المدرسة الوطنية للعلوم التطبيقية - بني ملال

Ecole Nationale des Sciences Appliquées - Béni Mellal

University Sultan Moulay Slimane

National School for Applied Sciences - Beni Mellal

N°: TDI-1/2024

End-of-Studies Project for obtaining the degree:

State Engineer in

Industrial Digital Transformation

Under the theme:

Development of a Desktop E-Commerce Application using PySide6 and QtQuick

Performed by:

Mouad AIT OUGRRAM

Supervised by:

Mr. ZOUINE Safouane (CISCM)
Pr. Hamid OUANAN (ENSA Beni Mellal)

Defended on 12 June 2024 before the jury members:

Pr. Ibtissame AOURAGHE	ENSA Beni Mellal	Chairperson-Reporter
Pr. Hamid SADIKI	ENSA Beni Mellal	Reporter
Pr. Hamid OUANAN	ENSA Beni Mellal	Educational Supervisor

Academic year: **2023-2024**

Université Sultan Moulay Slimane

Ecole Nationale des Sciences Appliquées - Béni Mellal

N° d'ordre : TDI-1/2024

Projet de Fin d'Etudes pour l'obtention du diplôme :

Ingénieur d'État en

Transformation Digitale Industrielle

Sous le thème :

Developpement d'une Application Desktop E-Commerce avec Pyside6 Et Qtquick

Réalisé par :

Mouad AIT OUGRRAM

Encadré par :

Mr. ZOUINE Safouane (CISCM)
Pr. Hamid OUANAN (ENSA Beni Mellal)

Soutenue le 12 juin 2024 devant le jury composé de :

Pr. Ibtissame AOURAGHE	ENSA Beni Mellal	Rapporteur (Président)
Pr. Hamid SADIKI	ENSA Beni Mellal	Rapporteur
Pr. Hamid OUANAN	ENSA Beni Mellal	Encadrant

Année universitaire: **2023-2024**

بسم الله الرحمن الرحيم

الحمد لله رب العالمين ۝ الرحمن الرحيم ۝

مالك يوم الدين ۝ إياك نعبد وإياك نستعين

۝ اهدنا الصراط المستقيم ۝

صراط الذين أنعمت عليهم

غير المغضوب عليهم

ولا الضالين ۝

Preface

Role: Software engineer

Company: CISCM

Location: Casablanca

Duration: Paid internship for 6 months, starting in February.

Work Mode: Hybrid, with 3 days on-site and 2 days remote.

Overview:

During my time at CISCM as an intern software engineer, I had the privilege of collaborating closely with a dynamic team, including a project manager, an assistant, and two software engineers. Our primary objective was to develop a user-friendly graphical interface using PySide6 and seamlessly integrate various functionalities through external APIs. This internship provided me with invaluable hands-on experience. I was deeply motivated by the challenges and opportunities for growth in software development, graphical interfaces, and API integration.

Responsibilities:

Throughout my internship, my responsibilities included:

- Designing and developing the graphical interface using PySide6.
- Working collaboratively with the team to grasp project requirements thoroughly.
- Integrating external functionalities using REST APIs.
- Testing and troubleshooting issues related to the graphical interface and API integration.

Skills Utilized:

In executing these tasks, I applied and further honed the following skills:

- Proficiency in Python programming.
- Knowledge of Qt, both in Python and C++.
- Application of design patterns.
- System Design.
- Ability to effectively work with external APIs.

Selection Process:

The selection process involved an initial exchange and interview with the human resources team, followed by a technical interview with one of CISCM's experts.

Résumé

Ce rapport présente mon PFE (Projet de Fin d'Études), réalisé dans le cadre de ma formation pour obtenir le diplôme d'ingénieur d'État en Transformation Digitale Industrielle à **l’École Nationale des Sciences Appliquées de Beni Mellal**, au sein de l'organisation **CISCM** Casablanca. L'objectif principal de ce projet était de développer et d'améliorer une plateforme de commerce électronique innovante dédiée aux détaillants et aux indépendants, constituant la première phase du projet **ISC**, une initiative visant à révolutionner la chaîne d'approvisionnement grâce à des technologies intelligentes et innovantes.

L'application utilise une gamme de technologies innovantes, tant pour le backend que pour le frontend. Parmi les exemples notables, on trouve Django et les technologies Qt de PySide6, ces composants ayant été sélectionnés pour répondre à des exigences spécifiques et tirer parti de leurs capacités respectives.

Pour mener à bien le projet, nous avons adopté l'approche de développement Agile Scrum, offrant un cycle de travail itératif et complet axé sur des objectifs collaboratifs et la livraison de produits de manière productive et créative. En conséquence, nous avons divisé la solution en sprints, permettant un développement efficace des différents modules.

Au départ, l'organisation hôte m'a accordé du temps pour apprendre et m'exercer. J'ai utilisé ce temps initial pour une auto-formation technique et fonctionnelle approfondie, couvrant tous les outils et technologies essentiels que je pourrais rencontrer pendant mon stage, comme Django Channels, les API, l'architecture de projet, PySide6, Qt6, QWidget, QML, les graphiques, le multi-threading, l'animation, etc. De plus, j'ai participé à plusieurs projets annexes pour me familiariser avec les nouvelles technologies et la méthodologie du projet employée.

Ensuite, j'ai été affecté au projet "AM", où j'ai consacré ma période de stage. Tout au long du projet, J'ai commencé par découvrir et comprendre le projet assigné grâce aux explications fournies par l'équipe. Nous avons ensuite procédé à la phase de conception, en veillant à respecter les meilleures pratiques modernes pour les applications de bureau et les conventions de l'architecture MVC. Enfin, nous avons progressivement implémenté chaque module de l'application.

Mots-clés : Django Channels, Redis, Qt6, Pyside6, QML, MVC, Scrum

Abstract

This report showcases my End-of-Studies project, conducted as part of my pursuit of a State Engineer diploma in Industrial Digital Transformation at the **National School for Applied Sciences of Beni Mellal**, within the organization **CISCM** Casablanca. The primary objective of this project was to develop and enhance an innovative E-Commerce platform dedicated to retailers and independents, serving as the First Phase of **ISC** Project an initiative aimed at revolutionizing the supply chain through intelligent and innovative technologies.

The application utilizes a range of innovative technologies, both in the backend and frontend. Notable examples include **Django, PySide6 Qt** Technologies, these components were selected to cater to specific business requirements and leverage their respective capabilities.

To effectively execute the project, we adopted the Agile Scrum development approach, providing an iterative and comprehensive work cycle that focuses on collaborative objectives and delivering products productively and creatively. Accordingly, we divided the solution into sprints, enabling efficient development of different modules.

Initially, the host organization provided me with some time to learn and practice, I used my initial time to do extensive technical and functional self-training covering all essential tools and technologies that I could encounter during my internship Like Django Channels, APIs, Project Architecture, PySide6, Qt6, QWidget, QML, Charts, Multi-Threading, Animation …and many more, Furthermore, I engaged in several side projects to familiarize myself with new technologies and the project methodology employed.

Subsequently, I was assigned to the “AM” project, where I dedicated my internship period. Throughout the project, I began by discovering and understanding the assigned project through explanations provided by the team. We then proceeded with the design stage, ensuring adherence to modern Desktop Application best practices and the conventions of the MVC architecture. Finally, we gradually implemented each module of the application.

Keywords: Django Channels, Redis, Qt6, Pyside6, QML, MVC, Scrum

ملخص

يعرض هذا التقرير مشروع التخرج الخاص بي، والذي تم إجراؤه كجزء من إجراءات الحصول على دبلوم مهندس دولة في تخصص " التحول الرقمي الصناعي " في المدرسة الوطنية للعلوم التطبيقية في بني ملال، بشراكة مع شركة CISCM بالدار البيضاء. كان الهدف الأساسي من هذا المشروع هو تطوير وتحسين منصة تجارة إلكترونية مبتكرة مخصصة لتجار التجزئة والمستقلين، وتعمل كالمرحلة الأولى من مشروعISC ، وهي مبادرة تهدف إلى إحداث ثورة في سلسلة التوريد من خلال التقنيات الذكية والمبتكرة.

يستخدم التطبيق مجموعة من التقنيات المبتكرة، سواء في Backend او في Frontend. تشمل الأمثلة البارزة Djangoو PySide6 Qt Technologies ، وقد تم اختيار هذه المكونات لتلبية متطلبات الأعمال المحددة والاستفادة من إمكانياتها المعنية.

لتنفيذ المشروع بشكل فعال، اعتمدنا نهج تطويرAgile Scrum ، مما يوفر دورة عمل شاملة وتكرارية تركز على الأهداف التعاونية وتقديم المنتجات بشكل منتج ومبدع. وفقًا لذلك، قمنا بتقسيم الحل إلى مراحل، مما يتيح التطوير الفعال لوحدات مختلفة.

في البداية، منحتني المؤسسة المضيفة بعض الوقت للتعلم والممارسة، واستخدمت هذه الفترة الأولى لإجراء تدريب ذاتي فني ووظيفي شامل يغطي جميع الأدوات والتقنيات الأساسية التي يمكن أن أواجهها أثناء فترة التدريب مثل Django Channelsو APIs و Project Architecture و PySide6 و Qt6 و QWidget و QML و Charts وMulti-Threadingو ... Animation وغيرها الكثير. علاوة على ذلك، شاركت في العديد من المشاريع الجانبية للتعرف على التقنيات الجديدة ومنهجية المشروع المستخدمة.

بعد ذلك، تم تعييني في مشروع"AM" ، حيث كرست فترة التدريب الخاصة بي. طوال المشروع، بدأت باكتشاف وفهم المشروع المخصص من خلال الشروحات التي قدمها الفريق. ثم تابعنا مع مرحلة التصميم، مع ضمان الالتزام بأفضل ممارسات تطبيقات سطح المكتب الحديثة , أخيرًا، قمنا بتنفيذ كل وحدة من وحدات التطبيق بشكل تدريجي.

الكلمات المفتاحية : Django Channels ، Redis، Qt6، Pyside6، QML، MVC، Scrum

Dedication

I would like to dedicate this work primarily to my dear parents, as no dedication can truly express the deep affection I have for them, nor the immense gratitude I feel for all the sacrifices they have made and continue to make for my education and studies. I also extend this dedication to my beloved brothers,

my dear sisters,

my extended family,

as well as all my teachers

and internship colleagues

And of course, to my dear friends...

Please accept my best wishes for success, happiness, and prosperity.

Mouad AIT OUGRRAM

Acknowledgments

First and foremost, I would like to express my gratitude to God Almighty for granting me the strength and patience needed to complete this humble work.

I am sincerely grateful to everyone who has supported me throughout my end-of-studies project to obtain the State Engineer degree in Industrial Digital Transformation.

I extend my deepest appreciation to the Director of the National School for Applied Sciences of Beni Mellal, Pr. Belaid BOUIKHALENE, and the Deputy Director, Pr. Soufiane BELHOUIDEG, for their leadership and guidance.

I am profoundly thankful to the coordinator of the program, Pr. Hamid OUANAN, for his unwavering support and valuable insights throughout my academic journey.

My heartfelt thanks go to Mr. ZOUINE Safouane, the Director of the host company where I completed my internship as a Software Engineer. His guidance, support, and valuable feedback during my internship have been invaluable to my professional growth.

I am incredibly grateful to my educational supervisor, Pr. Hamid OUANAN, for his constant support, guidance, and encouragement throughout the project. I also extend my gratitude to the administrative staff at the National School for Applied Sciences of Beni Mellal for their dedication and support.

Finally, I would like to thank the members of the defense jury, Pr. Ibtissame AOURAGHE (Chairperson-Reporter, ENSA Beni Mellal), Pr. Hamid SADIKI (Reporter, ENSA Beni Mellal), and Pr. Hamid OUANAN (Educational Supervisor, ENSA Beni Mellal), for taking the time to review my work and provide valuable feedback.

This project would not have been possible without the collective support and guidance of everyone mentioned above. Thank you all for your contributions to my academic and professional growth.

Table of Contents

List Of Figures

List Of Tables

List Of Abbreviations

Abbreviation	Designation
MVC	Model, View, and Controller
B2B	Business to Business
B2C	Business to Consumer
LLC/SARL	Limited Liability Company / Société à Responsabilité Limitée
AI	Artificial Intelligence
VSE/SME	Very Small Enterprise / Small and Medium-sized Enterprise
AM	Alma7al which is the name of our End-study Project
REST API	Representational State Transfer Application Programming Interface
KPI	Key Performance Indicator
CRM	Customer Relationship Management
US	User Story
CRUD	Create, Read, Update, Delete
ORM	Object-Relational Mapping
Pimpl	Pointer to Implementation
D-Pointer	Data Pointer
QML	Qt Modeling Language
UI/UX	User Interface / User Experience
ASGI	Asynchronous Server Gateway Interface
Redis	Remote Dictionary Server

General Introduction

Today, e-commerce occupies a prominent place in many international organizations (Stores), notably due to the challenges related to digital transformation that regularly make headlines. In this context, my End-of-Studies project is part of the CISCM Casablanca company, where we are participating in the development of an e-commerce platform. While the title highlights the e-commerce capabilities of the application, it is essential to understand that e-commerce in the modern context extends beyond online sales. As stated in a research paper from ResearchGate:

"*The real role of Electronic Commerce is to act on applications such as Supply Chain Management (SCM), ERP, and Customer Relationship Management (CRM) - a stretching of the company on the one hand to its supplier, on the other to its customer, as well as other partners such as employees, State, etc.*" [22] Gunson, John & Blasis, J.P.. (2001)

This broader definition underscores the integrated nature of e-commerce, which aligns perfectly with the functionalities of our application. By combining MVC architecture with the powerful Qt and Django technologies, this project represents an exceptional opportunity to explore and test innovative ideas. We are convinced that this approach will allow us to develop a robust, flexible, and user-friendly e-commerce platform. Furthermore, it will position us at the forefront of innovation in our sector.

This End-of-Studies project is a crucial step in our engineering training at the National School of Applied Sciences Beni Mellal. We are enthusiastic about contributing to the development of this e-commerce platform and bringing our expertise to the field of digital transformation.

This report consists of Four chapters:

First chapter: It presents the host organization of the company and its field of operation.

Second chapter: It is dedicated to an analysis study and specification of requirements.

Third chapter: It is devoted to the tools and technologies used during the development of this application, as well as the software architecture of the project.

The remaining chapter will be devoted to the implementation of the sprints, during which we will proceed with the analysis, design, and implementation of the various project tasks.

We conclude this report with a conclusion, showing the added value of our application and the perspectives we have envisaged for its evolution.

Chapter I: General Context of the Project

In this first chapter, we will present the general context of the project, by introducing the host organization, the client, the problem, and the work methodology.

I. Presentation of the Host Organization

1.1 Presentation of the Company

The figure below represents the logo of the company CISCM.

Figure 1: logo of CISCM : Consulting in Intelligent Supply Chain & Management

Consulting in Intelligent Supply Chain & Management (CISCM) is a firm specializing in technology, consulting, management, and market research. CISCM provides strategic support to businesses, helping them achieve their goals through innovative solutions and expert knowledge.

Thanks to its expertise in the field of commerce and technology, CISCM offers sustainable support to its clients, improving the customer experience, partners, and collaborators, as well as promoting the development of sales, awareness, and internal performance.

1.2 Organigram of Intern Project Team

The company CISCM is composed of multiple teams that work on various projects across different domains, including web app development, desktop app development (in my case), and machine learning. with talented and dedicated teams that are committed to delivering innovative solutions that cater to the diverse needs of the clients.

The figure below represents an Organigram of my team.

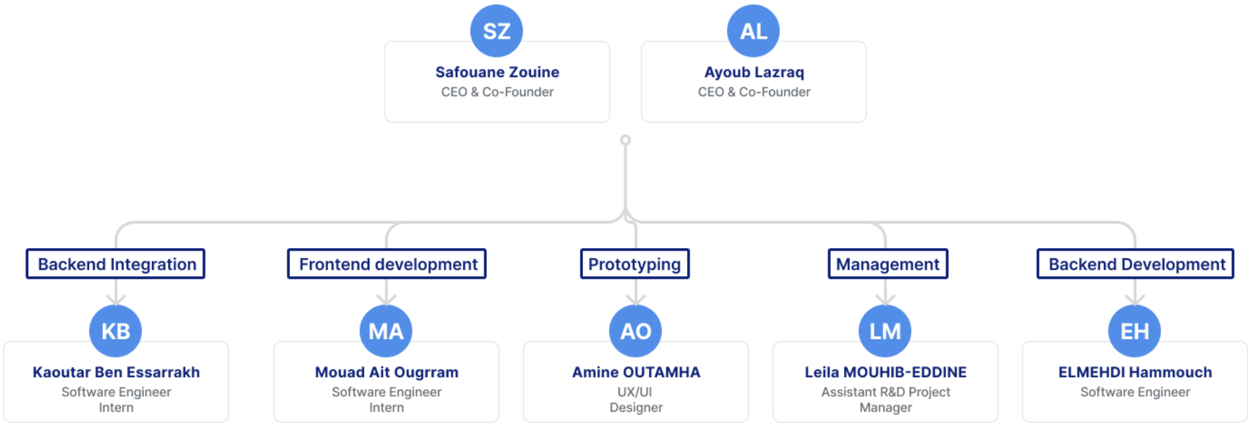


Figure 2: Organigram of Desktop App Development Team

1.3 CISCM Services

CISCM specializes in projects related to supply chain management, technology, and market research, integrating the use of innovative solutions to support businesses throughout the entire project lifecycle:

- Providing strategic advice to help B2B and B2C clients make informed decisions
- Research and development in numerical technology
- Coordination and project management
- Market studies and analysis

1.4 CISCM Company Profile

The figure below represents the company profile.

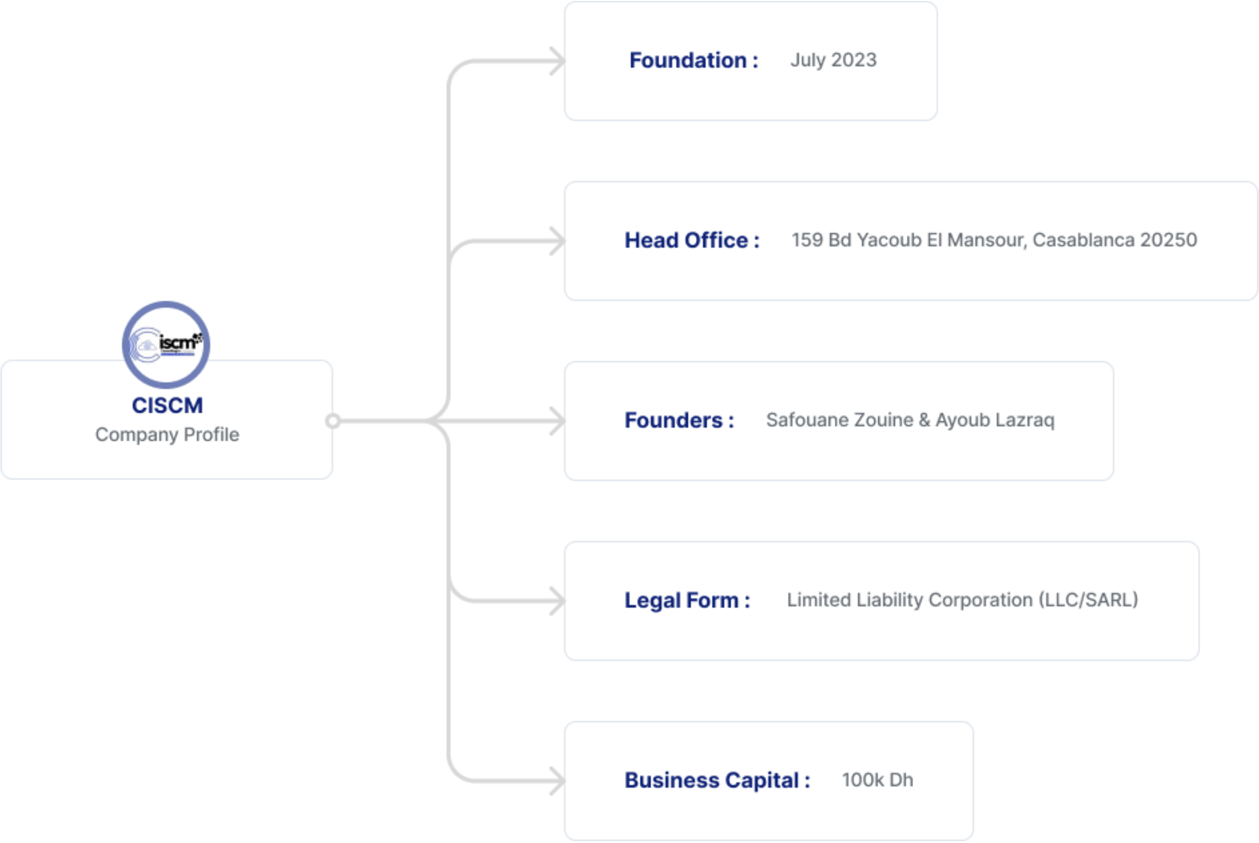

CISCM
Company Profile
Foundation : July 2023
Head Office : 159 Bd Yacoub El Mansour, Casablanca 20250
Founders : Safouane Zouine & Ayoub Lazraq
Legal Form : Limited Liability Corporation (LLC/SARL)
Business Capital : 100k Dh

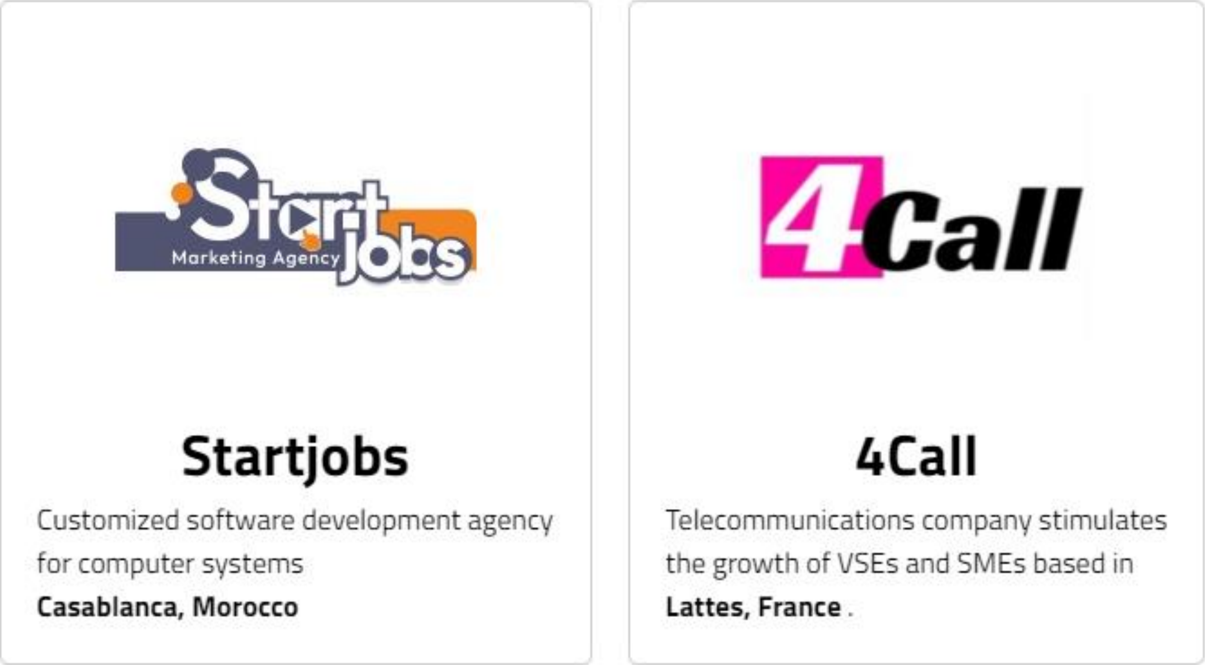
Figure 3 : Company Profile of CISCM

1.5 Business Sectors

Business Sectors that CISCM focuses on :

- Industry
- Services
- Fintech
- Supply Chain / Logistics
- AI
- Telecommunications
- Embedded systems

The figure below shows some CISCM partners:



Startjobs

Customized software development agency for computer systems

Casablanca, Morocco

4Call

Telecommunications company stimulates the growth of VSEs and SMEs based in **Lattes, France** .

Figure 4: CISCM Partners

II. Project Presentation

2.1 The General Framework of the Project

The Alma7al or “AM” project is a sub-project within the first phase of the ISC (Intelligent Supply Chain) initiative, which is an Eco-system and network of intelligent connected applications in the form of an intelligent supply chain that links investment, production, distribution, sale, delivery, and finally consumption.

Alma7al or “AM” focuses on developing an innovative platform for retailers and independents, aiming to meet their unique digital management needs. The primary goal is to simplify operations management, enhance efficiency, and support business growth through modern, intelligent solutions.

Figure 5: logo of Alma7al ("AM") Application

2.2 Problem Statement

The Enterprise Software market in Morocco is on a promising trajectory. It's projected that by the year 2024, the revenue generated from this sector will reach an impressive US$152.60 million. This upward trend signifies an increasing demand for business software solutions within the Moroccan market. The forthcoming chart will provide a visual representation of this market revenue growth, further emphasizing the potential and vitality of this sector. This data underscores the immense opportunities that lie within the realm of enterprise software development in Morocco.[1]

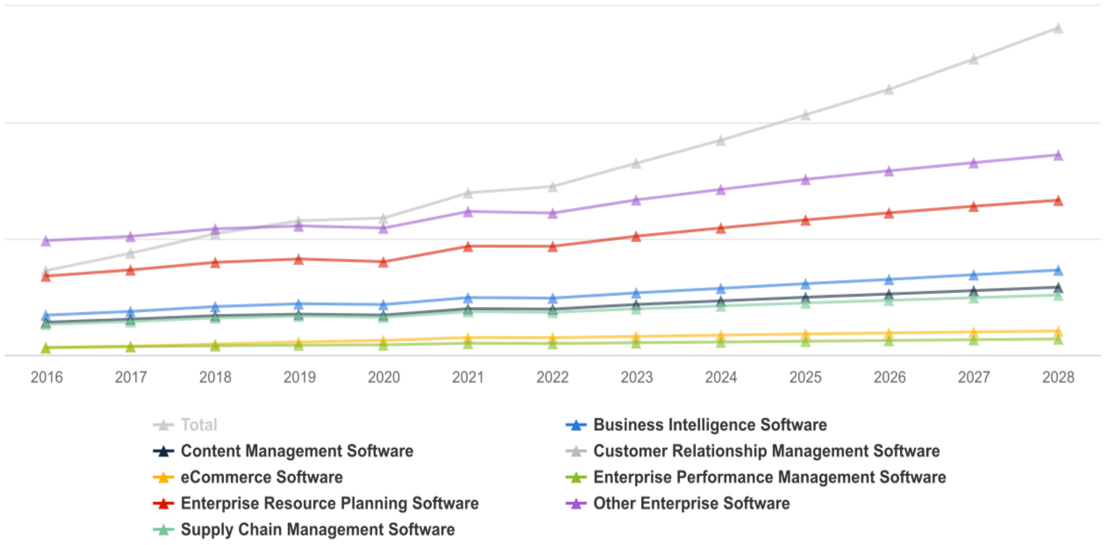


Figure 6 : Revenue in the Enterprise Software market in Morocco

The retail industry faces several challenges in managing operations like supply chain, and logistics, effectively. Traditional methods of managing these aspects often lead to inefficiencies, lack of visibility, monitoring, and multi-tasking, especially in Morocco, the retail industry is suffering from a lack of Information and just relies on estimations which is not good practice for good Business. And we can generalize all those challenges into categories:

- **Operational Challenges:**
    - ➢ Reliance on manual processes for inventory management, order tracking, and record-keeping.
    - ➢ Lack of real-time visibility into stock levels, sales data, and supply chain operations.
    - ➢ Inefficient communication and coordination with suppliers and logistics providers.
    - ➢ Difficulty in managing customer data and analyzing customer behavior patterns.
- **Technology Adoption:**
    - ➢ Resistance to change and reluctance to invest in new systems and tools.
    - ➢ Lack of technical expertise and training for staff to leverage digital solutions effectively.
- **Data and Analytics:**
    - ➢ Lack of data-driven decision-making capabilities due to limited access to relevant data.
    - ➢ Inability to leverage advanced analytics and business intelligence tools for insights and optimization.

Now to fulfill the needs of the traditional retailer we need a Software with these characteristics:

- **Operational Challenges:**
    - ➢ ~~Reliance on manual processes for inventory management, order tracking, and record-keeping.~~
    - ✓ Intuitive and user-friendly inventory management system for efficient stock tracking and updates.
    - ➢ ~~Lack of real-time visibility into stock levels, sales data, and supply chain operations.~~
    - ✓ Real-time visibility into stock levels, sales data, and supply chain operations through a centralized dashboard.
    - ➢ ~~Inefficient communication and coordination with suppliers and logistics providers.~~
    - ✓ Integrated order management system for streamlined order processing, tracking, and fulfillment.
    - ➢ ~~Difficulty in managing customer data and analyzing customer behavior patterns~~.
    - ✓ Customer relationship management (CRM) modules for capturing and analyzing customer data, including purchase history, preferences, and behavior patterns.
- **Technology Adoption:**

- ➢ ~~Resistance to change and reluctance to invest in new systems and tools.~~
- ✓ Intuitive and user-friendly interface with a minimal learning curve, reducing resistance to change.
- ➢ ~~Lack of technical expertise and training for staff to leverage digital solutions effectively.~~
- ✓ Dedicated Technical Support

- **Data and Analytics:**
  - ➢ ~~Lack of data-driven decision-making capabilities due to limited access to relevant data.~~
  - ✓ Robust data collection and integration capabilities, allowing retailers to consolidate data from various sources (e.g., sales, inventory, customer interactions).
  - ➢ ~~Inability to leverage advanced analytics and business intelligence tools for insights and optimization.~~
  - ✓ Built-in business intelligence and analytics tools for generating insightful reports, dashboards, and visualizations.

In conclusion, to address the operational challenges, technology adoption barriers, and data and analytics gaps faced by traditional retailers, a comprehensive software solution is imperative.

2.3 Proposed Solution

We propose to solve these problems by creating an e-commerce platform called “Alma7al.” Specifically, Alma7al is an integrated ERP, CRM software solution designed to facilitate the way retailers manage their supply chain and store operations by providing a comprehensive, user-friendly platform that centralizes data and is dedicated to meet the needs of retailers across their supply chain and it insists on providing 4 key features and functionalities:

Digital Collection:

The application would likely collect data from various sources across the supply chain, data related to inventory levels, product shipping, sales figures, client behavior information, Expense data, and other relevant information, creating a centralized data repository. This digital collection would provide a single source of truth for data across the entire supply chain, enabling better decision-making and forecasting.

Operations Visualization:

With the collected data, the application would offer intuitive dashboards and visualizations that provide a comprehensive view of the entire supply chain operations. These visualizations could include inventory levels across multiple locations, order fulfillment rates, and performance metrics. Users could quickly track shipments, and monitor key performance indicators (KPIs) through these visual representations.

Digital Transformation:

The application would facilitate the digital transformation of supply chain and retail operations by automating various processes and enabling data-driven decision-making. It could integrate with predictive analytics and machine learning models to forecast demand and optimize inventory levels. Also, Automated workflows and notifications could be set up to alert stakeholders of critical events or deviations from planned operations.

Facilitated Collaboration:

Users could share real-time updates, exchange documents, and engage in discussions between Store Employees within the platform,

Role-based access controls and secure data sharing would ensure that sensitive information is shared only with authorized parties.

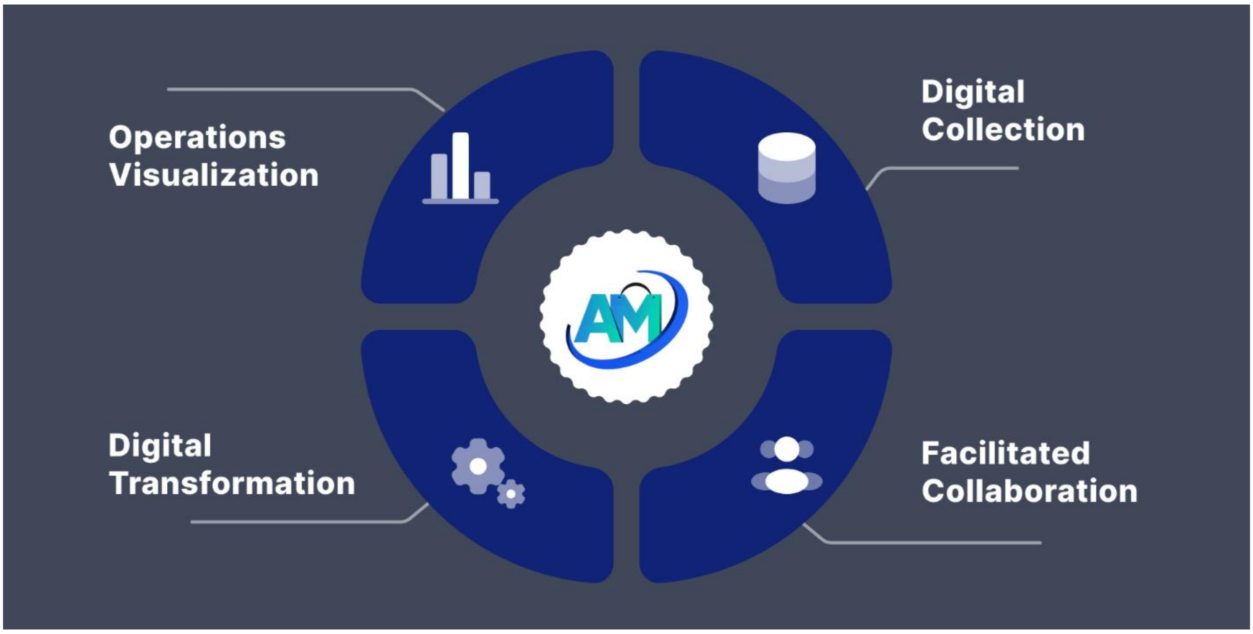


Figure 7: Representation of The Core features of ALma7al

2.4 Development Methodology

We have chosen to adopt the agile methodology, specifically the Scrum framework, for managing our project for several justified reasons. First, agility provides an iterative and collaborative approach that promotes adaptability and flexibility in the face of project changes and evolutions. By using Scrum, we can organize development into short cycles called "sprints," which enables continuous delivery and regular stakeholder feedback.

Furthermore, Scrum promotes transparency and communication within the team. Clearly defined roles, such as the Scrum Master and the Product Owner, ensure a clear distribution of responsibilities and effective coordination. Regular meetings, such as Daily Scrums and Sprint Reviews, allow the team to stay aligned with the project goals and make informed decisions.

Conclusion

In this chapter, we have introduced the CISCM company, defined the general project framework, the problem statement, and proposed solution. We also outlined the project's Core features and functionalities. Towards the end, we discussed the adopted development methodology. The next chapter will be dedicated to the analysis and specification of requirements.

Chapter II: Preparation Phase - Analysis and Specification of Needs

The stage of analysis and specification of needs plays a crucial role in enabling a better understanding of the required work through sprint planning, as well as defining the functional, and non-functional, objectives and technical needs.

I. Scrum Methodology

1.1 Definition

The company CISCM adopts the Scrum methodology for project management. Scrum is an iterative agile method that relies on short cycles. Each cycle, called a sprint, lasts from two to four weeks. The objective of each sprint is to prioritize the most important features and deliver a potentially usable product. This approach promotes rapid delivery of commercial value and offers great flexibility, allowing us to easily adapt to changes requested by the client. In summary, Scrum is characterized by short iterations, a focus on key features, and flexibility to meet changing client needs.[2]

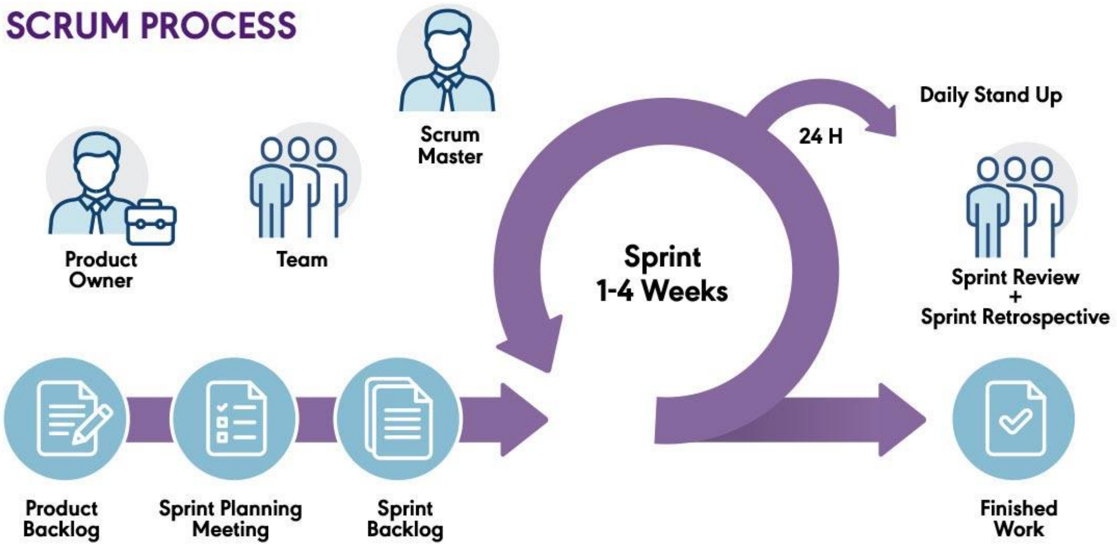


Figure 8 : Scrum methodology process

When it comes to Scrum, here are the key concepts to remember:

The Scrum Master: The Scrum Master is responsible for implementing Scrum in accordance with the Scrum Guide. Their role is to help everyone understand the theory and practice of Scrum, both within the Scrum team and within the organization. They are also responsible for ensuring the effectiveness of the Scrum team by helping it improve its practices.

The Team: The Scrum team is autonomous and responsible for product development. Its size should remain relatively small, usually 7 to 9 people, but can go up to 200 people. It operates autonomously and collaborates closely to achieve project objectives.

The Product Owner: The Product Owner is responsible for defining and prioritizing the list of product features, called the backlog. They are also responsible for choosing the date and content of each sprint based on information provided by the team.

The Product Backlog: This is a prioritized list of features to be developed or improved throughout the project. The backlog does not need to contain all features from the beginning of the project, it evolves in parallel with the client's needs. The backlog can be organized into User Stories (US), which follow the structure "as a..., I want to..., so that I can...". Each User Story is evaluated in terms of business value to prioritize backlog items. The team also evaluates the implementation complexity of each User Story.

Sprint Planning: This stage aims to define what can be delivered in the sprint and how the work will be done. Sprint planning is done in collaboration with the entire Scrum team to ensure a common understanding of objectives and tasks to be accomplished.

1.2 The Scrum Roles

The team consists of 6 people:

Table 1: Composition of The Team

Member	Role
Mr.LAZRAQ Ayoub	**Product Owner:** - Creation of the backlog. - Facilitates workshops with business teams. - Develops functional specifications for projects and updates them during development if necessary.
Mr.ZOUINE Safouane	**Scrum Master:** - Team management. - Organization and monitoring of work. - Ensuring daily project management. - Validating costs and deadlines.
Mr.HAMMOUCH El Mehdi	**Backend Developer**
Mrs.BEN ESSARRAKH kaoutar	**Backend Integrator (intern)**
Mr.OUTAMHA Amine	**Designer**
Mr.Mouad Ait OUGRRAM	**Frontend Developer (intern)**

II. Identification of Needs

In this section, we will describe all functional and non-functional requirements which our project should respond to

2.1 Functional requirements

The objective of the "Alma7al" project is to facilitate the overall management of various store resources (human resources, financial resources, goods resources, etc.) and to provide handy tools for day-to-day tasks. Additionally, it aims to offer an absolute view of the store's growth and serve as a digital twin of the store itself.

The main functional requirements of our application can be summarized in the following points:

Employee Management:

- Dedicated page for managing human resources (employees)
- Each store employee should have an account with assigned role-based permissions
- Ability to create/delete employee accounts
- Edit employee permissions for efficient task distribution
- Receive and act upon employee permission requests (accept or reject)

Expense Management:

- Visualize and monitor store expenses in real-time
- Provide detailed information on each expense, including comments, bill documents, and descriptions

Client Management:

- View clients' total expenses and total gains
- Real-time dashboard and charts to track client growth over time (year/month)
- Access detailed client information, such as total expenditure, net gain, loyalty points, trust level, online payments, cash payments, etc.

Product Catalog Management:

- Add new products and categorize them into categories and subcategories
- Provide product information, including model, description, color, initial quantity, stock quantity, price, brand, etc.

Communication:

- Real-time text/voice chat between employees for communication and file sharing

Profile/Store Settings:

- Change user information, such as name, password, profile image, and credentials
- Enable multi-factor authentication
- Access the history of all activities within the application for problem-handling and detection
- Generate bill documents by filling out a form

User Registration and Authentication:

- Implement a multi-channel authentication solution for various login options, such as email or one-time password (OTP) over SMS for added security

2.2 Non-functional requirements

These are the technical requirements describing the constraints that the system is subject to for its implementation and proper functioning. all extensions to be developed must meet the following needs:

Simplicity: Each user must be able to use the application intuitively.

Modularity and Scalability: It is crucial to adopt a modular and flexible approach in our system, using autonomous and easily replaceable components. This is particularly important when integrating new modules to meet evolving functional needs of the client. Additionally, the code must be simple, and easy to maintain and understand, in order to facilitate future adjustments and improvements.

Reliability: The results provided by the application must be reliable and accurately reflect the state of the database at the time of the query, i.e. during data updates.

Availability: Our application must be accessible at all times for all users, and it must be easily usable from any computer.

Security: Since our application processes personal and sensitive information, it must comply with computer system security standards. A secure access system based on authentication and password encryption is necessary.

Performance: Our system must meet this criterion while offering minimal response time using good development practices.

III. Global Use Case Diagram

The use case diagrams model the behavior of a system and help capture system requirements. They describe high-level functions and the scope of a system. These diagrams also identify interactions between the system and its actors. The following figure illustrates the overall use case diagram of the platform:

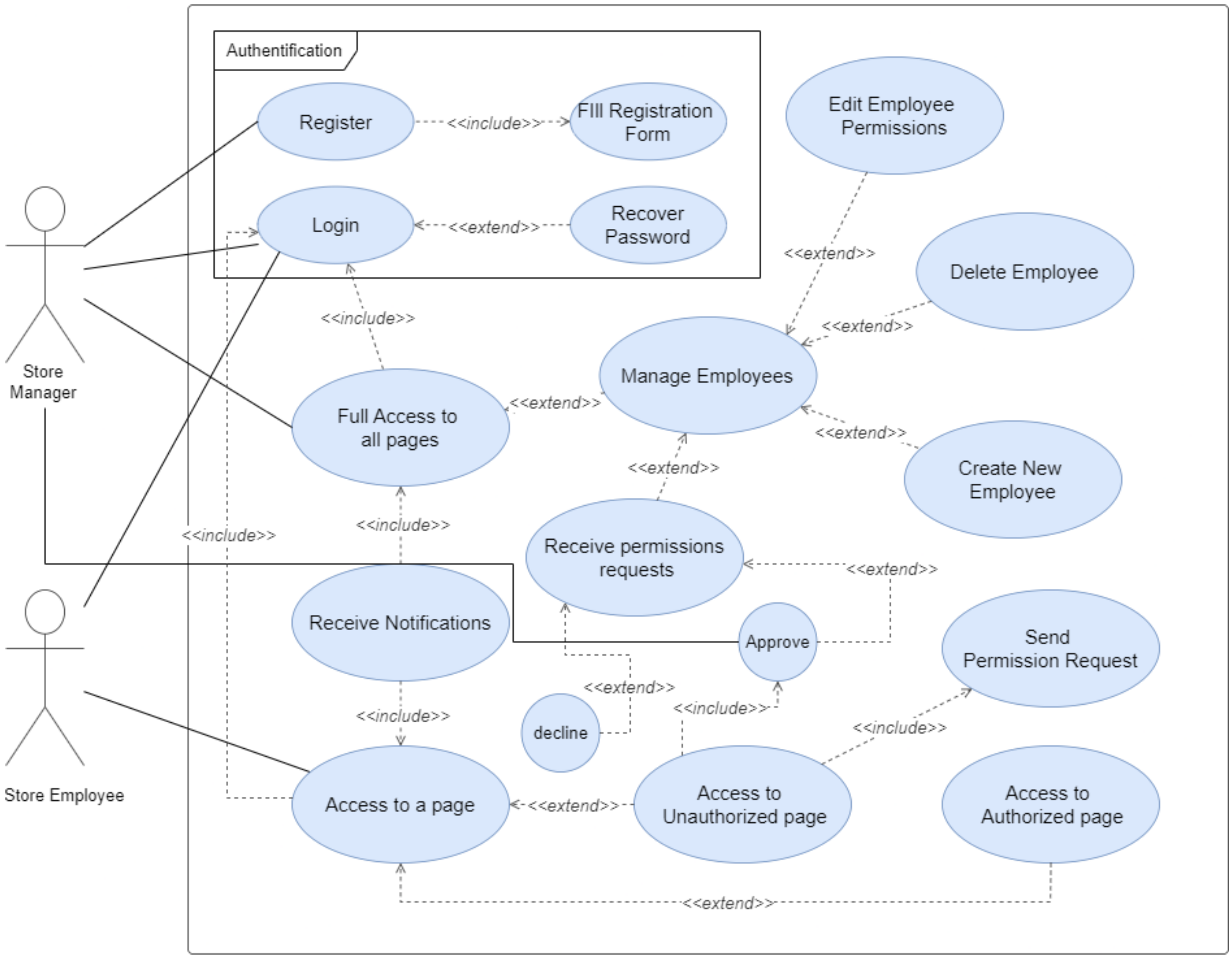


Figure 9 : Global Use Case Diagram

IV. Product Backlog

The Product Backlog is a set of deliverables to be completed that constitutes the queue of features to be accomplished during each sprint. It includes all tasks to be performed. The table below summarizes the tasks of the Product Backlog carried out during this project.

★★★★★ Five stars: high priority

★★★★☆ Four stars: very high priority

★★★☆☆ Three stars: high priority

★★☆☆☆ Two stars: medium priority

★☆☆☆☆ One star: low priority

The user stories were defined following the SMART axiom:

- Specific (simple, significant)
- Measurable (meaningful, motivating)
- Achievable (agreed, attainable)
- Relevant (reasonable, realistic)
- Time-bound (time-based, time-limited)

Table 2 : Product Backlog for the Project

User Story	Priority	Description
As a store manager, I want to manage employee accounts and permissions	★★★★★	- Implement a system to create, delete, and edit employee accounts with role-based permissions - Allow store managers to receive and act upon employee permission requests
As a store manager, I want to track and visualize store expenses in real-time	★★★★☆	- Develop an Expenses Page that displays store expenses in real-time - Provide detailed information on each expense, including comments, bill documents, and descriptions - Include relevant charts and visualizations
As a store manager, I want to manage client information	★★★★☆	- Display clients' total expenses, total gains, and other relevant variables - Implement real-time dashboards and charts to track client growth over time
As a store manager, I want to efficiently manage the product catalog	★★★★★	- Develop a Products Catalog Page that organizes products into domains, categories, and sub-categories - Include an explorer feature for easy navigation through the product catalog -Allow adding, updating, and removing products from the catalog
As an employee, I want to communicate with other employees in real-time through text and voice chat	★★★☆☆	- Implement a real-time communication system within the application -Enable text, image and voice chat functionality between employees with file-sharing
As a user, I want to manage my profile settings and access activity history	★★★☆☆	- Create a Profile/Store Settings page for users to manage their account information and access their activity history - Implement multi-factor authentication for enhanced security
As a new user, I want to register and authenticate securely using various login options	★★★★☆	-Implement multi-channel authentication, including email and OTP over SMS
As a store manager, I want to manage sponsorship activities and track revenue gained from sponsoring the application	★★☆☆☆	-Display revenue gained from sponsoring the application with options to invite others to use the application

V. Project Planification

5.1 Sprints Planification

The following table presents the project roadmap (SCRUM Roadmap) which describes the progression of the project sprints.

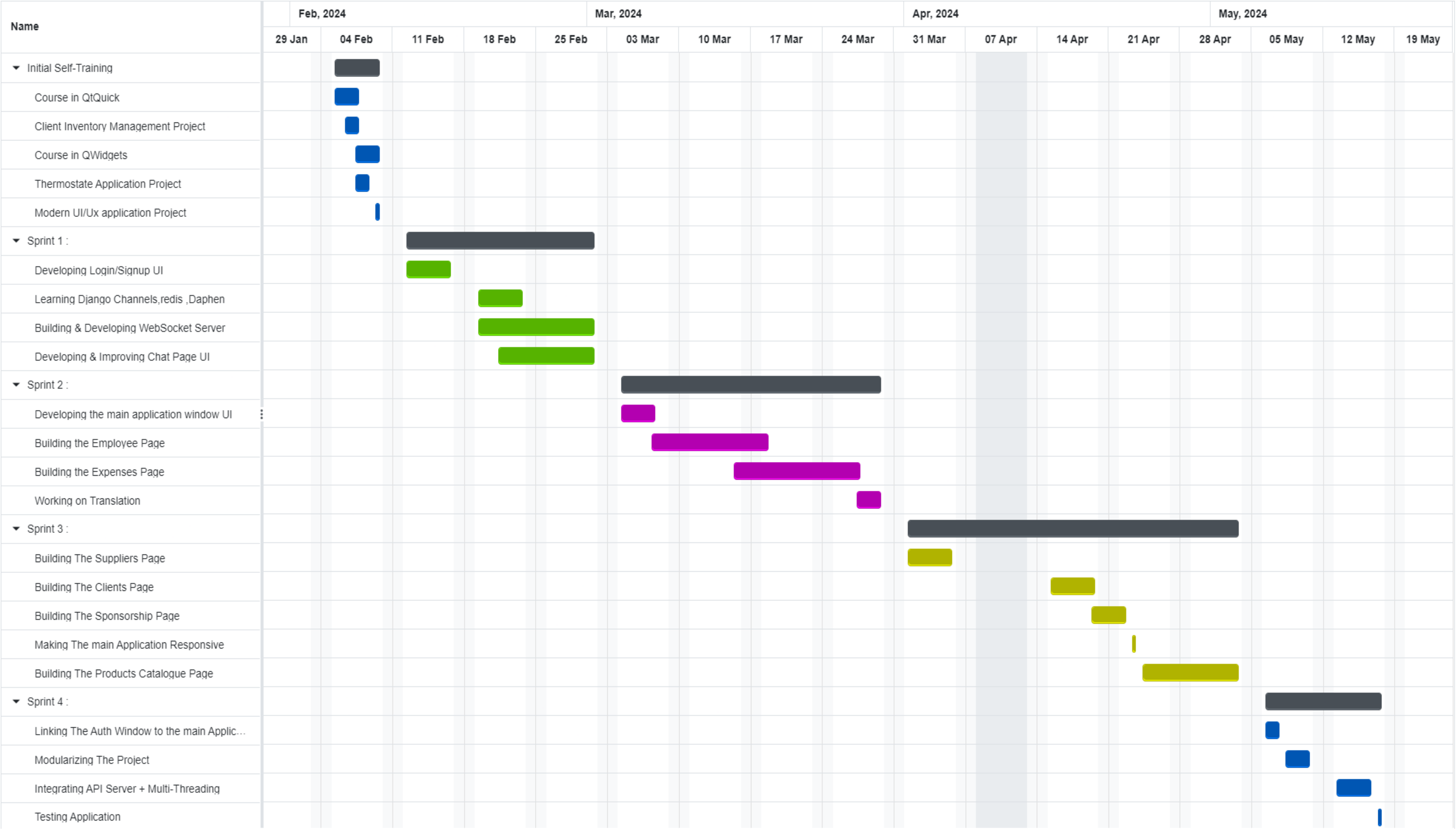
Table 3 : Sprints Planification Table

Sprint Planification		
Activity Name	Duration	Dates
Preparation Phase - analysis and specification of requirements - technical study -Initial Application Prototype	4 Months	2023/7 – 2024/02
Sprint 1: Building The Login/Signup + Develop real-time Chat Page	3 Weeks	12 Feb – 1 Mars
Sprint 2: Develop Employee Page and Expenses Page	4 Weeks	4 Mars – 29 Mars
Sprint 3: Develop Suppliers Page, Clients Page, Sponsorship Page, Product Catalog Page	4 Weeks	1 Apr – 3 May
Sprint 4: Integrate API communication for login/signup processes and employee page data management and CRUD operations	+ 2 Weeks	6 May - Ongoing

5.2 Gantt Chart

The Gantt chart in the figure below illustrates the progress of the internship over time:

Figure 10 : Gantt Chart



Conclusion

In this chapter, we have presented the various functional and non-functional requirements that the system must meet, as well as the use case diagram that has allowed us to identify the different features of the various modules. The next chapter is dedicated to the technical study.

Chapter III: Preparation Phase - Technical Study & Software Architecture

In this chapter, we will delve into the various technologies and tools used during the implementation, as well as the software architecture of our project.

I. Tools and Technologies Used

1.1 Frameworks and Development Technologies

a) The Back-end

	Django is a versatile web framework for Python that is widely used for developing web applications. It was created to help developers build web applications quickly and with less code. Django promotes rapid development and clean, pragmatic design. It is popular for creating scalable and secure web applications, and it includes an ORM (Object-Relational Mapping) system that simplifies database interactions.[21]
	Django Channels is an extension to Django that adds support for handling WebSockets, HTTP2, and other asynchronous protocols. It allows Django applications to handle real-time events, enabling features like live chat, notifications, and other interactive functionalities. Channels integrate seamlessly with Django and provide a way to manage asynchronous tasks efficiently.[3]
	Redis is an open-source, in-memory data structure store used as a database, cache, and message broker. It supports various data structures such as strings, hashes, lists, sets, and more. Redis is known for its high performance and flexibility, making it ideal

	for caching data to improve application speed and for managing real-time data streams.[4]
aws	AWS is the world’s most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. It provides a vast array of infrastructure technologies like computing, storage, and databases, as well as emerging technologies such as machine learning, artificial intelligence, data lakes, analytics, and the Internet of Things. AWS enables millions of customers, including the fastest-growing startups, largest enterprises, and leading government agencies, to lower costs, become more agile, and innovate faster. With its pay-as-you-go pricing, AWS allows businesses to scale their IT infrastructure in line with their needs, making it a cornerstone of modern IT strategies.[5]
python™	Python is a powerful, high-level programming language known for its simplicity and readability. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is widely used in web development, data analysis, artificial intelligence, scientific computing, and many other fields. Its extensive libraries and frameworks make it a popular choice for developers worldwide.[6]

b) The Front-end

Figure 11 : Qt technology logo

Qt is an important software toolkit for building applications that can run on different platforms like Windows, Mac, and Linux. It was created by a company called Digia (previously Trolltech). Qt allows developers to create powerful apps using the same codebase across various operating systems and devices. This is known as cross-platform development.

One of Qt's key advantages is providing a unified foundation for app development regardless of the underlying platform. Qt abstracts away the complex differences between operating systems and hardware. Developers can write code once and deploy it on multiple platforms with minimal changes. This cross-platform compatibility is made possible by Qt's use of advanced C++ techniques like the D-Pointer pattern.[7]

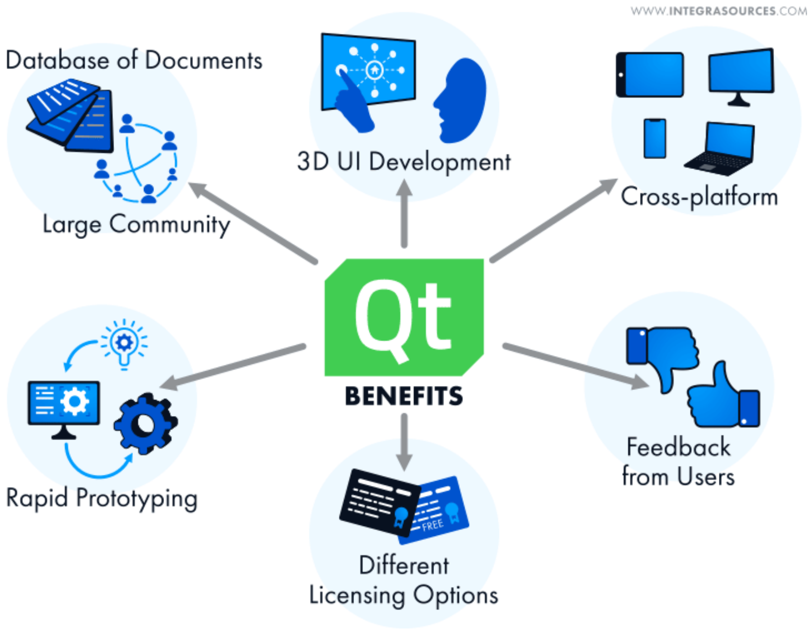


Figure 12 : Benefits of utilizing Qt technology

The Event Loop and Events in Qt

At the heart of every Qt app is the event loop, represented by the QEventLoop class. The event loop manages the flow of the application by continuously checking for and dispatching events to the appropriate event handlers. Events can originate from user interactions (e.g., mouse clicks, keyboard input), system notifications, or internal application signals.

The QCoreApplication class is the entry point for all Qt apps and plays a crucial role in managing events. It provides a centralized event loop and ensures that events are properly sent to the correct objects within the application. Understanding the event loop and event handling is essential for building responsive and well-behaved Qt apps.[8]

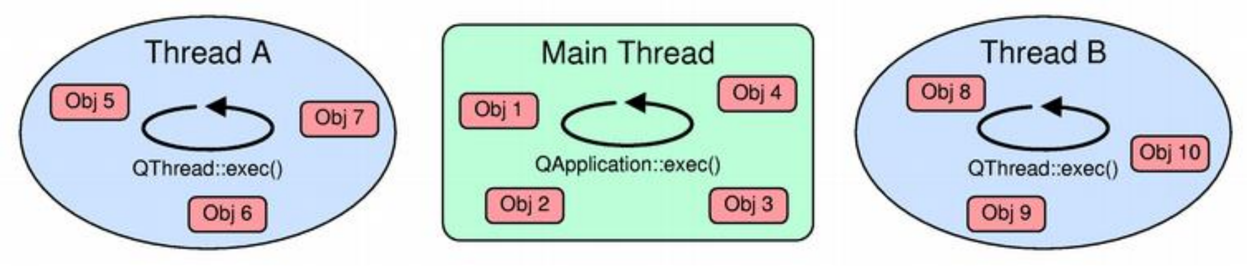


Figure 13 : Threads in Qt

Cross-Platform Adaptability with Qt

Qt's cross-platform design is deeply ingrained in its architecture and implementation. The framework uses advanced C++ techniques to ensure seamless adaptation across different platforms. One notable technique is the D-Pointer pattern, also known as the "Opaque Pointer" or "Pimpl" (Pointer to Implementation) idiom.

The D-Pointer pattern separates the public interface of a class from its private implementation details. This separation allows Qt to maintain platform-specific implementations while exposing a consistent API to developers. By encapsulating the platform-specific code behind an opaque pointer, Qt can swap out the underlying implementation without affecting the public interface, thereby achieving binary compatibility across different platforms.[9]

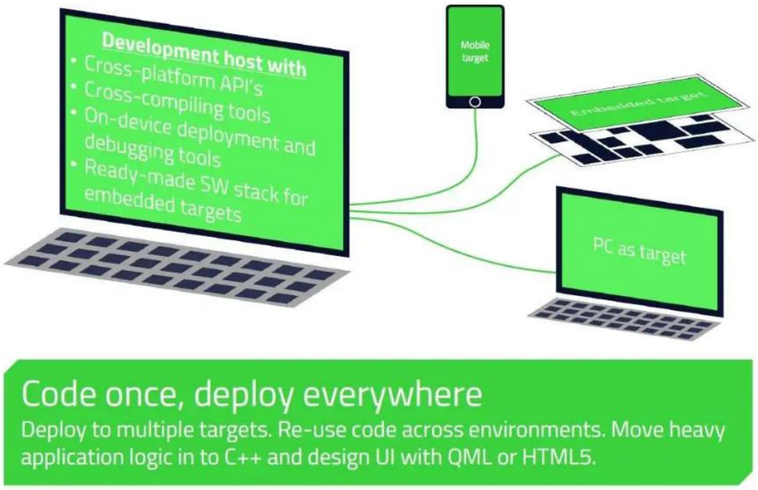

Development host with
• Cross-platform API's
• Cross-compiling tools
• On-device deployment and debugging tools
• Ready-made SW stack for embedded targets
Mobile target
Embedded target
PC as target
Code once, deploy everywhere
Deploy to multiple targets. Re-use code across environments. Move heavy application logic in to C++ and design UI with QML or HTML5.

Figure 14 : Qt's ability for crossplatform development

Signals vs. Events

While signals and events are both mechanisms for inter-object communication in Qt, they serve different purposes and originate from different sources.

Signals are notifications emitted by QObject-derived objects to indicate that a particular event or state change has occurred within the object. Other objects can connect to these signals using slots, which are essentially callable functions or methods that respond to the signal. Signals are typically used for inter-object communication within the same application or across different components of the application.

Events, on the other hand, typically originate from the operating system or external sources, such as user input or system notifications. Events are handled by event handlers, which are special methods or functions that process the specific event type. Events are often used to handle user interactions, system events, or other external stimuli that require a response from the application.[10]

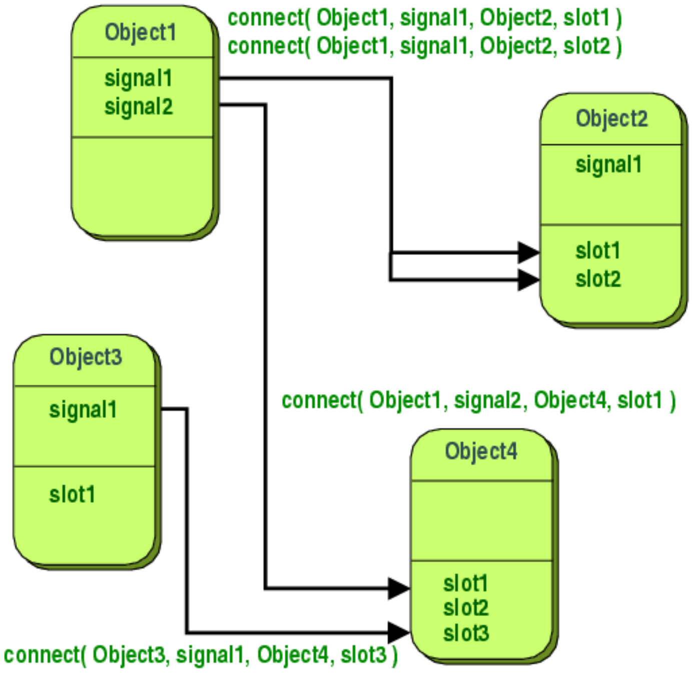

connect(Object1, signal1, Object2, slot1)
connect(Object1, signal1, Object2, slot2)
Object1
signal1
signal2
Object2
signal1
slot1
slot2
Object3
signal1
slot1
connect(Object1, signal2, Object4, slot1)
Object4
slot1
slot2
slot3
connect(Object3, signal1, Object4, slot3)

Figure 15 : Signals and Slots concept in Qt

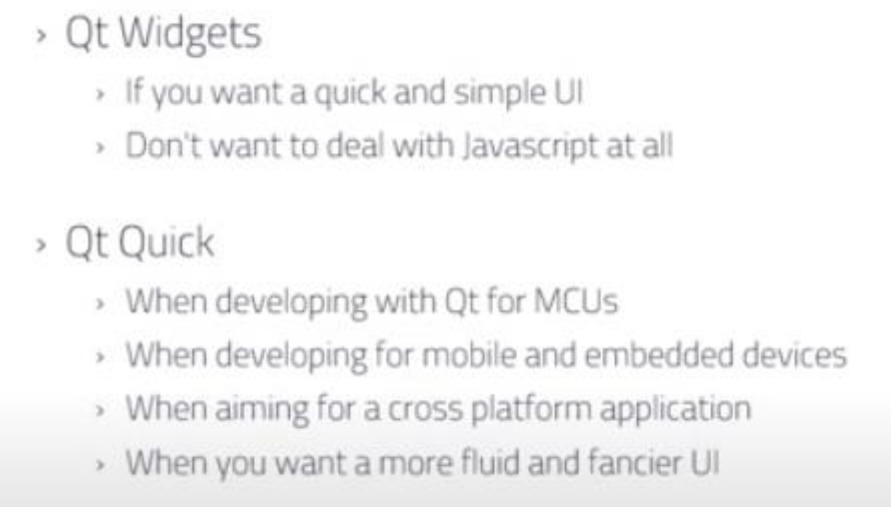
Graphical User Interfaces in Qt: QWidget vs. QML

Qt offers two different approaches for building graphical user interfaces (GUIs): the traditional QWidget approach and the more modern Qt Modeling Language (QML).

QWidget is a widget-based approach that focuses on pixel painting and low-level control over UI elements. It provides a rich set of widgets, such as buttons, menus, and dialogs, which can be composed to create complex user interfaces. QWidget is based on the concept of inheritance and subclassing, allowing developers to create custom widgets by inheriting from existing ones.

In contrast, QML is a declarative language that separates the UI definition from the underlying application logic. It is based on a rendering engine that optimizes the UI for high performance and smooth animation. QML leverages Qt's underlying rendering and scene graph infrastructure, providing a more modern and flexible approach to UI design.

While QWidget excels in scenarios that require fine-grained control over UI elements and low-level customization, QML offers a more intuitive and efficient way to create modern, fluid, and visually appealing user interfaces. The choice between QWidget and QML often depends on the specific requirements of the application, the target platform, and the development team's preferences and expertise.[11]



- Qt Widgets
  - If you want a quick and simple UI
  - Don't want to deal with Javascript at all
- Qt Quick
  - When developing with Qt for MCUs
  - When developing for mobile and embedded devices
  - When aiming for a cross platform application
  - When you want a more fluid and fancier UI

Figure 16 : Qt Widgets vs Qt Quick

PyQt vs PySide for Qt6

Qt also provides bindings for the Python programming language, allowing developers to create Qt applications using Python instead of C++. There are two main choices for Python bindings: PyQt and PySide.

Figure 17 : PyQt vs PySide

PyQt is the older and more established binding, developed by the Qt company itself (previously Trolltech/Digia). PyQt provides a comprehensive set of Python bindings for the entire Qt framework, enabling developers to access the full range of Qt's features and functionality from within Python code.

PySide, on the other hand, is a more recent project developed by the Qt Company (previously Nokia). It aims to provide an alternative set of Python bindings for Qt, with a focus on being more lightweight and easier to integrate with other Python libraries and tools.

Both PyQt and PySide offer similar functionality and allow developers to create cross-platform GUI applications using Qt's powerful UI tools and libraries. However, there are some key differences between the two:

- **Licensing:** PyQt uses a commercial license model, where developers need to purchase a license for commercial use. PySide, on the other hand, is released under the LGPL open-source license, making it free to use in both open-source and commercial projects.
- **API Compatibility:** PyQt aims to closely mirror the original C++ API of Qt, providing a more direct translation to Python. PySide takes a more pythonic approach, adapting the Qt API to better fit Python's conventions and idioms.
- **Community and Support:** PyQt has a larger and more established community, with more resources, tutorials, and third-party tools available. PySide, being a newer project, has a smaller but growing community.
- **Performance:** Both PyQt and PySide offer similar performance characteristics, as they both rely on Qt's underlying C++ libraries and APIs.

Used Technologies:

	PySide6 is the latest version of the PySide library, which provides Python bindings for the Qt framework. It allows developers to create desktop applications using Python instead of C++, while still taking advantage of Qt's powerful features and cross-platform capabilities. With PySide6, you can access and use all the Qt classes and modules from within your Python code. This includes everything from creating graphical user interfaces (GUIs) to handling network communication, databases, and more. PySide6 makes it easy to integrate Python and Qt, enabling you to leverage the strengths of both technologies in your desktop application development.[12]

	QML, or Qt Modeling Language, is a declarative language used for designing user interfaces in Qt applications. It separates the UI definition from the underlying application logic, making it easier to create modern, fluid, and visually appealing user interfaces. In your PySide6 desktop application, QML is likely used for defining the structure, layout, and appearance of the user interface. This includes elements such as windows, buttons, menus, and other UI components. QML files are written in a declarative syntax, which means you describe the UI elements and their properties, rather than coding them procedurally. This can make UI development more efficient and easier to maintain.[13]
	While QML is primarily used for defining the user interface, it also supports the use of JavaScript for adding interactivity and logic to your UI components. JavaScript code can be embedded directly into QML files, allowing you to define properties, functions, and event handlers that control the behavior of your UI elements. For example, you might use JavaScript in QML to: - Perform calculations or data manipulations - Respond to user interactions (e.g., button clicks, text input) - Communicate with the underlying PySide6/Qt application logic

	By combining QML's declarative UI definition with JavaScript's scripting capabilities, you can create rich, dynamic, and interactive user interfaces for your PySide6 desktop application.[14]

1.1 Modelling tools and technologies

draw.io	draw.io is a versatile web-based platform designed for creating and sharing diagrams and flowcharts. It offers a wide range of diagram types, from organizational charts to network diagrams, and integrates seamlessly with various storage platforms and applications. With its intuitive drag-and-drop interface and extensive shape libraries, draw.io simplifies the process of visualizing complex information.[15]
	Figma is a cloud-based design tool that revolutionizes the way teams collaborate on design projects. It's a comprehensive solution for UI/UX design, enabling real-time collaboration, prototyping, and code inspection. Figma's user-friendly interface and extensive plugin library make it an indispensable tool for modern designers.[16]

1.2 Project Management tools and technologies

git	Git is a distributed version control system that facilitates tracking changes in source code during software development. It supports non-linear development through its branching and merging capabilities, making it ideal for coordinating work among programmers.[17]
GitHub	GitHub is an online platform and service that provides a version control system for developers and teams to collaborate on software development projects. It allows developers to store, manage, and track changes to their source code, making it easier to work together on projects and manage the history of changes.[18]
Google Meet	Google Meet is a secure video communication service developed by Google. It’s designed for business and office use, allowing colleagues to conduct video meetings and collaborate effectively. Google Meet stands out for its ease of use and integration with other Google services.[19]

1.3 Development Tools

	Visual Studio Code is a free, open-source code editor developed by Microsoft. It supports a variety of programming languages and comes with features like debugging, intelligent code completion, snippets, and code refactoring. VSCode's customization options and extension marketplace cater to a wide range of development needs.[20]

II. General Architecture

2.1 Overall Software Architecture of the Project

The following figure represents the overall architecture of the project

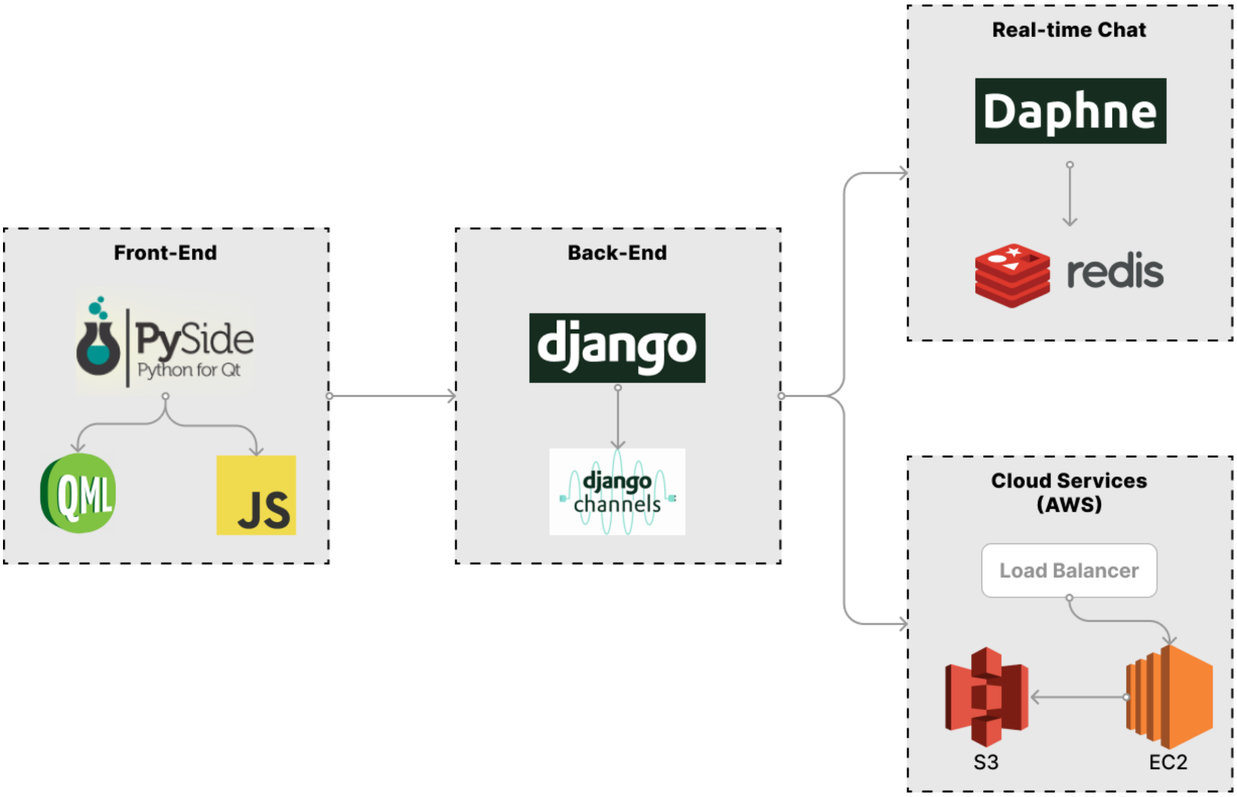


Figure 18 : Project Architecture Diagram

How the Technologies Interact

Django for Backend Management

Django serves as the backbone of the application, handling all the CRUD (Create, Read, Update, Delete) operations necessary for managing store data. This includes managing products, inventory, orders, and user information. Django's structured framework ensures that these operations are secure, efficient, and easy to maintain.

Django Channels and Real-Time Chat

Django Channels extends the functionality of Django to handle asynchronous tasks, including the real-time chat feature. When users engage with the chat module, Django Channels manages WebSocket connections, allowing messages to be sent and received instantly. This real-time interaction is facilitated by Daphne, an ASGI server that manages these WebSocket connections efficiently.

Redis as a Message Broker

Redis plays a crucial role in the smooth operation of real-time communications. It acts as a message broker, ensuring that messages sent from one user are correctly routed to the intended recipient. Redis handles this task swiftly, maintaining the performance and responsiveness of the chat feature even under heavy loads.

AWS for Hosting

The entire backend, including Django and its extended functionalities, is hosted on AWS. AWS provides a reliable and scalable environment, ensuring that the application can handle high traffic volumes and large amounts of data. This setup not only offers robust performance but also incorporates security features to protect sensitive store data.

Interaction Flow

When a user performs an action in the store manager application, such as adding a new product, Django processes this request and updates the database. For chat interactions, when a user sends a message, Django Channels handles the WebSocket connection, Redis routes the message to the convenient Group/Channel, and Daphne ensures the connection remains open and efficient.

In summary, Django manages the core functionalities and data of the application, Django Channels and Daphne enable real-time communication, Redis ensures message delivery, and AWS provides a secure and scalable hosting environment. Together, these technologies create a powerful and efficient store manager application with an integrated chat feature.

Conclusion

In this chapter, we presented the various modules and technologies used in our application, as well as the software architecture of the project. In the next chapter, we will discuss the Sprint Progress and Achievements.

Chapter IV: Sprint Progress and Achievements

This chapter outlines the key tasks and accomplishments for each scrum sprint, highlighting our progress in building the authentication UI, real-time chat features, various application pages, and integrating API communication with CRUD operations. Each sprint reflects our collaborative efforts and technical advancements.

I. Sprint 1: Building The Login/Signup UI & Development of real-time Chat Page

In this sprint, several key tasks were accomplished:

First, all UI components related to the authentication process were built, including fields, buttons, popups, and pages. These components were designed with smooth animations to enhance the user experience.

Next, the entire chat page was developed with comprehensive features. This includes customized chat bubbles for image and voice messages, a recording panel for voice messages, and a panel to view the current online users.

Additionally, a WebSocket server was built using Django Channels, Daphne, and Redis. for enabling real-time communication within the chat application.

1.1 Chat Backend Architecture:

Integrating a chat page into the application is extremely beneficial, as it facilitates direct communication between employees. This improves overall productivity and collaboration within the store. To achieve this, I developed a chat server using Django Channels, Daphne, and Redis, as illustrated by the architecture diagram below.[21]

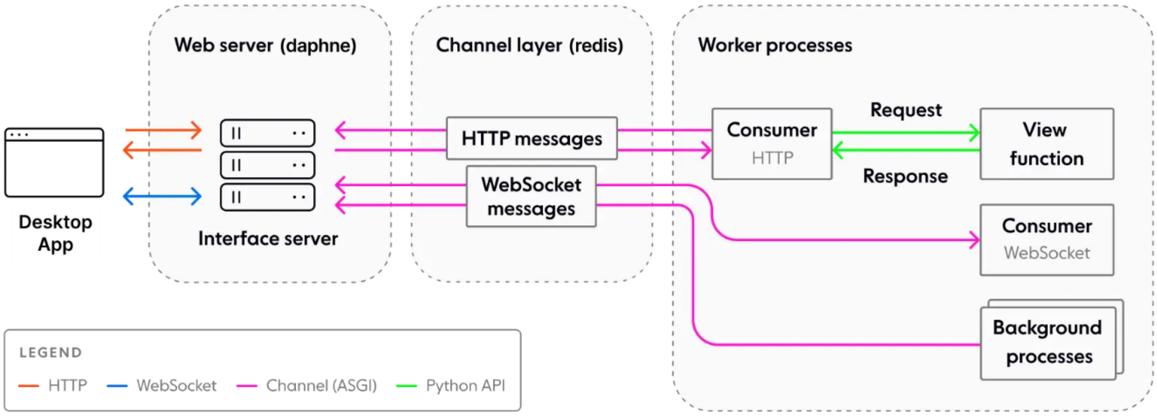


Figure 19 : basic Django Channels architecture for Chat

When a user starts a connection, such as opening a chat, the ASGI server (Daphne) manages this request. The server determines the type of connection (either HTTP or WebSocket) and routes it accordingly.

For WebSocket connections, the server forwards the request to a WebSocket consumer. This consumer accepts the connection and sets up the WebSocket session, allowing for real-time communication.

When the user sends a message through the WebSocket, it is received by the WebSocket consumer. The consumer processes the message and may use the channel layer to broadcast the message to other users. The channel layer, backed by Redis, manages the message passing between different parts of the application, ensuring that all relevant consumers receive the message.

1.2 Initial Frontend Architecture

During this sprint, the application was divided into two separate PySide6 projects. One project is dedicated solely to the login/signup UI, while the second project represents the main "Alma7al" application. This division showcases the modularity and abstraction of the overall application. At this stage, the two applications do not interact with each other, so each has its own QApplication() and QQmlApplicationEngine(). And follows this basic entity architecture:

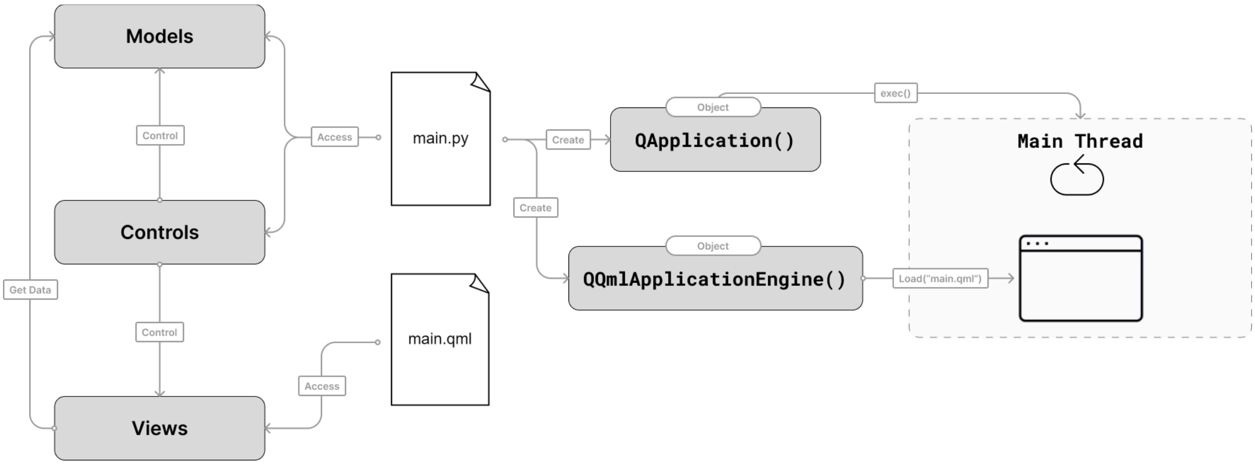


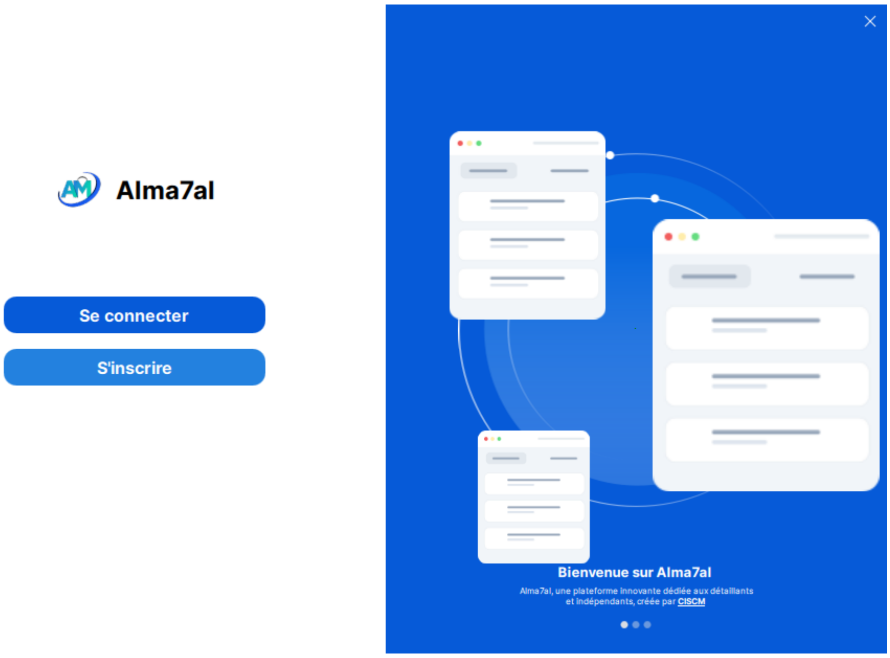
Figure 20 : Initial Frontend Architecture

This architecture consists of a structured folder system with models, controls, and views. The models and controls folders contain Python classes, while the views folder contains QML files. The entry points are two main files: `main.py` and `main.qml`. The `main.py` file instantiates `QApplication` and `QQmlApplicationEngine`,which loads the `main.qml` file, and starts the main event loop with the `exec()` method. The `main.qml` file contains the `ApplicationWindow` and serves as the starting point, invoking other views from the views folder.

the `main.py` file calls classes in the models and controls folders, while `main.qml` calls views in the views folder, which can further call other QML files. This organization allows for a clear separation of concerns, making the code more manageable.

1.3 Overview

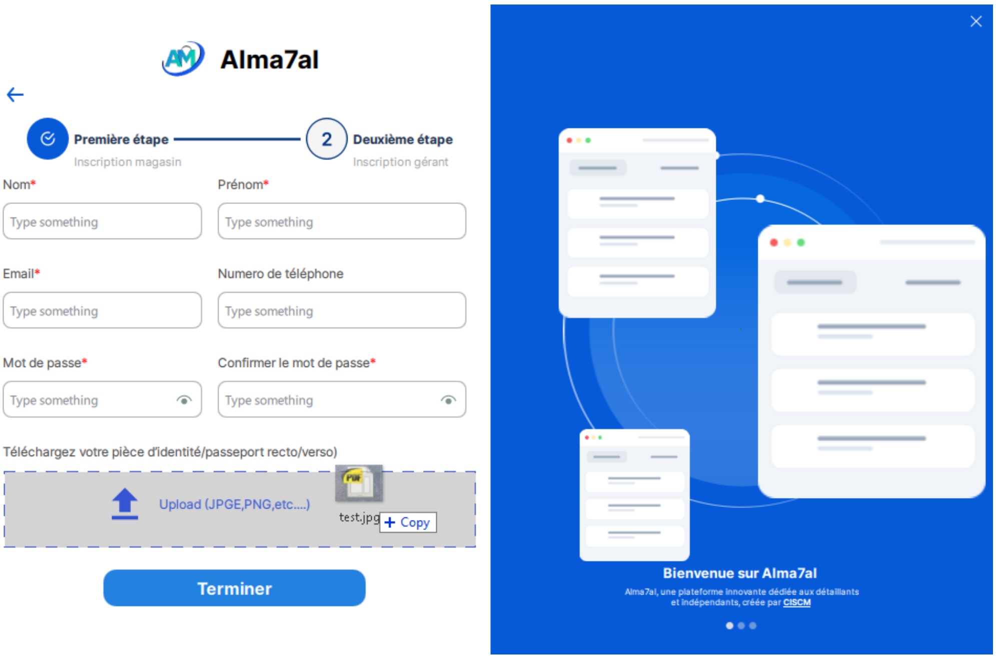
In this section, we present images showcasing various aspects of the application, providing a visual representation of its current state and features.



Bienvenue sur Alma7al

Alma7al, une plateforme innovante dédiée aux détaillants et indépendants, créée par CISCM

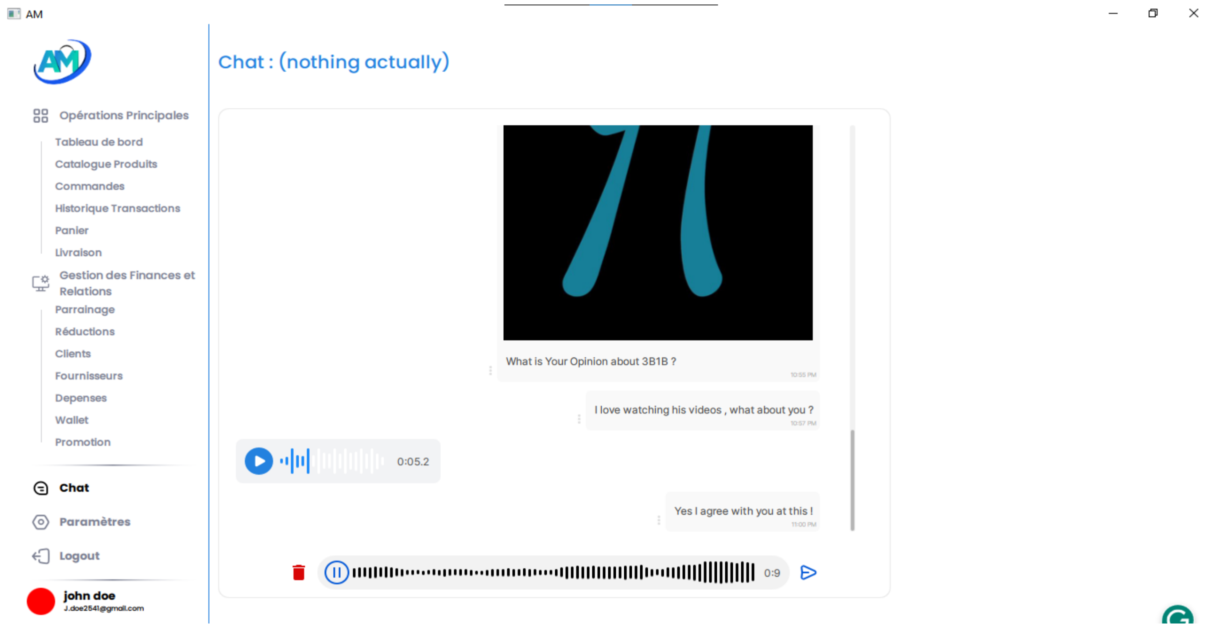
Figure 21:login/signup window



Bienvenue sur Alma7al

Alma7al, une plateforme innovante dédiée aux détaillants et indépendants, créée par CISCM

Figure 22:Registration Form

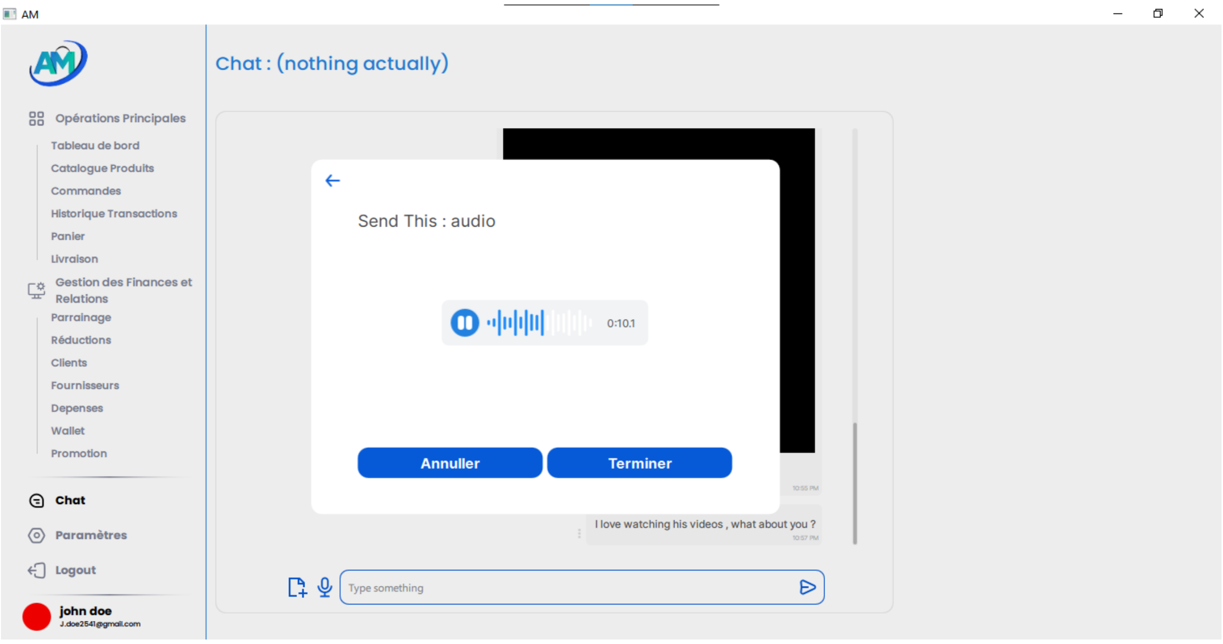


What is Your Opinion about 3B1B ?

I love watching his videos , what about you ?

Yes I agree with you at this !

Figure 23: Recording Voice in the chat page



Send This : audio

I love watching his videos , what about you ?

Figure 24:Sending Image/Audio in the chat

1.4 Conclusion

In this sprint, the application has taken its initial steps towards development and currently consists of two separate PySide6 applications. In the next sprint, we plan to merge the two applications so that once authentication is successful, the login application will close, and the main application, including the chat, will open.

II. Sprint 2 and 3: Development of the Application’s Pages

During this sprint, our objective was to develop and configure as many pages as possible. We successfully built the Employee Page, Expenses Page, Suppliers Page, Clients Page, Sponsorship Page, and Product Catalog Page. And we successfully merged the Login/signup project with the application respecting the application modularity and facilitating the error detection. Throughout this development journey,I discovered new aspects of PySide6 and expanded my knowledge significantly. This experience has left me with enough expertise to confidently design and build any software using PySide6 and Qt6.

2.1 Frontend Architecture

In the last sprint, the application was divided into two standalone pyside6 applications with separate Qapplication() instance and separate QqmlApplicationEngine, so the solution that I came up with to merge the two separate applications, is represented in this diagram.

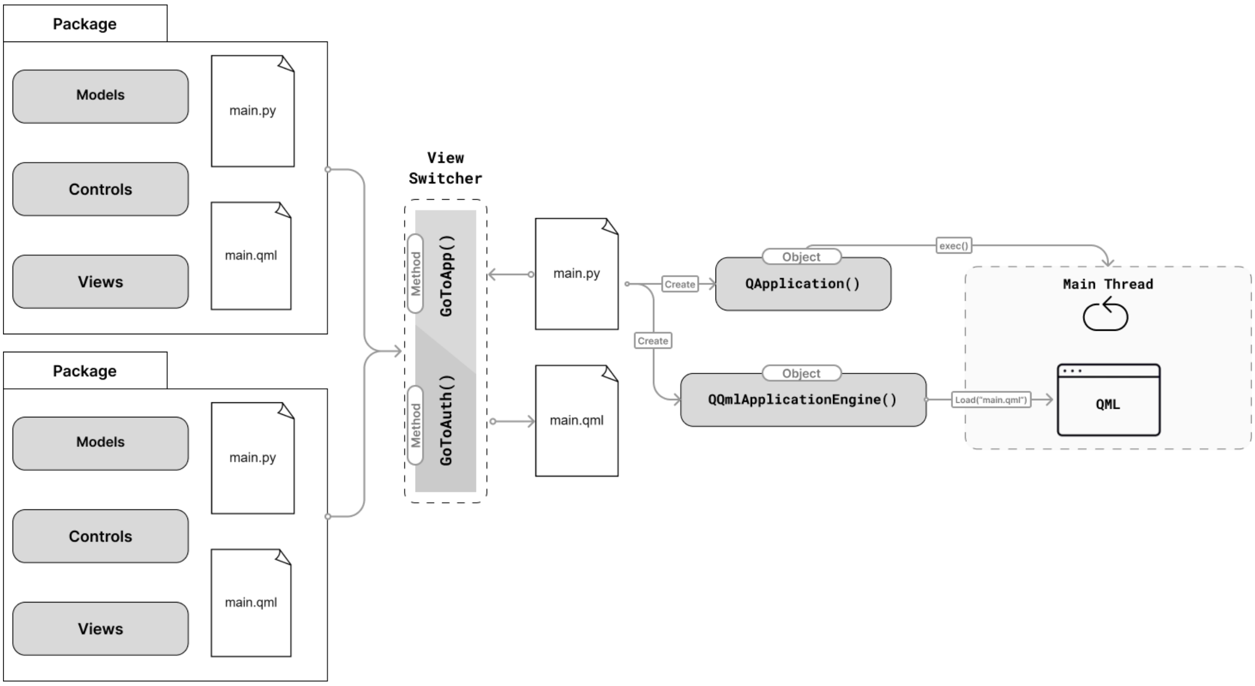


Figure 25 : Frontend Architecture of the "Alma7al"

To merge these two applications under a single `QApplication` instance and `QQmlApplicationEngine` instance, I created a new global structure. I removed the instantiation of `QApplication` and `QQmlApplicationEngine` from both applications and placed them in a new global folder containing global `main.py` and global `main.qml`.

In this global setup, I preserved the entity architecture for both applications, now treated as packages. The global `main.py` file instantiates `QApplication` and `QQmlApplicationEngine` and loads the global `main.qml` file. It includes a variable `window` -*that represents the current window or package*- to switch between package to package (representing the login/signup and store applications, respectively).

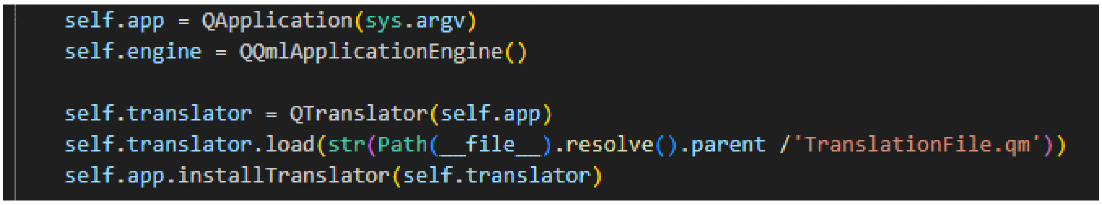
The global `main.qml` file uses a `Loader` with a `currentview` variable to dynamically load the appropriate view based on the current package. When switching between packages, the constructor of packages updates `currentview` to the path of the corresponding `main.qml` file. This setup allows seamless switching between the login/signup and store applications.

To call model or controller classes from the models or controls folders, I reference the `window`, which is an instance of either of the two Packages. This maintains a unified `QApplication` and `QQmlApplicationEngine` while preserving modularity and ensuring each package can access its own MVC structure

2.2 Translation in Pyside6

Since Our application targets retailers in Casablanca, there is a necessity to integrate translation in the application for 3 languages at least (Arabic, French, and Tamazight), To implement translation in a PySide6 application, we use `.ts` (translation source) and `.qm` (compiled translation) files. First, we generate a `.ts` file that contains the text elements to be translated. This can be done using the `pyside6-lupdate` tool, which scans our source code and extracts the text for translation. For example, running `pyside6-lupdate main.py -ts example_de.ts` will create a `example_de.ts` file for Arabic translations. Translators can then edit this `.ts` file using a tool like Qt Linguist to provide translations.

Once the `.ts` file is translated, we compile it into a `.qm` file using the `pyside6-lrelease` tool. The command `pyside6-lrelease example_de.ts -qm example_de.qm` will generate the `example_de.qm` file. In Our PySide6 application, you load this `.qm` file using a `QTranslator` object. For example:



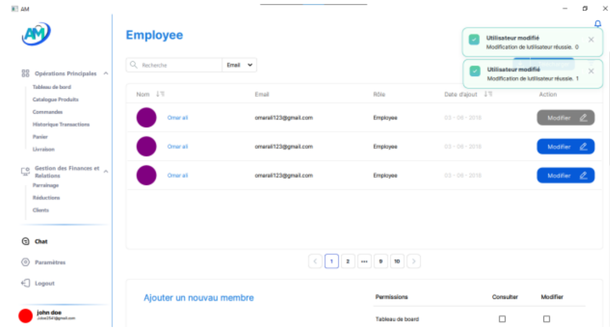
```
self.app = QApplication(sys.argv)
self.engine = QQmlApplicationEngine()

self.translator = QTranslator(self.app)
self.translator.load(str(Path(__file__).resolve().parent /'TranslationFile.qm'))
self.app.installTranslator(self.translator)
```

Figure 26 : Integration of Translation in PySide6

2.3 Overview

a) Employees Page:

The Employees Page is designed to help manage the human resources of the store. On this page, the store manager can assign tasks to employees by granting them specific permissions within the application. It also allows for the creation of employee profiles, where job roles and work history are stored.



Utilisateur modifié
Modification de lutilisateur réussie. 0

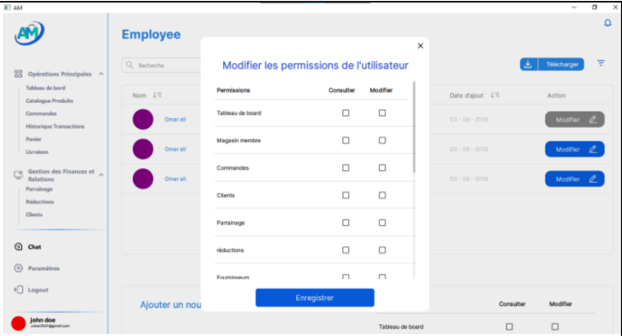
Utilisateur modifié
Modification de lutilisateur réussie. 1

Nom	Email	Rôle	Date d'ajout	Action
Omar ali	omarali123@gmail.com	Employee	03 - 06 - 2018	Modifier
Omar ali	omarali123@gmail.com	Employee	03 - 06 - 2018	Modifier
Omar ali	omarali123@gmail.com	Employee	03 - 06 - 2018	Modifier

Ajouter un nouvau membre

Permissions	Consulter	Modifier
Tableau de board	□	□

Figure 27 : Employee page



Modifier les permissions de l'utilisateur

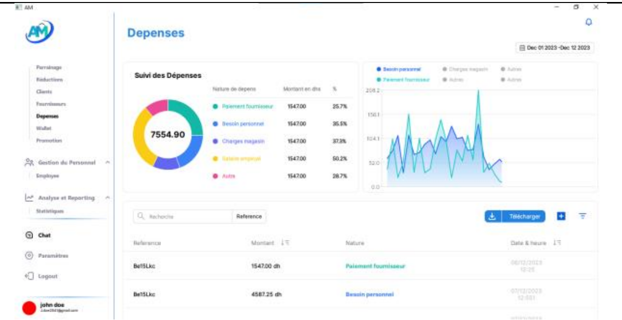
Permissions	Consulter	Modifier
Tableau de board	□	□
Magasin membre	□	□
Commandes	□	□
Clients	□	□
Parrainage	□	□
réductions	□	□
Fournisseurs	□	□

Enregistrer

Figure 28 : modify Employee permissions

b) Expenses Page

The Expenses Page is dedicated to monitoring the store's expenses. It allows you to track where money is spent and how much is spent. The page features relevant charts to visualize spending patterns over the year or month. You can see the breakdown of expenses into categories such as Supplier payments, personal needs, store charges, employee salaries, and other costs. This page helps identify trends and optimize budgeting by providing insights into how funds are allocated and highlighting areas for potential savings.

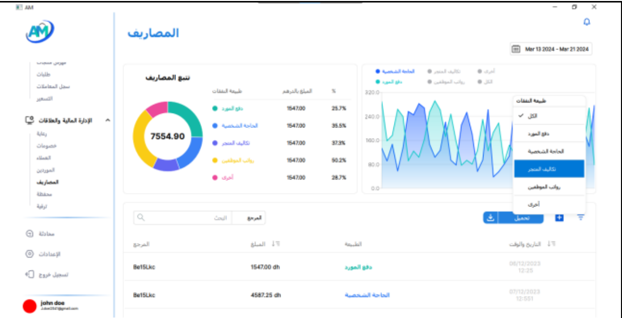


Suivi des Dépenses

Nature de depens	Montant en dhs	%
Paiement fournisseur	1547.00	25.7%
Besoin personnel	1547.00	35.5%
Charges magasin	1547.00	37.3%
Salaire employé	1547.00	50.2%
Autre	1547.00	28.7%

Reference	Montant	Nature	Date & heure
Be15Lkc	1547.00 dh	Paiement fournisseur	08/12/2023 12:25
Be15Lkc	4587.25 dh	Besoin personnel	07/12/2023 12:551

Figure 29 : Expenses Page



تتبع المصاريف

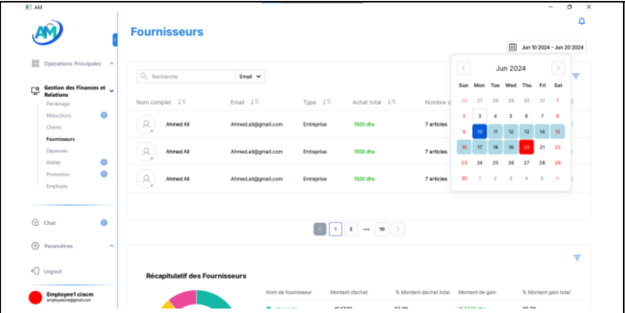
طبيعة النفقات	المبلغ بالدرهم	%
دفع المورد	1547.00	25.7%
الحاجة الشخصية	1547.00	35.5%
تكاليف المتجر	1547.00	37.3%
رواتب الموظفين	1547.00	50.2%
أخرى	1547.00	28.7%

المرجع	المبلغ	الطبيعة	التاريخ والوقت
Be15Lkc	1547.00 dh	دفع المورد	08/12/2023 12:25
Be15Lkc	4587.25 dh	الحاجة الشخصية	07/12/2023 12:551

Figure 30 : Expenses page in Arabic

c) Suppliers Page

The Suppliers Page provides detailed information about your store's suppliers. It includes data on purchases made from each supplier, including cash transactions. The page offers visualizations, such as pie charts, to help you understand the distribution of your spending among different suppliers. This page helps build strong relationships with suppliers by keeping all relevant information organized and accessible.

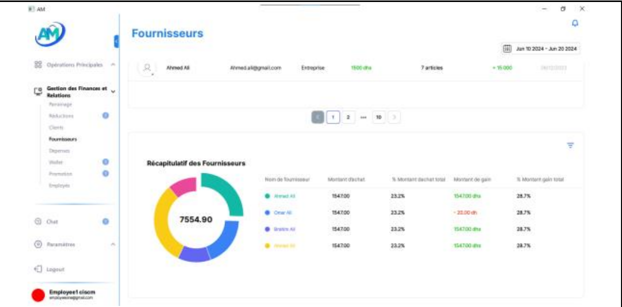


Nom complet	Email	Type	Achat total	Nombre d
Ahmed Ali	Ahmed.ali@gmail.com	Entreprise	1500 dhs	7 articles
Ahmed Ali	Ahmed.ali@gmail.com	Entreprise	1500 dhs	7 articles
Ahmed Ali	Ahmed.ali@gmail.com	Entreprise	1500 dhs	7 articles

Récapitulatif des Fournisseurs

Nom de fournisseur	Montant dachat	% Montant dachat total	Montant de gain	% Montant gain total

Figure 31 : Suppliers page



Ahmed Ali	Ahmed.ali@gmail.com	Entreprise	1500 dhs	7 articles	+ 15 000	[illegible]

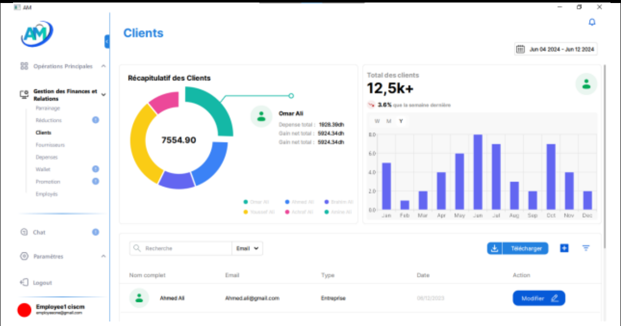
Récapitulatif des Fournisseurs

Nom de fournisseur	Montant dachat	% Montant dachat total	Montant de gain	% Montant gain total
Ahmed Ali	1547.00	23.2%	1547.00 dhs	28.7%
Omar Ali	1547.00	23.2%	- 20.00 dh	28.7%
Brahim Ali	1547.00	23.2%	1547.00 dhs	28.7%
Ahmed Ali	1547.00	23.2%	1547.00 dhs	28.7%

Figure 32 : recapitulation of Suppliers

d) Clients Page

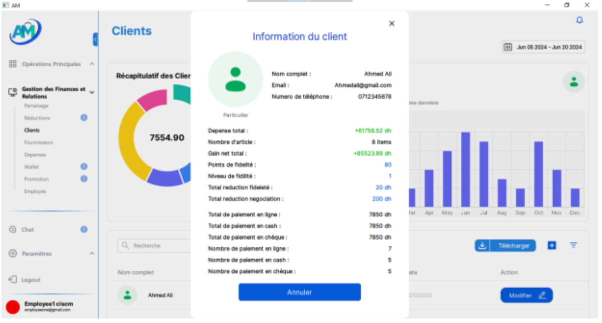
The Clients Page is dedicated for managing information about your store's clients. It tracks how much each client spends and includes a wide range of variables such as purchase history and fidelity points. The page allows you to analyze client behavior and preferences, enabling personalized marketing and loyalty programs. You can segment clients based on their purchasing patterns and tailor promotions to different groups. This page helps enhance customer satisfaction and drive sales through targeted engagement.



Omar Ali
Depense total : 1928.39dh
Gain net total : 5924.34dh
Gain net total : 5924.34dh

Nom complet	Email	Type	Date	Action
Ahmed Ali	Ahmed.ali@gmail.com	Entreprise	08/12/2023	Modifier

Figure 33 : Clients page Dashboard



Information du client

Nom complet : Ahmed Ali
Email : Ahmedali@gmail.com
Numero de téléphone : 0712345678
Particulier

Depense total :	+61756.52 dh
Nombre d'article :	8 items
Gain net total :	+85523.89 dh
Points de fidelité :	80
Niveau de fidlité :	1
Total reduction fidéleté :	20 dh
Total reduction negociation :	200 dh
Total de paiement en ligne :	7850 dh
Total de paiement en cash :	7850 dh
Total de paiement en chèque :	7850 dh
Nombre de paiement en ligne :	7
Nombre de paiement en cash :	5
Nombre de paiement en chèque :	5

Figure 34 : Add a client

e) Sponsorship Page

The Sponsorship Page focuses on sponsorship activities and shows information about the sponsees. It details how much revenue is gained from sponsoring the application and provides options to invite others to use the application., making it easier to attract and retain sponsors.

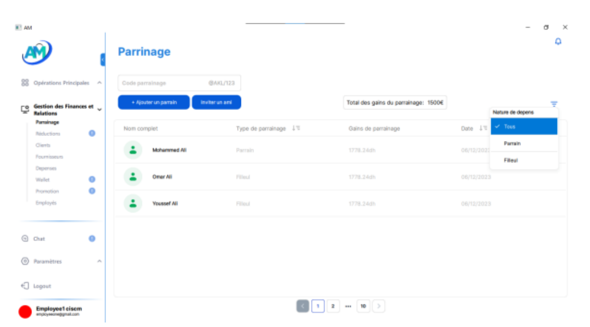


Figure 35: Sponsorship Page

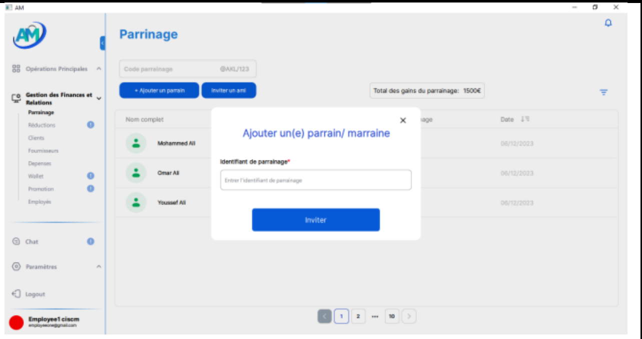


Figure 36 : Adding a sponsor

f) Products Catalog Page

The Products Catalog Page is dedicated to organizing and managing the products in the store. Products are categorized into domains, categories, and sub-categories for easy navigation. The page includes an explorer feature that simplifies browsing through the product catalog. It provides current information about products, including availability, size, specifications, and cost. This page helps keep the inventory up-to-date and ensures that staff can quickly find and access product details. It also supports adding new products, updating them, and removing discontinued items.

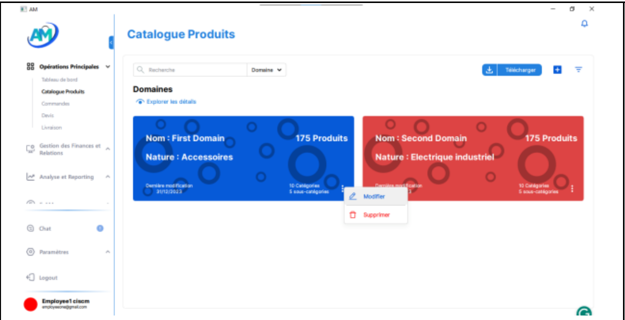


Figure 37 : Product Catalog Page Dashboard

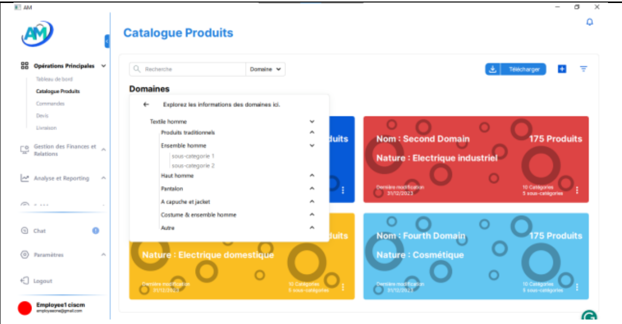

Catalogue Produits
Domaines
Explorez les informations des domaines ici.
Textile homme
Nom : Second Domain
Nature : Electrique industriel
Nature : Electrique domestique
Nom : Fourth Domain
Nature : Cosmétique

Figure 38 : Product Catalog Explorer

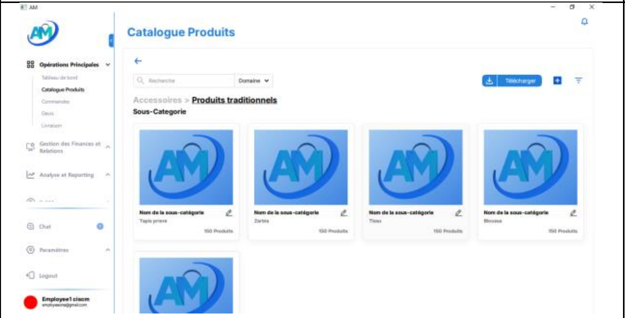


Figure 39 : Product Category

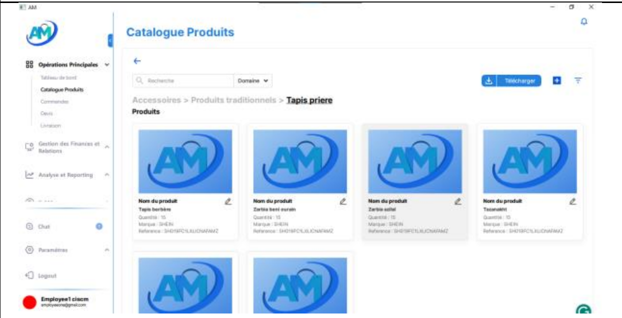


Figure 40 : Product Sub-Category

2.4 Conclusion

In these sprints, we explained the frontend architecture, introduced the translation feature, and showcased several important pages and their roles.

III. Sprint 4: Integration of API communication and CRUD operations

During this sprint, Server communication was integrated, enabling the application to interact with the backend for login and signup, complete with error detection messages. We introduced multi-threading to prevent the application from freezing, especially during server communication, and added a loading animation while waiting for server responses.

We also integrated the Employees page with the API for data management and CRUD operations. The employees' table is now displayed from the server, and we added search functionality by email/name. We implemented features to modify employee permissions, such as access to specific pages and setting access types (read-only or full edit). The ability to create new employees was added, and all actions are now based on user permissions to ensure only authorized users can perform specific tasks.

3.1 Backend Architecture

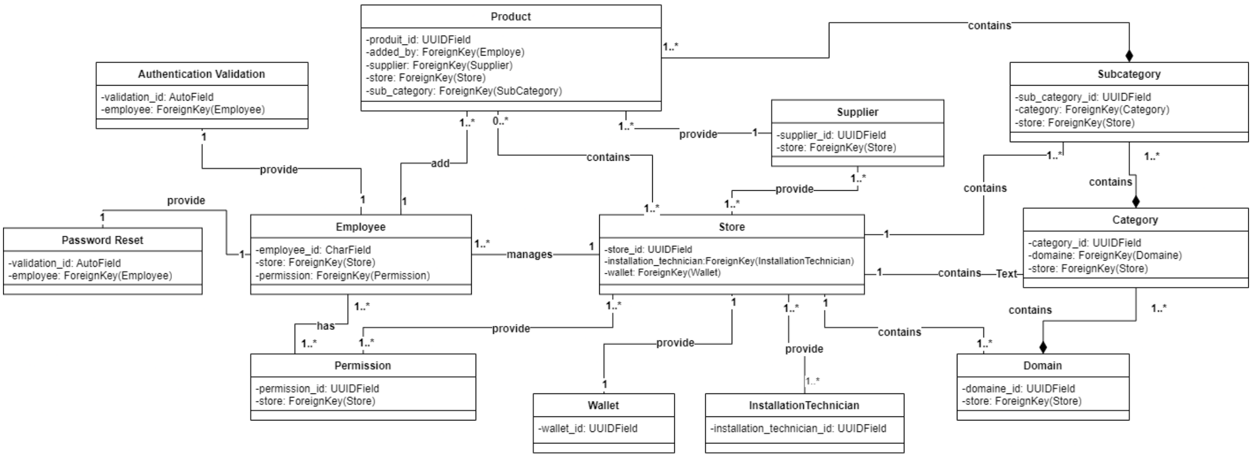

Product
-produit_id: UUIDField
-added_by: ForeignKey(Employe)
-supplier: ForeignKey(Supplier)
-store: ForeignKey(Store)
-sub_category: ForeignKey(SubCategory)
Authentication Validation
-validation_id: AutoField
-employee: ForeignKey(Employee)
Password Reset
-validation_id: AutoField
-employee: ForeignKey(Employee)
Employee
-employee_id: CharField
-store: ForeignKey(Store)
-permission: ForeignKey(Permission)
Permission
-permission_id: UUIDField
-store: ForeignKey(Store)
Store
-store_id: UUIDField
-installation_technician:ForeignKey(InstallationTechnician)
-wallet: ForeignKey(Wallet)
Supplier
-supplier_id: UUIDField
-store: ForeignKey(Store)
Subcategory
-sub_category_id: UUIDField
-category: ForeignKey(Category)
-store: ForeignKey(Store)
Category
-category_id: UUIDField
-domaine: ForeignKey(Domaine)
-store: ForeignKey(Store)
Domain
-domaine_id: UUIDField
-store: ForeignKey(Store)
Wallet
-wallet_id: UUIDField
InstallationTechnician
-installation_technician_id: UUIDField
provide
add
contains
manages
has
Text
1
1..*
0..*

Figure 41 : "Alma7al" Backend Architecture

The backend of our store management software is structured around a comprehensive set of classes that facilitate efficient management of various store operations. At the core, the Product class is used to maintain detailed information about the store's inventory, categorized through the Category and Subcategory and Domain classes to enable easy navigation and organization within the product catalog. The Supplier class manages interactions and transactions with suppliers, ensuring a streamlined supply chain.

To handle the human resources aspect, the Employee class includes features for managing employee profiles and roles. Complementing this, the Permission class allows for assigning specific tasks and permissions to employees, ensuring secure and efficient workflow management.

For security, the Authentication Validation and Password Reset classes provide robust mechanisms to protect sensitive data and support user authentication processes.

The Store class encompasses general store information and operations, while the Domain class helps in defining broader categories that may encapsulate multiple product categories. For financial management, the Wallet class tracks financial transactions and balances.

This interconnected system allows for robust management of store operations, from employee roles and permissions to product categorization and supplier relationships. The clear separation of concerns and detailed attribute tracking suggest a well-structured backend capable of supporting comprehensive store management functionalities.

3.2 Multi-threading for CRUD operations

QThread is a class in Qt that allows you to run code in parallel by creating and managing threads. This is useful for improving performance and keeping the user interface responsive. For example, I use QThread to handle server communication in a separate thread, so the user interface can still respond quickly to user actions like scrolling or clicking. QThread makes it easier to offload heavy tasks from the main thread, which is essential for applications that need to stay responsive while performing time-consuming operations.

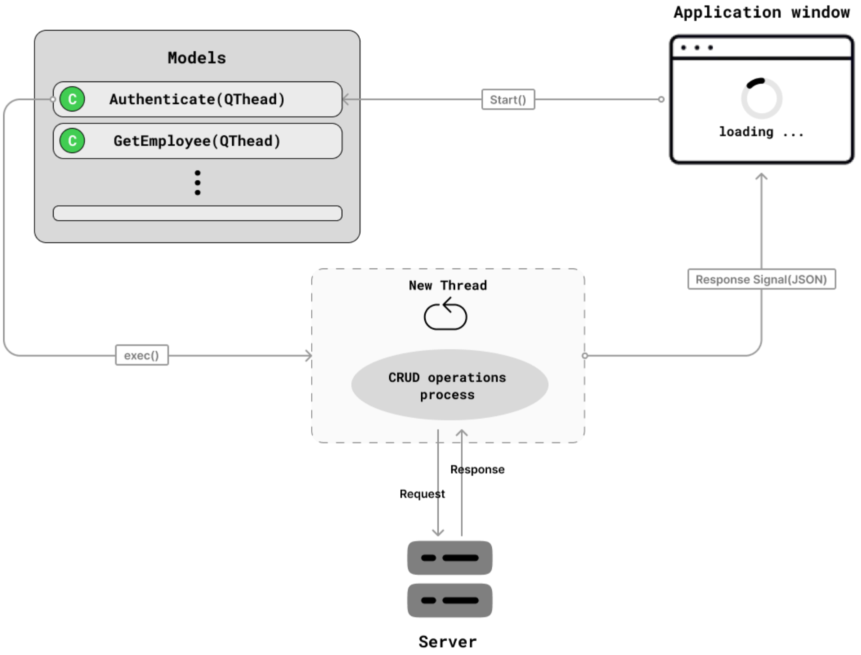
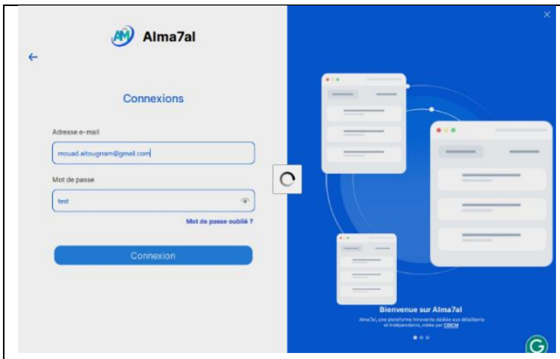


Figure 42 : the use of QThread for Server communications

3.3 Overview

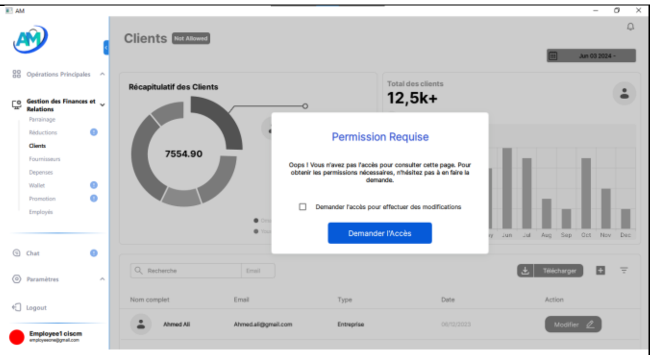
These screenshots highlight various features, including authentication, registration, permissions, and the ability to create, modify, and search for employees.



Adresse e-mail: mouad.aitougram@gmail.com

Mot de passe: test

Figure 43 : Authentication using QThread



Permission Requise

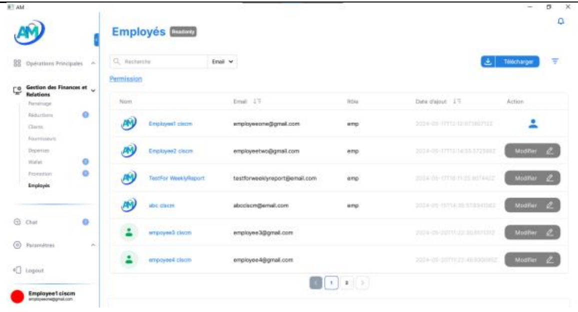
Oops ! Vous n'avez pas l'accès pour consulter cette page. Pour obtenir les permissions nécessaires, n'hésitez pas à en faire la demande.

☐ Demander l'accès pour effectuer des modifications

Demander l'Accès

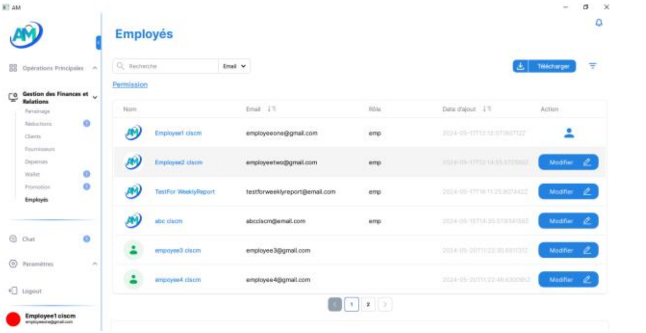
Nom complet	Email	Type	Date	Action
Ahmed Ali	Ahmed.ali@gmail.com	Entreprise	06/12/2023	Modifier

Figure 44 : a message popup up when user doesn't have permission



Nom	Email	Rôle	Date d'ajout	Action
Employee1 ciscm	employeeone@gmail.com	emp	[illegible]	
Employee2 ciscm	employeetwo@gmail.com	emp	[illegible]	Modifier
TestFor WeeklyReport	testforweeklyreport@email.com	emp	[illegible]	Modifier
abc ciscm	abcciscm@email.com	emp	[illegible]	Modifier
empoyee3 ciscm	employee3@gmail.com		[illegible]	Modifier
empoyee4 ciscm	employee4@gmail.com		[illegible]	Modifier

Figure 45 : the view when the user has read-only permission



Nom	Email	Rôle	Date d'ajout	Action
Employee1 ciscm	employeeone@gmail.com	emp	[illegible]	
Employee2 ciscm	employeetwo@gmail.com	emp	[illegible]	Modifier
TestFor WeeklyReport	testforweeklyreport@email.com	emp	[illegible]	Modifier
abc ciscm	abcciscm@email.com	emp	[illegible]	Modifier
empoyee3 ciscm	employee3@gmail.com		[illegible]	Modifier
empoyee4 ciscm	employee4@gmail.com		[illegible]	Modifier

Figure 46 : the view when the user has all permissions

Conclusion

This chapter demonstrated our systematic progress through each sprint, showcasing the successful completion of essential features such as the authentication UI, real-time chat functionality, various application pages, and API integration. Each sprint's achievements contributed significantly to the overall development and enhancement of the application.

General conclusion

In conclusion, this report presents the work completed for the End-of-Studies Project at the National School of Applied Sciences in Beni Mellal, in collaboration with CISCM Casablanca. The goal of this project was to develop an e-commerce platform (ERP CRM application for stores) using PySide6.

During the internship, I was able to deepen my theoretical and practical knowledge while benefiting from a professional environment that enhanced my skills and teamwork abilities.

However, various constraints and problems, such as time management issues and our need to become familiar with the technology used in development, and the large size of the project, have obstructed our progress. Additionally, multiple rejections and modifications in the backlog have caused delays. As a result, some tasks planned for the project, especially those related to the backend and implementation of CRUD operations, could not be fully completed before the presentation date. These include managing inventory and clients and suppliers, integrating a payment gateway, and other relevant tasks.

Regarding future perspectives, we aim to integrate additional systems and features into our platform, including advanced data analysis tools and more relevant economic ratios:Implement data analysis algorithms to provide deeper insights into sales trends and customer behavior, helping store managers make more informed decisions, Add more comprehensive economic ratios to better understand financial performance and optimize resource allocation, Integrate AI systems to predict inventory needs and automate restocking processes.

For the front end, we can continue to enhance the desktop application’s performance and speed, build more tools for data analysis, and focus on overall application improvement. Future features may include: Optimizing the application's performance for faster load times and smoother operation, Developing advanced data visualization tools to provide better insights into store operations, Adding offline capabilities

Bibliography

[1] Enterprise Software Growth – Morocco [Online]
[https://www.statista.com/outlook/tmo/software/enterprise-software/morocco#revenue]
[20/5/2024]

[2] Scrum Roles [Online]
[https://scrumguides.org/scrum-guide]
[20/5/2024]

[3] Channels [Online]
[https://testdriven.io/blog/django-channels/]
[20/5/2024]

[4] Redis [Online]
[https://www.ibm.com/topics/redis]
[20/5/2024]

[5] AWS [Online]
[https://www.techopedia.com/definition/26426/amazon-web-services-aws]
[20/5/2024]

[6] Python [Online]
[https://www.python.org/doc/essays/blurb]
[22/5/2024]

[7] Qt Benefits [Online]
[https://www.integrasources.com/blog/embedded-qt-hmi-development-why-where-and-how]
[26/5/2024]

[8] Event loop [Online]
[https://doc.qt.io/qt-6/threads-qobject.html]
[26/5/2024]

[9] Cross Platform Adaptability [Online]
[https://www.embedded.com/unifying-cross-platform-uis]
[26/5/2024]

[10] Signals and Slots [Online]
[https://doc.qt.io/qt-6/signalsandslots.html]
[26/5/2024]

[11] QWidget vs. QML [Online]
[https://www.youtube.com/watch?v=2RvhhEAZQxM]
[26/5/2024]

[12] PySide [Online]
[https://en.wikipedia.org/wiki/PySide]
[26/5/2024]

[13] QML [Online]
[https://en.wikipedia.org/wiki/QML]
[26/5/2024]

[14] JavaScript [Online]
[https://en.wikipedia.org/wiki/JavaScript]
[27/5/2024]

[15] Drawio [Online]
[https://www.computerhope.com/jargon/d/drawio.htm]
[27/5/2024]

[16] Figma [Online]
[https://en.wikipedia.org/wiki/Figma]
[27/5/2024]

[17] Git [Online]
[https://en.wikipedia.org/wiki/Git]
[27/5/2024]

[18] GitHub [Online]
[https://en.wikipedia.org/wiki/GitHub]
[27/5/2024]

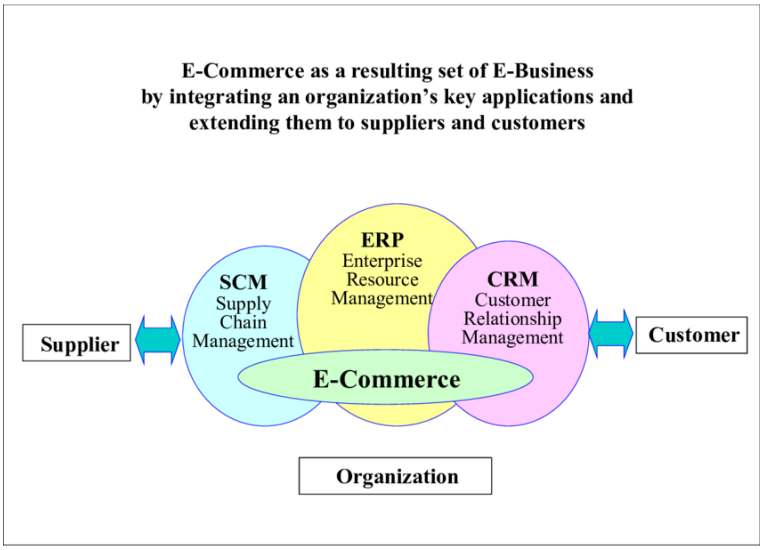
[19] Google Meet [Online]
[https://en.wikipedia.org/wiki/Google_Meet]
[27/5/2024]

[20] Visual Studio Code [Online]
[https://en.wikipedia.org/wiki/Visual_Studio_Code]
[27/5/2024]

[21] Django Channels architecture [Online]
[https://ably.com/topic/what-is-django-channels]
[27/5/2024]

[22] Gunson, John & Blasis, J.P.. (2001). The Place and Key Success factors of Enterprise Resource Planning (ERP) in the New Paradigms of Business Management. Ecole des Hautes Etudes Commerciales, Universite de Geneve-, Papers. [Online] [https://www.researchgate.net/publication/5063310_The_Place_and_Key_Success_factors_of_Enterprise_Resource_Planning_ERP_in_the_New_Paradigms_of_Business_Management]

[10/5/2024]


E-Commerce as a resulting set of E-Business
by integrating an organization's key applications and
extending them to suppliers and customers
ERP
Enterprise
Resource
Management
SCM
Supply
Chain
Management
CRM
Customer
Relationship
Management
Supplier
Customer
E-Commerce
Organization

Comments from the jury members

The jury members provided several insightful remarks on the end studies report. They appreciated the cohesive and comprehensive structure, noting that the report is divided into optimal chapters of similar length. One chapter is dedicated to the host company introduction, two chapters focus on the preparation phase, and the final chapter provides a concise and straightforward walkthrough of the sprints completed during the internship. The jury suggested:

- changing the French version title from "(... avec Pyside6 and QtQuick)" to "(... avec Pyside6 et QtQuick)."
- Additionally, they recommended including the abbreviation of ENSA,
- increasing the font size of the Table of Contents,
- and ensuring that the Gantt Chart fits the page properly.