People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Saad Dahlab University of Blida

Faculty of Sciences, Department of Computer Science

Master's Thesis

For obtaining the Maters's degree in Computer Science

Option: Software Engineering

Development of a Custom Stock Management System for Sonatrach Using Modern Software Frameworks

- **Represented by : MOHELLEBI ABDESSALEM**
- **Graduated in July , in front of the jury composed of :**
- **Dr. Nora Djemia: USDB - President**
- **Dr. Tobji Rachida: USDB - Examiner**
- **Co-superviser : ASSAKRI MALIKA (Information Technology Systems Engineer)**

Academic Year: 2024–2025

Acknowledgement

I would first like to express our sincere gratitude to Allah for granting me the health, perseverance, and strength necessary to bring this work to its completion.

My deepest appreciation is extended to my internship supervisor, Mrs. Assakri Malika, Information Technology Systems Engineer at Sonatrach Boumerdès. Her insightful guidance, unwavering support, and generous provision of her time and expertise were invaluable to me. i sincerely acknowledge her involvement, as well as the warm reception and assistance provided by the entire team at Sonatrach.

I also wish to convey my sincere thanks to my professors and the dedicated staff of the Computer Science Department at Saad Dahleb University, Soumaa, Blida. Their knowledge and commitment have served as a constant source of inspiration throughout my academic journey.

Furthermore, I warmly like to enthusiastically thank our firends and fellow students from the M2 Computer Science (2024/2025) for their collaborative spirit and the encouragement they provided along the way.

To all those who contributed to this project, whether directly or indirectly, I offer my sincere thanks.

Finally,I extend my profound gratitude to the esteemed members of the jury for the honor of reviewing and evaluating our work. Your time and consideration are deeply appreciated.

Thank you.

Dedication

I dedicate this study to my beloved family,
whose unwavering support and encouragement
have been my constant source of strength.

To my professors and mentors,
for their invaluable guidance throughout my academic journey.

And to all those who believed in me
even when I doubted myself.

Mohellebi Abdessalem

ملخص

تُتيح تنفيذ استراتيجية رقمنة الوثائق في الجزائر إدارة أكثر كفاءة وأمانًا للمعلومات. فهي تمهد الطريق نحو إدارة أكثر حداثة وشفافية وسهولة في الوصول، مما يُحسّن جودة وفعالية الخدمات العامة للمستخدمينفي هذا السياق، يهدف عملي إلى تصميم نظام جديد لإدارة الأصول يعتمد على قاعدة بيانات قوية وآمنة تجمع أصول مركز سوناطراك ، يتيح تطبيق سطح المكتب للمستخدمين المختلفين الوصول إلى النظام وأداء مهام إدارة الأصول المختلفة.

الكلمات المفتاحية: استراتيجية رقمنة الوثائق ، الخدمات العامة ، نظام إدارة الأصول ، مركز سوناطراك ، مهام إدارة الأصول.

Summary

Implementing a document digitization strategy in Algeria allows for more efficient and secure information management. This approach supports a modern administration and enhances public services by improving transparency, accessibility, and operational efficiency. It paves the way for a more modern, transparent, and accessible administration, ultimately improving the quality and effectiveness of public services for users. in this context, our work aims to design a new asset management system based on a robust and secure database that brings together all the assets of the Sonatrach center, the desktop application allows different users to access the system and carry out various asset management tasks.

Key words: Digitization strategy , Public services , Asset management system , Sonatrach , Asset management tasks.

Résumé

La mise en œuvre d'une stratégie de numérisation des documents en Algérie permet une gestion de l'information plus efficace et sécurisée. Cette approche favorise une administration moderne et améliore les services publics en renforçant la transparence, l'accessibilité et l'efficacité opérationnelle. Elle ouvre ainsi la voie à une administration plus moderne, transparente et accessible, améliorant en fin de compte la qualité et l'efficacité des services publics pour les usagers.dans ce contexte, mon travail vise à concevoir un nouveau système de gestion des biens basé sur une base de données robuste et sécurisée regroupant l'ensemble des actifs du centre Sonatrach. l'application deskto permet à différents utilisateurs d'accéder au système et d'effectuer diverses tâches liées à la gestion des biens.

Mots clés: Stratégie de numérisation , Services publics , Système gestion des biens , Sonatrach , Tâches liées à la gestion des biens.

Table of Contents

List of figures

List of Tables

List of Acronyms

AOT	Ahead Of Time
IA	Intelligence Artificielle
ISO	International Organization for Standardization
JIT	Just-In-Time
UML	Unified Modeling Language

GENERAL INTRODUCTION

Today, computer science plays an essential role in our lives, and its presence continues to grow thanks to its efficiency and usefulness [1]. It's a fast and effective way to run applications. In the past, people relied on paper to record information related to asset management , which often led to problems like wasted time searching for data or the deterioration of important documents. Over time, it became clear that more advanced tools were needed to improve both management . That's where computing comes in a field driven by the human desire to automate tasks that were once done manually, especially in areas like calculation and management.

Problem Statement

After gathering information at the Sonatrach Center in Boumerdes, several key challenges were identified, such as the risk of errors during manual verification of information, delays in the transmission of information related to certain assets, lack of data security, time loss, the large number of assets to manage, and concerns about overall system reliability.

The proposed desktop application should clearly provide an effective and relevant solution to address the identified challenges. It needs to streamline asset management, improve data accuracy, enhance security, and ultimately save time and resources for the Sonatrach Center in Boumerdes.

Objective

The objective of this work is to design and develop an asset management website for the center, with the aim of:

- Making it easier to search for and retrieve information by organizing data efficiently
- Creating an information system that includes a centralized database of all assets
- Storing data related to assets to track their allocation, condition, and quantity
- Improving the efficiency of data management processes
- Ensuring the security of asset-related data throughout the system
- Designing a user-friendly, intuitive interface for smooth and easy interaction

Study Outline

To achieve the defined objectives , this study is strcutured into the following chapters:

- **Chapter 1** presents the Literature Review, discussing asset management and its relation to digital systems in a general manner. It also introduces the modeling and detailed architecture of the system, identifying the main components and how they interact.
- **Chapter 2** is the Analysis of the Existing System, which provides an overview of Sonatrach and discusses the existing solution, explaining why it is no longer preferable.
- **Chapter 3** is the Architectural Design, covering UML and the diagrams used to model the application's architecture.
- **Chapter 4** presents the Implementation, including definitions of the programming languages and frameworks used, the reasons for their selection, and the developed user interfaces.

The study concludes with a general summary of the main pints discussed throughout it and suggestions for future research directions

Chapter 1 : Literature Review

1.1 Intorudction

Effective stock and asset management is a fundamental pillar of operational efficiency, especially for organizations that rely heavily on inventory, consumables, or physical assets. Poor inventory control can result in financial losses, operational disruptions, and dissatisfied customers. In contrast, well-structured stock systems allow businesses to maintain optimal inventory levels, forecast demand accurately, and reduce waste. This literature review explores the evolution of inventory systems, modern functionalities, enabling technologies, and the common challenges businesses face in deploying such systems.[2]

1.2 Transition to Digital and Automated Systems

The digital shift in inventory management began with the adoption of desktop software capable of managing item quantities, transaction histories, and basic supplier records. As technology matured, standalone systems gave way to enterprise level platforms. These platforms integrated procurement, warehousing, finance, and reporting modules into a unified interface.The use of relational databases significantly enhanced data accuracy and retrieval speed. Businesses could now access stock data instantly, reducing turnaround time for orders and improving service delivery. Custom-built inventory management software also allowed tailored workflows, automatic purchase orders, and audit trails.Moreover, modern systems enable centralization where multiple branches or warehouses feed data into a single platform. This ensures visibility across the supply chain, allowing managers to allocate resources effectively and prevent overstocking or understocking. [3]

1.3 Key Functionalities of Modern Inventory Systems

Contemporary inventory platforms are robust, offering automation and analytics that were unimaginable in traditional setups. They provide real-time stock visibility, allowing managers to monitor live inventory levels, identify fast-moving products, and track inbound and outbound stock. Automatic reordering capabilities minimize manual intervention by monitoring consumption patterns and replenishing items based on minimum and maximum thresholds. Batch and expiry tracking ensure compliance and facilitate recalls for perishable goods or pharmaceuticals. Barcode and RFID technologies enable efficient check-in and check-out processes, with RFID allowing passive, high-volume scanning. Supplier and purchase management features record and analyze lead times, prices, and performance to optimize procurement strategies. Role-based access restricts sensitive inventory data according to user permissions, ensuring security and accountability. Additionally, reporting dashboards generate insights on stock turnover, inventory value, and performance by location or item type. Together, these functionalities enhance operational efficiency while supporting strategic decision-making and long-term planning. [4]

1.4 Role of Cloud Computing and Integration Technologies

Cloud computing has revolutionized how inventory systems are deployed and accessed, as cloud-based platforms require no local servers and offer real-time access from any internet-enabled device—particularly beneficial for multi-site companies or those with remote and field operations. Integration capabilities are also essential, with REST APIs allowing connections to e-commerce platforms such as Shopify and WooCommerce, payment processors

like PayPal and Stripe, logistics providers including DHL and FedEx, and accounting systems such as QuickBooks and Xero. Additionally, cloud systems provide automatic backups, disaster recovery, and encryption protocols to safeguard business data, while vendors deliver regular updates to ensure security, compliance, and alignment with regulatory standards such as ISO 27001, SOC 2, and GDPR. [5]

1.5 Artificial Intelligence and Predictive Analytics

AI is rapidly transforming how businesses manage inventory. By analyzing historical data and external factors such as seasonal demand or economic indicators, AI can forecast demand spikes or slowdowns, suggest dynamic pricing to maximize revenue and minimize stock aging, identify irregularities including inventory shrinkage, fraud, or misplacement, and recommend redistribution of inventory across branches to balance supply. Machine learning models improve in accuracy over time as they process more data, allowing predictive analytics to shift organizations from reactive inventory management to proactive, scenario-based planning.

1.6 Common Challenges in Stock Management

Despite significant advancements in inventory technology, real-world implementation still faces a range of challenges. System complexity can be an issue, as some platforms are too intricate for small teams and require extensive configuration before delivering value. Employee adaptation is another common hurdle; without adequate training or a clear understanding of the system's benefits, data quality may suffer due to inconsistent usage. Cost and licensing considerations also pose difficulties, particularly for smaller firms, as subscription-based cloud solutions can accumulate significant expenses over time. Furthermore, internet dependency can limit the effectiveness of cloud-based platforms in regions with unreliable connectivity, and integration failures with third-party systems may result in data mismatches, delays, or downtime. Overcoming these challenges requires careful planning, strong stakeholder buy-in, targeted training programs, and phased implementation strategies to ensure smooth adoption and sustained operational efficiency. [6]

1.7 Conclusion

Inventory and stock management has matured into a strategic function, backed by powerful software systems and emerging technologies like AI and cloud computing. These systems support not just daily operations but also long-term business agility, customer satisfaction, and financial health.

Chapter 2 : Existing system analysis

2.1 Introduction

In this chapter, we examine the existing asset management application used at the Sonatrach Center in Boumerdes[7]. This step is essential to understand the current context, the existing features, and any potential shortcomings of the application.

2.2 General Overview of Sonatrach

According to Sonatrach Group, *Company Overview and Performance Report* [8] , Sonatrach is Algeria's state-owned oil and gas company and ranks as the 12^{th} largest oil consortium in the world. With 154 subsidiaries, Sonatrach operates across the entire hydrocarbon value chain—including upstream exploration and production, midstream transport and storage, and downstream refining and distribution.In 2002, the company reported gross sales of 1,530 billion Algerian dinars and a net income of 175 billion dinars. That same year, Sonatrach accounted for approximately 30% of Algeria's Gross National Product (GNP), employing around 120,000 people. Its annual production reached 206 million tonnes of oil equivalent (ToE), of which 24 million ToE—roughly 11.7%—was allocated for domestic consumption. By 2021, Sonatrach's performance had significantly improved. Export revenues surged by 75%, climbing from US$20 billion in 2020 to US$35 billion, driven largely by the global economic recovery and increased energy demand. Production levels reached 185 million ToE, with over 95 million ToE destined for export. The workforce comprised over 53,000 direct employees, supported by more than 150,000 workers across its subsidiaries and contractors. Sonatrach operates Algeria's most productive oil fields. Hassi Messaoud alone contributed approximately 440,000 barrels of crude oil per day in 2006. Meanwhile, the Hassi R'Mel field, one of Africa's largest gas fields, maintained a steady output of around 180,000 barrels per day. Other significant fields under Sonatrach's management include Tin Fouye Tabankort, Zarzaitine, Haoud Berkaoui/Ben Kahla, and Ait Kheir. The company maintains and operates over 3,900 kilometers of crude oil pipelines. These critical infrastructure assets include pipelines transporting crude oil from Hassi Messaoud to the coastal export terminals. Sonatrach also manages pipelines for condensate and liquefied petroleum gas (LPG), notably linking Hassi R'Mel to the industrial complex at Arzew. To meet growing demand, it is currently doubling the capacity of the Hassi Messaoud–Arzew line through the construction of a second pipeline. Beyond its domestic operations, Sonatrach holds strategic concessions in several international markets, including Libya, Mauritania, Peru, Yemen, and Venezuela. It is also expanding its portfolio into petrochemicals and seawater desalination to support energy diversification and meet future sustainability goals. Estimates suggest that Sonatrach manages hydrocarbon assets worth over 100 billion US $, making it not only a cornerstone of Algeria's economy but also a major player in the global energy sector.

2.3 General Organizational Chart of Sonatrach

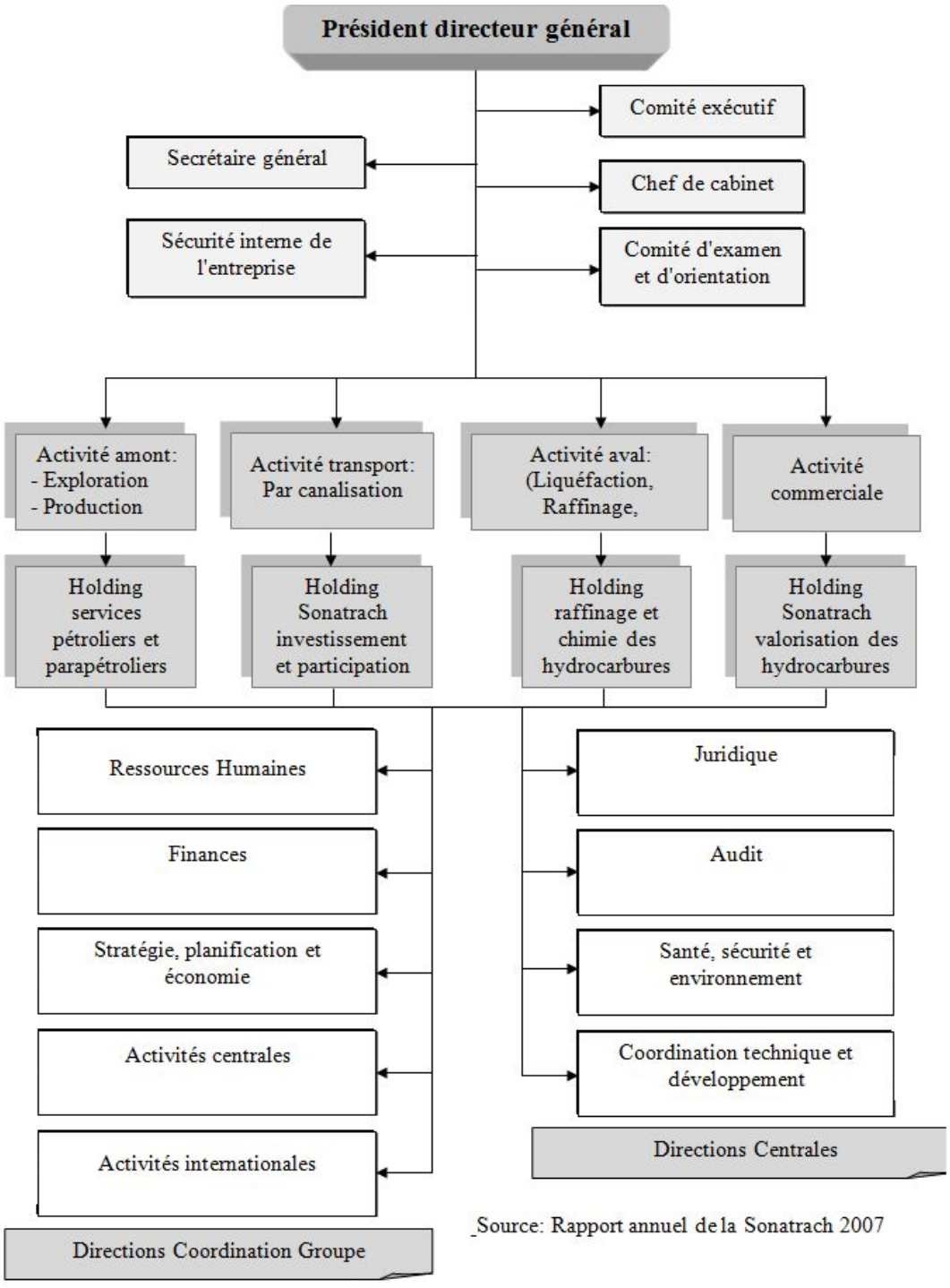

Figure 2.1: Organizational charts of Sonatrach [Sonatrach Annual Report, 2007.]

The diagram above illustrates the organizational structure of the national company **Sonatrach**, based on the 2007 annual report. It provides a comprehensive overview of the company's hierarchical and functional layout, highlighting the main divisions, operational activities, and support functions that enable its efficient operation.
At the top of the hierarchy is the **Chief Executive Officer (CEO)**, who is responsible for the overall strategic direction and management of the company. Directly reporting to the CEO are several key strategic entities, including the **General Secretary**, who oversees administrative coordination; the **Internal Security of the Company**, which ensures the protection of personnel, assets, and sensitive information; and consultative bodies such as the **Executive Committee, Chief of Staff, and the Review and Orientation Committee**, which play essential roles in decision-making and strategic planning.
Below the central governance, the diagram highlights the four main operational activities of Sonatrach. The **Upstream Activity** (Exploration and Production) is managed by the *Holding for Oilfield and Related Services*, responsible for hydrocarbon exploration and extraction. The **Transport Activity** (Pipeline) falls under the *Sonatrach Investment and Participation Holding*, which handles the movement of hydrocarbons through pipelines. The **Downstream Activity** (Liquefaction and Refining) is managed by the *Refining and Hydrocarbon Chemistry Holding*, focusing on processing and transforming oil and gas. Finally, the **Commercial Activity** is overseen by the *Sonatrach Hydrocarbon Valorization Holding*, responsible for sales, marketing, and market optimization.
In addition to these operational units, the company relies on a set of **support and coordination departments** that ensure the smooth functioning of all activities. The **Group Coordination Departments** provide support in human resources, finance, strategy, planning, economics, central activities, and international operations. Complementing these, the **Central Departments** are responsible for legal affairs, internal audit, health, safety, and environmental management, as well as technical coordination and development. These cross-functional structures maintain coherence across the group, optimize internal processes, and strengthen the company's capacity for growth, operational efficiency, and performance.

2.4 History of the Sonatrach Division Exploration Center – Boumerdès Case Study

The Sonatrach Division Exploration Center, located in Boumerdès, is a cornerstone of the upstream branch of **Sonatrach**, Algeria's national oil and gas company. Established to advance hydrocarbon exploration across the country, the center focuses on research, analysis, and evaluation of potential oil and gas fields. Its activities encompass conducting detailed geological and geophysical studies, processing and interpreting seismic and well data, and providing technical support to field teams involved in drilling and subsurface analysis. By developing and applying advanced technologies for data modeling, reservoir characterization, and predictive analytics, the center ensures that exploration operations are informed by the latest scientific and technological insights. Additionally, it maintains a comprehensive archive of exploration data, ensuring long-term accessibility for strategic planning and decision-making.

Over the years, the Boumerdès center has played a pivotal role not only in the surrounding region but also across multiple strategic areas of Algeria. Its collaborative framework extends beyond internal Sonatrach departments to include partnerships with university research institutions, national and international exploration companies, and specialized oilfield service providers. These collaborations foster knowledge exchange,

innovation, and adoption of best practices in exploration methodologies.

Through its work, the center contributes directly to Sonatrach's broader mission of ensuring national energy security. By identifying and developing new hydrocarbon resources, supporting technological advancement, and training specialized personnel, the Boumerdès Exploration Center strengthens Algeria's capacity to sustainably manage and expand its energy portfolio. Over time, it has become a benchmark for excellence in exploration, combining scientific rigor, operational efficiency, and strategic foresight to support the nation's energy ambitions.[9]

2.5 .Organizational chart of Sonatrach Division Exploration Center – Boumerdes

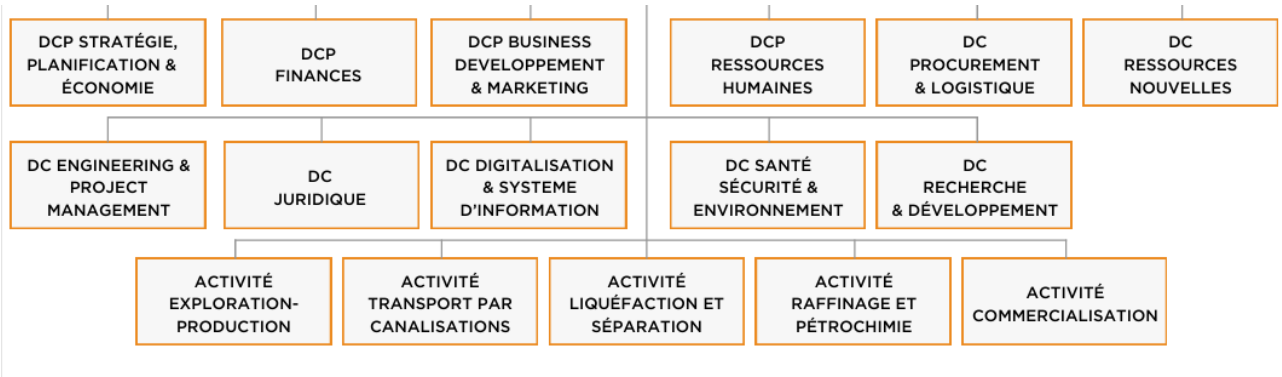

Figure 2.2 : Organizational charts of Sonatrach Division Exploration Center – Boumerdes [Annuel Report of Sonatrach 2012]

The diagram above presents a modernized and highly structured organizational framework of **Sonatrach**, Algeria's national oil and gas company. This configuration reflects the company's evolution toward greater efficiency, operational excellence, technological advancement, and strategic governance. By integrating both strategic leadership and operational units, the structure ensures coherent decision-making, rapid response to market and technical challenges, and seamless coordination across the entire hydrocarbon value chain.
At the apex of the organization are the **Direction Centrale (DC)** and **Direction Centrale des Pôles (DCP)**, representing the strategic and support divisions responsible for defining the company's long-term vision, monitoring performance, and providing cross-functional oversight. These divisions are tasked with translating national energy policies and corporate objectives into actionable programs while maintaining alignment with international standards, regulatory frameworks, and emerging energy trends.
Within these top tiers, the **DCP Stratégie, Planification & Économie** plays a pivotal role in shaping Sonatrach's corporate strategy. It conducts market analyses, evaluates national and global energy trends, and develops economic models that guide investment decisions, resource allocation, and growth initiatives. Its planning function ensures that all operational units are working toward clear, measurable goals that align with the company's broader objectives of energy security, sustainability, and profitability.
The **DCP Finances** department oversees financial governance, budgeting, and investment management. It ensures that all projects, from upstream exploration to downstream refining, are economically viable, cost-efficient, and aligned with the company's strategic priorities. This department also monitors cash flow, evaluates capital expenditure proposals, and

provides risk assessment for high-value investments, ensuring that Sonatrach maintains financial stability in volatile energy markets.
DCP Business Development & Marketing is responsible for identifying new market opportunities, developing strategic partnerships, and implementing marketing strategies that enhance Sonatrach's global footprint. It manages client relationships, negotiates commercial contracts, and evaluates emerging energy markets to ensure the company can expand its reach and adapt to the evolving energy landscape.
The human element is handled by **DCP Ressources Humaines**, which ensures workforce planning, recruitment, and continuous personnel development. This department designs training programs, manages talent pipelines, and fosters a corporate culture that supports innovation, safety, and operational excellence. By aligning human resources with strategic and operational needs, it ensures that Sonatrach possesses the skilled workforce necessary to implement complex technical projects.
On the operational support side, **DC Procurement & Logistique** coordinates the acquisition of materials, equipment, and services essential for exploration, production, transport, and refining activities. This department optimizes supply chains, reduces costs, and ensures timely delivery of critical resources, contributing to uninterrupted operations across all sites. **DC Ressources Nouvelles** focuses on exploring alternative resources, diversification initiatives, and emerging energy technologies, enabling Sonatrach to remain adaptive and future-oriented in a global energy transition context.
DC Engineering & Project Management ensures that complex technical projects—from drilling campaigns to refinery expansions—are executed efficiently, on schedule, and within budget. It integrates engineering expertise with project oversight, risk management, and quality assurance to maximize productivity while maintaining compliance with safety and environmental standards. **DC Juridique** provides legal support and regulatory compliance guidance, helping the company navigate international contracts, local laws, and industry regulations, thus minimizing legal and operational risks.
The **DC Digitalisation & Système d'Information** department drives the company's digital transformation initiatives, overseeing IT infrastructure, automation, data analytics, and cybersecurity. Its role is critical for modern exploration techniques, real-time monitoring of operational performance, and decision support systems that leverage AI and predictive analytics. Similarly, **DC Santé, Sécurité & Environnement** ensures occupational health, safety, and environmental protection, enforcing compliance with global HSE standards and promoting a culture of safety at all operational sites. Finally, **DC Recherche & Développement** fosters innovation by developing new technologies, improving exploration and refining processes, and supporting scientific research that advances Sonatrach's technical capabilities.
The bottom tier of the diagram highlights Sonatrach's **operational activities**, which form the backbone of the company's day-to-day functions. The **Exploration-Production Activity** identifies, evaluates, and extracts hydrocarbon reserves, employing cutting-edge geophysical surveys, seismic data interpretation, and advanced drilling technologies. The **Transport by Pipelines** unit manages an extensive network of pipelines, ensuring safe, reliable, and cost-efficient transportation of oil and gas from extraction sites to refineries, storage facilities, and export terminals.
The **Liquefaction and Separation Activity** handles gas processing and liquefaction, converting raw hydrocarbons into marketable LNG and other products while adhering to strict quality and safety standards. The **Refining and Petrochemicals Activity** converts crude oil and gas into refined fuels, chemicals, and petrochemical products, employing sophisticated technologies to maximize yield, minimize environmental impact, and maintain product quality. Finally, the **Commercialization Activity** oversees domestic and

international sales, marketing, and client relationship management, ensuring that Sonatrach maintains a competitive position in global energy markets while optimizing revenue streams. By integrating these strategic, support, and operational units into a cohesive structure, Sonatrach achieves several critical objectives: enhanced coordination between leadership and field operations, accelerated decision-making processes, improved risk management, and increased adaptability to global market fluctuations. The structure also facilitates innovation, digital adoption, and operational excellence, positioning the company as a resilient and forward-looking energy leader capable of meeting Algeria's current and future energy demands. This organizational model exemplifies how large-scale national companies can combine governance, technical expertise, and operational efficiency to sustain growth, maintain competitiveness, and contribute to national energy security.

2.6 Functional Analysis

This section provides a general overview of the functional environment in which the system operates. It highlights the main elements necessary for the system to function properly, including the human, material, and digital resources involved. The analysis will be broken down into several subsections to better understand how each component contributes to the overall operation and performance of the system.

2.6.1 Human Resources:

The IT department consists of a diverse and specialized team of professionals, each playing a critical role in supporting the company's technological and operational needs:

- **Eleven Database Engineers:** responsible for designing, maintaining, and optimizing relational and non-relational databases; ensuring data integrity, security, and high availability; implementing backup and recovery strategies; performing regular performance tuning; and supporting the development team with efficient query design and data modeling.
- **Ten Network Engineers:** specialized in network design, configuration, and troubleshooting; managing local area networks (LANs), wide area networks (WANs), and virtual private networks (VPNs); ensuring network security, firewall configuration, and intrusion detection; monitoring network performance; and providing technical support to prevent downtime and maintain seamless connectivity across multiple sites.
- **Three IT Infrastructure and System Administration Engineer:** responsible for managing servers, storage systems, virtualization platforms, and cloud resources; performing system updates, patch management, and configuration; monitoring overall system health; implementing disaster recovery solutions; and ensuring high availability and reliability of IT infrastructure.
- **Five Software Engineer:** tasked with developing, maintaining, and improving internal applications and software tools; automating repetitive processes; integrating applications with existing systems; writing clean, maintainable code; and collaborating with other engineers to enhance overall productivity and system efficiency.
- **Multiple Data Entry Agents:** handling accurate input, validation, and updating of critical operational and administrative data; maintaining data quality standards; assisting in reporting and data preparation for analytics; and supporting various departments by ensuring timely and error-free data processing.

- **One Accountant:** responsible for financial record-keeping and reporting for the IT department; managing budgets, tracking expenses, and preparing financial statements; coordinating with other departments for procurement and invoicing; ensuring compliance with corporate financial policies; and supporting strategic planning through financial analysis.
- **IT Support Technicians (Multiple):** providing first-line technical support to staff; troubleshooting hardware and software issues; installing and configuring new equipment; assisting with system updates; and ensuring users can operate efficiently without IT-related disruptions.
- **Cybersecurity Specialist:** responsible for protecting the department and company-wide IT assets from cyber threats; implementing security protocols, monitoring networks for breaches, and conducting regular vulnerability assessments; training staff on security best practices; and ensuring compliance with national and international cybersecurity standards.
- **Business Analyst:** acts as a liaison between technical teams and operational units; gathers requirements, documents workflows, analyzes business processes, and proposes IT solutions that align with company goals; helps prioritize projects and ensures technology supports strategic objectives.
- **Project Coordinator:** manages timelines, resources, and deliverables for IT projects; ensures tasks are completed according to schedule; communicates progress to management; identifies risks; and coordinates between multiple teams to achieve smooth project execution.

2.6.2 Hardware Resources:

All computers are connected via a fully operational Local Area Network (LAN), ensuring smooth communication, file sharing, and centralized access to network resources across the organization. This network infrastructure supports essential services such as internet connectivity, printer sharing, access to internal servers, and remote administration. The LAN is secured using firewalls and access control policies to prevent unauthorized access and ensure data integrity.

The specifications of the PCs are generally decent and vary depending on their intended function within the organization. Basic administrative and office tasks are typically carried out on standard machines equipped with mid-range processors, sufficient RAM, and solid-state drives for better responsiveness. More advanced configurations, including higher-performance CPUs and dedicated graphics cards, are reserved for technical users such as engineers or designers, who require greater computational power for simulations, design work, or data processing.

The computer fleet operates on a diverse range of operating systems, including Windows, Linux (primarily Ubuntu and Debian distributions), and macOS. The choice of operating system is dictated by the user profile and the technical demands of their responsibilities. For instance, Linux is favored by developers and system administrators for its flexibility and control, while Windows remains the standard for general office use and proprietary enterprise applications. macOS may be used in departments focused on media production or design.

Equipment	Approximate Quantity
Desktop PCs	500
PC Screens	700
Laptops	120
Printers	50
Scanners	30
Interphones	40
Network Switches	25
Network Routers	15
Wi-Fi Access Points	40
Servers (Database/Web/File)	10
Projectors (Meeting Rooms)	12
UPS (Uninterruptible Power Supply)	20
External Backup Drives/Units	15
Firewalls (Hardware)	5
NAS (Network Attached Storage)	6
Surveillance Cameras (IP-based)	60
VoIP Phones	35
Time Attendance Machines (Biometric)	10
Smart TVs (Meeting/Training Rooms)	8
Air Conditioners (Server Rooms)	4
Cabling and Patch Panels	50
Workgroup Printers (Shared)	10
Video Conferencing Systems	5
Network Racks and Cabinets	12
Barcode Scanners	25
External Monitors for Laptops	60
Wireless Keyboards and Mice Sets	200
Docking Stations	50
Speakers (Office/Meeting Rooms)	40
Headsets (VoIP/Online Meetings)	80
Label Printers	15
Backup Power Generators	3
Smart Boards (Interactive Displays)	6
Digital Signage Screens	10
Tablets (Field Work/Inventory Use)	20
Network Testers and Tools	10
Security Alarms and Sensors	25

Table 2.3: Summary of a Sample of Available Hardware Resources

All servers within the infrastructure run on Linux due to its stability, performance, and security features. These servers handle various critical services such as web hosting, database management[18], internal communications, and authentication. System updates and maintenance are regularly performed by the infrastructure engineer to minimize downtime and ensure optimal performance. Moreover, routine monitoring and backups are implemented to further enhance the reliability and availability of the system.

2.6.3 Software Resources:

Sonatrach currently uses a legacy cross-platform desktop system for asset management, developed in C++. While C++ is a powerful and efficient language, the major issue lies in the fact that Sonatrach does not possess the source code or licensing rights for the software. Only the end product was acquired. This effectively turns the company into a passive user, completely dependent on the original vendor for any future modifications, bug fixes, or enhancements. In a time where agility and adaptability are critical for digital systems, relying on a "black box" solution is a severe limitation. The inability to evolve the software internally means the company cannot keep pace with operational changes, compliance updates, or user feedback — all of which are essential in modern IT environments.

In addition to that, the application uses Oracle Database 11g, which was released in 2007 and has not been updated or migrated since. This is highly problematic from a security and maintainability standpoint. Running outdated software increases exposure to vulnerabilities that have long since been discovered and exploited. It also means the system does not benefit from newer performance optimizations, support for modern data types, or integration capabilities with newer tools and frameworks. Worse yet, the server hosting the database is known to crash or become unresponsive multiple times a day, often requiring manual restarts. This is not only inefficient but also dangerous, as it introduces frequent downtimes, risks data loss, and impacts the reliability of the system.[20]

From a best-practices perspective, this entire setup violates several foundational principles of modern system architecture: lack of maintainability, poor fault tolerance, and absence of software ownership. In any enterprise-grade environment, especially for a company of Sonatrach's scale and strategic importance, these issues are unacceptable. The combination of technical obsolescence and vendor lock-in makes the current system unsustainable in the long term and calls for a robust, scalable, and future-proof replacement.

2.7 Requirements Analysis

Here is an analysis of the essential requirements for developing a desktop asset management application. The goal is to identify the functionalities, non-functional requirements, and technical aspects necessary for their design and implementation:

2.7.1 Functional Requirements

- Once authenticated, the administrator can access his features within the application.
- The desktop asset management application facilitates the search and retrieval of information by efficiently organizing data.
- It is designed as an information system with a centralized database that consolidates all assets.
- Data related to these assets is stored in the database.
- The system aims to enhance security.
- It contributes to cost reduction.

- The application allows for the entry of new assets, the modification of existing ones, and the transfer of assets between different locations.
- The administrator has the authority to create, modify, and delete accounts for data entry agents and accountants, but cannot delete or modify their own account.
- The accountant is responsible for verifying invoice information.
- The asset management application is used to manage and track assets within an inventory.
- Barcodes can be used to track the condition and location of assets .

2.7.2 Non-Functional Requirements

- The security of users personal information and asset-related data must be ensured throughout the entire process.
- The desktop application should be accessible across multiple platforms, including Linux, macOS, and Windows.
- The application must have fast loading times to ensure responsiveness.
- The application must be easy to use, with a clear and intuitive user experience.
- The system must comply with current desktop development standards and best practices.
- The application must be maintainable over the long term, allowing for updates and scalability.
- The application must deliver high performance, even when handling a large volume of data.
- A visually appealing and user-friendly interface should be designed.
- File integrity and security must be preserved in the storage of asset data.
- The system should be fault-tolerant, continuing to operate reliably in the event of minor failures.
- The application should support modularity, allowing different components to be developed, tested, and deployed independently.

2.8 Summary of Findings

Through the study of the existing system, several key issues have been identified that highlight the need for a more robust and efficient asset management solution:

- There is a high risk of error due to the manual verification of information.
- Tracking assets is difficult because of the lack of a reliable and centralized system.
- The transmission of information related to certain assets is slow and inefficient.
- Data security is insufficient, exposing the system to potential risks.
- A significant amount of time is lost in daily operations due to outdated processes.

2.8.1 Objectives

The desktop application aims to improve the search and retrieval of information by organizing data efficiently. The main objectives are:

- Design an information system with a centralized database that includes all assets.
- Store data related to these assets in an organized database.
- Enhance data security.
- Reduce operational costs.
- Once authenticated, the administrator can access and manage their functionalities in the application.
- The asset management application allows adding new assets, modifying existing ones, and transferring assets between locations.
- Barcodes can be used to track the condition and location of assets.
- Ensure fast loading times of the asset management desktop application.
- Ensure ease of use through a user-friendly and intuitive interface.
- Improve the visibility of the system and encourage user engagement.
- Create an interface that scans asset barcodes and extracts the corresponding data, which should then be stored in an organized manner to enable efficient information management.
- Improve the overall efficiency of data management.

2.9 Conclusion

In conclusion, this chapter offered an overview of the existing system currently in use at the Sonatrach Exploration Center in Boumerdès. It provided a general understanding of how the

system operates within its environment and how it supports the center's daily activities. The analysis also helped identify some of the main strengths and weaknesses of the current application, showing that while it meets several operational needs, there is still room for improvement to make it more efficient and better adapted to users' expectations. This reflection serves as a starting point for the following chapter, which will focus on the architecture design of the proposed solution and how it can respond to the challenges identified in this study.

Chapter 3 : Architecture Design

3.1 Introduction

In this chapter, we present the design of our desktop application for asset management using the UML (Unified Modeling Language) modeling language. UML provides a set of standardized notations and techniques to graphically represent various aspects of a software system. We used three types of UML diagrams in our design process to model the system's functional aspect (use case diagram), static structure (class diagram), and dynamic behavior (sequence diagram).

3.2 Application Modeling

I used the UML language in my design process. UML is a standardized modeling language used in software engineering to represent, specify, visualize, and document software systems and development processes. It provides a set of diagrams that allow for the representation of different perspectives and phases of a software system, from design to implementation.[10]

The primary goal of UML is to facilitate communication and understanding among development team members, analysts, designers, and project stakeholders. It enables the capture of requirements, modeling of system structure and behavior, identification of relationships between elements, and documentation of design decisions.

3.3 Diagrams

3.3.1 Use case diagram

Definition of the Use Case Diagram : In UML language, use case diagrams model the behavior of a system and illustrate either its entire set of requirements or key parts of it. The use case diagram consists of actors, use cases, and the relationships between them.[11]

3.3.2 Components of the Use Case Diagram

Actors: An actor represents an external entity that interacts with the system being modeled. This entity can be a human user, an external system, an organization, or even a hardware device.

In the context of use case diagrams, actors play a crucial role in defining system boundaries and expectations. Each actor initiates interactions with the system to achieve a specific goal. For instance, in a goods management system, a "Warehouse Manager" may be an actor responsible for updating stock levels, while an "Inventory Auditor" may only view data.

There are two main types of actors:

- **Primary actors:** Those who initiate the use case and are the main beneficiaries of the system's functionality.
- **Secondary actors:** Those who support or are used by the system during the execution of a use case, such as databases or authentication servers.

Identifying actors correctly is a foundational step in understanding how users or external systems expect to interact with the software. It also helps define access control, roles, and permissions during both the design and implementation phases.

Use Cases: A use case describes a specific function that the system performs to fulfill a user's goal. Each use case should produce a visible and useful outcome for the actor involved, based on that actor's functional needs.

Use cases are central to understanding how the system behaves in various situations. They help define the required behavior of the application and ensure that developers and stakeholders share a common understanding of what the system is supposed to do.

Relationships Between Actors and Use Cases: These relationships are visually represented by arrows that indicate the direction of communication. For example, an arrow from an actor to a use case signifies that the actor initiates that specific use case.

The diagram may also include *extends* or *includes* relationships, which represent optional behavior or reused functionality, respectively. These mechanisms help in organizing complex scenarios, such as handling exceptions or common sub-processes across multiple use cases.

3.3.3 Representation of the Use Case Diagram

Administrator: The administrator is responsible for creating, deleting, and modifying user accounts for both data entry agents and accountants—but is restricted from altering or deleting their own account. This role may also include managing user credentials, such as updating login information.

Generally, the administrator holds a higher level of access than regular users and is authorized to perform tasks that fall outside the permissions of standard accounts. In addition to managing user accounts, the administrator is also responsible for controlling access to specific areas or features of the system. This includes granting or restricting permissions to ensure that each user can only interact with the parts of the application relevant to their role.

Data Entry Agent: A data entry agent in an asset management system is responsible for accurately entering information related to assets, as well as modifying, deleting existing records, and performing transfers between locations.

The role of the data entry agent is critical, as input errors can lead to asset management issues that may have negative consequences. It is essential that the data entry agent is properly trained in using the asset management system and in the correct procedures for data entry to ensure the accuracy and reliability of the recorded data.

Accountant: An accountant in asset management is a professional responsible for monitoring and verifying invoice information against the system, as well as recording financial transactions related to asset management.

Their primary role is to maintain financial records and ensure that all financial information is accurate and up to date.

3.3.4 Use cases

3.3.4.1 Identification and Representation of Use Cases

The following diagram provides a general representation of how the asset management system functions, with different actors each playing their own specific role within the system.

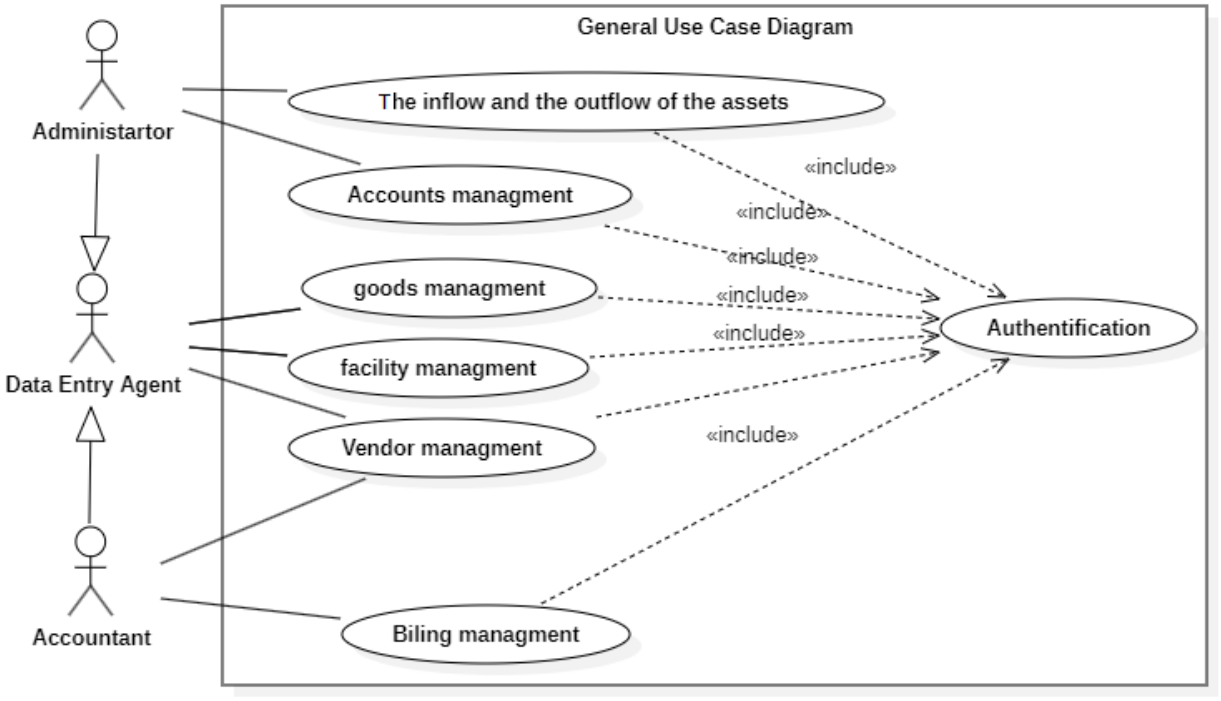

Figure 3.1 : General Use Case Diagram

3.3.4.2 Use Case Diagram for Accounts managment use case

User Account Management Features

- **Create Accounts**

  This functionality allows the administrator to create accounts for data entry agents and accountants by entering their login credentials. This task is performed through the web application interface.

- **Modify Accounts**

  This refers to the ability to change the credentials associated with a user account. Adjusting account settings is often necessary for the administrator to maintain security and ensure compliance with organizational data policies. Ultimately, account modification is an important method for administrators to effectively manage users and data within the organization.

- **Enter New Credentials**

  The administrator inputs new login information in order to update user accounts.

- **Delete Accounts**

  The administrator has the right to delete the accounts of data entry agents and accountants when they are no longer needed or if there are security concerns. However, the administrator cannot delete or modify their own account.

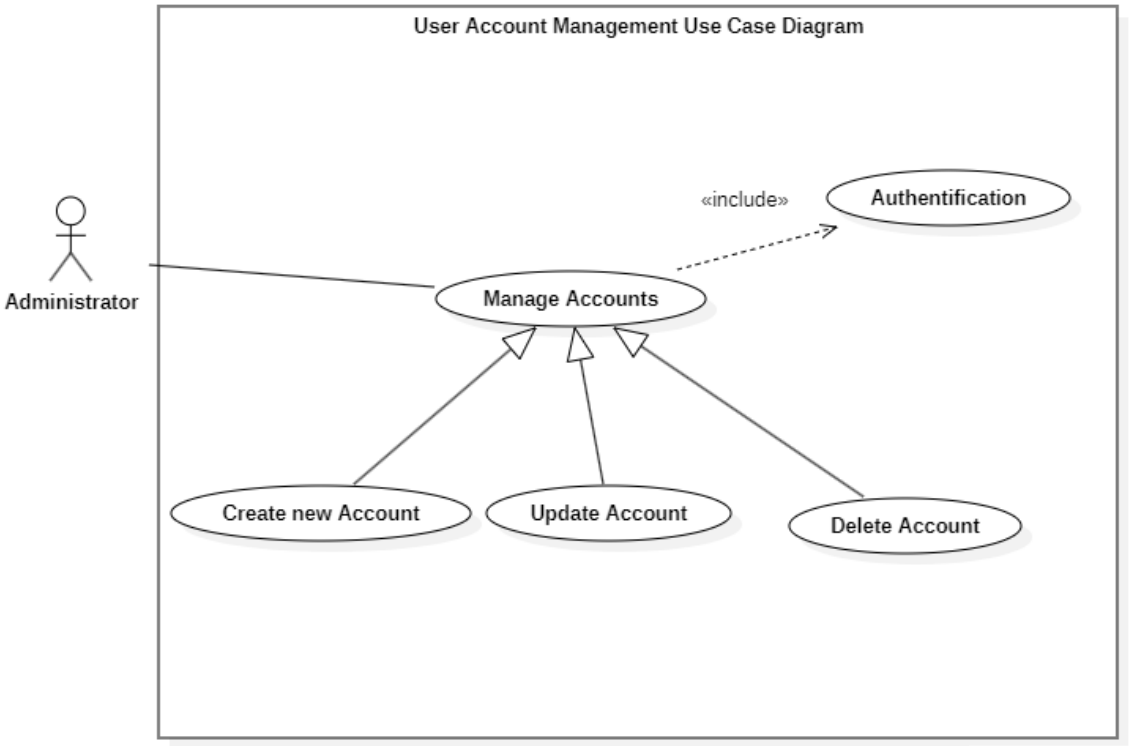

Figure 3.2 User Account Management Use Case Diagram

3.3.4.3 Use Case Diagram for Authentification Process

All users of the system—including administrators, accountants, and data entry agents—must authenticate themselves before gaining access to their respective accounts. To do so, each user is required to enter a valid username and password combination. If either the username or password is incorrect, the system prompts the user to re-enter their credentials. Only upon successful authentication is the user routed to their designated interface, where they can perform their assigned tasks securely and efficiently.

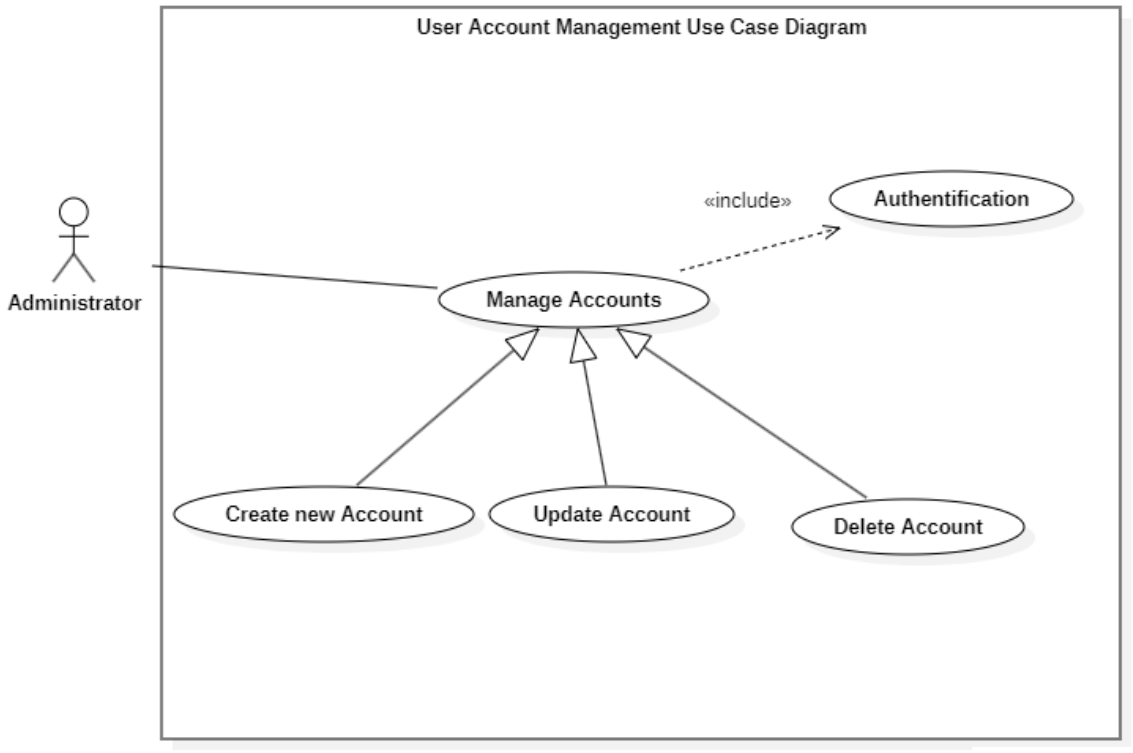

figure 3.4 Authentification Process Use Case Diagram

3.3.4.4 Use Case Diagram for Goods managment use case

Barcode Scanning: Thanks to the barcode scanners already available within the company, users can scan asset labels directly. The data associated with the scanned assets is automatically captured and transmitted to the MongoDB database[17]. The desktop asset management application then retrieves this information and presents it in a user-friendly interface to the data entry agents. This process allows the agents to quickly view the scanned asset details and take any necessary actions, such as updating statuses or initiating transfers. Barcode scanning significantly enhances inventory accuracy and operational efficiency by reducing manual data entry and minimizing errors.

Adding New Assets: This functionality enables users to register new assets into the asset management desktop application database. It ensures that the lifecycle of each asset—from acquisition through to allocation and usage—is well documented. By recording these assets, the organization can track their quantity, location, condition, and movements throughout the company. Maintaining a precise inventory helps determine current stock levels, optimize reordering, and support better decision-making. The continuous addition of new assets is essential for keeping the system up to date and reflecting the real-world inventory situation accurately.

Transferring Assets: Asset transfer refers to the process of moving recorded assets from one location to another within the organization using the asset management application. This process includes transferring the associated data and status updates along with the physical relocation. Such a feature is vital for tracking asset usage across multiple sites and

maintaining an accurate and synchronized inventory system across different localities or departments.

Deleting Assets: Deleting an asset involves the permanent removal of its associated records from the system. This action may be required for a variety of reasons: duplicate entries, damaged or obsolete equipment, recycling, or simply optimizing storage space. The removal process ensures that the system remains clean and reflects only assets that are in active use. Proper deletion procedures also contribute to compliance, data integrity, and operational transparency.

Modifying Assets: Modifying assets allows users to update or change the information tied to existing entries in the database, such as quantity, value, or location. This feature is essential for reflecting changes that occur over time due to usage, reallocation, depreciation, or adjustments. Keeping asset data accurate and current supports more reliable reporting, forecasting, and strategic planning. Ultimately, it ensures that decision-makers can act based on the most relevant and trustworthy information available.

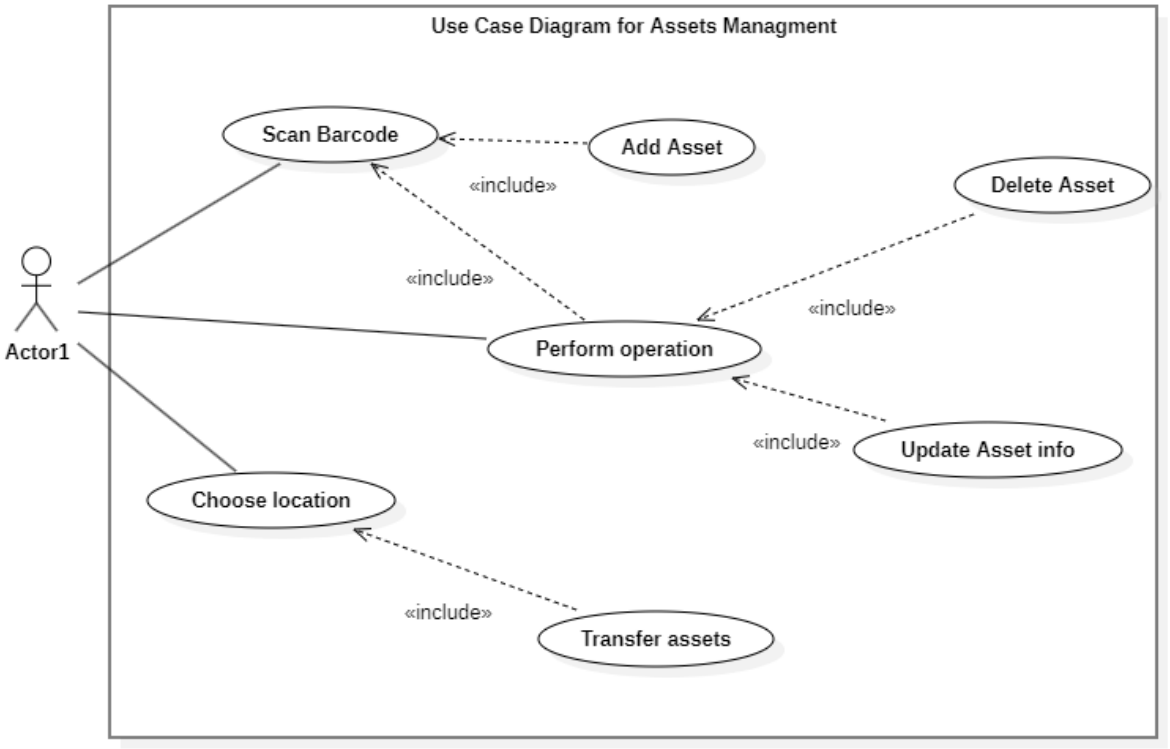

Figure 3.5 Goods Management Use Case Diagram

3.3.4.5 Use Case Diagram for biling managment use case

Verifying Invoice Information Against the System: After the data entry agents have completed their tasks within the asset management application, it becomes the accountant's responsibility to review and validate the invoice data to ensure it matches the information stored in the system. This verification process is essential for maintaining consistency and accuracy in financial records.

By comparing invoice details such as item quantities, pricing, and descriptions with the actual data recorded in the application, the organization can ensure that the billed assets correspond precisely to real transactions. This helps identify potential billing errors and prevents discrepancies that could lead to overpayment or loss.

Furthermore, the verification process contributes to better inventory management. By cross-referencing invoiced quantities with actual stock levels, businesses can detect mismatches and use this insight to make informed decisions regarding future purchasing and restocking. This reduces the risk of overstocking or understocking, thus optimizing operational efficiency. Additionally, this practice enhances transparency and trust among stakeholders. Providing accurate and verifiable invoice data to both clients and suppliers improves business relationships and fosters confidence in the company's financial integrity. A reliable invoice verification process is also important for auditing purposes and regulatory compliance.

Resending Invoices to Data Entry Agents: When a data entry agent makes an error in the asset management application, it can result in significant issues for the company, including stock inconsistencies, shipment delays, and financial discrepancies. To address these situations efficiently, a structured correction procedure must be in place.

One effective corrective measure is for the accountant to resend the affected invoice back to the data entry agent. This action includes highlighting the specific error, such as incorrect item names, wrong quantities, or mismatched supplier details, and providing all relevant references (invoice number, asset name, supplier, quantity, etc.). The accountant is solely responsible for initiating this resending process.

This practice ensures that errors are corrected at their source and helps maintain the accuracy of the inventory records. Providing clear instructions and necessary context allows the agents to amend the data swiftly and appropriately. Timely corrections prevent cascading errors that could affect reporting, analytics, and operational performance.

In summary, verifying invoice data and implementing a clear invoice correction workflow are critical components of an effective asset management system. They uphold data integrity, minimize financial risk, and promote internal accountability and process efficiency.

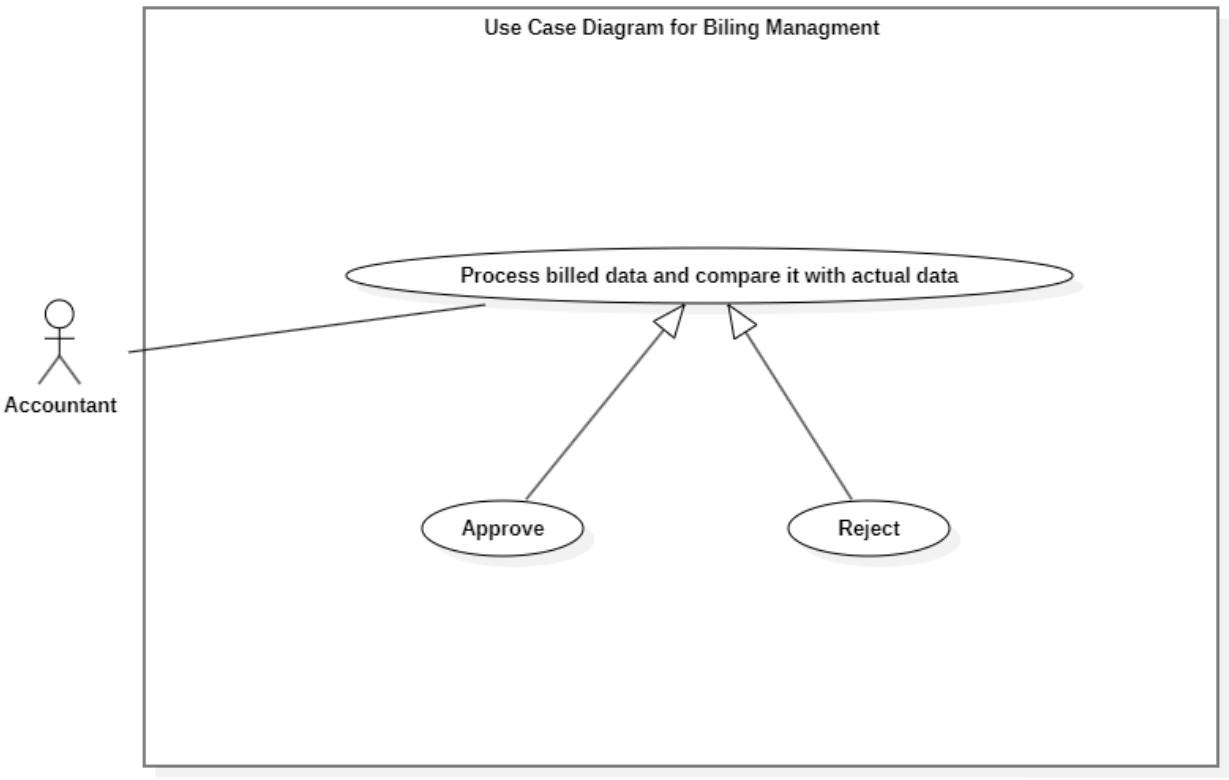

figure 3.6 Biling Management Use Case Diagram

3.3.5 Sequence Diagram

3.3.5.1 Definition of the Sequence Diagram

The sequence diagram is a modeling tool that visually and sequentially represents the interactions between objects in a software system or process. It illustrates how objects communicate with each other by exchanging messages and highlights the chronological order of these interactions.[12]

3.3.5.2 Representation of the Sequence Diagram for authentification

This scenario describes the process of authenticating an administrator. The authentication begins when the administrator enters a valid username and password combination. These credentials are transmitted to the system, which queries the database to verify the information.

If the entered credentials are correct and successfully validated by the database, the administrator is granted access to the desktop application. Otherwise, the system notifies the administrator of the failure and prompts them to re-enter the correct information.

This interaction ensures a secure and controlled access mechanism to protect sensitive administrative features. The sequence diagram visually outlines this flow, showing the order of communication between the user interface, the authentication service, and the database. It also helps identify potential points of failure and improve the robustness of the login process.

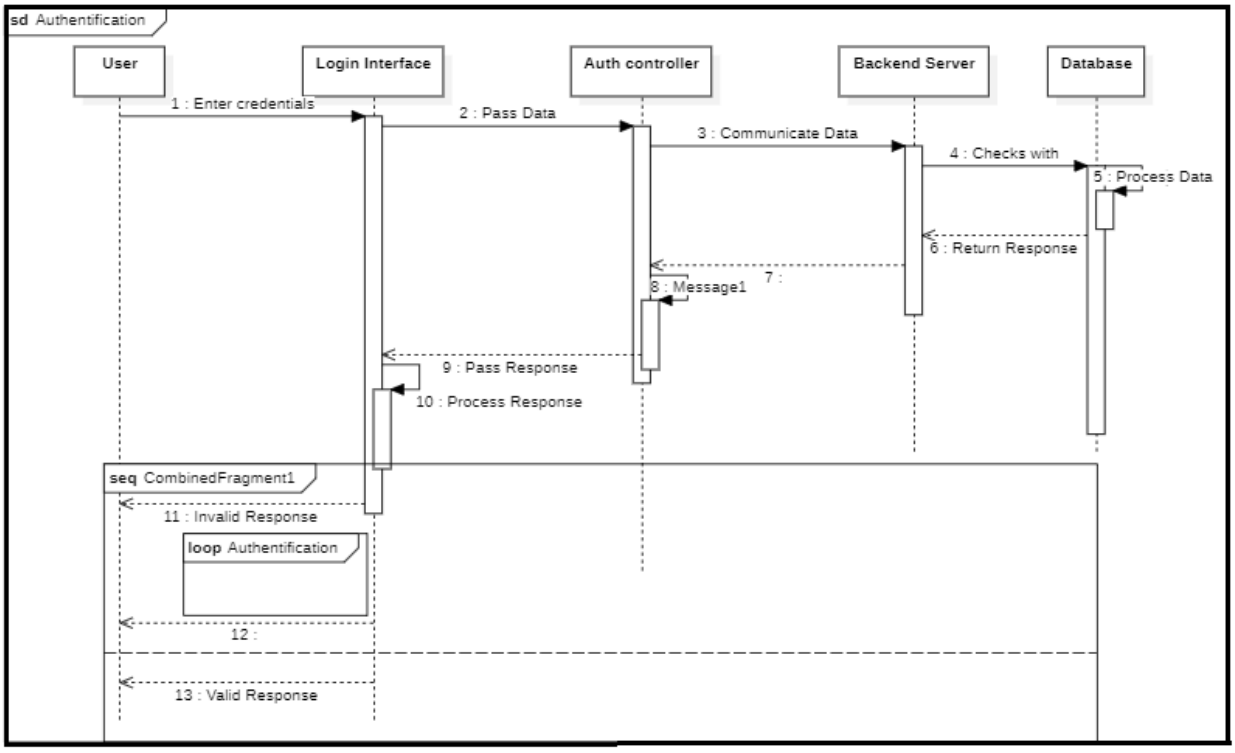

Figure 3.7 Authentification Proccess Sequence Diagram

3.3.5.3 Sequence Diagram for Account Management

This scenario describes the responsibilities of the administrator, which include the creation, modification, and deletion of user accounts such as data entry agents and accountants.

The administrator starts by entering the application through successful authentication. Once logged in, the administrator is presented with a dashboard offering access to user management features.

Creating a user involves entering key information such as username, password, role type, and department. This information is then validated and stored in the central database. Modifying a user account allows the administrator to update existing credentials or user roles, ensuring the right access levels are maintained across the organization.

Deleting a user account is reserved for cases where an agent is no longer part of the organization or when access needs to be revoked for security reasons. However, administrators cannot modify or delete their own accounts, a restriction put in place to prevent potential misuse and to uphold the security policy of the system.

These user management tasks are crucial for maintaining operational security and ensuring that only authorized personnel have access to the system based on their roles and responsibilities.

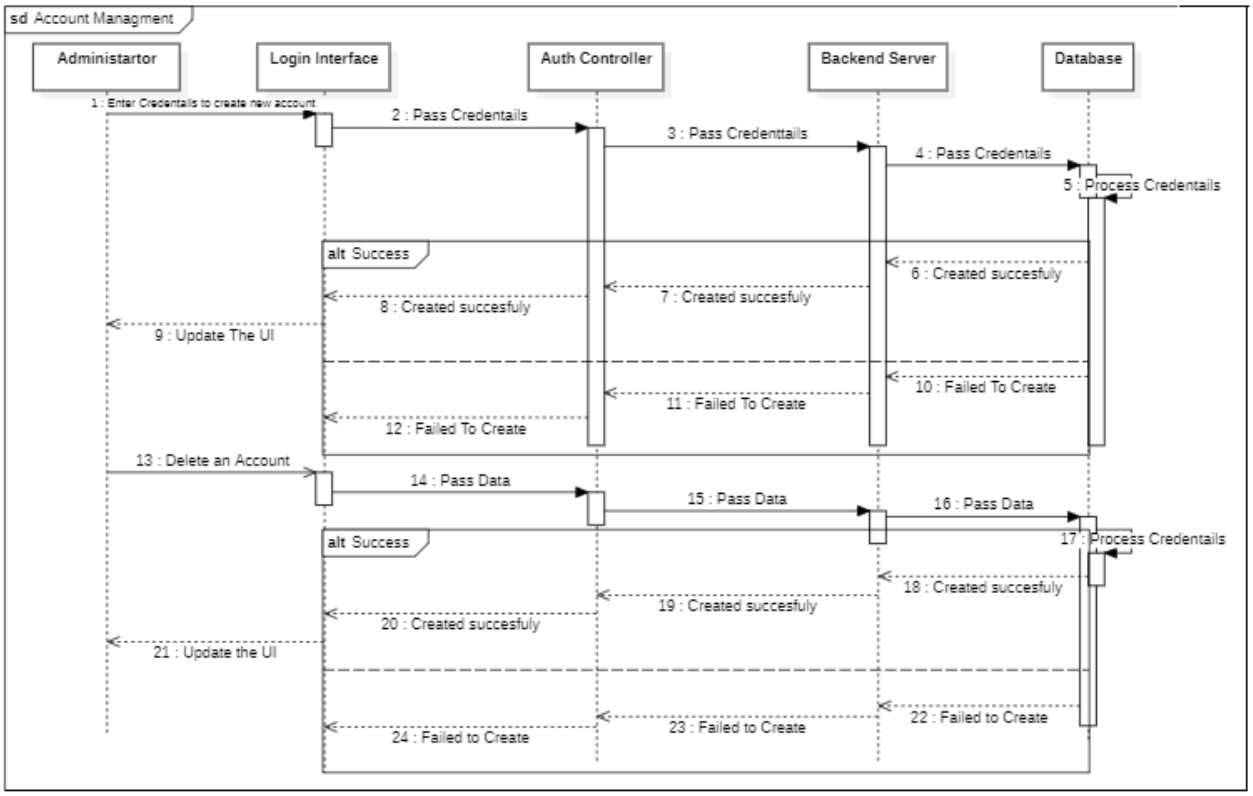

Figure 3.8 : Account Management Sequence Diagram

3.3.5.4 Sequence Diagram for Goods Managment

Barcode Scanning and Asset Synchronization Scenario

This scenario describes how the Desktop application, when used with barcode scanners available within the organization, enables users to scan asset barcodes efficiently.

Once a barcode is scanned, the associated data—such as the asset identifier, type, location, and status—is immediately recorded. This information is securely transmitted and stored in a centralized MongoDB database, where it is organized into collections to ensure efficient access and robust data integrity. The backend system ensures that each asset record is uniquely identifiable and that all updates are traceable.

Simultaneously, scanned asset files such as documentation or images are uploaded to Firebase Storage. The web application, which is integrated with the same backend ecosystem, retrieves this information and presents it through a user-friendly interface.

Data entry agents can then easily view the scanned asset details in real-time, verify their status, and carry out related operations such as edits, transfers, or deletions. This system minimizes manual errors, improves traceability, and supports efficient asset management across multiple departments or locations.

This scenario highlights how combining barcode technology with cloud storage and synchronized desktop–web integration leads to streamlined workflows and more reliable inventory data.

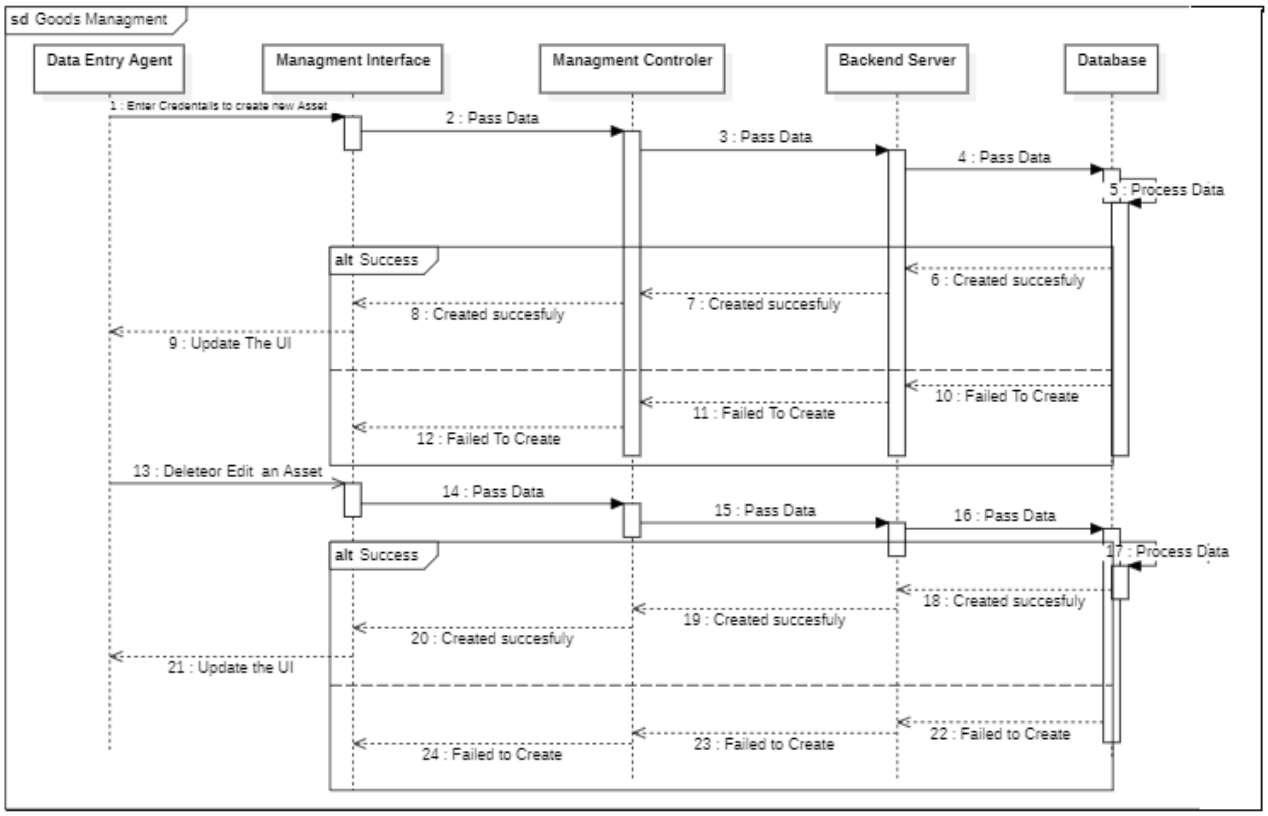

Figure 3.9 : Goods Managment Sequence Diagram

3.3.6 Class Diagram

3.3.6.1 Definition of the Class Diagram

The class diagram models the static structure of a system in terms of classes and the relationships between them . It serves as a blueprint for the system architecture by defining the key building blocks and how they are connected. This type of diagram is fundamental in object-oriented analysis and design.[13]

3.3.6.2 Components of a Class Diagram

Class — A class is an abstract description of a set of objects that share common properties within the domain of the application. It defines the structure (attributes), behavior (operations), and relationships of its instances. Classes are the core elements in object-oriented modeling and represent entities from the problem domain.

Attribute — An attribute is a named property of a class that describes the range of values its instances can possess. Attributes represent characteristics or data associated with an object of the class. A class may have no attributes or multiple attributes, depending on its purpose in the system.

Operation — An operation is an abstraction of what an object of a class can do. It represents a specific action or behavior that the object can perform. Operations define the functionalities available to instances of a class, allowing objects to interact with one another or manipulate their own data.

Relationships Between Classes — UML provides different types of relationships that define how classes are connected. These semantic links model how objects interact and depend on each other in an object-oriented system.

- **Association** — A structural relationship that specifies that instances of one class are connected to instances of another. It defines a link between two classes, often representing a usage or communication path.
- **Aggregation** — A special form of association representing a whole–part relationship. It indicates that one class can be composed of multiple instances of another class. This type of relationship is used when components can exist independently of the aggregate.
- **Composition** — A strong form of aggregation where the lifecycle of components is tied to the lifecycle of the composite. If the composite (whole) is created or destroyed, its components are also created or destroyed accordingly. This ensures tight containment and ownership of components.
- **Generalization** — This relationship models class hierarchies, where a child class inherits attributes and operations from a parent class. The child class is more specialized and can override inherited behavior or extend it with new features. Generalization promotes code reuse and establishes polymorphism within the system.

These components together form a comprehensive structure that defines the internal logic and external relationships of the software system, allowing developers to visualize dependencies and make design decisions early in the development cycle.

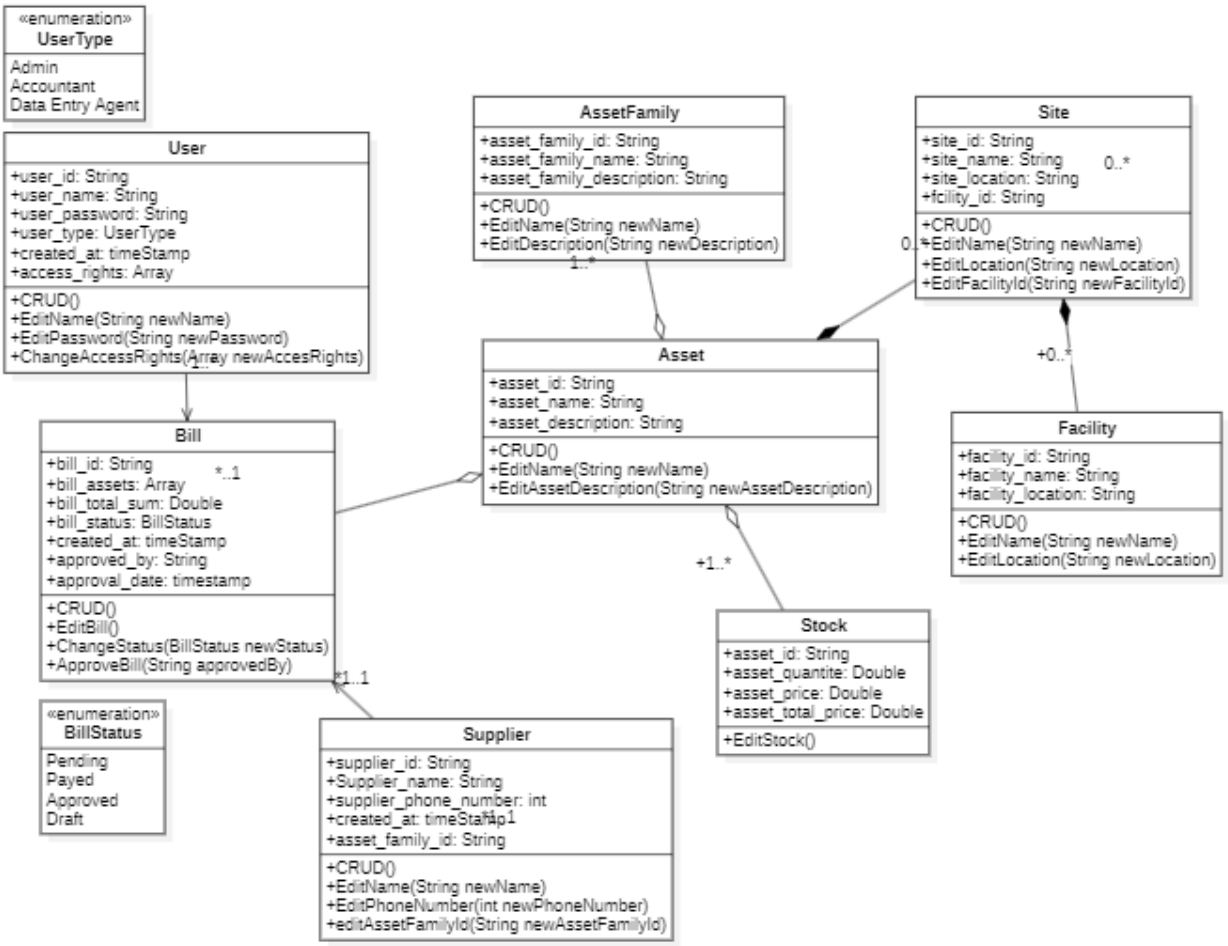

Figure 3.10 : General Class Diagram

3.4 Conclusion

The proposed design of this desktop asset management application using the UML (Unified Modeling Language) was essential to ensure an efficient development process and a shared understanding of the system's objectives.

By using UML's functional, structural, and static diagrams, the application's requirements were accurately captured, its internal structure and external behavior were effectively modeled, and the relationships between the system's key components were clearly identified.

This modeling process helped lay a solid foundation for implementation, improving clarity, minimizing development errors, and maintaining consistency throughout the software development lifecycle.

Chapter 4 : Implementation

4.1 Introduction

In this chapter, I present the implementation of the desktop application designed for asset management at the Sonatrach Center in Boumerdès. This desktop application, developed using the Flutter [16]framework, provides a complete and user-friendly interface that streamlines the process of managing physical assets across different departments and locations.

The application is built to support key functionalities such as scanning barcodes, adding, updating, transferring, and deleting assets, and reviewing asset details. Its design focuses on both usability and efficiency to ensure that daily operations are carried out with minimal friction.

For data management, I used MongoDB as the primary database system due to its flexibility in handling semi-structured data, which is suitable for asset records. The backend server is developed in Golang (Go), which ensures fast, concurrent, and scalable performance for processing and serving data.

Together, this technology stack—Flutter for the front end, Golang for the backend, and MongoDB for data storage—creates a robust ecosystem that supports real-time, efficient, and secure asset management operations tailored to the needs of Sonatrach's internal processes.

4.2 Hardware Environment

The development and execution of the desktop asset management application were carried out on the following hardware configuration:

- **Operating System:** Windows Server 2022 Datacenter (64-bit)
- **Processor:** AMD EPYC 9654 — 96 Cores / 192 Threads (2.4 GHz Base, up to 3.7 GHz Boost)
- **System Architecture:** x86_64 (AMD64)
- **RAM:** 128 GB DDR5 ECC Registered @ 4800 MHz
- **Storage:** 2 × 2 TB NVMe Gen4 SSD (RAID 1) + 4 × 4 TB NVMe Gen4 SSD (RAID 10)
- **Network:** Dual 10 GbE Ethernet Ports
- **Graphics Card:** NVIDIA A100 80 GB

This environment provides a stable and sufficient platform for developing and testing the Flutter-based desktop application, ensuring compatibility with modern system requirements and allowing smooth interaction with the MongoDB database and Go backend services.

4.3 Software Environment

4.3.1 Visual Studio Code (VS Code)

Visual Studio Code is a lightweight yet powerful source code editor developed by Microsoft. It is widely adopted by developers across various disciplines for building web applications, mobile applications, desktop software, and even games. Designed to support modern software development workflows, VS Code offers a seamless user experience and an intuitive user interface, which makes it particularly suitable for both beginners and experienced developers.

4.3.2 MongoDB

MongoDB is a widely-used, open-source, NoSQL document database designed to handle modern application data requirements. It is renowned for its flexibility, scalability, and capability to manage structured, semi-structured, and unstructured data. Unlike traditional relational databases (RDBMS) that store data in rigid tables with fixed schemas, MongoDB stores data as flexible, JSON-like documents grouped into collections. This makes it particularly suitable for applications requiring a dynamic and adaptable data model.

Key Features of MongoDB:

- **NoSQL Database:**
  MongoDB is a NoSQL database, meaning it does not rely on the rigid schema constraints of traditional SQL-based relational databases. This enables faster and more flexible development cycles.
- **Document-Oriented Storage:**
  Data is stored in documents using key-value pairs, similar in structure to JSON. Each document represents a real-world entity, which can be nested and indexed for efficient querying.
- **Flexible Schema:**
  Documents within the same collection do not require the same structure. This schema-less design allows developers to iterate quickly without needing to perform complex migrations.
- **Collections:**
  MongoDB organizes documents into collections, which serve a similar role to tables in relational databases but without the rigid schema requirements.
- **BSON Format:**
  MongoDB uses BSON (Binary JSON) for document storage. BSON extends the JSON model by including additional data types such as dates and binary data, optimizing both storage and query performance.
- **Scalability and High Availability:**
  MongoDB supports sharding (horizontal scaling) and replication (data redundancy) to ensure high performance, availability, and the ability to scale to massive datasets.
- **Common Use Cases:**
  MongoDB is commonly used in web and mobile applications, content management systems, real-time analytics platforms, and any scenario where flexible data modeling and scalability are critical.

- **Community and Ecosystem:**
  Backed by a large community and a rich ecosystem of tools, MongoDB offers comprehensive resources for development, data management, and performance monitoring.

In this project, MongoDB was chosen for its adaptability and its ability to efficiently manage complex asset-related data across the desktop and backend systems. Its seamless integration with Flutter and GoLang made it an ideal choice for storing and retrieving asset data in real-time.

4.4 Used Programming Languages:

4.4.1 Go Programming Language

Go, also known as Golang, is a general-purpose programming language developed by Google in 2007 and officially released in 2009. Designed by Robert Griesemer, Rob Pike, and Ken Thompson, Go was created to improve productivity in the era of multicore computing, large-scale distributed systems, and cloud-based development.

Go combines the simplicity and clarity of a dynamically typed language with the performance and safety of a statically typed language. It offers a clean and minimalistic syntax, making it easy to read and write, especially for beginners in programming.[14]

Key Features of Go:

- Simplicity and Clarity.
- Concurrent Programming.
- Cross-Platform Compilation.
- Robust Standard Library.
- Fast Compilation.
- Garbage Collection and Memory Safety.
- Strong Tooling Support.
- Widespread Adoption and Community.

Thanks to these features, Go is a good fit for our case study.

4.4.2 Dart Programming Language

Dart is a modern, object-oriented, general-purpose programming language developed by Google and first released in 2011. Originally designed to replace JavaScript as the primary language for client-side web development, Dart has since evolved into a versatile language used in web, mobile, desktop, and server development—most notably as the backbone of the Flutter framework. Dart was originally built by Google engineers Lars Bak and Kasper Lund, and it was forked from Google's own V8 JavaScript engine, the same engine that powers Google Chrome. Due to this origin, Dart shares many syntactic similarities with

JavaScript and TypeScript, making it approachable and intuitive for developers coming from those ecosystems.[19]

Key Features of Dart:

- Approachable and Safe.
- Cross-Platform and Portable.
- High Productivity.
- Optimized for UI Development.
- Object-Oriented and Familiar Syntax.
- Robust Package Ecosystem.
- Used by Google and Beyond.

Thanks all of this , Dart is a good fit for our case study.

4.4.3 Flutter Framework

Flutter is an open-source UI software development kit (SDK) created by Google. It was first announced in 2015 and officially released in its stable version in December 2018. Designed to enable developers to build high-performance, visually attractive, and natively compiled applications from a single codebase, Flutter has quickly become one of the most popular cross-platform development frameworks worldwide.

Flutter allows for the development of applications for multiple platforms including Android, iOS, Windows, macOS, Linux, the web, and even embedded systems—all using the same codebase. Its ability to produce native ARM machine code ensures optimal performance, while its expressive UI toolkit provides fine-grained control over every pixel rendered on screen.[15]

Key Characteristics of Flutter:

- Developed and Maintained by Google.
- Cross-Platform Capabilities.
- Built on Dart.
- Beautiful and Customizable UI.
- High Performance.
- Hot Reload and Fast Development.
- Open Source and Growing Ecosystem.
- Web and Desktop Support.
- Strong Community and Adoption.
- Integration with Native Code. [21]

Thanks to all of this , Flutter is a good fit for our case study.

4.5 User Interface Design (UI) :

4.5.1 Loading Screen

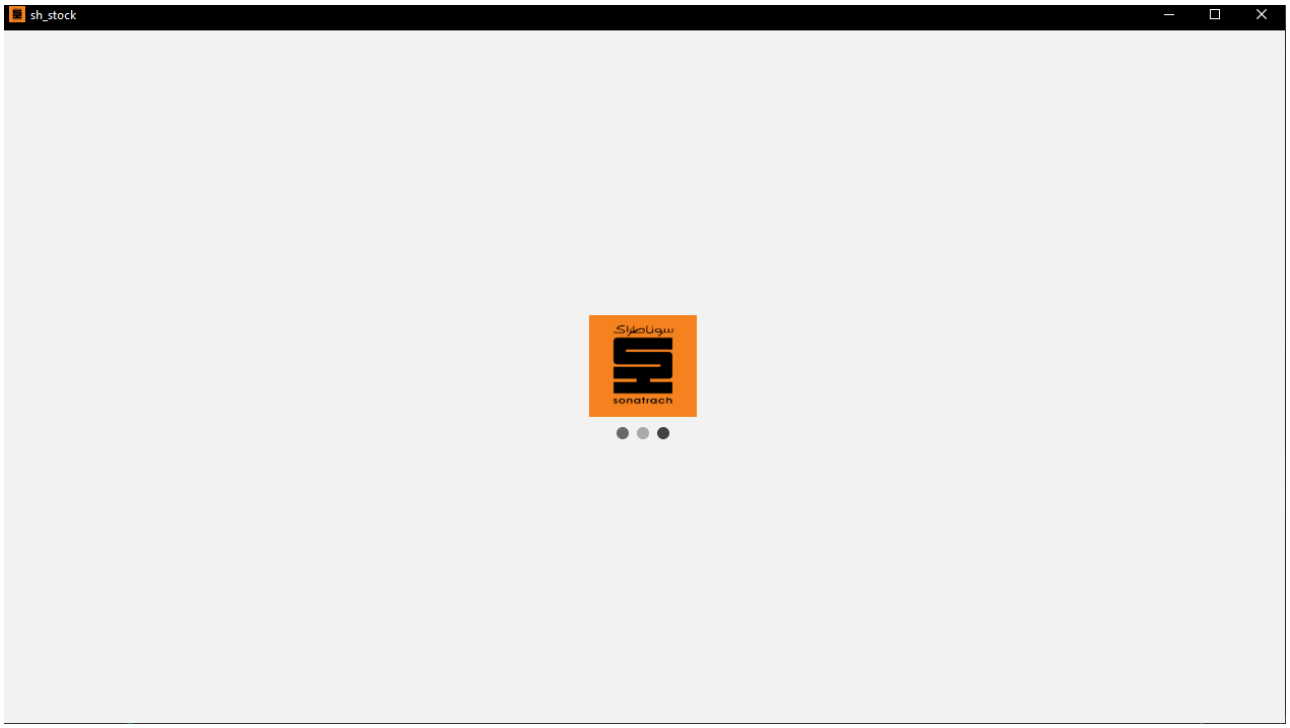

Figure 4.1 : Loading Screen

On this screen, the app loads and checks for the availability of the necessary resources, displaying any errors if they occur.

4.5.2 Authentification Screen empty

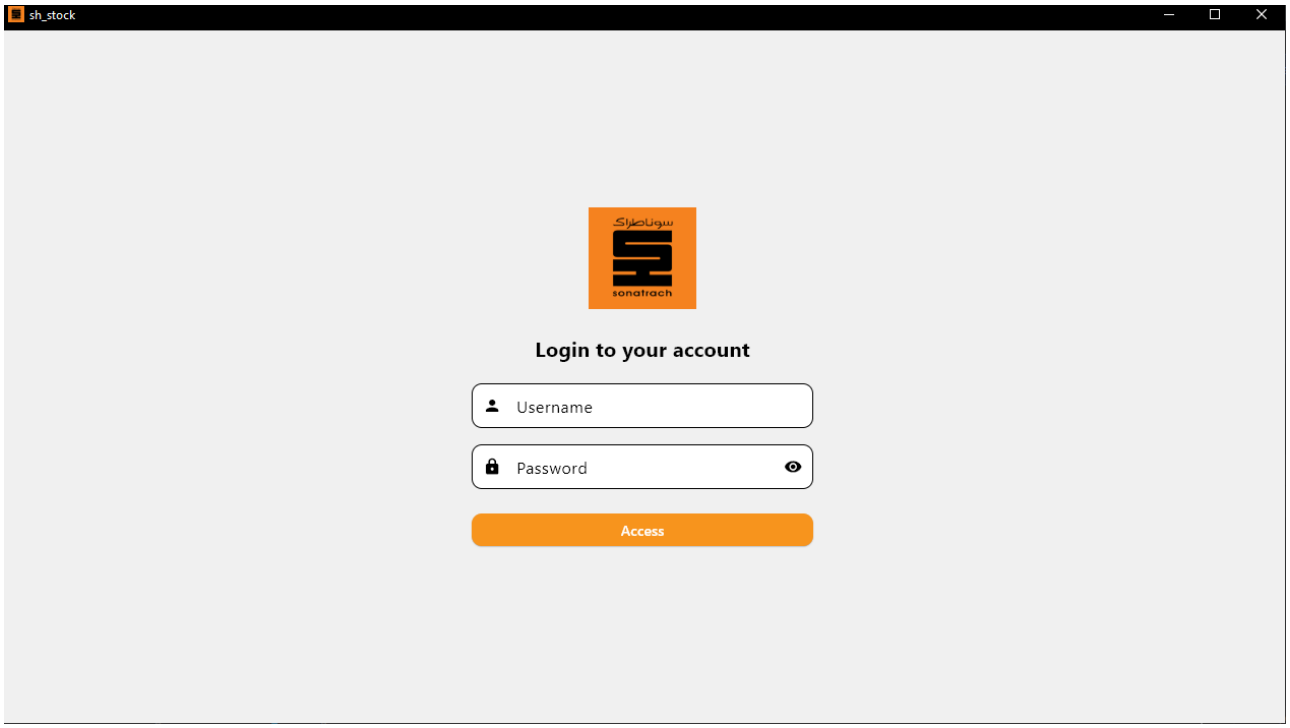

Figure 4.2 : Authentification Screen empty

On this screen, users can authenticate using their username and password.

4.5.3 Authentification filled

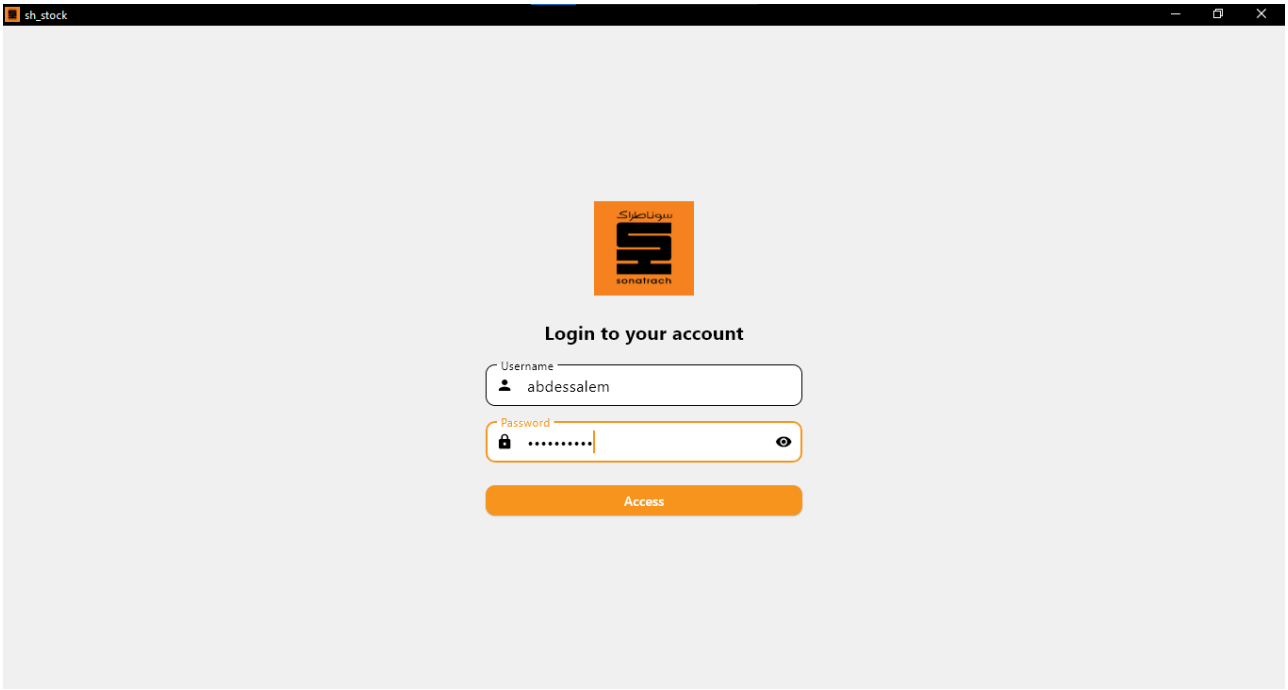

Figure 4.3 : Authentification Screen empty filled

This interface represents the **authentication screen** for the Sonatrach asset management desktop application. Its purpose is to allow authorized users to securely log in to the system using their credentials.
The login form includes:

- A **Username field** to input the user's identifier.
- A **Password field** with optional visibility toggle for enhanced usability.
- A **Login button** .

An organization logo is placed prominently at the top to reinforce branding. The entire interface is centered and constrained for better aesthetics and accessibility across various screen sizes.
Upon clicking the login button, the application collects the entered credentials and processes them for authentication. Although this example prints the credentials to the console for demonstration purposes, in a production environment, these values would be securely validated against a backend system.
This authentication step is critical for ensuring that only authorized personnel can access the administrative functions of the application.

4.5.4 Admin Panel Screen

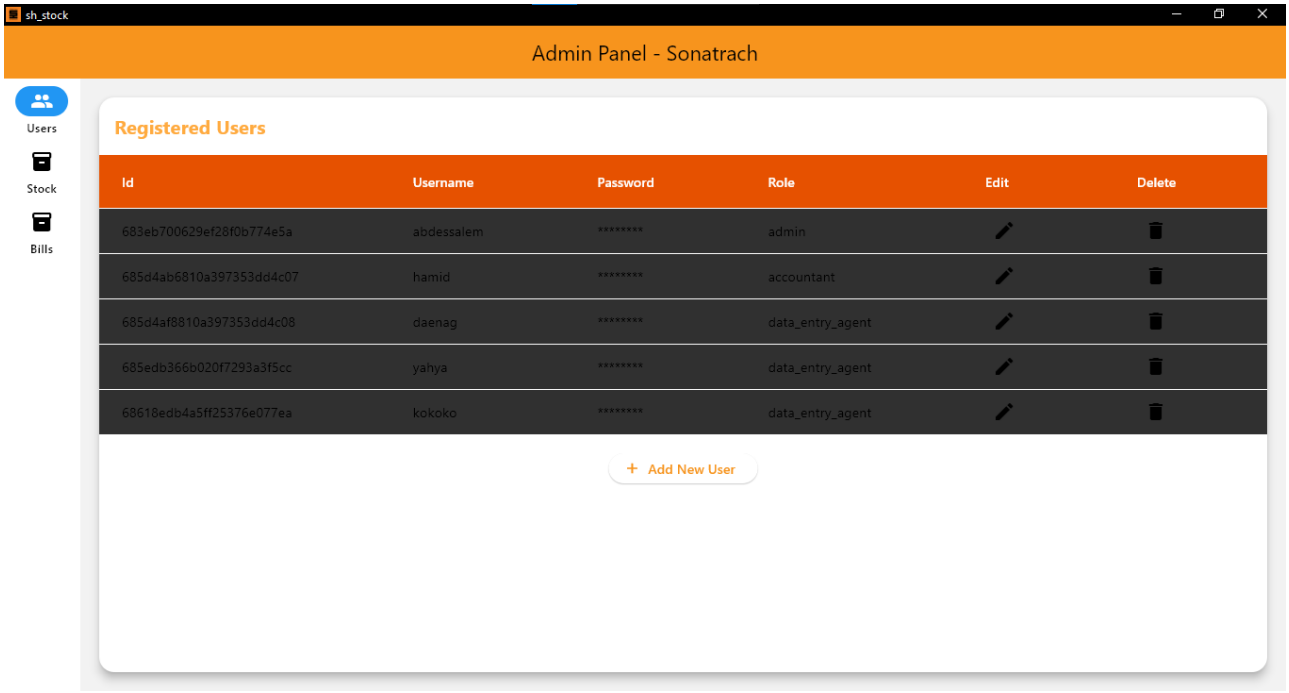

Figure 4.4 : Admin Panel screen

Figure 4.5 : Edit Users Dialog

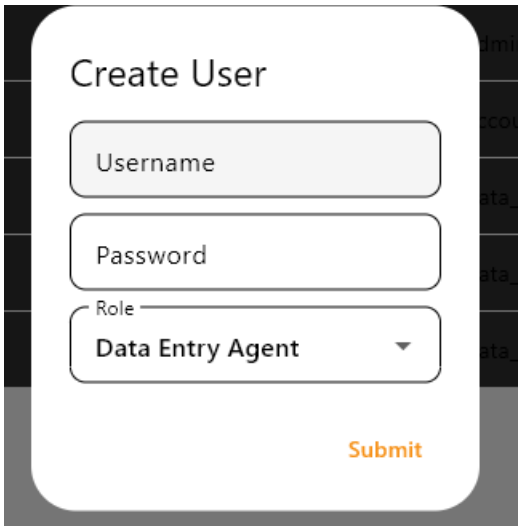

Figure 4.6 : Create Users Dialog

This interface represents the **Administrator Panel** of the desktop asset management application. Designed with a user-friendly layout and a color scheme inspired by Sonatrach's corporate identity (orange and black), the interface provides administrators with a centralized view and control panel to manage user accounts. The main component of this screen is a **data table** that displays the list of registered users within the system. Each row in the table corresponds to a specific user and includes three essential pieces of information:

- **Username** – the unique identifier used by the user to log into the system.
- **Password** – the authentication credential associated with the user .
- **Role** – the role or level of permission granted to each user, such as "Data Entry Agent", "Accountant", or "Administrator".
- **Edit And Delete options**

This panel enables the administrator to review user accounts at a glance and ensure proper distribution of access rights throughout the system. It plays a crucial role in maintaining system security, user accountability, and overall organizational control, it also contains other tabs to create new users and see bills and stock movements.

4.5.5 Stock Summary Screen

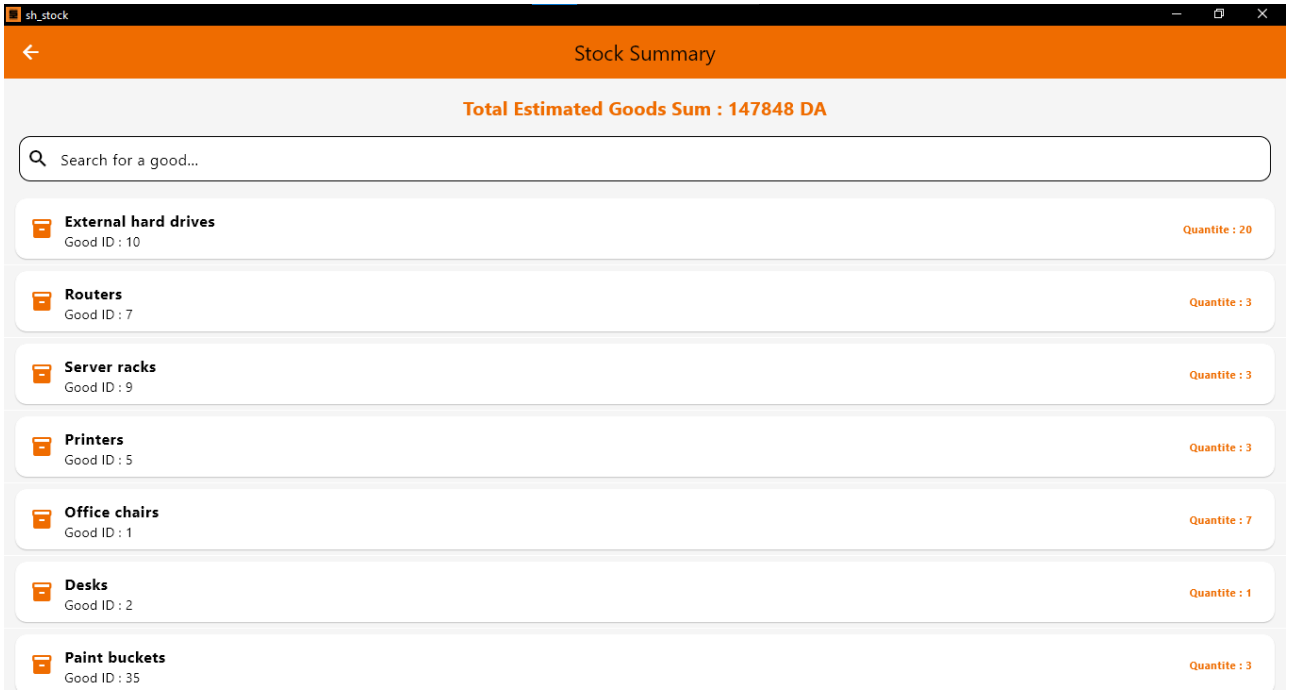

Figure 4.7 :Stock Summary screen

On this screen the user can view the current stock status and search for items in it.

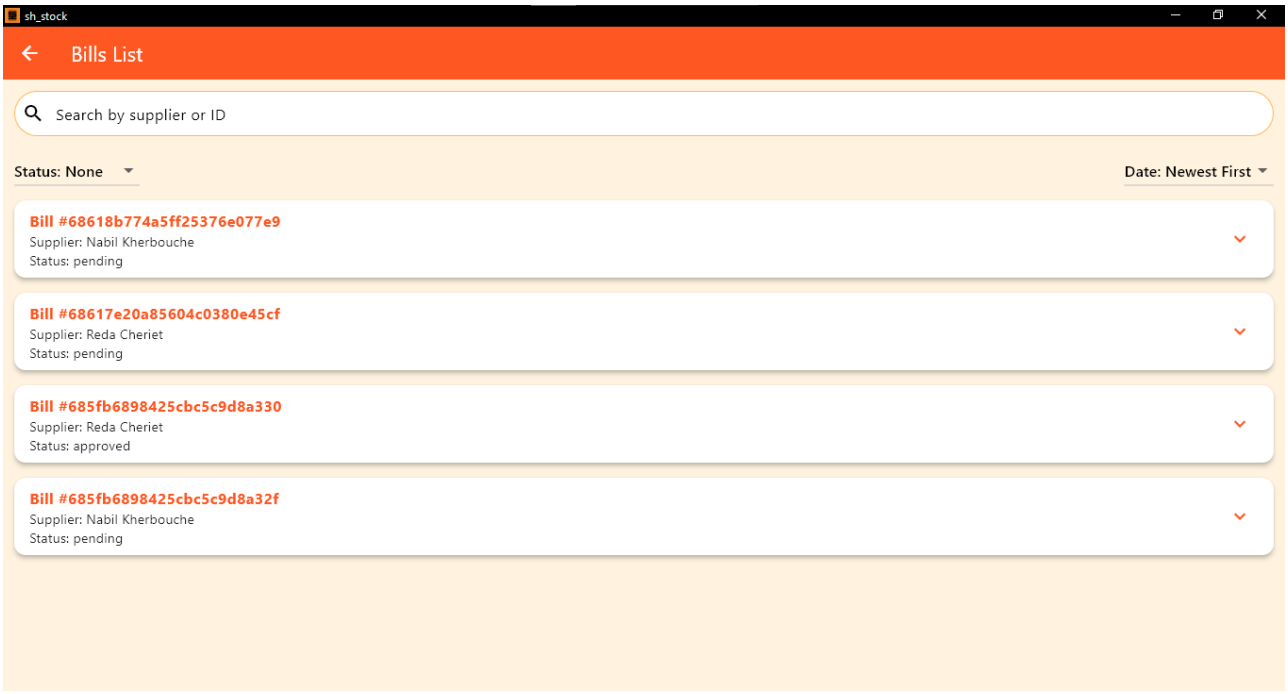

Figure 4.8 : Bills List screen

On this screen the user can view the bills that only he have access to.

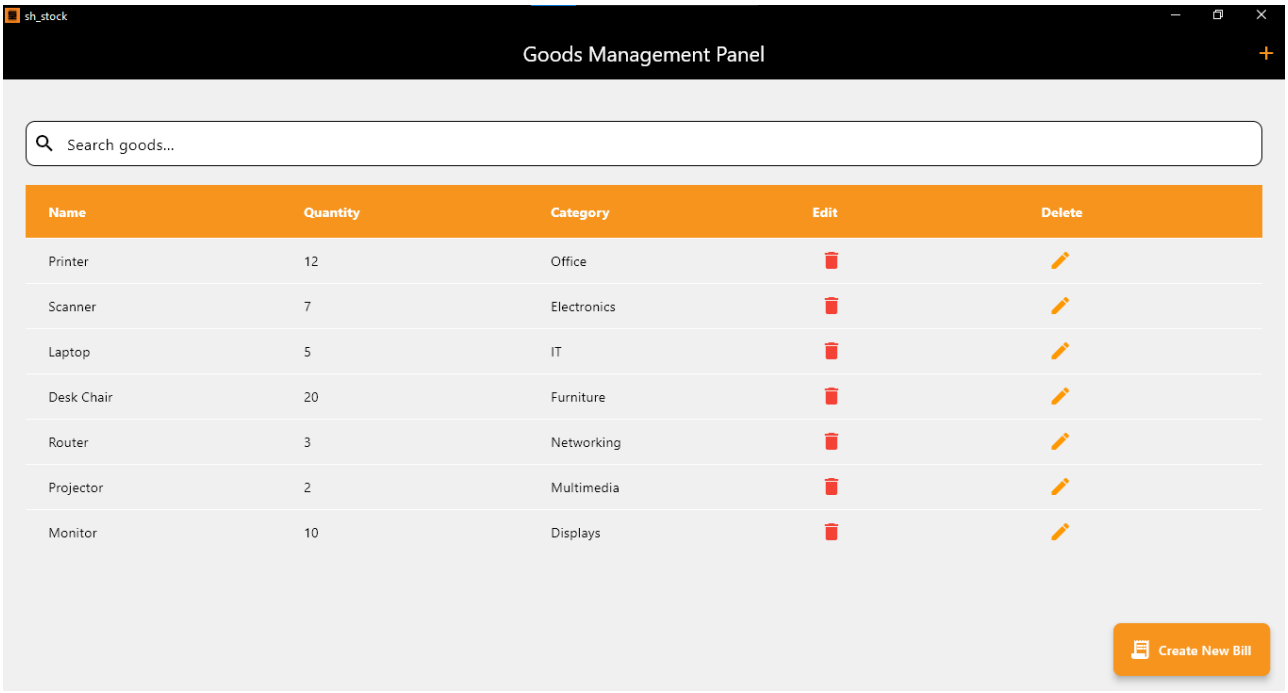

Figure 4.9 : Goods Management screen

The **Goods Summary Screen** is an essential part of the desktop application designed for the agents responsible for managing inventory and assets within the Sonatrach center. This screen provides an intuitive and efficient interface for viewing, editing, and organizing goods.

Key Components of the Interface:

- **Search Bar:** Positioned at the top of the screen, the search bar allows users to quickly filter goods based on keywords in their names or categories. It supports real-time searching as the user types, helping to locate specific items in large datasets easily.
- **Data Table:** Below the search bar, a structured data table displays the list of goods with the following columns:
  - **Name:** The name of the item (e.g., Printer, Laptop).
  - **Quantity:** The number of available units.
  - **Category:** The classification of the good (e.g., Office, IT, Furniture).
  - **Actions:** Includes buttons to *edit* or *delete* items, allowing for immediate updates to the inventory.

4.5.6 Supplier Managment Screen

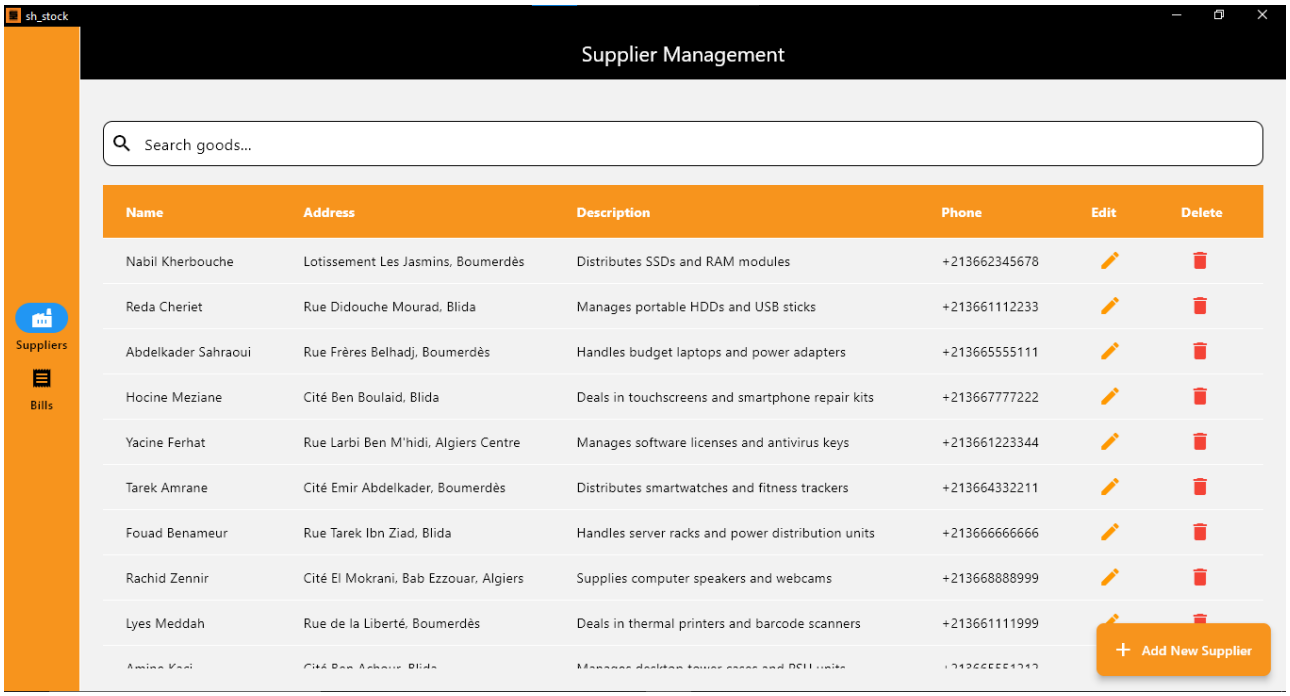

Figure 4.10 : Supplier Management screen

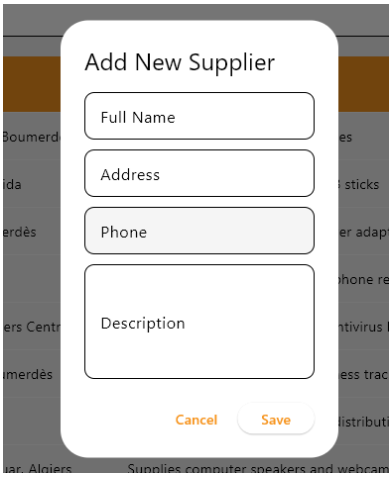

Figure 4.11 : Edit Supplier info

Figure 4.12 : Search For Supplier

The **Supplier Management Screen** is a dedicated interface that allows administrators or agents to manage the list of suppliers associated with the system. It is part of the goods and logistics workflow and ensures that all supplies are linked to valid, up-to-date supplier entries.

4.5.7 Bill Management

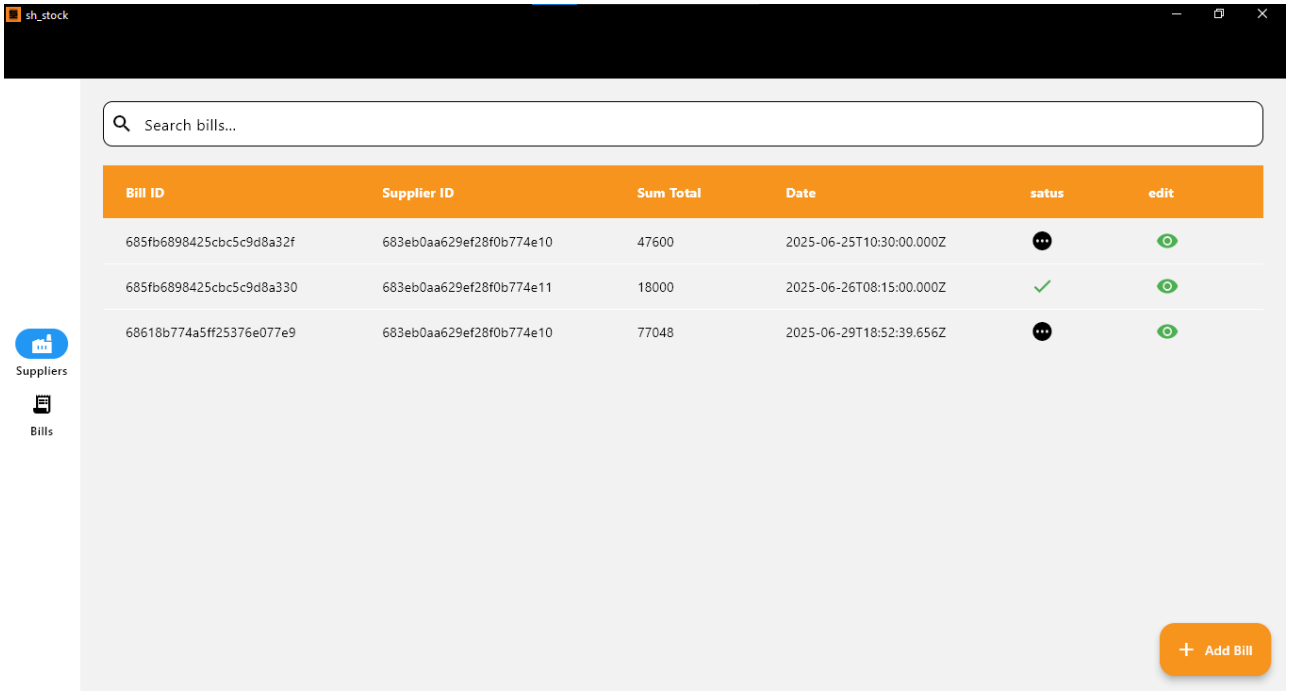

Figure 4.13 : Bills Management screen

On this screen users can check search for bills based on their id and check the status of these bills.

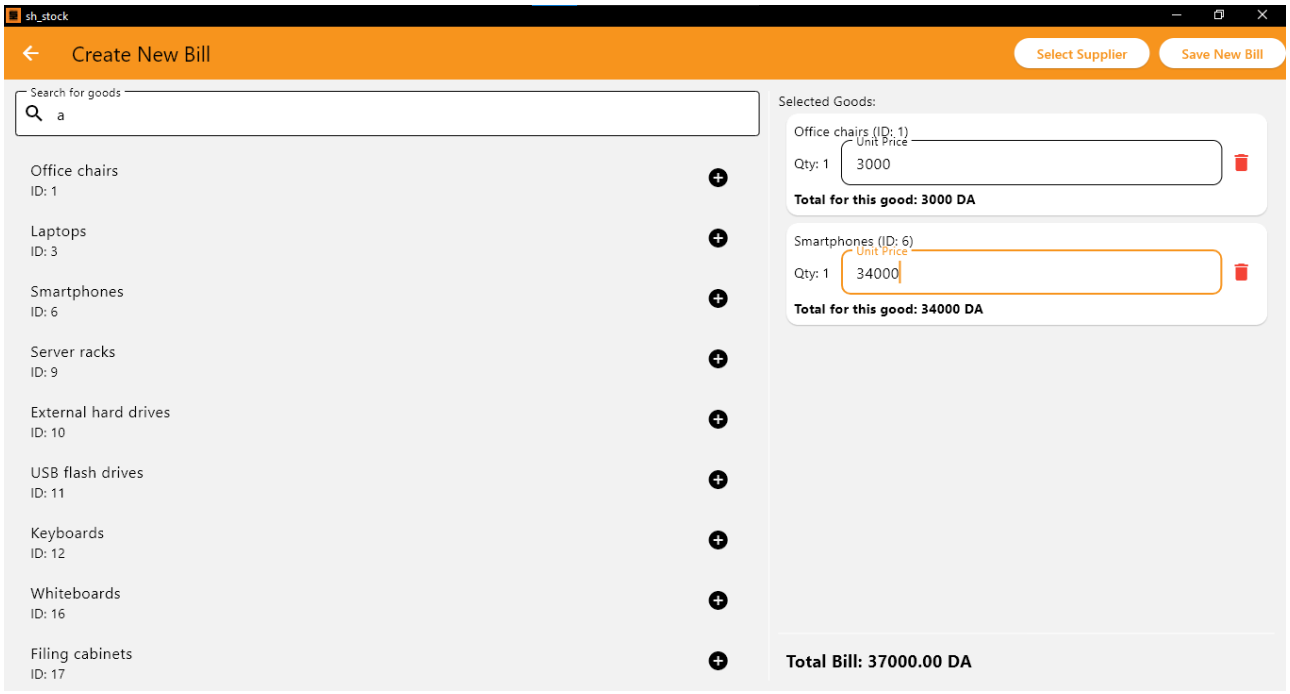

Figure 4.14 : New Bill Screen

On this screen users can view/edit bills contents.

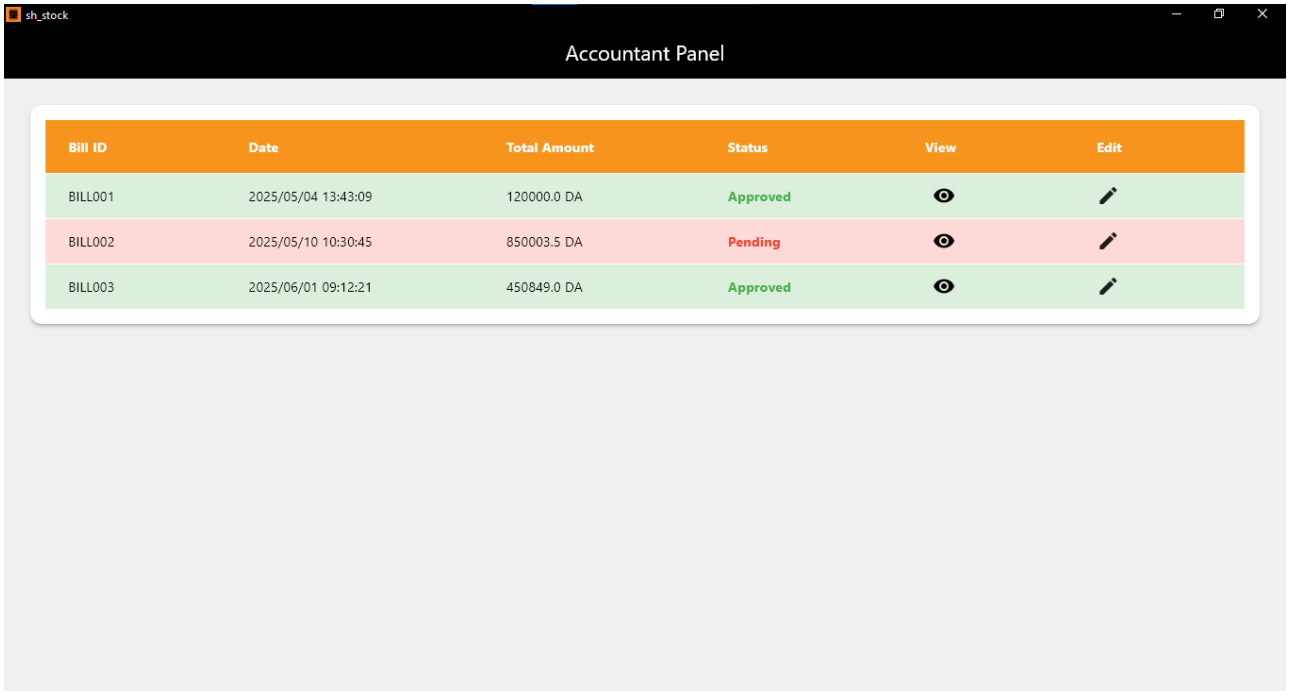

Figure 4.15 : Bills Status Screen

On this screen users can view and edit bills status.

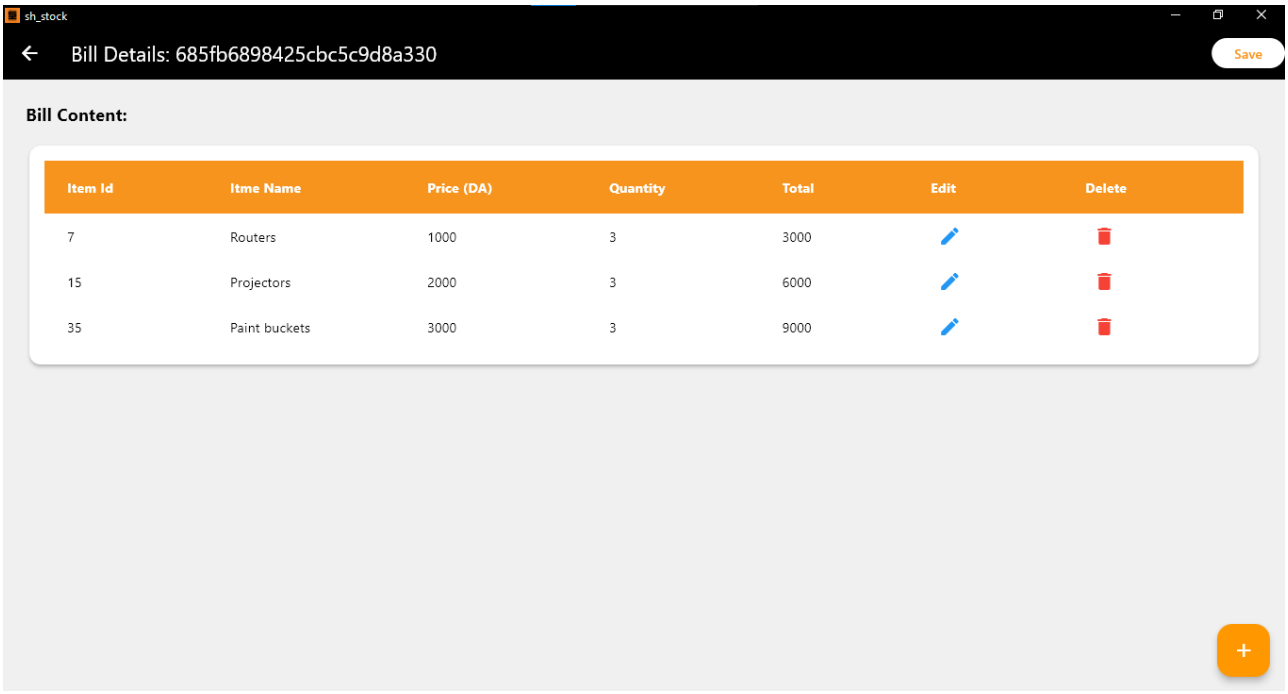

Figure 4.16 : More Details About Bills Screen

On this screen users can view/edit bills contents.

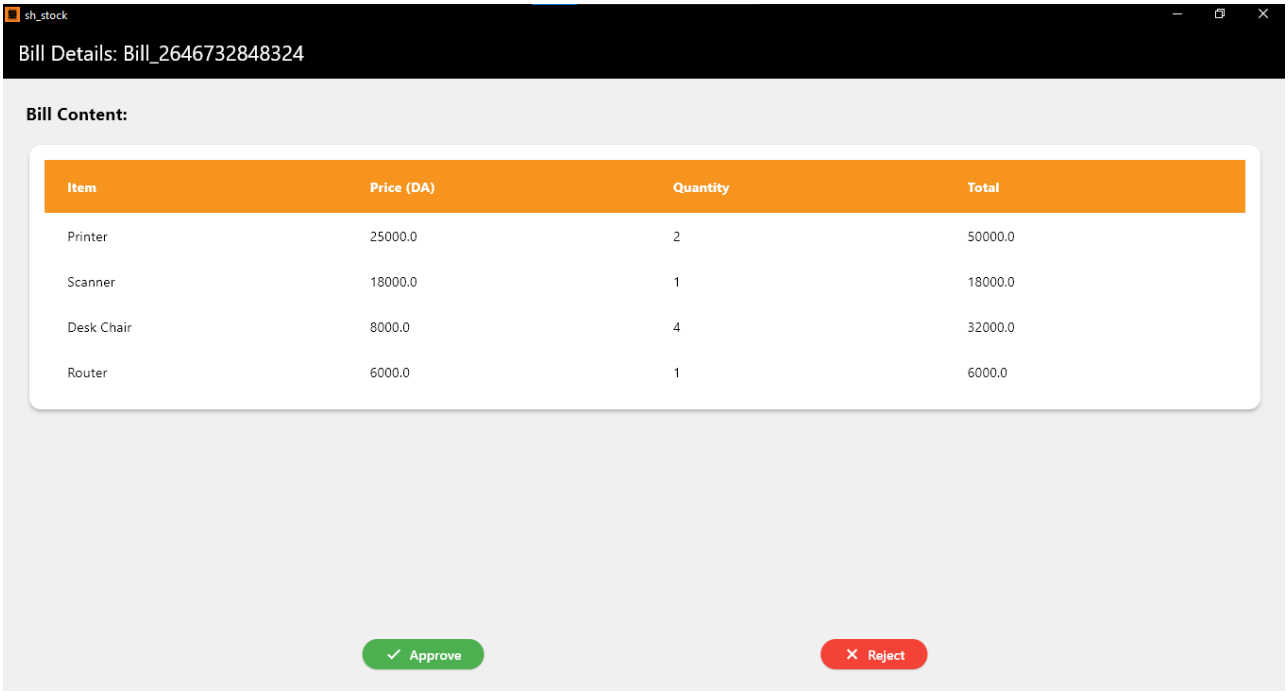

Figure 4.17 : Approve/Reject Bill Screen

This screen accountants to monitor, review, and manage bills. The main functionalities and design choices of this screen are as follows:

- **Bills Data Table:** The core element of the interface is a DataTable that lists bills. Each bill is represented as a row in the table, displaying:
  - **Bill ID** (e.g., BILL001)
  - **Date and Time** (timestamp of the bill creation)

- **Total Amount** (in Algerian Dinar)
- **Status** (either *Approved* or *Pending*)
- **View** icon button to inspect the full content of the bill
- **Edit** icon (placeholder for future features)

- **Status Highlighting:** Rows in the table are color-coded for clarity:
  - Green for approved bills
  - Red for pending bills

This interface facilitates smooth interaction for accountants managing large sets of bills and ensures clarity in financial record review and decision-making.

4.5.8 Barcode Scanner Interface

Figure 4.18 : Barcode SCanning Sczreen

This interface represents a conceptual design of a barcode scanner screen within the Sonatrach-themed application, using orange and black colors. The purpose of this screen is to provide a user-friendly layout where a data entry agent or storekeeper can simulate scanning barcodes of goods or assets.

4.6 Results

The final outcome of this project is a modern, reliable, and user-friendly desktop stock management system specifically tailored to meet the operational needs of Sonatrach's logistics department. The application empowers users to manage inventory, track stock

movements, access historical records, and generate summary reports through an intuitive and responsive interface.
Developed using modern software frameworks, the system performs efficiently across multiple operating systems (Windows, Linux, and macOS) and integrates seamlessly with the company's existing LAN infrastructure, requiring no modifications. A major accomplishment was ensuring compatibility with the wide range of barcode scanners already in use, enabling rapid item identification and significantly reducing manual input errors.
The system also overcame platform-specific networking challenges—particularly on macOS—ensuring stable and consistent communication across all environments. Compared to the previously used outdated system, this new solution offers superior performance, increased stability, and an enhanced user experience. It establishes a practical and scalable foundation for future improvements and may be extended to other departments encountering similar inventory management needs.

Aspect	**Previous System**	**New System**
Source Code Ownership	Not available (third-party closed system)	Fully open and maintainable by Sonatrach
Performance	Frequent crashes, slow queries	Stable, fast response times, optimized handling of operations
Platform Support	Tied to outdated Linux setup	Cross-platform: Windows, Linux, macOS
Scalability	Difficult to update or extend	Modular and future-proof architecture
Database Technology	Rigid, old relational database (Oracle 11)	Flexible, schema-less modern database for easier data handling
User Interface	Outdated, not intuitive	Modern, clean, and user-friendly interface
Barcode Integration	Limited or incompatible	Supports multiple existing barcode readers
Network Setup	Legacy server configuration, hard to modify	Compatible with current LAN, adapted even for macOS
Data Accuracy	Delayed or missing updates	Real-time updates, consistent data integrity
Maintenance	Dependent on external vendors	Internal teams can maintain and improve the system

Table 4.6 Comparison between the Previous and New System

This table represents a detailed comparison between the previously used inventory management system and the newly developed one. Each row highlights a critical operational or technical aspect, illustrating how the new system improves upon limitations of the former. Key improvements include full source code ownership, enhanced performance and stability, broader platform compatibility, and significantly improved scalability and maintainability. These advancements not only resolve past inefficiencies but also position the system for long-term adaptability and internal control.

4.7 conclusion

This combination of tools and technologies resulted in a comprehensive goods management solution that enhances efficiency, ensures reliable data handling, and provides a smooth user experience for the Sonatrach center in Boumerdes.
The next chapter presents the general conclusion of this study, summarizing the main results, highlighting the achieved objectives, and suggesting possible improvements for future work.

General Conlusion

At the end of this study project, a modern desktop application was designed and developed to manage goods at the Sonatrach center in Boumerdès. This solution replaces the old manual system, which was often subject to human errors, made asset tracking difficult due to the absence of a robust structure, slowed down the transmission of information, and lacked adequate data security.

By incorporating modern development tools and an intuitive user interface, the new application enhances efficiency, accuracy, and reliability. It enables users to perform daily tasks more easily and securely, while establishing a foundation for potential future extensions such as barcode scanning or integration with other internal systems.

SH_STOCK provides an effective solution by ensuring better organization of data, which simplifies information retrieval and management. It relies on a centralized database that stores all relevant information about goods, including their status, quantity, and assignment details. Furthermore, the system integrates user account management features that allow the administrator to adjust account settings and remove data entry or accountant accounts when necessary, especially in the case of security concerns. This contributes to a safer and more structured working environment for all users.

The user interface follows a clean and intuitive layout that ensures a pleasant and straightforward experience. The design allows users—regardless of their technical background—to easily navigate **SH_STOCK** and complete their tasks efficiently. Thanks to its simplicity and logical structure, the interface supports effective asset management without confusion or the need for extensive training.

Accessibility, readability, and ease of use were prioritized to deliver a practical and professional interface tailored to the operational needs of Sonatrach staff.

The UML methodology was used to model the system, clearly defining its components and their interactions. This step was crucial in establishing a solid foundation for the application's architecture. Once the design was completed, these concepts were implemented using appropriate tools and technologies to develop the **SH_STOCK** solution efficiently and reliably.

The study project met the specific needs of the Sonatrach center in Boumerdès by delivering a complete digital solution for asset management. Through **SH_STOCK**, the system provides improved data organization, secure account management, a user-friendly interface, and increased visibility for the site's users. This application contributes to better asset tracking and management within the center by optimizing processes and reducing the risks of errors and data loss.

It is important to note that this work is not limited to the case of the Sonatrach center. The proposed solution can be adapted and extended to other organizations or sectors facing similar challenges in asset management. Digital solutions represent powerful tools for automating and improving management processes, which are essential for maintaining competitiveness and efficiency in a constantly evolving environment. Future developments may include greater automation of asset management through the integration of smart sensors capable of monitoring real-time usage, detecting anomalies, or issuing maintenance alerts.

Bibliography

[1] Donald E. Knuth, *The Art of Computer Programming*, Addison-Wesley, 1997.

[2] Robert Davis, *An Introduction to Asset Management: A simple but informative introduction to the management of physical assets*, Published by blah d blah design Ltd (subsidiary of EA Technology Ltd), Capenhurst Technology Park, Chester, United Kingdom, 2010, ISBN 978-0-9571508-3-6.

[3] Zimmermann, A., Schmidt, R., & Jain, L. C. (Eds.). (2021). *Architecting the Digital Transformation: Digital Business, Technology, Decision Support, Management*. Intelligent Systems Reference Library, Vol. 188. Springer.

[4] Council of Supply Chain Management Professionals, *The Definitive Guide to Inventory Management: Principles and Strategies for the Efficient Flow of Inventory across the Supply Chain*, Pearson Education, 2014. Available at: https://www.oreilly.com/library/view/the-definitive-guide/9780133448856/.

[5] Marvin Waschke, *How Clouds Hold IT Together: Integrating Architecture with Cloud Deployment*, Apress, 2015. Available at: https://doi.org/10.1007/978-1-4302-6167-4.

[6] Dinesh Shenoy & Roberto Rosas, *Problems & Solutions in Inventory Management*, Springer Cham, 2018.

[7] Sonatrach, *Official Website and Public Reports*, Available at:https://www.sonatrach.com , last access : 06/22/2025.

[8] Sonatrach Group, *Company Overview and Performance Report*, Available at: https://sonatrach.com/, Accessed: October 2025.

[9] Sonatrach Group – Division Exploration Center (Boumerdès), *History of the Sonatrach Division Exploration Center – Boumerdès*, Boumerdès, Algeria, [2012].

[10] Marco Brambilla, Jordi Cabot, and Manuel Wimmer, *Model-Driven Software Engineering in Practice*,Morgan / Claypool Publishers, 2017. Available at: https://doi.org/10.2200/S00751ED2V01Y201701SWE004.

[11] Grady Booch, James Rumbaugh, and Ivar Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley Professional, 2nd Edition, 2005. Available at: https://doi.org/10.5555/1076321.

[12] Martin Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, Addison-Wesley Professional, 3rd Edition, 2004. Available at: https://martinfowler.com/books/uml.html.

[13] Grady Booch, James Rumbaugh, and Ivar Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley Professional, 2nd Edition, 2005. Available at: https://doi.org/10.5555/1076321.

[14] Alan A. A. Donovan and Brian W. Kernighan, *The Go Programming Language*, Addison-Wesley Professional, 2015. Available at: https://golang.org/doc/.

[15] Google LLC, *Flutter Framework Documentation*, Available at: https://flutter.dev/docs.

[16] Google, *Flutter Documentation*, Available at: https://flutter.dev/docs

[17] MongoDB Inc., *MongoDB Documentation*, Available at: https://www.mongodb.com/docs/.

[18] Elmasri, Ramez and Navathe, Shamkant, *Fundamentals of Database Systems*, 7th Edition, Pearson, 2015.

[19] Google, *Dart Documentation*, Available at: https://dart.dev/docs

[20] Alapati, S. R., *Expert Oracle Database 11g Administration*, Apress, 2009. Available at: https://link.springer.com/book/10.1007/978-1-4302-1016-0

[21] Karl Fogel, *Producing Open Source Software: How to Run a Successful Free Software Project*, O'Reilly Media, 2021.