Republic of Tunisia
The Ministry of Higher Education and Scientific Research

The University of Carthage

The Higher Institute of Information and Communication Technologies

MINI PROJECT REPORT

Submitted in Partial Fulfillment of the Requirements for the
Bachelor Degree in Computer Science

Field of Study : Software Engineering and Information Systems

FlexFit : workouts and diets managments

By
Oussema, Eya, Azza

Conducted within ISTIC

Publicly defended on May 25, 2024 in front of the jury members :
Academic Supervisor :
Wafa Tbourski

Academic Year : 2023-2024

Table des matières

Introduction Générale **1**

1 Requirements specification **2**
1.1 Introduction 2
1.2 Requirements Specification 2
1.2.1 Functional requirements 2
1.2.2 Non-Functional requirements 2
1.2.3 Actors Identification 4
1.3 Technological choices 5
1.3.1 Backend 5
1.3.2 Frontend 5
1.4 Global use case diagram 6
1.5 Product backlog 8
1.6 Conclusion 8

2 Identification of Backlog for Release 1 Sprint 0 **9**
2.1 Introduction 9
2.2 Sprint 0 9
2.2.1 Identifying the sprint0 backlog 9
2.2.2 Refinement of sprint 0 9
2.3 Conclusion 18

3 Identification of Backlog for Release 2 Sprint 1 **19**
3.1 Introduction 19
3.2 Sprint 1 19
3.2.1 Identifying the sprint1 backlog 19
3.2.2 Refinement of sprint 1 20

Conclusion Générale **34**

Bibliographie **35**

Table des figures

1.1 PHP 5
1.2 Angular 5
1.3 HTML5 6
1.4 CSS3 6
1.5 Visual Studio Code 6
1.6 Git 6
1.7 global use case diagram 7

2.1 Authentication use case diagram 11
2.2 Authenticate class Diagram 11
2.3 Authentication sequence diagram 12
2.4 Authentication Screenshot 13
2.5 Authentication Screenshot 13
2.6 ”Register” use case diagram 15
2.7 Register class Diagram 15
2.8 Register sequence diagram 16
2.9 Register ScreenShot 17
2.10 Register ScreenShot 17

3.1 user case diagram of story update information 21
3.2 update information class Diagram 22
3.3 update information sequence Diagram 23
3.4 update information screenshot 24
3.5 user case diagram of story follow Subscriber 26
3.6 Consult subscriber class Diagram 28
3.7 Consult subscriber class Diagram 29
3.8 consult subscriber Screenshot 30
3.9 Delete subscriber class Diagram 32
3.10 Delete And Consult subscriber Screenshot 33

Liste des tableaux

2.1 Detailed description of the ”Authentication” use case 11
2.2 Detailed description of the ”Register” use case 15

3.1 Detailed description of the ”Update Information” use case 21
3.2 Detailed description of the ”Delete Subscriber” use case 32

Introduction Générale

The Minister of Youth and Sport, Kamel Deguiche, stated this Thursday, October 20,2022, that 83% of Tunisians do not practice any sporting activity and that walking is the second favourite sport in Tunisia after football.

this is an alarming number that requires great attention from the population. Sports play a major role in battling several death-threatening diseases due to their major benefits including :
weight-gain-control, lower-cholesterol-Levels, improving blood circulation. Proving its effectiveness in battling common major death-threatening diseases in Tunisia related to heart diseases and diabetes.

So with our application, we aim to reduce this number and help clients easily tackle these issues by providing a workout plan and diet specific for each individual, considering their budget, goals, and physical health state.
During the developing phase of our project, we pursued Scrum as a working methodology and therefore had an aspectual approach by splitting the project into the following four chapters :

— The first chapter entitled as "Project Context" shows the exploratory study.

— The second chapter entitled as "Requirements Specification" explains the requirement specification.

— The third chapter entitled as "Release 1 : Implementation of the Project Management" demonstrates the implementation of the project management on our customer relationship management platform.

— The fourth chapter entitled as Project Follow-Up "Release 2 : Implementation of the Project follow-up" demonstrates the implementation of the project follow-up on our customer relationship management platform.

Eventually, we finalize our report with a general conclusion and a viewpoint of the project and beyond.

Chapitre 1

Requirements specification

1.1 Introduction

In this chapter, we focus on the users' needs addressed in our project through functional and non-functional specifications, as well as the external entities that will interact with the system. Finally, the global use case diagram is presented to deliver a high-quality application that meets the client's requirements

1.2 Requirements Specification

Requirements' specification establishes the main foundation for the product development. It defines how a developed system is set to function, as well as the constraints to be implemented on its design. Laying out this foundation on solid bases guarantees that the team working on the project development creates a product satisfying the customers' needs. Therefore, it characterizes the following functional and non-functional requirements :

1.2.1 Functional requirements

Functional requirements, known also as the functional specification, are a collection of requirements that outlines the main goal of the system and the purpose of its offerings to all users. It features the systems functionalities that the team must fulfil during the process of development, what helps them to keep track of their progress.

— Consult Sport
— Authenticate
— Register
— save personal information of sport
— update personal information
— follow subscriber information
— manage sport

1.2.2 Non-Functional requirements

The non-functional requirements describe the developed system qualities when it is performing one of its use cases. In fact, these requirements are more of a set that defines the manners to hold the functional ones on a line that is bound to the software system

judgements followed by its security, portability, usability etc. . . . Therefore, the system won't meet the expectations of neither the owner nor the user unless these requirements are fulfilled. Yet, even if the require- ments are not respected, the system would still work.

— Security
— Usability
— Ergonomics
— Maintainability
— Extensibility
— Modularity
— manage sport

Progressive Web Apps (PWAs) possess several characteristics that distinguish them from traditional web applications. Here are some key features :

— **Progressive Enhancement** :

  PWAs are built with progressive enhancement in mind. This means they are designed to work for every user regardless of browser choice because they're built with progressive enhancement as a core tenet.

— **Responsiveness** :

  PWAs are responsive and adapt to various screen sizes and devices, providing a seamless user experience across desktop, mobile, and tablet devices.

— **Connectivity Independence** :

  PWAs can function even in low or no internet connectivity environments through features like service workers, enabling offline access to cached content and data.

— **App-like Experience** :

  PWAs offer an app-like experience to users, with features such as push notifications, home screen icons, and fullscreen mode, blurring the lines between web and native applications.

— **Secure** :

  PWAs are served over HTTPS, ensuring a secure connection between the user and the application, which is crucial for data integrity and user privacy.

— **Discoverability** :

  PWAs are discoverable by search engines and can be indexed, allowing users to find them through traditional web searches.

— **Installability** :

  PWAs can be installed by users directly from the browser without the need to visit an app store. Once installed, they appear on the user's home screen or app launcher, just like native applications.

— **Up-to-date Content** :

PWAs can update themselves automatically in the background, ensuring users always have access to the latest content and features without requiring manual updates.

— **Engagement** :

PWAs leverage features such as push notifications to engage users and keep them informed about new content, updates, or relevant events.

— **Offline Capabilities** :

PWAs can cache content and data locally using service workers, allowing users to access certain features and content even when they are offline or experiencing a poor network connection.
Overall, PWAs combine the best features of web and native applications, offering users a fast, reliable, and engaging experience across various devices and network conditions.

1.2.3 Actors Identification

An actor is an individual or entity involved in or affected by the action or project in question. Therefore, it is necessary to start by clearly specifying with respect to which action or sequence of actions we are seeking to determine who the actors are and what they are. Our platform contains four actors who interact directly with the system :

— **Visitor** : Any personnel accessing the FlexFit platform has the ability to navigate through the platform and consult existing workouts and diets.
— **Subscriber** : Any person accessing the FlexFit platform has the possibility to record their sports-related data, request their BMI, update their information, and receive a workout and diet plan tailored to their specific data entered.
— **Admin** :It is the supervisor who possesses all control permissions to manage the workouts and diets based on the data entered by the subscriber. Additionally, they monitor this data along with the corresponding BMI feedback from the subscribers.

Actor	Role
Administrator	— Authenticate — Track subscriber data — Manage sports.
Subscriber	— Register — Authenticate — Record sports data. — Update informations
Invité	— Consulter Workout and Diet

Description detaillé des acteurs

1.3 Technological choices

For this part, we will be presenting our technological choices that we thought about and followed during the process of developing the system. These technological choices can be classed into front-end technologies and back-end technologies.

1.3.1 Backend

PHP : PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP : Hypertext Preprocessor. It is widely used for creating dynamic web pages and can be embedded into HTML. PHP code is executed on the server, generating HTML content that is then sent to the client's web browser

FIGURE 1.1 – PHP

1.3.2 Frontend

Angular : Angular is an open-source client-side JavaScript framework that enables the development of single-page applications. It is based on the concept of the MVC (Model View Controller) architecture to separate data, views, and various actions that can be performed. The source code of Angular is written in TypeScript.

FIGURE 1.2 – Angular

HTML5 : HTML5 is the abbreviation of "Hypertext Markup Language," and it comprises markup symbols or codes inserted into a file intended to be displayed on a World Wide Web browser page. The markup instructs the web browser on how to display the words and images of a web page to the user. Each individual code is called an element.

CSS3 : The abbreviation of "Cascading Style Sheet" is CSS, which is used to format the layout of web pages. They can be used to define text styles, table sizes, and other aspects of web pages that previously could only be defined in the HTML code of a page. CSS helps web developers create a consistent look across multiple pages of a website.

FIGURE 1.3 – HTML5

FIGURE 1.4 – CSS3

Visual Studio Code : A lightweight yet powerful source code editor that runs on your desktop and is available for Windows, macOS, and Linux is Visual Studio Code. It comes with built-in support for JavaScript, TypeScript, and Node.js and has a rich ecosystem of extensions for other languages (such as C++, Java, Python, PHP, Go) and runtime engines (such as .NET and Unity).

FIGURE 1.5 – Visual Studio Code

Git : It is a decentralized version control system. It was created by Linus Torvalds, who is also the creator of the Linux kernel. This project is licensed under GPL and primarily developed in C, with some Shell and Perl as well.

FIGURE 1.6 – Git

1.4 Global use case diagram

this Figure 1.7 shows

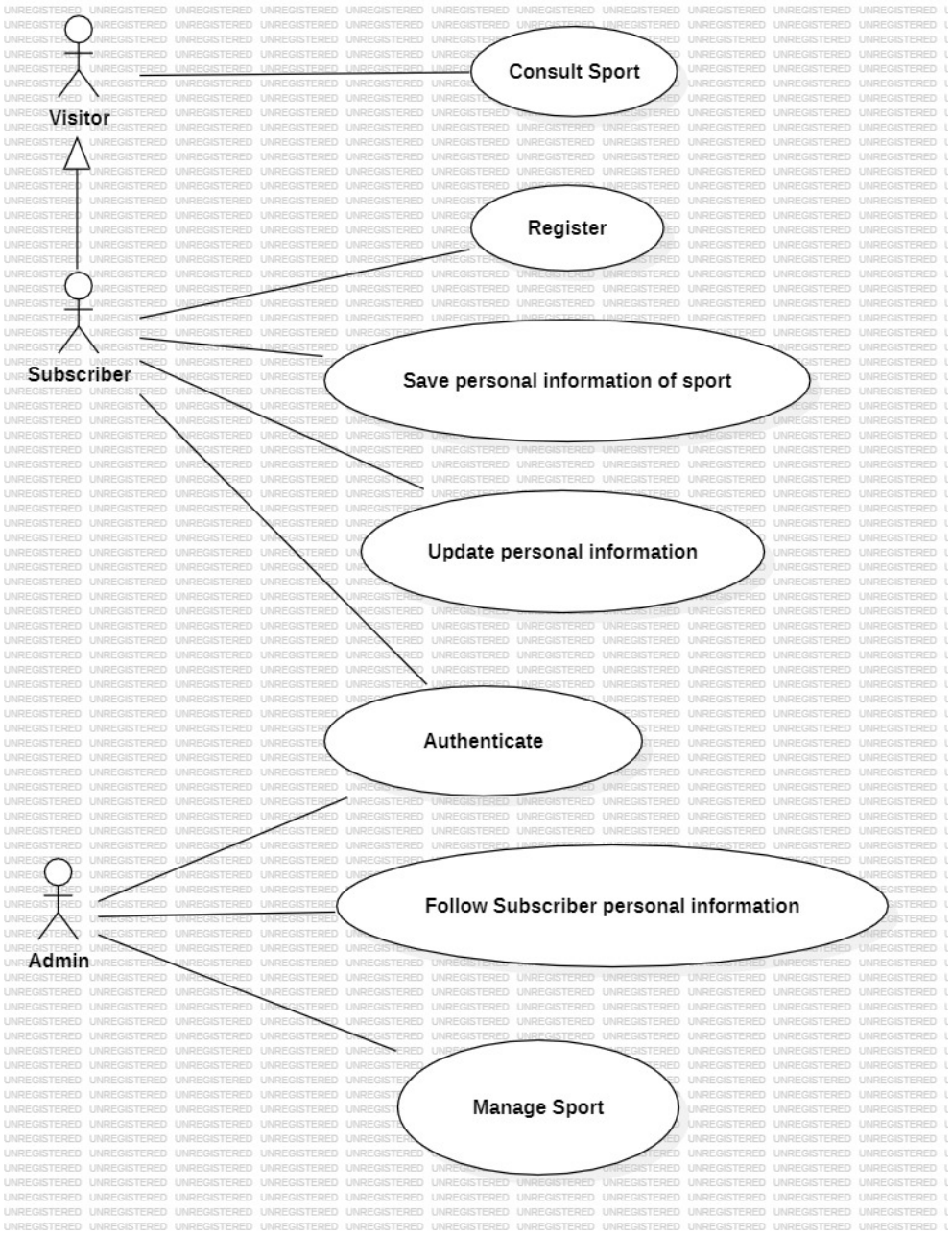

FIGURE 1.7 – global use case diagram

1.5 Product backlog

Backlog de Produit	Priorité	Estimation	Planification
As a subscriber, I can register.	1	Moyen	Sprint 0
As an administrator, I can authenticate	1	Moyen	Sprint 0
As an administrator, I can manage the workouts and diets.	1	Moyen	Sprint 0
As a guest, I can view the available workouts and diets.	2	Moyen	Sprint 1
As a subscriber, I can update my information.	2	Moyen	Sprint 1
As an administrator, I can track the data of subscribers.	2	Moyen	Sprint 1

1.6 Conclusion

Dans ce chapitre , nous avons fait un d ecoupage fonctionnel d etaill e de notre future solution par le biais de diagramme de cas d'utilisation et le backlog de produit.

Chapitre 2

Identification of Backlog for Release 1 Sprint 0

2.1 Introduction

Now that we set up a perfect understanding of our system by analyzing its use cases, classifying them by priorities into 2 sprints of 1 releases, and identifying their actors relatively, we proceed to the next chapter and the first phase of development. We will be going through combined use cases of our system, which have the highest priority, by explaining their backlogs, elaborating their user stories with the refinements, then eventually showcasing them by implementing the class and sequence diagrams

2.2 Sprint 0

We will be presenting the following user stories in the first sprint :
— Authenticate
— Register
— manage sport

2.2.1 Identifying the sprint0 backlog

The following table contains the backlog elements that are realised during the sprint 0 :

Backlog de Produit	**Priorité**	**Estimation**	**Planification**
As a subscriber, I can register.	1	Moyen	Sprint 0
As an administrator, I can authenticate	1	Moyen	Sprint 0
As an administrator, I can manage the workouts and diets.	1	Moyen	Sprint 0

2.2.2 Refinement of sprint 0

In this section, we analyze the different use-case scenarios of the first sprint

Refinement of the user story ”authenticate” :

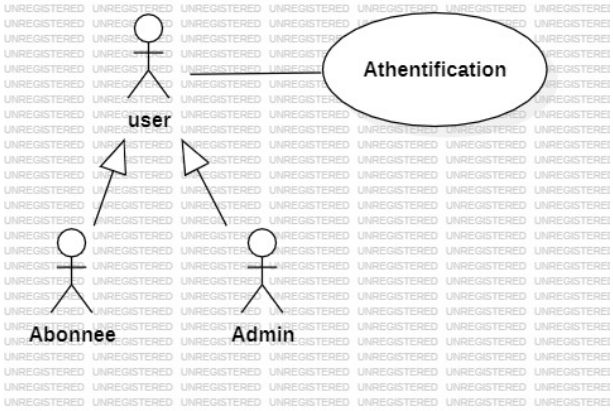

FIGURE 2.1 – Authentication use case diagram

TABLE 2.1 – Detailed description of the "Authentication" use case

Use Case Scenario	**As a user, I can authenticate**
Actors	User
Pre-Conditions	The User must have an account
Post-Conditions	Authenticate
Main Scenario	- The user enters their email and password. - The user clicks on the login button. - The system verifies whether the credentials exist in the database or not. - If they do : The system redirects the user to their specified Dashboard. - If not : The system notifies the user that their credentials are incorrect and
Exception	The system displays an error message if the data is incorrect.

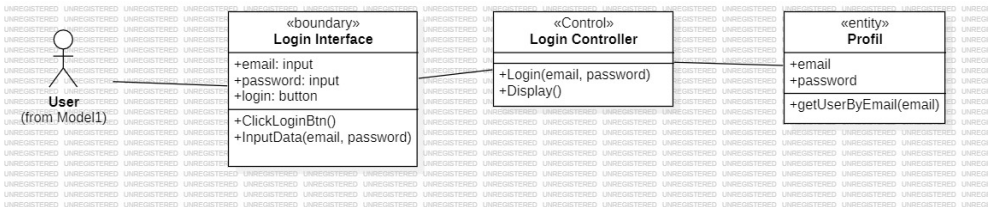

FIGURE 2.2 – Authenticate class Diagram

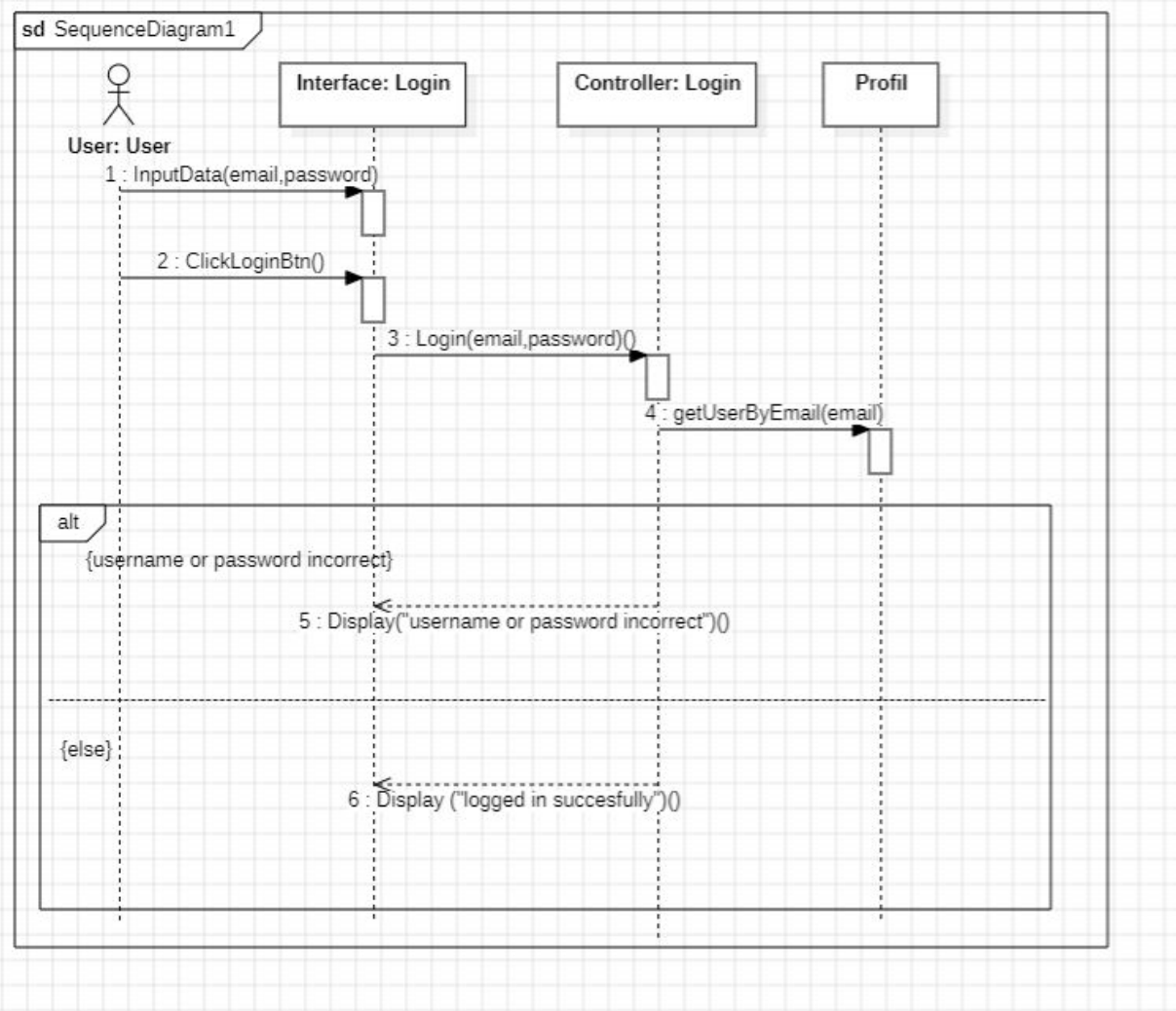

FIGURE 2.3 – Authentication sequence diagram

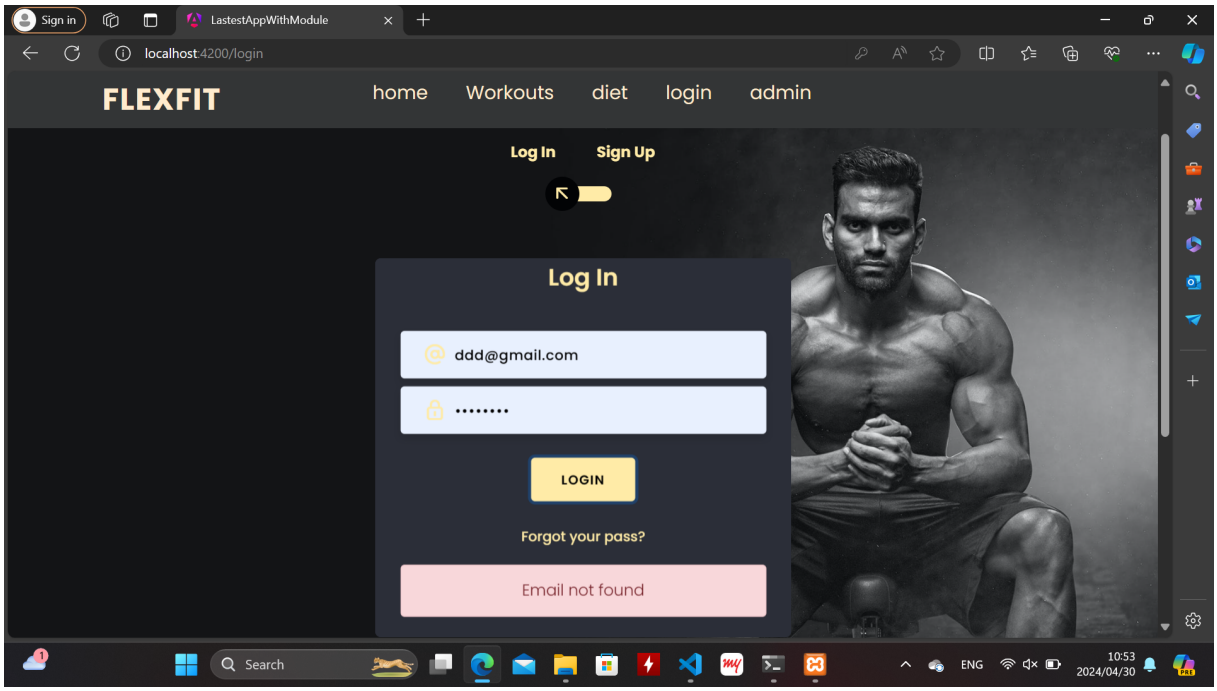

FIGURE 2.4 – Authentication Screenshot

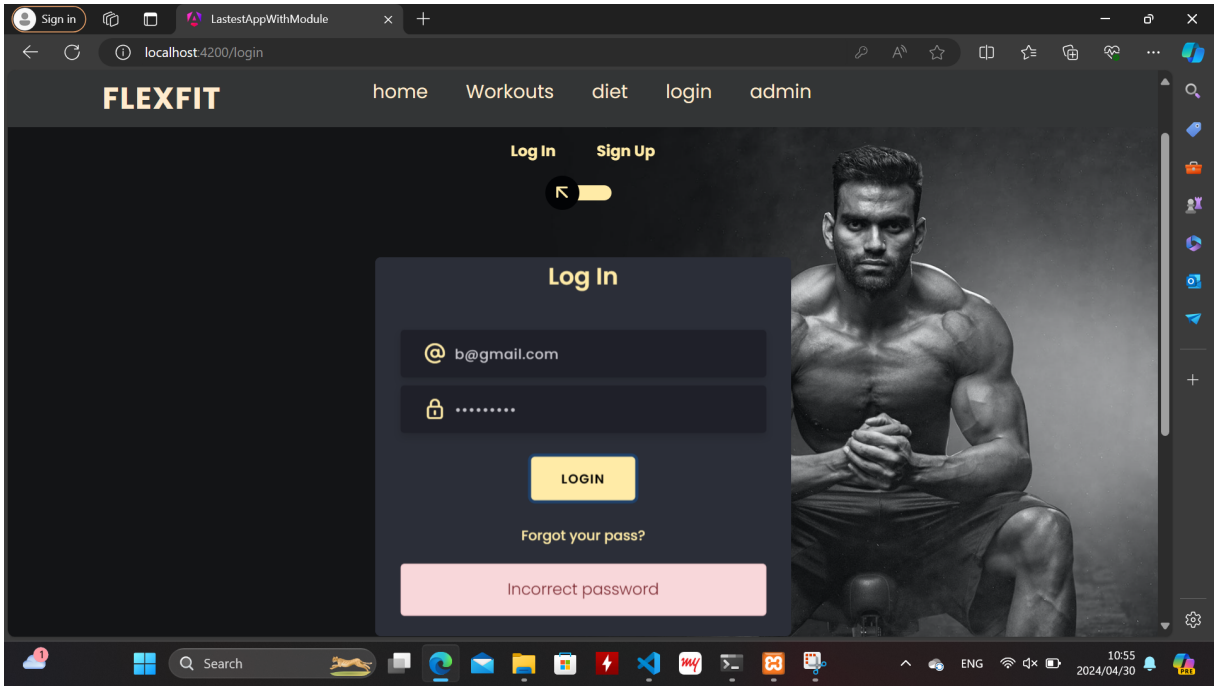

FIGURE 2.5 – Authentication Screenshot

Refinement of the Subscriber story "Registre"

The following figure showcases the use case . The following figure showcases the sequence diagramm. The following figure showcases the class Diagram.

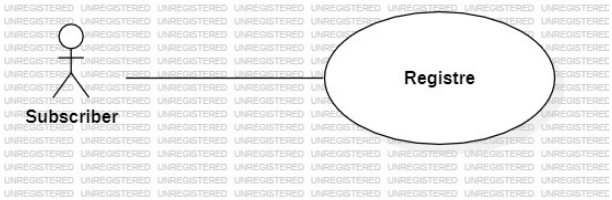

FIGURE 2.6 – "Register" use case diagram

TABLE 2.2 – Detailed description of the "Register" use case

Use Case Scenario	**As a Subscriber, I can Register**
Actors	Subscriber
Pre-Conditions	The system is running. The email address and password are validated.
Post-Conditions	Registration established.
Main Scenario	- The system displays the registration interface. - The Subscriber enters their personal details. - The subscriber clicks on the "Sign up" button. - The system verifies the data. - The system displays a message confirming the successful registration.
Exception	The system displays an error message if the data is incorrect.

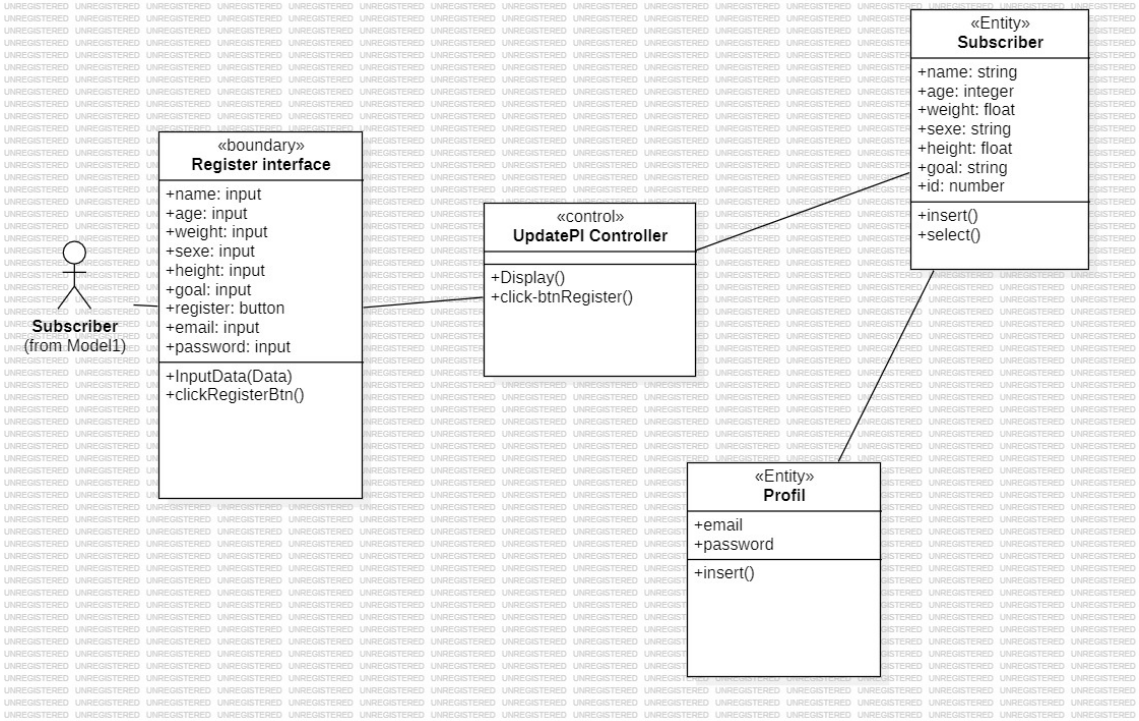

FIGURE 2.7 – Register class Diagram

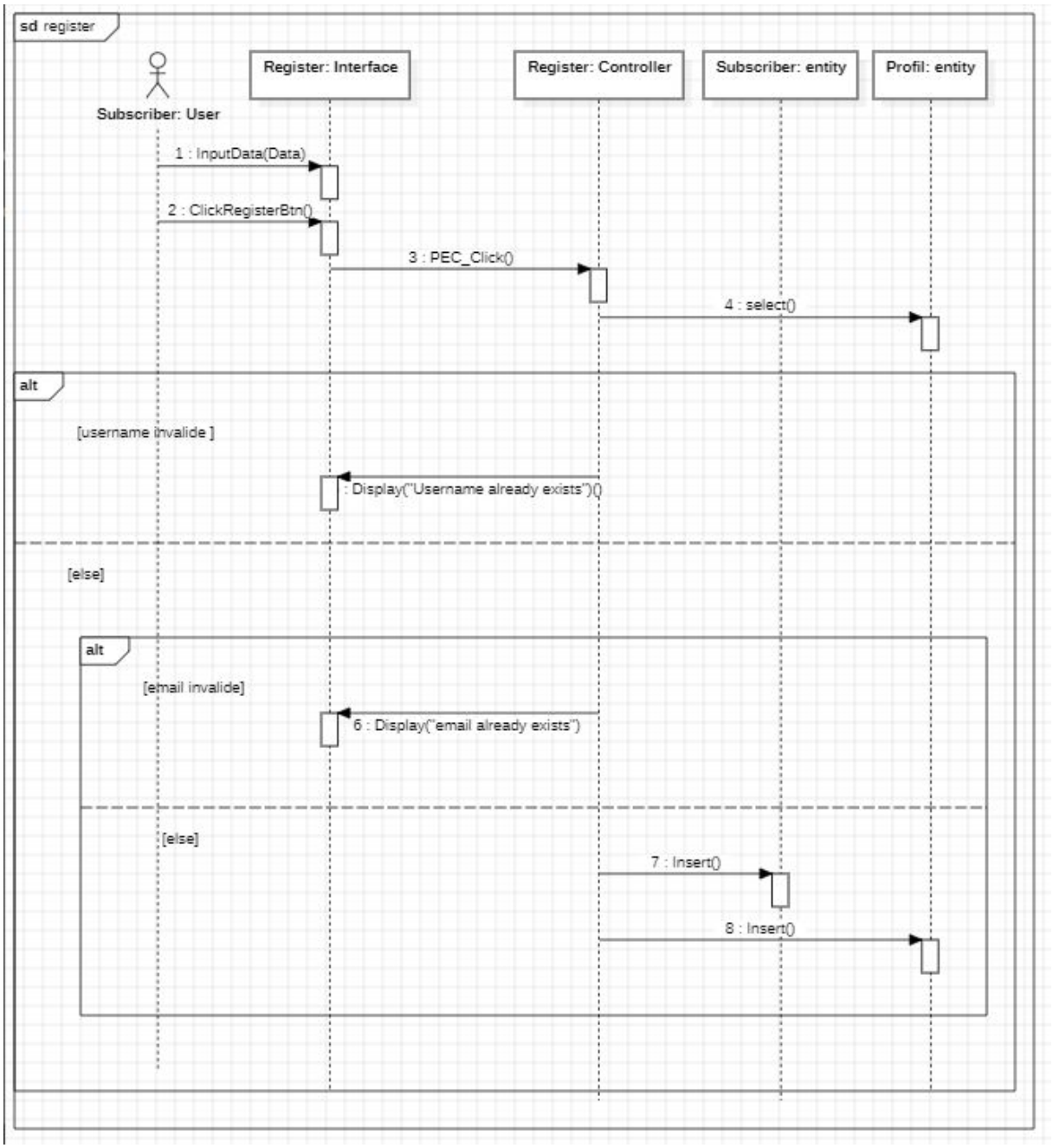

FIGURE 2.8 – Register sequence diagram

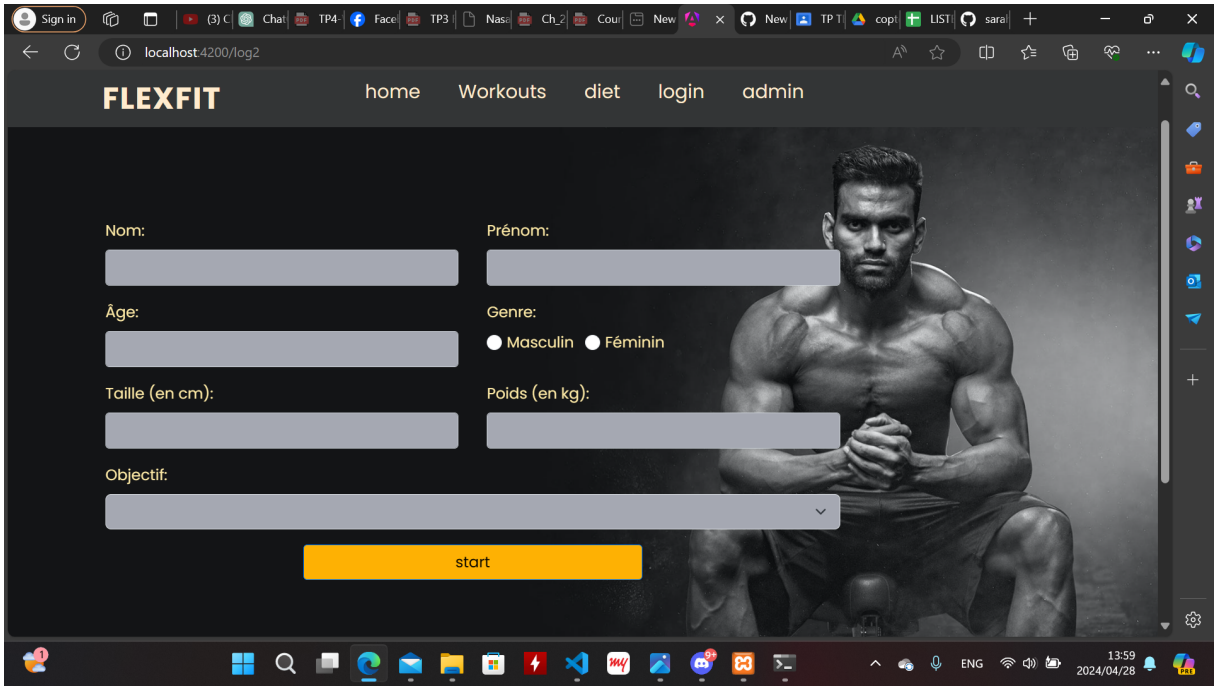

FIGURE 2.9 – Register ScreenShot

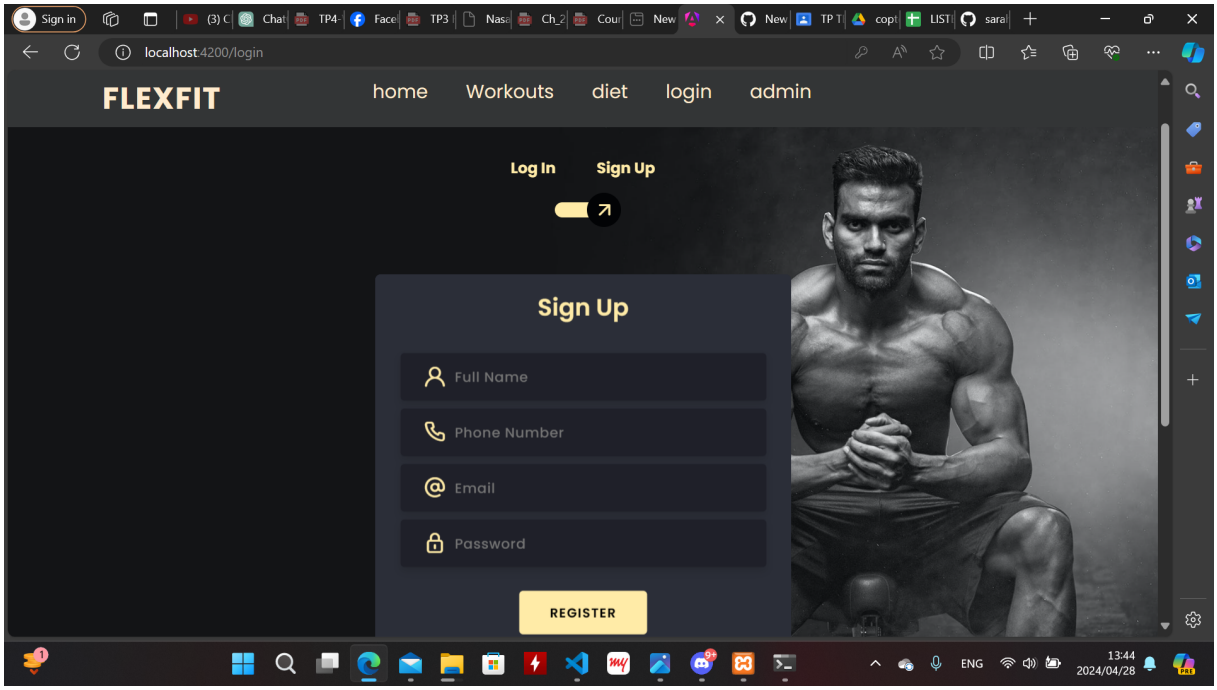

FIGURE 2.10 – Register ScreenShot

2.3 Conclusion

In this chapter, we have carried out the use cases of release 1, followed by the presentation of some interfaces along with their analyses. In the next chapter, we present the second release.

Chapitre 3

Identification of Backlog for Release 2 Sprint 1

3.1 Introduction

Now that we set up a perfect understanding of our system by analyzing its use cases, classifying them by priorities into 4 sprints of 2 releases, and identifying their actors relatively, we proceed to the next chapter and the first phase of development. We will be going through combined use cases of our system, which have the highest priority, by explaining their backlogs, elaborating their user stories with the refinements, then eventually showcasing them by implementing the class and sequence diagrams in addition to the final reveal of the developed interfaces.

3.2 Sprint 1

We will be presenting the following user stories in the first sprint :
— Consult available workouts and diets
— Update information
— track subscriber data

3.2.1 Identifying the sprint1 backlog

The following table contains the backlog elements that are realised during the sprint 1 :

Backlog de Produit	**Priorité**	**Estimation**	**Planification**
As a visitor, I can Consult available workouts and diets.	2	Moyen	Sprint 1
As a subscriber, I can Update my data.	2	Moyen	Sprint 1
As an administrator, I can track Subscribers data	2	Moyen	Sprint 1

3.2.2 Refinement of sprint 1

In this section, we analyze the different use-case scenarios of the second sprint

Refinement of the user story "Update information"

The following figure showcases the use case . The following figure showcases the class Diagram.

The following figure showcases the class Diagram.

FIGURE 3.1 – user case diagram of story update information

TABLE 3.1 – Detailed description of the "Update Information" use case

Use Case Scenario	**As a Subscriber, I can Update my data**
Actors	Subscriber
Pre-Conditions	The User must be authenticated
Post-Conditions	data updated
Main Scenario	- The system displays the "Update Personal Information" interface. - The system displays the user's personal data. - The Subscriber clicks on the "Update" button to change their personal data - The Subscriber clicks on the 'import an image' button to change their profil
Exception	The system displays an error message if the data is incorrect.

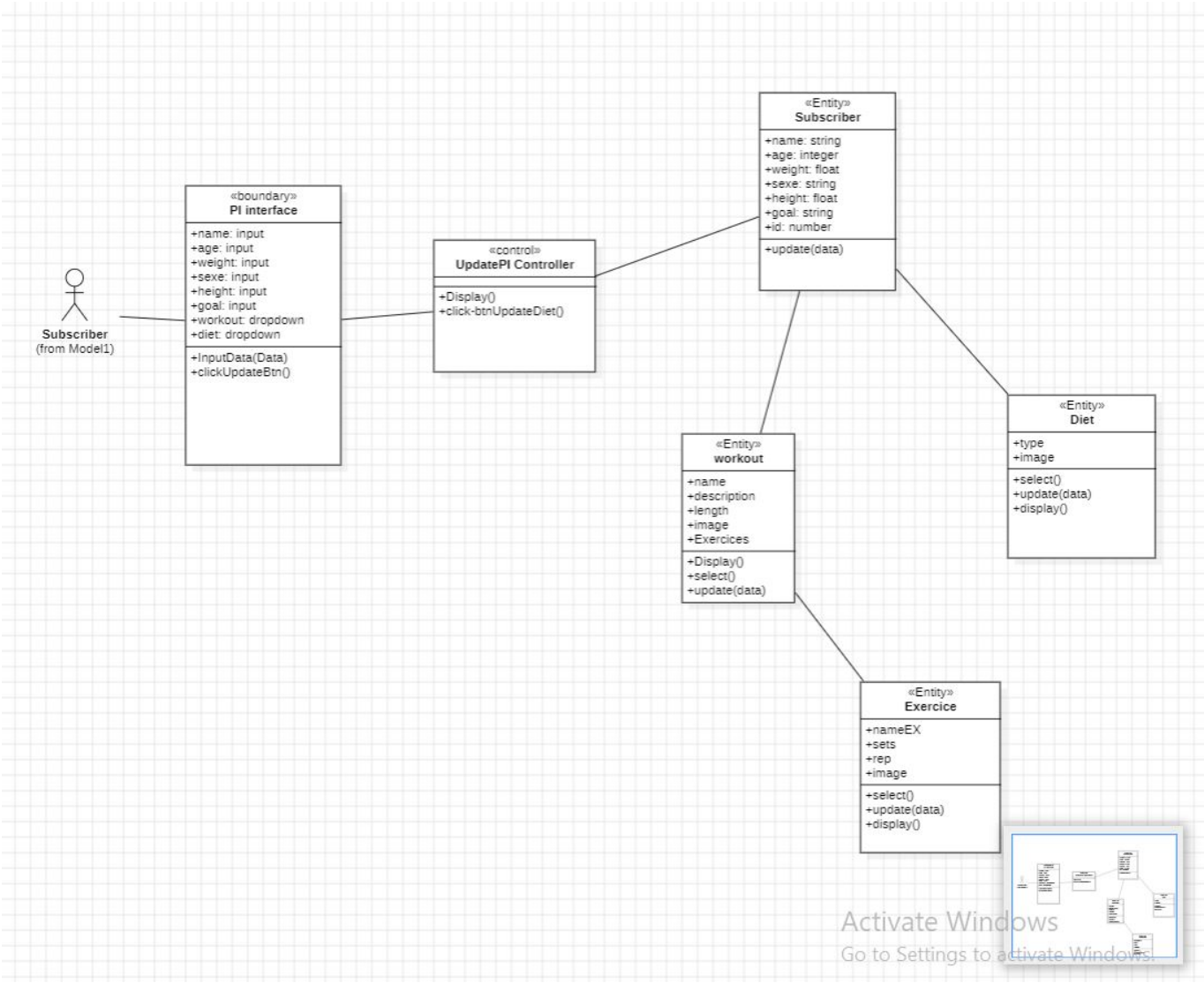

FIGURE 3.2 – update information class Diagram

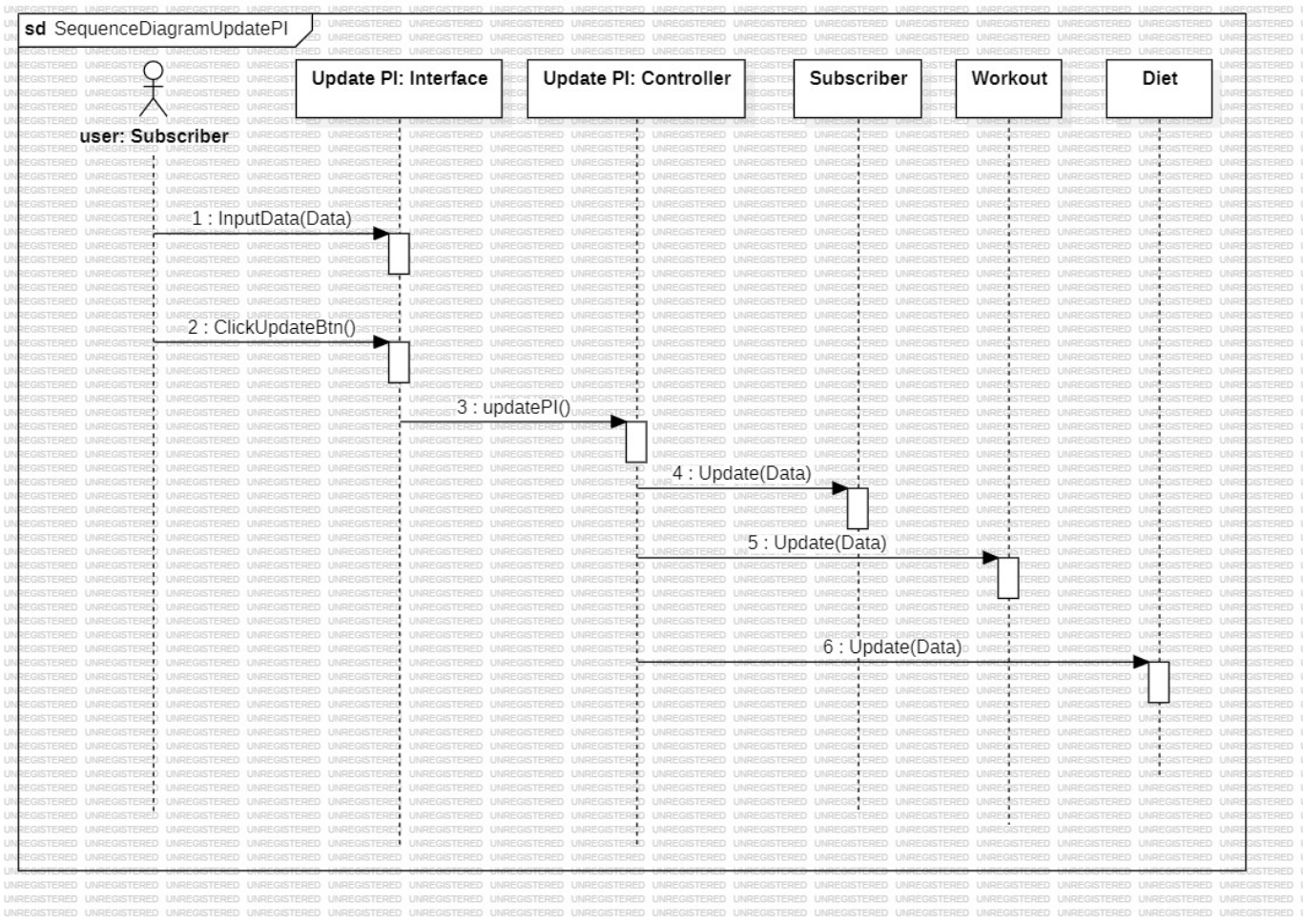

FIGURE 3.3 – update information sequence Diagram

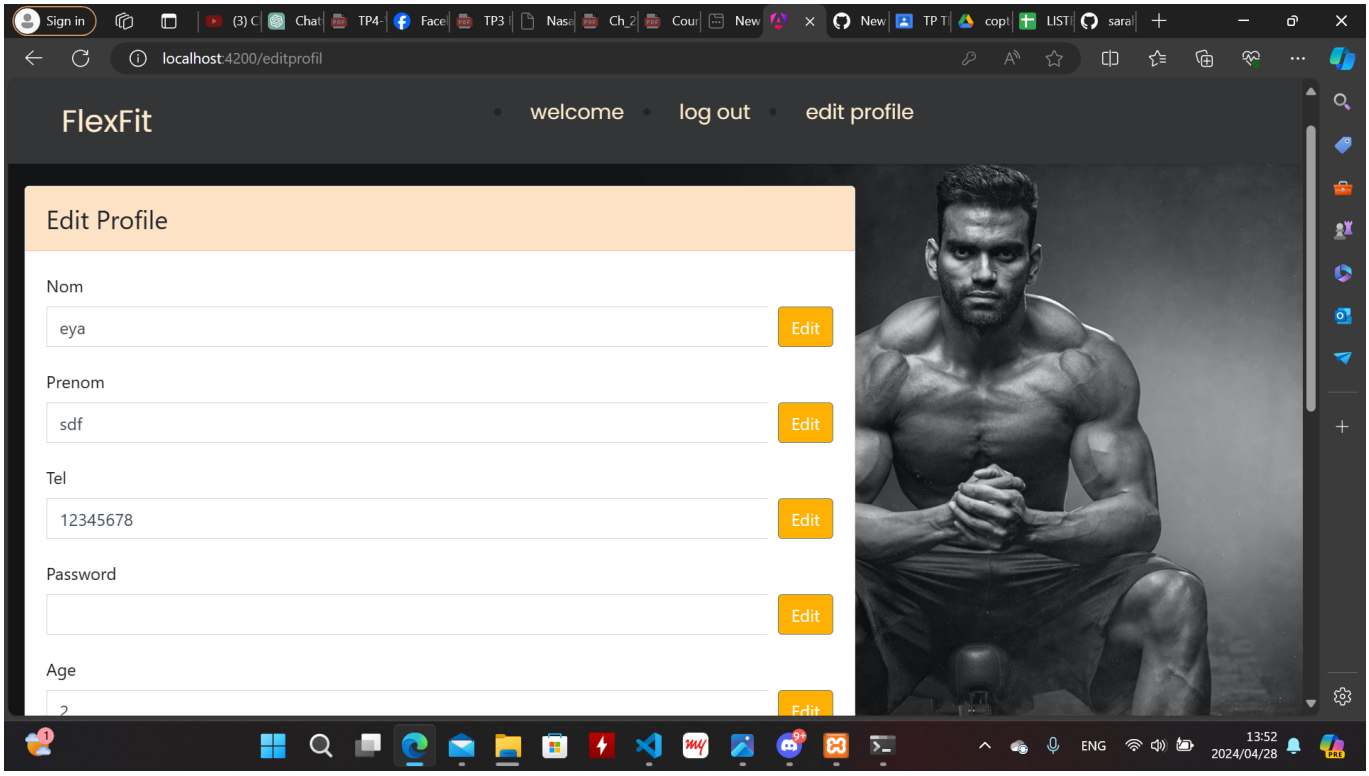

FIGURE 3.4 – update information screenshot

Refinement of the user story "follow Subscriber"

The following figure showcases the use case .

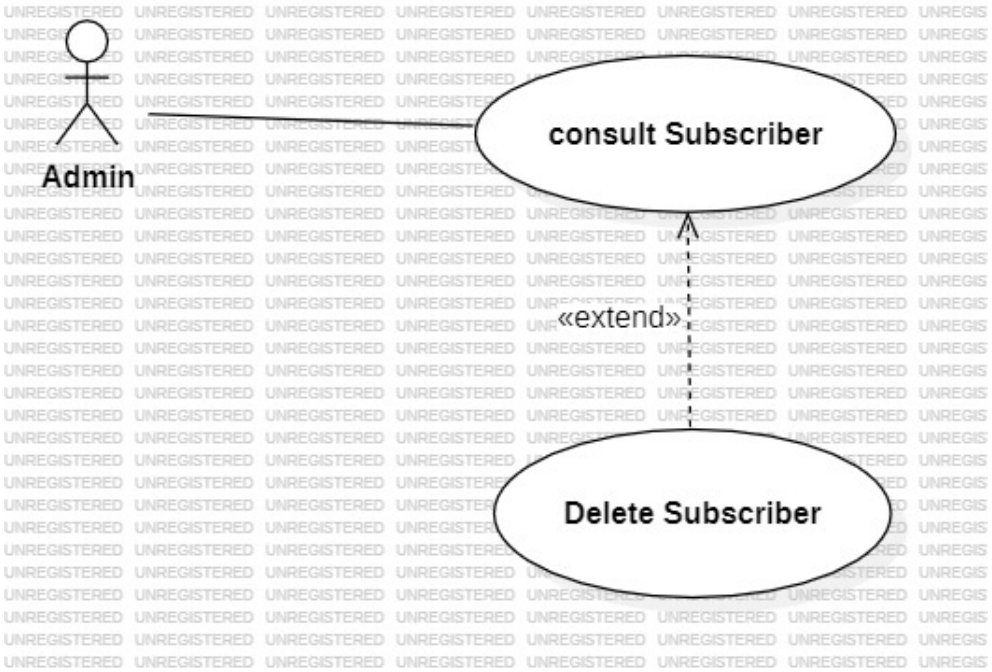

FIGURE 3.5 – user case diagram of story follow Subscriber

Refinement of the user story "consult Subscriber"

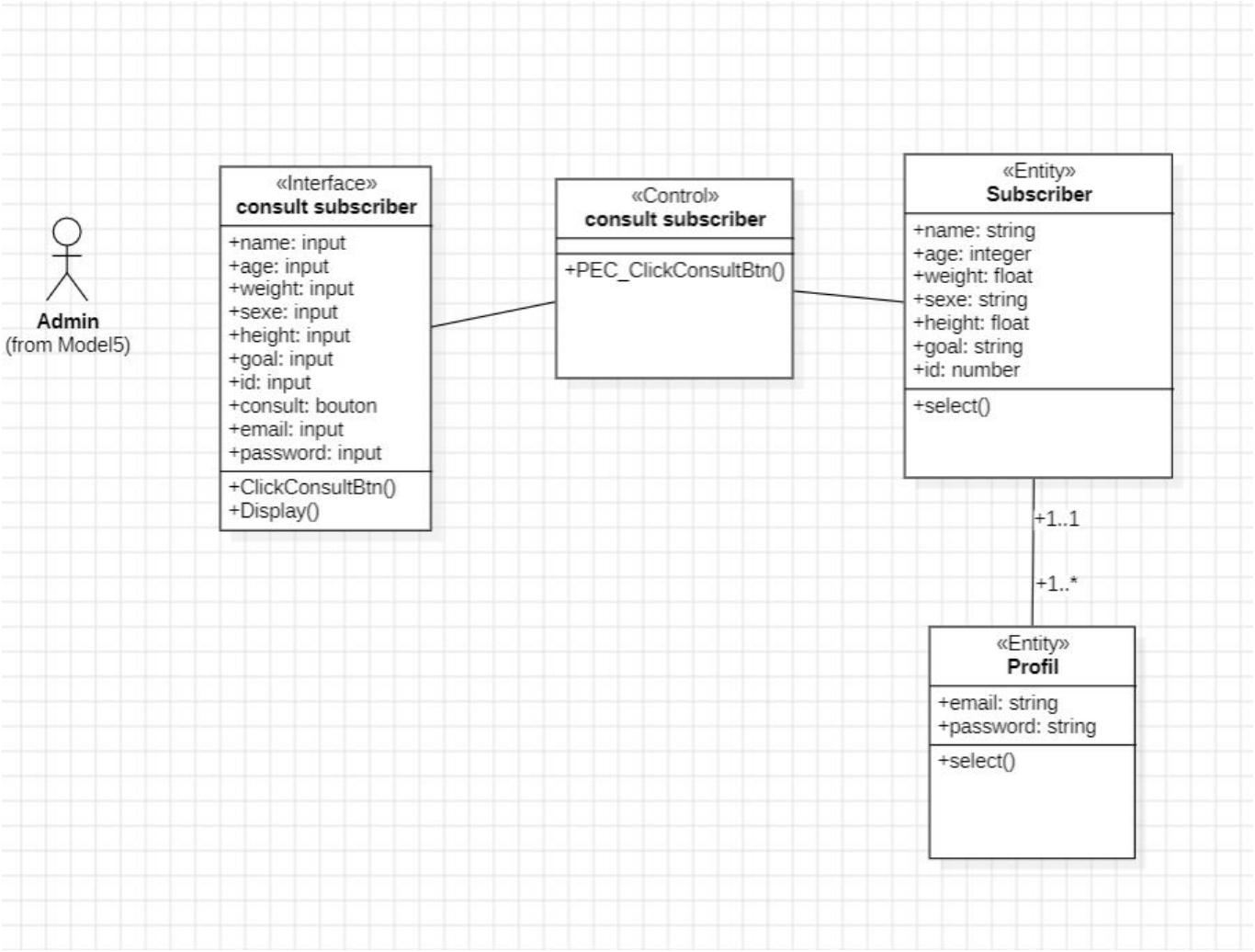

FIGURE 3.6 – Consult subscriber class Diagram

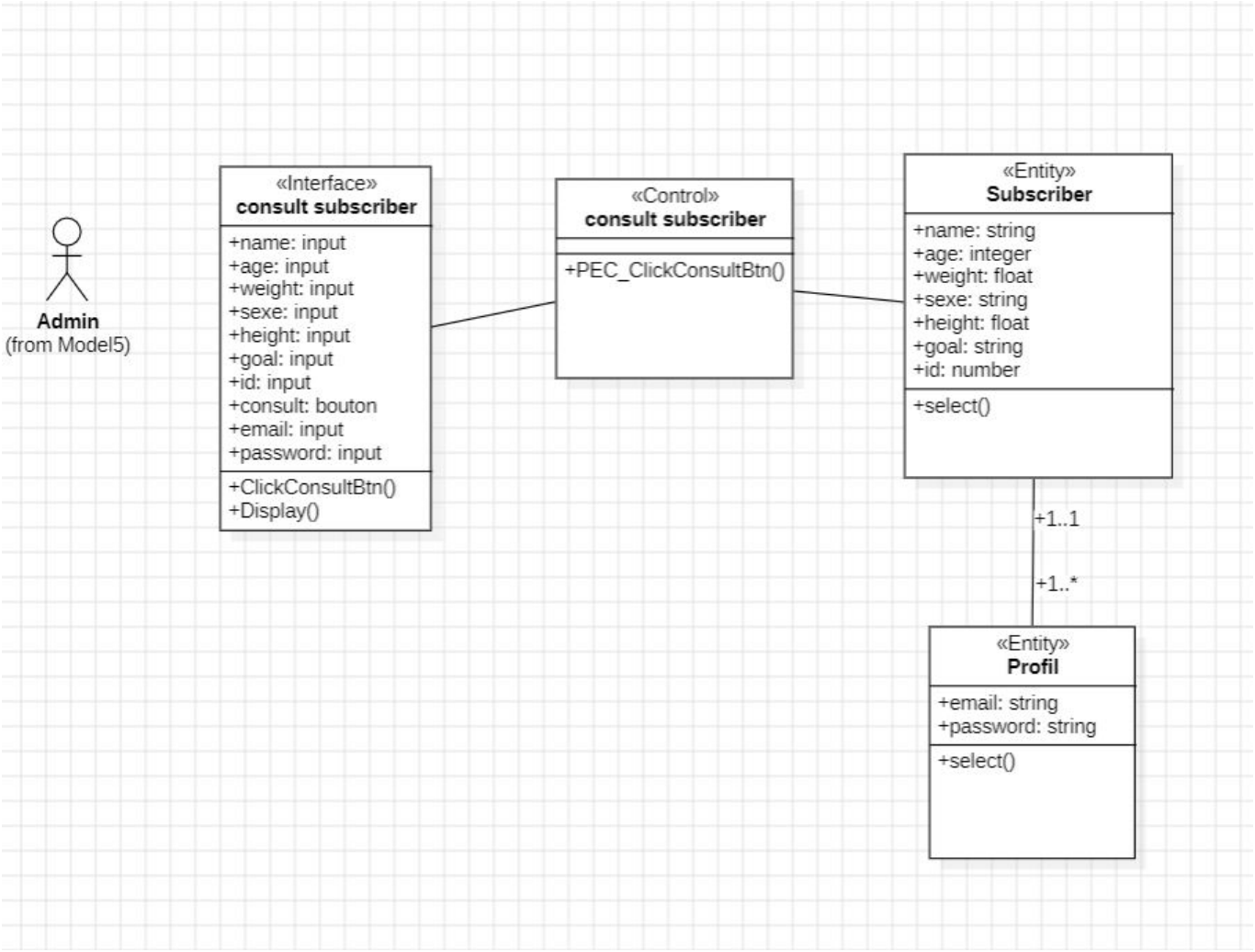

FIGURE 3.7 – Consult subscriber class Diagram

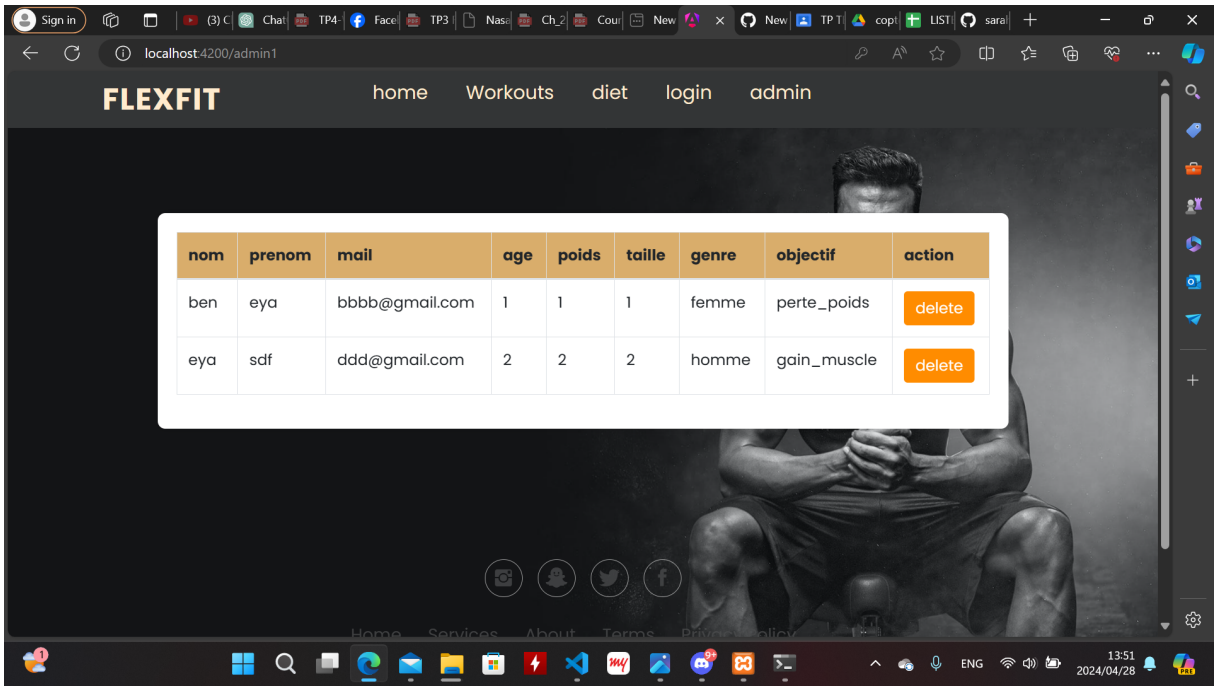

FIGURE 3.8 – consult subscriber Screenshot

Refinement of the user story "Delete Subscriber"

The following figure showcases the use case The following figure showcases the class Diagram.

TABLE 3.2 – Detailed description of the "Delete Subscriber" use case

Use Case Scenario	**As an administrator, I can delete Subscribers**
Actors	The administrator
Pre-Conditions	The administrator must be connected
Post-Conditions	Subscriber Deleted
Main Scenario	- The system displays the "Delete Subscriber" interface. - The administrator clicks on the delete button – The system drops the selected user from the database

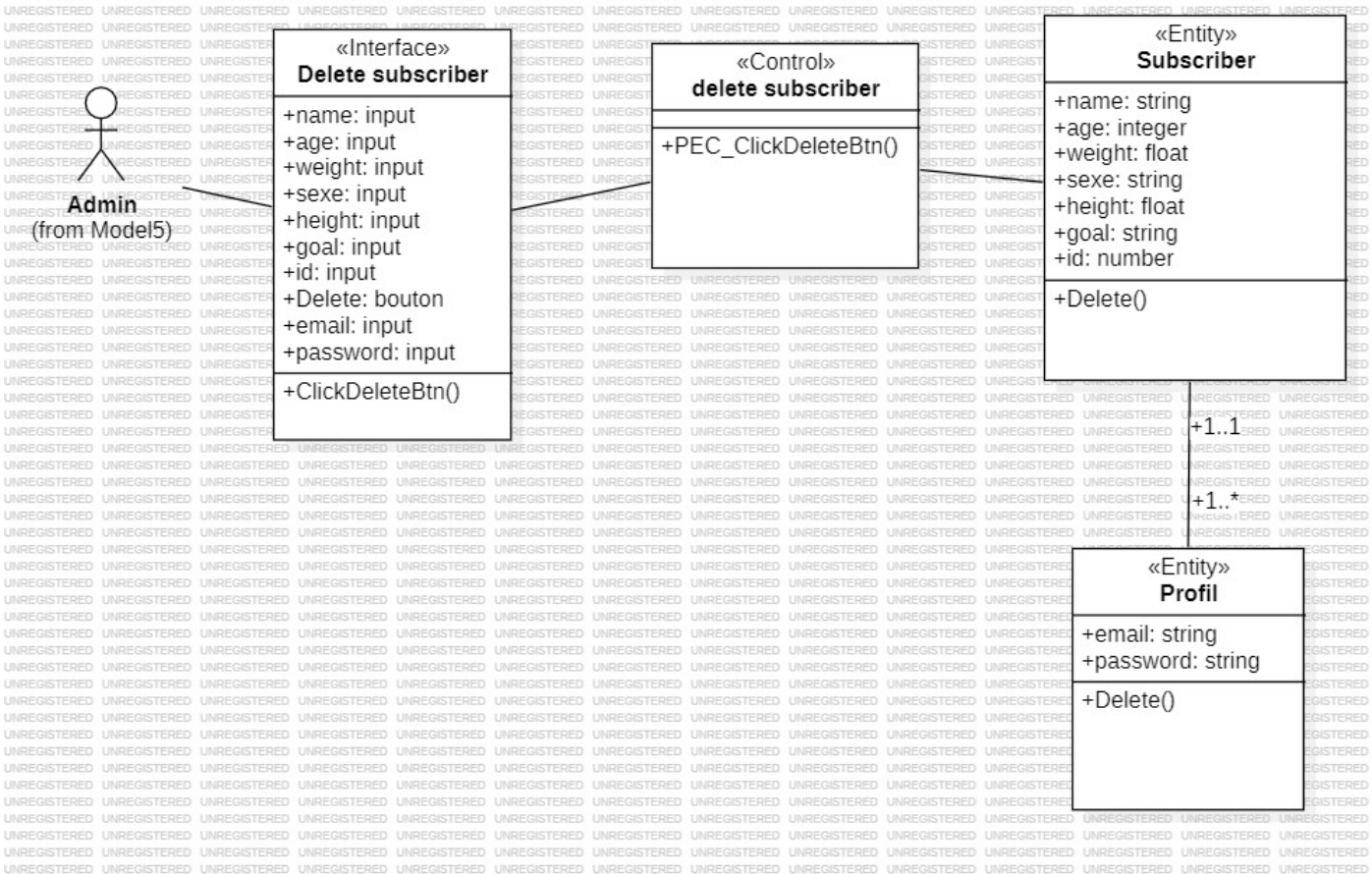

FIGURE 3.9 – Delete subscriber class Diagram

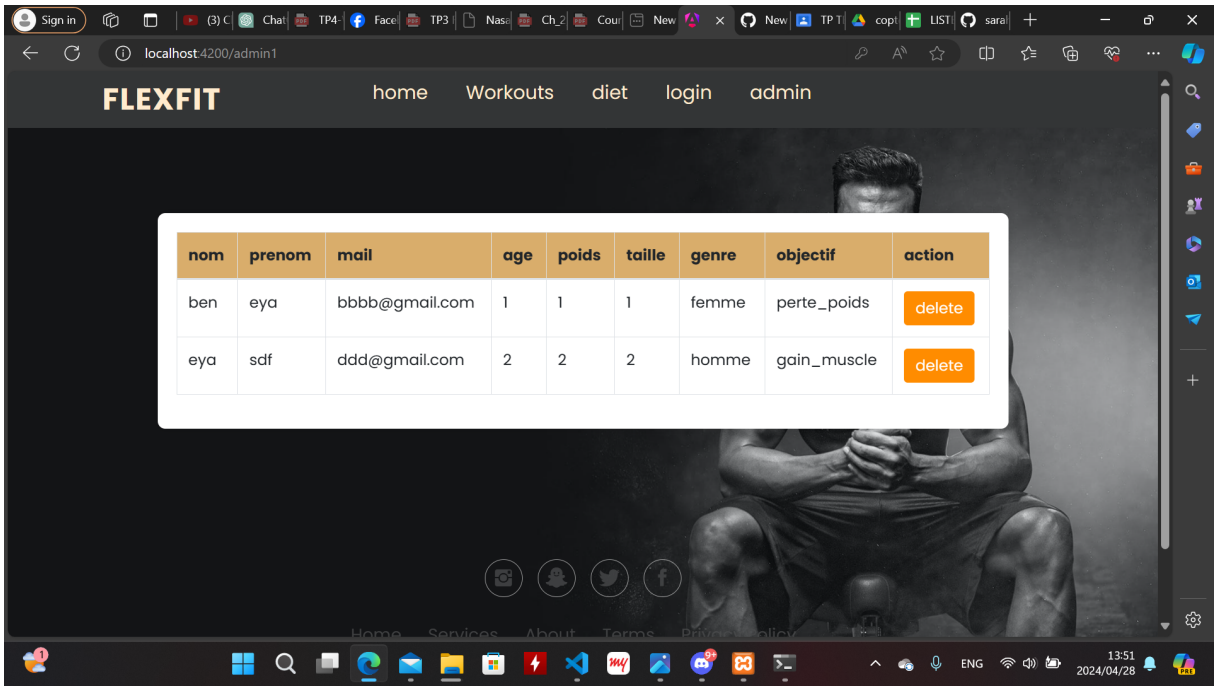

FIGURE 3.10 – Delete And Consult subscriber Screenshot

General Conclusion

In conclusion, the development of our mini web project focused on workouts and diets has been a fulfilling journey. Through the implementation of various use cases and the creation of intuitive interfaces, we have successfully provided users with a platform to track their fitness journey effectively.

The use of diagrams, including use case diagrams, sequence diagrams, and class diagrams, has greatly aided in the visualization and understanding of the system's functionalities. These diagrams have served as valuable tools for both development and documentation purposes, ensuring clarity and coherence throughout the project lifecycle.

Moreover, the iterative approach adopted in the development process has allowed for continuous refinement and improvement of the system. User feedback and testing have played a crucial role in identifying areas for enhancement, resulting in a more user-friendly and feature-rich platform.

Overall, our mini web project demonstrates the importance of user-centered design and iterative development methodologies in creating successful digital solutions. By prioritizing user needs and incorporating feedback loops, we have delivered a product that addresses real-world challenges and empowers users to achieve their fitness goals effectively.

Bibliographie