المدرسة المغربية لعلوم المهندس

**The Moroccan School of Engineering Sciences**

**l’École Marocaine des Sciences de l'Ingénieur**

|  |
| --- |
| **Internship project:**  **The Creation and Implementation of a Web-based Platform for Procurement, Quality testing and Automation of Raw Material Purchases** |

#### 

|  |  |
| --- | --- |
| Submitted by: | **RABII Yahya** |
| Host Company: | **FinaTech Group** |
| **Guidance:** | **Mr. El BALMANY Chawki** |
| **Supervisor:** | **Mr. KHADDARI Yassin** |

**Acknowledgments**

**T**his endeavour could not have come to fruition without the assistance and support of several individuals whom I sincerely wish to extend my gratitude to and dedicate this work to.

**M**r. Chawki El Balmany, our mentor, who continuously provided me with his guidance and spared no effort in contributing to the success of this work.

**I**n this regard, I would also like to extend special thanks to Mrs. Hajar, who supported and advised me, and above all, shared with me her expertise in the field.

**M**r. KHADDARI Yassin, for the directions he provided during the internship period.

**T**he personnel of the company "FinaTech Group" for the enriching and engaging experience they provided me with.

**A**ll of our professors, and particularly the members of the jury who agreed to evaluate my work.

**O**ur Institute, which provided us with the opportunity to acquire professional training.



Yahya rabii

**Table of Contents**

[General Introduction 9](#_bookmark0)

[Chapter 1: General Context of the Internship 10](#_bookmark0)

1. [Company Overview 11](#_bookmark0) 
   1. [General introduction 11](#_bookmark1)
   2. [FinaTech Group](file:///C:\Users\yahya\AppData\Roaming\Microsoft\Word\yahya_RAPPORT.docx) 12
   3. FinanceCom 12
   4. [Technical sheet of FINATECH Group](#_Fiche_technique_de) 13
   5. [Activities of FinaTech Group 14](#_TOC_250005)
2. General work framework 16
3. [Project Presentation 16](#_bookmark2) 
   1. [Project Overview 16](#_bookmark3)
   2. [Objectives to Achieve 16](#_bookmark4)
   3. [Required Work 16](#_bookmark5)
4. Project Planning 17

[Chapter 2: Requirements Analysis and Specifications 18](#_bookmark0)

[Introduction 19](#_bookmark6)

1. [Preliminary Study 19](#_bookmark7) 
   1. [Existing System’s Study 19](#_bookmark8) 
      1. [Existing System’s Description 19](#_bookmark9)
      2. [Existing System’s Critique 19](#_bookmark10)
   2. [Proposed Solutions 20](#_bookmark11)
2. [Requirements Specification 20](#_bookmark12) 
   1. [Functional Requirements 20](#_bookmark13)
   2. [Non-functional Requirements 21](#_TOC_250004) 
      1. [Ergonomic constraints 21](#_bookmark14)
      2. [Technical constraints 21](#_bookmark14)

[Conclusion 21](#_bookmark6)

[Chapter 3: Conceptual Study 22](#_bookmark0)

[Introduction 22](#_bookmark15)

1. [Project Development Life Cycle 22](#_bookmark16)
2. [UML Language 25](#_TOC_250003) 
   1. [Introduction to UML 25](#_bookmark17)
   2. [Importance of modelling 25](#_bookmark18)
   3. [advantages of UML 25](#_bookmark18)
3. [Design with UML 26](#_TOC_250002) 
   1. [Modelling Tool 26](#_bookmark19)
   2. [Modeling with Use Case Diagrams 26](#_bookmark20)
   3. [Modeling with sequence diagrams 34](#_bookmark20)
   4. [Modeling with Class Diagram 38](#_bookmark20)
   5. Code [MAP 39](#_bookmark20)

[Conclusion 39](#_bookmark15)

[Chapter 4: Implementation 40](#_bookmark0)

[Introduction 40](#_bookmark15)

1. [Working environment 40](#_bookmark21) 
   1. [Hardware environment 40](#_bookmark22)
   2. [Software environment 42](#_TOC_250001)
2. [Development 42](#_TOC_250000)
3. [Operating System 42](#_bookmark22)
4. [Development Tools and Technologies 42](#_bookmark22)
5. [Integrated Development Environment 44](#_bookmark22)
6. [Database administration Tool 45](#_bookmark22)
7. [Containerization Tool](#_bookmark22)  [46](#_bookmark22)
8. [Version Control Tool](#_bookmark22)  [46](#_bookmark22)
9. [Programming language 47](#_bookmark22)
10. Graphical interfaces 48
    1. [Interfaces 48](#_bookmark22)
11. Authentication interface 48
12. Registration interface 49
13. Home interface 50
    1. [Documentation 52](#_bookmark22)
    2. [GitHub-Action 52](#_bookmark22)
    3. [Deployment 52](#_bookmark22)

[General conclusion 56](#_bookmark0)

**List of figures**

Figure 1: The company Logo 12

Figure 2: Membership of the FINANCECOM group 12

Figure 3: Agile-Scrum Model 24

Figure 4: The use case diagram (Administrator) 27

Figure 5: The use case diagram (Supplier) 28

Figure 6: The use case diagram (Purchasing Department Manager) 29

Figure 7: The use case diagram (Purchasing Department Receptionist) 30

Figure 8: The use case diagram (Finance Department Manager) 31

Figure 9: The use case diagram (Quality testing Department Manager) 32

Figure 10: The use case diagram (Project Manager) 33

Figure 11: The Sequence diagram (Supplier’s Registration) 34

Figure 12: The Sequence diagram (Supplier’s Validation) 35

Figure 13: The Sequence diagram (Supplier’s Authentication) 36

Figure 14: The Sequence diagram (Full Diagram) 37

Figure 15: The Class diagram (Full Diagram) 38

Figure 16: The Code Map 39

Figure 17: ASP Logo 42

Figure 18: ASP.Net Logo 43

Figure 19: ASP.Net Core MVC Logo 44

Figure 20: VS Code Logo 44

Figure 21: SQL server Express Logo 45

Figure 22: SQL server Logo 45

Figure 23: Docker Logo 46

Figure 24: Git Logo 46

Figure 25: GitHub Logo 46

Figure 26: C# Logo 47

Figure 27: Authentication Interface 48

Figure 28: Registration Interface 49

Figure 29: Administrator Home Interface 50

Figure 30: User Home Interface 51

Figure 31: Swagger API Logo 53

Figure 32: CI/CD Pipeline to azure 53

Figure 33: Microsoft Azure SQL database Logo 53

Figure 34: Microsoft Azure Portal Logo 53

## List of Acronyms

* **UML:** Unified Modeling Language
* **OS:** operating system
* **RAM**: Random Access Memory
* **ASP**: Active Server Pages
* **MVC**: Model View Controller
* **SQL:** Structured Query Language
* **SSMSE:** SQL Server Management Studio Express
* **Git:** Global Information Tracker
* **GUI**: Graphical User Interface

**List of tables**

**Table 1**: Technical Sheet of FINATECH Group 13

**General Introduction**

Since ancient times, humanity has never ceased its relentless quest for various means to transmit its messages, thereby establishing vital communication pathways. Across successive eras, this pursuit has mobilized both ingenuity and physical energy to uncover adapted and efficient methods of communication.

It was until the 20th century, a revolution in the field of telecommunications emerged, fuelled by rapid technological advancements and the widespread dissemination of the Internet. This progress significantly hastened the transmission of information between senders and receivers. Nowadays, information flows extend extensively, diversify, adopt bidirectional directions, and has become easily accessible.

In our era, within a thriving web application market, driven by the rise of computing tablets, the number of downloaded mobile applications experiences exponential growth year after year.

Amidst this vibrant context, an increasing number of companies operating in the telecommunications sector are considering substantial investments in this domain. Among these pioneers stands FinaTech Group.

In the scope of my internship project, I have been entrusted with the task of designing an Android application aimed at connecting FinaTech Group's purchasing service with its various suppliers. As a result, I have structured my report around four chapters:

In the first chapter, "General Framework," I have elucidated the context of my project.

In the second chapter, "Analysis and Requirements Specification," I have identified the future system's stakeholders, both functional and non-functional requirements.

In the third chapter, "Conceptual Study," I have modelled a design with use case diagrams and a comprehensive sequence diagram.

In the final chapter, "Implementation," I present the hardware and software environment, along with the key interfaces of the developed application.

**1st Chapter**

# General Context

# Of

# The Internship

**Chapter 1: General Context of the Internship**

### **Introduction**

In this chapter, I place my work within its broader context. First and foremost, I have introduced the internship environment through a brief presentation of the hosting organization. Subsequently, I have outlined the overall framework of this project.

### **Company Overview**

In this section, I introduce the hosting organization that proposed this subject.

### **General introduction**

**FinaTech** Group is a prominent technological integrator that was established in 2007, stemming from O CAPITAL GROUP's desire to equip itself with a technological arm capable of deploying high-value solutions to operators and strategic stakeholders in Morocco and Africa.

**FinaTech** Group is engaged in a wide range of activities through its in-house business units and its various subsidiaries.

At the heart of **FinaTech** Group's approach lies the consideration of the challenges and objectives of its partner clients. This consideration allows the company to design solutions and infrastructures that cater to the genuine needs of its partner clients, while also ensuring long-term sustainability and environmental responsibility.

**FinaTech** Group leverages its expertise across various domains through a fully integrated framework. This framework spans from collaborative ideation with partner clients for solution design to operational deployment and proactive as well as reactive maintenance.

### **FinaTech Group Logo**

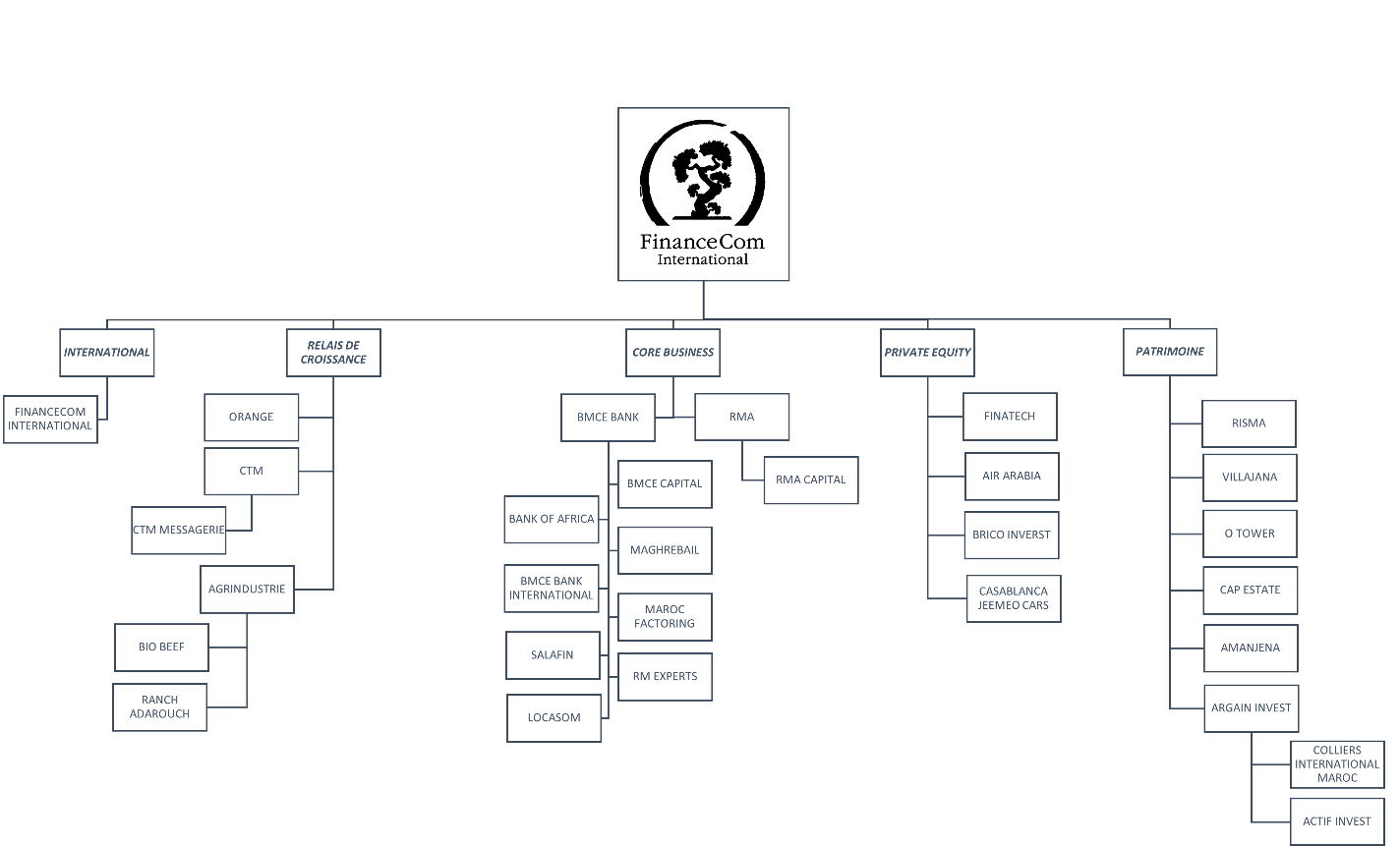


*Figure 1: The Company Logo*

1. **FINANCECOM**

Since its establishment in 2007, FinaTech Group has been an integral part of **O Capital Group** (formerly known as **Financecom**). **FinaTech** Group serves as the technological arm of the Group.

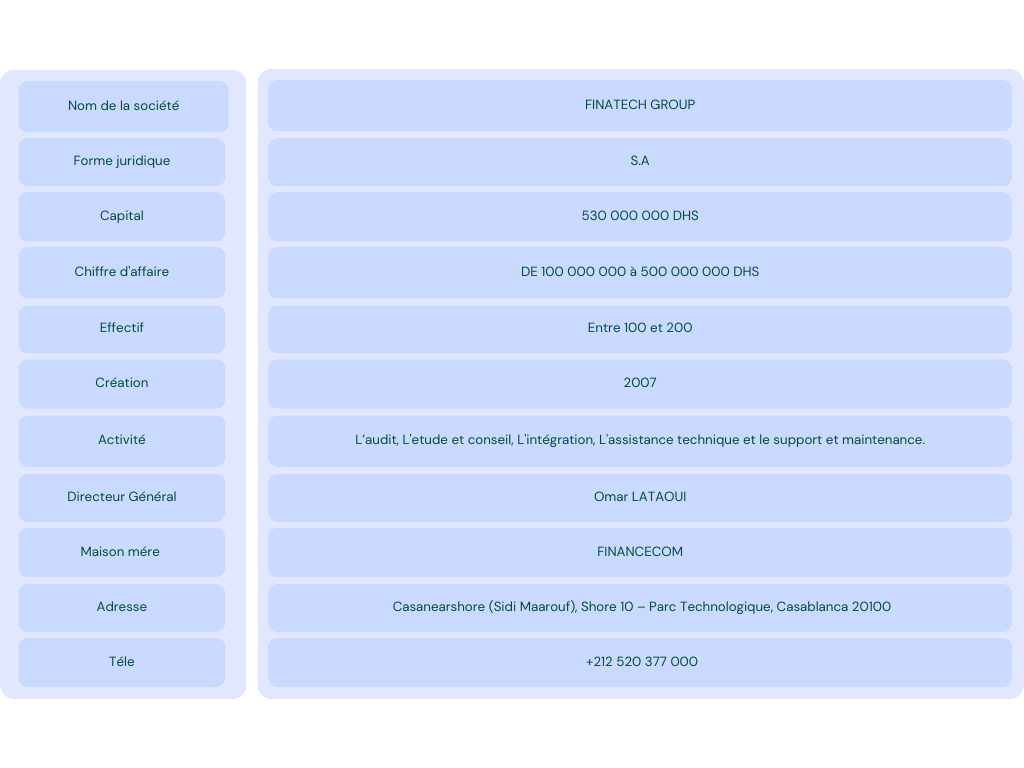
**Financecom** Group stands as one of the largest economic and financial consortia in the Kingdom of Morocco. The Group has a presence in pivotal and high-potential sectors such as banking, insurance, telecommunications, real estate, hospitality, transportation, technology, and more, both in Morocco and across approximately 23 African countries.



*Figure 2:* *Membership of the FINANCECOM Group*

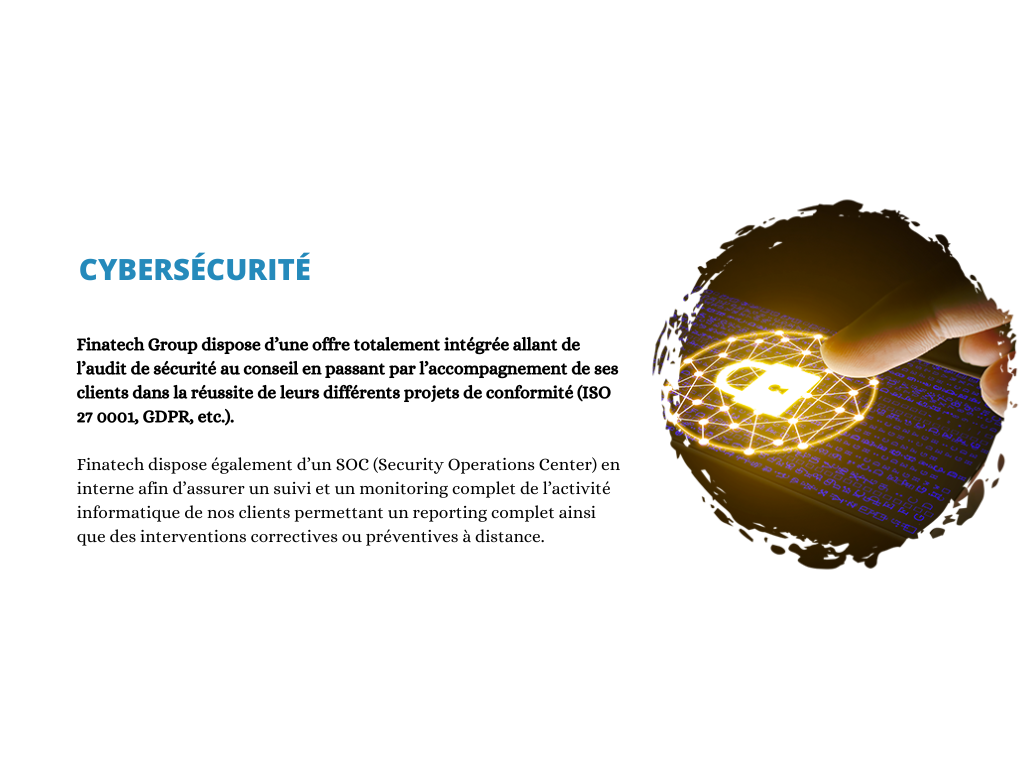
### 

### **Technical Sheet of FINATECH Group**

****

*Tableau 1: Technical Sheet of FINATECH Group*

### **Activities of FinaTech Group:**











1. **General Work Framework:**

For this internship, the objective is to design and develop a web platform aimed at enhancing communication with suppliers and automating raw material purchasing. This solution aims to simplify the quality control processes, thereby contributing to the overall profitability of the company.

### **Project Presentation:**

### **Project Overview:**

This work is carried out within the scope of an internship project. The platform is designed to be compatible with all devices and web browsers. It provides employees of the company with a simpler way to communicate with suppliers and test the quality of purchased products, with the goal of enhancing workplace efficiency.

### **Objectives to Achieve:**

* + - Facilitating and organizing employees' work.
    - Ensuring proper customer service.
    - Saving time and achieving customer satisfaction.
    - providing a better work experience through seamless navigation, as a web application is often more ergonomic than using paper

### **Required Work:**

My responsibilities consist of:

* + - * Contributing to the development of a dynamic web application.
      * Establishing an optimized connection between the platform and the database.
      * Optimizing performance for a seamless experience across various devices.
      * Integrating synchronization functionalities.
      * Providing comprehensive documentation to simplify the application's usage.
      * Conducting regular tests to identify and resolve potential compatibility issues.

### **Project Planning:**

**Project start date: 20/07/2023**

**1st Sprint:**

**Main objective: Define project specifications and requirements, and validate the initial design.**

1. Sprint planning: 20/07/2023 - 21/07/2023
2. Design and prototyping: 22/07/2023 - 05/08/2023
3. Internal review and improvement: 06/08/2023 - 10/08/2023
4. Sprint review and retrospective: 11/08/2023

**2nd Sprint:**

**Main objective: Complete and validate the required prototypes, and develop the application.**

1. Sprint planning: 12/08/2023 - 13/08/2023
2. Development and prototyping: 14/08/2023 - 05/09/2023
3. Internal review and improvement: 06/09/2023 - 10/09/2023
4. Sprint review and retrospective: 11/09/2023

**3rd Sprint:**

**Main objective: Test the application according to the established specifications.**

1. Sprint planning: 12/09/2023 - 13/09/2023
2. Development and testing: 14/09/2023 - 18/09/2023
3. User acceptance testing (UAT): 19/09/2023 - 23/09/2023
4. Sprint review and retrospective: 24/09/2023

**Deployment (Release to Production):**

The final deployment of the application is scheduled for **25**/**09/2023** marking the official launch of the product for end users.

### **Conclusion:**

In this chapter, I have placed this project within its broader context. I began with a general description of the company "FinaTech". I presented its Logo and its activities. Following that, I introduced the overall framework of this project, starting with a brief overview, the objective to achieve, and the required tasks. I concluded by providing details about the development environment and the timeline



**Requirements Analysis and Specifications 2nd Chapter**

**Chapter 2: Requirements Analysis and Specifications**

### **Introduction:**

In this second chapter, I will start by discussing the existing system at "FinaTech." Through our study we have identified some communication challenges, with suppliers that directly affect efficiency and customer satisfaction. This analysis forms the basis, for introducing a solution aimed at improving service quality.

### **Preliminary Study:**

To ensure the achievement of objectives, it is essential to start with a clear and straightforward view of the various expected needs. Hence, an examination of the existing system will enable me to justify and formulate a solution.

### **Existing System’s Study:**

### **Existing System’s Description:**

The issue lies in the absence of an efficient means of communication with suppliers and the warehouse control, which has resulted in delays that adversely affect customer satisfaction. Presently, communication takes place through traditional methods such as email or direct telephone contact with the sales department of each supplier. After an order is placed (specifying the product and quantity), a notification is received containing a form (quotation) that must be completed out and signed by the employee. This form includes details such as the product type and desired quantity upon order confirmation.

### **Existing System’s Critique:**

Inevitably, it is evident that promptly identifying the issue at hand and devising effective solutions are necessary to ensure the company's proper service towards its clients

### **Proposed Solutions:**

To fix the identified issues and ensure the professionalism of services offered by "FinaTech" my aim is to design and develop a web platform aimed at automating and computerizing communication between the company's employees and suppliers, while ensuring efficient warehouse management. This initiative aims to ensure the speed and security of service exchanges.

### **Requirements Specification:**

### **Functional Requirements:**

The future platform offers the following functionalities to different users:

For Administrators:

* User Management: Addition, modification, deletion, activation, and deactivation.
* Supplier Management: Activation confirmation.
* Access to product history.
* Access to supplier history.
* Addition, deletion, and modification of claims.

For the Project Managers:

* Project Management: Addition, modification, deletion.
* Access to product history.
* Addition of products to projects.

For the Purchasing Department Managers:

* Creation of purchase requests.
* Validation of sales offers: Access to history, validation.
* Creation of purchase quotations.

For the Purchasing Department Receptionists:

* Reporting the arrival of purchased products.
* Generating receipt reports.

For the Finance Department Managers:

* Creation of bills.

For the Quality Control Department Managers:

* Creation of quality test reports.
* Validation of Payments.

### **Non-Functional Requirements:**

The non-functional requirements represent the rules to adhere to in order to ensure the quality of the future system to be completed and its proper functioning, they can be summarized as follows:

### **Ergonomic constraints:**

The application aims to achieve several crucial imperatives:

Ensure a smooth learning curve, enabling users to quickly familiarize themselves with the content and functionalities.

Prioritize simplicity and clarity in the interface design to ensure an intuitive and easily understandable user experience.

Organize interactive elements such as buttons, tables, and forms logically and intuitively to enhance overall ergonomics.

### **Technical constraints:**

* **Hosting:** Microsoft Azure Cloud for availability and scalability.
  + - * **Access (Ordinary Users):** Reserved for authorized employees with secure authentication.
      * **Access (Administrator):** Grant full control to the administrator.
      * **Language:** C# with ASP.NET for power and stability.
      * **Database:** SQL Server for reliable data storage.
      * **Design Tool:** Enterprise Architect for well-structured architecture.
      * **Version Control Tool:** Git / GitHub for tracking changes and collaboration.
      * **Containerization Tool:** Docker for easy deployment and dependency management.

**Conclusion:**

In this chapter, I conducted a study of the existing system's operations to identify an acceptable solution. I described both the functional and non-functional requirements of the new system.



**3rd Chapter**

**Conceptual Study**

**Chapter 3: Conceptual Study**

### **Introduction:**

“Anticipate before acting, establish plans before building, conceptualize upstream and develop downstream” is the essential approach to follow during the development of an application, as well as for the success of any project.

Indeed, the design phase of an information system holds paramount importance, exerting a direct influence on the overall quality and reliability of the application.

In this chapter, I will introduce the selected project's life-cycle model, thereby providing a methodological framework. Subsequently, I will delve into the design phase in detail, during which I will precisely present the overall architecture of the application. Lastly, I will highlight various diagrams, namely the use case, the sequence diagrams, and the class diagram as well, which will complement this design stage.

### **Project Development Life Cycle:**

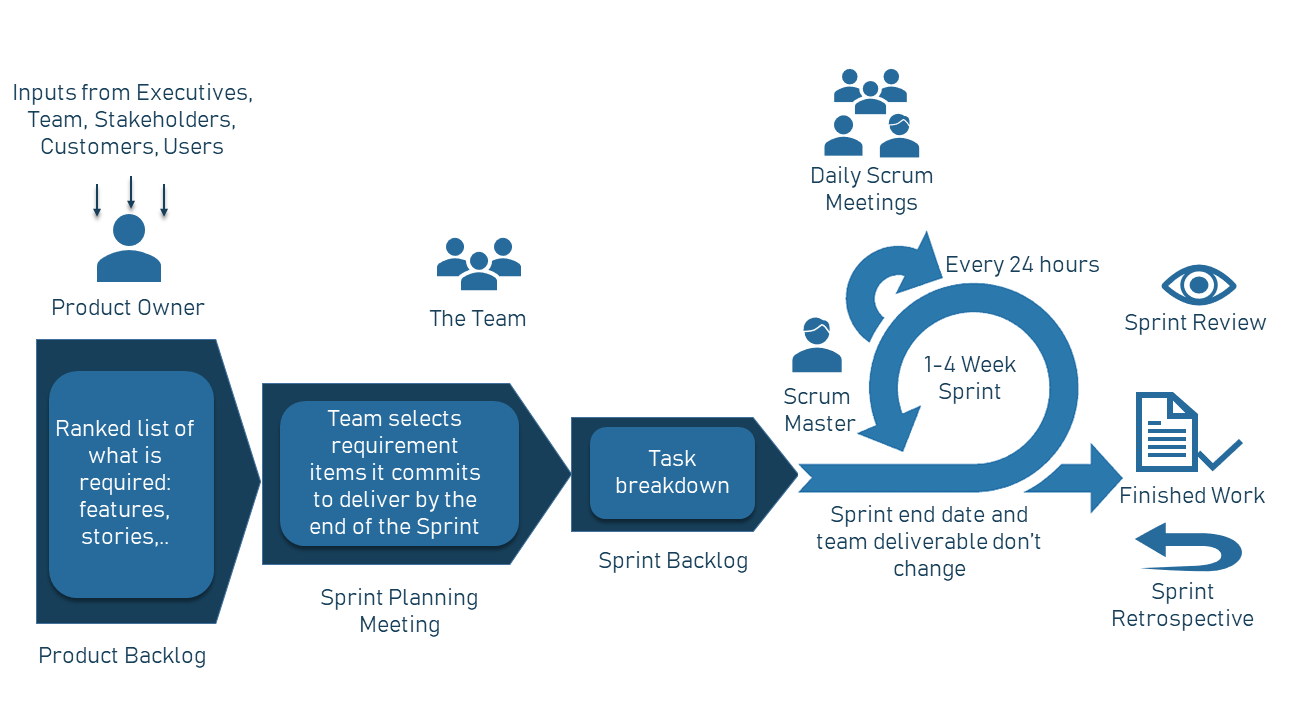
The software life cycle, often referred to as the "SLC" encompasses all stages of application development, from conception to eventual release. This structuring aims to establish intermediate steps to confirm software development, ensuring the software's alignment with expressed needs and verifying the development process by maintaining consistency in adopted methodologies. To achieve optimal realization, I introduced the agile methodology, specifically the Scrum methodology, an approach that emerged in the 1990s.

Within this approach, each stage is designed to be flexible and adaptive. Instead of rigidly segmented phases, Scrum operates through iterations known as "sprints." Each sprint, typically of short duration, aims to produce a functional version of the software. The Scrum approach promotes close collaboration among team members, encouraging open and constant communication.

At the core of Scrum lies the "Product Backlog," a dynamic list of items to be accomplished that evolves based on user feedback and changing priorities. At the start of each sprint, the team selects backlog items to address, creating the "Sprint Backlog" that guides sprint efforts. Key Scrum roles, such as the "Scrum Master" and "Product Owner," work synergistically to ensure the success of each sprint and the achievement of objectives.

At the end of each sprint, a "Sprint Review" is held to demonstrate completed work to stakeholders. This is followed by a "Sprint Retrospective" that allows the team to analyze the past sprint and identify improvements for future iterations.

The Scrum approach within the agile framework offers exceptional flexibility to adapt to project evolution and changing requirements. It promotes transparency, team engagement, and continuous delivery of features, making it a relevant methodology for complex and dynamic software development environments



*Figure 3: Agile - Scrum Model*

### **UML Language:**

### **Introduction to UML:**

**UML** (Unified Modeling Language) is a graphical language for modeling data and processes. It's a form of object-oriented Modeling used in software engineering. UML serves various objectives which makes it an accurate communication tool:

* Understanding and describing requirements.
* Specifying a system.
* Establishing a software architecture.

### **Importance of modeling:**

The use of conceptual modeling in the development of an information system allows for a more adequate consideration of application needs and provides an abstract representation of certain aspects of physical and human systems.

### **advantages of UML:**

* UML is a formal and standardized language it.
* gains precision.
* motivates the use of tools.
* enhances stability and consistency.

UML standardizes object concepts, and its graphical representation enable the expression of an object-oriented solution, simplifying the comparison and assessment of solutions. UML frames object analysis not only represents object concepts but implies an analytical approach that iteratively reproduces an object solution through diagrams, which support abstraction.

UML diagrams are a graphical representation, and different views correspond to diagrams that are distributed according to their static or dynamic aspects:

* ***Static aspects:*** 
  + Use case
  + Classes
  + Components
  + Valuables
  + Deployment

* ***Dynamic aspects:*** 
  + Sequences
  + Activity
  + Transitional State
  + Collaboration
* These diagrams are not necessarily all produced during modelling. The most commonly used ones include class diagrams, use case diagrams, and sequence diagrams.

### **Concepting with UML:**

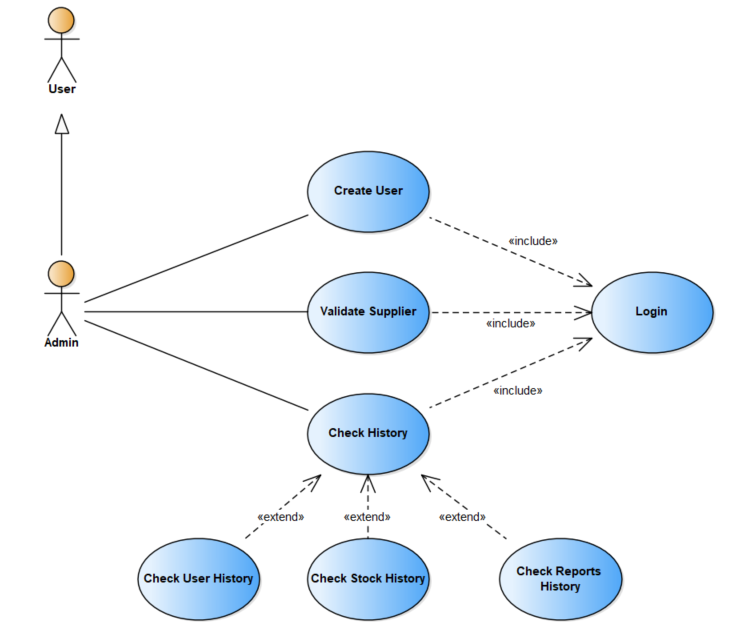
### **Modelling Tool:**

**SparxEA**, formerly known as "Sparx Enterprise Architect," is a visual modeling and design tool used in software development and database management. It is developed by Sparx Systems and primarily focuses on assisting businesses in designing, modeling, and managing their databases, business processes, and software architecture.

### **Modeling with Use Case Diagrams:**

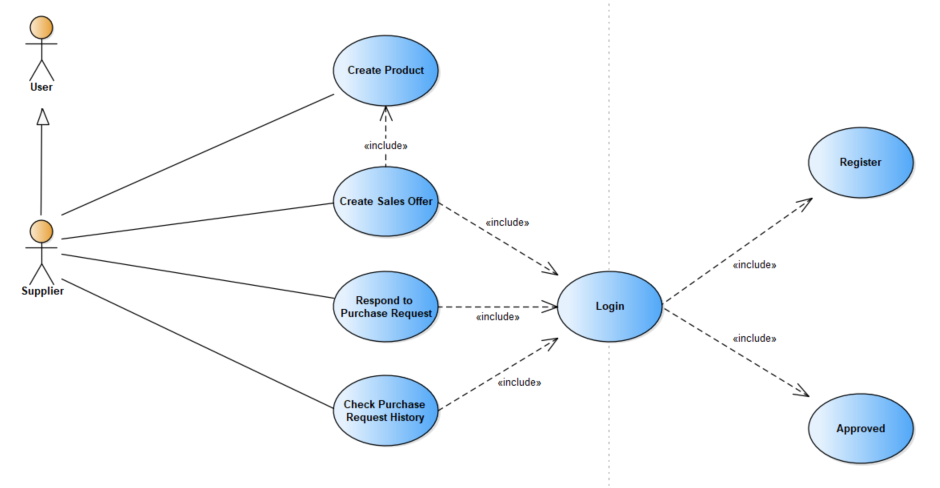
The use case diagram helps determine the potential interactions between the system and the actors, which means identifying all the functionalities that the system should provide. It also helps define the boundaries of the system.

* Each action performed by actors is represented by a use case.
* Each use case signifies a functionality offered to it to produce the expected result.
* The use case diagram describes the interaction between the system and the actor, identifying user needs and outlining all the system's actions for the actor.
* **Admin**



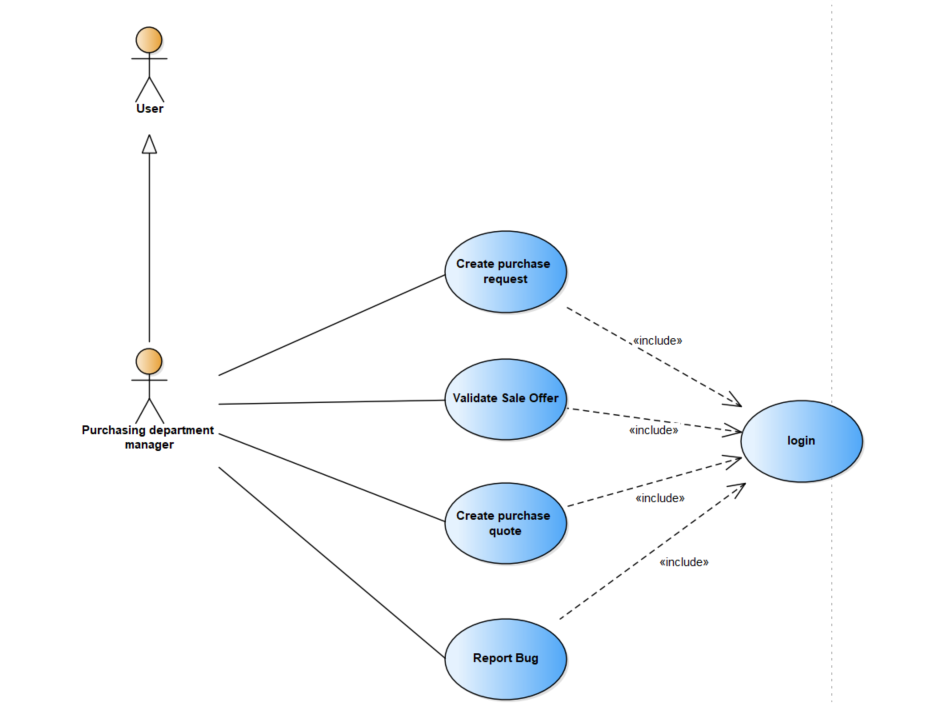
*Figure 4: The Use Case Diagram (Administrator)*

* **Supplier**



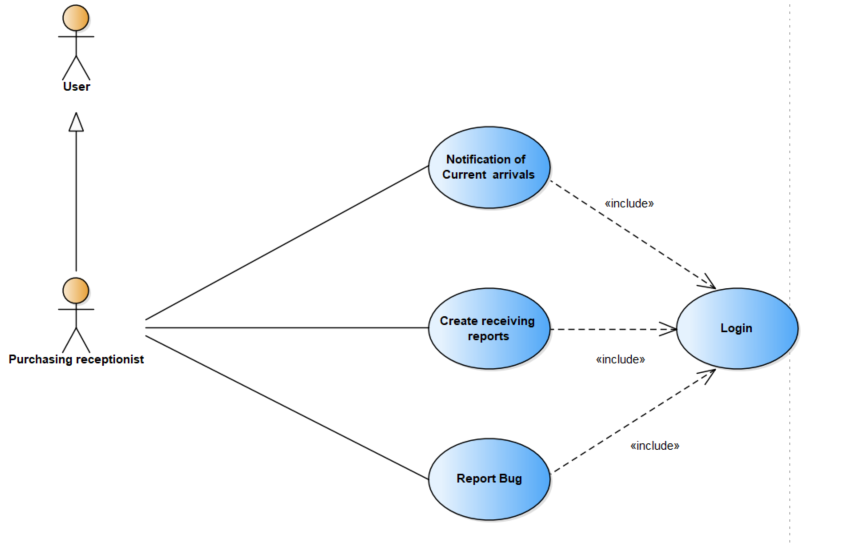
*Figure 5: The Use Case Diagram (Supplier)*

* **User: (****Purchasing Department Manager)**



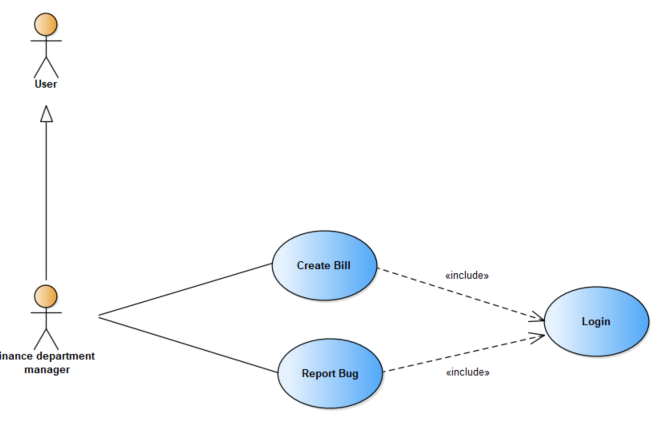
*Figure 6: The Use Case Diagram (Purchasing Department Manager)*

* **User:** **(****Purchasing Department** **Receptionist)**



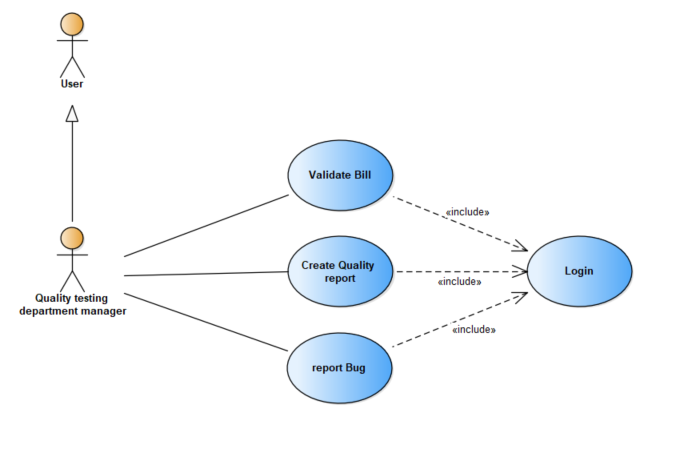
*Figure 7: The Use Case Diagram (Purchasing Department Receptionist)*

* **User: (Finance Department** **Manager)**



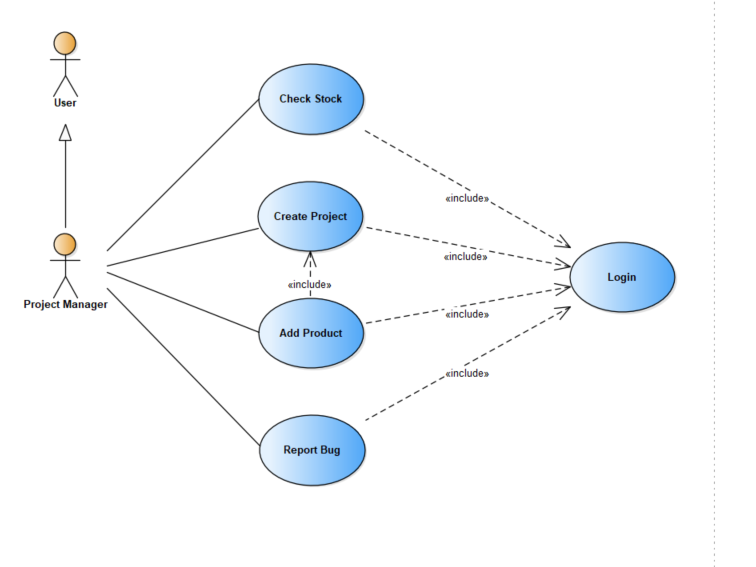
*Figure 8: The Use Case Diagram (Finance Department Manager)*

* **User:** **(Quality testing Department** **Manager)**



*Figure 9: The Use Case Diagram (Quality testing Department Manager)*

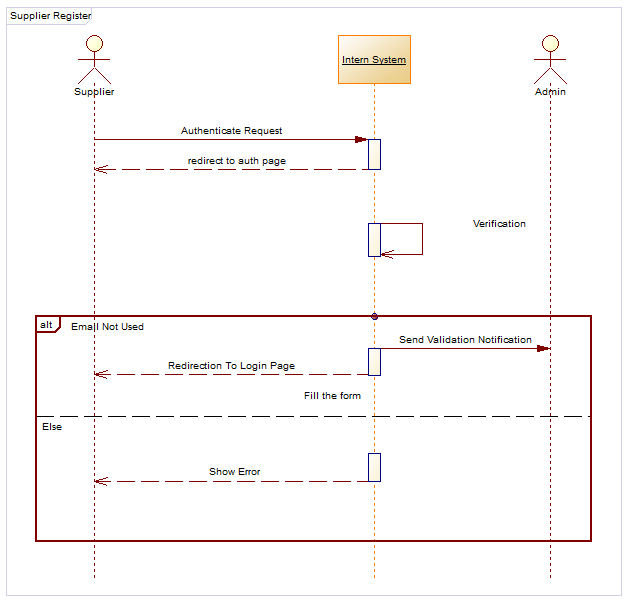
* **User: (Project** **Manager)**



*Figure 10: The Use Case Diagram (Project Manager)*

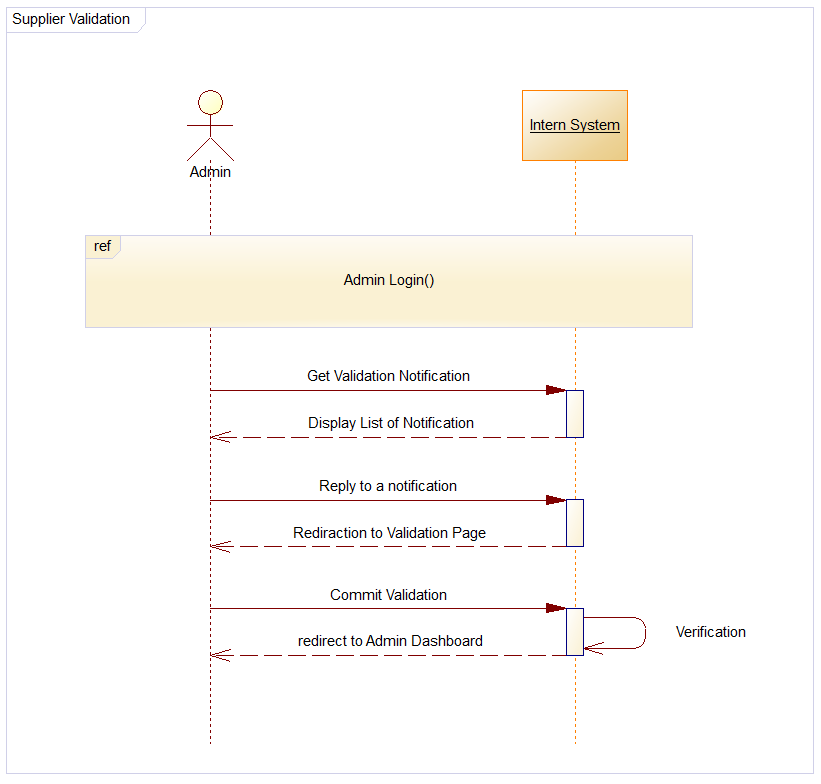
### **Modeling with sequence diagrams:**

* **Supplier’s Registration:**



*Figure 11: The Sequence Diagram (Supplier’s Registration)*

* **Supplier’s Validation:**



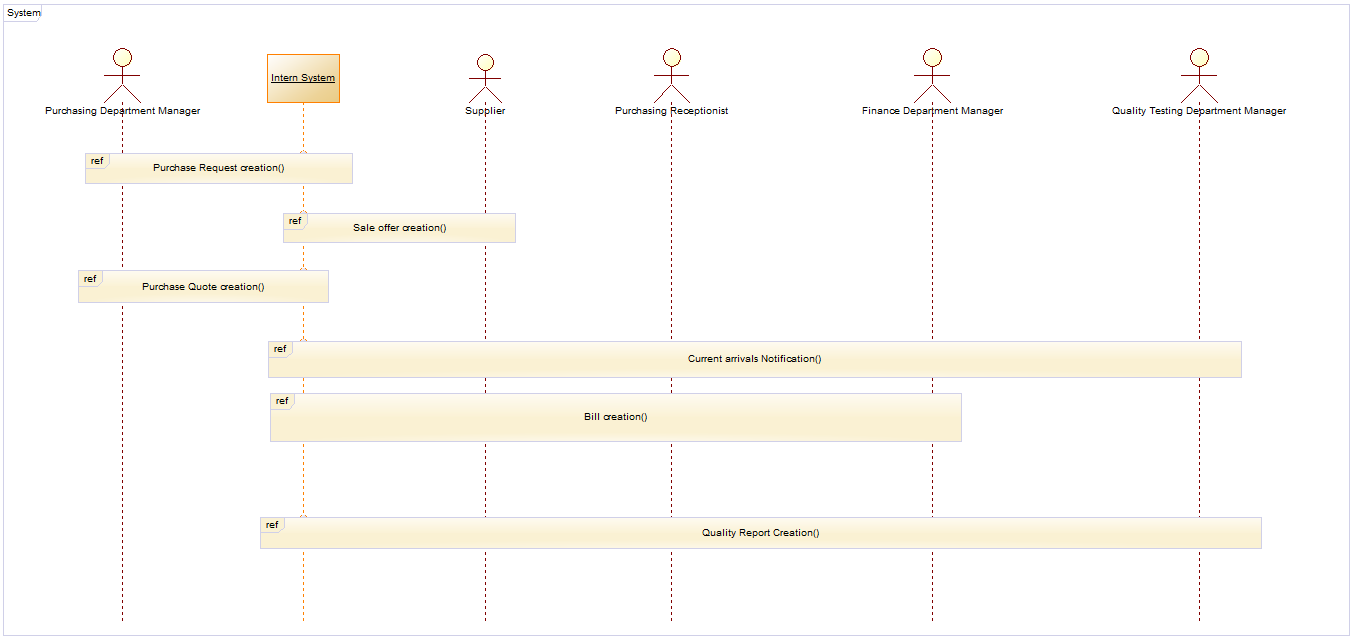
*Figure 12: The Sequence Diagram (**Supplier’s Validation)*

* **Supplier’s Authentication**



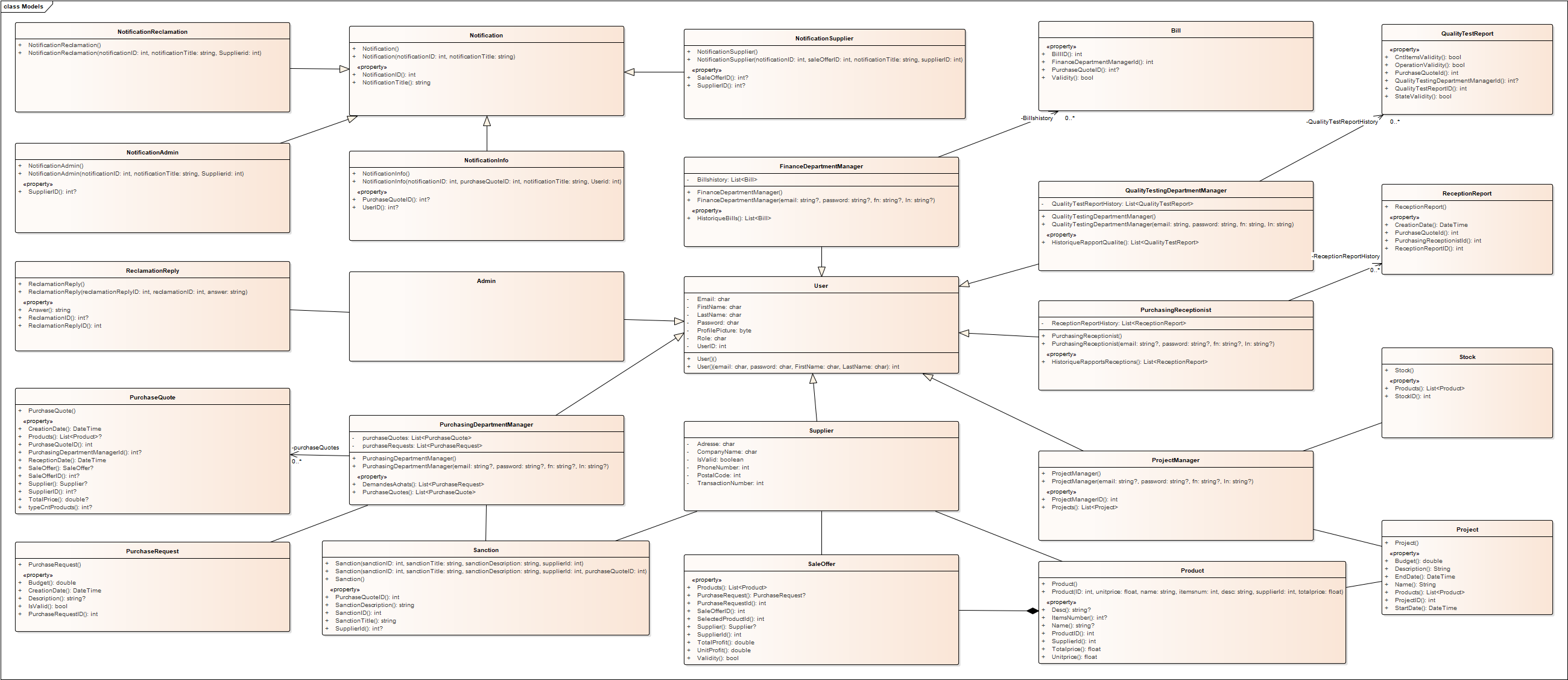
*Figure 13: The Sequence Diagram (Supplier’s Authentication)*

* **Full diagram**



*Figure 14: The Sequence Diagram (Full Diagram)*

### **Modeling with Class Diagram:**

**

### 

*Figure 15: The Class Diagram (Full Diagram)*

### **Code MAP:**

### 

*Figure 16: The Code Map*

### Conclusion:

This chapter was devoted to UML modeling of the system. I presented different diagrams of the use case diagram, and sequences in order to specify in detail the functional, dynamic and static aspects of the system. In the next chapter, I will present the development and testing of this system.



**4th Chapter**

# The

# Implementation

Chapter 4: The Implementation

### **Introduction:**

I have now reached the final phase of my report. This last step is of paramount importance as it materializes all the theories presented earlier. To begin, I will present the software production environment. Then, I will highlight several key interfaces of my solution, as well as some scripts that I have identified as particularly crucial.

### **Working environment:**

Within this section, I will emphasize the various hardware and software tools essential to the development process of my application.

### **Hardware environment:**

I will mention the specifications of the computers on which I developed the application, as they can provide insight into the working conditions.

The application was developed on two laptops with the following main specifications:

* ***1st*** ***Laptop***
* Name: HP
* Processor: Intel® Core™ i9-10210U (1.6 GHz base frequency, up to 4.2 GHz)
* Installed Memory (RAM): 12.00GB (7.89GB usable)
* System Type: 64-bit Operating System
* ***2nd Laptop***
* Name: Asus ROG Strix G15
* Processor: AMD Ryzen™ 9 6900HX Mobile Processor (8-core/16-thread, 20MB cache)
* Installed Memory (RAM): 16.00GB DDR4-4800 SO-DIMM
* System Type: 64-bit Operating System

### **Software environment:**

Throughout this section, I will list the various tools used for the study and implementation of our application during this project.

### **Development:**

* 1. **Operating System:**

I used the following operating systems:

* 1st Laptop: Microsoft Windows 11 Professional
* 2nd Laptop: Microsoft Windows 10 Professional

### **Development Tools and Technologies:**

* + - **ASP (Active Server Pages):**

ASP, short for Active Server Pages, is a technology pioneered by Microsoft to design dynamic web pages by utilizing server-side executed scripts. Using ASP, developers were able to embed script code (usually in VBScript or JScript) within HTML pages to generate dynamic content. However, ASP's limitations in terms of scalability, maintenance, and separation of concerns became evident.



*Figure 17: ASP Logo*

* + - **ASP.NET:**

ASP.NET represents the evolution of ASP, featuring a more sophisticated and comprehensive web development framework. This advancement comes with various improvements, including a more structured approach to building web applications, better separation between code and presentation, and an object-oriented programming model. Among the major features of ASP.NET is the introduction of Web Forms, which offered a drag-and-drop interface for creating web pages, and expanded support for different programming languages, including C# and VB.NET. Additionally, ASP.NET also introduced a powerful event-based programming model, coupled with improved application state management.



*Figure 18: ASP.Net Logo*

* + - **ASP.NET Core MVC:**

ASP.NET Core MVC represents the latest advancement of Microsoft's web development framework, built on the foundation of the ASP.NET Core platform. Here are the key distinctions compared to previous versions:

* + - * Cross-Platform Compatibility: ASP.NET Core MVC is versatile, working seamlessly on Windows, Linux, and macOS. This flexibility makes it adaptable to a variety of hosting environments.
      * Open-Source: ASP.NET Core is open-source and actively developed on the GitHub platform. This encourages community contributions and transparency in the development process.
      * Modular Architecture: ASP.NET Core allows developers to integrate only the necessary components, minimizing overhead and making applications leaner.
      * Performance: ASP.NET Core is designed with performance in mind. It's faster and more efficient compared to previous iterations of ASP.NET.
      * Dependency Injection: ASP.NET Core simplifies dependency injection, making it easier to manage and test dependency relationships within an application.
      * Unified MVC and Web API Framework: ASP.NET Core MVC merges the previously separate MVC and Web API frameworks in ASP.NET. This streamlines development and facilitates creating web applications and APIs within the same framework.

****

*Figure 19: ASP.Net Core MVC Logo*

### **Integrated Development Environment:**

* + - **Visual Studio**

**Visual Studio (VS)** is a powerful integrated development environment (IDE) created by Microsoft for designing a wide range of software applications. With a user-friendly interface, it offers comprehensive tools for programming, debugging, and deploying applications across various platforms. Developers can code in languages such as C#, C++, Python, while benefiting from features like auto-completion, advanced debugging, and integrated version control. VS supports web, desktop, mobile, cloud, and game development, and can be extended through plugins, making it essential for developers seeking a feature-rich and productive environment.



*Figure 20: Visual Studio (VS) Logo*

### **Database administration Tool:**

* + - **SQL Server**

SQL Server, designed by Microsoft, is a crucial relational database management system (RDBMS) for managing, storing, and retrieving structured data. This solution provides a robust and scalable platform, ideal for efficient data processing within businesses and applications. SQL Server offers a range of editions tailored to various needs, catering to small businesses as well as large enterprises.

* + - * For the SQL Server database management system:
      * **SQL Server Express:** SQL Server Express is a lightweight and free edition of Microsoft's SQL Server database management system. Designed for smaller-scale applications, it is suitable for development, testing, or limited production workloads. SQL Server Express includes a subset of the features available in the full version of SQL Server.
      * **SQL Server Management Studio Express (SSMSE):** SQL Server Management Studio Express is a graphical user interface tool used to manage SQL Server Express databases. It allows users to create, modify, and manage databases, as well as perform tasks such as querying, designing database schemas, and managing security.





*Figure 21:SQL Server Express Logo Figure 22: SQL Server Log*

### **Containerization Tool:**

### **Docker:**

### Docker is a game-changing virtualization tool that packages applications and their dependencies in isolated containers, ensuring compatibility and consistent execution across different environments. It accelerates deployment, streamlines management, and allows for flexible application scalability.

### In my project, I used Docker to seamlessly install and manage a SQL Server database within a containerized environment. By encapsulating the SQL Server instance, its configuration, data, and dependencies in a self-contained container, I was able to ensure consistent and reproducible deployment, regardless of the deployment environment. This modular approach enhances efficiency, portability, and deployment consistency, while simplifying application management and maintenance.

*Figure 23: Docker Logo*

### **Version Control Tool:**

### **Git / GitHub:**

### The version control tool Git, paired with the GitHub platform, represents a crucial pillar for collaborative management and traceability in software development. Git provides an advanced mechanism for tracking changes made to source code, allowing developers to work concurrently on separate branches and seamlessly merge their contributions. GitHub acts as a centralized platform to host, share, and collaborate on Git projects. This enables teams to track code evolution, conduct reviews and discussions, and manage issues transparently. The combination of Git and GitHub promotes agile workflows, efficient project management, and detailed change history, making it an essential component of modern development processes.

### 

*Figure 24: Git Logo Figure 25: GitHub Logo*

### **Programming language:**

* + - **C#:**

C# is a modern, versatile, and object-oriented programming language developed by Microsoft. It is widely used to create a diverse range of software applications, from desktop to web and mobile applications. Known for its simplicity and readability, C# incorporates features from various programming languages, making it both intuitive for beginners and powerful for experienced developers. With a strong emphasis on type safety and memory management, C# ensures robust and reliable code. It's an integral part of the .NET framework, offering seamless integration with other Microsoft technologies.



*Figure 26: C# Logo*

### **Graphical interfaces:**

### **Interfaces:**

The graphical user interface (GUI) holds a paramount importance in the design process of an application as it provides a pleasant user experience throughout their navigation. Consequently, this factor has the potential to differentiate between two applications even if they share similar functionalities. Here is a series of screenshots illustrating key entry points of the application:

### **Authentication interface:**



*Figure 27: Authentication Interface*

### **Registration interface:**



*Figure 28: Registration Interface*

The first figure illustrates the authentication interface, where users are prompted to enter their email address and password to access the application's home space. The second figure portrays the registration interface designed for suppliers.

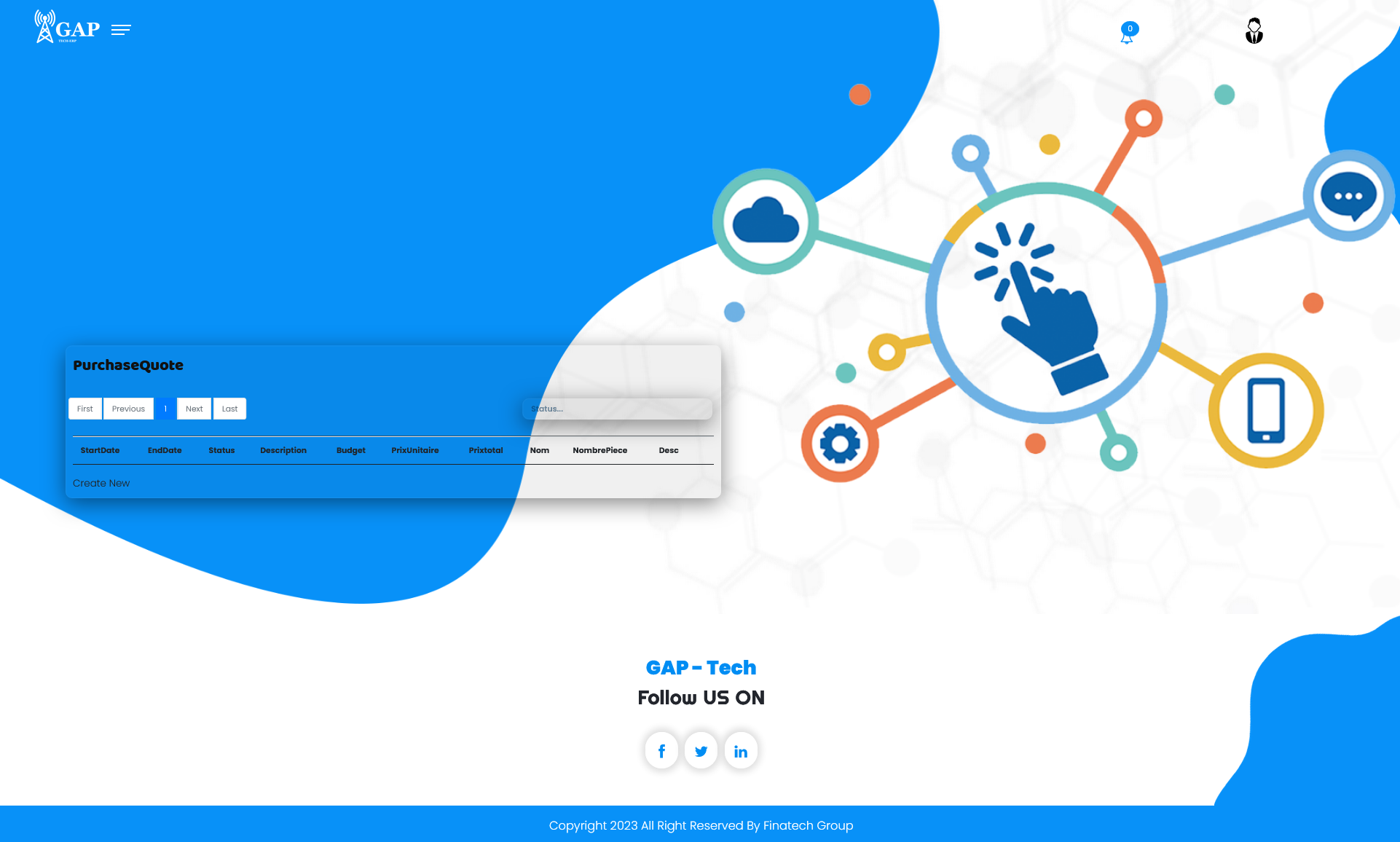
### **Home Interface:**

* **Administrator Home Interface:**



*Figure 29:* *Administrator Home Interface*

* **User Home Interface:**

**

*Figure 30: User Home Interface*

When users enter their username and password, the home interface is displayed. If the user is an administrator, the "Administrator Dashboard" page will open. This page provides an overview of the application's content, stock logs, and interfaces to create different types of users. On the other hand, if the user is not an administrator, the user-specific home interface will be presented. Each type of user enjoys unique content based on their respective responsibilities.

### **Documentation:**

I integrated Swagger into my project to streamline the process of generating routing documentation for my API. It was a straightforward process that involved several key steps. First, I added the Swagger UI and Swagger Core libraries to the project's dependencies, making sure they were properly installed. Then, I configured Swagger by adding annotations directly to my codebase to specify essential API details such as endpoints, request/response models, and models description. Once the configuration was in place, I ran Swagger's documentation generation tool, which automatically scanned the code to produce a comprehensive API specification.

To make this documentation accessible and user-friendly during the development stage, I seamlessly integrated it into our project. developers can easily explore and test the API endpoints through Swagger's interactive Swagger UI, which I embedded right into the application. This integration provides a user-friendly interface for the API interaction and testing.

### 

*Figure 31: Swagger Logo*

### **GitHub-Action:**

Integrating GitHub Actions into the project was a game-changer when it came to deploying our application. The process was remarkably straightforward and immensely beneficial. First, I configured GitHub Actions by creating a workflow file in the repository. This file defined the steps and conditions for our building and deployment process, specifying which branches trigger deployments and which deployment services to use.

Then i defined the deployment environment and connected it to our hosting platform, in our case it's a cloud service provided by Microsoft: Azure.

GitHub Actions allowed me to securely store sensitive deployment secrets and environment variables, ensuring the deployment process was both automated and secure.

Whenever a code change is pushed to the master branch, GitHub Actions automatically triggers the workflow, seamlessly building and deploying our application to the defined environment. This not only saves us significant time but also reduces the likelihood of deployment errors, as the process is consistent and automated.

With GitHub Actions, we've achieved a reliable and efficient deployment pipeline, making it easier than ever to deliver updates and improvements to our application.

### Using Github Actions with Azure App Services for Web Apps | by Kasun Rajapakse | Bits and Pieces

*Figure 32: (CI/CD) pipeline to azure.*

### **Deployment:**

I successfully deployed my ASP.NET Core MVC application on Microsoft Azure, leveraging its cloud infrastructure for seamless hosting and scaling. Using Azure's intuitive deployment tools, I efficiently configured and published my application, ensuring its availability to a broader audience. Furthermore, I integrated the deployment process with Azure's managed database services, utilizing their database solution to store and manage the application's data. This integration not only streamlined the deployment process but also ensured reliable and secure data storage. By harnessing the capabilities of Azure, I achieved an end-to-end solution, with my ASP.NET Core MVC application hosted and supported by Azure's cloud services, and the associated database effortlessly managed through their provided database services.

* Here are some of the steps I followed to successfully deploy:
* Creating Azure Resources: After logging into the Azure portal, I set up the necessary resources to host my application. I configured an App Service to host my ASP.NET Core MVC application and selected the appropriate runtime stack.
* Application Deployment: I used Azure deployment tools such as Azure DevOps or Azure Publish features in Visual Studio to deploy my ASP.NET Core MVC application to the Azure App Service. I ensured proper application configuration, including database connection strings.
* Configuring the Database Service: I explored Azure's managed database services, particularly Azure SQL Database. I created a database instance through the Azure portal, adjusted its settings, and noted the connection details.
* Database Migration: If there were existing data, I considered using database migration tools or scripts to transfer data to the Azure database. I ensured that my ASP.NET Core application was configured to connect to the new Azure database.
* Updating Connection String: I updated the connection strings in my ASP.NET Core MVC application to point to the newly created Azure database. This ensured that my application was communicating with the correct database instance.
* Testing and Verification: I thoroughly tested my application on the Azure platform, verifying its interaction with the Azure database. I validated that data retrieval, storage, and manipulation were functioning as expected.
* Security and Permissions: I took necessary security measures, establishing firewall rules for the Azure database and configuring appropriate access permissions to ensure data security.





*Figure 31: Microsoft Azure SQL Database Logo Figure 32: Microsoft Azure Portal Logo*

Conclusion:

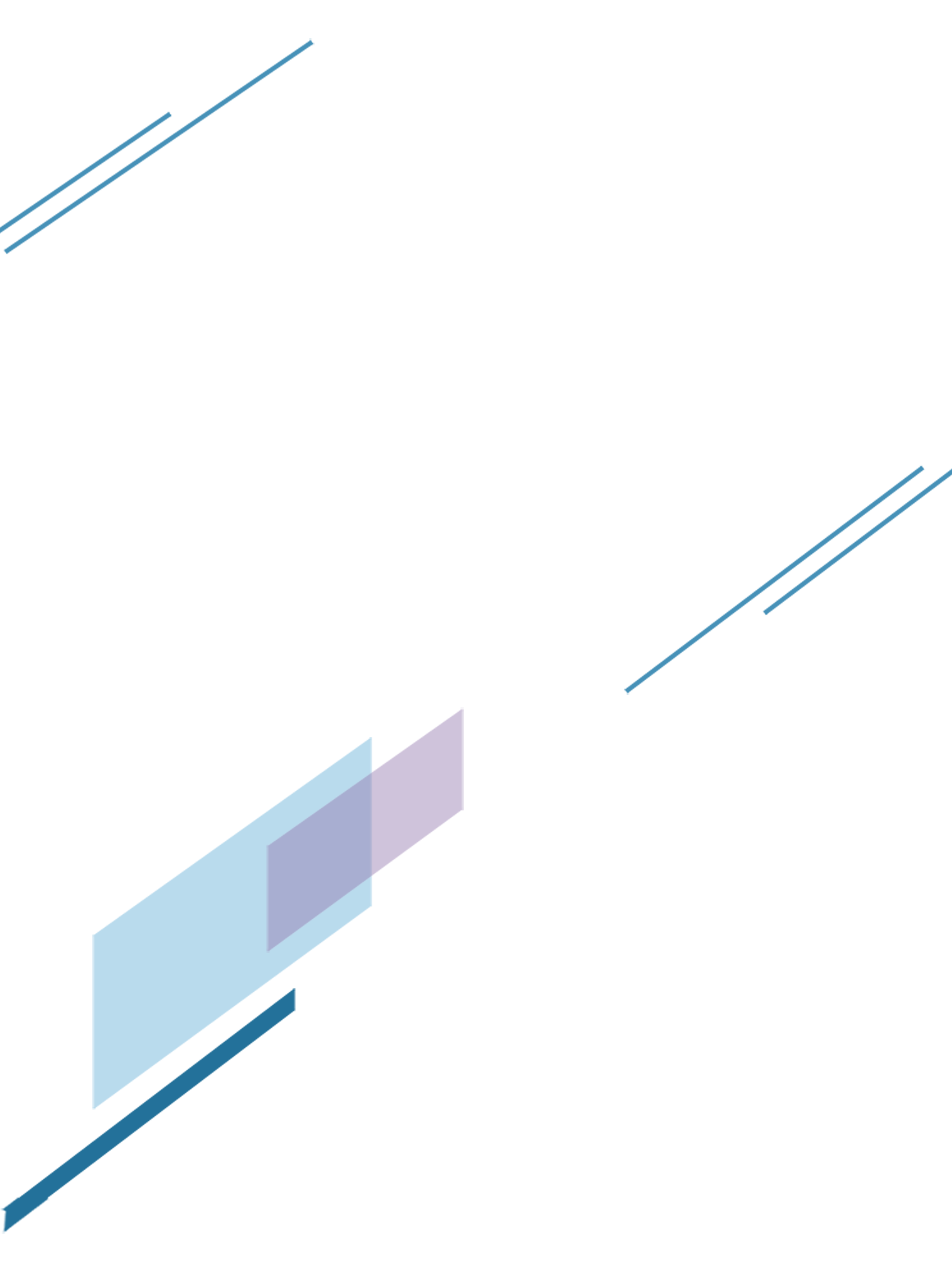
The last section was dedicated to navigation within the application. It serves as the final part of this report and is intended to present the software and hardware environment of implementation. It also describes the main functionalities that were implemented after development, accompanied by screenshots illustrating the key interface elements of the application.

**General conclusion**

Contributing to the creation of this platform has been an exceptional opportunity that allowed me to gain practical professional experience in the field of web application development using ASP.NET. This experience has been deeply enriching on multiple levels. By putting my theoretical knowledge into practice, this project not only helped me solidify my understanding but also anchored it in a real and tangible context. I deepened my programming language and platform skills, successfully exploring and leveraging them.

This report diligently details the various essential steps in designing and developing this innovative platform. Moreover, this endeavour served as an invaluable opportunity to acquire new skills, particularly in the expert use of SQL Server, Visual Studio, Docker, and Git. By mastering these tools, I further solidified my programming language skills and related technologies, notably C# and ASP.NET Core MVC.

Undoubtedly, these newly acquired skills will play a pivotal role in my future professional experiences, providing me with a distinct advantage in the job market



###### This internship has been an exceptional professional experience.

###### University life certainly provides a glimpse into our future careers, but it quickly becomes evident that practice often surpasses theoretical knowledge, offering an accelerated and concrete form of learning.