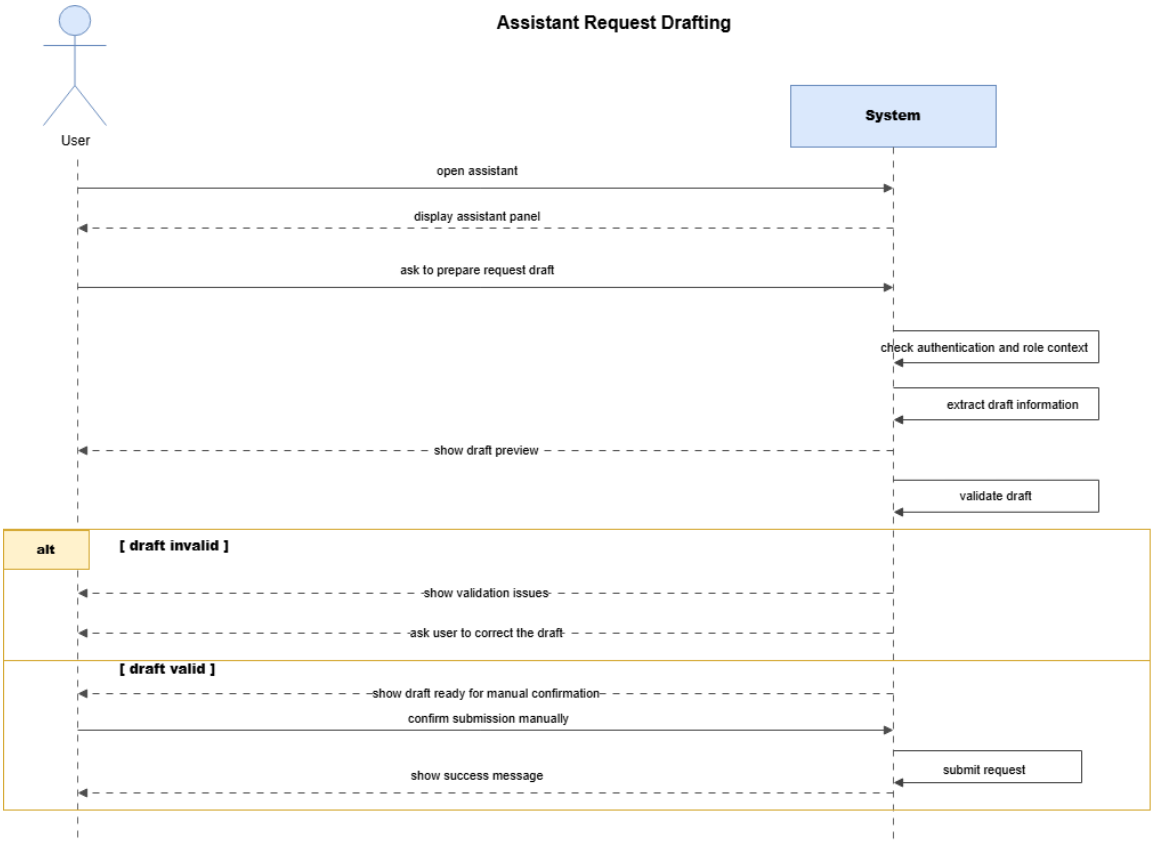


Figure 188: System Sequence Diagram – Assistant Request Drafting

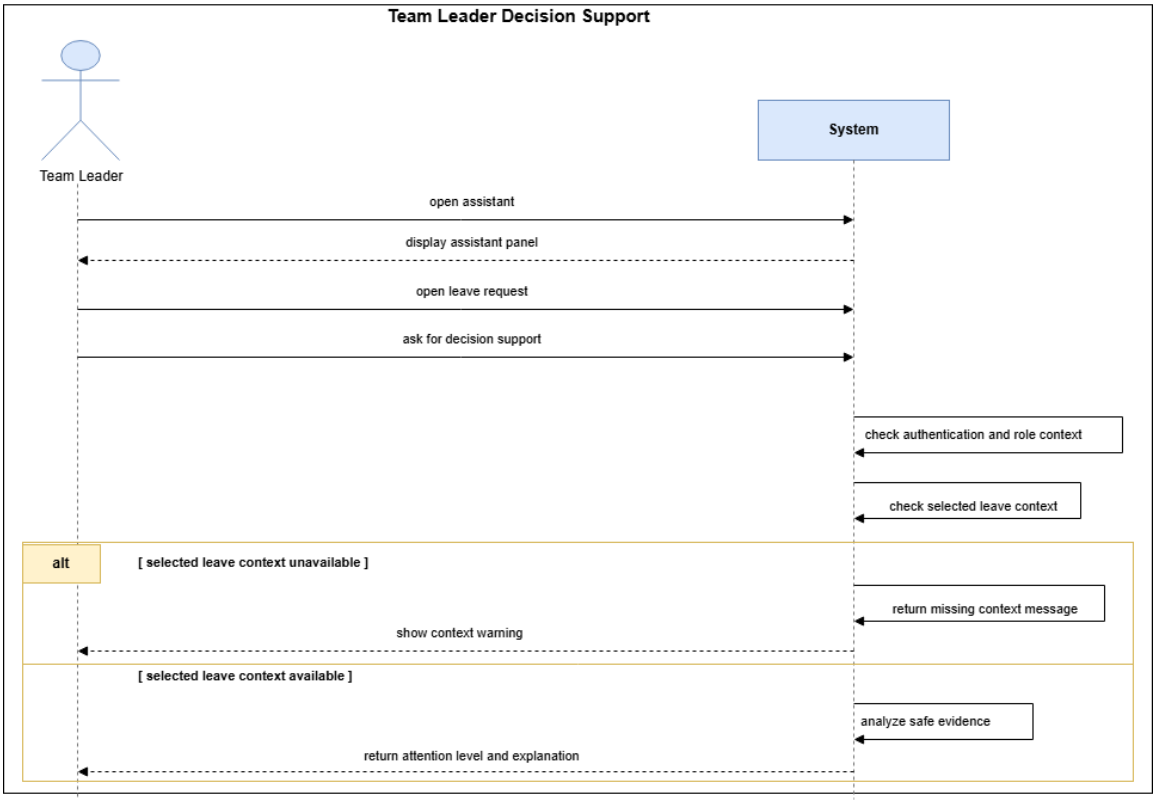


Figure 189: System Sequence Diagram – Team Leader Decision Support

Analysis and Design Decisions

Sprint 6 is documented with three complementary views: the domain model for the assistant context data, the participant class view for the gateway interaction, and the design class view for the safe assistant structures. Together, these diagrams separate persistent domain data from the assistant-specific object structure.

Domain Model

The domain model below shows the real Spring Boot entities used to build the assistant context.

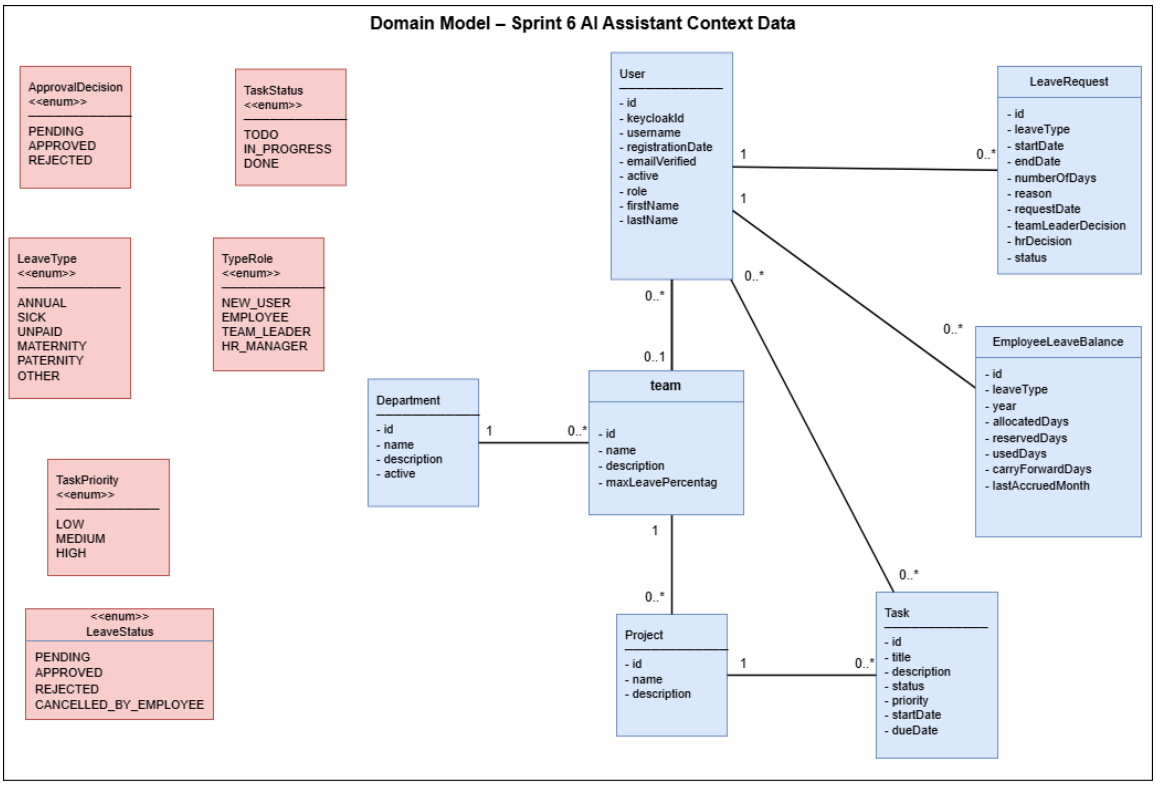


Figure 190: Domain Model – Sprint 6 AI Assistant Context Data

Participant Class Diagrams

The participant class diagram below captures the assistant gateway and the safe context contract used during the Sprint 6 assistant flow.

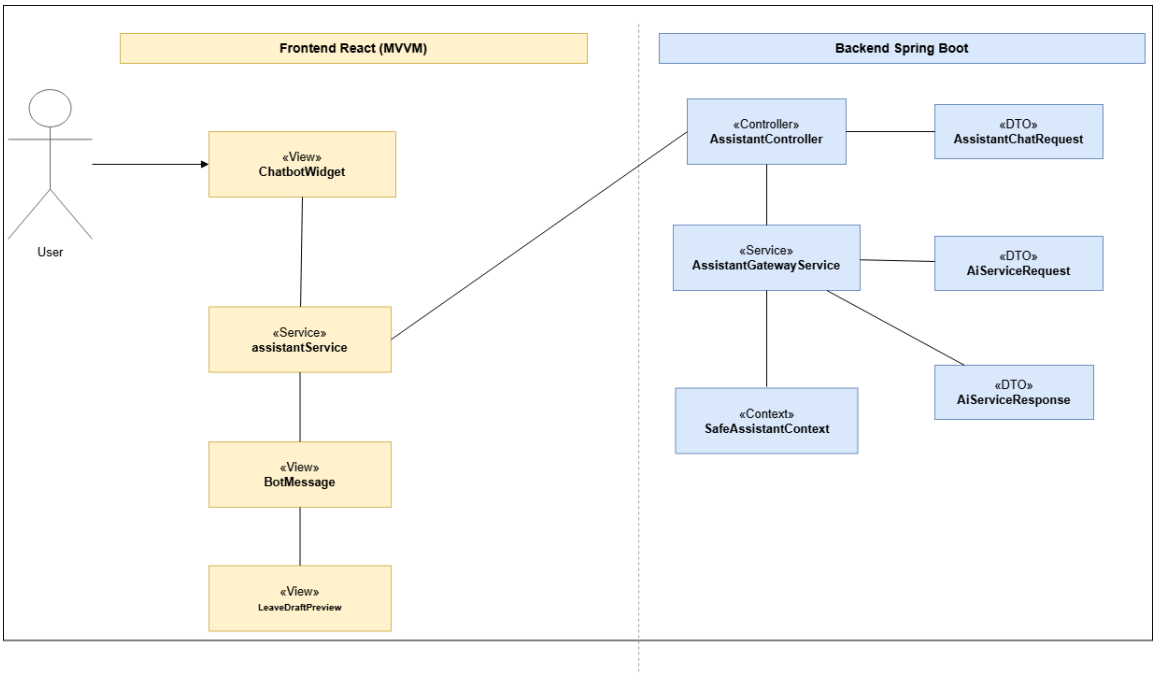


Figure 191: Participant Class Diagram – AI Assistant Gateway and Safe Context

Design Class Diagrams

The design class diagram below presents the assistant gateway and safe context classes used by the backend implementation.

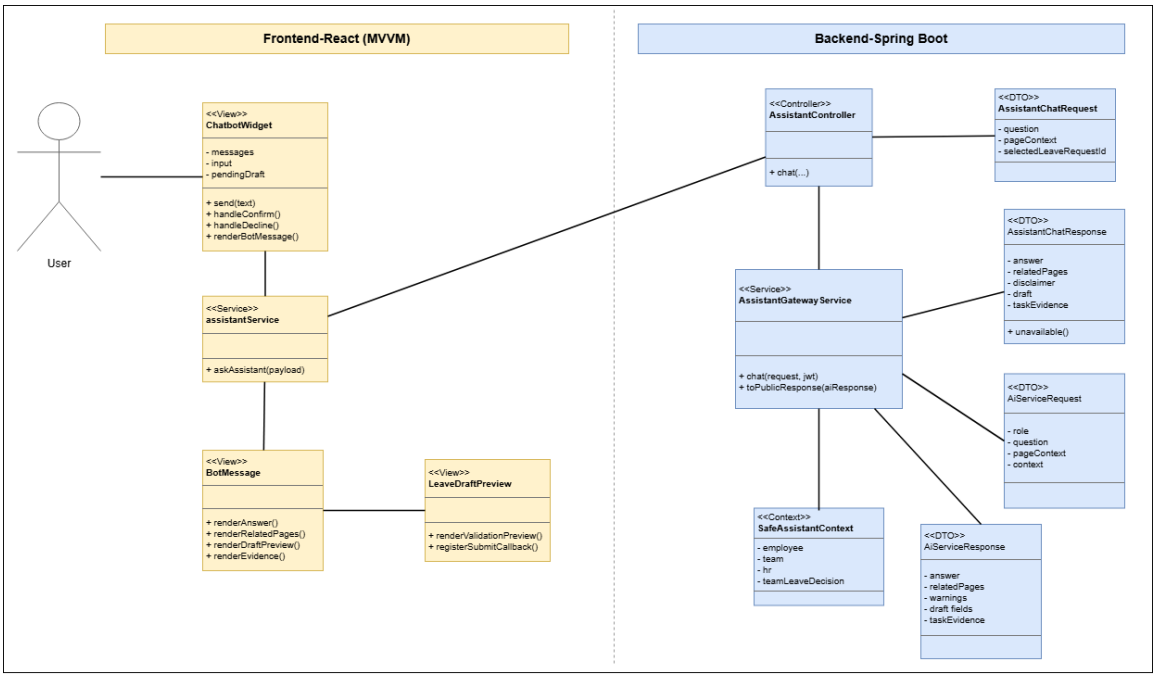


Figure 192: Design Class Diagram – Assistant Gateway and Safe Context

Conception Sequence Diagrams

The Sprint 6 conception sequence diagrams describe the frontend and backend realization of the assistant flows. The diagrams keep the assistant guidance-only: user questions, request drafting, and Team Leader decision support remain controlled by the authorized user.

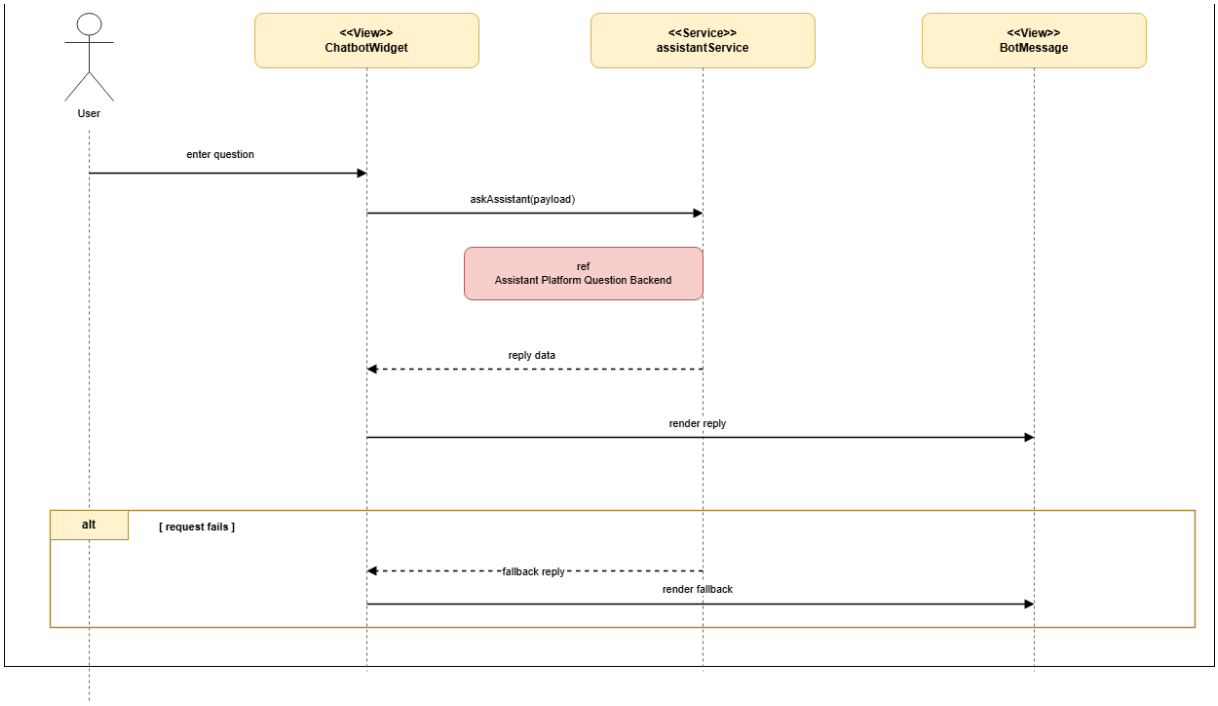


Figure 193: Conception Sequence Diagram – Assistant Platform Question Frontend

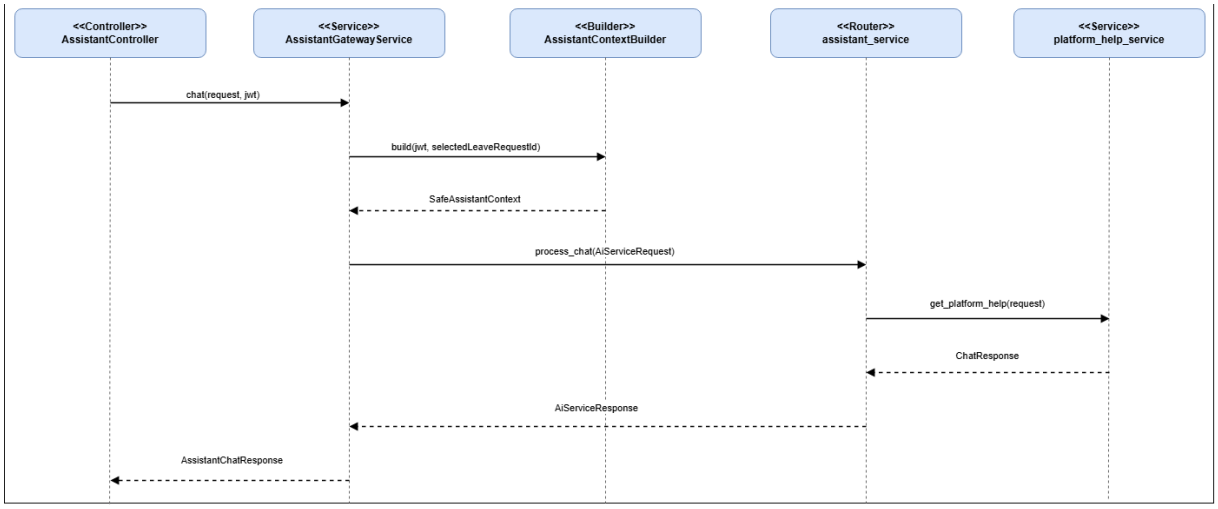


Figure 194: Conception Sequence Diagram – Assistant Platform Question Backend

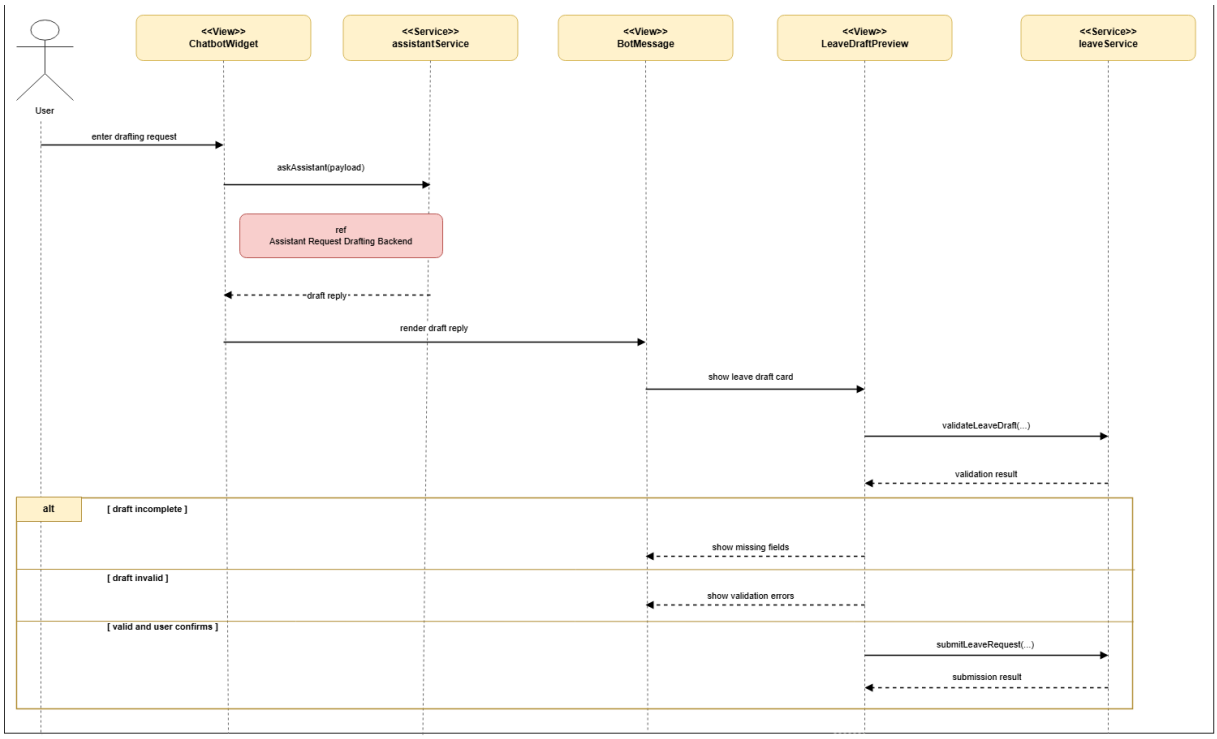


Figure 195: Conception Sequence Diagram – Assistant Request Drafting Frontend

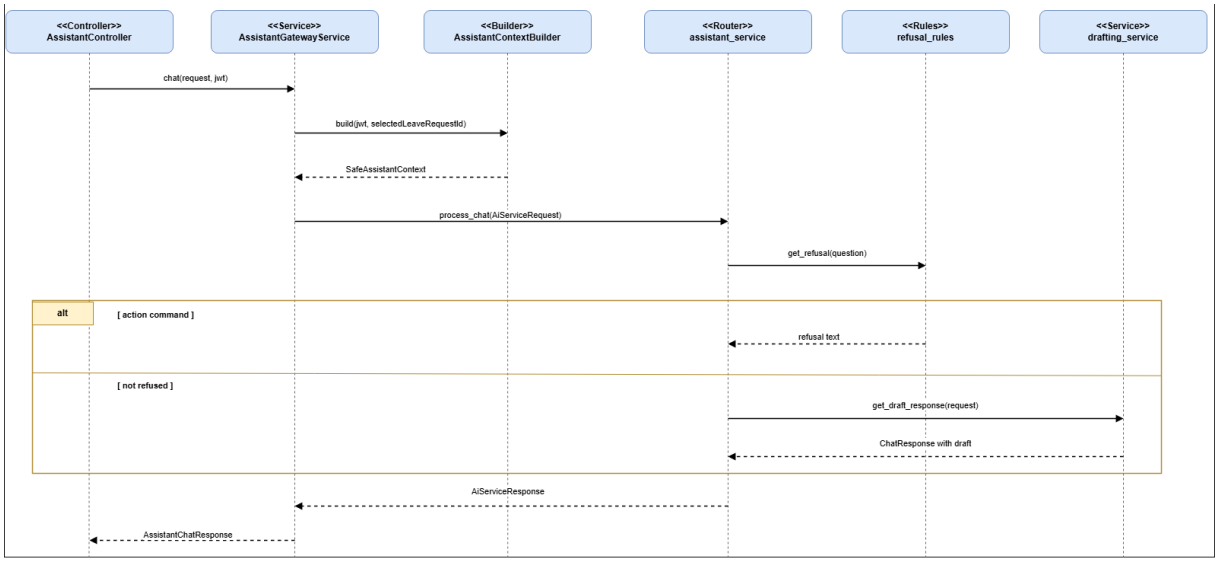


Figure 196: Conception Sequence Diagram – Assistant Request Drafting Backend

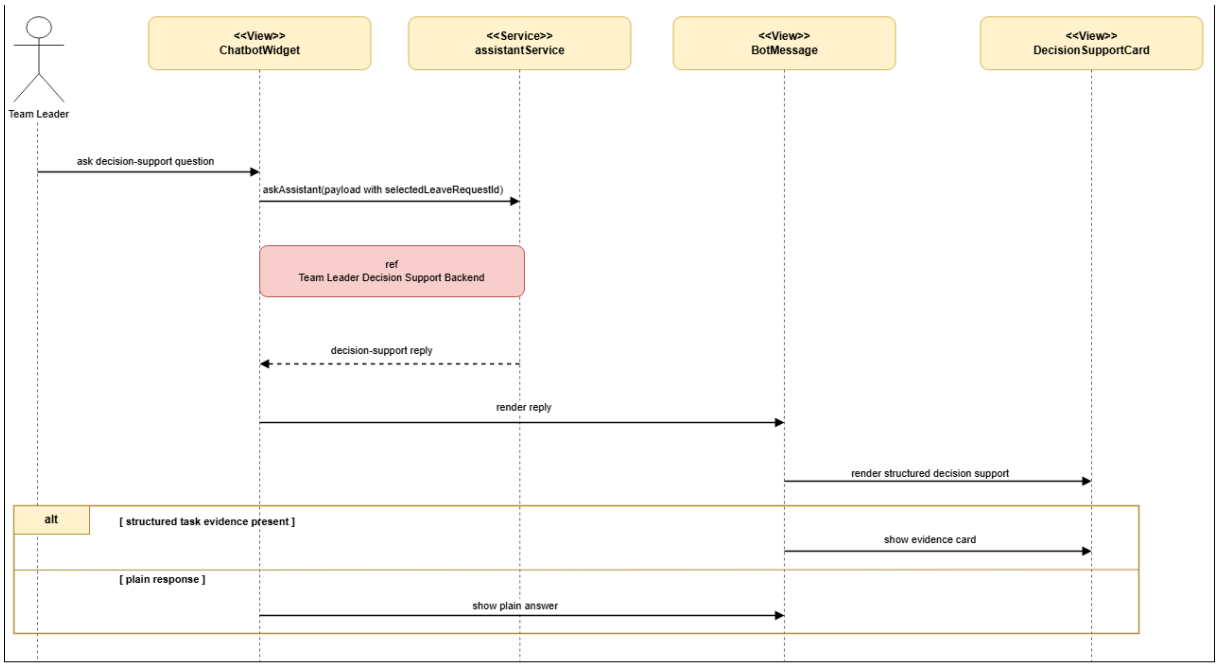


Figure 197: Conception Sequence Diagram – Team Leader Decision Support Frontend

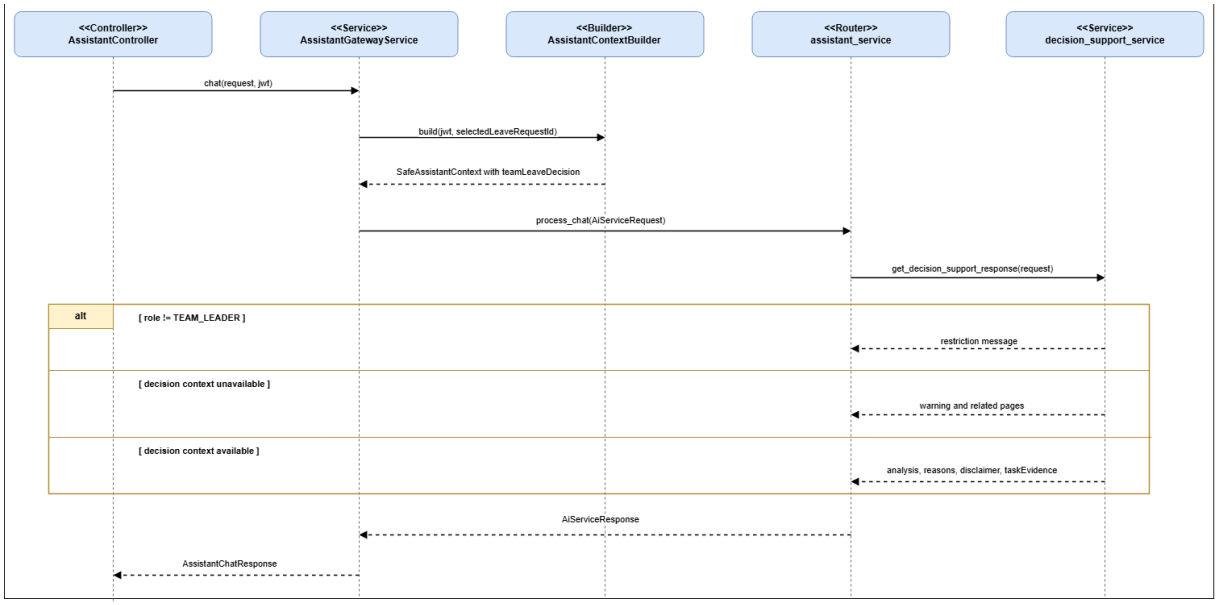


Figure 198: Conception Sequence Diagram – Team Leader Decision Support Backend

Realization

The realization section presents the assistant widget, draft preview cards, decision-support panel, and the local packaging used to validate the sprint scope.

Interface Screenshots

The following screenshots document the Sprint 6 assistant interfaces.

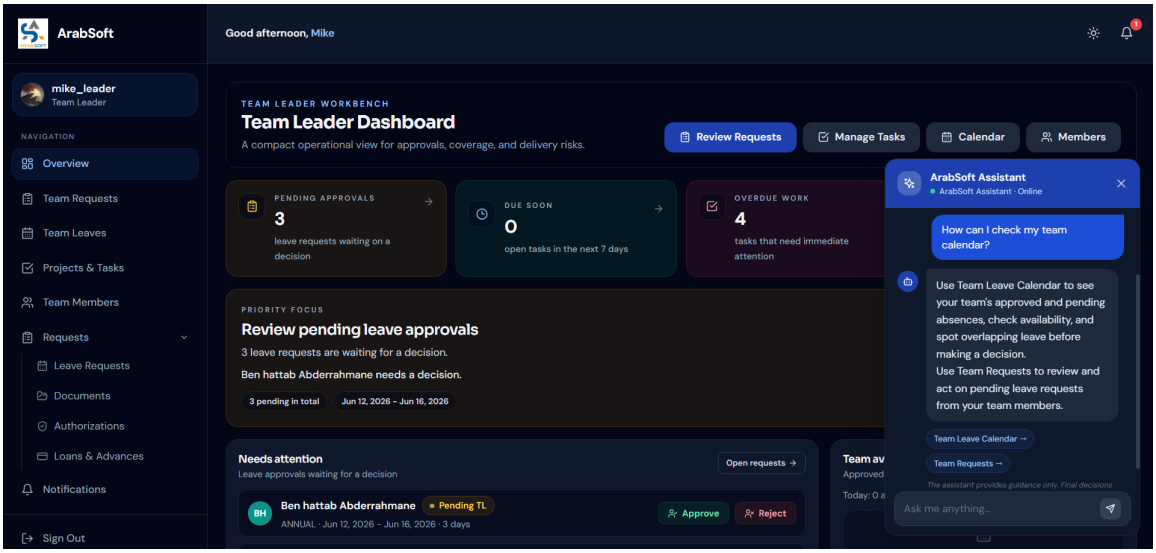


Figure 199: Assistant Platform Guidance Interface

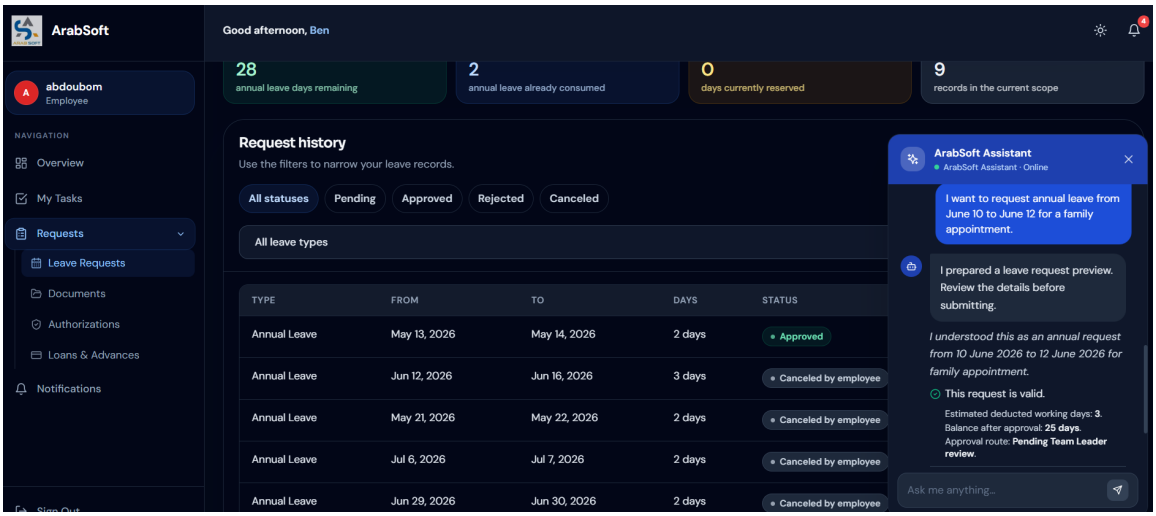


Figure 200: Assistant Leave Draft Preview Interface

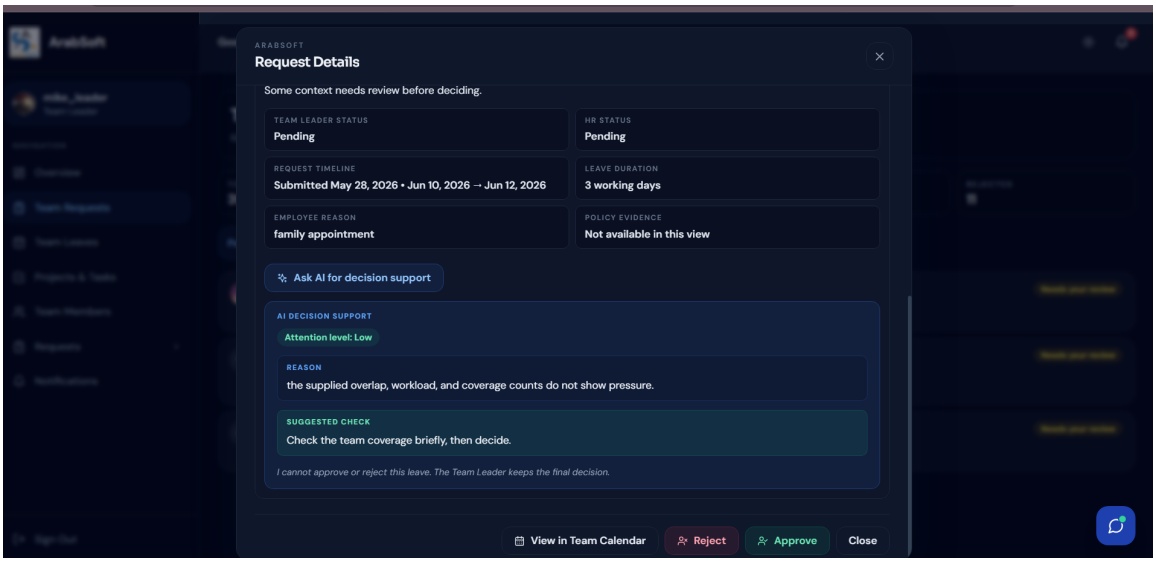


Figure 201: Team Leader Decision Support Interface

Testing and Validation

This section summarizes the regression coverage and safety checks already present in the implementation.

5. Sprint Review

Sprint 6 completed the assistant scope added in Release 3. The assistant was integrated through the React widget, Spring Boot gateway, and FastAPI AI service. [18] It supports platform questions, request-draft preparation, draft validation preview, suggested page navigation, unsafe-action refusal, and Team Leader decision-support guidance. The review confirmed that the assistant remains guidance-only: it does not approve, reject, submit, cancel, change roles, or modify workflow states on behalf of the user. The sprint also covered local packaging and final stabilization checks.

6. Sprint Retrospective

What went well	What needs improvement
The assistant was integrated through a controlled backend gateway while keeping workflow actions under user control.	**Request drafting, validation preview, suggested navigation, and Team Leader decision-support guidance were documented and tested.**
Future work should keep the assistant scope narrow and avoid claiming actions that are not implemented.	**Any future deployment work should remain clearly separated from the current local packaging and validation scope.**

Table 11: Sprint Retrospective – Sprint 6

General Conclusion

This end-of-studies project focused on the design and development of an internal platform for managing requests, projects, and tasks at ArabSoft. The main objective was to centralize internal workflows, improve traceability, and provide employees, team leaders, and HR managers with a structured digital environment adapted to their responsibilities.

The platform was developed progressively through six sprints. The first releases established the core foundations of the system, including authentication, role-based access, leave management, administrative requests, HR supervision, calendars, project tracking, task assignment, and reporting support. These modules helped replace fragmented manual processes with clearer and more reliable digital workflows.

The final release added advanced support features, including reporting, Excel export, and an AI-assisted guidance layer. The assistant was integrated through a controlled architecture using a React widget, a Spring Boot gateway, and a FastAPI service. [18] Its role remains limited to platform guidance, request drafting support, validation preview, suggested navigation, and Team Leader decision-support explanations, while all workflow decisions remain under user control.

This project also allowed me to strengthen my skills in React, Spring Boot, PostgreSQL, Keycloak, Kafka, FastAPI, UML modeling, testing, and Scrum-based organization. In conclusion, the project delivered a functional platform aligned with the needs identified at the beginning of the internship, while providing a solid foundation for future improvements such as UI refinement, broader local deployment preparation, and additional analytics features.

Bibliographie et Nétographie

[1] Object Management Group – Unified Modeling Language specification and resources: consulted on 27/05/2026, url: https://www.omg.org/spec/UML/

[2] Scrum Guides – The official Scrum Guide: consulted on 27/05/2026, url: https://scrumguides.org/scrum-guide.html

[3] Microsoft – Visual Studio Code documentation: consulted on 27/05/2026, url: https://code.visualstudio.com/docs

[4] JetBrains – IntelliJ IDEA documentation: consulted on 27/05/2026, url: https://www.jetbrains.com/help/idea/

[5] Postman – Postman documentation: consulted on 27/05/2026, url: https://learning.postman.com/docs/

[6] diagrams.net – diagrams.net documentation: consulted on 27/05/2026, url: https://www.diagrams.net/doc/

[7] Git – Official Git documentation: consulted on 27/05/2026, url: https://git-scm.com/doc

[8] GitHub Docs – GitHub documentation: consulted on 27/05/2026, url: https://docs.github.com/

[9] Atlassian – Jira Software documentation: consulted on 27/05/2026, url: https://support.atlassian.com/jira-software-cloud/

[10] The LaTeX Project – LaTeX documentation and project information: consulted on 27/05/2026, url: https://www.latex-project.org/help/documentation/

[11] React Documentation – Official documentation for building user interfaces with React: consulted on 27/05/2026, url: https://react.dev/

[12] Tailwind CSS Documentation – Official documentation for the utility-first CSS framework: consulted on 27/05/2026, url: https://tailwindcss.com/docs

[13] Spring Boot Documentation – Official documentation for creating Spring Boot applications and REST services: consulted on 27/05/2026, url: https://spring.io/projects/spring-boot

[14] PostgreSQL Documentation – Official PostgreSQL database documentation: consulted on 27/05/2026, url: https://www.postgresql.org/docs/

[15] Keycloak Documentation – Official documentation for identity and access management: consulted on 27/05/2026, url: https://www.keycloak.org/documentation

[16] Docker Docs – Official Docker documentation for containerizing applications: consulted on 27/05/2026, url: https://docs.docker.com/

[17] Apache Kafka Documentation – Official documentation for distributed event streaming: consulted on 27/05/2026, url: https://kafka.apache.org/documentation/

[18] FastAPI Documentation – Official documentation for building APIs with Python: consulted on 27/05/2026, url: https://fastapi.tiangolo.com/

[19] Apache POI – Official documentation for Java APIs used to read and write Microsoft Office files: consulted on 27/05/2026, url: https://poi.apache.org/

Résumé du sujet PFE

Résumé

Ce rapport présente un projet de fin d'études réalisé au sein d'ArabSoft. Il consiste en la conception et le développement d'une plateforme interne destinée à la gestion des demandes, des projets et des tâches. La solution centralise les workflows administratifs, améliore la traçabilité des opérations et fournit un espace numérique adapté aux employés, aux chefs d'équipe et aux responsables RH. Elle couvre la gestion des comptes, des congés, des documents, des prêts, des autorisations, des équipes, des projets, des tâches, des rapports et des notifications. La plateforme repose sur React, Spring Boot, PostgreSQL, Keycloak, Kafka et un service FastAPI dédié à l'assistance intelligente. Le projet a été modélisé avec UML et organisé selon Scrum.

Mots clés : Plateforme interne, gestion des demandes, gestion de projets, gestion des tâches, React, Spring Boot, PostgreSQL, Keycloak, Kafka, FastAPI, assistant intelligent, UML, Scrum.

Abstract

This report presents an end-of-studies project carried out at ArabSoft. It focuses on the design and development of an internal platform for managing requests, projects, and tasks. The solution centralizes administrative workflows, improves operational traceability, and provides a digital workspace adapted to employees, team leaders, and HR managers. It supports account management, leave requests, document requests, loans, authorizations, teams, projects, tasks, reports, and notifications. The platform is built with React, Spring Boot, PostgreSQL, Keycloak, Kafka, and a FastAPI service for intelligent assistance. The project was modeled using UML and organized according to Scrum.

Keywords : Internal platform, request management, project management, task management, React, Spring Boot, PostgreSQL, Keycloak, Kafka, FastAPI, AI assistant, UML, Scrum.

الملخص

يقدم هذا التقرير مشروع ختم الدراسة المنجز داخل شركة ArabSoft. يتمثل المشروع في تصميم وتطوير منصة داخلية لإدارة المطالب والمشاريع والمهام. تهدف هذه المنصة إلى مركزية سير العمل الإداري، تحسين تتبع العمليات، وتوفير فضاء رقمي مناسب للموظفين ورؤساء الفرق ومسؤولي الموارد البشرية. تشمل المنصة إدارة الحسابات، مطالب العطل، الوثائق، القروض، التراخيص، الفرق، المشاريع، المهام، التقارير، والإشعارات. تعتمد المنظومة على واجهة React، وخادم خلفي Spring Boot، وقاعدة بيانات PostgreSQL، ونظام Keycloak للمصادقة، و Kafka للتبادلات غير المتزامنة، وخدمة FastAPI للمساعدة الذكية. كما تم اعتماد UML في التصميم ومنهجية Scrum في تنظيم مراحل الإنجاز.

الكلمات المفتاحية : منصة داخلية، إدارة المطالب، إدارة المشاريع، إدارة المهام، React، Spring Boot، PostgreSQL، Keycloak، Kafka، FastAPI، مساعد ذكي، UML، Scrum.