DEMOCRATIC AND POPULAR REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
Mohamed Khider University – BISKRA
Faculty of Exact Sciences
Department of Computer Science

Order number ... /M2/2025

Graduation Project Report

2nd year Master's academic degree
Specialty: Applied Informatics in Enterprise Management

Startup for the Design and Development of a Human Resource Management Platform for the Algerian Public Sector

---

Conducted by

- Nesrine Tabbi Anneni
- Yasmine Rahma
- Khedidja Mekhatria

Supervised by

- Toufik Bendahmane
- Mohamed Djalleb
- Ali Sriti

Defended on 16/06/2025 before the jury composed of:

Meadi Mohamed Nadjib	President
Bendahmane Toufik	Supervisor
Zerarka Nour El Houda	Examiner

Academic year 2024–2025

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Acknowledgments

First and foremost, we would like to express our deepest gratitude to **Allah**, whose guidance and blessings have supported us throughout this journey. Without His help, this achievement would not have been possible.

We also thank ourselves for the effort, patience, and perseverance we showed in completing this project.

Our sincere and profound thanks go to our supervisor and mentor, **Mr. Ali Sriti** . His constant support, valuable advice, and exceptional guidance were crucial to every step of this work. His dedication, availability, and encouragement had a significant impact on our learning experience, and we are truly grateful for his mentorship.

We also extend our appreciation to **Mr. Toufik Bendahmane** and **Mr. Mohamed Djalleb** for their supervision, guidance, and helpful feedback throughout this project.

Special thanks go to **Madam Laila** , Director of Human Resources at the Directorate of Youth and Sports, and **Mr. Mohamed** computer engineer at DJS Biskra, and **Mr. Soufiene**, DRH at the National Sports Office Complex in Biskra, for welcoming us during our internship and for their help in testing the application in a real-world environment. Their cooperation was invaluable to our progress.

We are truly thankful to everyone who supported us during this journey.

Dedication

With deep gratitude, I dedicate this humble work to:

My beloved mother and father, who have been there for me since the day I was born, and never doubted me. Who always encouraged me and rewarded me for my hard work.

To my lovely sister, my soulmate and my twin, **Ilham Tabbi Anneni**, who did complain about the lights being on at night, but never turned them off so I could keep studying.

To my dearest friends and classmates, **Yasmine Rahma** and **Khedidja Mekhatria**, who always listened to all my nagging about the smallest details and also complained, but still listened. Who shared with me my ambitions and dreams, and always motivated me with their hard work and dedication.

To my supervisor, **Ali Sriti**, whom I've always considered to be more than just a teacher he was a mentor and a second father. He encouraged me and believed in me when I didn't believe in myself, and made me realize that I can be much more than I thought I could be. You have my utmost gratitude and thanks.

And finally, to all my dear friends that I met and took this path alongside thank you, all of you.

Nesrine Tabbi Anneni

Dedication

In the name of Allah, the Most Gracious, the Most Merciful,

I dedicate this work, first and foremost:

To myself for enduring, persevering, and staying strong through every challenge.
For always pushing forward, because I know I am worthy.

To my beloved family, who stood by my side, supported my choices, and believed in me completely, never once doubting my ability.

To my two sisters by heart, not by blood ***Loudjeyna Djeddou*** *and* ***Nour El Houda Djeddou*** *whose presence turned every hardship into something bearable. You never let go of my hand, and that made all the difference.*

To all my dear friends, too many to name, who brought joy into my days and strength into my journey. Your silent pride and uplifting energy meant more than words could say.

To my friends and classmates, ***Tabbi Anneni Nesrine*** *and* ***Rahma Yasmine****, whose contributions to this work were filled with professionalism, care, and generosity.*

To my professor and role model, ***Ali Sriti****, a true blessing and an answered prayer. Thank you for helping me see my future with hope and clarity.*

To every soul who supported me with a kind word, a warm smile, or even a silent prayer your encouragement carried me further than you'll ever know.

And to every teacher, mentor, and classmate who shaped my path since elementary school, who planted in me a lifelong love for learning your influence still shines within me.

To all of you, I offer my deepest gratitude and heartfelt prayers.

Mekhatria Khedidja

Dedication

All praise to Allah for His mercy and guidance. With a grateful heart, **I dedicate this work to...**

To myself for never giving up. For the sleepless nights, the hard work, the tears, and the triumphs. For standing tall after every fall, for pushing through the difficult moments, and for believing that I could always reach higher. I am proud of all that I've achieved during my university journey.

To my beloved mother your endless love, constant care, and unwavering support, both emotional and physical, have carried me more than words can express. You've always believed in me, even when I struggled to believe in myself.

To my father you've been my foundation. Your words, "Studying is good — do more," still echo in my heart. You opened the right paths for me and reminded me to keep going, to keep achieving. You gave me strength, and I'll always carry that with me.

To my sister ***Rania.M*** *thank you for always being there. For every word of encouragement, every celebration of my small and big wins, and for always knowing how much I can do — even when I forget. You've always been my quiet strength and biggest cheerleader.*

To my dear brother ***Ahmed Rami*** *the funny soul and the youngest light of our family thank you for every bit of support, for every surprised face when I achieve something new, and for always cheering me on in your own unique way. You are truly the light of my heart, and your presence brings joy and comfort to my journey.*

To my supervisor, ***Ali Sriti****,my deepest gratitude goes to you, not only for your academic guidance but also for being a mentor who helped me discover the real Yasmine. Your constant support, encouragement, and belief in me changed the course of my journey. Thank you for walking beside me during this master's program and, hopefully, beyond.*

To my dear friends, ***Tabbi Anneni Nesrine*** *and* ***Khedidja Mekhatria,****you are not just teammates or classmates. You are dream-sharers. May Allah preserve our bond, bless our friendship, and guide us toward the success we dream of. You've been with me in every step, and I'm lucky to walk this path with you.*

And finally, to everyone who supported me throughout my university life friends, colleagues, teachers, and family , thank you all.

Rahma Yasmine

Contents

List of Figures

Abstract

KYN-HR is a project that focuses on the development of a human resources management (HRM) system tailored specifically for Algerian public institutions. While digital transformation has modernized many sectors globally, HR management in Algeria's public sector still relies heavily on outdated, paper-based processes. Existing digital tools are either non-compliant with national laws, unable to generate official documentation, or overly dependent on cloud infrastructure, raising security concerns. To address these issues, we designed a web-based HRM platform that complies fully with Algerian labor laws, supports local and cloud hosting, and automates key HR functions such as employee management, promotions, leave tracking, and legal documentation. This report presents the motivations behind the project, its technical design, and the results of its implementation, aiming to improve efficiency in public sector HR management.

Key words: Human Resource Management, HRM, algerian public institution, web application.

ملخص

يهدف هذا المشروع إلى تطوير نظام لتسيير الموارد البشرية يدعى KYN-HR مخصص للمؤسسات العمومية في الجزائر. بالرغم من التقدم التكنولوجي، ما زالت العديد من هذه المؤسسات تعتمد على طرق تقليدية وإجراءات ورقية تسبب ضياع الوقت وارتكاب الأخطاء الحلول الموجودة حاليا لا تتماشى مع القوانين الجزائرية، ولا تُمكّن من توليد وثائق قانونية رسمية، وبعضها يعتمد على التخزين السحابي مما يطرح مشاكل في خصوصية البيانات النظام الذي تم تطويره في هذا المشروع يقدم حلاً رقميًا كاملاً، يُراعي القوانين الجزائرية، ويُتيح الاستضافة محليًا أو على السحابة. يوفر وظائف مهمة مثل إدارة الموظفين، تتبع الترقية، إدارة العطل، وتوليد الوثائق القانونية بشكل آلي يعرض هذا التقرير خلفية المشروع، طريقة تصميمه وتطويره والنتائج المحققة، بهدف تسهيل وتحسين تسيير الموارد البشرية في القطاع العمومي.

الكلمات المفتاحية

تسيير الموارد البشرية، المؤسسات العمومية الجزائرية، تطبيقات الويب.

General Introduction

Over the past few decades, the world has witnessed a technological revolution that has profoundly transformed the way businesses and institutions operate. From internal management to production and strategic decision-making, technology has revolutionized every aspect of organizational functioning. It has enabled data collection, centralization and analysis, providing decision-makers with clear, real-time insights into their organization's performance and long-term trends.

One of the most impactful changes has been the digitization of internal management systems, including core functions such as budgeting, inventory management, enterprise resource planning (ERP), and human resource management (HRM).

Among these, HRM plays a particularly crucial role. Employees must be carefully selected, managed, motivated, and positioned in roles where they can deliver maximum value. However, with the large number of employees in most institutions, tracking and managing their records manually becomes a tedious and error-prone task.

In Algeria, public institutions still face considerable challenges in effectively managing human resources. These challenges largely come from outdated, paper-based management systems that are time-consuming, labor-intensive, and prone to errors. Unlike many developed countries that have embraced digital HRM systems, Algerian public institutions have yet to adopt such solutions.

While some technological tools do exist both developed locally and internationally none are currently being used in Algerian public sector HRM. This is due to several key limitations:

- Lack of compliance with Algerian public sector laws;
- Inability to generate official legal documentation;
- Dependence on cloud-based infrastructure, which raises concerns about data privacy and security.

These challenges have inspired us to develop **KYN-HR**, a dedicated human resource management system designed specifically for the public sector of Algeria. The objective of this project is to digitize HRM in Algerian public institutions by offering a local platform that:

- Complies fully with Algerian labor and administrative laws;
- Automatically generates legally compliant documents;
- Available as a local install or cloud-based solution, depending on each institution's security needs and preferences.

The proposed platform integrates all the essential HRM functionalities needed, including:

- Employee data management;
- Promotion tracking;
- Leave management;
- Position and grade management;
- Regulation and decision handling;
- Authentication and authorization.

This report outlines the full scope of the project, detailing the rationale behind its development, the design and implementation process, and the results achieved. It is organized into five chapters: the general context; the rationale for using web-based platforms and the adopted architecture; a specification and analysis chapter; and finally, chapters dedicated to design, development, and implementation.

This project aims to provide substantial value to Algeria's public institutions by modernizing and simplifying HR processes and reducing dependency on inefficient paper-based systems.

Chapter 1

HRM and Web Applications: Concepts and Technologies

1.1 Introduction

Human Resource Management (HRM) is essential for managing the employee lifecycle and ensuring organizational efficiency. In Algeria, public institutions follow specific legal frameworks such as Law No. 03-06, which requires HRM tools to be both effective and compliant with national regulations.

This chapter explores HRM in the context of Algerian public administration and the digital solutions used to support it. It focuses on the advantages of web applications over desktop and mobile alternatives, and reviews common architectural models—from simple structures to advanced approaches like Clean Architecture. The chapter concludes by explaining why Clean Architecture is best suited for developing a scalable, maintainable, and law-compliant HRM system for Algeria.

1.2 Overview of HRM

In this section, we will explain what Human Resource Management (HRM) is, look at how it is practiced in Algeria, and review the different software solutions available for managing HR tasks.

1.2.1 Human Resource Management Definition

Human resource management (HRM or simply **HR**) is the term most commonly used to describe all those organisational activities concerned with recruiting and selecting, designing work for, training and developing, appraising and rewarding, directing, motivating and controlling workers. In other words, HRM refers to the framework of philosophies, policies, procedures and practices for the management of the relationship that exists between an employer and worker. [1]

1.2.2 HRM Functions

Human Resource Management (HRM) is responsible for every stage of the employee life-cycle within an organization, including recruitment, training, development and performance evaluation. While these activities are common across most organizations, their implementation often differs based on internal policies, legal requirements, and specific working conditions. However, despite these variations, the core functions they share remain largely consistent and include: [1]

- People resourcing, ensuring optimal staffing for current and future business needs through activities that include human resource planning, recruitment, selection, talent management, succession planning and the termination of the employment relationship (including managing retirement and redundancy).
- Managing performance, managing individual and team performance and the contribution of workers to the achievement of organisational goals.
- Managing reward, designing and implementing reward and pay systems covering individual and collective, financial and non-financial reward, including employee benefits, perks and pensions.
- Human resource development, identifying individual, team and organisational development requirements and designing, implementing and evaluating learning and development interventions.
- Employment relations, managing employee 'voice', communication and employee involvement (EI) in organizational decision making, handling union-management relations, managing employee welfare, and handling employee grievances and discipline.

1.2.3 HRM in Algerian Public Sector

Human Resource Management in the Public Service refers to all the tasks and processes involved in the management of employees in public institutions and administrations that follow the regulations of **General Civil Service Law (Law No. 03-06)**. This law explains the rights and duties of public employees and highlights the main activities of HR management in this sector, including the following:

- Recruitment: hiring the right people for public jobs
- Probation period management: guiding and evaluating new employees during their trial period
- Career administration: managing the different steps of an employee's professional journey
- Performance evaluation: checking how well employees are doing their jobs

- Training and development: helping employees grow their skills and knowledge
- Promotion: giving employees higher positions when they deserve it
- Compensation and rewards: making sure employees are fairly paid and appreciated for their work

These activities ensure that public employees are selected, supported, and motivated in a way that improves the efficiency and fairness of public service. [2]

1.2.4 HRM Systems

A Human Resource Management System (HRMS), refers to the systems and processes at the intersection between human resource management (HRM) and information technology. It merges HRM as a discipline and in particular its basic HR activities and processes with the information technology field. [3]

1.2.5 HRM System Types

Human resource management systems have evolved significantly since the advent of digital technology, offering diverse solutions specific to organizational needs. Broadly, HRM systems can be categorized into comprehensive solutions (covering multiple HR functions) and stand-alone solutions (focused on specific HR tasks).

A. Comprehensive Solutions

These systems integrate multiple HR functions into a unified platform, simplifying processes such as payroll, talent management, and analytics. The three main types are: [4]

- **Human Resources Information System (HRIS):**
  HRIS is designed to manage data from various HR processes such as benefits, workforce management, payroll, and core HR functions. The most sophisticated HRISs are interactive information management systems that standardize HR tasks and processes while facilitating accurate recordkeeping and reporting.
- **Human Resource Capital Management (HCM):**
  Expanded on the HRIS to offer a more complete suite of software that organizations could use to manage internal HR functions.
- **Human Resource Management System (HRMS):**
  HRMS Is now commonly used to describe a complete suite of HR applications, built in the cloud, that are designed to improve the employee experience. An HCM solution today often incorporates digital assistants, AI, and other tools that enable users to collaborate and share information across teams. Additional functionality includes advanced talent management tasks such as performance management, learning, succession planning, and compensation planning.

The differences between these solutions can be summarized in the table below:

Aspect	**HRIS**	**HRMS**	**HCM**
Focus	Administrative tasks	HR operations + employee lifecycle	Strategic workforce planning & employee value
Scope	Core HR data (payroll, records)	HRIS + training, recruitment, performance	HRMS + talent strategy, analytics, succession
Features	Data storage, compliance, payroll	HRIS + onboarding, development tools	HRMS + AI tools, workforce modeling, planning
Strategic Level	Low	Moderate	High
Technology Base	Often on-premise or basic software	Mix of on-premise and cloud	Mostly cloud-based with advanced capabilities

Table 1.1: Comparison of HRIS, HRMS, and HCM systems

This table compares the HRIS, HRMS, and HCM systems. HRIS handles basic HR tasks like payroll and record keeping. HRMS adds features to manage employee training, recruitment, and performance. HCM goes further by focusing on strategic planning and using advanced tools such as AI to better manage the workforce. In general, systems grow from simple administration to more strategic and data-driven HR management.

B. Stand-alone Solutions

These systems target specific HR functions and often integrate with comprehensive platforms:

- **Applicant Tracking System (ATS):** An applicant tracking system (ATS) logs all open positions within an organization and optimizes workflows to fill them as quickly as possible. ATS functions include syndicating job postings to multiple job boards, screening applications to identify qualified candidates, and identifying bottlenecks within the hiring process. [5]

- **Performance Management System:** A performance management system stores employee information on performance evaluations. It supports employee retention, promotion, transfer, job rotation, contract termination needs, and other talent management concerns. A performance management system helps managers take action if an employee is underutilized or needs extra support. [5]

- **Benefits Administration System** A benefits administration system is a tool within HR that helps manage employee compensation and benefits programs. It supports HR leaders in tracking benefit usage, ensuring employees are informed about available

benefits, and making data-driven decisions to adjust or improve the benefits offered as part of total compensation. [6]

1.2.6 Existing HRM Solutions

After going over the different types of HRM systems, it's important to take a look at some of the solutions that are already available both locally and globally. In this section, we'll briefly review a few well-known HRM tools on the market, then compare them to our own solution, with a focus on what makes ours more suitable for Algerian public institutions.

A. Global Solutions

- **Oracle HCM Cloud** [7] is a globally recognized human capital management platform offering extensive features such as employee record management, self-service tools, performance tracking, and workforce planning. While technically advanced, it is primarily designed for multinational environments and lacks adaptation to Algerian laws and administrative procedures. Furthermore, as a fully cloud-based system, it poses data residency concerns especially for government bodies that are sensitive about where employee data is stored.
- **BambooHR** [8] is another global solution, known for its simplicity and user-friendly interface. It provides features like time-off tracking, performance management, and employee databases. However, it does not accommodate the Algerian legal framework, and like many global systems, it is hosted on the cloud raising data security concerns for local public institutions.

B. Local Solutions

- Several HRM solutions have emerged on the Algerian market, such as **OnyxHR** [9] by **Solutions Intelligentes Informatiques.** This platform offers functionalities such as employee record management, attendance, and payroll processing. However, it is targeted to private institutions and often fall short in terms of compliance with the public service civil law. furthermore, it lacks comprehensive feature like administrative documentation and auto file generation
- Another solution that is occasionally used in Algeria, is **Odoo's Employees App** [10], which is part of the broader Odoo ERP suite. It includes employee management, contracts, time-off tracking, and organizational hierarchy features. While **Odoo** is flexible and relatively cost-effective, it often requires technical customization and lacks specific built-in features related to Algerian public sector regulations.

1.2.7 Comparison

Each of the previous solutions is different and meets specific needs. The table below shows the main differences between them, with a focus on what matters most for HR systems in

Algeria's public sector.

Feature	**Oracle HCM Cloud**	**OnyxHR**	**Odoo Employees**	**Our HRM Solution**
Legal Compliance (Algeria)	No	No	No	Fully integrated
Language	English	French/Arabic	Multilingual	Arabic-native
Hosting	Cloud	Cloud	Cloud	both local and cloud
Public Sector Adaptation	Not tailored	Limited	Custom setup	Built for public institutions

Table 1.2: Comparison of Existing HRM Solutions

1.2.8 Our Solution

Ou solution **KYN-HR** is a human resource management system specifically developed for Algerian public institutions. From the outset, it was designed to meet the exact needs of government organizations. It offers both local and web-based deployment options, ensuring flexibility and adaptability across various infrastructures. Entirely in Arabic, it supports ease of use for administrative staff, and most importantly, it ensures full compliance with Algerian labor laws, down to the smallest detail.

In contrast, most global HR tools, while rich in features, are not built with the Algerian context in mind. They often lack legal compliance with national regulations and rely heavily on cloud-based infrastructure, which many government clients cannot adopt. Local tools offer a closer fit but still fall short in several key areas, especially in automation, legal support, and the integration of modern HR functionalities.

While other solutions attempt to adjust to our environment, KYN-HR was built around it.

1.3 Web Applications

1.3.1 The Different types of applications

In today's diverse software landscape, applications vary widely in their design, purpose, and complexity. Understanding the different types of applications helps us choose the right development approach and tools. This section provides an overview of the main categories of applications relevant to our project context.

A. Desktop Applications

A desktop application is a software installed and run directly on a personal computer or workstation. It operates independently of a web browser and can function offline. These applications are typically platform-dependent and are built using frameworks specific to the operating system, such as WPF or WinForms for Windows, Cocoa for macOS, or GTK for Linux. [11].

(a) Microsoft Word (b) Apple Pages

Figure 1.1: Examples of Desktop Applications

B. Mobile Applications

A mobile application is a specialized software program engineered to execute on handheld computing devices (e.g. smartphones, tablets) by leveraging the device's native operating system runtime environment and hardware abstractions. [12]

(a) android play Store (b) IOS apple Wallet

Figure 1.2: Examples of Mobile Applications

There are four major development approach when building mobile applications [13]

- **Native Apps:** Built for a single platform (Android or iOS) using platform-specific

languages (Java/Kotlin or Swift). High performance and full hardware access (e.g., WhatsApp, Spotify).

- **Web Apps:** Run in browsers using HTML, CSS, JavaScript. Cost-effective but limited hardware access (e.g., Google Docs mobile).
- **Hybrid Apps:** Combine web and native code in one package using tools like Ionic or Cordova. Work on multiple platforms and access device features (e.g., Instagram, Uber).
- **Cross-Platform Apps:** Single codebase for all platforms using frameworks like Flutter or React Native (e.g., Airbnb, Tencent).

C. Web Applications

A web application (web app) is a software program hosted on a remote server and accessed via a web browser over the internet. Unlike traditional desktop applications, which require installation and are often platform-dependent, web applications run within the browser interface and can be used across multiple devices including desktops, smartphones, and tablets without the need for installation. Web apps are designed for a broad range of users and purposes, from individuals to enterprises, and include services such as webmail, social networking platforms, e-commerce systems, and online productivity tools. Their ability to function across various browsers and operating systems makes them highly accessible and easier to maintain compared to mobile or desktop applications. [14]

1.3.2 Rationale for Choosing a Web Application

In designing our HRM system for the public sector, we carefully considered the available technologies and chose to develop a web application. This decision was based on a set of practical and strategic benefits that align with the goals of our project and the constraints of the Algerian public administration

a. **Accessibility:** With a web-based solution, we ensure that the application can be accessed from any device equipped with a standard web browser. This allows HR managers to use the system from their workstations without the need for specific software installations in the case of the web-based option. For locally hosted installations, a one-time setup is required, after which the system remains easily accessible within the organization's internal environment. In contrast, desktop applications require installation on each individual computer, and mobile apps are often limited to specific operating systems, which can restrict accessibility.

b. **Uniform User Experience:** Web applications offer a consistent interface across different operating systems and devices. Whether the system is accessed from a desktop running Windows or Linux, the experience remains the same. This uniformity reduces

training time and increases usability. On the other hand, Desktop applications often have different versions tailored for each OS, while mobile apps vary significantly between platforms like Android and iOS, leading to inconsistent user experiences.

c. **Simplified Maintenance and Updates:** One key advantage for choosing web applications is centralized management. Updates or fixes we apply it directly on the server, so all users benefit instantly without needing to reinstall or manually update software. In contrast, desktop and mobile applications usually require users to download and install updates separately, which can cause delays and inconsistencies in versions across users, and may also require hardware upgrades to support new features or performance improvements.

d. **Flexible Deployment Options:** Choosing a web application architecture allows us to offer two modes of deployment either as a fully web-based solution accessible via a browser or as a locally hosted application within an organization's internal network. This flexibility makes the solution suitable for the Algerian public sector's needs of security.By comparison Desktop and mobile apps are generally tied to their installation environment and lack such deployment flexibility.

e. **Cost Efficiency:** Developing and maintaining a single web application is generally more affordable than creating separate native apps for each operating system. Organizations benefit from reduced development time, shared codebases, and fewer resources required for distribution and updates.On the contrary, desktop and mobile apps require multiple versions and ongoing maintenance for each platform, increasing costs.

In summary, choosing a web application allowed us to deliver a powerful, flexible, and maintainable system that respects the operational context of the Algerian public sector all while ensuring simplicity, consistency, and efficiency in daily use.

1.3.3 Types of Web App Architectures

Web applications can be designed using various architectural patterns, each defining how data flows and how responsibilities are distributed. These models help developers build scalable, maintainable, and testable systems.

a. **Layered (N-Tier) Architecture :** this architecture divides an application into logical layers and physical tiers, such as presentation, business logic, and data access layers, promoting separation of concerns and maintainability. [15]

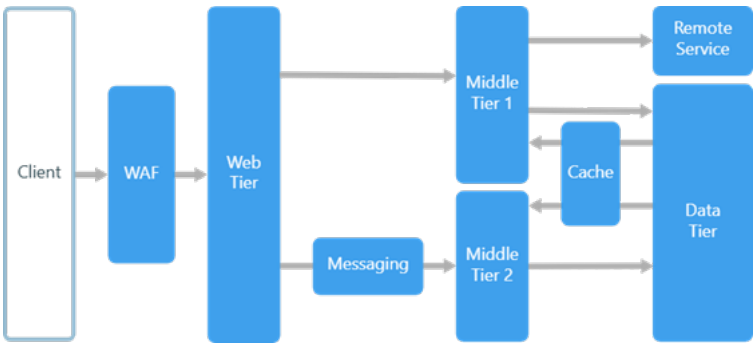


Figure 1.3: Example of a Layered (N-Tier) Application Architecture

b. Client-Server Architecture : Client-server architecture mba3d 3aytilha (see Figure 1.4) is a computing model that divides tasks or workloads between service providers, called servers, and service requesters, called clients. This model is designed to improve the efficiency and management of resources by centralizing server functions, which handle the processing and management of data, while clients interact with the user and request services from the server . [16]

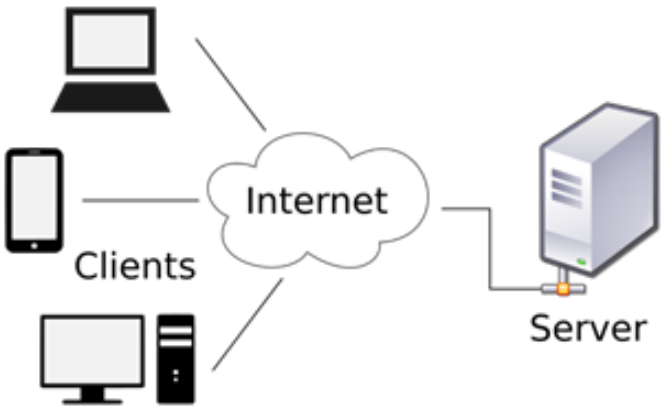


Figure 1.4: Example of Client-Server Architecture

c. Microservices Architecture : Microservices architecture mba3d 3aytilha (see Figure 1.5) is a software design pattern in which a complex application is broken down into a suite of small, independent services. Each service is designed to fulfill a specific business capability, runs in its own process, and communicates with other services through lightweight protocols (typically HTTP/REST or messaging queues). These services are independently deployable, loosely coupled, and organized around business domains, which enhances scalability, maintainability, and organizational agility. [17]

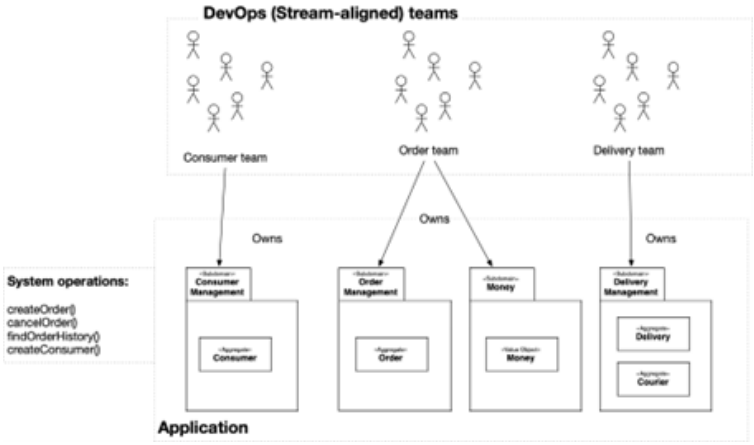


Figure 1.5: Example of Illustration of Microservices Architecture with Domain-Aligned Teams

d. Clean Architecture : Clean Architecture mba3d 3aytilha (see Figure 1.6) is a software design approach that organizes a system into well-defined layers typically including entities, use cases (or interactors), and interface adapters. The main goal is to separate concerns and keep the business logic (core rules) independent of external technologies such as databases, frameworks, or user interfaces. This makes the system easier to maintain, test, and adapt over time. A key principle is that dependencies flow inward from outer layers (like UI or infrastructure) toward the core logic ensuring the core remains unaffected by changes in external components. [18]

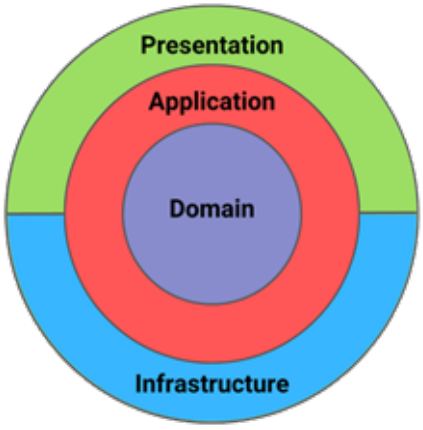


Figure 1.6: Layered Structure of Clean Architecture Following the Onion Model

1.3.4 Clean Architecture as Our Design Approach

When designing our HRM system for the Algerian public sector, we took time to evaluate different architecture models like Layered (N-Tier), Client-Server, and Microservices. While each of these has its strengths, we found that Clean Architecture was the best fit for our needs, especially considering the long-term maintainability and complexity of public sector systems.These are the main reasons why we chose to adopt Clean Architecture for our project:

- **Separation of Concerns**
  One of the main reasons we went with Clean Architecture is how it truly separates the business logic from things like the UI, database, and external frameworks. While the N-Tier model also tries to separate logic, we noticed that in practice, the layers often end up tightly connected. With Clean Architecture, we're able to keep our core HR rules like promotions, recruitment, or position upgrades totally independent, so we can update or test them without affecting the rest of the app.

- **Easier to Maintain Over Time**
  For a system like ours that's expected to evolve and be used over many years, we needed something that wouldn't break every time a technology changes. Clean Architecture lets us switch the database or UI framework without touching the core logic. Compared to the N-Tier or Client-Server models, where changing one part can often mean reworking others, Clean Architecture gives us much more flexibility.

- **Better for Team Collaboration**
  Clean Architecture helped our team work more efficiently. Since everything is clearly separated UI, business rules, and infrastructure we were able to assign different parts to different people without getting in each other's way. Frontend and backend could move in parallel, and merging work was much easier. This really supported our Agile workflow.

- **Easier to Test**
  Because our business logic doesn't depend directly on the database or UI, we were able to write unit tests without needing a full setup or real data. This made it easier to make sure our HR rules work as expected and saved us a lot of debugging time.

- **Built to Last**
  Public sector systems aren't just short-term projects. They need to last. With Clean Architecture, we know we won't have to rebuild everything from scratch in a few years. The core logic is protected from tech trends so even if we change frameworks or tools, our system stays solid.

Summary Comparison with Other Architectures

Criteria	**Layered (N-Tier)**	**Client-Server**	**Microservices**	**Clean Architecture**
Separation of Concerns	Partial; layers often mix logic	External focus only	Strong; per service	Strong; clear layer boundaries
Maintainability	Moderate; layers interdependent	Low; tightly coupled	High; modular but complex	High; logic isolated from tech
Testing	Needs DB setup	Needs full setup	Easy per service	Logic easily testable
Team Collaboration	Basic; not enforced	Limited; centralized server	High; needs coordination	Easy; clear separation of roles
Scalability	Limited; monolithic	Limited; impacts whole app	High; per service scaling	Moderate; scale per layer
Fit for our HRM Project	Rigid; not flexible	Good for flow, not structure	Too complex	Best fit; maintainable, testable

Table 1.3: Comparison of Architecture Models

Clean Architecture gave us the right balance of structure, flexibility, and long-term maintainability. It allowed us to build a solid foundation for a critical public-sector system while keeping the code clean, testable, and easy to evolve.

1.4 Conclusion

In summary, the effective management of human resources within Algerian public institutions requires systems that are both legally compliant and technically robust. Web applications provide a flexible and accessible foundation for such systems, offering advantages over traditional desktop or mobile solutions. Among the architectural models reviewed, Clean Architecture stands out for its clear separation of concerns, testability, and long-term maintainability. Its adoption ensures that HRM solutions remain adaptable to evolving legal and organizational needs while maintaining a solid and scalable technical structure.

Chapter 2

Specification Analysis

2.1 Introduction

This chapter focuses on the specification analysis phase, which plays a vital role in ensuring that the system meets real user needs and operates within its intended context. It begins with an analysis of the case study, including a detailed overview of DJS and the methods used during the internship to gather relevant information. The key challenges faced by the organization are identified to guide the definition of system requirements. Based on this analysis, the chapter presents a structured specification document covering the project scope, system users and roles, and both functional and non-functional requirements. Finally, it discusses the chosen project management methodology highlighting the comparison between classic and agile approaches, justifying the choice of Scrum, and outlining the task distribution and project timeline using a Gantt chart.

2.2 Analysis

In this section, we'll share the background of our project by walking you through a case study and what we noticed during our internship. We'll also highlight the key challenges the institution is facing, that helped shape the design of our solution.

2.2.1 Case Study

DJS – (fr : *Direction de la Jeunesse et des Sports wilaya de Biskra*) (en : *Direction of Youth and Sports of Biskra*)

The Direction of Youth and Sports of Biskra is a decentralized administrative body operating under the supervision of the Ministry of Youth and Sports. According to Executive Decree No. 06-345 of September 28, 2006, its main mission is to develop, promote, coordinate, evaluate, and supervise institutions, structures, and activities related to youth, physical education, and sports within the Biskra province.

DJS Biskra is composed of the following key departments:

- Physical Education and Sports Department .
- Youth Activities Department.
- Investment and Equipment Department.
- Training, Administration, and Logistics Department.

Each department is subdivided into three specialized offices responsible for implementing public policies in their respective areas. [19]

2.2.2 Internship to collect information

As part of the specification analysis, we began by studying the Algerian public civil service laws to understand how Human Resource Management (HRM) is legally structured within the public sector. Our goal was to gain a solid foundation on the official procedures related to employee management, career progression, promotions, and administrative documentation.

To complement this legal understanding with practical insights, we also created an online questionnaire and distributed it to several public institutions. This allowed us to collect a wide range of responses from HR professionals working in various Algerian public organizations, providing us with valuable information about how HR tasks are performed and how data is managed.

Following this, we conducted an internship at the Direction of Youth and Sports of Biskra (DJS Biskra), specifically within the Human Resources (DRH) department. During the internship, the director of the HR department explained in detail how the actual procedures from recruitment and position tracking to promotions and document handling are applied on the ground and how they align with the legal framework.

Based not only on our observation, but also on the online questionnaire responses and discussions during the internship, we concluded that :

- link to the online questionnaire :
https://forms.gle/Ai5smuK2yB1jzXuU6

The department currently relies on traditional tools such as Microsoft Excel and Word to manage HR-related data and to generate official documents. This manual approach, while functional for basic tasks, presents several limitations in terms of efficiency, accuracy, and scalability. The lack of an integrated digital Human Resources Management (HRM) system limits the department's ability to automate repetitive tasks, track employee data effectively, and ensure timely updates across all processes.

2.2.3 Major Problems They Face

From our observations and discussions during the internship at DJS Biskra, along with the insights collected through our online questionnaire, we identified several major problems faced by HRM departments in the public sector.

- **Scattered and Uncentralized HR Data**
  HR data is spread across multiple files, formats, and often physical documents, making it difficult to maintain consistency, ensure data accuracy, or apply updates efficiently.

- **Difficulties in Employee Tracking**
  Without a digital system, tracking the status, history, or performance of employees becomes cumbersome. Important information such as promotions, or leave balances often requires manual searches through spreadsheets.

- **Error-Prone and Time-Consuming Processes**
  Manual data entry increases the risk of human error. Mistakes in employee records, duplicated entries, or incorrect document generation are common and can cause significant administrative delays.

- **Delays in Decision-Making**
  Because information is not readily accessible or updated in real-time, responding to requests or making strategic HR decisions (like staff reassignment or promotion eligibility) often takes longer than necessary.

- **Inefficient Document Generation**
  The reliance on Word documents for official paperwork like contracts, promotions, and evaluations demands a lot of manual editing and formatting, which slows down the process.

- **Limited Data Security and Backup**
  Files stored locally are vulnerable to accidental loss, unauthorized access, or system failure, as there is no robust backup or access control mechanism in place.

- **Poor Scalability and Reporting**
  As the number of employees grows, the traditional methods become harder to manage, and generating statistical or analytical reports becomes nearly impossible without spending hours compiling data.

2.3 Specification Document

This section outlines the project scope, user roles, system requirements both functional and non-functional as well as the expected operating environment for the application.

2.3.1 Project scope

This project aims to develop a web-based Human Resources Management (HRM) system specifically adapted for the public sector, with a focus on the needs of the Direction de la Jeunesse et des Sports (DJS) of Biskra. The goal is to replace the current manual methods

mostly based on Microsoft Excel and Word with a more efficient, automated solution that simplifies the management of employee data and HR processes.

The new system, as a web application, will allow users to access and manage HR tasks from any device connected to the network, increasing accessibility and flexibility for the HR team. In addition, the solution is designed to support both cloud-based and local installation options, depending on the specific needs and constraints of each institution. It will support key HR functions such as employee records, job positions, career tracking, promotions, and leaves. The application will also help generate official documents, reducing the chances of delays and human errors.

By centralizing data in a web-based environment and making it easier to access and update, the system will enable the HR department to save time, stay organized, and work more efficiently.

In short, this project is about supporting the digital transformation of a public institution by introducing a practical tool that improves transparency, reliability, and performance in human resource management.

2.3.2 System Users and Roles

In the current version of the Human Resource Management (HRM) system, the sole defined user role is the HR Manager. This role has full access to all system functionalities, including employee management, leave tracking, promotion handling, and regulation configuration.

2.3.3 Requirement Specification

The requirement specification marks the initial phase in the development of any application. This stage defines all the functionalities that the application must provide. In our project, we carefully analyzed and combined the needs identified during our internship and field study at the DJS Biskra. As a result, we outlined a clear set of system requirements, categorized into two main types: functional and non-functional requirements.

2.3.3.1 Functional requirements

- The system must allow user authentication through a login interface .
- The system must provide a user-friendly interface to streamline HR tasks and improve efficiency across the department.
- The system must enforce validation and confirmation on all critical actions (e.g., deletion of data, promotion, document removal) with success, error, and warning alerts to avoid accidental operations.
- The system must allow managing employee records: add, update, delete, and view employee profile .

- The system must allow the HR manager to view each employee's profile and update all related information, including personal details, family information, educational background, career path and leave records.
- The system must support the uploading and storage of employee documents (e.g., contracts, ID documents), and allow retrieval when needed.
- The system must allow the creation, viewing, updating, and deletion (CRUD) of job positions of the institution .
- The system must allow the creation, viewing, updating, and deletion (CRUD)operations for grades of the institution .
- The system must allow the HR manager to promote employees in terms of both grade and advancement .
- The system must manage leaves : create, update, assign each leave to the correct employee, and specify the type of leave (e.g., sick leave, annual leave) and it duration .
- The system must track all leave history for all employees and ensure leaves are correctly associated with dates and status.
- The system must include configuration management for HR-related regulations and allow the HR manager to update them in line with legal changes.
- The system must allow the HR manager to define and order the applicable regulations for each type of decision so that official documents are generated in the legally required structure, with only the relevant laws included as per the specific case.
- The system must automatically generate official decisions (e.g., promotions, leave approvals) based on configured templates and assigned data.
- The system must allow the HR manager to perform advanced search, filtering, and sorting on data such as employee lists, positions, and grades.
- The system must allow the HR manager to upload data lists based on specific criteria they select.

2.3.3.2 Non- Functional requirements

- **Performance :** The application must provide a smooth user experience with fast loading times and seamless interactions.
- **Security :** The protection of user data is essential, with appropriate encryption and authentication mechanisms implemented to ensure data confidentiality and secure access.

- **Reliability :** The application must be stable and reliable, minimizing crashes and unexpected errors to ensure continuous service.
- **Maintainability :** The system must be easily maintainable, allowing for simple updates, bug fixes, and the addition of new features without major disruptions.
- **Portability :** The application must be usable across various operating systems and web browsers .
- **Data Integrity :** The system must preserve the accuracy and consistency of data across all operations, especially during updates and modifications.
- **Scalability :** The application must support increasing numbers of users and data without compromising performance or functionality.
- **Usability :** The interface must be user-friendly, intuitive, and efficient, enabling HR staff and managers to perform tasks with minimal training.
- **Compliance :** The system must comply with HR regulations applicable to the public sector.

2.3.3.3 Operating Environment

The Human Resource Management (HRM) web application is designed to function as a desktop-oriented web solution accessible through modern web browsers. It operates entirely within a web environment, allowing authorized users to interact with the system through commonly used browsers such as Google Chrome, Microsoft Edge, and Mozilla Firefox.

The application runs on a secure server infrastructure and communicates with a relational database management system to ensure efficient handling of HR-related data.

This setup ensures that the application can be deployed within standard institutional IT environments, requiring only a compatible browser and internet access to function effectively.

However, internet access is only required in the case of cloud-based deployment. For local deployment, the application can operate without an internet connection .

2.4 Project Management

Project management is a fundamental pillar in the success of any software or information system development. It involves organizing, planning, executing, and controlling the activities necessary to achieve the project's objectives. Two major approaches stand out: the classical (also known as traditional) method and the Agile method.

2.4.1 Agile Vs Classic Methods

Classical methods are based on rigorous upfront planning, with a sequential execution of the project phases (analysis, design, development, testing, deployment). The framework is

structured, responsibilities are clearly defined, and changes during the project are difficult to incorporate.

Agile methods, on the other hand, are iterative, focused on collaboration and flexibility. They allow the project to adapt to evolving customer needs and promote frequent deliveries of functional versions. [20]

Comparison Table

Criterion	**Classical Method**	**Agile Method**
Approach	Sequential, linear	Iterative and incremental
Adaptability to changes	Low – changes are hard to integrate	High – continuous consideration of evolving needs
Client involvement	Low – mostly involved at the beginning and end	High – constant collaboration and regular feedback
Deliverables	Final product delivered at the end of the project	Frequent deliveries of usable partial versions
Risk management	Risks identified late in the process	Risks detected and addressed throughout the project
Progress visibility	Limited – based on milestones	Transparent – supported by regular meetings
Response to change	Costly and slow	Fast and continuous
Best suited for	Stable projects with well-defined requirements	Evolving or innovative projects

Table 2.1: Comparison between Classical and Agile Methods [21]

From the table, it's clear that the Classical method is more rigid and follows a step-by-step process, with less client involvement and little room for changes so it works best for projects that are stable and well-defined. On the other hand, the Agile method is flexible and iterative, with ongoing collaboration, regular deliveries, and continuous risk management. Agile makes it easier to adapt to changes and keeps everything transparent, which is great for projects that are evolving or new. Overall, Agile feels like a better fit when things aren't set in stone, while Classical suits projects that need a more fixed plan.

2.4.2 Types of Agile Methods

Agile is not just one method it's a group of methods that help teams work better and faster by breaking big projects into smaller parts. All Agile methods follow the same goal: to deliver working software step by step, get feedback quickly, and adapt easily to changes. Here are four common Agile methods:

1. Scrum

Scrum (see Figure 2.1) is a very popular Agile method. It breaks the work into short cycles called sprints (usually 1 to 4 weeks long). During each sprint, the team works on a small part of the project.

There are three main roles:

- **Product Owner** – decides what the team should work on .
- **Scrum Master** – helps the team follow Scrum rules .
- **Development Team** – – builds the product .

Every day, the team has a short meeting (called a Daily Scrum) to check progress and plan the day. At the end of each sprint, the team shows what they've done and gets feedback. [22]

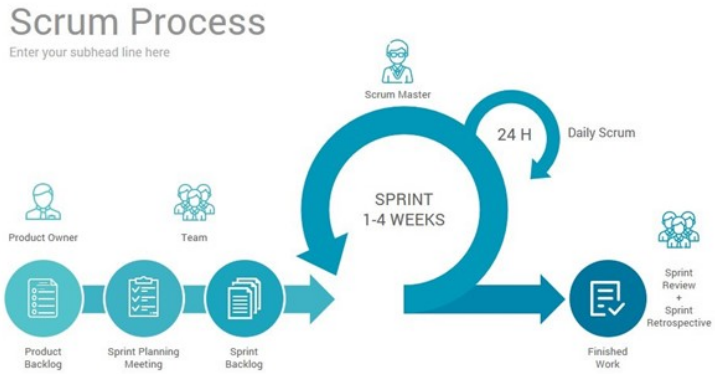


Figure 2.1: Overview of the Scrum process from backlog to sprint review.

2. Kanban

Kanban is a simple and flexible method that helps teams manage their work visually. Tasks are shown as cards on a Kanban board with columns like "To Do," "In Progress," and "Done." This helps the team see what everyone is working on and avoid doing too many things at once. Kanban is great for teams that handle many small tasks like bug fixing or customer support.[23]

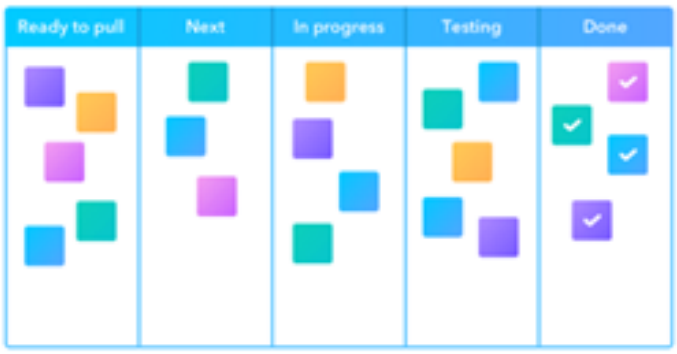


Figure 2.2: Example of a Kanban board illustrating task flow across different stages.

3. Extreme Programming (XP)

Extreme Programming (XP) focuses on writing high-quality code and working closely as a team.

It uses techniques like:

- Writing tests before the code (Test-Driven Development) .
- Two people working together on one computer (Pair Programming) .
- small updates often (Continuous Integration) .

XP is useful for projects where the requirements change a lot and the software must be very reliable. [24]

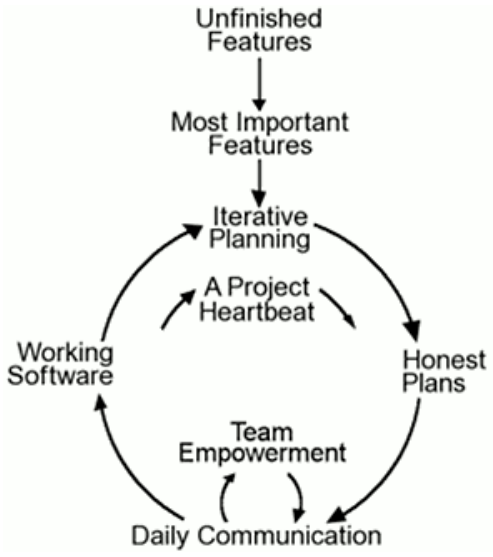


Figure 2.3: XP process cycle with iterative planning, team collaboration, and continuous delivery.

4. Lean Software Development

Lean comes from manufacturing and helps teams work faster by removing unnecessary items. The idea is to do only what adds value to the customer and avoid wasting time or effort.

Lean focuses on simple solutions, fast delivery, and learning from feedback. [25]

What All Agile Methods Have in Common ?

- Iterative development cycles
- Frequent delivery of functional products
- Customer collaboration throughout the project
- Emphasis on individuals and interactions over tools and processes

2.5 Justification for Using Scrum in Our Project

- **Frequent Client Involvement:** We have a client who is continuously involved, so we deliver working features regularly and get their feedback often. This helps us make sure the system meets their expectations at every stage.
- **Small, Manageable Deliverables:** We break down the project into small parts — for example, starting with grades and positions management — which allows us to focus on one feature at a time and deliver usable components quickly.
- **Flexibility to Adapt to Changes:** The Algerian public sector has evolving regulations and requirements. Scrum lets us adjust priorities and features after each sprint, ensuring the product stays relevant and compliant.
- **Early Problem Detection:** By delivering working increments every few weeks, we identify and fix issues early before they become costly or complex.
- **Clear Team Communication:** Daily stand-ups keep our team aligned and enable us to solve blockers fast, which is critical for a project with many stakeholders and detailed requirements.
- **Continuous Improvement:** After each sprint, we review what worked and what didn't, helping us improve our processes and product quality over time.
- **Delivering Real Value Quickly:** Each sprint ends with something functional that the client can review or start using, building trust and ensuring the project delivers tangible benefits throughout development.

2.5.1 Tasks Distribution

To keep our work organized and efficient, we followed the Scrum methodology, breaking the project into sprints, each focused on a specific module and structured into four phases:

- **Information Collection** : Meetings with the HR managers and staff of DJS Biskra to understand needs and gather detailed requirements.
- **UI Design** : Designing the interface screens using mockups .
- **Development** : Implementing the core functionalities of the module.
- **Testing** : Conducting functional, unit, and user acceptance tests to validate the work.

We also applied Clean Architecture to keep things modular and assign tasks based on our strengths. This helped us collaborate better and deliver higher quality results.

- **Rahma Yasmine** with her strong background in UI/UX design and front-end development, was responsible for designing the user interface screens through mockups and implementing the visual components of the application.
- **Khedidja mekhatria** took on both front-end and back-end development responsibilities. She ensured smooth integration between the interface and the system's core logic, effectively connecting the user experience with the server-side functionalities.
- **Tabbi Anneni Nesrine** specialized in back-end development and database management, was in charge of building the data layer and managing storage operations. She ensured data consistency, integrity, and secure interactions with the system's back end.

While each of us had our main responsibilities, we often helped each other when needed even during development to stay flexible and keep the project moving smoothly.

2.5.2 Scrum Sprints for Our Project

This section presents the different sprints we carried out during the development of our project.

Sprint 1: Position and Grade Management

Information Collection : Met with HR managers and staff to understand the structure of positions and grades in the public sector.

UI Design : Created forms and lists for adding, editing, deleting and viewing positions and grades.

Development : Implemented CRUD (Create, Read, Update, Delete) operations.

Testing : Ensured data is saved and retrieved accurately.

Sprint 2: Employee Registration

Information Collection : Gathered requirements on what data is needed to register a new employee.

UI Design : Designed the registration form with validation rules and creation steps .

Development : Developed the employee creation module with input validation.

Testing : Checked form submission, error handling, and duplicate prevention.

Sprint 3: Employee Personal Information management

Information Collection : done in sprint 2

UI Design : Built a clean and intuitive layout for viewing and editing the employee personal details.

Development :Developed add ,edit features for employee personal details .

Testing : Ensured data is saved and retrieved accurately.

Sprint 4: Employee Education management

Information Collection : done in sprint 2

UI Design : Built a clean and intuitive layout for viewing and editing and adding the employee education details.

Development : Developed add ,remove and edit features for Employee Education.

Testing : Ensured data is saved and retrieved accurately.

Sprint 5: Employee Career management

Information Collection :Defined how career movements (transfers, positions , etc..) are tracked.

UI Design : Built a clean and intuitive layout for viewing ,editing and adding the employee Career .

Development :Implemented logic to track changes in the employee careers .

Testing :Ensured data is saved and retrieved accurately.

Sprint 6: Promotion Management

Information Collection : Discussed the promotion rules, approval process, and criteria.

UI Design :Designed forms for promoting an employee , and viewing list of employees to promote .

Development : develop the logic of promoting an employee .

Testing : Verified promotion logic and system response.

Sprint 7:Management of promotions in employee profiles

Information Collection : Clarified how promotion history should appear in the employee profile.

UI Design : Integrated a view section in the profile to display promotion history.

Development : Fetched and presented relevant data for each employee.

Testing : Ensured accurate linking and display of promotion records.

Sprint 8: Regulation Management

Information Collection : Gathered legal documents and procedures that affect HR decisions.

UI Design : Built CRUD interfaces for managing legal texts and rules.

Development : Implemented the logic to store and update regulatory content according to the laws .

Testing : Evaluated content visibility and user experience for clarity, accessibility, and ease of use.

Sprint 9: Decision Generation and document Management

Information Collection : Discussed which documents (promotion decisions , leaves approval, etc ..) need auto-generation and to be retrieved from the app .

UI Design : Designed templates for documents storing and generations buttons .

Development : Developed dynamic generation, and saving of PDF documents and decisions.

Testing : Tested document formatting and data accuracy.

Sprint 10: Leave Management

Information Collection : Held meetings with HR staff to understand how leave requests, approvals, and employee leave balances are managed at DJS Biskra.

UI Design : Designed user-friendly forms for submitting a,d update leave of any type .

Development : Implemented features for submitting leave requests and tracking and updating leave balances for each employee.

Testing : Verified data validation, date overlapping, leave eligibility and leave plannings.

2.5.3 Sprint-Based Development Timeline

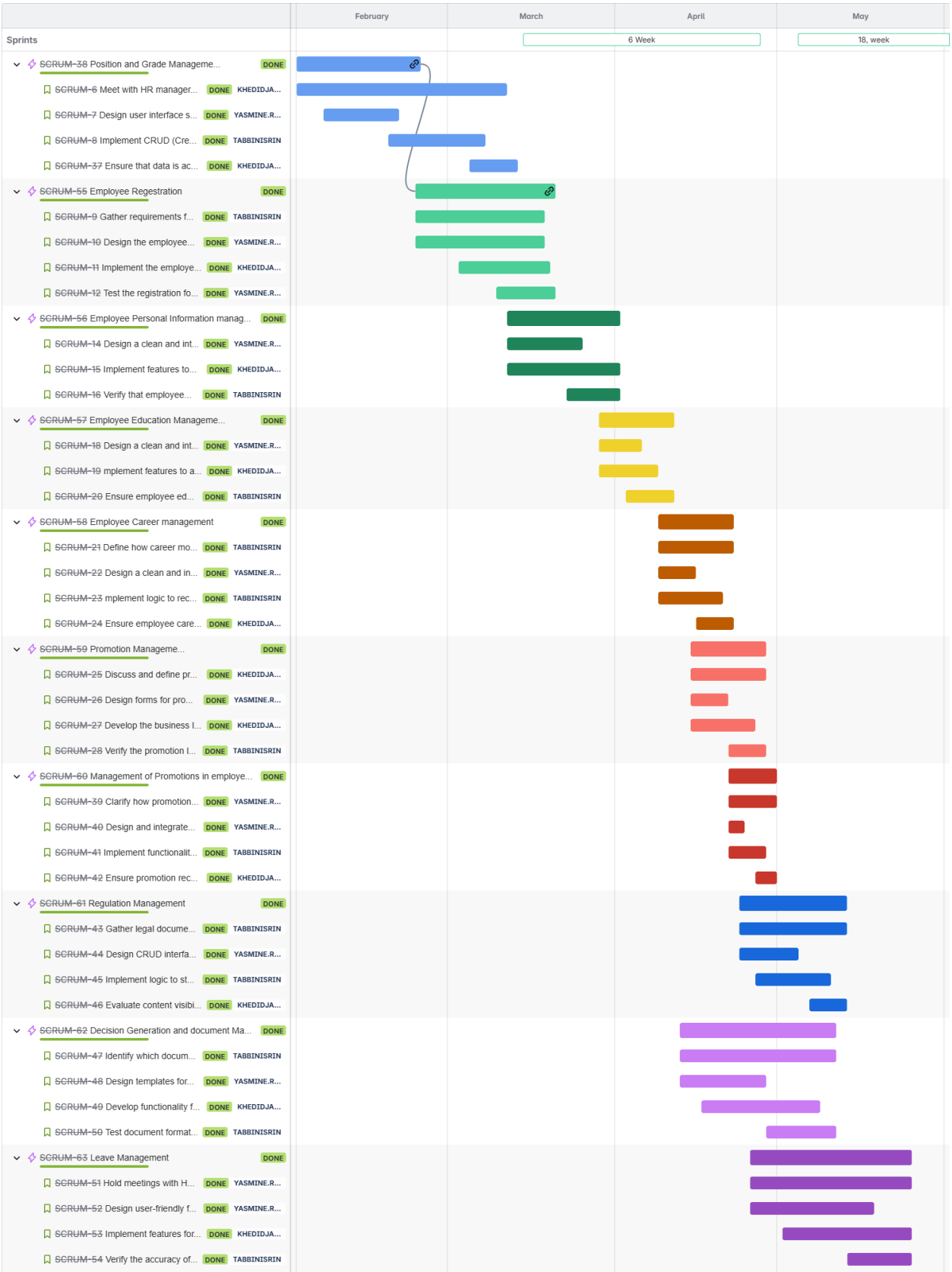


Figure 2.4: Gantt D Representing the Timeline and Task Distribution of the HRM Web Application Development

This figure gives a clear overview of the timeline and tasks for developing the HRM web app, broken down into eight sprints. Each sprint focused on a specific feature, starting with position and grade management, then moving on to employee registration, personal info, education, career tracking, promotions, and regulation management. It also shows how long each sprint took and how they overlap or follow each other, making it easy to see how the work was organized over time.

2.6 Conclusion

This chapter set the foundation for the HRM system in the public sector by exploring the real-world context, user needs, and technical requirements. Through close collaboration with the DJS Biskra staff and thoughtful analysis, we identified the major challenges and translated them into clear project specifications. Choosing Agile particularly Scrum enabled us to manage the project in a structured yet flexible way, adapting to feedback and ensuring continuous improvement. By combining practical insights with an organized development approach, we set the stage for building a solution that is not only functional but also meaningful and aligned with the users' expectations.

Chapter 3

System Design

3.1 Introduction

The design of a software system is a fundamental step in the development process. It involves determining how various parts of the system will interact, what components are needed, and how data will move through the application to meet the required functionality. In this chapter, we focus on essential modeling techniques such as the Unified Modeling Language (UML) and the Merise methodology. These tools play a central role in expressing the system's structure and behavior clearly, helping developers and stakeholders to better understand and communicate the underlying design before implementation begins.

3.2 Unified modeling language(UML)

UML is a standardized modeling language that uses diagrams to visualize, design, and document software systems.

3.2.1 Use Case Diagram

A use case diagram illustrates user interactions with the system, highlighting its main functionalities and clarifying requirements from the user's perspective.

3.2.1.1 Employees Management Use Case Diagram

This diagram (see Figure 3.1) highlights the role of HR manager in handling employee information efficiently within the system.. It defines the main functionalities such as adding, updating, viewing, and deleting employee records.

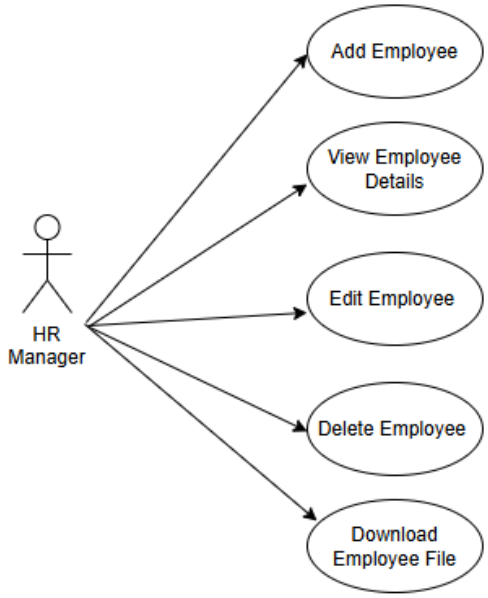


Figure 3.1: Employees Management use case diagram

3.2.1.2 Promotions Management Use Case Diagram

This use case diagram (see Figure 3.2) represents the process of managing employee promotions, whether through advancement or grade. The system supports automatic generation of promotion decisions and provides the ability to download the decision documents at any time.

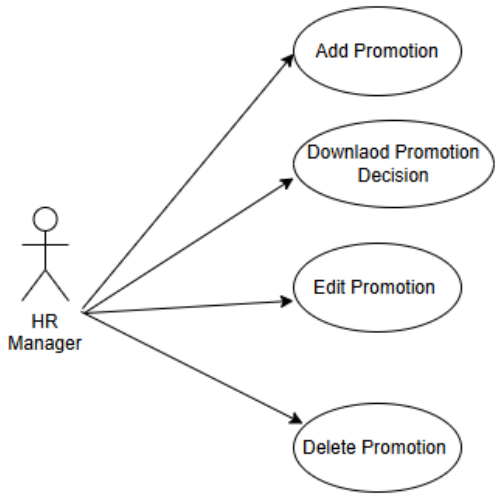


Figure 3.2: Promotions Management Use Case Diagram

3.2.1.3 Grades and Positions Management Use Case Diagram

This diagram (see Figure 3.3) illustrates the functionalities related to managing grades and positions within the institution. It allows Hr manager to view the full list of available grades and positions, as well as to access the list of employees assigned to each grade or position.

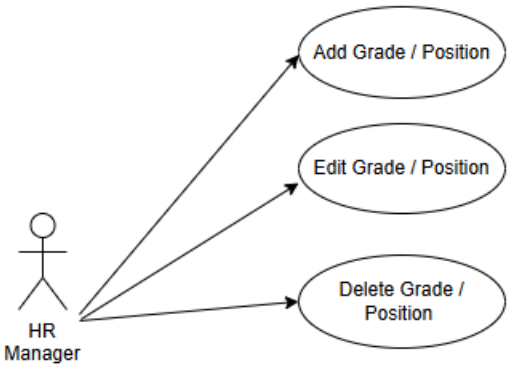


Figure 3.3: Grades and Positions Management use case diagram

3.2.1.4 Regulations Management Use Case Diagram

This diagram (see Figure 3.4) illustrates the management of regulations associated with various decisions, such as promotion decisions and others. It enables the HR manager to add, update, or delete regulations as needed. Additionally, the system allows users to view regulations linked to a selected decision type and sort them according to the order in which the decision is applied.

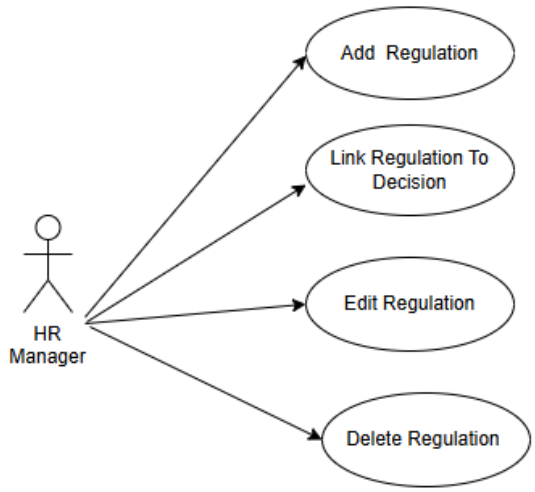


Figure 3.4: Regulations Management use case diagram

3.2.2 Sequence Diagram

A sequence diagram shows how system components interact over time by detailing the sequence and flow of exchanged messages for a specific process.

3.2.2.1 Add New Employee

This sequence (see Figure 3.5) illustrates how the manager can add a new employee by entering all necessary details related to the employee's profile.

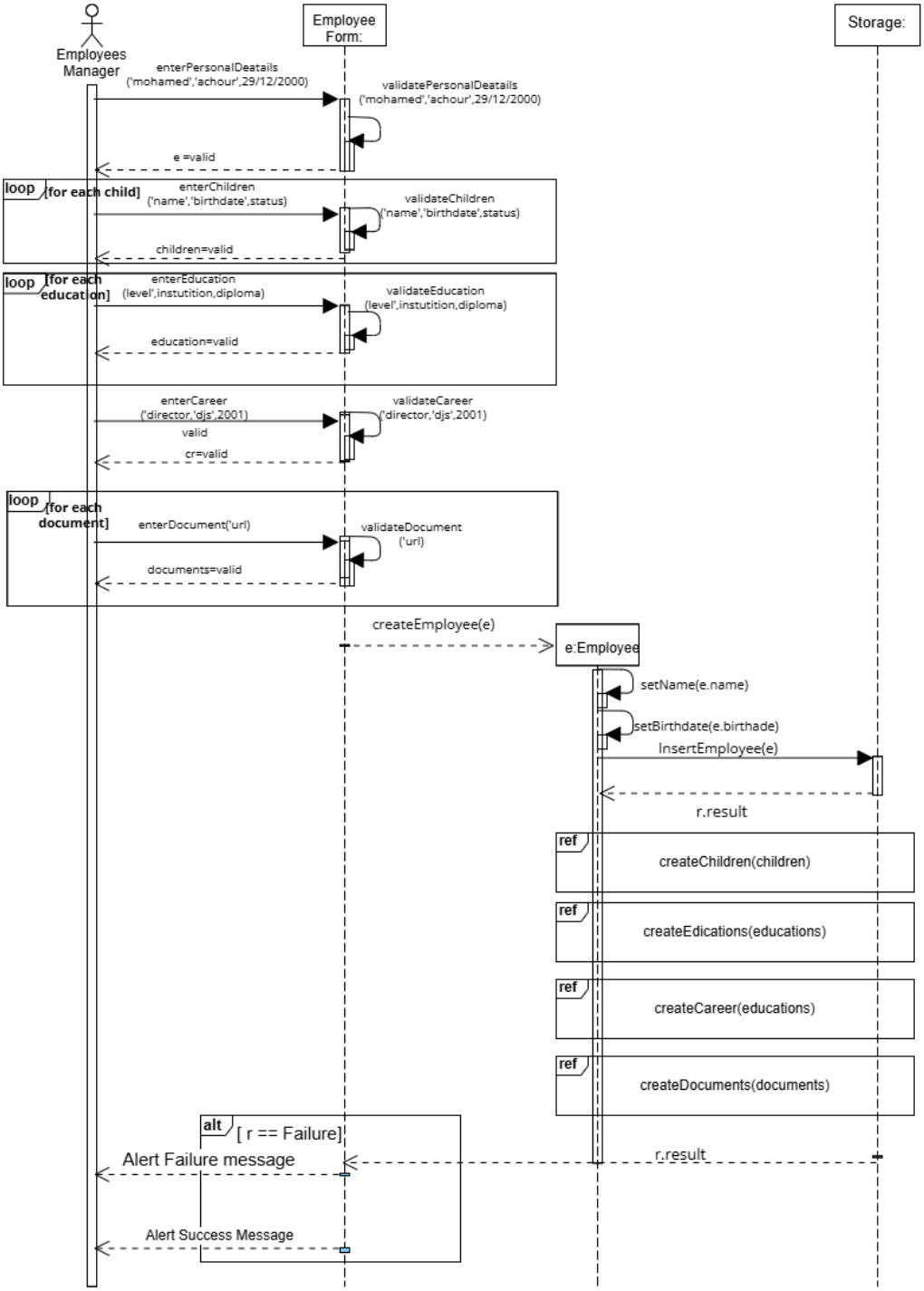
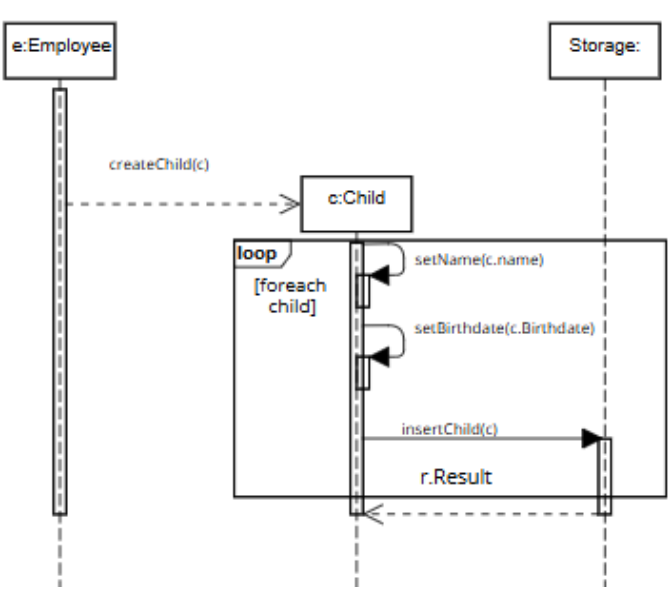
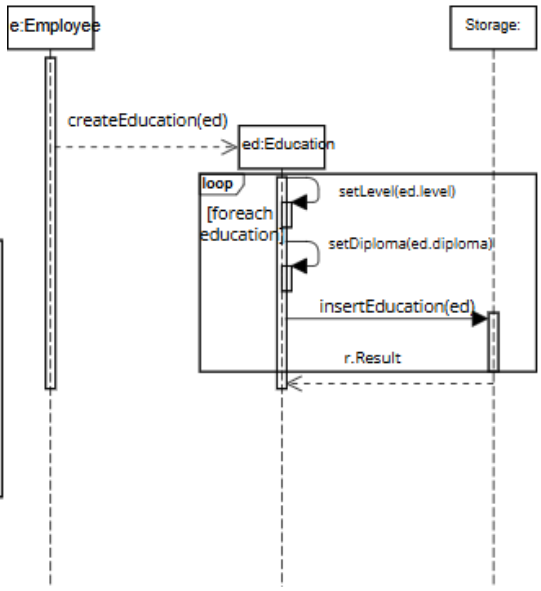


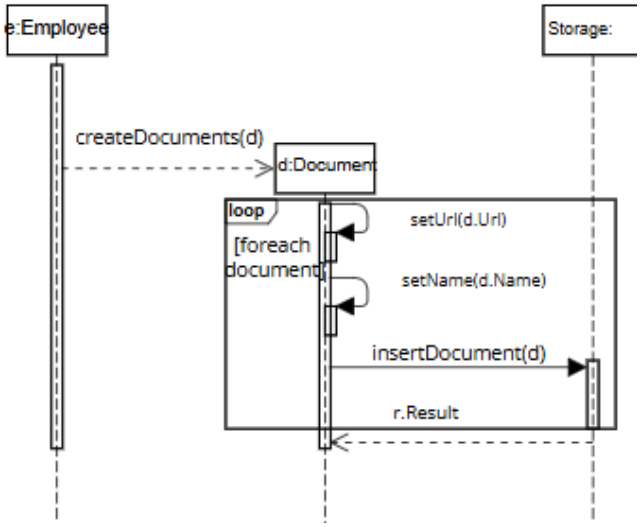
Figure 3.5: Add New Employee Sequence Diagram



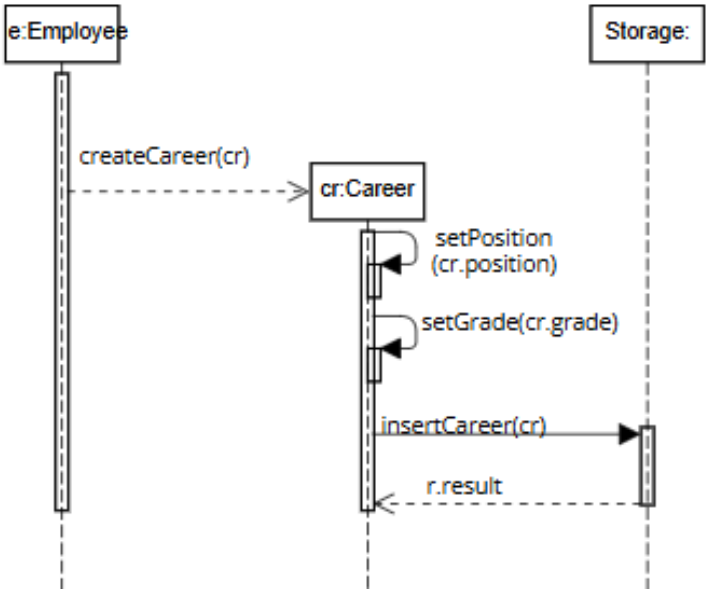
(a) Create Children Sequence Diagram



(b) Create Educations Sequence Diagram



(a) Create Documents Sequence Diagram



(b) Create Educations Sequence Diagram

3.2.2.2 Promote Employee

The promotions manager has the possibility to promote an employee in advancement and download the promotion decision, as illustrated in the diagram below (see Figure 3.8).

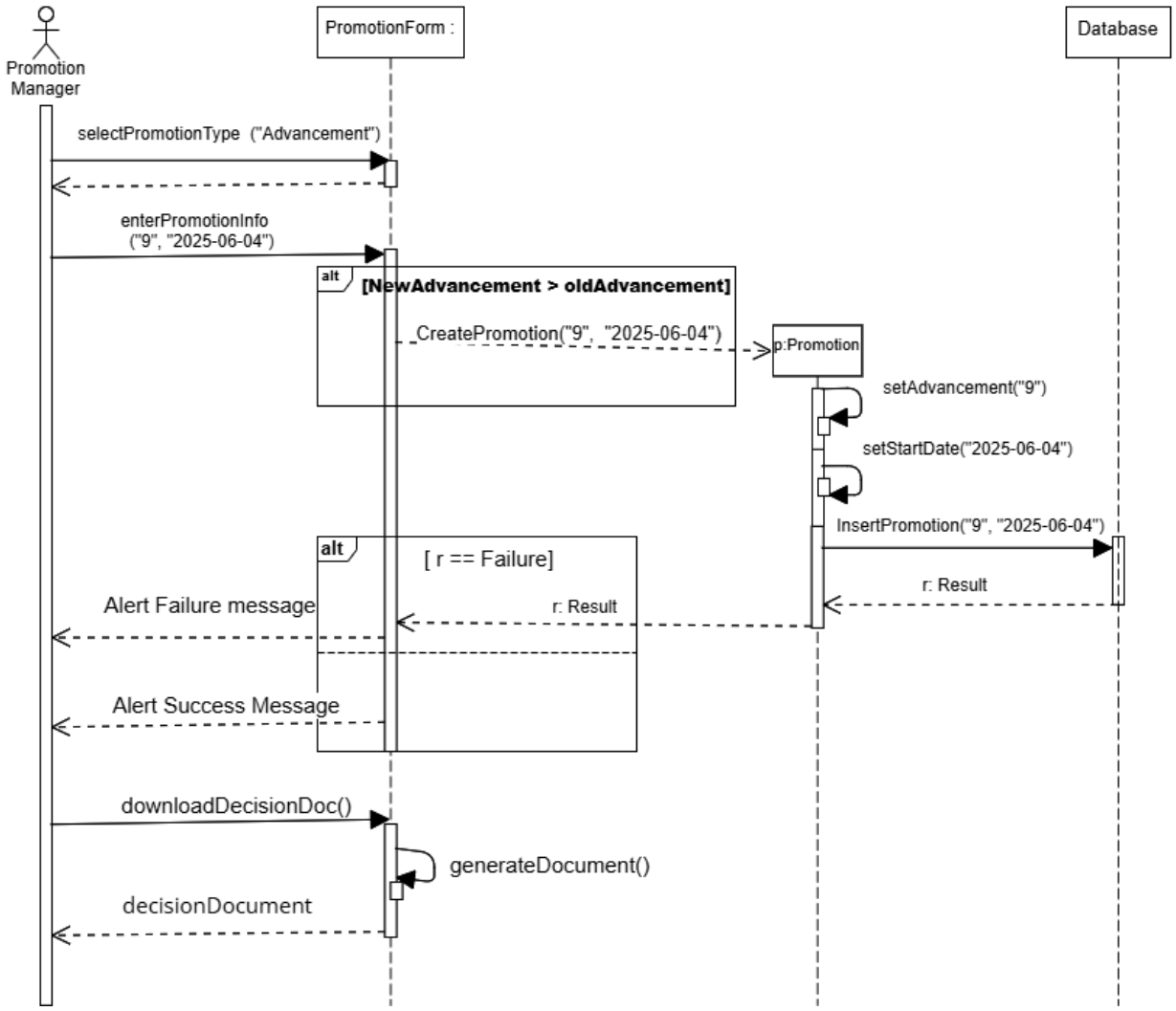


Figure 3.8: Promote Employee Sequence Diagram

3.2.3 Activity diagram

An activity diagram represents the dynamic flow of activities in a system, visualizing workflows, business processes, and conditional or parallel actions.

3.2.3.1 Add New Leave Activity Diagram

The following activity diagram (see Figure 3.9) illustrates the process of adding a new leave in the system. It describes the sequence of actions performed by the user and the system, including validations and conditional flows.

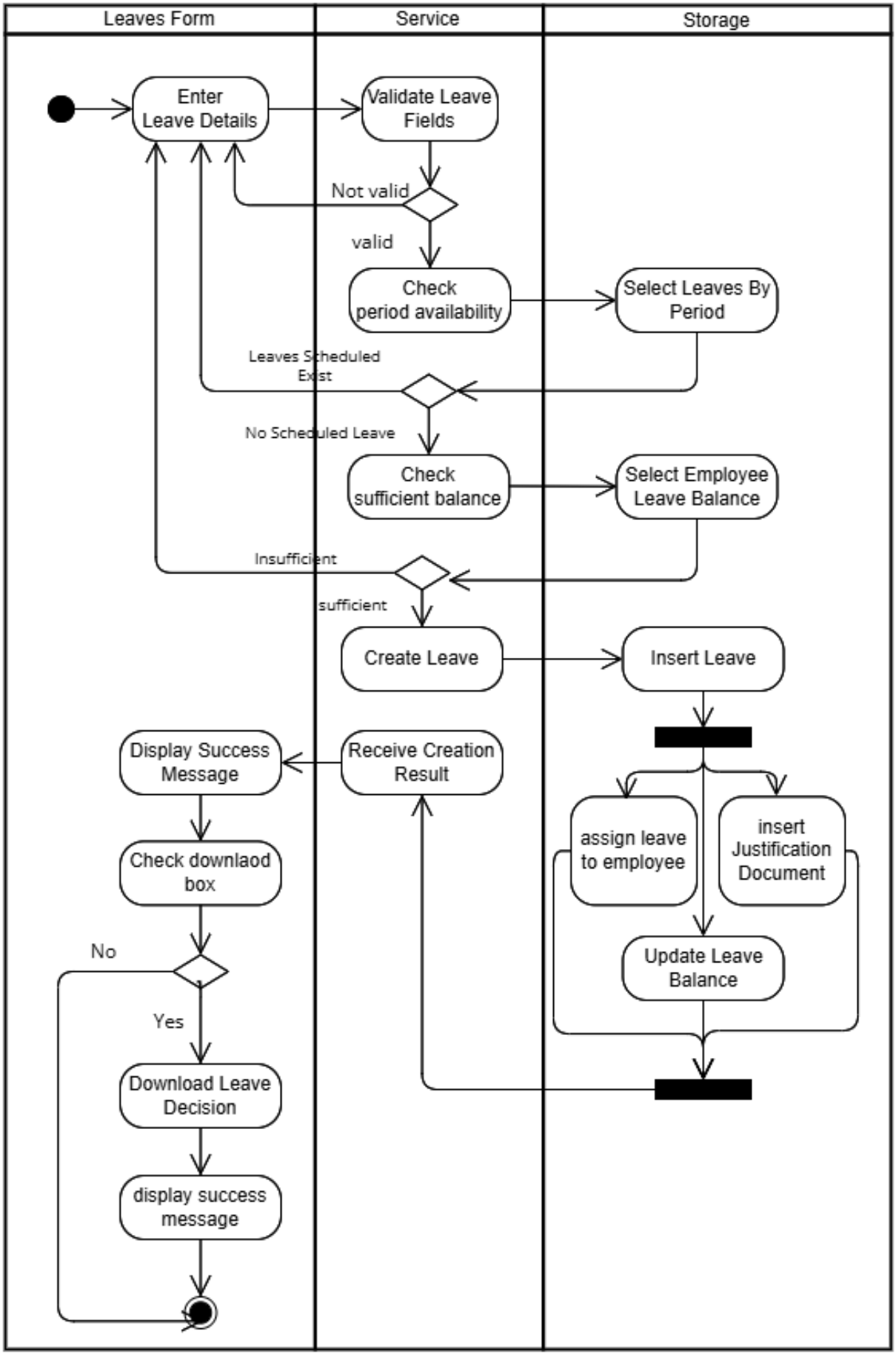


Figure 3.9: Add New Leave Activity Diagram

3.3 Merise Methodology

Merise is a general-purpose modeling methodology used in information systems development. It separates data and process modeling into conceptual, logical, and physical stages. This structured approach enables systematic analysis and design of databases and business processes.

3.3.1 MCD (Conceptual Data Model)

The purpose of this MCD (see Figure 3.10) is to provide a high-level, technology independent representation of our system's data to clarify requirements, structure information, and validate business rules.

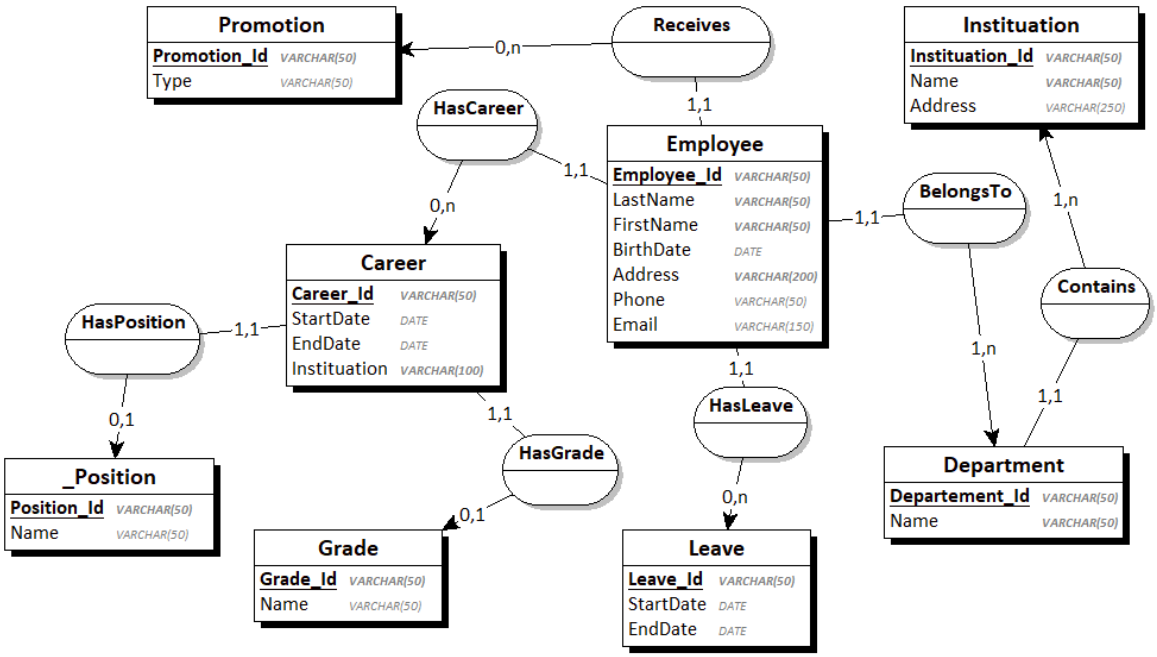


Figure 3.10: System MCD

3.3.2 MLD (Logical Data Model)

The purpose of the MLD is to transform the conceptual data model into a logical structure suitable for database implementation, defining tables, keys, and relationships while preparing for physical design.

- **Instituation** = (Instituation_Id: VARCHAR(50), Name: VARCHAR(50), Address: VARCHAR(250))
- **Department** = (Departement_Id: VARCHAR(50), Name: VARCHAR(50), #Instituation_Id: VARCHAR(50))
- **Grade** = (Grade_Id: VARCHAR(50), Name: VARCHAR(50))
- **Position** = (Position_Id: VARCHAR(50), Name: VARCHAR(50))

- **Employee** = (Employee_Id: VARCHAR(50), LastName: VARCHAR(50), FirstName: VARCHAR(50), BirthDate: DATE, Address: VARCHAR(200), Phone: VARCHAR(50), Email: VARCHAR(150), #Departement_Id: VARCHAR(50))
- **Career** = (Career_Id: VARCHAR(50), StartDate: DATE, EndDate: DATE, Instituation: VARCHAR(100), #Position_Id: VARCHAR(50), #Grade_Id: VARCHAR(50), #Employee_Id: VARCHAR(50))
- **Promotion** = (Promotion_Id: VARCHAR(50), Type: VARCHAR(50), #Employee_Id: VARCHAR(50))
- **Leave** = (Leave_Id: VARCHAR(50), StartDate: DATE, EndDate: DATE, #Employee_Id: VARCHAR(50))

3.3.3 MPD (Physical Data Model)

The purpose of the MPD is to transform the logical data model into a physical structure ready for implementation in a specific database management system. It defines the exact table structures, data types, constraints (such as primary and foreign keys), and storage details.

```
CREATE TABLE Institution(
   Institution_Id VARCHAR(50),
   Name VARCHAR(50) NOT NULL,
   Address VARCHAR(250),
   PRIMARY KEY(Institution_Id)
);

CREATE TABLE Department(
   Departement_Id VARCHAR(50),
   Name VARCHAR(50) NOT NULL,
   Institution_Id VARCHAR(50) NOT NULL,
   PRIMARY KEY(Departement_Id),
   FOREIGN KEY(Instituation_Id) REFERENCES
     Instituation(Institution_Id));
```

```
CREATE TABLE Employee(
   Employee_Id VARCHAR(50),
   LastName VARCHAR(50) NOT NULL,
   FirstName VARCHAR(50) NOT NULL,
   BirthDate DATE,
   Address VARCHAR(200) NOT NULL,
   Phone VARCHAR(50),
   Email VARCHAR(150),
   Departement_Id VARCHAR(50) NOT NULL,
   Leave_Id VARCHAR(50) NOT NULL,
   Career_Id VARCHAR(50) NOT NULL,
   Promotion_Id VARCHAR(50) NOT NULL,
   PRIMARY KEY(Employee_Id),
   FOREIGN KEY(Departement_Id) REFERENCES Department(Departement_Id));
```

```
CREATE TABLE Grade(
   Grade_Id VARCHAR(50),
   Name VARCHAR(50),PRIMARY KEY(Grade_Id));


CREATE TABLE _Position(
   Position_Id VARCHAR(50),
   Name VARCHAR(50),
   PRIMARY KEY(Position_Id)
);

CREATE TABLE Career(
   Career_Id VARCHAR(50), StartDate DATE,
   EndDate DATE,
   Instituation VARCHAR(100) NOT NULL,
   Position_Id VARCHAR(50) NOT NULL,
   Grade_Id VARCHAR(50) NOT NULL,
   PRIMARY KEY(Career_Id),
   UNIQUE(Position_Id),
   UNIQUE(Grade_Id),
   FOREIGN KEY(Position_Id) REFERENCES _Position(Position_Id),
   FOREIGN KEY(Grade_Id) REFERENCES Grade(Grade_Id)
   FOREIGN KEY (Employee_Id) REFERENCES Employee(Employee_Id));
```

```
CREATE TABLE Promotion(
   Promotion_Id VARCHAR(50),
   Type VARCHAR(50),
   Employee_Id VARCHAR(50),
  FOREIGN KEY (Employee_Id) REFERENCES Employee(Employee_Id)
);
CREATE TABLE Leave(
   Leave_Id VARCHAR(50),
   StartDate DATE,
   EndDate DATE,
   Employee_Id VARCHAR(50),
   PRIMARY KEY(Leave_Id),
   FOREIGN KEY (Employee_Id) REFERENCES Employee(Employee_Id)
);
```

3.4 Conclusion

System design plays a vital role in the development of effective information systems. Methodologies such as **UML** and **Merise** offer structured approaches for modeling both behavioral and structural elements of a system. By promoting clarity, minimizing misunderstandings, and guiding implementation, these tools support consistent communication among stakeholders and provide lasting documentation throughout the software development lifecycle.

Chapter 4

System Implementation

4.1 Introduction

This chapter presents the implementation phase of our web application, **KYN-HR**. Following thorough planning, analysis, and system design, development marks the transition from concept to working solution. Here, we outline the technologies adopted, the development environment, and the essential tools that supported the system's construction. We also explain the strategies used to ensure functionality, reliability, and alignment with the initial requirements. Finally, the user interfaces are presented to demonstrate how the application responds to the identified challenges and meets the practical needs of end users.

4.2 Development Environment

In this section, we present the set of tools and technologies used throughout the various phases of the application's development. These tools supported activities such as UI/UX design, front-end and back-end development, version control and testing. Each subsection below details the specific tools employed in its respective phase.

4.2.1 Used UI/UX Design Tool

- **Figma Design** : is a cloud design platform to create, share, and test digital product interfaces like websites and mobile applications. It enables real-time collaboration between designers, product managers, writers, and developers so that all the stakeholders can contribute and give feedback in real-time [26].

Figure 4.1: Figma Logo

4.2.2 Used Front-end language and Framework

- **Blazor** : is a free and open-source web framework developed by Microsoft that enables developers to build interactive web UIs using `C#` instead of JavaScript. It supports both client-side (via WebAssembly) and server-side hosting models. Blazor uses Razor syntax to create reusable web UI components with `C#` and HTML [27].

Figure 4.2: Blazor logo

- **HTML**: HTML (HyperText Markup Language) is the standard language used to create and structure content on the web. It defines the meaning and structure of web elements such as headings, paragraphs, and links [28].
- **CSS**: is a language used to describe the presentation and layout of web pages. It controls design aspects such as colors, fonts, spacing, and positioning of HTML elements [29].
- **JavaScript**: is a versatile, high-level programming language primarily used to create interactive and dynamic content on websites. It runs in the browser and enables features like animations, form validation, and real-time updates [30].

Figure 4.3: HTML, CSS, and JavaScript logos

- **Tailwind CSS**: is a free and open-source utility-first CSS framework used for front-end web development. Rather than offering pre-designed components, Tailwind provides low-level utility classes that allow developers to build custom designs directly in their HTML. [31]

Figure 4.4: Tailwind Logo

4.2.3 Used Database System

- **Microsoft SQL Server** : is a relational database management system (RDBMS) that supports storing and retrieving data using Transact-SQL (T-SQL). It provides

features like data integrity, security, and scalability. To interact with the database, **SQL Server Management Studio (SSMS)** is commonly used—a graphical interface that enables developers and database administrators to manage databases, write queries, and monitor performance. This combination is widely used in enterprise environments [32, 33].

Figure 4.5: SQL Server logo

4.2.4 Database Hosting Platform

- **SmarterASP.NET**: is a web hosting provider that specializes in ASP.NET and .NET Core applications. It supports SQL Server, MySQL, and offers various hosting plans including shared, VPS, and dedicated hosting, tailored for developers using Microsoft technologies [34]. We used SmarterASP.NET during the development phase to facilitate remote access and collaboration.

Figure 4.6: Smarter Asp Logo

4.2.5 Used Back-end Frameworks

- **.NET**: is a free, cross-platform, open-source developer platform used to build diverse applications, including web, desktop, mobile, cloud, and microservices. In this project, we utilized `NET 9 with C#` as the primary programming language, leveraging its strong typing, object-oriented features, [35] and rich tooling support within the Visual Studio environment 4.8.

Figure 4.7: `Microsoft.Net Logo`

4.2.6 Used Integrated Development Environment

- **Visual Studio 2022** : is a robust development environment designed to support the entire software development lifecycle in one platform. It serves as a full-featured IDE

that enables you to write, edit, debug, build, and deploy applications. With built-in compilers, code suggestions, version control integration, and a wide range of extensions [36].

Figure 4.8: Visual Studio 2022 Logo

4.2.7 Version Control System

- **GitHub** : is a cloud-based platform that allows developers to host, manage, and collaborate on code projects. It uses Git to track changes, support version control, and enable collaborative workflows. Teams can review, share, and improve code together without disrupting each other's work. [37].

Figure 4.9: Github Logo

4.3 Test and Validation Strategy

To ensure the quality and reliability of the developed application, we begin by clarifying the difference between testing and validation. Understanding their distinct roles allows us to apply the appropriate strategies for each. Following this clarification, we detail the specific approaches we adopted to carry out both testing and validation throughout the development process.

4.4 Test vs Validation

It is essential to distinguish between testing and validation, as the two are often mistakenly used interchangeably. While both are critical components of the system development lifecycle, they serve different purposes.

Aspect	Testing	Validation
Definition	Process of evaluating a system or component to determine if it satisfies specified requirements.	Process of evaluating software during or at the end of development to ensure it meets business needs.
Focus	Checks internal consistency and correctness.	Checks usability, completeness, and if the system meets user expectations.
Who performs it	Mainly developers and testers.	Users, stakeholders, and testers.
Tools used	MSTest (.NET), Selenium, manual unit tests.	Usability tests, feedback forms, manual system-level checks.
When performed	Throughout development cycle, especially after builds.	After every deliverable of the software

Table 4.1: Comparison between Testing and Validation

In conclusion, while both testing and validation aim to ensure software quality, they serve distinct purposes: testing focuses on identifying defects during development, whereas validation ensures that the final product meets user expectations and requirements. Together, they form a complementary process that ensures both functional correctness and user satisfaction.

4.5 Test Strategy

Our project adopts two primary testing approaches: **unit testing** to verify individual components and **UI/UX** testing to assess the user interface and experience.

4.5.1 Unit testing

Unit testing was performed by the development team after each task. Initially, units were tested manually to establish the right behavior. Once verified, automated tests were executed using the MSTest framework in.NET to enable consistent and repeatable validation of functionality.

Figure 4.10: Unit testing `C#` with MSTest

4.5.2 UI/UX Testing

The user interface was tested both manually and using automation. Manual testing helped assess usability and visual consistency, while Selenium IDE was used to automate common interaction scenarios, ensuring the front-end behaved as expected across different user actions.

Figure 4.11: Test With Selenium IDE

4.6 Validation Strategy

Validation is first performed by the development team and later in collaboration with clients to ensure the system meets real-world requirements.

4.6.1 Data Validation and Business Rule Validation

We apply thorough data validation to ensure the accuracy and completeness of user inputs. For instance, certain fields like **Position Name**, **Grade Name**, are mandatory and cannot be left empty. We also validate input formats, such as ensuring that email addresses follow a valid pattern (e.g., `namedomain.com`) and that date fields are in the correct format (e.g., `DD/MM/YYYY`).

In addition to data validation, we enforce business rules to maintain logical consistency. For example, the system prevents:

- The duplication of a position name within the same institution.
  - The duplication of an education level within the same employee's record.
  - A promotion in advancement if the proposed advancement is lower than or equal to the employee's current advancement.

4.6.2 User Acceptance Testing (UAT) and Feedback

Beyond internal validation, we placed significant emphasis on real user feedback. To achieve this, we deployed the system in several real-world environments, including the **Directorate of Youth and Sports (DJS)**, the **Multi-Sports Complex Office (Biskra Stadium)**, and the **University of Biskra**. These installations allowed us to conduct thorough User Acceptance Testing (UAT) under real working conditions.

We closely accompanied users during their testing sessions, providing assistance when necessary and documenting their observations and comments in real time. This approach enabled us to gather both spontaneous feedback and structured evaluations.

In addition to direct interaction, we developed and distributed a structured satisfaction survey designed to serve two purposes: first, to help users evaluate their experience with the system, and second, to assist our team in analyzing customer satisfaction. The survey covered key aspects such as:

- **User Interface:** visual design, clarity, and intuitiveness.
- **Usability:** ease of use, task completion without support.
- **Task Execution:** ability to accomplish day-to-day operations.
- **Feedback and Issues:** sections dedicated to reporting problems, submitting suggestions, and providing general comments.

To explore the structure and content of the evaluation form we used, please refer to the following link:
https://forms.gle/X6s3V4hF4RYE3g5Y9

4.6.3 Sample Feedback

To further illustrate how users interacted with the evaluation form, we include below a sample of actual feedback responses. These reflect both positive impressions and suggestions for improvement, gathered directly from end-users during field testing.

Figure 4.12: Sample Users Feedback

4.7 Presentation of the application Interfaces

4.7.1 Employees List Page

This page represents the list of employees, displaying their main information and providing management functionalities.

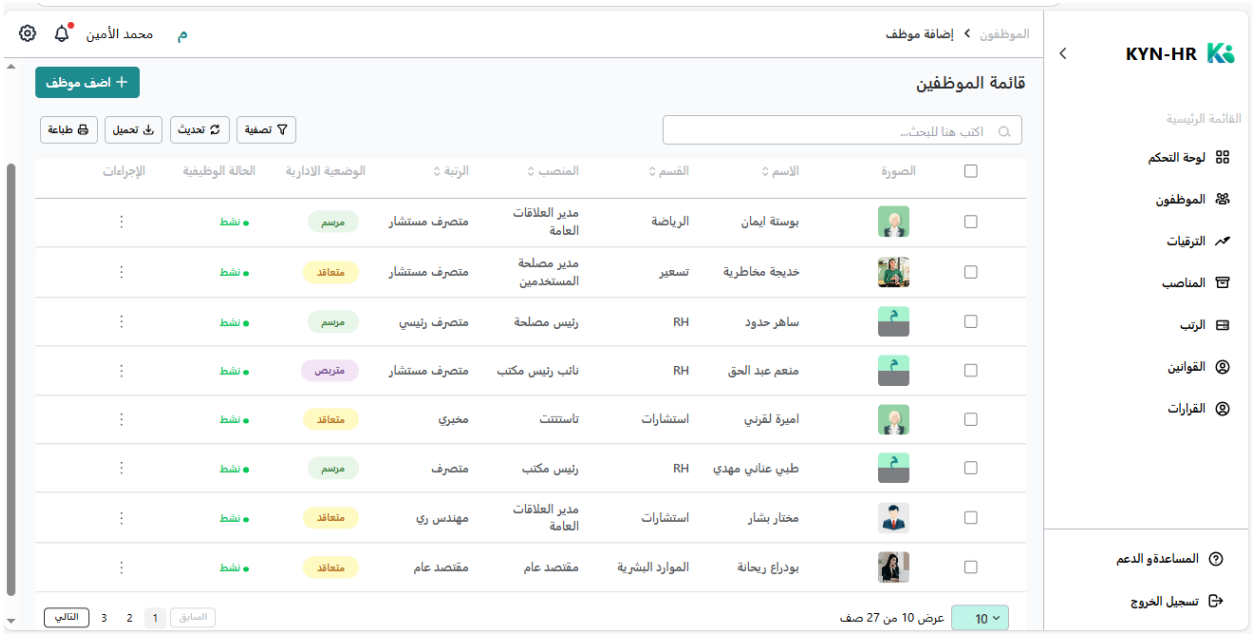


Figure 4.13: Employees List Page

4.7.2 Add New Employee Page

This page allows the manager to enter all employee-related details step by step.

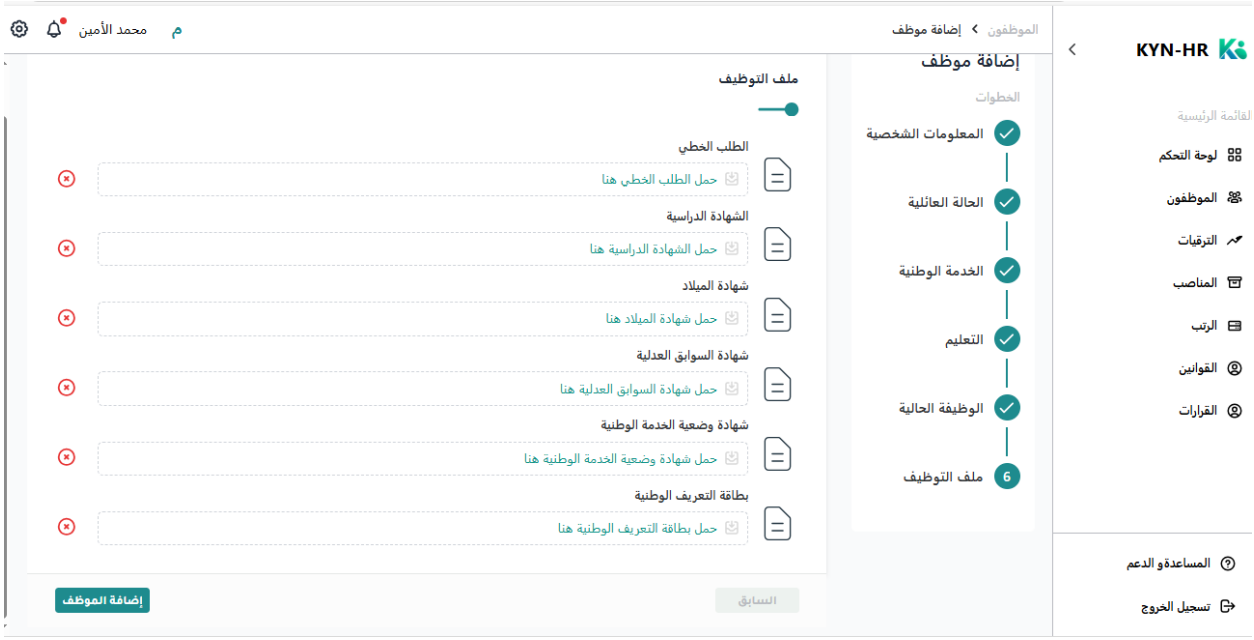


Figure 4.14: Add New Employee Page

4.7.3 Employee Details Page

This page allows navigation through all employee details and provides management options.

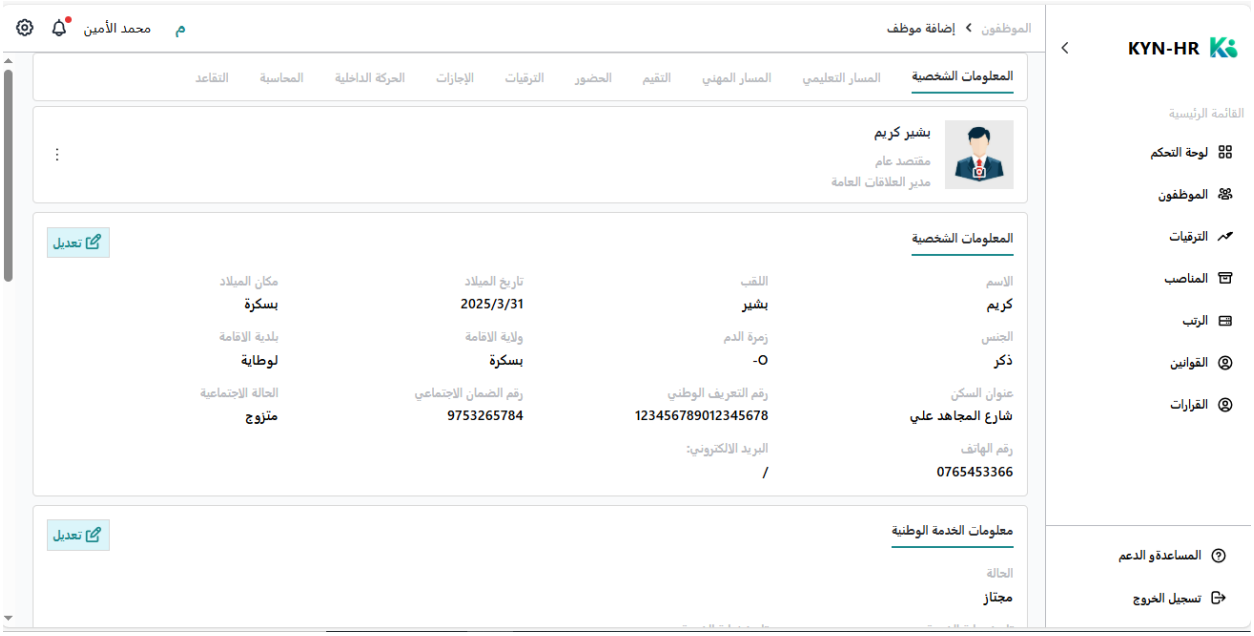


Figure 4.15: Employee Details Page

4.7.4 Leave Management Pages(Add/Update)

These pages are designed for managing employee leaves by allowing the manager to add a new leave or update an existing one.

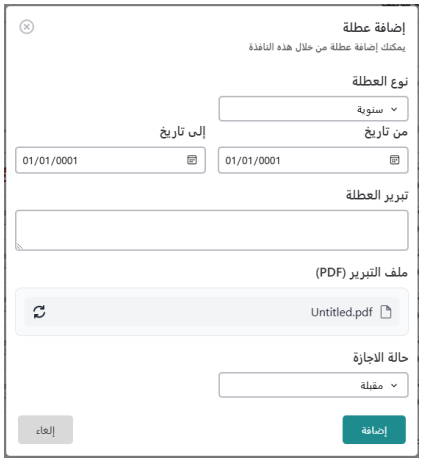


Figure 4.16: Add Leave Page

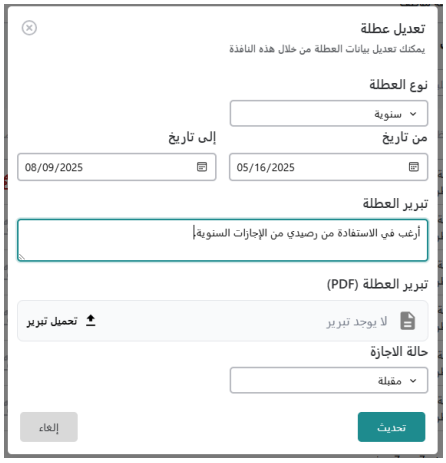


Figure 4.17: Update Leave Page

4.7.5 Regulations Management Page

This page facilitates the management of internal regulations associated with decisions, enabling manager to order them as needed.

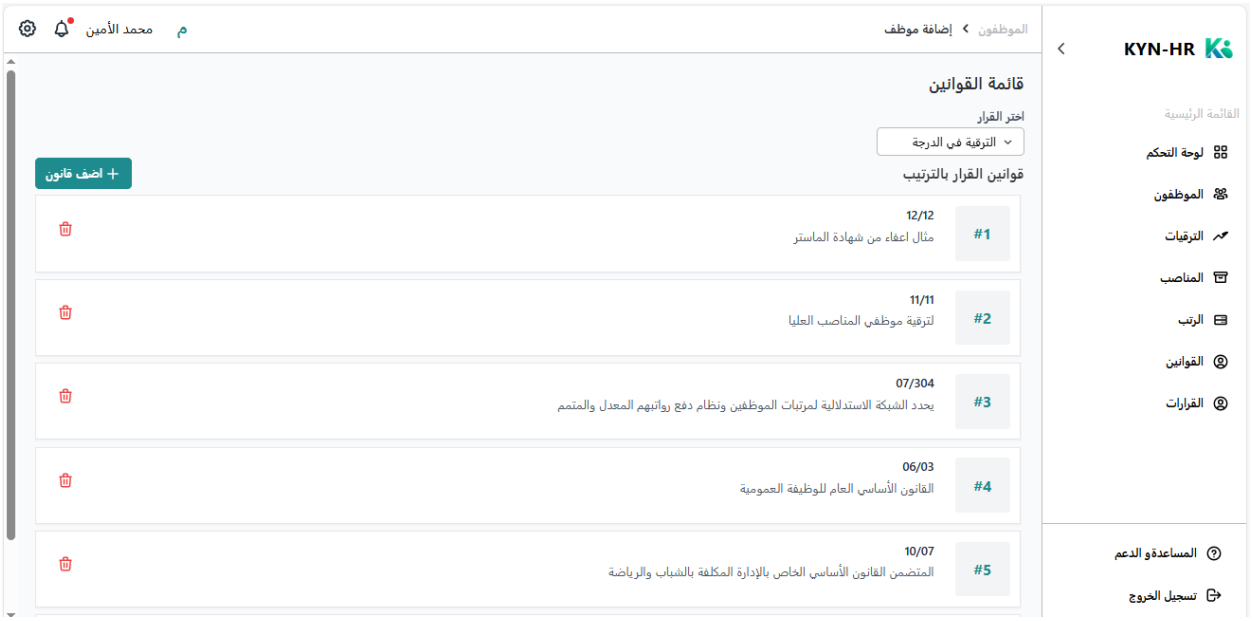


Figure 4.18: Regulations Management Page

4.7.6 Decision Generation and Download

The system allows the generation of an official decision document. When the user clicks on the **Download Decision** button, the system compiles the relevant information, generates a formal document based on a predefined template, and downloads it automatically to the user's device.

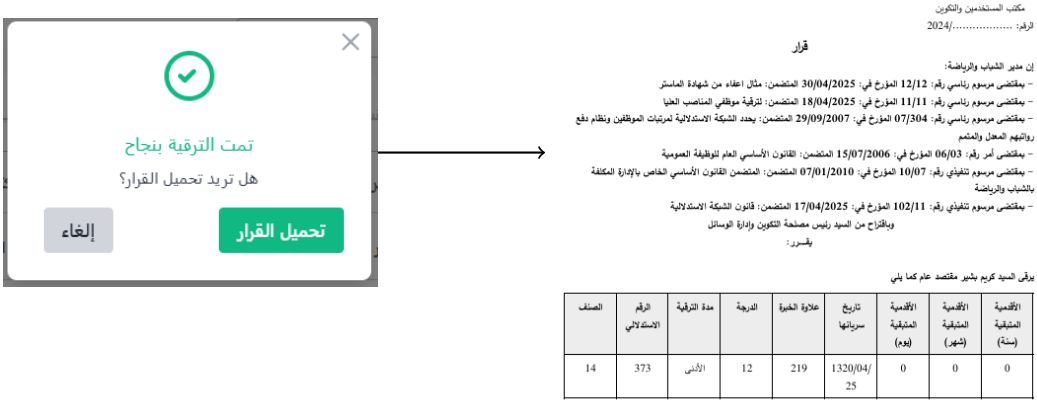


Figure 4.19: From Download Page to Generated Decision Document

4.7.7 Employee Document Management

All employee documents are centralized in one place within the system (Employee Profile Page, see Figure 4.15). Managers can upload, print, and download any document at any time with ease. Additionally, the system allows downloading all employee documents at once into a single folder for streamlined access and archiving.

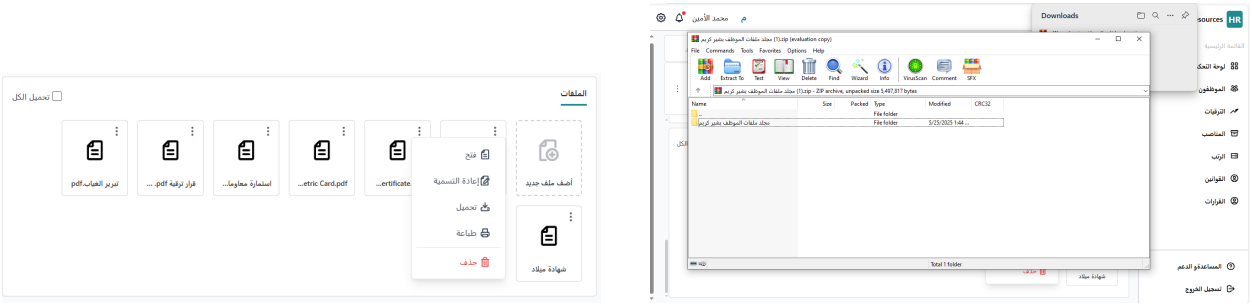


Figure 4.20: Employee Documents Management and Bulk Download

4.8 Conclusion

The implementation chapter illustrated how the proposed HRM system was developed using the selected technologies and tools. Each component was built according to the design and specifications defined in earlier phases. The application interfaces were carefully crafted to provide a clear and intuitive user experience. Testing and validation ensured that the system functions correctly and meets user expectations. This phase reflects the successful transition from design to a fully working application, ready to be used in a real administrative environment.

General Conclusion

This project set out to design and implement a Human Resource Management (HRM) system built specifically for Algerian public sector institutions. The primary motivation was to address the persistent challenges these organizations face—namely, reliance on paperwork, difficulty in tracking employee careers, and delays in administrative procedures. By building a localized software solution that complies with Algerian labor laws, the project aimed to modernize and streamline HR operations while maintaining strict data privacy and legal integrity.

Throughout the development process, several key functionalities were successfully implemented. The system supports comprehensive employee file management, career tracking, leave and promotion management, and the generation of administrative decisions based on a legal rules engine. Built as a locally installed web application, it ensures that sensitive employee data remains secure within the institution, addressing concerns commonly raised about cloud-based solutions in the public sector.

The resulting system offers clear benefits: it improves efficiency by automating repetitive tasks, enhances transparency by standardizing HR processes, and supports better decision-making through organized employee data and regulation-based workflows. Most importantly, it brings HR practices in line with national legal requirements, which is often lacking in many existing local solutions.

Nonetheless, the project has certain limitations. For example, it does not yet include modules for payroll integration, biometric attendance, or advanced analytics features that could add significant value in the future. Moreover, while the system is designed for local deployment, a future hybrid model could be explored to balance security with scalability. In the future, further improvements can be made by incorporating artificial intelligence to assist in administrative decision-making, generating predictive reports.

In conclusion, this project contributes not only to the digital transformation of HR management in Algeria but also highlights the importance of aligning software development with legal, organizational, and societal contexts. It reflects a growing need for tailored technological solutions that respect national priorities while driving institutional progress.

Bibliography

[1] N. Wilton. *An Introduction to Human Resource Management*. SAGE Publications, 2025. ISBN: 9781036202224. URL: https://books.google.dz/books?id=sJMeEQAAQBAJ.

[2] Ben Atta Allah El Almi. "Human Resource Management in Public Institutions and Administrations in Algeria: A Reading of the General Statute of the Civil Service 06/03." In: *Al-Naqid Journal for Political Studies* 7.1 (Apr. 2023). Published on April 1, 2023, pp. 482–493.

[3] A. S. Syed Navaz et al. "Human Resource Management System." In: *International Organization of Scientific Research - Journal of Computer Engineering* 8 (Sept. 2013), 62–71, (Impact Factor : 1.686).

[4] Oracle. *HRIS vs. HRMS vs. HCM: What is the Difference?* Accessed: 2025-05-16. 2024. URL: https://www.oracle.com/human-capital-management/hrms/hris-hrms-hcm/.

[5] TechnologyAdvice. *Types of Human Resource Information Systems (HRIS)*. Accessed: 2025-05-16. 2025. URL: https://technologyadvice.com/blog/human-resources/types-of-hris-software/.

[6] V. Antony Joe Raja and S. Balasubramanian. "E-HRM in Software Organizations." In: *International Journal of Management Research and Development (IJMRD)* 1.1 (2011). Posted Online: 11 Mar 2020, pp. 1–5.

[7] Oracle Corporation. *Oracle Human Capital Management (HCM)*. Accessed: 2025-05-18. 2025. URL: https://www.oracle.com/za/human-capital-management/?utm_source=chatgpt.com.

[8] BambooHR LLC. *BambooHR - Human Resources Software for Small and Medium Businesses*. Accessed: 2025-05-18. 2025. URL: https://www.bamboohr.com/.

[9] Solutions Intelligentes Informatiques SII. *OnyxRH Human Capital Management System (HCM)*. https://www.solutionsinformatiques.dz/Human-Capital-Systems. Accessed: 2025-05-18. n.d.

[10] Odoo S.A. *Employees App — Odoo*. Accessed: 2025-05-18. 2025. URL: https://www.odoo.com/app/employees.

[11] *windows desktop apps and privacy*. Accessed: 2025-05-08. Microsoft. URL: https://support.microsoft.com/en-us/windows/windows-desktop-apps-and-privacy-8b3b13bc-d8ff-5460-8423-7d5d5c1f6665.

[12] Rohit Manglik. *Mobile Application Development*. English. Digital Edition, EduGorilla Prep Experts. n.p.: EduGorilla Publication, 2024, p. 124. ISBN: 9789370937123, 9370937129.

[13] Prabhu TL. *Mobile App Development: From Concept to Creation – Mastering the Art of Building Engaging and Scalable Apps*. English. Digital Edition. n.p.: NestFame Creations Pvt Ltd., Dec. 2024, p. 82.

[14] Rosario Moscato. *Web App Development Made Simple with Streamlit: A Web Developer's Guide to Effortless Web App Development, Deployment, and Scalability*. English. Digital edition. Birmingham, UK: Packt Publishing, Feb. 2024, p. 350. ISBN: 9781835085936, 1835085938.

[15] *N-tier architecture style*. Accessed: 2025-05-08. microsoft. URL: https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/n-tier.

[16] *Client-Server Model*. Accessed: 2025-05-12. geeksforgeeks. URL: https://www.geeksforgeeks.org/client-server-model/.

[17] *Microservice Architecture pattern*. Accessed: 2025-05-9. microservices. URL: https://microservices.io/patterns/microservices.html.

[18] Diogo Ribeiro. *Clean Architecture: Principles and Practices for Sustainable Code*. Accessed: 2025-05-17. LinkedIn. Jan. 2024. URL: https://www.linkedin.com/pulse/clean-architecture-principles-practices-sustainable-ribeiro-da-silva-retvf.

[19] *Direction de la jeunesse et des sports (DJS)*. Accessed: 2025-05-10. Ministry of Youth and Sports of Algeria. Nov. 2019. URL: https://www.mjs.gov.dz/index.php/fr/formation-et-emploi-3/djs/1973-direction-de-la-jeunesse-et-des-sports-djs.

[20] *What is Agile?* Accessed: 2025-05-16. agilealliance. URL: https://www.agilealliance.org/agile101/.

[21] Sergey V. Zykov, Alexander Gromoff, and Nikolay S. Kazantsev. *Software Engineering for Enterprise System Agility: Emerging Research and Opportunities*. Digital Edition. IGI Global, 2018, p. 218. ISBN: 9781522555902, 1522555900. URL: https://www.google.dz/books/edition/Software_Engineering_for_Enterprise_Syst/WtJeDwAAQBAJ?hl=fr&gbpv=0.

[22] Cherie C. Trumbach, Kenneth R. Walsh, and Sathiadev Mahesh, eds. *Agile Scrum Implementation and Its Long-Term Impact on Organizations*. Digital Edition. IGI Global, 2020, p. 268. ISBN: 9781799848868.

[23] *Kanban Methodology — A Visual System for Managing Work*. Accessed: 2025-05-16. startinfinity. URL: https://startinfinity.com/project-management-methodologies/kanban.

[24] Don Rosenberg, Doug Rosenberg, and Matt Stephens. *Extreme Programming Refactored: The Case Against XP*. Digital Edition. Apress, Aug. 13, 2003, p. 432. ISBN: 9781430208105.

[25] *Lean, Agile and Scrum: A Simple Guide [2024]*. Accessed: 2025-05-16. brainhub. URL: https://brainhub.eu/library/differences-lean-agile-scrum.

[26] Figma. *What is Figma?* Accessed: 2025-05-06. 2024. URL: https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma.

[27] Microsoft Corporation. *ASP.NET Core Blazor*. Accessed on May 7, 2025. 2025. URL: https://learn.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-8.0.

[28] Mozilla Developer Network. *HTML - MDN Web Docs*. Accessed: 2025-05-09. 2024. URL: https://developer.mozilla.org/en-US/docs/Web/HTML.

[29] Mozilla Developer Network. *CSS - MDN Web Docs*. Accessed: 2025-05-09. 2024. URL: https://developer.mozilla.org/en-US/docs/Web/CSS.

[30] Mozilla Developer Network. *JavaScript - MDN Web Docs*. Accessed: 2025-05-09. 2024. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript.

[31] Tailwind Labs. *Tailwind CSS - Documentation*. Accessed on May 7, 2025. 2025. URL: https://tailwindcss.com/docs.

[32] Microsoft Corporation. *What is SQL Server - SQL Server Documentation*. Accessed on May 7, 2025. 2025. URL: https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16.

[33] Microsoft Corporation. *SQL Server Management Studio (SSMS)*. Accessed on May 7, 2025. 2025. URL: https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms.

[34] SmarterASP.NET. *ASP.NET Web Hosting - SmarterASP.NET*. https://www.smarterasp.net/asp.net_hosting. Accessed: 2025-05-06. 2025.

[35] Microsoft Corporation. *What is .NET? - .NET Documentation*. Accessed on May 7, 2025. 2025. URL: https://learn.microsoft.com/en-us/dotnet/core/introduction.

[36] Microsoft Corporation. *What is Visual Studio?* Accessed on May 7, 2025. 2025. URL: https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022.

[37] GitHub, Inc. *About GitHub - Documentation*. Accessed on May 7, 2025. 2025. URL: https://docs.github.com/en/get-started/start-your-journey/about-github-and-git.