



GRADUATION PROJECT

NATIONAL ENGINEERING DEGREE

SPECIALTY : SAE

KYC Platform

By: *Rakrouki Sabri*

Academic supervisor: *Ms. Sbissi Samia*

Corporate Internship Supervisor: *M. Alaeddine Aziz*



Je valide le dépôt du rapport PFE relatif à l'étudiant nommé ci-dessous / I validate the submission of the student's report:

- Nom & Prénom /Name & Surname : Rakrouki Sabri

Encadrant Entreprise/ Business site Supervisor

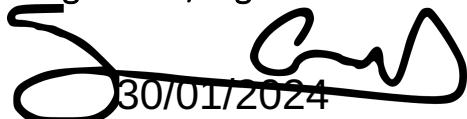
- Nom & Prénom /Name & Surname: **Mr Alaeddine Aziz**

Cachet & Signature / Stamp & Signature

Encadrant Académique/Academic Supervisor

- Nom & Prénom /Name & Surname: **Madam Sbissi Samia**

Signature / Signature



The image shows a handwritten signature in black ink, consisting of stylized loops and curves. Below the signature, the date "30/01/2024" is written in a smaller, straight-line font.

Ce formulaire doit être rempli, signé et scanné/This form must be completed, signed and scanned.

Ce formulaire doit être introduit après la page de garde/ This form must be inserted after the cover page.

Abbreviations List

API= Application Programming Interface

NOSQL= Not Only Structured Query Language

REST= Representation State Transfer

SPA= Single Page Application

JSON= JavaScript Object Notation

DB=Data base

IRS=Internal Revenue Service

FATCA=Foreign Account Tax Compliance Act

CRS= Comprehensive Ranking System

KYC=Know Your Customer

Table of Contents

General Introduction	1
Chapter 1: Project Context	1
Introduction.....	1
1.1. Company Presentation	1
1.1.1. Hydatis activities.....	1
1.1.2. Organizational structure.....	2
1.2. Project presentation.....	2
1.2.1. Problematic	2
1.2.2. Study of Existence	3
1.2.3. Proposed solution.....	3
1.3. Used Methodology	3
1.3.1. Different Methodologies Presentation	3
1.3.2. Scrum.....	4
Conclusion	5
Chapter 2: Project Analysis.....	6
Introduction.....	6
2.1. Functional part	6
2.1.1. Actor Identification.....	6
2.1.2. Function requirement	6
2.1.3. Non-Function requirement	6
2.1.4. Global use case diagram.....	7
2.1.5. Product backlog.....	8
2.2. Branch technology.....	9
2.2.1. Technologies.....	9
2.2.2. Work Tools.....	10
2.2.3. Application architectures	11
2.2.4. Sprints Planning.....	17
Conclusion	17
Chapter 3: Sprint 1- Question Management.....	18
Introduction.....	18
3.1. Functional specification.....	18
3.1.1. Sprint Backlog	18
3.1.2. Use Case Diagram of Question Management	18
3.1.3. Add Question Scenario	19

3.1.4.	Edit Question Scenario	20
3.1.5.	Delete Question Scenario.....	21
3.1.6.	Show All Question Scenario.....	22
3.2.	Realisation	23
3.2.1.	Show all Question Admin side.....	23
3.2.2.	Add Question.....	23
3.2.3.	Edit Question	24
3.2.4.	Delete Question.....	24
3.2.5.	Show the Question User Side	25
	Conclusion	27
Chapter 4: Sprint 2-Answer Management		28
Introduction.....		28
4.1.	Functional Specification	28
4.1.1.	Sprint Backlog.....	28
4.1.2.	Answer Management Use Diagram.....	28
4.1.3.	Add Answer Scenario.....	29
4.1.4.	Delete Answer Scenario	30
4.1.5.	Show all the Answers Scenario.....	31
4.2.	Realisation	31
4.2.1.	Answers Dashboard.....	31
4.2.2.	Delete Answer	32
	Conclusion	32
Chapter 5: Sprint 3- Form Management		33
Introduction.....		33
5.1.	Function Specification	33
5.1.1.	Sprint Backlog.....	33
5.1.2.	Form management Use case Diagram	34
5.1.3.	Class Diagram	34
5.1.4.	Upload Form Scenario	35
5.1.5.	Download Form Scenario	36
5.1.6.	Edit Form Scenario	37
5.1.7.	Delete Form Scenario	38
5.1.8.	Show all Forms Scenario.....	39
5.1.9.	Verify Form Scenario	39
5.1.10.	Auto recommendation form	40
5.2.	Realisation	41

5.2.1.	Form Dashboard	41
5.2.2.	Edit FW9 Form Interface	42
5.2.3.	Delete FW9 Form Interface	42
5.2.4.	Verify Fw9 Form Interface	43
5.2.5.	Auto recommendations from User side	43
5.2.6.	manual selection form User Side	44
5.2.7.	Upload file for user	44
	Conclusion	45
	Chapter 6: Sprint 4-Xml Generation and Management	46
	Introduction.....	46
6.1.	Functional Specification	46
6.1.1.	Sprint Backlog	46
6.1.2.	Xml Generation and Management Use case.....	47
6.1.3.	Class diagram of FATCA xml report schema.....	47
6.1.4.	Class diagram of CRS XML report schema.....	48
6.1.5.	Auto Generation XML Report.....	49
6.1.6.	Manual Generation XML Report	50
6.2.	Realisation	51
6.2.1.	XML Dashboard	51
6.2.2.	XML view Interface	51
	Conclusion	52
	General Conclusion	53
	Bibliography.....	54

List of figures

<i>Figure 1: Hydatis organization chart</i>	2
<i>Figure 2:SCRUM Methodology</i>	4
<i>Figure 3: Global Use case Diagram</i>	7
<i>Figure 4: Microservice Architecture</i>	12
<i>Figure 5: Global architecture</i>	14
<i>Figure 6:Logic Architecture</i>	15
<i>Figure 7 : Use case of Question Management</i>	18
<i>Figure 8:Add Question Sequence Diagram</i>	19
<i>Figure 9>Edit Question Sequence Diagram</i>	20
<i>Figure 10>Delete Question Sequence Diagram</i>	21
<i>Figure 11>Show All Question Sequence Diagram</i>	22
<i>Figure 12:Interface of all question Admin side</i>	23
<i>Figure 13: Add question Interface.</i>	23
<i>Figure 14>Edit question Interface</i>	24
<i>Figure 15:Delete Question Conformation Modal</i>	24
<i>Figure 16:FATCA or CRS choose interface</i>	25
<i>Figure 17:Individual or Entity FATCA Interface</i>	25
<i>Figure 18:Question Entity Interface</i>	26
<i>Figure 19:Question Individual Interface</i>	26
<i>Figure 20:Answer Management Use Diagram</i>	28
<i>Figure 21:Add Answer Scenario</i>	29
<i>Figure 22:Delete Answer Scenario</i>	30
<i>Figure 23>Show All Answers Scenario</i>	31
<i>Figure 24:Answer Dashboard</i>	31
<i>Figure 25:Delete Answer Interface</i>	32
<i>Figure 26:Form Use case Diagram</i>	34
<i>Figure 27: Form classes</i>	34
<i>Figure 28:Upload Form Scenario</i>	35
<i>Figure 29:Download Form Scenario</i>	36
<i>Figure 30>Edit Form Scenario</i>	37
<i>Figure 31:Delete Form Admin</i>	38
<i>Figure 32>Show All Forms!</i>	39
<i>Figure 33:Verify Form Scenario</i>	39
<i>Figure 34:Auto Recommendation Form</i>	40
<i>Figure 35:FW9 dashboard interface</i>	41
<i>Figure 36:Fw9 form Interface</i>	42
<i>Figure 37:Delete Fw9 Form Interface</i>	42
<i>Figure 38: Verify Fw9 Form Interface</i>	43
<i>Figure 39: auto recommendations form User interface</i>	43
<i>Figure 40:manual selection user interface</i>	44
<i>Figure 41:upload file for user interface</i>	44
<i>Figure 42: Xml Generation and Management Use case</i>	47
<i>Figure 43: XML FATCA schema</i>	47
<i>Figure 44:XML CRS schema</i>	48
<i>Figure 45:Auto Generation XML Report</i>	49
<i>Figure 46:Manual Generation XML report</i>	50
<i>Figure 47:XML Dashboard</i>	51
<i>Figure 48:XML view interface</i>	51

List of Tables

<i>Table 1:Company roles</i>	2
<i>Table 2:Scrum team presentation</i>	5
<i>Table 3:Product Backlog</i>	9
<i>Table 4: Question Management Sprint Backlog</i>	18
<i>Table 5: Sprint Backlog of Answer Management</i>	28
<i>Table 6:Sprint Backlog Form Management</i>	33
<i>Table 7: Sprint Backlog XML Generation and management</i>	46

DEDICACES

To express my gratitude to all those who helped and encouraged me to carry out this modest work, I dedicate this project to:

To my parents:

No matter how profound the words, they can never truly encapsulate the immense value I place on the unwavering material and spiritual support you've consistently provided, along with the sacrifices you've selflessly made, in token of my gratitude, I offer this modest endeavour, extending my wishes for your health, happiness, and a long life so that I may, in turn, reciprocate your kindness.

To my Siblings:

I offer this work as a symbol of the deep and close connections that tie us together, and as an expression of gratitude for your unwavering support and motivation. May your future be abundant in success and happiness.

To my Friends:

To my cherished friends and hardworking colleagues, I dedicate our joint achievements as a token of my gratitude for your consistent support, camaraderie, and collaborative spirit. I eagerly anticipate the path ahead, filled with further accomplishments and cherished shared moments.

To All my family

To all those who have supported me and continue to support me

Rakrouki Sabri

ACKNOWLEDGEMENTS

This project is the fruit of six months' work at HYDATIS, during which I benefited from the guidance, encouragement, and support of many of my colleagues, during which I benefited from the guidance, encouragement, and support of numerous people, to whom I wish to express my deep and sincere THANKS.

I would first like to express my gratitude to my supervisors at my host company **Mr Alaeddine Aziz ,Ms. Hziem Abir** and **Ms. Zammouri Semia**, who supported me from the outset, agreed to take on the responsibility of guiding me, monitoring my work and, above all, explaining the work to me to take on the task occupations.me, following my work and, above all, explaining it to me required, despite his many occupations as well as **Ms. Ikbel Souelmi**, our project PO, for her vision and support throughout the development of the project.

I would also like to thank my supervisor at Private University of Tunis **Ms. Sbissi Samia**, for monitoring and supervising my project. His encouragement edified me greatly.

My deepest thanks also go to all the teachers at Private University of Tunis, thanks to whom I was able to complete this work.

A special thank you to the Hydatis development team, whose professionalism, mutual assistance, and team spirit have created a stimulating and enriching work environment. your collaboration was essential to achieve the objectives set and to make this experience more enriching.

Finally, I'd like to thank all the people who have helped me, both directly and indirectly.

General Introduction

In the age of the digital banking industry and online transactions, information privacy is a paramount concern. Banks handle big amounts of very sensitive data, including personal and financial data of their customers. Maintaining the confidentiality and security of this information is essential to foster trust and ensure the integrity and cybersecurity defences are imperative to safeguard against breaches and cyberattacks that compromise customer privacy and lead to significant financial losses.

To enhance information security and mitigate financial risks, banks often implement Know Your Customer (KYC) procedures. KYC is a set of regulations and processes that require banks to verify the identity of their customers, understand their financial activities, and assess potential risks associated with those customers. KYC measures are not only crucial for regulatory compliance but also serve as a vital component in the protection of customer data and financial assets. By ensuring that customers are who they claim to be and that their transactions are legitimate, KYC plays a critical role in maintaining the privacy, security, and overall stability of the banking industry.

contextually , our final year internship project "KYC platform" will automatise the process of validating information, generating xml codes and saving it in the database and so on in the first chapter we will present the host company ,the project that we are working on and the methodology that , while in the second one will deal with the presentation of the global conceptual study, actors and function and non-function requirements that we have, also we presented the technologies and work tool we are using and sprint planning we have, moving on to first sprint that we specifically provide the question management through the diagrams, description and scenarios that we have, next thing we will present the second sprint answer management through the diagrams, description and scenarios we provided, additionally we have third sprint that specified for Form management that we will present it through diagrams, description and scenarios we have and finally we have the last sprint xml generation and management we will present xml schema for FATCA and CRS.

Chapter 1: Project Context

Introduction

In this first chapter, I will first introduce my host company, followed by a brief description of the project I worked on. We will also present the study of existent. And finally, we'll pose a few problems and propose the solutions which we will be describing throughout this document.

1.1. Company Presentation

In order to carry out a high-quality project, it is often essential to draw up a comprehensive study of the host organization. This section is devoted to presentation of the "Hydatis" company. Figure 1 shows the Hydatis logo.

Hydatis, a dynamic consulting and services company



IT engineering. Its mission is to support its customers and partners in the implementation of efficient and innovative solutions in the fields of administration, finance, and industry.

1.1.1. Hydatis activities

Services are at the heart of Hydatis' approach. As a true research and Hydatis' team provides high value-added services in the fields of embedded systems, information systems and mobility applications. The Hydatis offering is characterized by a high level of productivity, capitalization on experience and a customer-oriented approach. Thanks to the technological expertise of its teams in the field of information systems, Hydatis advises and works in close collaboration with its customers, respecting their constraints in terms of cost,

performance and efficiency. respecting their constraints in terms of costs and deadlines to transform their visions into reality.

Hydatis teams help their customers with the functional aspects of their tools. tools. A network of IT services partners and business experts are at their disposal to their customers' various needs.

1.1.2. Organizational structure

Hydatis Engineering, Limited Liability Company (LLC), incorporated in 2020. Figure 2 shows the Hydatis organization chart.

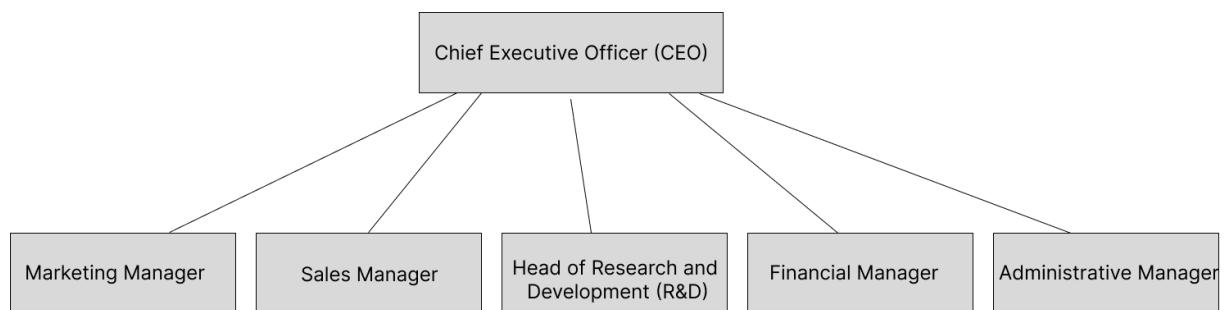


Figure 1: Hydatis organization chart

According to figure 2, here are the main responsibilities associated with each position within the company:

Position	Primary role
Chief Executive Officer (CEO)	Oversees and directs the company's global strategy.
Marketing Manager	Develops and implements marketing strategies.
Sales Manager	Leads the sales team and builds relationships with customers.
Head of Research and Development (R&D)	Guide the research and development of new products and services.
Financial Manager	Manages the company's finances and financial strategy.
Administrative Manager	Oversees administrative functions and regulatory compliance.

Table 1: Company roles

1.2. Project presentation

1.2.1. Problematic

The FACTA system are all manual and are facing a lot of problem like delaying and backlogging the manual document verification process which is time-consuming also we can mention that human interfering in the process can lead to compliance violation and potential fraud incidents.

1.2.2. Study of Existence

We have identified three products have some similarity in the way of business model that we are aiming for in our project:

Sumsub: Sumsub is an identity verification and compliance platform used by businesses to streamline customer onboarding and meet regulatory requirements.

Onfido: Onfido is another identity verification platform that uses AI and machine learning to verify identities. It enables businesses to verify the identities of their users by analyzing various documents (such as IDs, passports, or driver's licenses) and biometric data.

Plaid: Plaid is a different kind of platform compared to Sumsub and Onfido. It's a data connectivity platform that enables applications to connect with users' bank accounts. It offers APIs that allow developers to integrate banking and financial data into their applications securely.

1.2.3. Proposed solution

Based on what we provide in the study of existence and the problematic that we had. We conclude that we don't have a specific solution for our product that is why Hydatis provided this solution which is KYC platform that can identify, verify, and export a suitable XML report with IRS rules for every year.

1.3. Used Methodology

As far as IT projects are concerned, the methodologies used have evolved considerably evolved. We've gone from classic methodologies to highly advanced ones designed to meet certain needs and solve relevant problems.

1.3.1. Different Methodologies Presentation

Older methodologies boil down to cascading development cycles which have many negative points. They require the customer's requirements to be requirements to be complete and defined at the outset of the project, which means that a lot of documentation. In addition, quality control is only carried out at the end of the cycle. at the end of the cycle. All this adds up to a higher failure rate for the projects in question.

The emergence of new methodologies has solved some of these problems. These methodologies, known as agile, give greater importance to the individuals involved in the

project, reinforcing collaboration, and avoiding a purely solely on the process. To achieve this, waterfall development cycles have been replaced by incremental and iterative cycles. In addition to this, these methodologies are minimalist and open to change. They also enable problems to be detected problems early in the project, while providing continuous improvements.

1.3.2. Scrum

Scrum is simple. Try it as is and determine if its philosophy, theory, and structure help to achieve goals and create value. The Scrum framework is purposefully incomplete, only defining the parts required to implement Scrum theory. Scrum is built upon by the collective intelligence of the people using it. Rather than provide people with detailed instructions, the rules of Scrum guide their relationships and interactions.[1]

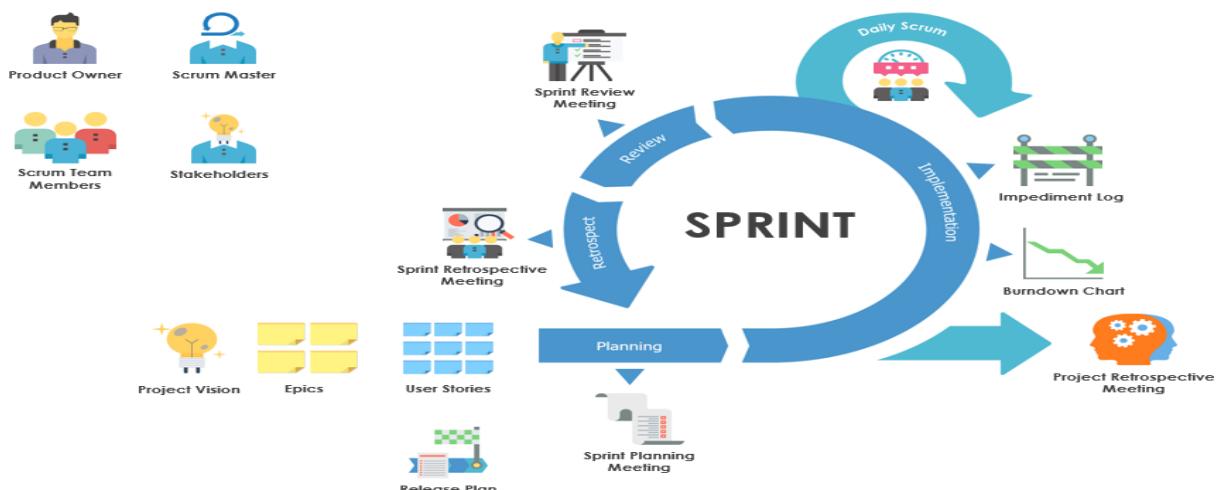


Figure 2:SCRUM Methodology

During my internship, I adopted the agile methodology called scrum. This methodology enables us to maintain good communication between the members of the team working on our project, but also to improve the project in a simple way. To be able to follow this methodology, you need to start by knowing the role of each member, since the scrum process clearly defines everyone's role as well as the overall way in which the project is managed.

The project team is made up of four members. The product owner is responsible for the application, defining requirements and deciding on the direction the platform will take. Then there's the scrum master, who is the guarantor of the scrum methodology. His role is to

ensure that the methodology is respected and properly applied, by resolving any problems that may arise and maintaining communication with the entire team. Finally, the development team, which is responsible for producing the product in question.

Role	Full Name	Responsibility
Product owner	Mme. Ikbel Souelmi	Defining the product vision
Development team	Rakrouki Sabri (Myself)	<ul style="list-style-type: none"> • Develop all parts of the application (Front-end and Back-end). • Manage application architecture
Scrum Master	Mme. Samia Zammouri	<ul style="list-style-type: none"> • Ensure that the team follows Scrum principles. • Facilitate Scrum meetings such as the sprint planning meeting, sprint review and retrospective. • Help resolve obstacles that the team's progress

Table 2:Scrum team presentation

Conclusion

In this chapter, we presented the host organization, also we presented the study of existent, and finally, we presented the problem and proposed solution that we will solve. In the next chapter we will present conceptual study of and methodology we worked with.

Chapter 2: SPRINT 0-Project Analysis

Introduction

In this chapter we will present project analysis start with actor identification where we will define the actor that we have in our application move forward to function requirements and non-function requirements , global class diagram after that we will present the product backlog also we will present the technologies and tools we will use in the development of this project and the Application Architecture we had we will finish by the print planning we have.

2.1. Functional part

2.1.1. Actor Identification

We have identified three actors in our project that they have some privilege.

Visitor: a visitor that he can access the first login page and authenticate in our project.

Admin : an admin who has the dashboard to manage the data that we have in the project and export the xml report manually and consult the other data.

Simple user: a simple user he can answer the quiz and upload the document that he has. A simple user can be an individual or an entity.

2.1.2. Function requirement

During the discussion phase, the following needs were identified:

- **Question Management:** This model allows you to create, modify and delete a question from the quiz's pages.
- **Answer Management:** this module allows you modify and delete an answer from database.
- **Forms Management:** this module allows you to auto identification, modify, delete and verify multiple form on dashboard.
- **Xml Report Generation and Management:** this module allows you to auto generate yearly and manually and delete an XML report.

2.1.3. Non-Function requirement

Non-functional requirements guarantee that the platform will function correctly following implementation of the functional requirements. We have concluded that the platform must meet the following requirements:

- **Performance:** our project verifies and processes the given information within seconds
- **Scalability:** the platform can handle multiple requests without impacting the response time.
- **Ergonomic:** our platform uses user friendly interface and web base applications that make run at any device.
- **Flexibility:** our platform gives the administrator the ability to configure the question through an easy interface and give the developer an easy written code with clean code rules to maintain.

2.1.4. Global use case diagram

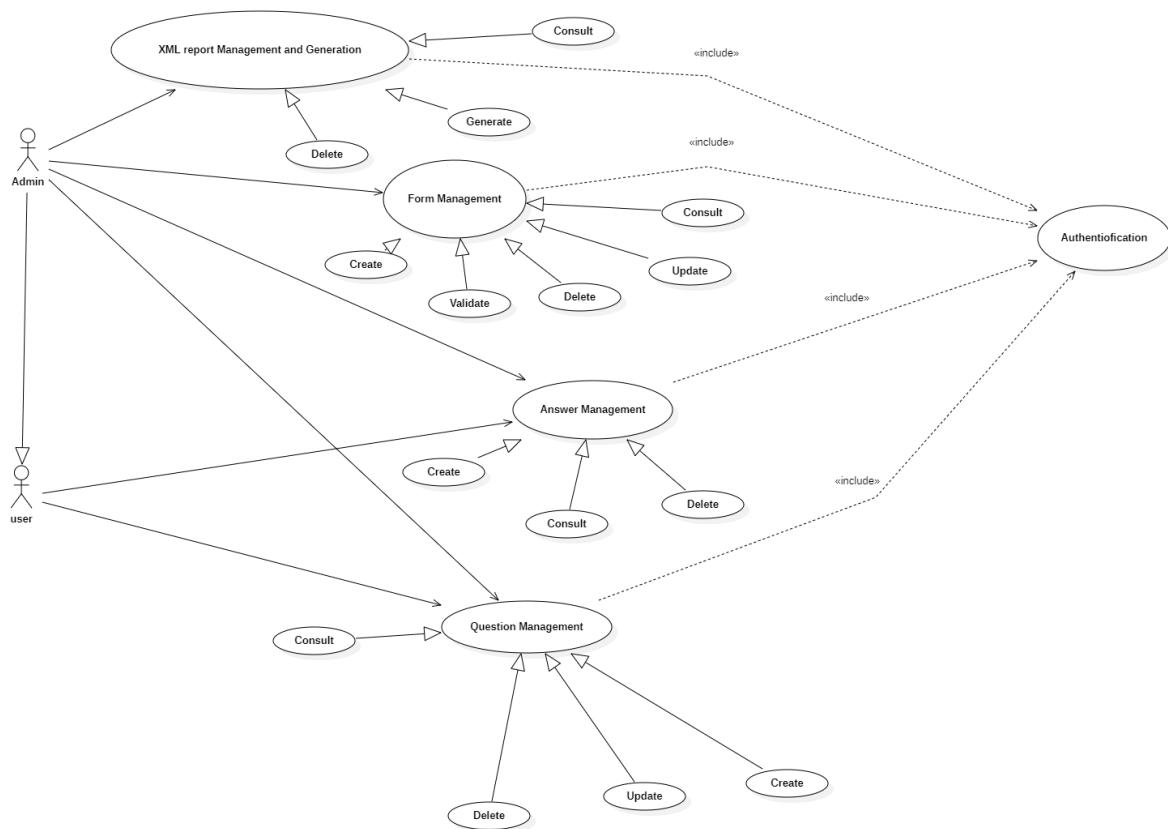


Figure 3: Global Use case Diagram

2.1.5. Product backlog

Having defined these requirements, we're now going to detail them in the form of user stories in the product backlog. This allows us to get as close as possible to the customer's vision of the product.

Feature ID	Feature	Story ID	User Story	Priority
0	Study of the project	0.1	Documentation on and technologies.	Average
		0.2	Identification of functional and non-functional.	High
		0.3	Conception	High
		0.4	Tool installation and preparation environment	High
		0.5	Project creation template front-end and back-end	Average
		0.6	creation databases	High
1	Question Management	1.1	As an administrator I can add questions	High
		1.2	As an administrator I can modify questions	High
		1.3	As an administrator I can delete questions	High
		1.4	As an administrator I can consult questions	Average
		1.5	As a User I can Consult the question	High
2	Answer Management	2.1	As a user I can choose between FATCA or CRS	High
		2.2	As a user I can choose between Individual or Entity	High
		2.3	As a user I can answer the questions	High
		2.4	As a user I can submit my answers	High
		2.5	As an administrator I can consult the answers	Average
		2.6	As an administrator I can delete the answers	Average
3	Form Management	3.1	As a user I get recommended a form by an algorithm base on user answers	High
		3.2	The user choose entity an email will send to shareholders to answer the question and upload the form suitable.	High
		3.3	As a user I can choose between the manual selection or the recommendation	Average
		3.4	As a user I can upload the form that have been fill out with user sign in	High

		3.5	As an administrator I can download the file that the user has been uploaded	High
		3.6	As an administrator I can delete the form that the user has been uploaded	High
		3.7	As an administrator I can modify the form that the user has been uploaded	High
		3.8	As an administrator I can verify the form that the user has been uploaded	High
4	XML Report generation and Management	4.1	As an administrator I can consult the xml reports	High
		4.2	As an administrator I can manual generate FATCA report	High
		4.3	As an administrator I can manually generate CRS report	High
		4.4	The platform can automatically generate XML FATCA report yearly	High
		4.5	The platform can automatically generate XML CRS report yearly	High

Table 3:Product Backlog

2.2. Branch technology

At this level, we'll discuss the technologies and tools we've employed. We will unveil the logical and physical architecture for the development of our application.

2.2.1. Technologies

Java is an object-oriented language, which means all programs are made of entities representing concepts or physical things known as “objects”. Java programs are found in desktops, servers, mobile devices, smart cards and Blu-ray Discs (BD)[2]



Spring boot is an open-source Java-based framework used to create a micro-Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications. This chapter will give you an introduction to Spring Boot and familiarizes you with its basic concepts.[3]



Kafka is a distributed event store and stream-processing platform. It is an open-source system developed by the Apache Software Foundation written in Java and Scala.[4]



Angular is an application-design framework and development platform for creating efficient and sophisticated single-page apps.[5]



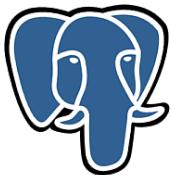
CQRS pattern Command Query Responsibility Segregation (CQRS) is the segregation of the responsibilities of the commands and queries in a system. That means that we're slicing our application logic vertically. In addition to that, we're segregating state mutation (command handling) from the data retrieval (query handling). [6]

2.2.2. Work Tools

IntelliJ IDEA is an integrated development environment (IDE) written in Java for developing computer software written in Java, Kotlin, Groovy, and other JVM-based languages. It is developed by JetBrains [7]



PostgreSQL is an advanced, enterprise class open-source relational database that supports both SQL (relational) and JSON (non-relational) querying. It is a highly stable database management system [8]



PostgreSQL

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.[9]



Microsoft Azure, often referred to as **Azure** is a cloud computing platform run by Microsoft. It offers access, management, and the development of applications and services through global data centres. [10]



2.2.3. Application architectures

On this figure right blow we have the entities that we worked with through this project.

2.2.3.1. Microservice Architecture

Microservices are a service-oriented architecture model in which applications are built by grouping together several small, independent service units. This software engineering approach emphasizes the decomposition of an application into single-function modules, each with clearly defined interfaces. Figure 4 illustrates a microservices-based architecture.

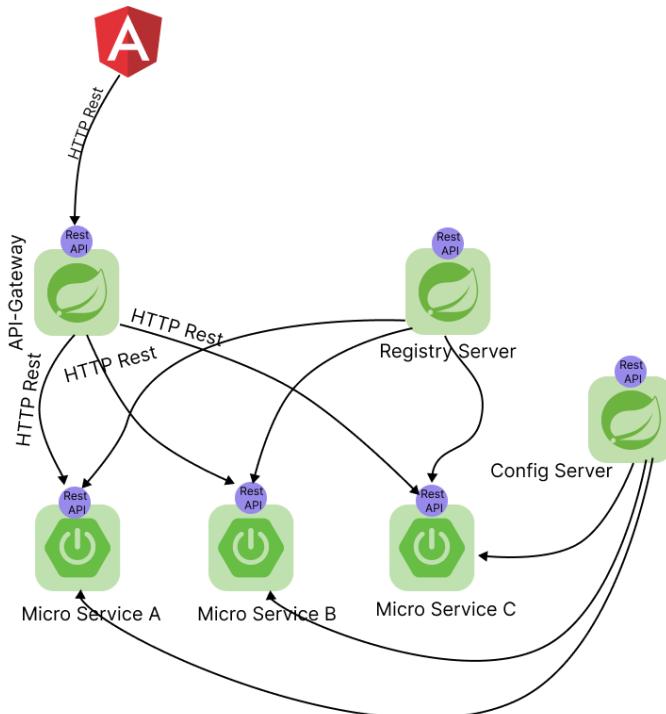


Figure 4: Microservice Architecture

The architecture of the "KYC Platform" application, illustrated in figure 4, is based on the principle of microservices architecture. This modular design makes it possible to break down the application's key functionalities into a series of independent microservices. The advantages of this approach are multiple:

- **Reducing dependency:**

By isolating the various functionalities into autonomous microservices, we prevent the system from becoming too interdependent. This reduces the risk of cascading errors and makes it easier to maintain and upgrade each service individually.

- **Improved availability**

Each microservice can be deployed, scaled, and maintained independently of the others. This means that a failure in one service will not directly affect the others, which resilience, and overall system availability.

In this architecture, we note the presence of the following elements:

- **Spring Boot backend**

Microservices: These elements form the functional core of the application. They represent the various modules that work independently, guaranteeing a high degree of modularity and flexibility.

- **Microservice API Gateway**

Serves as a single point of entry for all requests from the frontend. This microservice orchestrates and directs requests to the appropriate microservices, facilitating managing and securing interactions between the frontend and microservices.

- **Angular Frontend**

The frontend is built using the Angular framework. This technical choice enables us to create a responsive, modern user interface, capable of efficiently consuming the APIs exposed by the API Gateway.

In summary, the microservices architecture implemented for the "KYC Platform" application favors better dependency management and a significant improvement in availability, while enabling fluid and secure integration thanks to the API Gateway microservice and a dynamic frontend developed with Angular.

2.2.3.2. Global architecture

The microservices architecture will be adopted for our application because of its many advantages, including the ability to manage complex, scalable applications. This approach offers benefits such as the possibility of using heterogeneous technologies, increased resistance to failure and ease of deployment.

For exchanges between microservices, we opted for:

- The Event Sourcing pattern: a microservice architecture built around an event-driven approach. This guarantees the modularity, scalability and stability of our platform. Event Sourcing is a powerful architectural model that records all changes to an

application's state as a sequence of events, in the chronological order in which they were applied.

- The CQRS (Command and Query Responsibility Segregation) approach in conjunction with Event Sourcing: this model separates the logic of order processing from that of query processing. By separating commands and queries, the CQRS model enables developers to use different models for reading and writing data. Commands write updates to the corresponding topics, while queries use consumers to create data projections in the topics.
- For data management, we have chosen to adopt a polyglot persistence approach in line with the CQRS approach. We have divided data persistence into a read database and a write database to facilitate fluid access to the same data source. What's more, this approach allows multiple data sources to be used within the same microservice.

Figure 5 describes the overall architecture of our application.

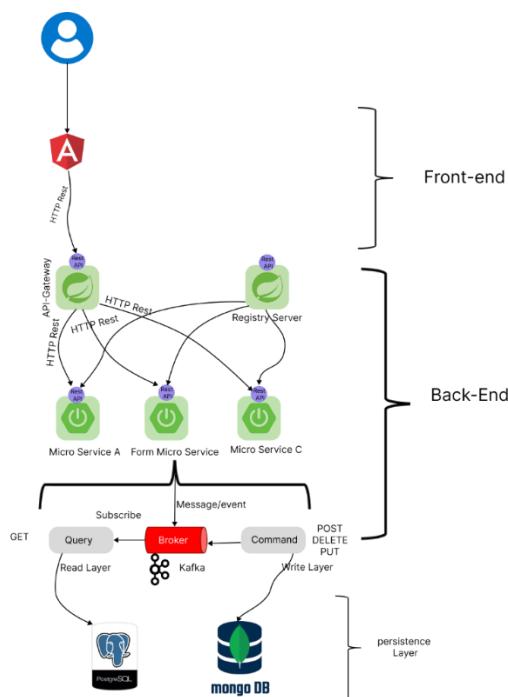


Figure 5: Global architecture

2.2.3.3. Logic Architecture

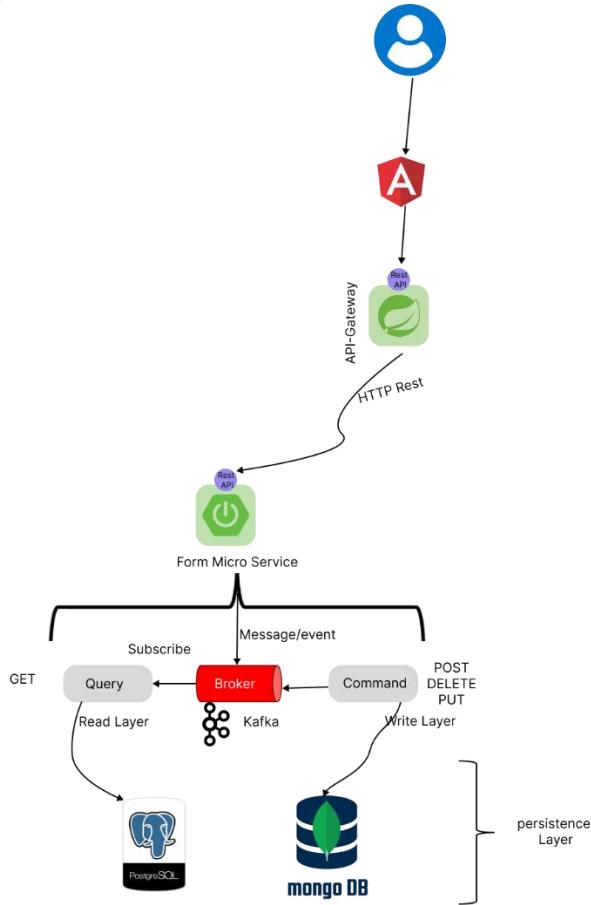


Figure 6:Logic Architecture

Angular follows the MVVM (Model-View-Model-View) architecture, as follows

- Model: represented by a class that is referenced by the data access layer (TypeScript class or interface).
- View:
 - Contains the layout and appearance visible to the user.
 - Reacts to user interactions, such as clicks, inputs, etc.
- Model View: replaces the controller in MVC (Model-View-Controller) architecture, connected to the view via data binding. In Angular, it is represented by a `nom.component.ts` file.

The following layers are available for Spring boot:

Command Layer / Command Handling:

- Responsible for receiving and handling commands that change the state of the system.
- Converts commands into domain events.
- Typically involves command validation, authorization, and dispatching to the appropriate domain service or aggregate.

Domain Layer:

- Contains the core business logic and domain entities.
- Handles domain events and updates the state of aggregates/entities.
- Enforces business rules and invariants.

Event Layer / Event Handling:

- Captures and processes domain events.
- Responsible for updating read models or projections.
- May involve handling compensating events in case of failures.

Event Store:

- Persists domain events in an event store.
- Retrieves events for replay and projection.
- Often used in event sourcing to rebuild the state of aggregates.

Command Bus:

- Dispatches commands to the appropriate command handlers.
- May involve asynchronous communication for better scalability.

Query Layer / Query Handling:

- Handles read requests by querying read models or projections.
- Optimized for fast retrieval of data.
- Decoupled from the command layer to support CQRS.

Read Model / Projection Layer:

- Contains denormalized data optimized for specific read operations.
- Updated asynchronously in response to domain events.
- Supports efficient queries for read operations.

Event Bus / Message Broker:

- Facilitates communication between different parts of the system using events.
- Decouples producers and consumers of events.
- Ensures that events are delivered reliably.

Infrastructure Layer:

- Contains infrastructure-related concerns such as database access, messaging, and external services integration.
- Provides implementations for event stores, message brokers, and other external dependencies.

Application Layer:

- Glues together the different layers.
- Orchestrates the flow of commands, events, and queries.
- Typically contains application services that coordinate the interaction between the domain and the infrastructure.

API Layer:

- Exposes APIs for external clients or other services to interact with the application.
- Includes RESTful endpoints, GraphQL, or other communication mechanisms.

Configuration Layer:

- Handles configuration of the application components.
- Manages external configurations and dependencies.

2.2.4. Sprints Planning

Using the Scrum method, we had planned our sprints in time intervals according to the evaluation of each task.

- Sprint 1: Question Management
- Sprint 2: Answer Management
- Sprint 3: Form Management
- Sprint 4: XML report and Management

Conclusion

In this chapter, we had detailed the functional part, which includes the identification of the functional and non-functional requirements of our solution, and the identification of the stakeholders. We then move on to the technical side of things, with its various components. We close this study with a sprint plan. The next chapter will focus on the first sprint, where we'll explain the work carried out in this sprint.

Chapter 3: Sprint 1- Question Management

Introduction

This chapter is dedicated to the first sprint. We will be looking at the design of the Question management task and its implementation. To achieve this, we're going to break down the module into several simple titles.

3.1. Functional specification

3.1.1. Sprint Backlog

Sprint Id	Sprint Name	Story Id	User Stories	Priority
1	Question Management	1.1	As an administrator I can add questions	High
		1.2	As an administrator I can modify questions	High
		1.3	As an administrator I can delete questions	High
		1.4	As an administrator I can consult questions	Average
		1.5	As a User I can Consult the question	High

Table 4: Question Management Sprint Backlog

3.1.2. Use Case Diagram of Question Management

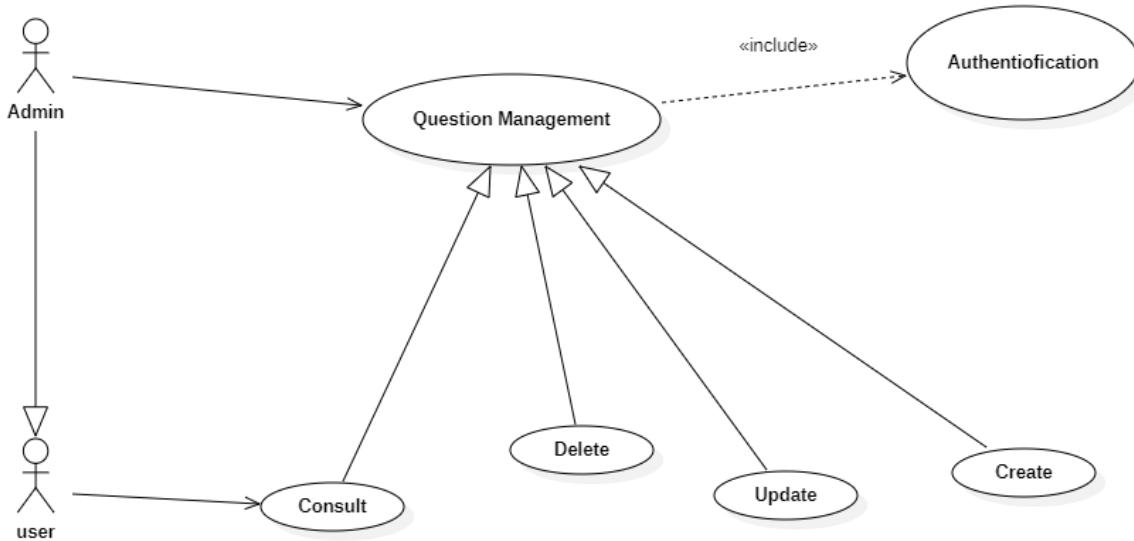


Figure 7 : Use case of Question Management

3.1.3. Add Question Scenario

When admin add question, he will click on create button on the dashboard and fill the fields necessary and submit after that a request will sent to QuestionController that will process the request, save event in write database and will produce an Add Question event that will consume it the QuestionEventConsumerService and update the read database.

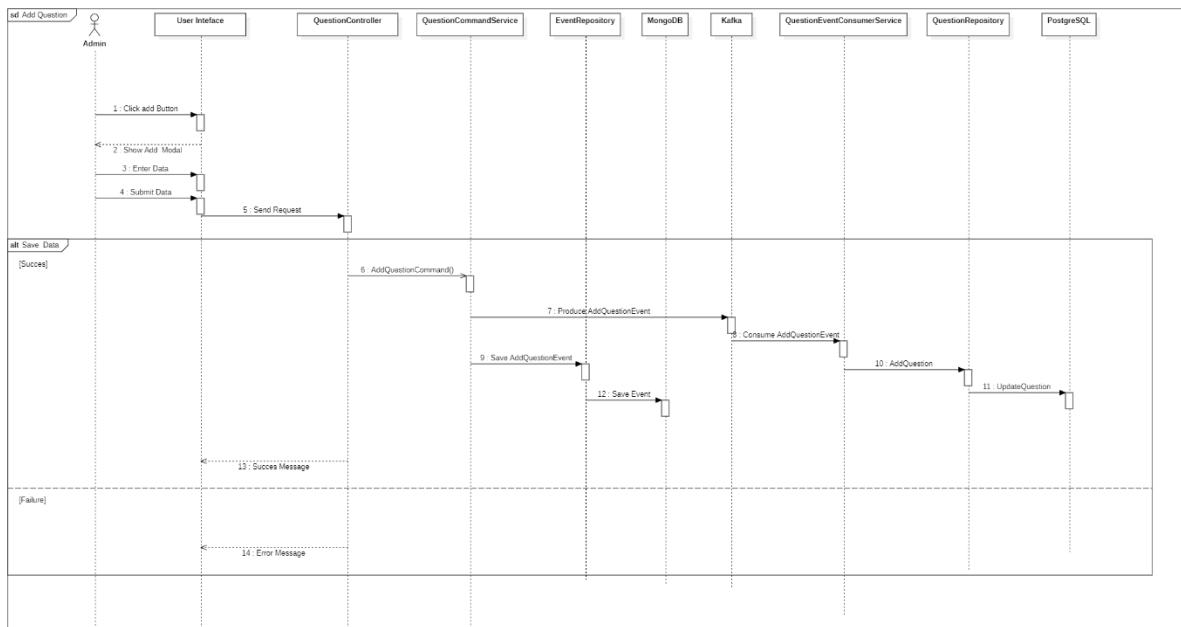


Figure 8:Add Question Sequence Diagram

3.1.4. Edit Question Scenario

When the admin will click the edit button, fill the data necessary and submit the data a update request will sent to QuestionCommandService that will process the request if the data pass successfully QuestionCommandService will save event into the write database and produce event to Kafka after that the QuestionEventConsumerService will consume the event and update the read database. If the data produce error will send error messages for admin.

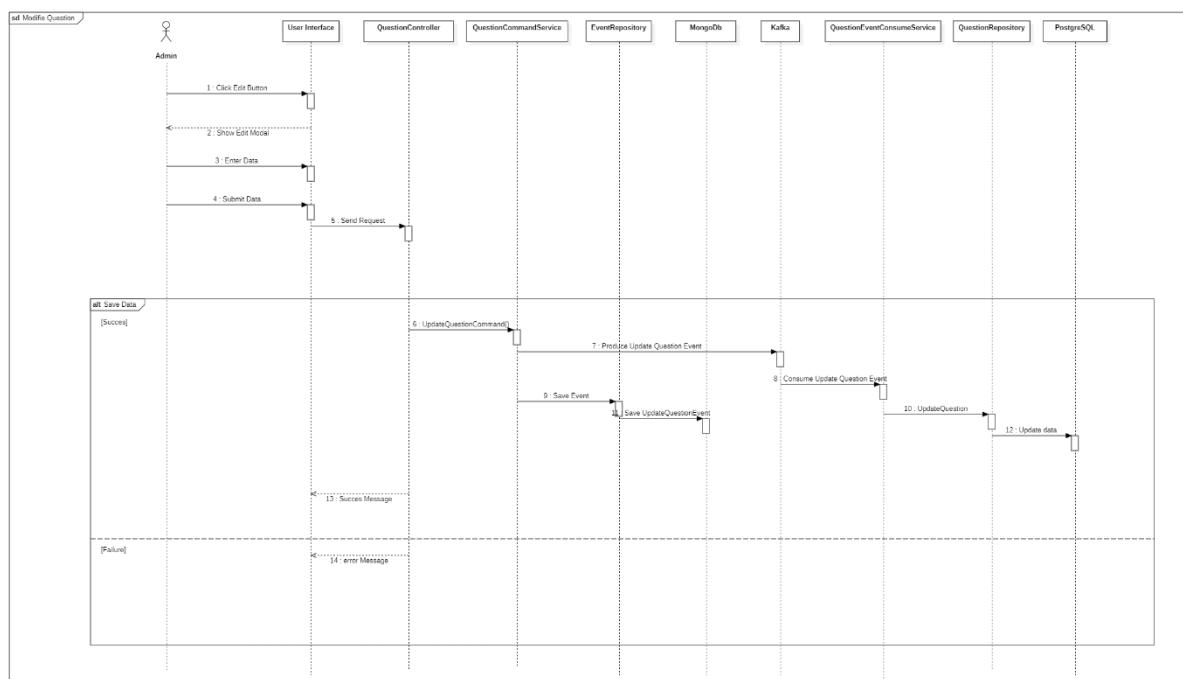


Figure 9:Edit Question Sequence Diagram

3.1.5. Delete Question Scenario

When the admin will click the edit button and submit a delete request will be sent to microservice that will process the request if the data pass successfully QuestionCommandService will save event into MongoDB and produce event to Kafka after that the QuestionEventConsumerService will consume the event and update the read database. If the data produce error will send error messages for admin.

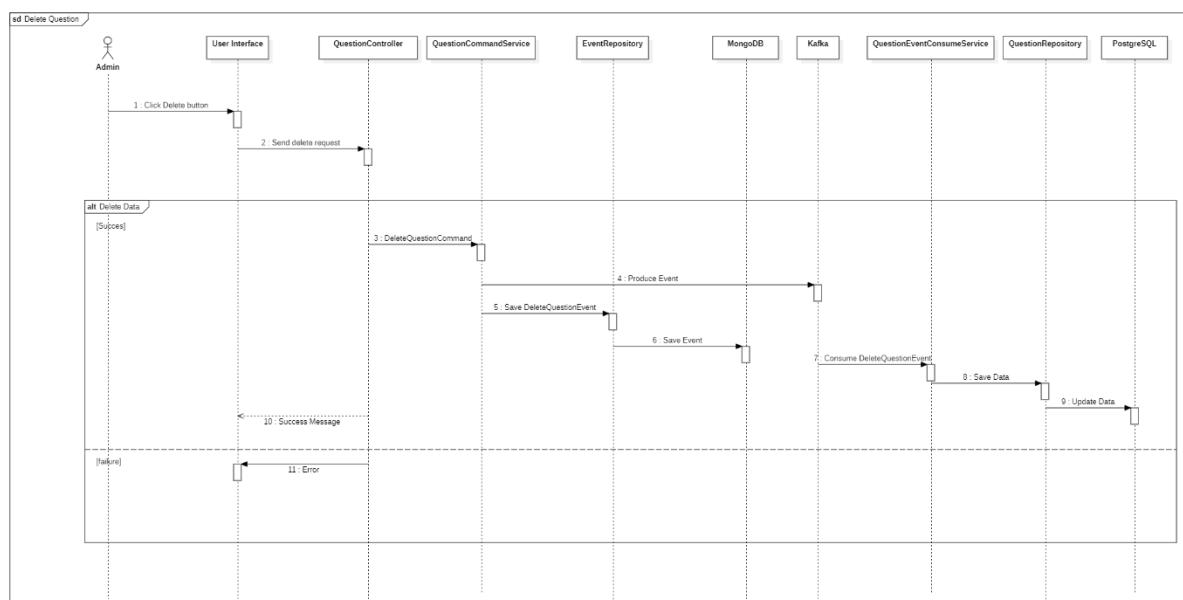


Figure 10:Delete Question Sequence Diagram

3.1.6. Show All Question Scenario

After the admin select question on the dashboard will send request for QuestionQueryController that will launch a query that will retrieve the data from the read database and send to user interface.

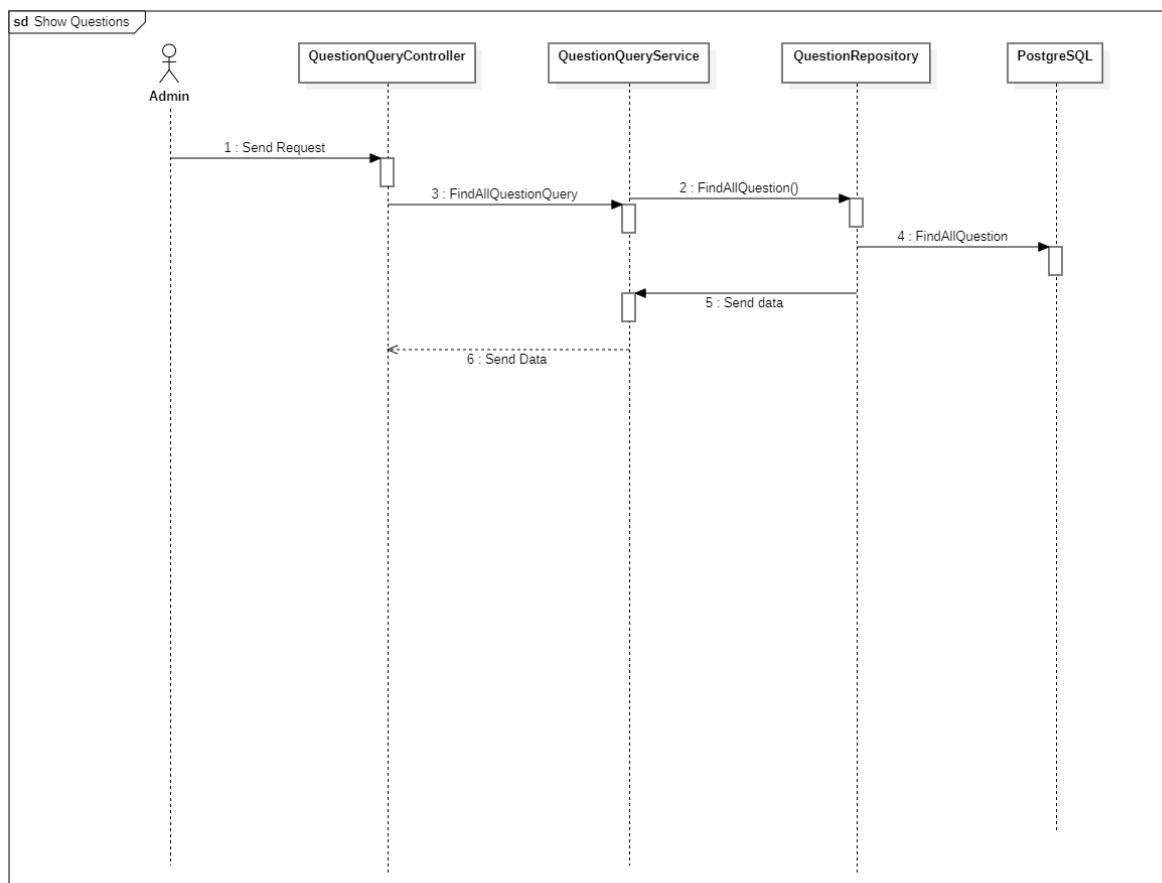


Figure 11:Show All Question Sequence Diagram

3.2. Realisation

3.2.1. Show all Question Admin side

In this interface we can see all questions that have been added and we can see add question button on top also there is edit and delete button for every question.

No.	Question Content	Score	Critical	Type of Question	Edit	Delete
bec6d028-a35c-4f0d-b80f-7f0f6c592c04	The company is incorporated under US law/incorporated in the US	5	false	Entity	<button>Edit</button>	<button>Delete</button>
bb64e999-2dd0-47c9-80ea-de32ba261d52	The company has a TIN code	5	false	Entity	<button>Edit</button>	<button>Delete</button>
030992d6-fcb1-4d39-a200-d3af90db75	Etablissement aux Etats Unis (sté mère américaine)	5	false	Entity	<button>Edit</button>	<button>Delete</button>
49cbe0f7-9840-451a-82ec-8a356016cdde	Mailing address in the USA	5	false	Entity	<button>Edit</button>	<button>Delete</button>
2d982641-26e5-4243-b514-a3ad77d97333	The capital of the company is held by one or more shareholders who are US taxpayers, holding at least 10% of the capital.	5	false	Entity	<button>Edit</button>	<button>Delete</button>

Items per page: 5 | 1 - 5 of 15 | < < > >|

Figure 12: Interface of all question Admin side

3.2.2. Add Question

On the interface below we can see how the question can be added base on score, if question is critical that mean he is use citizen if not he will identify as use indica also there is two type question: Entity question, Individual question

ADD QUESTION

Question Content*

Score*

Check if it's critical Question

Select for whom the Question*

Figure 13: Add question Interface.

3.2.3. Edit Question

In the interface we can see that how the question can be edited.

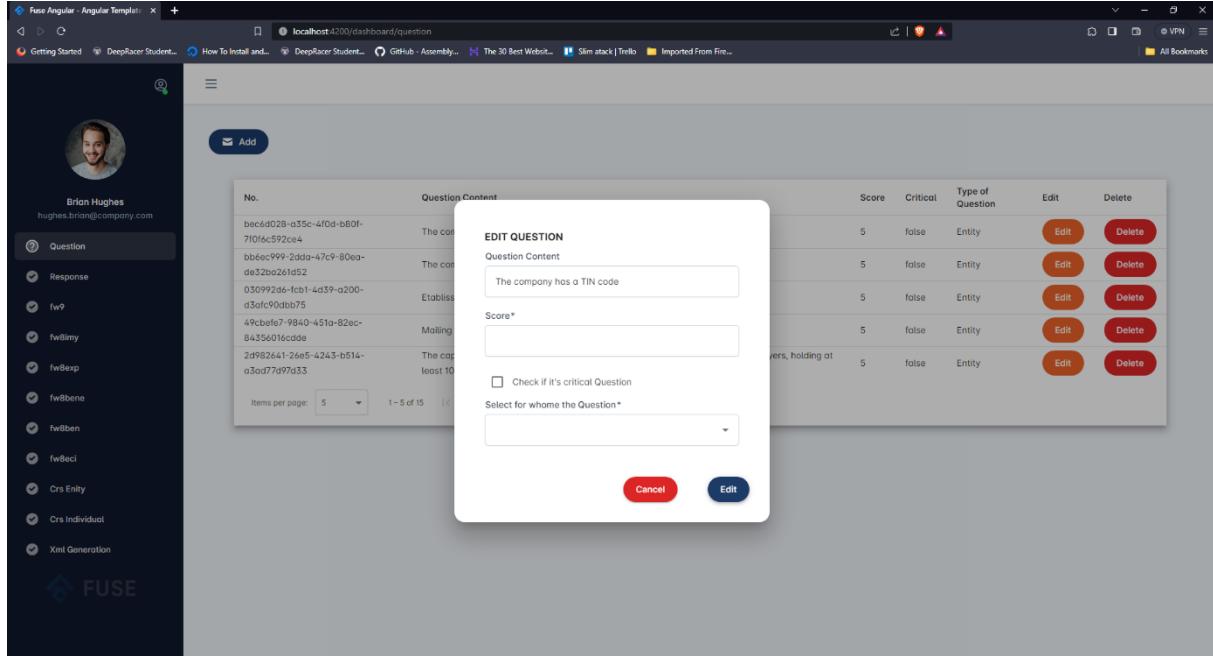


Figure 14:Edit question Interface

3.2.4. Delete Question

When the admin click delete a modal will pop up with confirmation message.

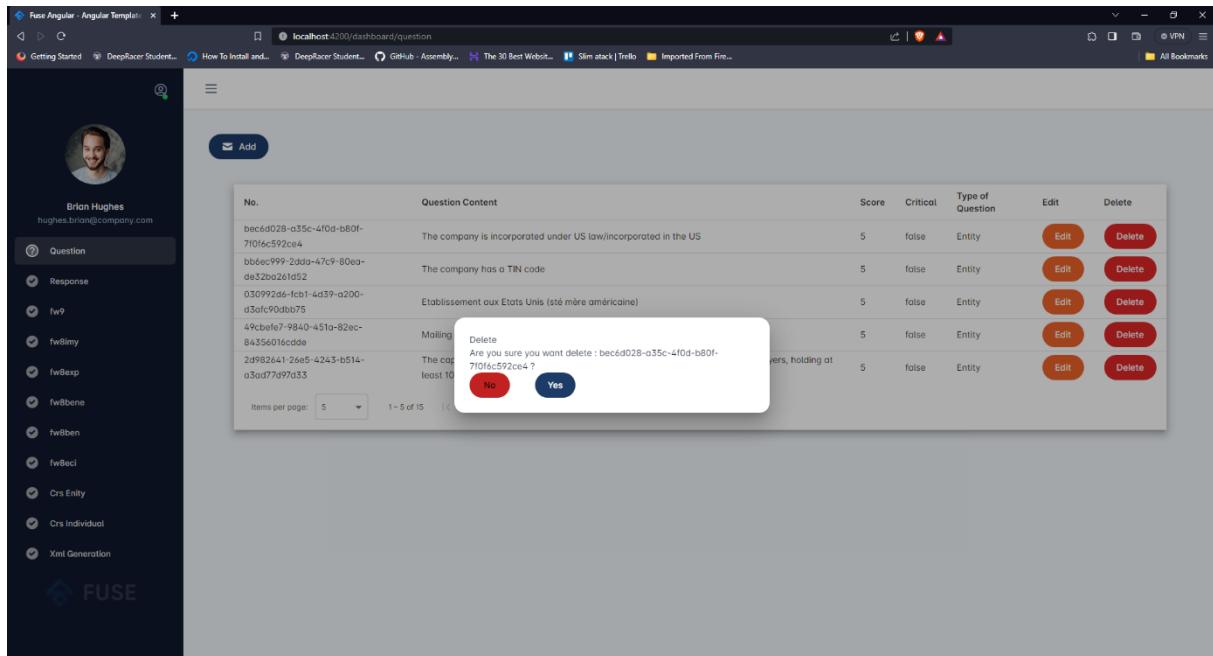


Figure 15:Delete Question Conformation Modal

3.2.5. Show the Question User Side

After login the user will got choose between FATCA and CRS.

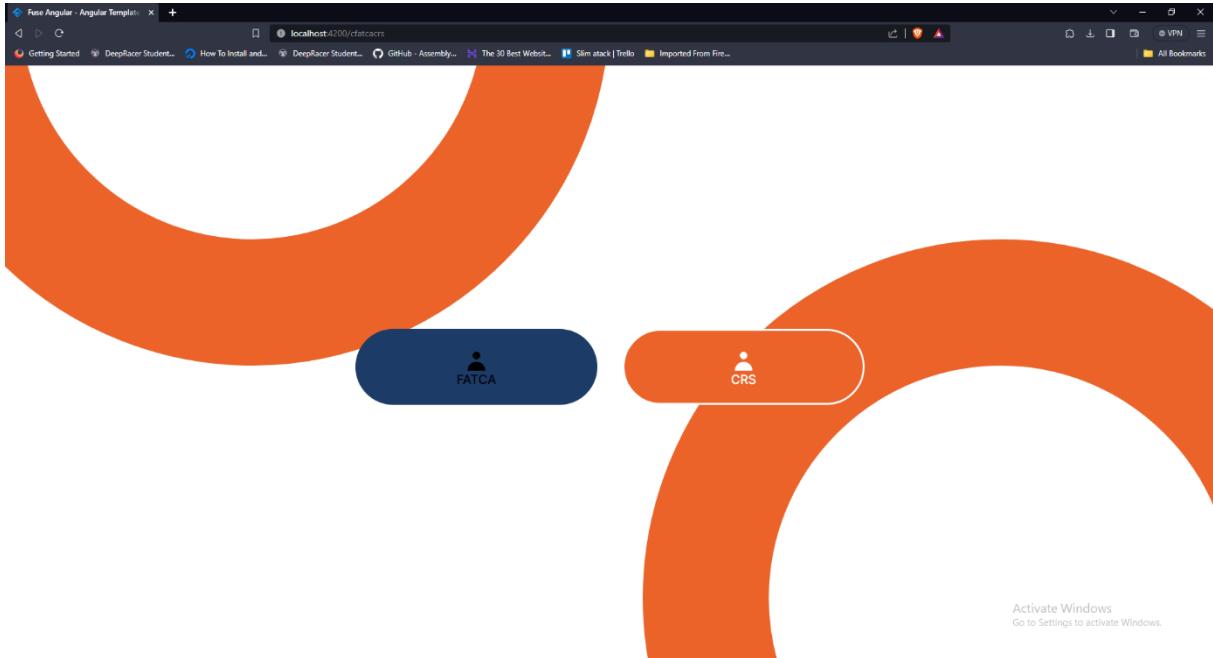


Figure 16:FATCA or CRS choose interface

After choosing FATCA we will move to the next selection between FATCA Individual or Entity that will lead us to different interfaces.

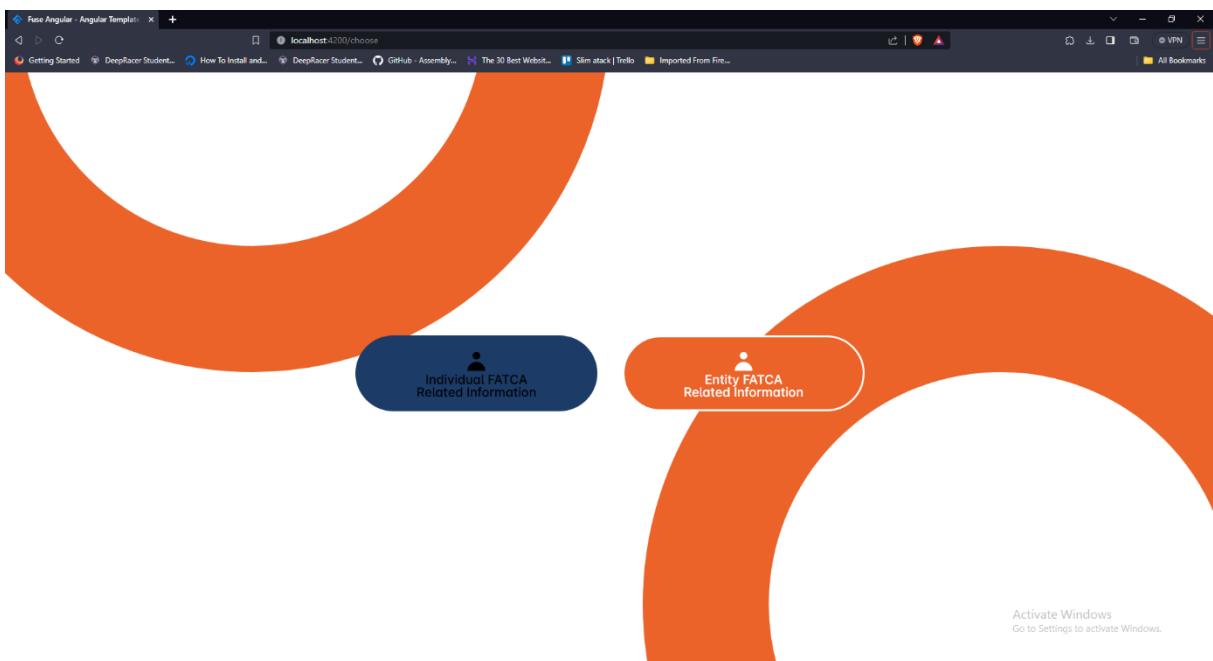


Figure 17:Individual or Entity FATCA Interface

Chapter 3: Sprint 1-Question Management

If the User choose Entity FATCA will get the interface below after that user will type the shareholders numbers after that will send them mails and answer the questions.

The screenshot shows a web browser window with a blue header bar containing the text "FATCA Related information". Below the header, there is a form with several questions and radio button options. At the bottom right of the form is a red "Submit" button. A watermark for "Activate Windows" is visible at the bottom right of the page.

Shareholder Number *	<input type="text"/>
The company is incorporated under US law/incorporated in the US	<input type="radio"/> Yes <input type="radio"/> No
The company has a TIN code	<input type="radio"/> Yes <input type="radio"/> No
Etablissement aux Etats Unis (sté mère américaine)	<input type="radio"/> Yes <input type="radio"/> No
Mailing address in the USA	<input type="radio"/> Yes <input type="radio"/> No
The capital of the company is held by one or more shareholders who are US taxpayers, holding at least 10% of the capital.	<input type="radio"/> Yes <input type="radio"/> No

Figure 18: Question Entity Interface

Else if the user chooses individual FATCA will get the interface below that user will answer questions and submit.

The screenshot shows a web browser window with a blue header bar containing the text "FATCA Related information". Below the header, there is a form with several questions and radio button options. At the bottom right of the form is a red "Submit" button. A watermark for "Activate Windows" is visible at the bottom right of the page.

Holding a TIN code	<input type="radio"/> Yes <input type="radio"/> No
Mailing address in the USA	<input type="radio"/> Yes <input type="radio"/> No
American telephone line	<input type="radio"/> Yes <input type="radio"/> No
Power of attorney for the benefit of a person having an address in the USA	<input type="radio"/> Yes <input type="radio"/> No
Standing instructions for transferring funds to an account in the USA	<input type="radio"/> Yes <input type="radio"/> No
adresse portant la mention « à l'attention de » ou « envoi à garder en instance » qui est l'unique adresse du titulaire du compte	<input type="radio"/> Yes <input type="radio"/> No
GREEN CARD ownership	<input type="radio"/> Yes <input type="radio"/> No
American Nationality	<input type="radio"/> Yes <input type="radio"/> No
American Residence	<input type="radio"/> Yes <input type="radio"/> No
Place of birth in the USA	<input type="radio"/> Yes <input type="radio"/> No

Figure 19: Question Individual Interface

Conclusion

In this chapter we present the first sprint through use case and several sequences diagram and scenarios and user interfaces from the project. In the next chapter we will present the next sprint which is focused on the Answer Management.

Chapter 4: Sprint 2-Answer Management

Introduction

This chapter is dedicated to the second sprint. We will be looking at the design of the answer management task and its implementation. To achieve this, we're going to break down the module into several simple titles.

4.1. Functional Specification

4.1.1. Sprint Backlog

Sprint Id	Sprint Name	Story Id	User Stories	Priority
1	Answer Management	2.1	As a user I can choose between FATCA or CRS	High
		2.2	As a user I can choose between Individual or Entity	High
		2.3	As a user I can answer the questions	High
		2.4	As a user I can submit my answers	High
		2.5	As an administrator I can consult the answers	Average
		2.6	As an administrator I can delete the answers	Average

Table 5: Sprint Backlog of Answer Management

4.1.2. Answer Management Use Diagram

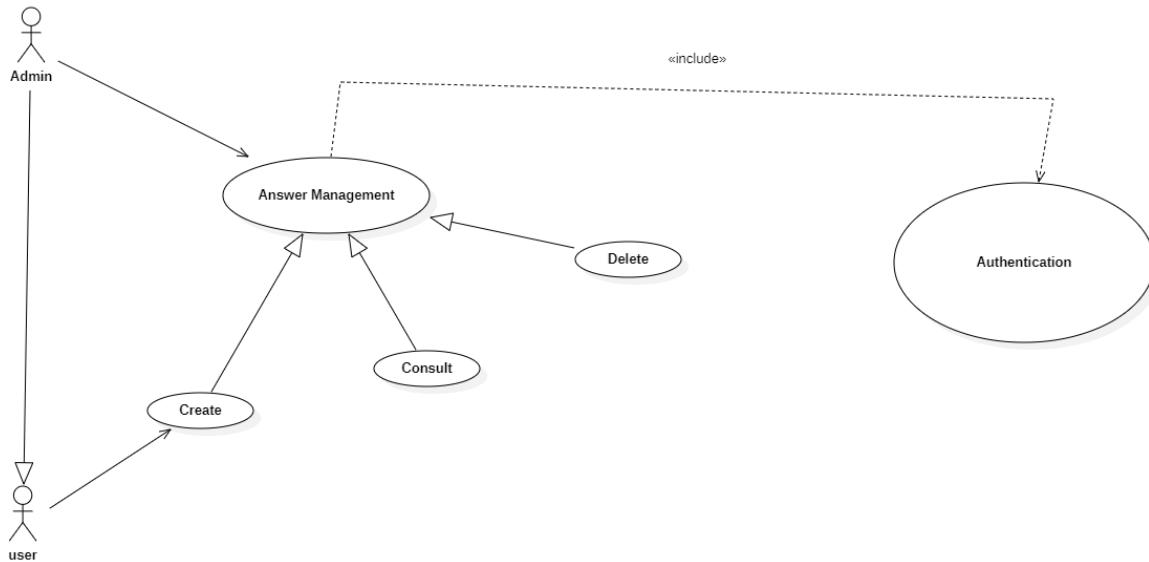


Figure 20: Answer Management Use Diagram

4.1.3. Add Answer Scenario

When the user submits answers, data will transfer to AnswerCommandController if data has successfully saved into MongoDB the AnswerCommandService will produce event to Kafka after that AnswerEventConsumerService will consume the event update PostgreSQL if not the Microservice error Message.

If the actor who is submitting the answer is an entity the platform will automatically ask for shareholders email and sent them emails to submit answers individually on the platform.

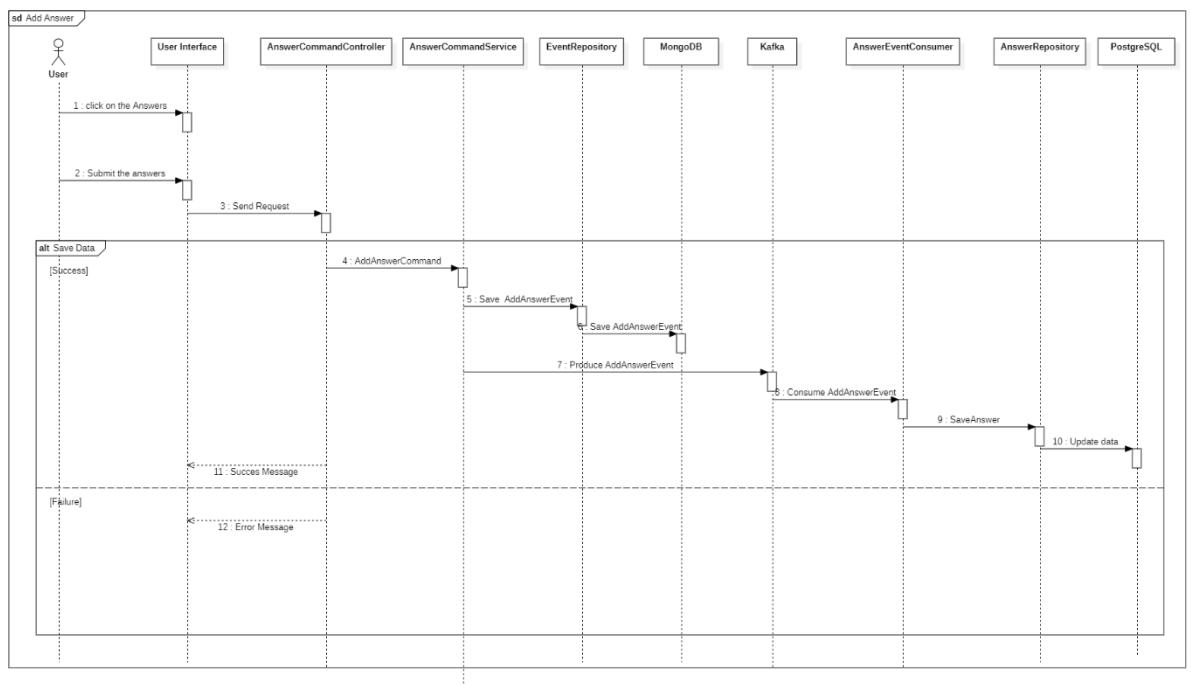


Figure 21: Add Answer Scenario

4.1.4. Delete Answer Scenario

When the admin click delete button it will send delete request if request pass successfully AnswerCommandController will produce an event to Kafka and save the database on MongoDB after that will successfully send messages for the user interface.

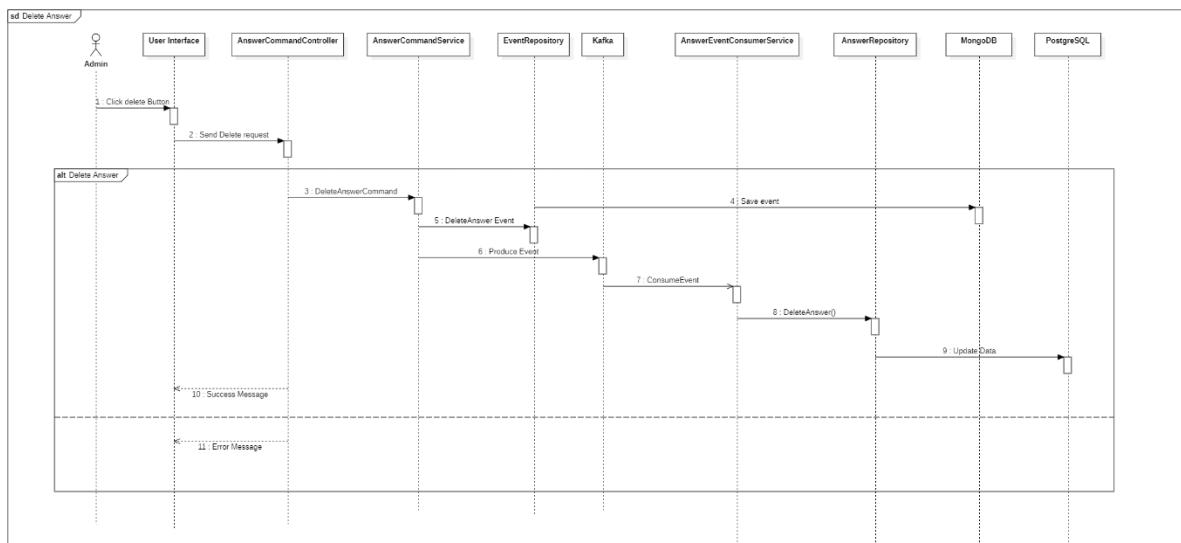


Figure 22:Delete Answer Scenario

4.1.5. Show all the Answers Scenario

When admin send on Answers button on the dashboard an Find All request will sent to AnswerQueryService after that will retrieve the data from PostgreSQL and sent the data and show to user.

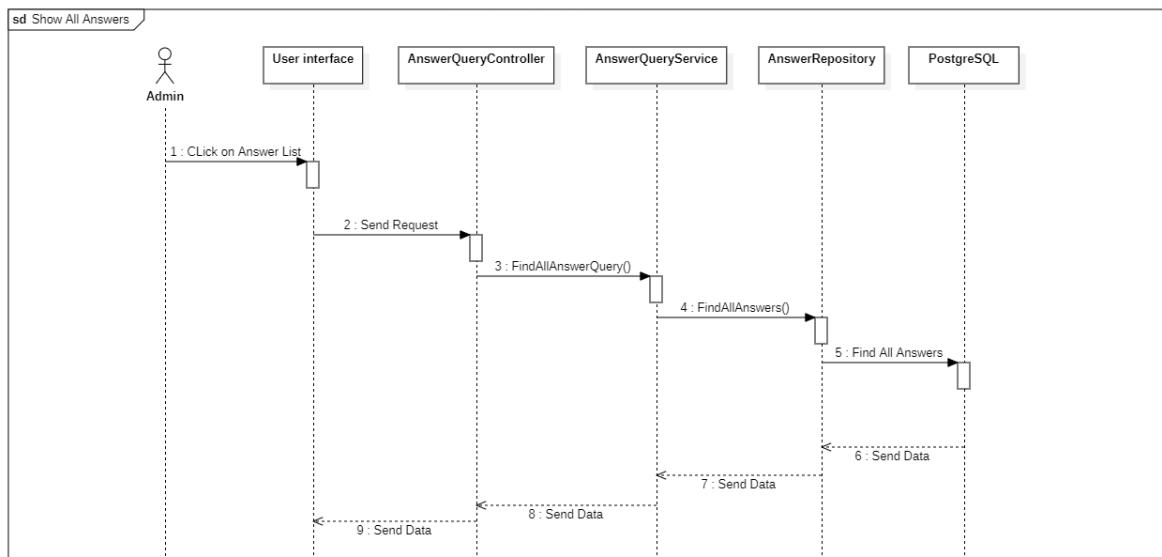


Figure 23:Show All Answers Scenario

4.2. Realisation

4.2.1. Answers Dashboard

On this interface will show the answer Dashboard interface that the admin consults the answers and delete answers.

No.	Question Content	Answer	User	Edit	Delete
fee251e7-d8fb-4ec1-8026-182796d5d401	The company is incorporated under US law/incorporated in the US	true		<button>Edit</button>	<button>Delete</button>
f92cc40e-1dce-4f41-9f09-66536a44c979	The company has a TIN code	true		<button>Edit</button>	<button>Delete</button>
5cced088-3c25-47ab-8826-fabbdfe4e466	Etablissement aux Etats Unis (siège mère américaine)	true		<button>Edit</button>	<button>Delete</button>
840da912-1c15-4c97-9c52-ff050094699	Mailing address in the USA	false		<button>Edit</button>	<button>Delete</button>
5830af82-3c21-4995-a7e0-7aa895b2a0f	The capital of the company is held by one or more shareholders who are US taxpayers, holding at least 10% of the capital.	false		<button>Edit</button>	<button>Delete</button>

The screenshot shows a web browser window displaying the 'Answer Dashboard'. On the left, there is a sidebar with a user profile picture of Brian Hughes and an email address (brian.brian@company.com). Below the profile, there is a list of navigation items: Question, Response (which is selected), fw9, fw8my, fw8xp, fw8bene, fw8ben, fw8reci, Crs Entity, Crs Individual, and Xmi Generation. At the bottom of the sidebar, there is a FUSE logo. The main content area shows a table with a list of responses. The table has columns for No., Question Content, Answer, User, Edit, and Delete. Each row contains a unique ID, a question text, a boolean value for 'Answer', a boolean value for 'User', and two buttons for 'Edit' and 'Delete'.

Figure 24:Answer Dashboard

4.2.2. Delete Answer

When the user click delete Answer will pop up a delete modal with delete confirmation message.

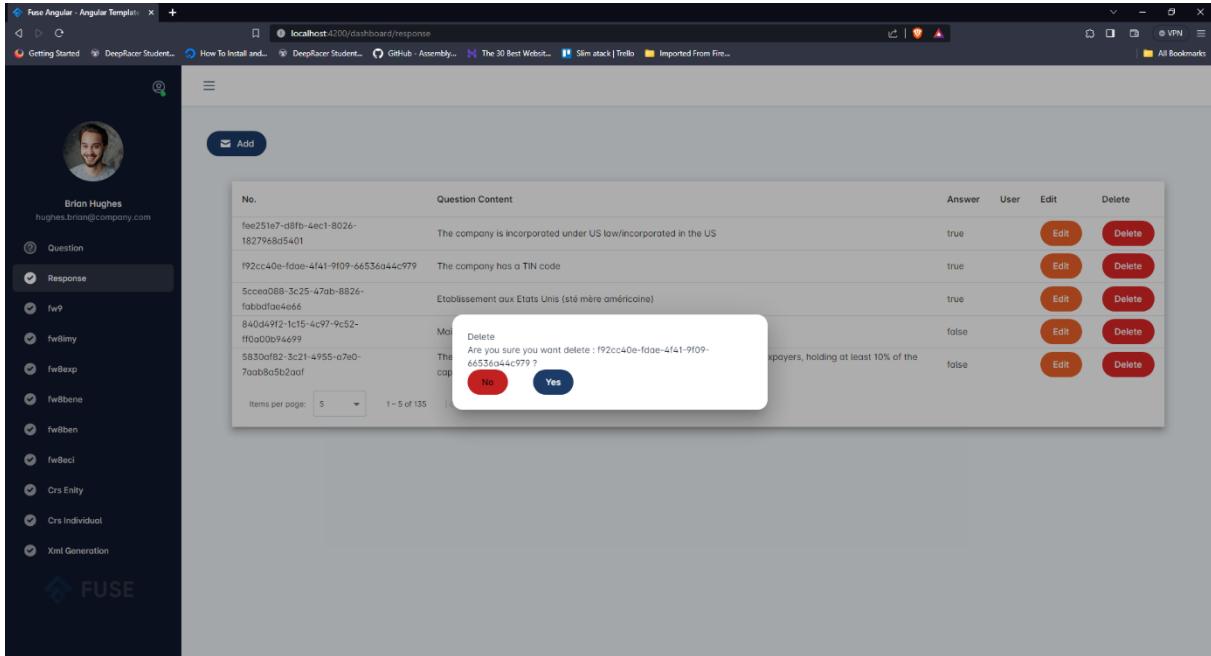


Figure 25:Delete Answer Interface

Conclusion

In this chapter, we presented the second chapter that speak about the answer management trough use case diagram, serval sequence diagram, scenario, and user interfaces from the project. In the next chapter we will present the next sprint which is focused on the Form Management.

Chapter 5: Sprint 3- Form Management

Introduction

In this chapter we will present the sprint 3 that specifically for form management we will go through the sprint backlog, some sequence diagrams and we class diagram that describes the form we are talking about.

5.1. Function Specification

5.1.1. Sprint Backlog

Sprint Id	Sprint Name	Story Id	User Stories	Priority
3	Form Management	3.1	As a user I get recommended a form by an algorithm base on user answers	High
		3.2	The user choose entity an email will send to shareholders to answer the question and upload the form suitable.	High
		3.3	As a user I can choose between the manual selection or the recommendation	Average
		3.4	As a user I can upload the form that have been fill out with user sign in	High
		3.5	As an administrator I can download the file that the user has been uploaded	High
		3.6	As an administrator I can delete the form that the user has been uploaded	High
		3.7	As an administrator I can modify the form that the user has been uploaded	High
		3.8	As an administrator I can verify the form that the user has been uploaded	High

Table 6:Sprint Backlog Form Management

5.1.2. Form management Use case Diagram

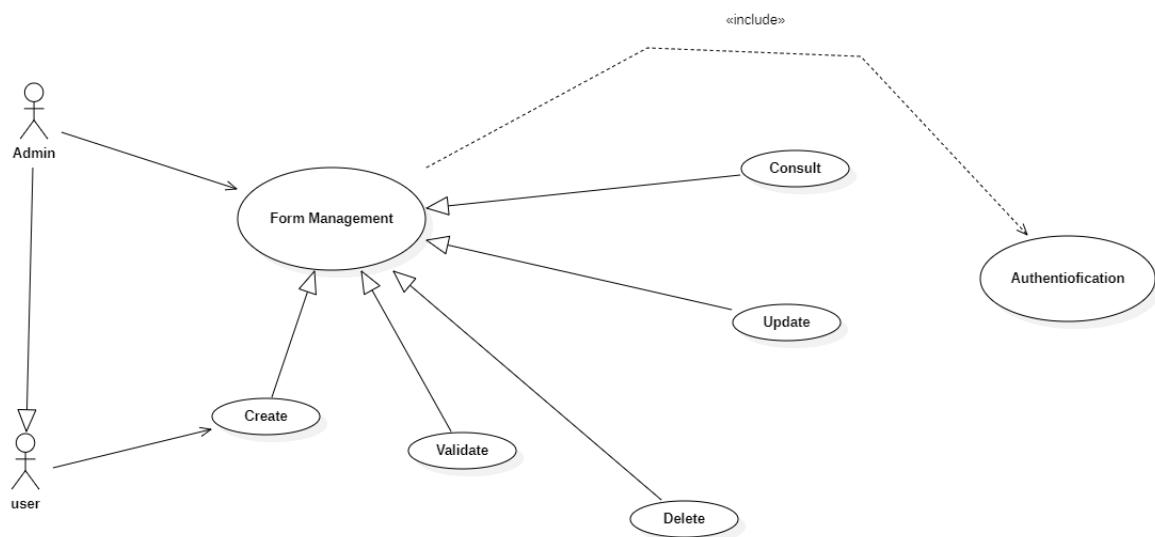


Figure 26: Form Use case Diagram

5.1.3. Class Diagram

Those classes are form we are going to talk about in next paragraphs all the next functions will be mentioned can be apply on one of those classes separately.

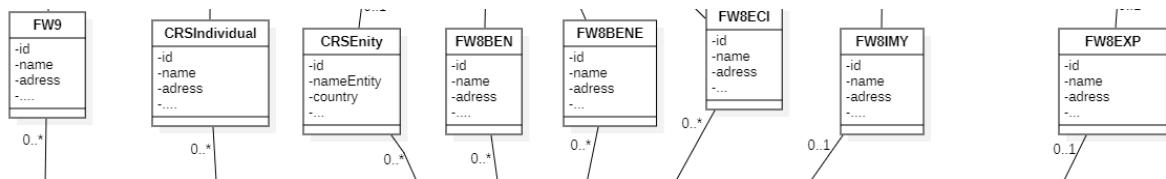


Figure 27: Form classes

5.1.4. Upload Form Scenario

When the User choose the form he will upload after answering the questions he will download the form and fill, sign it after that he will click upload and select the form he will be coded to base 64 and submitted a create form request to FormCommandController if the data have been successfully saved in the MongoDB FormCommandService will produce add Form Event and consume it the FormEventConsumerService and update PsogreSQL database and send success Message.

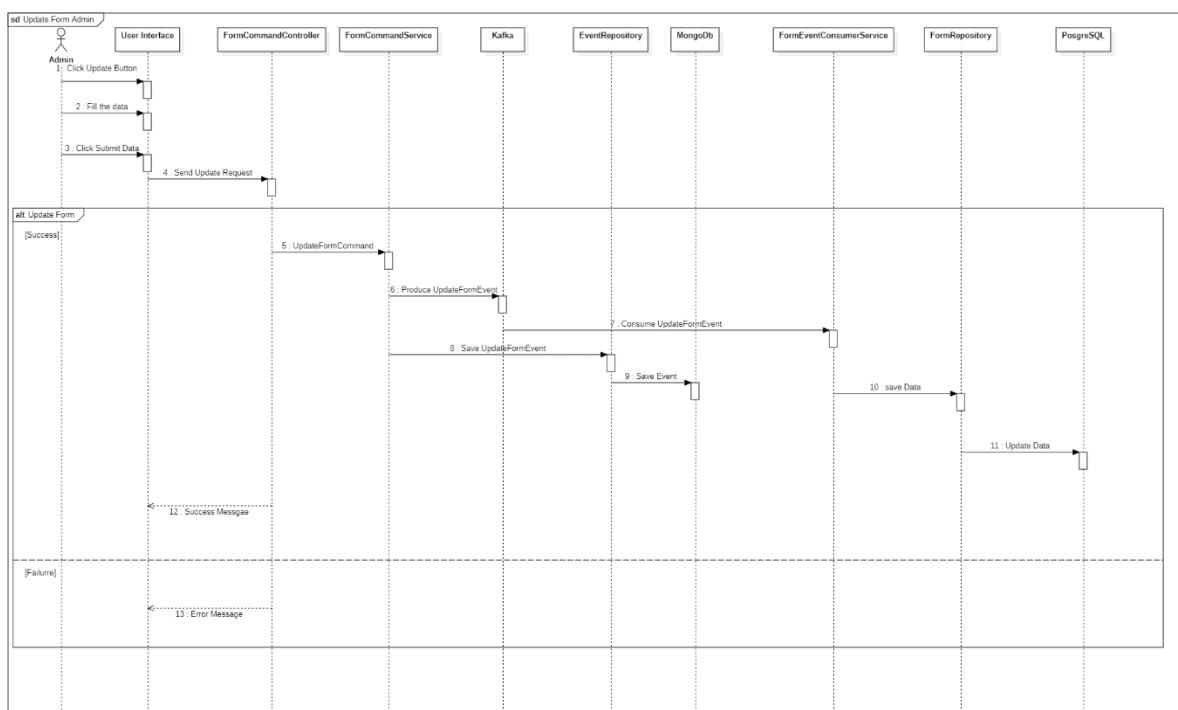


Figure 28:Upload Form Scenario

5.1.5. Download Form Scenario

After the user uploads the form into the platform the administrator will go to form and click the download button and send request to retrieve the form from the PostgreSQL and decode the form from base64 to pdf file and save in the device.

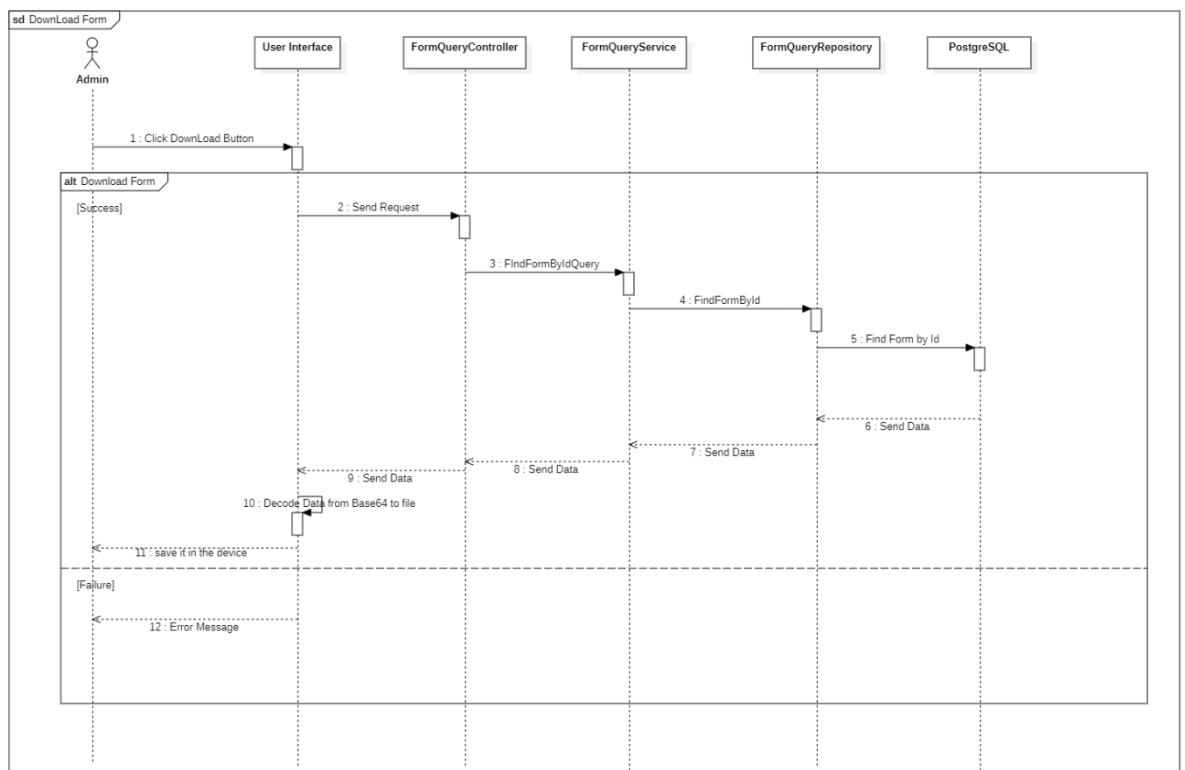


Figure 29: Download Form Scenario

5.1.6. Edit Form Scenario

We the admin click the edit button will be transferred to another interface to fill the data from pdf document that have been uploaded by user after submitting the edit request will be transfer to FormCommandController that will process the data and save in the MongoDB and produce event update from and consume it the FormEventConsumerService then update PostgreSQL database and sent success message. If the data didn't pass FormCommandController will send error message.

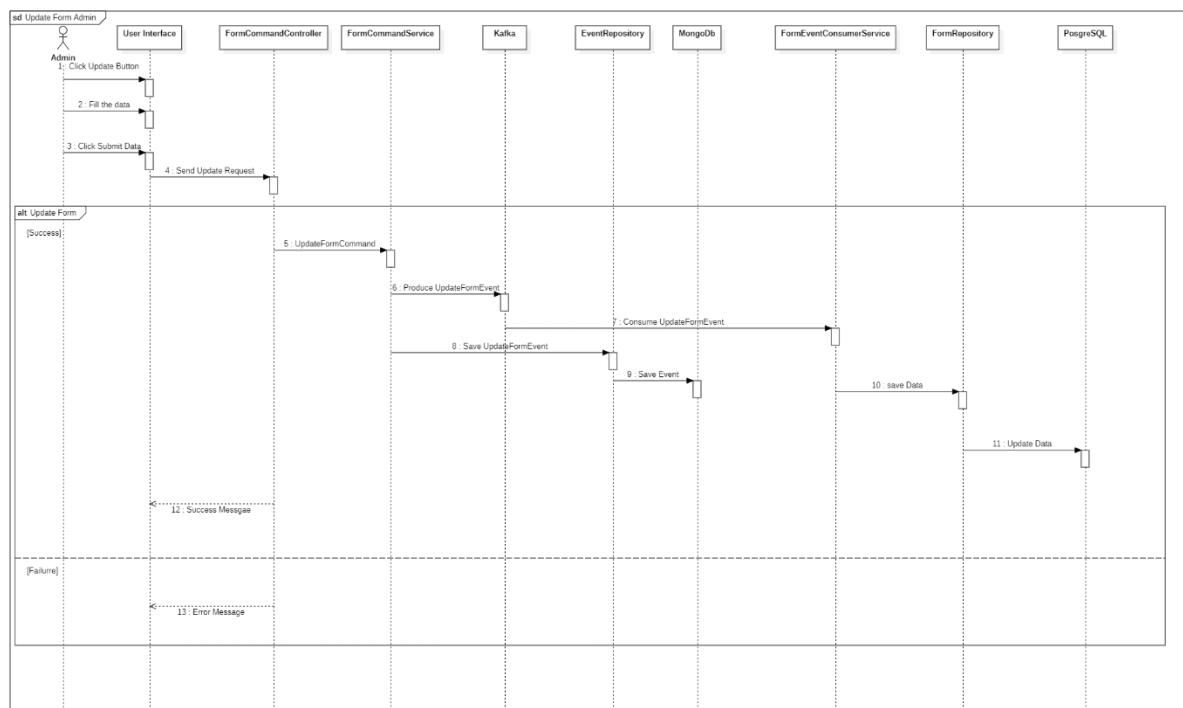


Figure 30:Edit Form Scenario

5.1.7. Delete Form Scenario

When the admin click delete button will send a delete request to FormCommandController that will save the data in MongoDB and produce a Delete form event to KAFKA Following this FormEventConsumerService will consume the delete form event and update PostgreSQL.

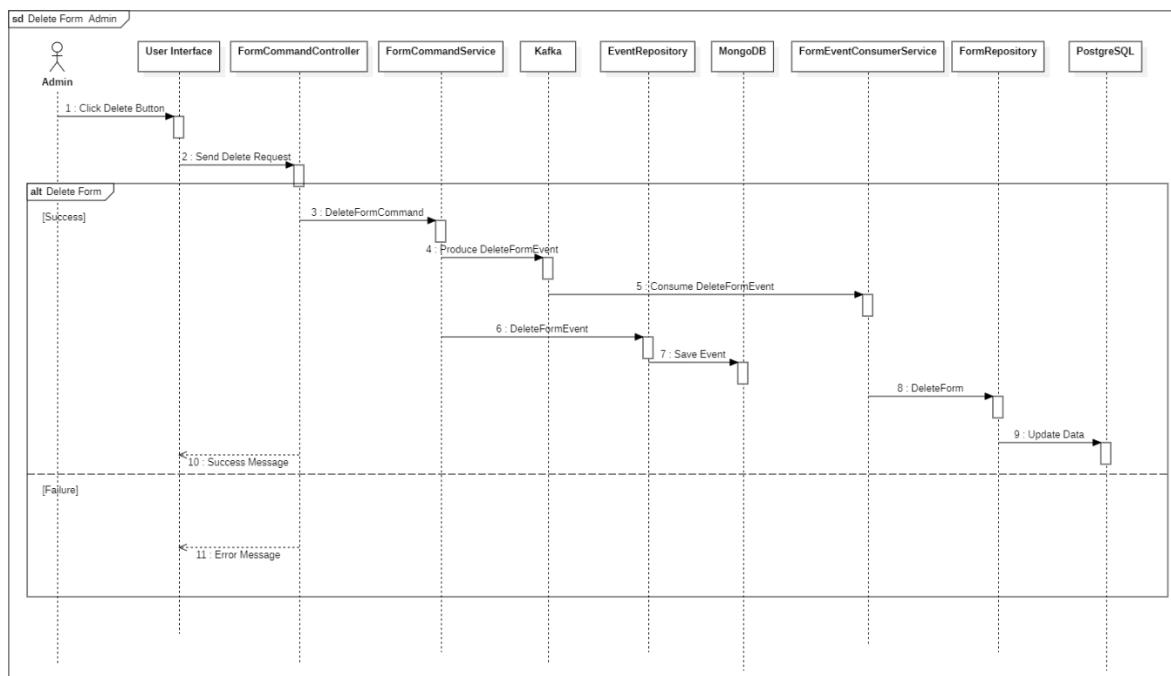


Figure 31:Delete Form Admin

5.1.8. Show all Forms Scenario

When admin click on one the forms in dashboard a find All request will be sent to Microservice-Query and retrieve the data from PostgreSQL and send it back to interface to show it. Else will be sent an error message.

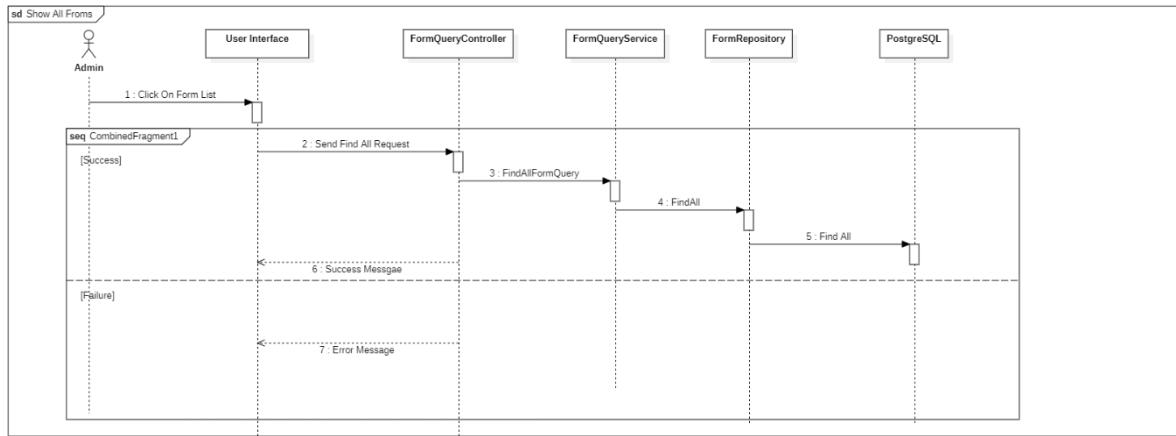


Figure 32:Show All Forms!

5.1.9. Verify Form Scenario

When admin clicks verify button will send a request for FormCommandController following that will save the data in MongoDB and produce verify form event to Kafka after that microservice-Query will consume it and update PostgreSQL if this process pass successfully will FormEventConsumerService will send success message to the admin otherwise will sent error message.

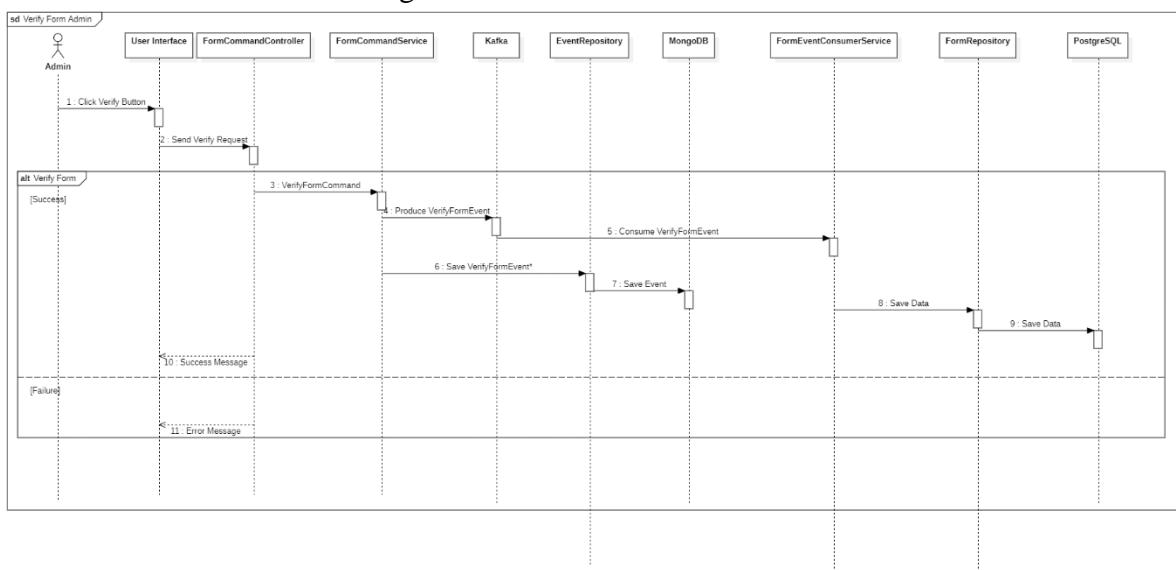


Figure 33:Verify Form Scenario

5.1.10. Auto recommendation form

After answering the question, the data will be sent to the FormCommandController and will run an algorithm that will identify the best form suitable for use and will save the auto recommendation form into MongoDB and produce auto recommendation form event to Kafka following that FormEventConsumerService will consume and update PostgreSQL database.

The algorithm we are using here will identify form base on the actor and he is answering score so based on that we are covering all the situations we have on this platform actor can be an entity or individual.

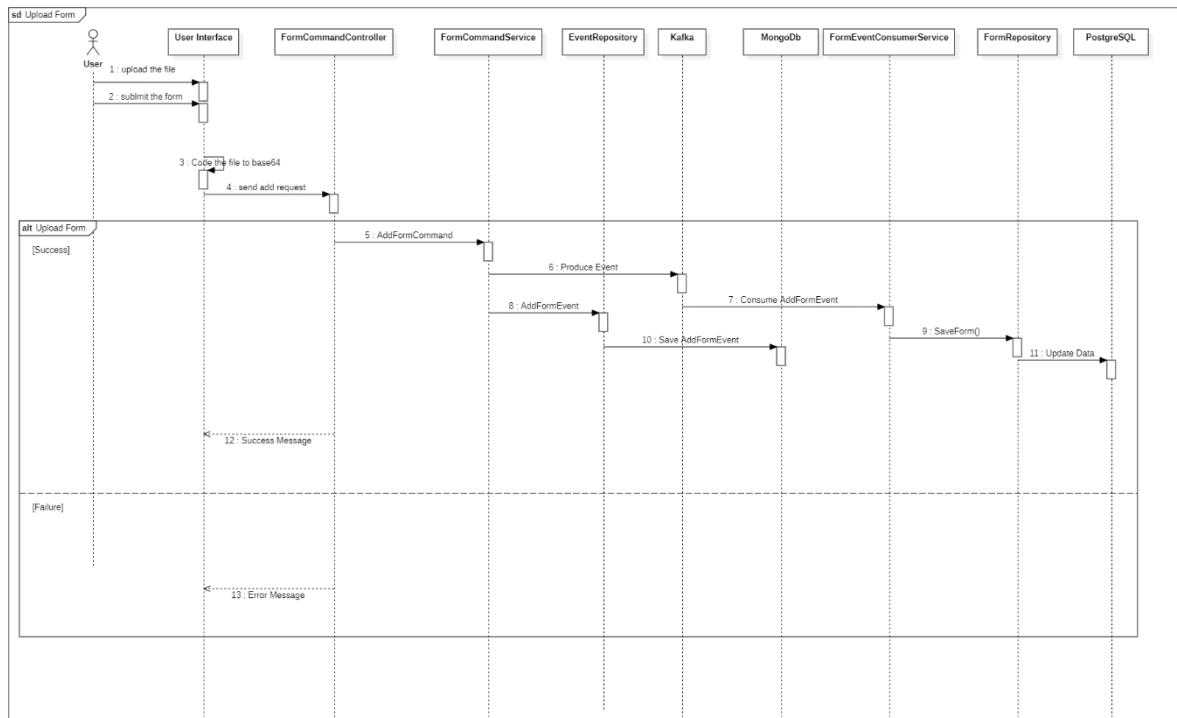


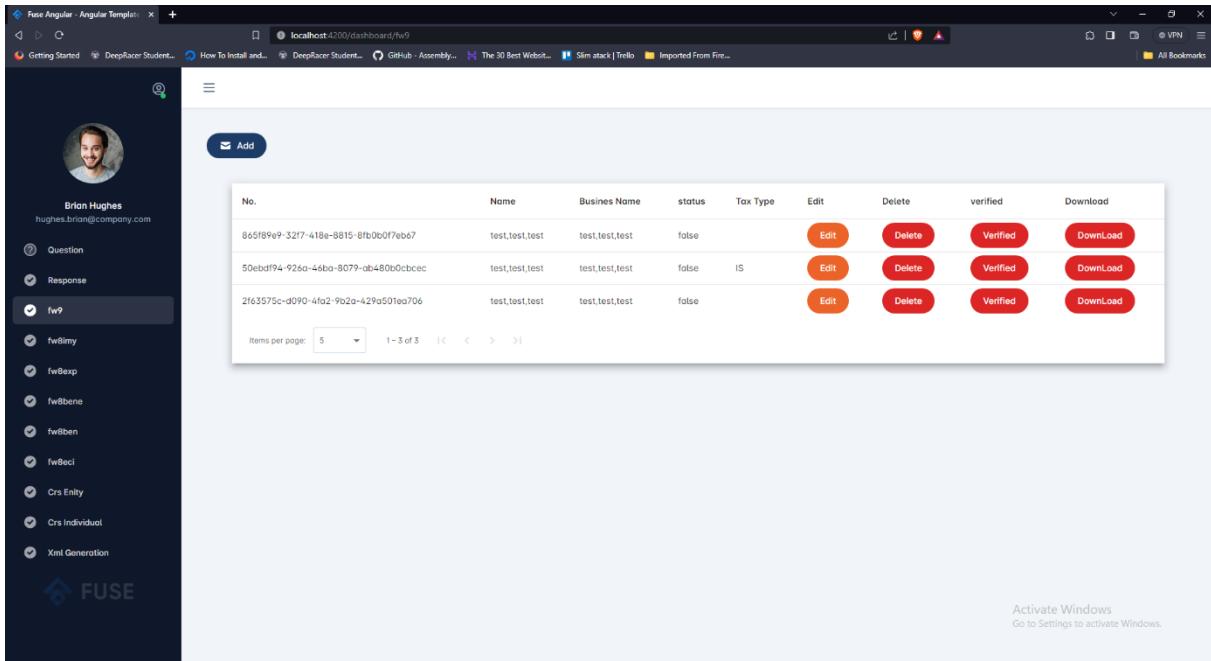
Figure 34:Auto Recommendation Form

5.2. Realisation

We will show one the forms that we worked on throughout the processes and its example from the forms we have for the remainder we have 6 different forms.

5.2.1. Form Dashboard

This interface FW9 Form dashboard through this we can manage the whole form we can edit, delete, verify, and download the form uploaded from the user.



The screenshot shows a web browser window titled "Fuse Angular - Angular Template". The address bar indicates the page is "localhost:4200/dashboard/fw9". The main content area displays a table of forms. The table has columns: No., Name, Business Name, status, Tax Type, Edit, Delete, Verified, and Download. There are three rows of data:

No.	Name	Business Name	status	Tax Type	Edit	Delete	Verified	Download
865f89e9-32f7-418e-8815-0fb0b0f7eb67	test,test,test	test,test,test	false		<button>Edit</button>	<button>Delete</button>	<button>Verified</button>	<button>DownLoad</button>
50ebd194-926a-45b0-8079-ab480b0cbcec	test,test,test	test,test,test	false	IS	<button>Edit</button>	<button>Delete</button>	<button>Verified</button>	<button>DownLoad</button>
2163575c-d090-4fa2-9b2a-429a501ea706	test,test,test	test,test,test	false		<button>Edit</button>	<button>Delete</button>	<button>Verified</button>	<button>DownLoad</button>

Below the table, there is a pagination control labeled "Items per page: 5" and a message "1–3 of 3". On the far right, there is a "Activate Windows" message: "Activate Windows Go to Settings to activate Windows." The left sidebar contains a user profile for "Brian Hughes" and a list of items including "Question", "Response", "fw9" (which is checked), "fw8lmy", "fw8esp", "fw8benie", "fw8ben", "fw8eci", "Crs Entity", "Crs Individual", and "Xml Generation". The FUSE logo is at the bottom of the sidebar.

Figure 35: FW9 dashboard interface

5.2.2. Edit FW9 Form Interface

On this interface we have the fw9 form interface that we can add the data from uploaded file that user have.

The screenshot shows the Fw9 form interface for the W-9 form. The title is "Request for Taxpayer Identification Number and Certification". The left panel shows the "W-9" logo and a note about the form's purpose. The middle panel contains fields for "Name" and "Business name". The right panel contains sections for "Exemptions" and "List account number(s) here (optional)". A note at the bottom right says "Give Form to the requester. Do not send to the IRS".

Figure 36:Fw9 form Interface

5.2.3. Delete FW9 Form Interface

When the admin click Delete button will pop up a confirmation modal.

The screenshot shows a list of Fw9 forms in a table. The columns include No., Name, Business Name, status, Tax Type, Edit, Delete, verified, and Download. A confirmation dialog box is overlaid on the table, asking "Are you sure you want delete : 2f63575c-d090-4f02-9b2a-429a501ea706 ?" with "Yes" and "No" buttons.

Figure 37:Delete Fw9 Form Interface

5.2.4. Verify Fw9 Form Interface

After adding the information of user and checking the data the administrator will click the verify button after that will pop up a confirmation modal of the verification.

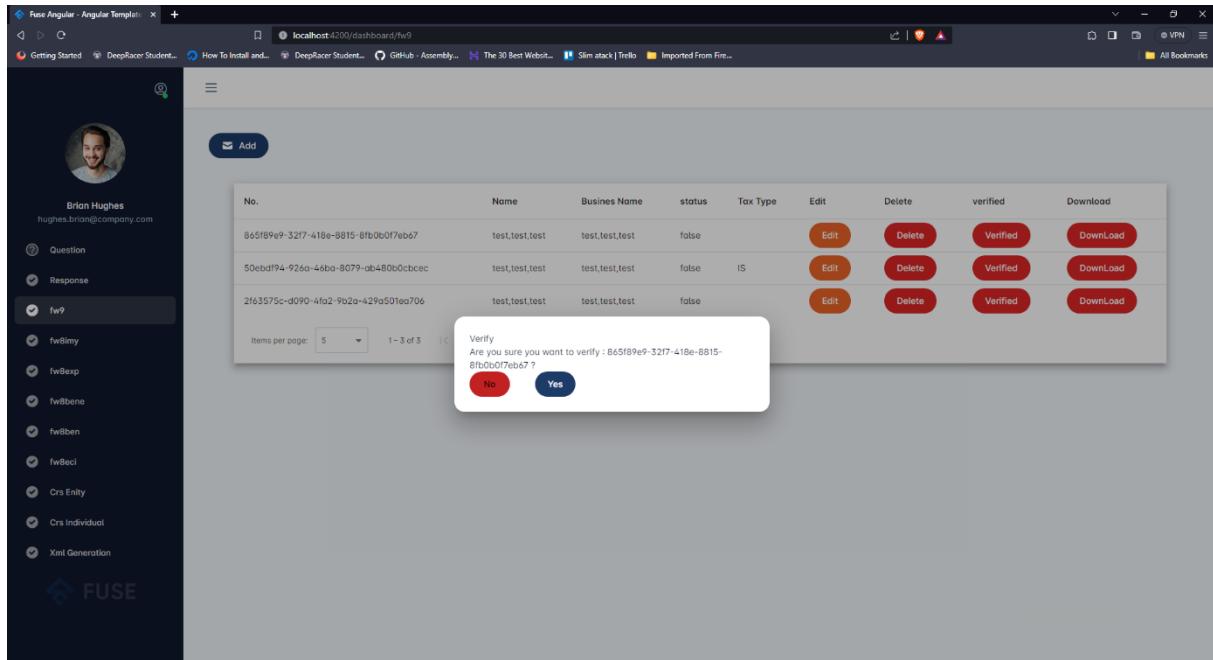


Figure 38: Verify Fw9 Form Interface

5.2.5. Auto recommendations from User side

After answering questions the identification algorithm will recommend for user a form base on he is answers.

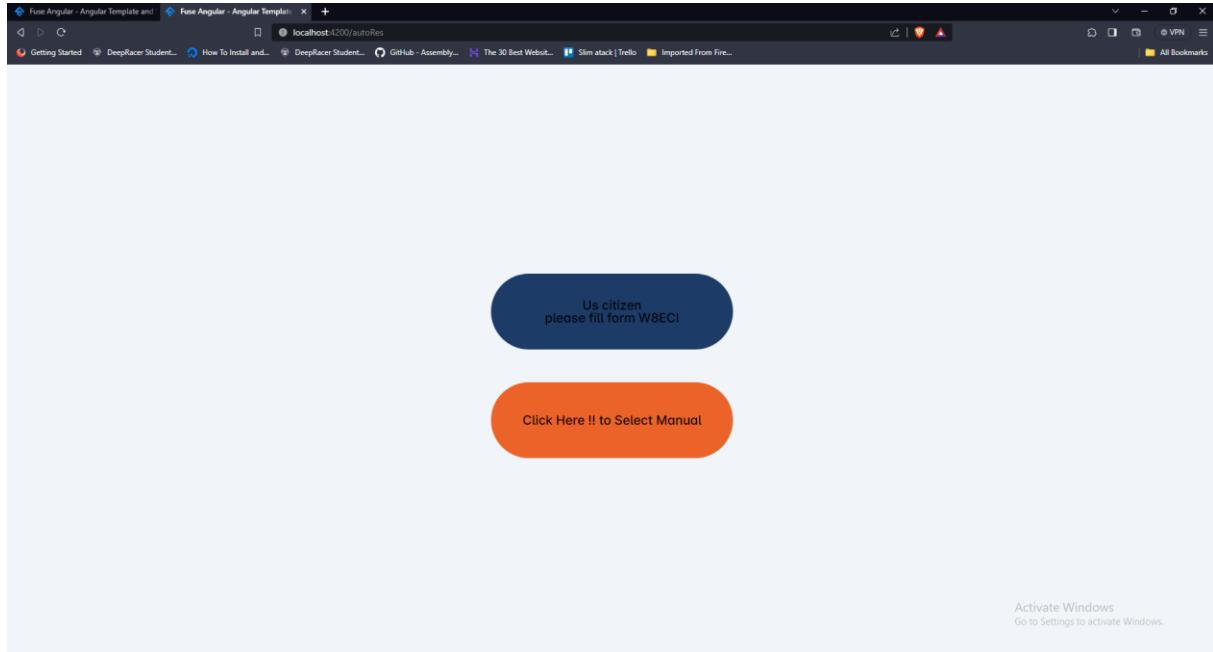


Figure 39: auto recommendations form User interface

5.2.6. manual selection form User Side

If user want to choose manually will have other form selection base on that will choose the form that he will upload.

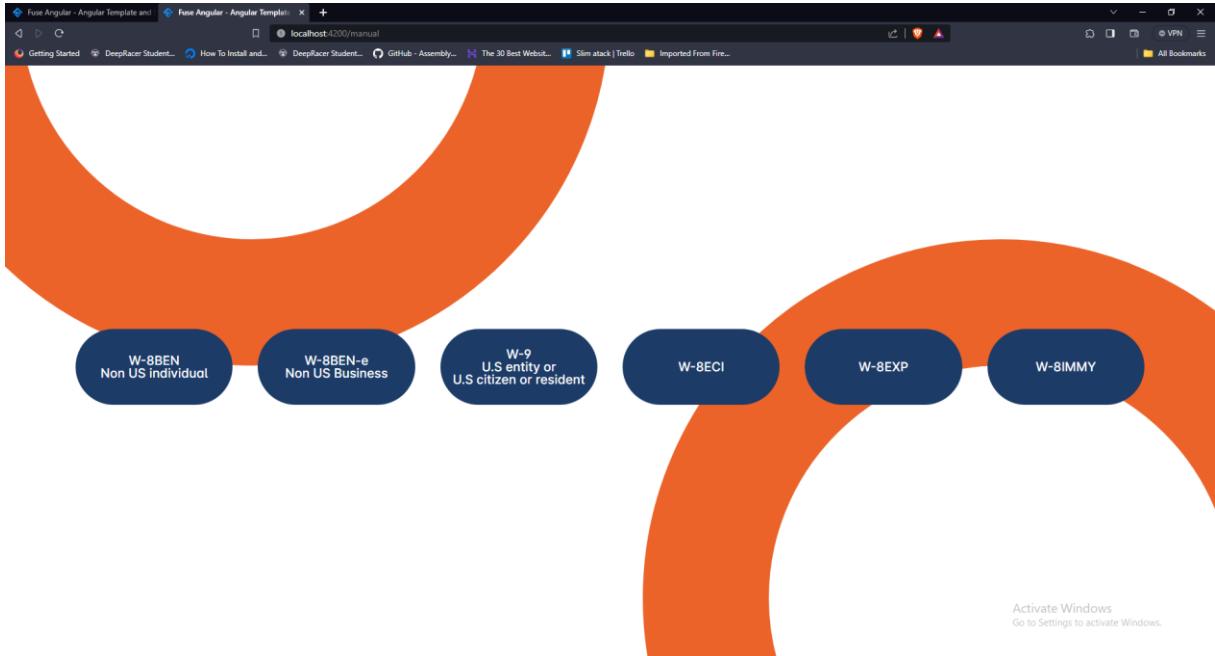


Figure 40:manual selection user interface

5.2.7. Upload file for user

After choosing the form user will upload the file and click upload to submit the form.

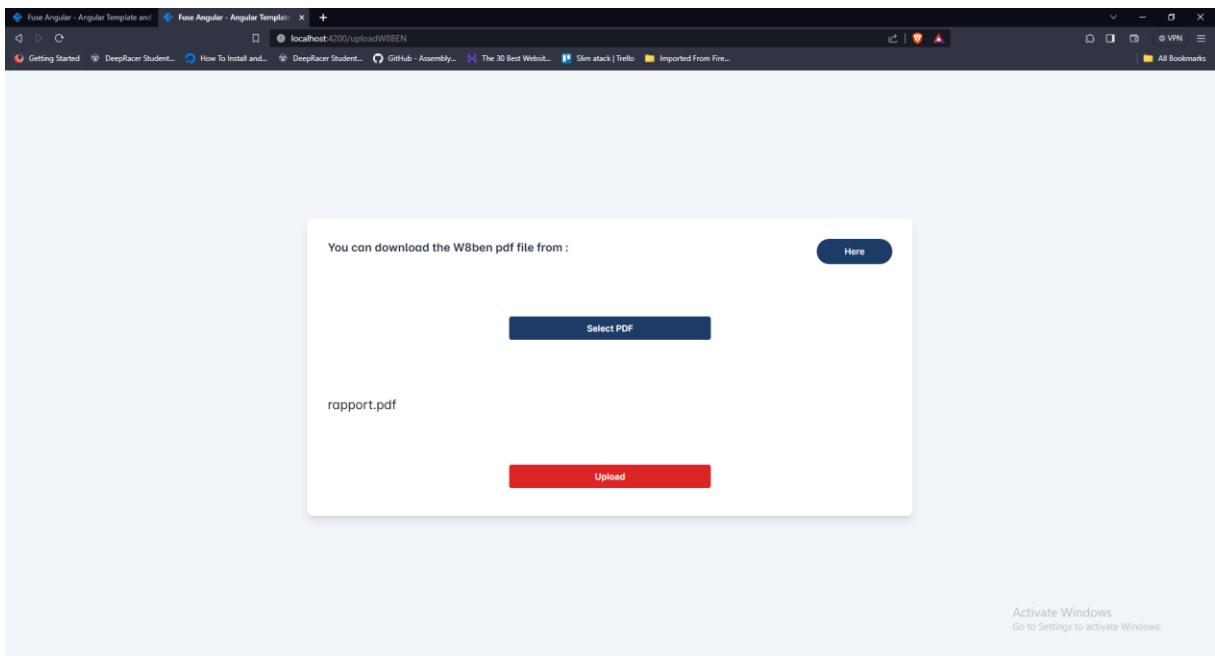


Figure 41:upload file for user interface

Conclusion

In this chapter, we presented the third chapter that speak about the Form management through use case diagram, serval sequence diagram, scenario, Class diagram and user interfaces from the project. In the next chapter we will present the next sprint which is focused on the XML Generation and Management.

Chapter 6: Sprint 4-Xml Generation and Management

Introduction

In this chapter we will present the sprint 4 that specifically for xml generation and management we will go through the sprint backlog, some sequence diagrams and we class diagram that describes the form we are talking about.

6.1. Functional Specification

6.1.1. Sprint Backlog

Sprint Id	Sprint Name	Story Id	User Stories	Priority
4	XML Generation and Management	4.1	As an administrator I can consult the xml reports	High
		4.2	As an administrator I can manual generate FATCA report	High
		4.3	As an administrator I can manually generate CRS report	High
		4.4	The platform can automatically generate XML FATCA report yearly	High
		4.5	The platform can automatically generate XML CRS report yearly	High

Table 7: Sprint Backlog XML Generation and management

6.1.2. Xml Generation and Management Use case

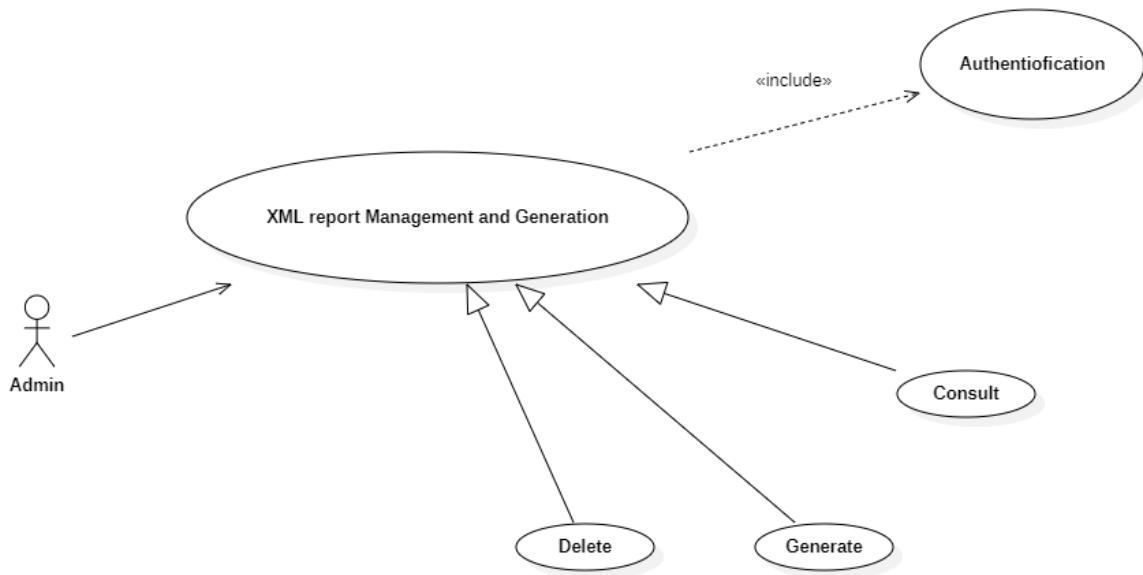


Figure 42: Xml Generation and Management Use case

6.1.3. Class diagram of FATCA xml report schema

Base on the IRS documentation we created java class that will deserialize the data from database to xml FATCA file on the figure right blew a class diagram describe whole schema base on rules that IRS put on.

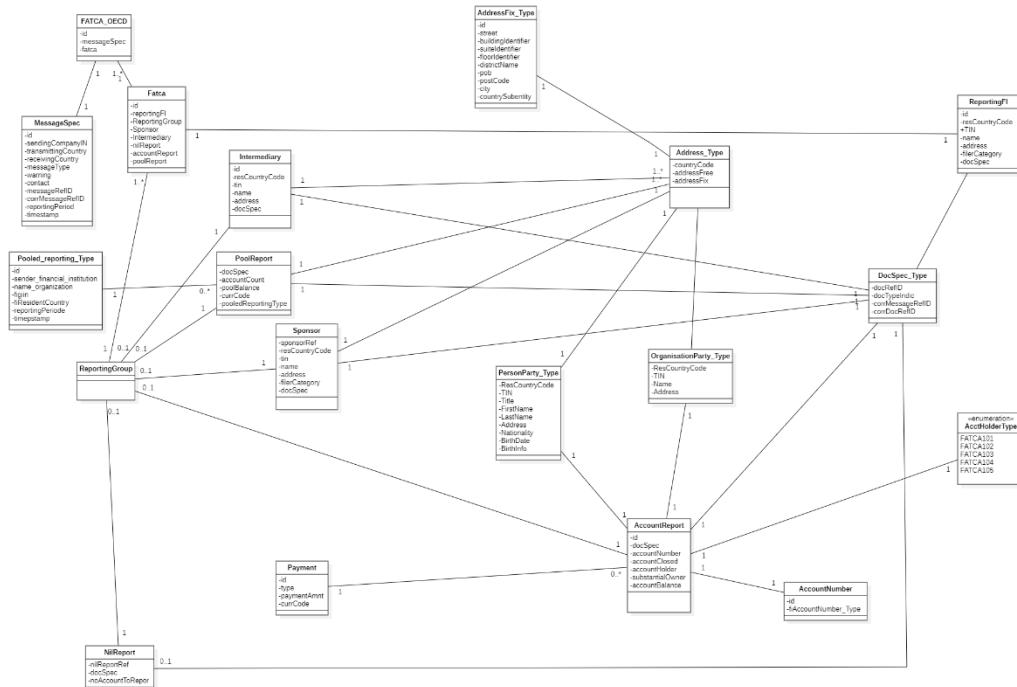


Figure 43: XML FATCA schema

6.1.4. Class diagram of CRS XML report schema

Base on the IRS documentation we created java class that will deserialize the data from database to xml CRS file on the figure right blew a class diagram describe whole schema base on rules that IRS put on.

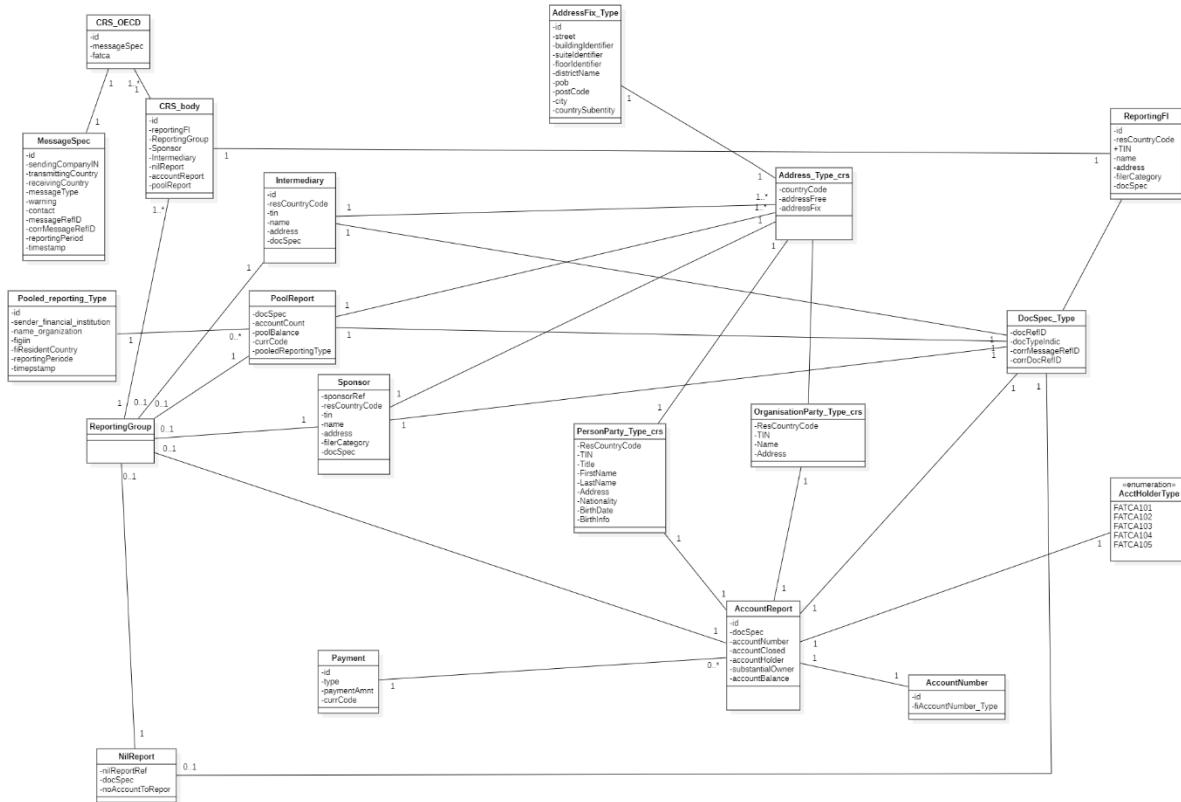


Figure 44:XML CRS schema

6.1.5. Auto Generation XML Report

On every first day on the year will be generated xml report from the data we have through the last year a request will send XmlCommandController will lunch a xml generation function what will take all the data and convert it to xml reports adhering to IRS schemas for FATCA or CRS and save the report on MongoDB and produce auto generation xml report event to Kafka and XmlEventConsumerService will consume the xml generation report event update the PostgreSQL database.

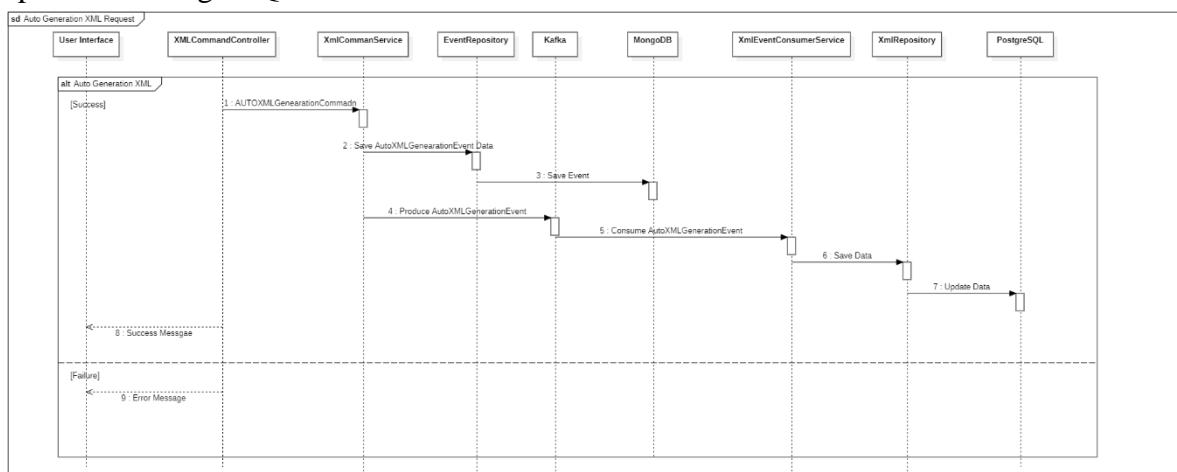


Figure 45:Auto Generation XML Report

6.1.6. Manual Generation XML Report

When the admin click on manual generation xml report it will sent request for XmlCommandController that will launch a xml generation function what will take all the data and convert it to xml reports adhering to IRS schemas for FATCA or CRS from the date that function to beginning of that year and save the report on MongoDB and produce auto generation xml report event to Kafka and XmlEventConsumerService will consume the xml generation report event update the PostgreSQL database.

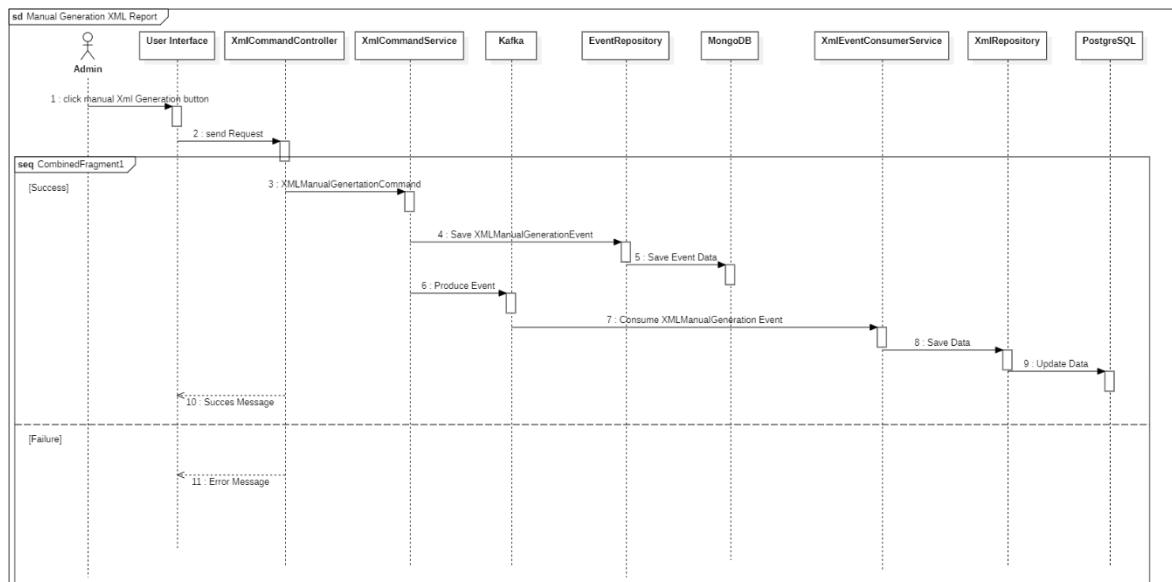


Figure 46:Manual Generation XML report

6.2. Realisation

6.2.1. XML Dashboard

On this figure we can see xml dashboard on this interface we manual generate FATCA or CRS XML code or delete the xml code or auto generate the xml code.

The screenshot shows a web-based XML dashboard. On the left, there is a sidebar with a user profile picture of Brian Hughes and a list of options: Question, Response, fw9, fw8my, fw8exp, fw8ben, fw8ben, fw8eci, Crs Entity, Crs Individual, and Xmt Generation. The Xmt Generation option is selected. The main area has tabs for 'Manual Generation Fata' and 'Manual Generation Crs'. Below these tabs is a table listing five XML report entries. Each entry includes a 'No.' column (e.g., b8605d4-657c-4ece-9208-fab005e06337), a 'Creation Date' column, a 'Type of Report' column, a 'Delete' button, and a 'View' button. At the bottom of the table, there is a pagination control showing 'Items per page: 5' and '1 - 5 of 61'. A footer at the bottom right says 'Activate Windows Go to Settings to activate Windows.'

Figure 47:XML Dashboard

6.2.2. XML view Interface

After generating the xml code we can observe the schema that we illustrate before on the code with rules of the IRS.

The screenshot shows a browser window displaying a large block of XML code. The XML is structured with various namespaces and elements, including ftc:MessageSpec, ftc:FATCA, ftc:ReportingGroups, ftc:AccountReport, ftc:AccountHolder, ftc:AccountReport, ftc:AccountHolder, ftc:AccountReport, ftc:AccountHolder, and ftc:AccountReport. The XML uses sfa:ResCountryCode, sfa:Name, sfa:Address, and sfa:AddressFree elements. A footer at the bottom right says 'Activate Windows Go to Settings to activate Windows.'

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ftc:FATCA_OEG_m2xsi><http://www.w3.org/2001/XMLSchema-instance> xsi:schemaLocation="urn:oecd:ties:fatca:v2" version="2.0" xmlns:ns2=<http://www.w3.org/2000/xmlns/>
<ftc:MessageSpec>
    <SendingCompanyIN>e1d3cc0-c7ae-4e38-b231-934e4c36ec92</SendingCompanyIN>
    <ReceivingCountry>US</ReceivingCountry>
    <MessageType>FATCA</MessageType>
    <MessageRefId>2b4c54cb-0da1-4a05-98e5-2ee89f460795</MessageRefId>
    <Timestamp>2023-12-05T13:43:37.363+01:00</Timestamp>
</ftc:MessageSpec>
<ftc:FATCA>
    <ftc:ReportingGroups>
        <ftc:AccountReport>
            <ftc:AccountHolder>
                <sfa:ResCountryCode>test,test,test</sfa:ResCountryCode>
                <sfa:Name>
                    <sfa:FirstName>test</sfa:FirstName>
                    <sfa:MiddleName>test</sfa:MiddleName>
                    <sfa:LastName>test</sfa:LastName>
                </sfa:Name>
                <sfa:Address>
                    <sfa:AddressFree>test,test,test</sfa:AddressFree>
                    <sfa:AddressFix>
                        <sfa:Street>test,test,test</sfa:Street>
                        <sfa:BuildingIdentifier>test</sfa:BuildingIdentifier>
                        <sfa:PostCode>test,test,test</sfa:PostCode>
                        <sfa:City>test,test,test</sfa:City>
                    </sfa:AddressFix>
                </sfa:Address>
            </ftc:AccountHolder>
        </ftc:AccountReport>
        <ftc:AccountReport>
            <ftc:AccountHolder>
                <sfa:ResCountryCode>test,test,test</sfa:ResCountryCode>
                <sfa:Name>
                    <sfa:FirstName>test</sfa:FirstName>
                    <sfa:MiddleName>test</sfa:MiddleName>
                    <sfa:LastName>test</sfa:LastName>
                </sfa:Name>
                <sfa:Address>
                    <sfa:AddressFree>test,test,test</sfa:AddressFree>
                    <sfa:AddressFix>
                        <sfa:Street>test,test,test</sfa:Street>
                        <sfa:BuildingIdentifier>test</sfa:BuildingIdentifier>
                        <sfa:PostCode>test,test,test</sfa:PostCode>
                        <sfa:City>test,test,test</sfa:City>
                </sfa:Address>
            </ftc:AccountHolder>
        </ftc:AccountReport>
    </ftc:ReportingGroups>
</ftc:FATCA>

```

Figure 48:XML view interface

Conclusion

In this chapter, we presented the third chapter that speak about the xml generation and management trough use case diagram, serval sequence diagram, scenario, Class diagram and user interfaces from the project.

General Conclusion

The modern banking system operates within a framework of regulatory compliance and customer due diligence to ensure financial transparency and security. Know Your Customer (KYC) guidelines form the cornerstone of this system, requiring banks to verify and authenticate the identity of their clients, assessing their risk profiles, and monitoring transactions to prevent money laundering and financial crimes. The Foreign Account Tax Compliance Act (FATCA) is a U.S. regulation that obliges foreign financial institutions to report information about accounts held by U.S. taxpayers to the Internal Revenue Service (IRS) to curb tax evasion. Similarly, the Common Reporting Standard (CRS) is an international standard developed by the Organisation for Economic Co-operation and Development (OECD) that requires participating countries to exchange financial information automatically to combat tax evasion and ensure tax compliance globally. Together, these regulatory frameworks establish protocols for banks to collect, maintain, and share crucial information, fostering greater transparency and accountability in the global financial system while addressing tax-related challenges across borders.

In this context, our project objectives to make the whole process of KYC digital and easy for the client and workers in the fields with providing Question management services that you can update the question regularly and answer Management services that will help client with he is answer to recommend the best form that he will suitable him, and Form management that will help administrator to manage all the form and verify the document have been uploaded and Xml generation and management service that will provide xml report on FATCA or CRS that IRS demand him every year.

Through this year I got familiar with the technologies that Hydatis have been provide it for solving this problem which gave me opportunity to learn more about the fields and practices what I learned before and see it in real life. For the project still in the beginning of the development will need more work on it to add more reliable feature like online sign in so we make it all process digitally or make a good script function that can scrap the whole information form the document that the client can provide.

Bibliography

- 1: <https://scrumguides.org/scrum-guide.html>
- 2: <https://www.techopedia.com/definition/3927/java>
- 3: https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm
- 4: https://en.wikipedia.org/wiki/Apache_Kafka
- 5: <https://angular.io/docs>
- 6: <https://www.eventstore.com/cqrs-pattern>
- 7: <https://www.eventstore.com/cqrs-pattern>
- 8: <https://aws.amazon.com/rds/postgresql/what-is-postgresql/#:~:text=PostgreSQL%20is%20used%20as%20the,%2C%20geospatial%2C%20analytics%20applications.>
- 9: <https://en.wikipedia.org/wiki/MongoDB>
- 10: https://en.wikipedia.org/wiki/Microsoft_Azure



ESPRIT SCHOOL OF ENGINEERING

www.esprit.tn - E-mail : contact@esprit.tn

Siège Social : 18 rue de l'Usine - Charguia II - 2035 - Tél. : +216 71 941 541 - Fax. : +216 71 941 889

Annexe : Z.I. Chotrana II - B.P. 160 - 2083 - Pôle Technologique - El Ghazala - Tél. : +216 70 685 685 - Fax. : +216 70 685 454