Design and Development of Software for HR Management

Project proposed by :

Mrs HAMOUMA OUARDA :
Defended on :04 /06/2025   **Presented by :** Drouiche Loubna
                                                                       Khelifi Ilham
                                                                       Mahdjoub Ines
                                                                       Zeghar Meroua
                                                                       Aouadi Asma

In front of the jury composed of:

Mr HAMDAH, MOHAMED, President
Mrs MERABET FATIMA, Member

Group n° : 29 / 2025

Table of Contents

Table of Figures

List of Tables

Introduction

In the era of digital transformation, the dematerialization and automation of business processes are key drivers for enhancing the performance and transparency of public institutions. In Algeria, human resources management (HRM) within many organizations still heavily relies on traditional office tools, or even manual processing, leading to significant inefficiencies in terms of reliability, traceability, and productivity.

In this context, our final-year project is part of a broader effort to modernize and streamline HR processes through the development of a desktop application dedicated to the National Observatory for Education and Training (ONEF). The main objective is to design a robust, secure, and user-friendly software solution that enables integrated management of various HR functions.

This thesis presents the different stages of the software development life cycle carried out, from the functional analysis of needs to the technical implementation, including data modeling, software architecture design, and the integration of intuitive user interfaces. It also highlights the technological choices made, as well as the specific constraints related to the targeted institutional environment.

Through this initiative, we aim to illustrate the key role that a well-designed digital solution can play in optimizing administrative processes and enhancing human capital within the public service.

Problematic

Currently, human resources management at ONEF is mainly based on manual methods, using Excel and Word files, and physical archives to organize and track staff data. Although this organization is still functioning, it is quickly showing its limits in an administrative environment that requires efficiency, security and traceability. Indeed, this approach leads to several major problems:

- Dispersed data across different media, making it difficult to synchronize, update and access in real time.
- The lack of automated control leading to inconsistencies, duplication and loss of integrity of information.
- The non-existence of traceability of actions, increasing the risk of errors and confusion.
- High workload for personnel, reducing productivity and increasing the likelihood of human error.
- The difficulty of quickly accessing all the data because of the lack of centralization.
- The significant loss of time spent researching, editing and archiving documents.
- The vulnerability of sensitive data, which could be lost in the event of a disaster or human error.
- The lack of a local network and Internet access within the HR department, limiting any connected or collaborative solution.

A modern, centralized and automated HR management system would therefore be essential to meet these challenges, ensure information security and improve overall efficiency.

Project objectives

We propose the implementation of an automated solution to replace the current management, which relies on manual, risky procedures and prone to many inconsistencies such as human error, lack of documentation and visibility. The goal of this solution is to develop a desktop application using recent technologies, capable of meeting the human resources management needs of the National Observatory of Education and Training (ONEF).

The main objective is to design a centralized, secure and efficient platform to automate all essential HR processes. The solution must:

- Centralize all personnel data in a single database that can be accessed quickly, structured, and securely.
- Facilitate complete personnel management, including adding, editing, searching, and archiving data.
- Integrate leave and absence management with request tracking, validation workflows, and historical records.
- Enable monitoring of employee training and performance evaluations
- Guarantee access security through role-based authentication (administrator, HR,).
- Ensure complete traceability of the actions with a logging system.

- Be entirely autonomous and operable on a single shared workstation, without relying on internet or local network connectivity..

This project aims to provide ONEF with a high-performance, scalable and easily maintainable platform that enhances HR management in a modern and efficient manner.

Chapter I: Analysis of the existing situation

In order to fully understand how the current system operates and to design an effective alternative, it is essential to go through an "Existing System Analysis" phase. During this stage, we will carefully examine all the components of the current system, including workstations and the documents in use.

This chapter aims to identify the main problems and propose appropriate solutions by suggesting the necessary changes.

1.1. Study of the existing system
1.1.1. Presentation of the organization (ONEF)

The National Observatory of Education and Training (ONEF) is a public institution under the supervision of the Ministry of National Education. It was established by Presidential Decree n° 03-46 of 5 November 2003, in accordance with the recommendations of the National Commission for the Reform of the Education System.

The mission of the ONEF is to contribute to the continuous improvement of the national education system through the observation, analysis, evaluation and proposal of actions aimed at optimizing the quality of education and training in Algeria.

It collects and analyzes statistical data, monitors performance indicators, coordinates prospective studies and disseminates results to public decision-makers.

1.1.2. Internal organization

The National Observatory of Education and Training (ONEF) is structured around a General Directorate, supported by a General Secretariat, and organized into several specialized functional directorates. Each directorate comprises several departments responsible for specific missions related to the evaluation of the education system, foresight studies, documentation, cooperation, administrative management, and data analysis.

All these structures work in a complementary manner to produce studies, performance indicators, pedagogical tools, and recommendations aimed at improving the performance of the national education and training system.

The official organizational chart of ONEF illustrating this structure is presented in the **Annex.**

1.1.3. Workstation Study

This section describes the different roles involved in the current system, along with their main responsibilities.

Based on our observations, two main roles have been identified:

HR Administrator: Responsible for overall supervision, the HR administrator manages employee records, approves leave requests, monitors training programs, and coordinates various HR functions. They are also in charge of organizing and archiving internal data.

HR User: An administrative officer with more operational responsibilities, such as enteringemployee information, tracking files on a daily basis, and forwarding documents. This person often interacts directly with employees for internal procedures.

1.1.4. Study of the HR documents used

This section highlights the official documents used within the department and the informal information flows that result from them. The goal is to gather the necessary data for the design phase, to understand the types of documents generated, and to assess the resulting data outputs.

The official documents used are listed in the **Annex.**

1.2. Critique of the Existing System

An analysis of the current environment within ONEF's Human Resources department reveals several functional and organizational limitations that undermine the efficiency and reliability of personnel management. The main shortcomings identified include:

- Fragmented and unsynchronized data spread across multiple formats and locations.
- Complex and rigid procedures, often with redundant task sequences.
- Difficulties in information sharing between staff members, often resulting in a loss of context.
- High risk of errors due to a lack of validation mechanisms or assisted data entry
- Strong dependence on staff memory and habits, due to the absence of an intelligent monitoring system.

1.3. Proposal of solutions

To address these challenges, a dedicated software solution would significantly improve workflow efficiency while reducing manual workload. The goal is not simply to replace Excel with a database, but to provide a coherent, structured, and interactive framework that can:

- Support decision-making through dashboards and historical records.
- Facilitate request processing via integrated digital forms.
- Standardize data entry to ensure information consistency.
- Ensure data privacy through controlled access rights.
- Simplify searches and updates using filters, dynamic queries, and data export functions.

1.4. Comparison between the current and the new proposed system

Criteria	Current system	Proposed system
Tools used	Excel, Word, papier	Desktop application with database
Data centralization	No, multiple independent files	Yes, centralized in a single shared database
Access security	No control mechanism	Authentication + Roles
Search for information	Slow	Fast
Traceability	Nonexistent	Automatic logging
Data update	Manual	Integrated control via interface
Productivity	Low for repetitive tasks	Significant time savings

Table 1: Comparison between the current and the new proposed system

Chapter II: Analysis and design of the solution
2.1. Specification of requirements
Before proceeding with system modeling, it is essential to clearly define the requirements of the target system. This step identifies what the software must accomplish in terms of functional requirements, as well as the quality and performance expectations referred to as non-functional requirements.

2.1.1. Functional requirements
Functional requirements define the essential services the application must provide, based on user roles. These were identified through analysis of the current system, user interviews, and the project's defined objectives.

Here are the main features expected:

Authentication and Role Management :
- Secure login with username and password.
- Role-based access: HR Administrator, HR User.

Employee Management :
- Add, edit, delete, or archive employee records.
- Search employees using filters (e.g., name, ID number).
- View detailed personnel profiles.

Leave Management :
- Add, edit, or delete employee training records.

Assessment Management :
- Add, edit, or delete assessments (annual or semi-annual).
- View historical evaluation records.

Action History (Logging) :
- Automatically log all user actions within the system.
- Allow administrators to view the full action history.

Data Export :
- Generate PDF or Excel documents for administrative use.

2.1.2. Non-functional requirements

Non-functional requirements specify the quality characteristics and technical constraints to ensure the system is effective and sustainable.

Identified needs include:

Performance: Fast and responsive user experience during regular operations.
Availability: Fully operable at all times on the dedicated workstation, independent of network connectivity.
Reliability: Accurate data recording, field validation to minimize input errors.
Scalability: Modular design allowing future feature expansion.
Compatibility: Operates seamlessly on Windows without complex setup.
Compliance: Ensures confidentiality and integrity of HR data.

2.2. UML Modeling

The analysis and design phase is a cornerstone of any software development process. It aims to model the system's functional and technical aspects based on observations of the existing environment. This modeling forms the logical and structural foundation for the future application, ensuring alignment between user needs and the final solution.

This section presents the main UML diagrams used to formalize use cases, system interactions, and data models. It also outlines the database structure and the general architecture of the application.

2.2.1. Use Case Diagram

The Use Case Diagram represents the primary functionalities of the system from the user's perspective. It illustrates the different interactions between actors and the system through a set of use cases.
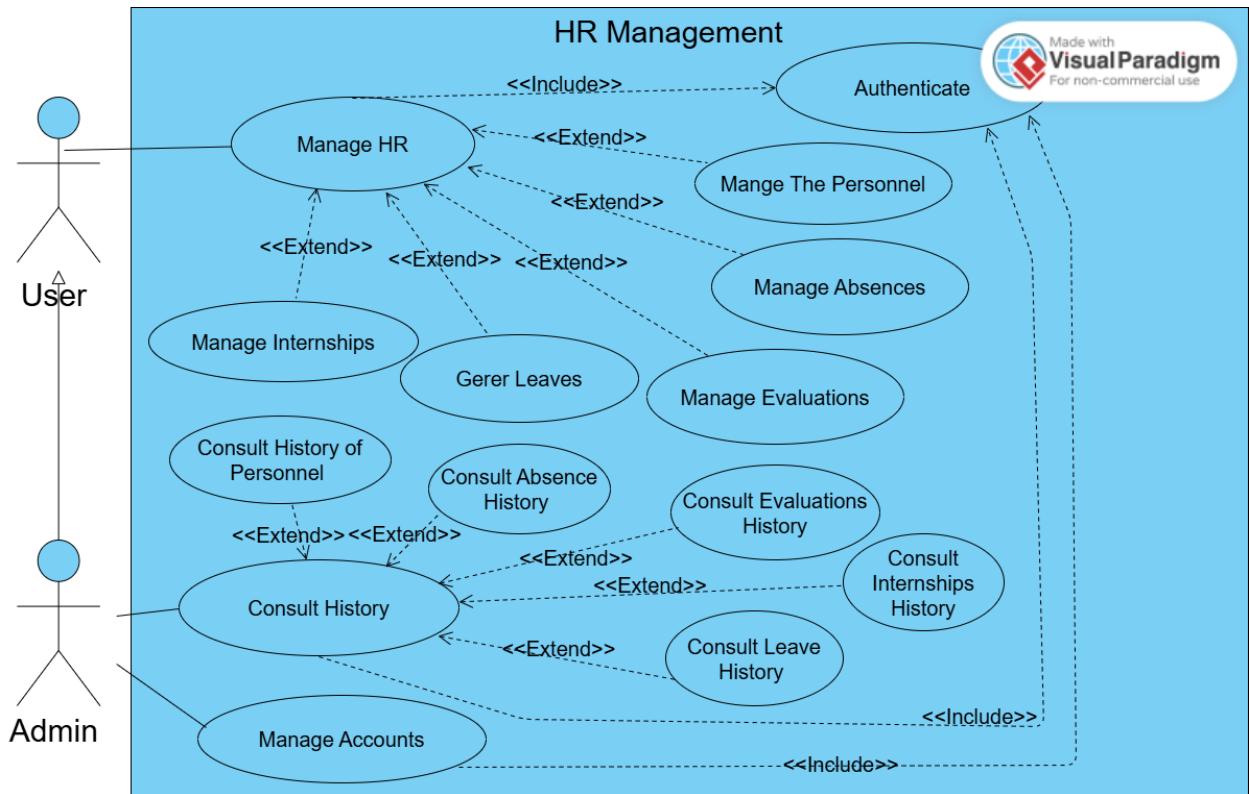
Figure 1: Overall Use case Diagram

Additional use case diagrams related to personnel management, leave, training, absences, and evaluations are provided in the **Annex.**

2.2.2. Sequence Diagrams

Sequence diagrams are UML behavioral diagrams used to model the sequence of interactions over time between users (actors) and the system. They detail the message exchanges, method calls, and responses for specific scenarios using lifelines.
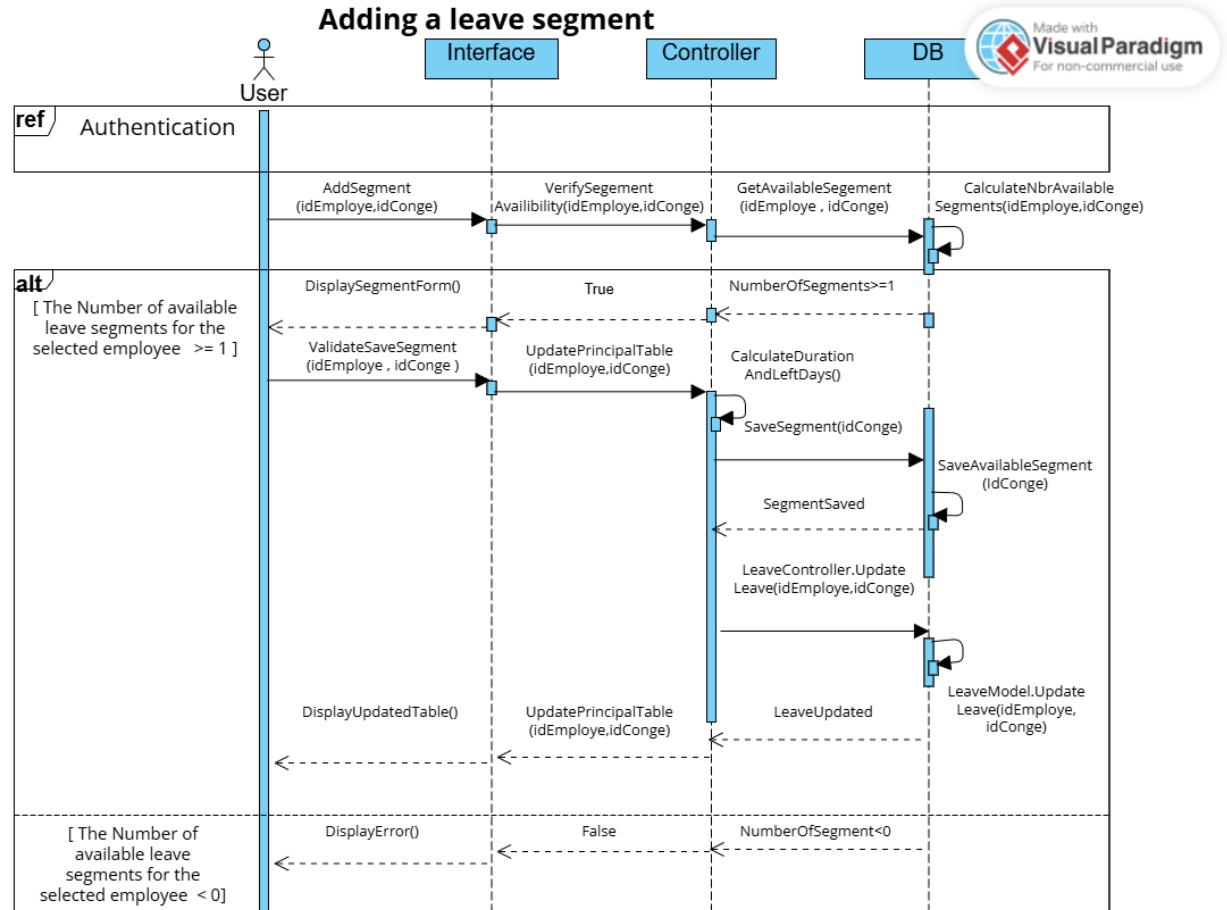
Figure 2 : Sequence diagram for adding a leave segment

An additional sequence diagram for authentication is presented in the **Annex**.

2.2.3. Class diagram

The Class Diagram offers a static representation of the system's object model. It defines key entities, their attributes, methods, and relationships. This serves as a blueprint for object-oriented design and the foundation for implementation.
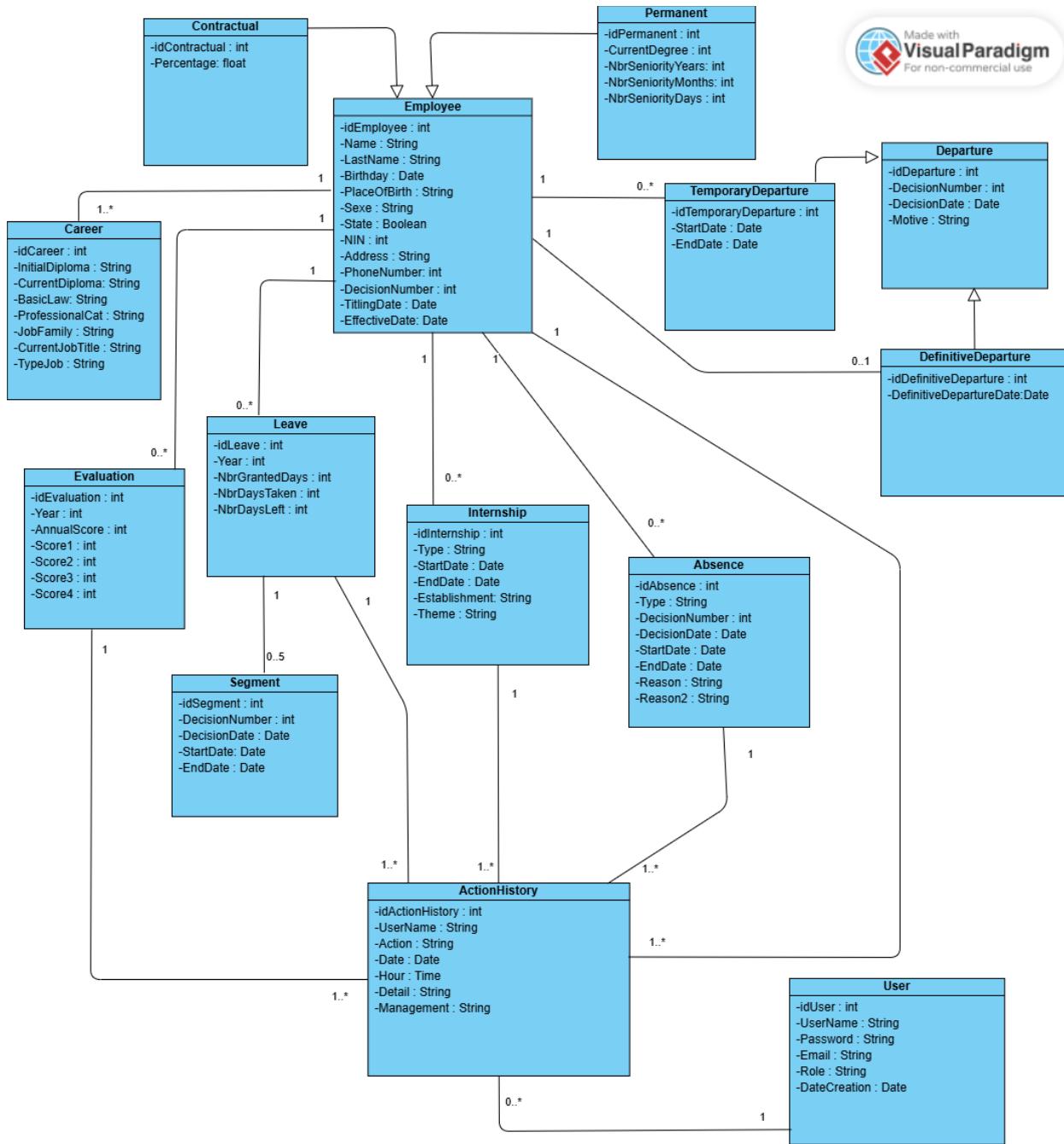
Figure 3 : Class Diagram

2.3. General architecture of the application

The application follows the Model-View-Controller (MVC) architectural pattern, which separates responsibilities into three core components:

- **Model :** Manages business logic and data (e.g., employee records, departures)
- **View :** Provides the user interface, built with PyQt and supporting Arabic (RTL) and dark mode
- **Controller :** Handles user actions, manages data flow between Model and View, and updates the interface

This architecture ensures clarity, facilitates future enhancements, and improves overall maintainability. [7]

2.4. Database Design

2.4.1. ORM Architecture (SQLAlchemy)

Role of Object-Relational Mapping (ORM)

To streamline data management and ensure a clean separation between data storage and business logic, the Object-Relational Mapping (ORM) approach was adopted. ORM allows developers to interact with database records using Python objects rather than writing raw SQL queries. This significantly simplifies code maintenance, enhances readability, and reduces the potential for SQL-related errors. [3]

Rationale for Using SQLAlchemy

For this project, we chose SQLAlchemy, one of the most powerful and widely adopted ORM libraries in the Python ecosystem. Several factors motivated this choice:

- **High flexibility:** SQLAlchemy supports both the high-level declarative syntax (class = table) and a lower-level core expression language, making it suitable for projects of varying complexity. [3]
- **Native MySQL support:** It integrates seamlessly with MySQL, our selected database management system, chosen for its robustness and efficiency. [4]
- **Clean and structured code:** SQLAlchemy promotes modular and maintainable code, encouraging a clear separation between the user interface (PyQt5) and the persistence layer.[6]
- **Portability:** The ORM abstracts the database layer, allowing easy migration to other RDBMS (e.g., PostgreSQL, SQLite) with minimal changes to the application logic. [4]
- **Strong community and documentation:** Its popularity ensures excellent documentation and community support, which is invaluable for long-term maintenance and scalability.
- **Automatic relationship management**: SQLAlchemy natively handles relationships such as one-to-many and many-to-many, as well as constraints and transactions, reducing the likelihood of integrity issues.

By mapping each Python class in our application directly to a database table, SQLAlchemy enforces a strong consistency between the data model and the underlying database schema. This ensures reliability and makes development more intuitive and less error-prone. [3]

2.4.2. Data Logic Model

- **employees (idEmployee, …)** → in the data table (Annex)
- **permanent_staff (idPermanent**, idEmployee*, CurrentRank, YearsOfSeniority, MonthsOfSeniority, DaysOfSeniority)
- **contract_staff (idContract**, idEmployee*, Percentage)
- **careers (idCareer**, idEmployee*, …) → in the data table (Appendix D)
- **trainings (idTraining**, idEmployee*, Type, StartDate, EndDate, Institution, Topic)
- **evaluations (idEvaluation**, idEmployee*, Year, AnnualScore, Score1, Score2, Score3, Score4)
- **leaves(idLeave**, idEmployee*, Year, DaysAllocated, DaysTaken, DaysRemaining)
- **leave_segments (idSegment**, idLeave*, DecisionNumber, DecisionDate, StartDate, EndDate)
- **absences (idAbsence**, idEmployee*, Type, DecisionNumber, DecisionDate, StartDate, EndDate, Reason, AdditionalReason)
- **temporary_departures (idTemporaryDeparture**, idEmployee*, idDeparture*, StartDate, EndDate)
- **departures (idDeparture**, DecisionNumber, DecisionDate, Reason)
- **final_departures (idFinalDeparture**, idEmployee*, idDeparture*, FinalDepartureDate)
- **action_history (idHistory**, idUser*, idEmployee*, idTraining*, idEvaluation*, idLeave*, idAbsence*, Username, Action, Date, Time, Details, Module)
- **users (idUser**, Username, Password, Email, Role, CreationDate)

Chapter III: Implementation of the application

3.1.Development Environment

The development of the human resources management application was conducted in a technical environment adapted to the specific constraints of the project. The aim was to build a desktop, single-user application, while ensuring scalability, performance, and usability. The chosen technologies are all open-source, offering flexibility, extensibility, and ease of integration.

3.1.1. Languages and frameworks used

Programming Language: Python

Python was selected as the primary development language due to its simplicity, clarity, and rich ecosystem of libraries. It is particularly suited for business applications thanks to its support for modern development paradigms such as object-oriented programming, Model-View-Controller (MVC) architecture, and seamless database integration. Furthermore, Python offers excellent cross-platform compatibility, especially for Windows environments, which aligns with the deployment context of this project. [1]

Graphical User Interface Framework: PyQt5

The PyQt5 framework was used to build the graphical user interface. PyQt5 provides a powerful and flexible set of tools for developing professional, dynamic, and customizable interfaces, including complex components such as forms, tables, and dialogs. [2]

Key advantages of PyQt5 include:
- Seamless integration with Python
- Native OS styling for improved user experience
- Support for drag-and-drop, event handling, multilingual applications, and right-to-left (RTL) languages, essential for Arabic support
- Separation of concerns between interface design (frontend) and application logic (backend)

This framework allowed us to build an ergonomic, user-friendly, and accessible interface adapted to the specific needs of HR agents.

Database Management System: MySQL

We opted for MySQL, a robust and high-performance relational database management system, to handle all structured data in the application. Its integration with Python and SQLAlchemy is straightforward and efficient. MySQL provides a reliable storage solution for sensitive information such as:

- Employee records
- Leave and absence tracking
- Training and professional development
- Evaluation histories

Its scalability and widespread use make it a suitable choice for long-term maintenance and evolution. [4]

Object-Relational Mapping (ORM): SQLAlchemy

To facilitate interaction between the Python application and the MySQL database, SQLAlchemy was used as the ORM layer. This abstraction allows the application to interact with database tables through Python classes, offering several advantages:

- Object-oriented manipulation of data, simplifying development.
- Automatic handling of relationships (e.g., one-to-many, many-to-many) between entities.
- Simplified and secure query construction, reducing the need for direct SQL.
- Transparent transaction management.
- Easy adaptation to changes in the database schema.

3.2. Presentation of the application and its interfaces

3.2.1. Main features

To support and enhance human resources management operations within ONEF, the application offers several essential modules.

Below are screenshots illustrating the main features of personnel management, including the employee table along with the statistics sidebar:
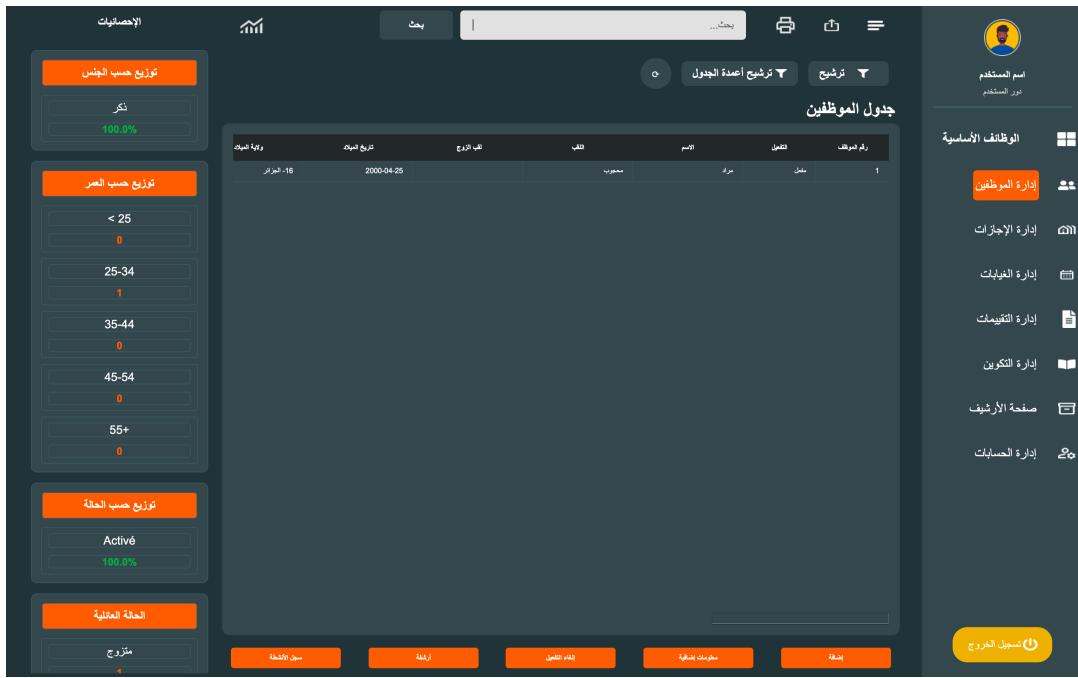
Figure 4: Main Page

Screenshot of the employee registration form, allowing the entry of personal and administrative information:

Figure 5 : Employee Registration

Screenshot of the employee deactivation form, used to manage temporary or permanent departures :

Figure 6 : Temporary Departure

Screenshot illustrating the PDF generation feature from the application's data:
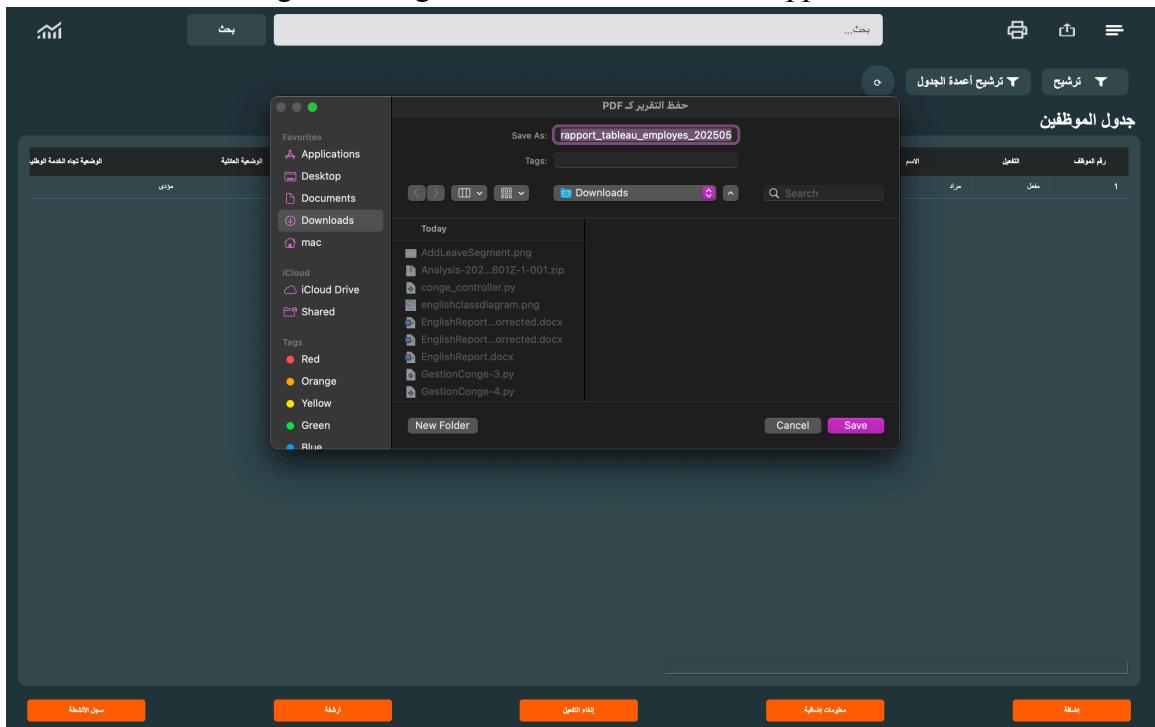
Figure 7 : PDF File Generation

Screenshot of the archived employees table, displaying all information related to the archiving of each staff member:

Figure 8 : Archive Page

Action History Table (Logging), tracking all operations performed within the system:

Figure 9 : History

17

3.2.2. Navigation between modules

The application is built on a modular architecture, designed to provide intuitive and efficient navigation. A navigation panel located on the right side of the interface provides direct access to the main functional modules.
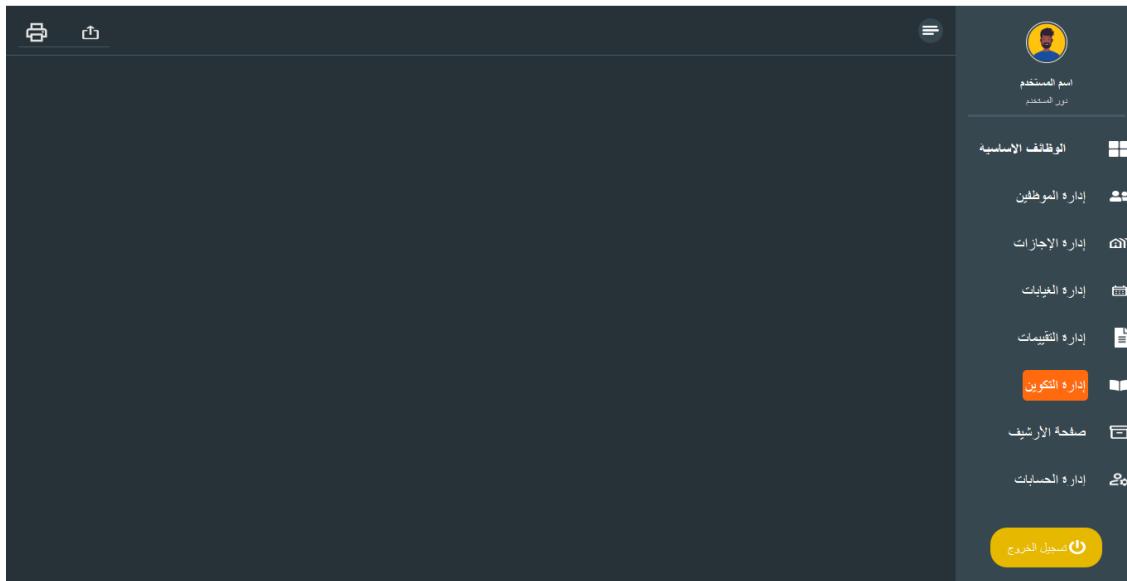
*Figure 10 : Sidebar*

3.2.3. Security and User Roles

To access our application, a verification of the username and password is performed.Here is an interface that illustrates how it works:
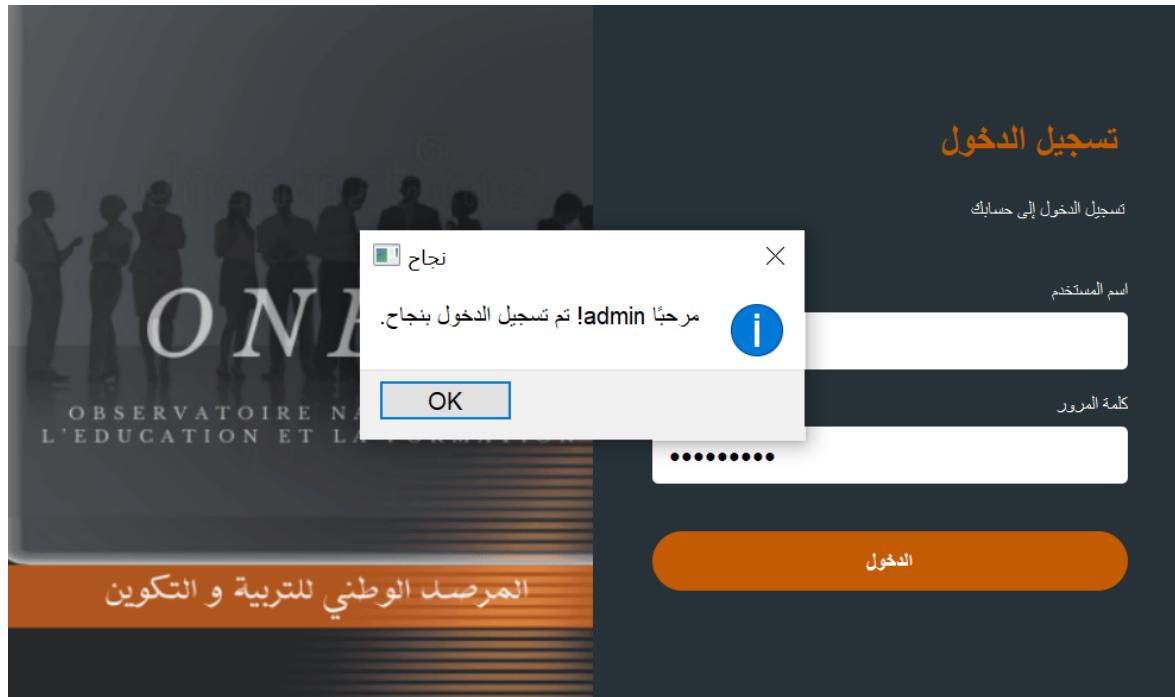
Figure 11 : Authentification Interface

And here is an example illustrating the deletion of an account:
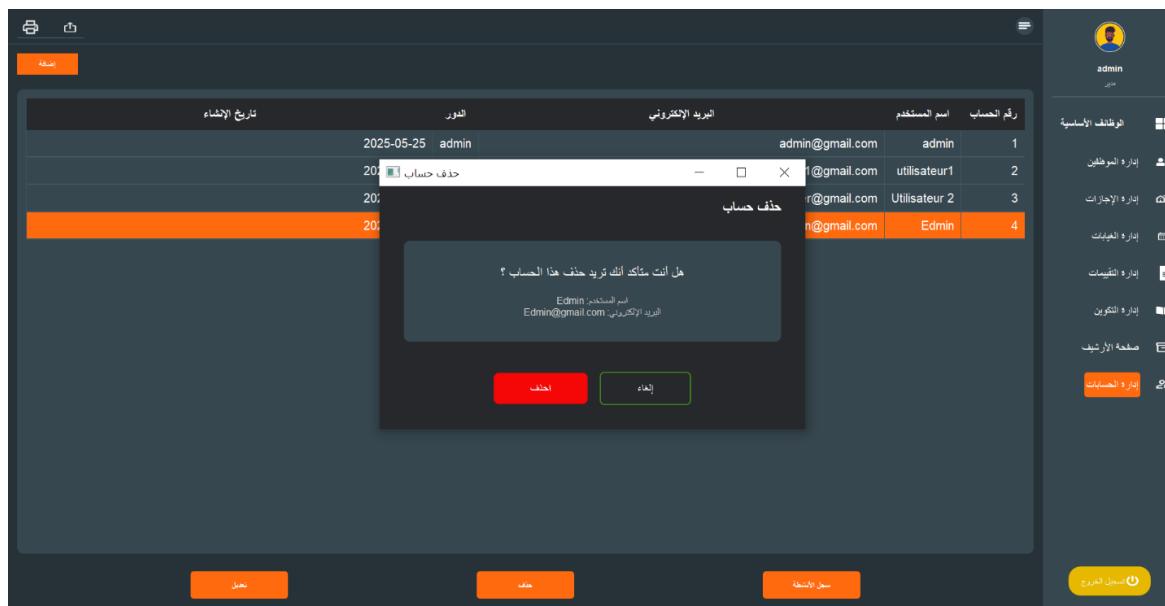

Figure 12 : Delete Account

For more examples of interfaces, please refer to the **Annex.**

General conclusion

This project focused on the design and development of a desktop application dedicated to human resources management for the National Observatory of Education and Training (ONEF). Faced with a management system still largely manual, based on office documents and paper archives, the organization had an urgent need to modernize its tools and practices.

The main objective of the project was to offer a centralized, standalone, and secure digital solution, specifically adapted to the constraints of ONEF, including the absence of a local network and Internet access. The developed application allows for the centralization of all personnel data, efficient management of leave, departures (temporary and permanent), training programs, and ensures rigorous monitoring of actions through a logging system. Special attention was paid to the user interface's ergonomics, support for right-to-left (RTL) Arabic text, and the integration of a dark theme to enhance readability.

On the technical side, the project relied on PyQt5 for the graphical user interface, SQLAlchemy as the Object-Relational Mapper (ORM) for data access, and MySQL for secure and structured information storage. The entire system operates offline, while ensuring traceability, security, and ease of use for HR personnel.

However, the implementation of the project raised several challenges, including:

- The lack of a local network and Internet access, which required the development of a fully autonomous solution installed and running on a single workstation without external dependencies;
- The complexity and lack of formalization in HR processes, which required in-depth needs analysis to translate them into relevant digital functionalities;
- Adaptation to Arabic writing (RTL), requiring precise adjustments in alignment, input, and text display;
- The dispersion and heterogeneity of existing data, which imposed a rigorous process of centralization, cleaning, and structuring.

Despite these constraints, the project resulted in a fully functional application, tested and aligned with the defined objectives, offering a reliable solution tailored to ONEF's specific context.

While the application currently meets the fundamental needs of the institution, several avenues for improvement could be considered to further develop the system:

- Transition to a client-server architecture in case a local network is deployed, to enable multi-user access and better collaboration;
- Addition of new features such as automated generation of administrative documents, employee performance evaluations, or the integration of graphical statistics and interactive dashboards;
- Implementation of full multilingual support (French/Arabic) to broaden accessibility for all user profiles.

In conclusion, this project represents a crucial first step toward the digital transformation of human resources management at ONEF. It lays a solid foundation for a modern, scalable system aligned with the ongoing goals of public service improvement in the educational sector in Algeria.

Bibliographie

[1] Python Documentation

- **Python Software Foundation: https://docs.python.org/3/**

[2] PyQt5 Documentation

- **Riverbank Computing: https://www.riverbankcomputing.com/static/Docs/PyQt5/**

[3] SQLAlchemy Documentation

- **SQLAlchemy: https://docs.sqlalchemy.org/**

[4] MySQL Documentation

- **Oracle Corporation: https://dev.mysql.com/doc/**

[5] UML Distilled

- **A Brief Guide to the Standard Object Modeling Language, Fowler, M. (2004). Addison-Wesley Professional.**

[6] Clean Code: A Handbook of Agile Software Craftsmanship

- **Martin, R. C. (2009). Pearson Education.**

[7] Design Patterns: Elements of Reusable Object-Oriented Software

- **Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Addison-Wesley Professional.**

[8] Human Resources Management

- **Peretti, J.-M. (2020). Vuibert.**

[9] Digital Transformation of Organizations

- **Bounfour, A. (2018). Dunod.**

[10] Online Articles and Resources

- **Towards Data Science: https://towardsdatascience.com/**

- **Real Python: https://realpython.com/**

[11] Institutional Reports and Guidelines

• **Ministry of National Education (Algeria).**

• **Official documents related to ONEF and Algerian administrative standards.**

[12] Tutorials and Forums

• **Stack Overflow: https://stackoverflow.com/**

• **GitHub: https://github.com/**