

TUNISIAN REPUBLIC



PROJECT BOOK

Women Empowerment

Prepared by

Farouk Gassara

Achraf BENZrig

Marwa Boudellaa

Mohamed Ali Charrad

Mohamed Ali Omri

Loujein Hragua

Abstract

It is true that men and women live in poverty, face discrimination and face barriers. But throughout history, in every country in the world, women and girls face additional barriers and discrimination solely because of their gender.

Fortunately, technology is available to help showcase their problems. Now it's time to move to a virtual assistance plan, a plan that protects women from being devalued and discriminated against, and this is what this project offers among the benefits of online employment, comprehensive digitization, a forum for discussing women's problems among themselves and among experts and providing assistance to all women. In the event that there is a need in addition to providing workshops and charitable events. Through this project, we hope to do our utmost to help women around the world.

List of Figures

I.1	an overview of the application Find Sisterhoods	7
I.2	an overview of the application Girls Who Code Loop	8
I.3	an overview of the application Girls Who Code Loop	8
I.4	Elements of the Unified Process	10
I.5	Comparative Study	11
II.1	Forum Use Case	17
II.2	Training Use Case	18
II.3	Training Use Case	18
II.4	User Use Case	19
II.5	Events Use Case	19
II.6	Use Case Diagram	20
II.7	Class Diagram	21
II.8	System Sequence Diagram Authentification	22
II.9	System Sequence Diagram Delete Training	23
II.10	System Sequence Diagram Edit User	24
II.11	an overview of the application Physical architecture	25
III.1	Manage Publications Scenario	27
III.2	Edit Publications Scenario	28
III.3	Training Scenario	28
III.4	Manage Users Scenario	29
III.5	Manage Reports Scenario	29
III.6	JOB APPLICATION Scenario	30
III.7	Manage Events	30
III.8	Manage Donation	31
III.9	Sequence diagram Update Publication	32
III.10	Sequence diagram Update Training	33
III.11	Sequence diagram RDV	34
III.12	Package class diagram : Training	35
III.13	Package class diagram : User	36
III.14	Package class diagram : Forum	37
III.15	Package class diagram : RDV	38
III.16	an overview of the application Operational architecture	39
III.17	an overview of the application Logical architecture	39
III.18	Deployment diagram	40
IV.1	Spring Boot	41
IV.2	Angular Advantages	42
IV.3	Job Market Angular	42

IV.4 Learning Curve Angular	43
IV.5 Angular	43
IV.6 Login	44
IV.7 Sign up	44
IV.8 Users List	45
IV.9 Stats	45
IV.10Events	46
IV.11Events	46
IV.12Login	47
IV.13Sign up	47
IV.14Users List	48
IV.15Stats	48
IV.16Login	49
IV.17Sign up	49
IV.18Users List	50
IV.19Stats	50
IV.20Login	51
IV.21Login	51
IV.22Login	52
IV.23Login	52
IV.24Login	53
IV.25Login	53
IV.26Login	54
IV.27Login	54
IV.28Login	55
IV.29Login	55
IV.30Login	56
IV.31Login	56
IV.32Login	57
IV.33Login	57
IV.34Login	58
IV.35Login	58
IV.36Login	59
IV.37STS	60
IV.38VSC	60
IV.39Postman	60
IV.40GitHub	61
IV.41Adobe XD	61
IV.42Adobe XD	61

Contents

Abstract	1
List of Figures	3
Table of Contents	5
General Introduction	6
I Project Overview	7
I.1 Introduction	7
I.2 Existing Slution Analysis	7
I.2.1 Criticism of the existing :	8
I.3 Problem and proposed solution :	9
I.4 Methodological choice :	9
I.4.1 UP VS SCRUM :	10
I.4.2 Comparative Study :	11
I.4.3 Choosing UP Method :	11
I.5 Conclusion :	13
II Requirements Analysis	14
II.1 Introduction	14
II.2 Specification of Requirements	14
II.2.1 Specification of Functional Requirements :	14
II.2.2 Specification of Non-Functional Requirements :	15
II.3 Identification of actors :	16
II.4 Identification of use cases :	17
II.5 Use Case Diagram	20
II.6 Class Diagram	21
II.7 System Sequence Diagram	22
II.8 Physical architecture	25
II.9 Conclusion :	25
III Conception	26
III.1 Introduction	26
III.2 Refinement and detailed description of use cases	27
III.2.1 Text description of use cases	27
III.2.2 Sequence diagram	32
III.3 Conception class diagram	35
III.3.1 Package class diagram : Training	35
III.3.2 Package class diagram : User	36
III.3.3 Package class diagram : Forum	37
III.4 Global solution architecture	39
III.4.1 Operational architecture	39

III.4.2 Logical architecture	39
III.5 Static Aspect	40
III.5.1 Deployment diagram	40
III.6 Conclusion :	40
IV Implementation	41
IV.1 Introduction	41
IV.2 Framework and technical tools	41
IV.2.1 Spring boot framework	41
IV.2.2 Angular framework	42
IV.2.3 Data mining	43
IV.3 Coding and test	44
IV.3.1 Spring Boot (Postman)	44
IV.3.2 Data mining (R Studio/Python)	47
IV.3.3 Angular	51
IV.4 Working environment	59
IV.4.1 Materiel Environment	59
IV.4.2 Software Environment	60
IV.5 Conclusion	61

General Introduction

Women are the primary caretakers of children and elders in every country of the world. International studies demonstrate that when the economy and political organization of a society change, women take the lead in helping the family adjust to new realities and challenges. They are likely to be the prime initiator of outside assistance, and play an important role in facilitating (or hindering) changes in family life. However, women still suffer from violence, gender equality, racism and economic inequality and..., that's why the phenomenon of women's empowerment has appeared, which is defined to promote women's sense of self-worth, their ability to determine their own choices, and their right to influence social change for themselves and others. It is closely aligned with female empowerment – a fundamental human right that's also key to achieving a more peaceful, prosperous world. Being a future engineer doesn't make us less committed than any other citizen to the causes of women's anguish and that is the main reason behind working on this project. It consists of an application that centralizes and provides women with the resources to resist against any issue that could face them, our platform is an outlet that allows women around the world to express, to communicate, to exchange, and to help each other to find jobs to find support and to learn new skills with the guidance of experts.

The next pages will give details about how we brought this application to an end. Though, we planned our report as follows : First of all we present the project General framework. Next chapter treats our needs of analysis statements. Last but not least, we explain our conception. Finally, we will explain the achievements.

Chapter I

Project Overview

I.1 Introduction

This chapter, we present the General presentation of the project, an existant study in which we will show the strenth and the weakness, then we will define the problem and the proposed solution and finally we will explain our Methodological choice.

I.2 Existing Slution Analysis

Find Sisterhood :

Founded by Ana Pompa Alarcon, the findSisterhood app was created as a safe space for women to anonymously share their stories with each other without the concerns of judgment. Thanks to cryptography, the app makes your identity untraceable, allowing the user to ask any question they want or seek support without embarrassment. [1]



Figure I.1: an overview of the application Find Sisterhoods

Girls Who Code Loop :

Girls Who Code, the national nonprofit working to close the gender gap in technology, created a custom-built app designed exclusively to connect members of the program. The app helps the Girls Who Code, students and alumni stay in the loop and support each other through job and internship postings, questions regarding coding, and more [2]

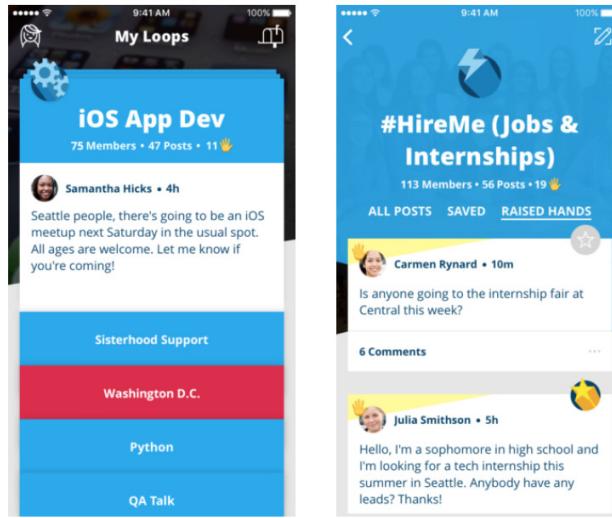


Figure I.2: an overview of the application Girls Who Code Loop

Maven Clinic :

Entrepreneur Katherine Ryder founded Maven Clinic, a virtual clinic dedicated to women's and family health. Through the app, users can meet OB-GYNs, therapists, nutritionists, and more. They can also schedule virtual appointments, video chat with practitioners, and pick up prescriptions [3]

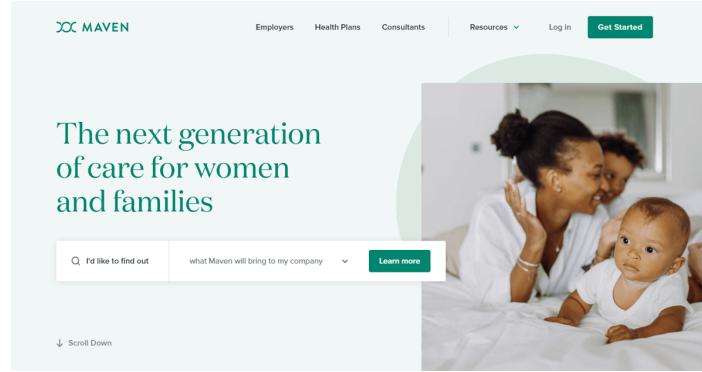


Figure I.3: an overview of the application Girls Who Code Loop

I.2.1 Criticism of the existing :

A scrutiny comparative study of the previous applications led us to the following weakness which are summarized in the following table :

	Find Sisterhood	Girls Who Code Loop	Maven Clinic
Authentication	YES	YES	YES
Paid application	NO	NO	NO
Shared data	NO	NO	NO
High-level visual design	NO	YES	YES
Security	YES	YES	NO
Managing Forum	YES	YES	NO
Report Publications	NO	YES	NO
Users Support	NO	YES	YES

Alalyzing this table we notice a big diffrence between our application and the others. this gap is the fruit of hard searching in order to fill in the gabs of others applications

I.3 Problem and proposed solution :

Women are the pioneers of nation they are man's helpmate, partner and comrade. They sacrifices her personal pleasure and ambitions, sets standard of morality, relieves stress and strain, tension of husband, maintains peace and order in the household. Thereby, they are the source of inspiration to whole family for high endeavour and worth achievements in life. That's why we should accord theme more support, help and attention to bypass all her lifetime problems.

The purpose of the present solution is to develop an application which will serve as a tool to make an end to limit the detention process

To well make it ,we shall adress these diffrent requirement :

- Manage forum : share publications in the home page and check if the content is respectful or not
- Dashboard :Interactive dashboard representing statistics to help the admin analyzing, and make decisions
- Manage Users and Roles : Implement a module for managing users and roles with maximum of security (Does not access confidential information if it is not useful for the proper functioning of the application)
- Intuitiveness : A great website anticipates what your visitor is thinking and caters directly to their needs, and has elements arranged in a way that makes sense. If a visitor is searching for one of your products or services on a search engine or directory where your site is listed, it's important that your website have a landing page that is directly relevant to what they searched for rather than forcing them to filter through all of your information. Remember, the shortest distance between two points is a straight line.
- Users Support : the application must offer options for enhanced customer support. Application are developed so customer support is accessed at any time, and they can help drive customer loyalty.

I.4 Methodological choice :

The Shape Unified process (UP) Method is based on Basecamp's own internal product development approach. It is an architecture-centric, use-case driven, iterative and incremental development process

that leverages unified modeling language and is compliant with the system process engineering meta-model. Unified process can be applied to different software systems with different levels of technical and managerial complexity across various domains and organizational cultures.

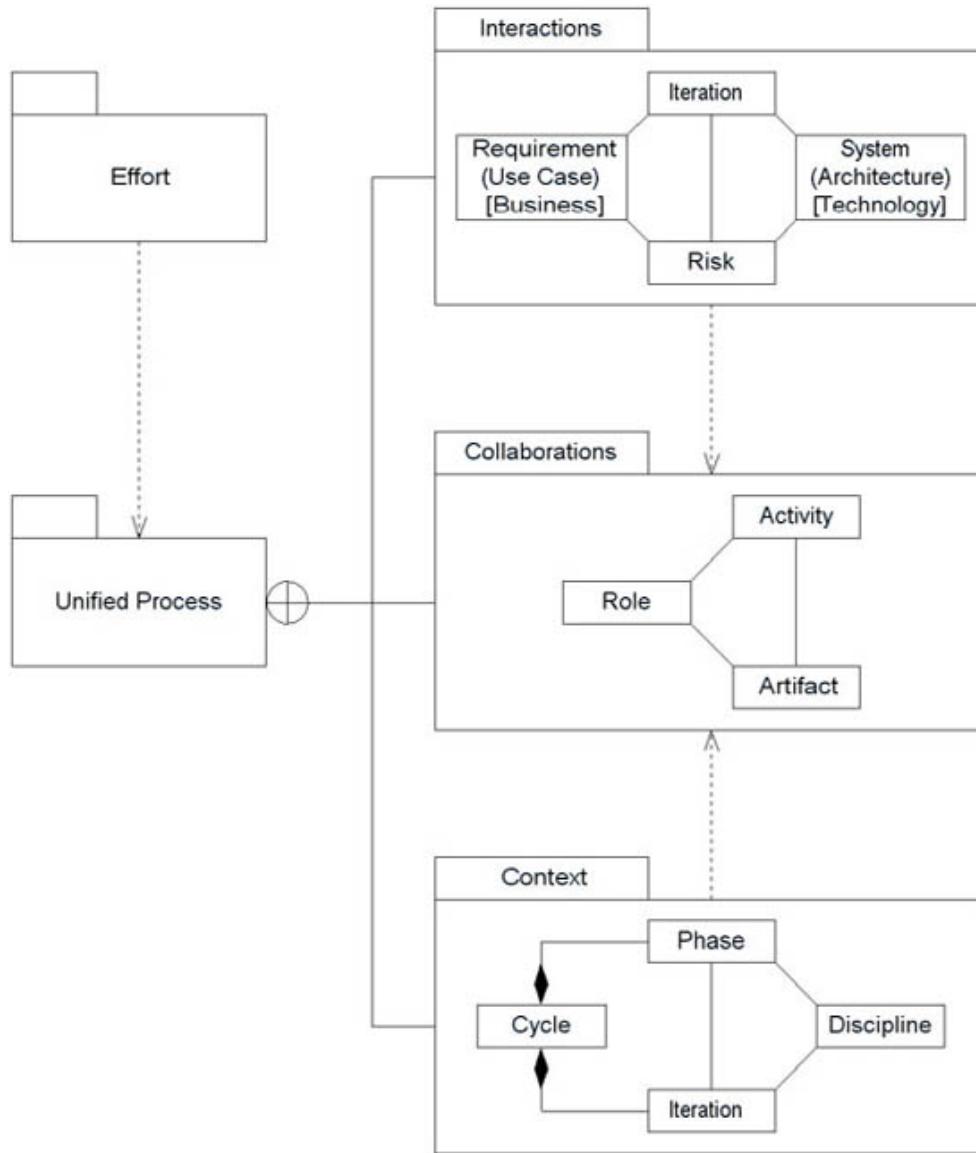


Figure I.4: Elements of the Unified Process

I.4.1 UP VS SCRUM :

Both methodologies are considered to be Agile and approach project activities in the iterative way. They have in common :

Time-boxed approach to delivering software.

The scope is flexible during the time-box. However, there are limits to this flexibility. As long as it meets the intended objective of the Iteration or the Sprint, flexibility is permitted.

Refinement is necessary to make work-ready.

The team has full control on how to split up work in tasks and how the work is performed.

I.4.2 Comparative Study :

	UP	SCRUM
Approach	Iterative	Iterative
Cycle	Formal Cycle is defined across 4 phases, but some workflows can be concurrent.	Each sprint (iteration) is a complete cycle.
Planning	Formal project plan, associated with multiple iterations, is used. The plan is end-date driven and also has intermediate milestones.	No end-to-end project plan. Each next iteration plan is determined at the end of the current iteration (NOT end-date driven). Product Owner (Key Business User) determines when the project is done.
Artifacts	Vision Document, Formal functional requirements package, system architecture document, development plan, test plan, test scripts, etc.	The only formal artifact is the operational software.
Type of Project/Product	Recommended for large, long-term, enterprise-level projects with medium-to-high complexity.	Recommended for quick enhancements and organizations that are not dependent on a deadline.

Figure I.5: Comparative Study

I.4.3 Choosing UP Method :

Unified Process is based on the enlargement and refinement of a system through multiple iterations, with cyclic feedback and adaptation. The system is developed incrementally over time, iteration by iteration, and thus this approach is also known as iterative and incremental software development. The strengths of the latter was the main reason why we have chosen the scrum method as the working methodology for our project.

The iterations are spread over four phases where each phase consists of one or more iterations :

- Inception : the first and the shortest phase in the project. It is used to prepare basis for the project, including preparation of business case, establishing project scope and setting boundaries, outlining key requirements, and possible architecture solution together with design tradeoffs, identifying risks, and development of initial project plan—schedule with main milestones and cost estimates. If the inception phase lasts for too long, it is like an indicator stating that the project vision and goals are not clear to the stakeholders. With no clear goals and vision the project most likely is doomed to fail. At this scenario it is better to take a pause at the very beginning of the project to refine the vi-

sion and goals. Otherwise it could lead to unnecessary make-overs and schedule delays in further phases.

- Elaboration : during this phase the project team is expected to capture a majority of system's requirements, to perform identified risk analysis and make a plan of risk management to reduce or eliminate their impact on final schedule and product, to establish design and architecture, to create a plan for the next phase.

- Construction : the longest and largest phase within Unified Process. During this phase, the design of the system is finalized and refined and the system is built using the basis created during elaboration phase. The construction phase is divided into multiple iterations, for each iteration to result in an executable release of the system. The final iteration of construction phase releases fully completed system which is to be deployed during transition phase

- Transition : the final project phase which delivers the new system to its end-users. Transition phase includes also data migration from legacy systems and user trainings

Each phase and its iteration consists of a set of predefined activities. The Unified Process describes work activities as disciplines—a discipline is a set of activities and related artifacts in one subject area . The disciplines described by Unified Process are as follows :

- Business modeling : domain object modeling and dynamic modeling of the business processes

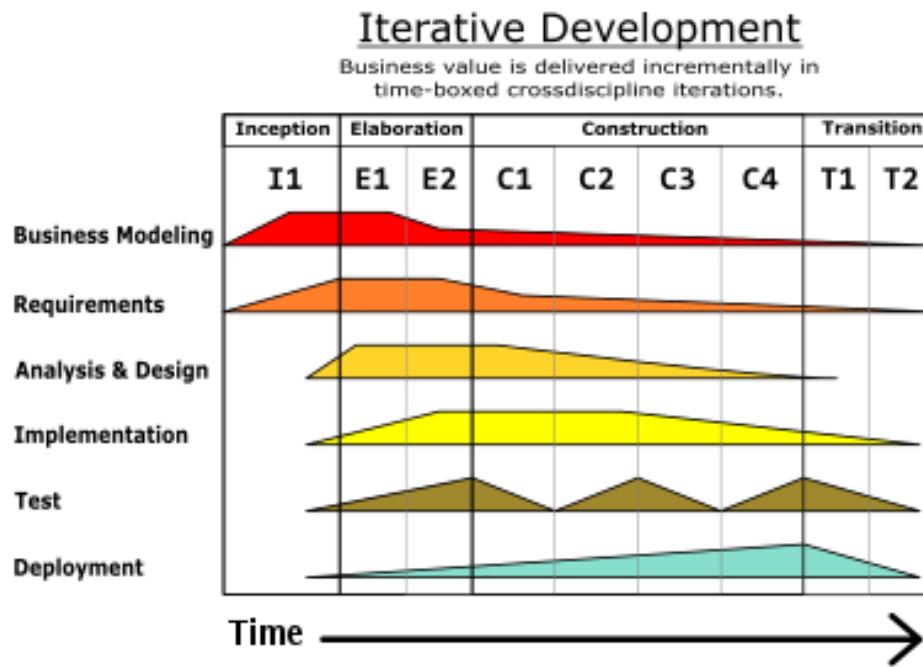
- Requirements : requirements analysis of system under consideration. Includes activities like writing use cases and identifying nonfunctional requirements

- Analysis and design : covers aspects of design, including the overall architecture

- Implementation : programming and building the system (except the deployment)

- Test : involves testing activities such as test planning, development of test scenarios, alpha and beta testing, regression testing, acceptance testing

- Deployment : the deployment activities of developed system. The disciplines and phases of Unified Process are given in Fig.2 where the phases are columns and the disciplines are rows. It clearly shows that the relative effort across disciplines changes over time from iteration to iteration .



I.5 Conclusion :

In this chapter, we presented our plateforme General presentation, then we have identified the study and the critical of exciting followed by the definition of define the problem and the proposed solution and finally we have explained the Methodological choice.

Chapter II

Requirements Analysis

II.1 Introduction

This chapter is devoted to the identification of the functional and the non-functional needs in a first step. Then we will focus on identifying the actors and their use cases. At the end of this chapter, we will detail the realization of the diagrams and the physical Architect.

II.2 Specification of Requirements

II.2.1 Specification of Functional Requirements :

Functional Requirements: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

- Account management :

Account management is the practice of providing customers with service, support and improvement opportunities to increase their consumption of the service

- Subscriptions management :

Subscription management is the process of managing your customers' subscriptions and making sure that their experience with the application is a happy one. The process starts once a customer "subscribe", and it ends when the customer cancels their subscription.

- Interactive statistics :

Interactive analytics refers to the use of visual analytics and master data management systems to process massive, unstructured datasets, analyze, visualize, and model data via interactive analytics dashboards in real-time.

- Publications management :

A publications management is responsible for the production of print and online publications. ... Publications managers are responsible for the creation of technical documents, pictures , status , business proposals, and a wide variety of written material

- Ad management

Advertising management is a planned managerial process designed to oversee and control the various advertising activities involved in a application to communicate with a firm's target market

- Chat

One of the features of the application is Chat. Chat is a conversation between two persons inside an app.

- Training management:

first of all the admin is going to authenticate, then he's going to add a train and reserve a location and materials for this train or update An existing one also he can delete it, then he can add a trainer for every single train at the end of the train the user can comment the train and rate the trainer through a form

- Certificate management :

In the back office application, the admin could manage the Certification of the participant.

- Job management :

In the front-office application, we have created an interface that manage job opportunities.

- Candidate management :

In the back-office application, the job application could be managed and filtered.

- Charity event management

in back-office application we have created an interface that manage events And filter them and publish the events.

- Complaints management

In the front-office application any user could add a complaining request.

- Appointments booking

In the front-office application, we have created an interface that allowed the users to take an appointment with experts.

II.2.2 Specification of Non-Functional Requirements :

Non-functional requirements: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

Ergonomic Requirements

The ergonomic constraints are the constraints related to the adaptation between the application functionalities, their interfaces and their use. For our application, we must obey the following ergonomic constraints :

- Provide quick access to information.
- Simple and understandable interface(A modern graphical interface).
- The organization of headings, tabs,...
- The application must guide the customer to have his request, that is to say that it must be developed with an understanding language by the user, present the information in a simple and clear way, display the customer's choices or input
- Any interface of the application must be homogeneous, indeed, the different pages must follow the same model of representation (colors, images, scrolling texts, ...).

Technical Requirements

Maintainability:

- The code must be extensible and maintainable to facilitate any improvement or optimization.

Security:

- You must have an authentication interface and control access to the various features.

Performance and efficiency:

- response time : application loading, screen opening and refresh times
- Processing time : functions, calculations,data imports/exports

Reliability :

- In a perfect world, a reliable piece of software is completely defect free, does not create downtime, and performs correctly in every scenario.

Scalability :

- the ability to handle a growing number of customers, clients, or users. Application scalability guarantees that, in time, the app will be capable of efficiently sustaining a high growth of its user base.

Responsive :

- Responsive Web design is a web design approach that aims to develop sites that provide an optimal reading and browsing experience for the user regardless of their range of devices (mobile phones, tablets, e-readers, desktop monitors).

II.3 Identification of actors :

Un acteur est une personne physique ou morale prenant part ou affecté par l'action ou le projet en question. Il faut donc commencer par bien préciser par rapport à quelle action ou suite d'actions on cherche à déterminer qui sont et ce que sont les acteurs.

in our application we have the following actors :

- Administrator
- EXPERT(PSY, Lawyer, Doctor)
- Women
- Company

II.4 Identification of use cases :

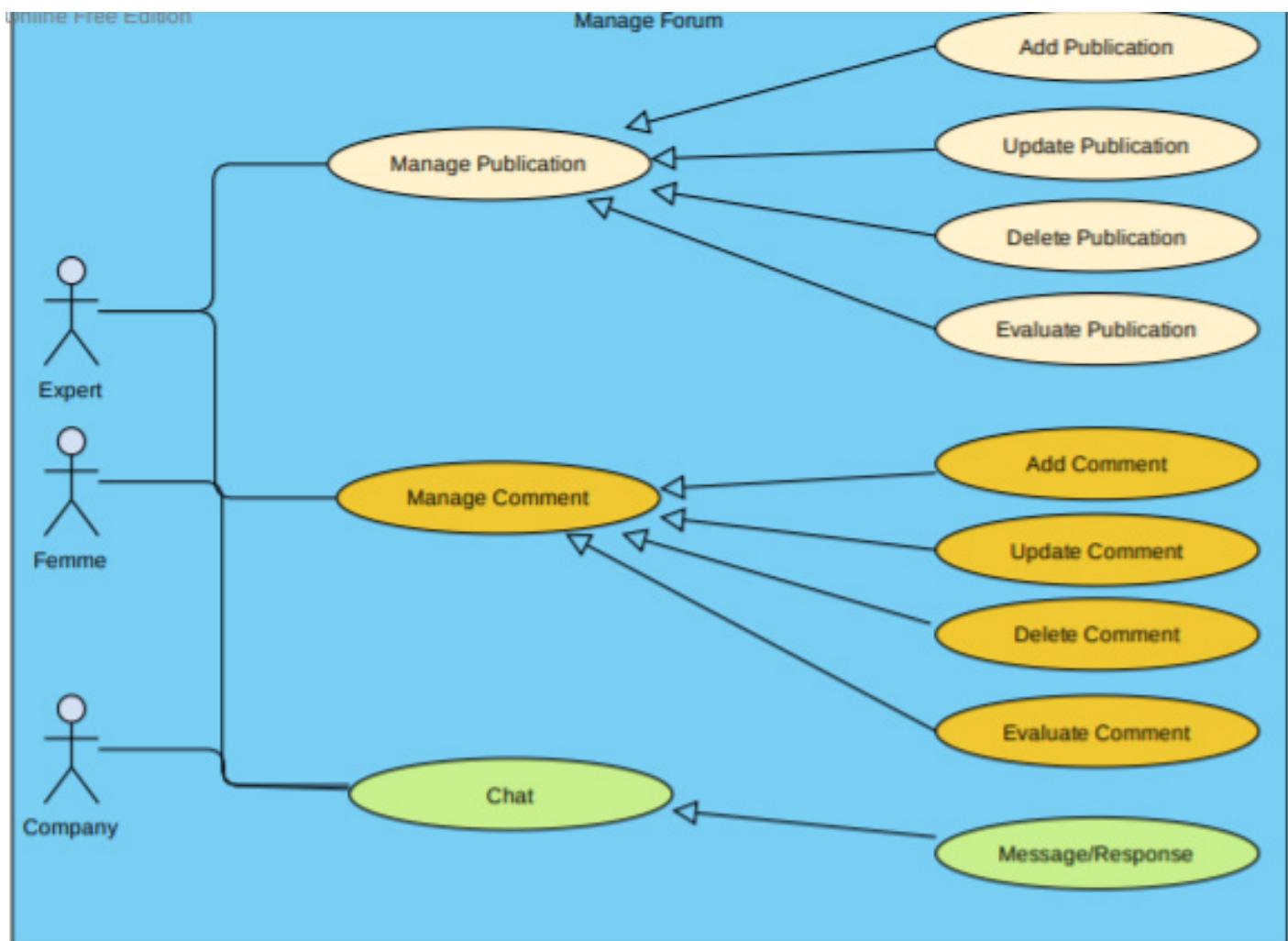


Figure II.1: Forum Use Case

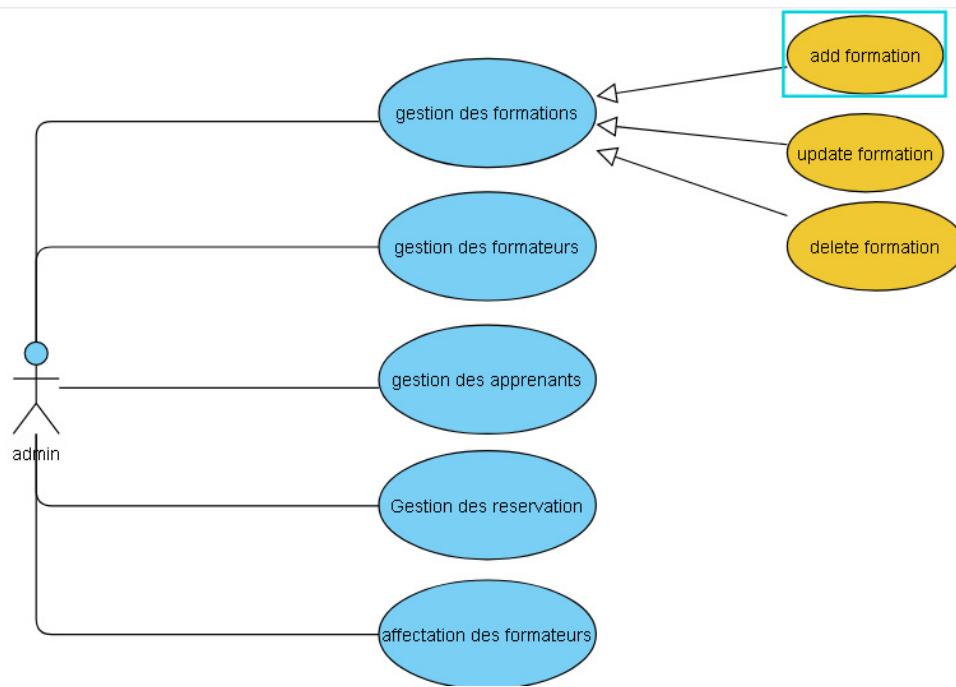


Figure II.2: Training Use Case

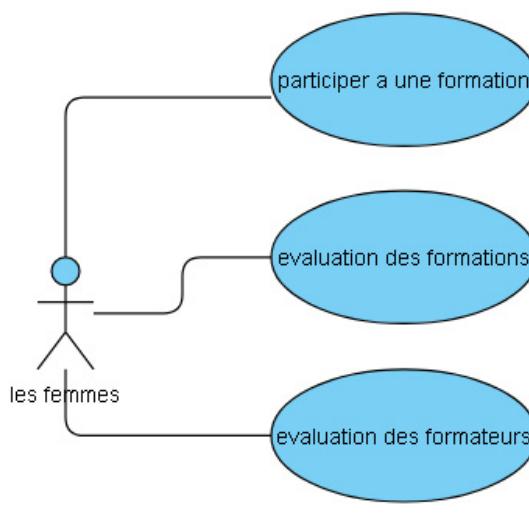


Figure II.3: Training Use Case

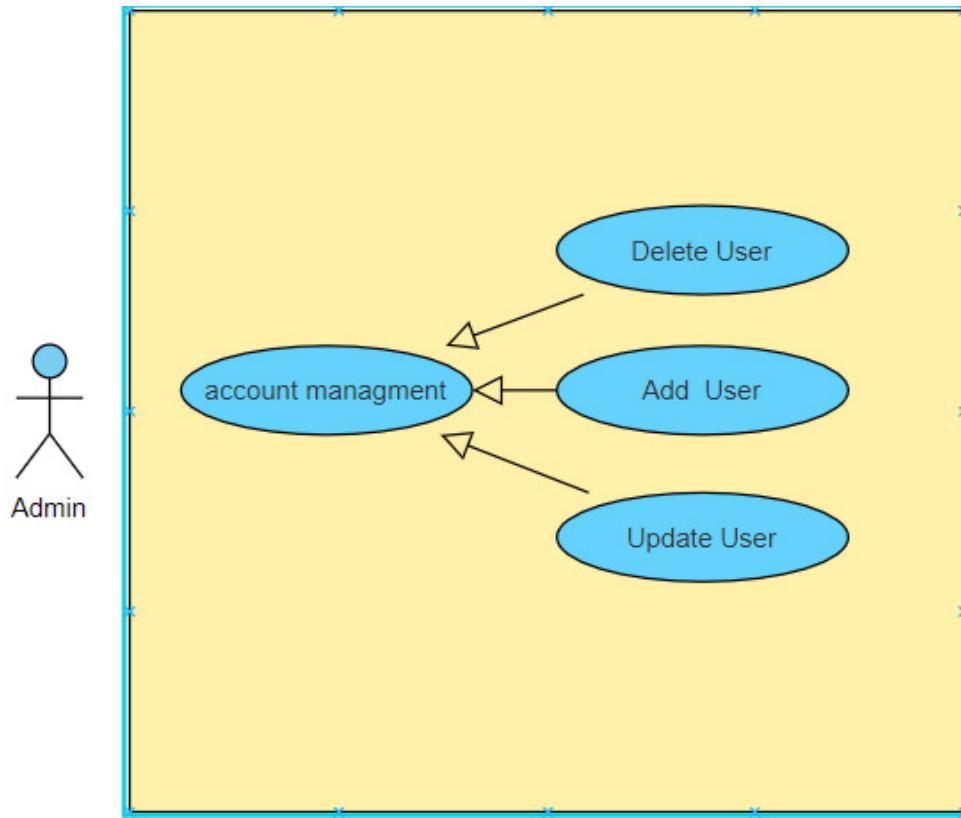


Figure II.4: User Use Case

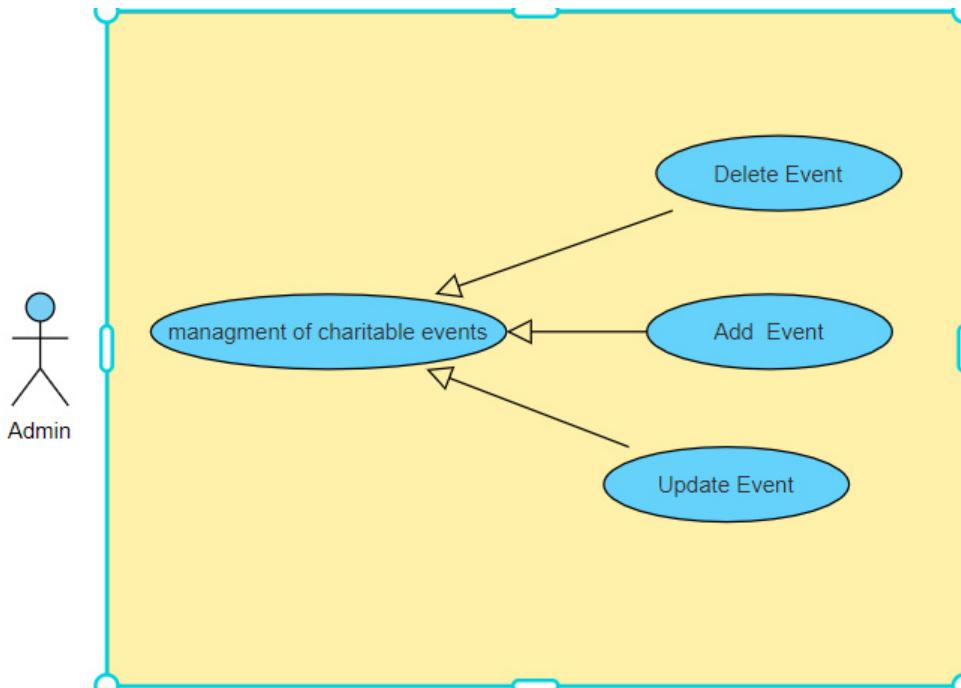


Figure II.5: Events Use Case

II.5 Use Case Diagram

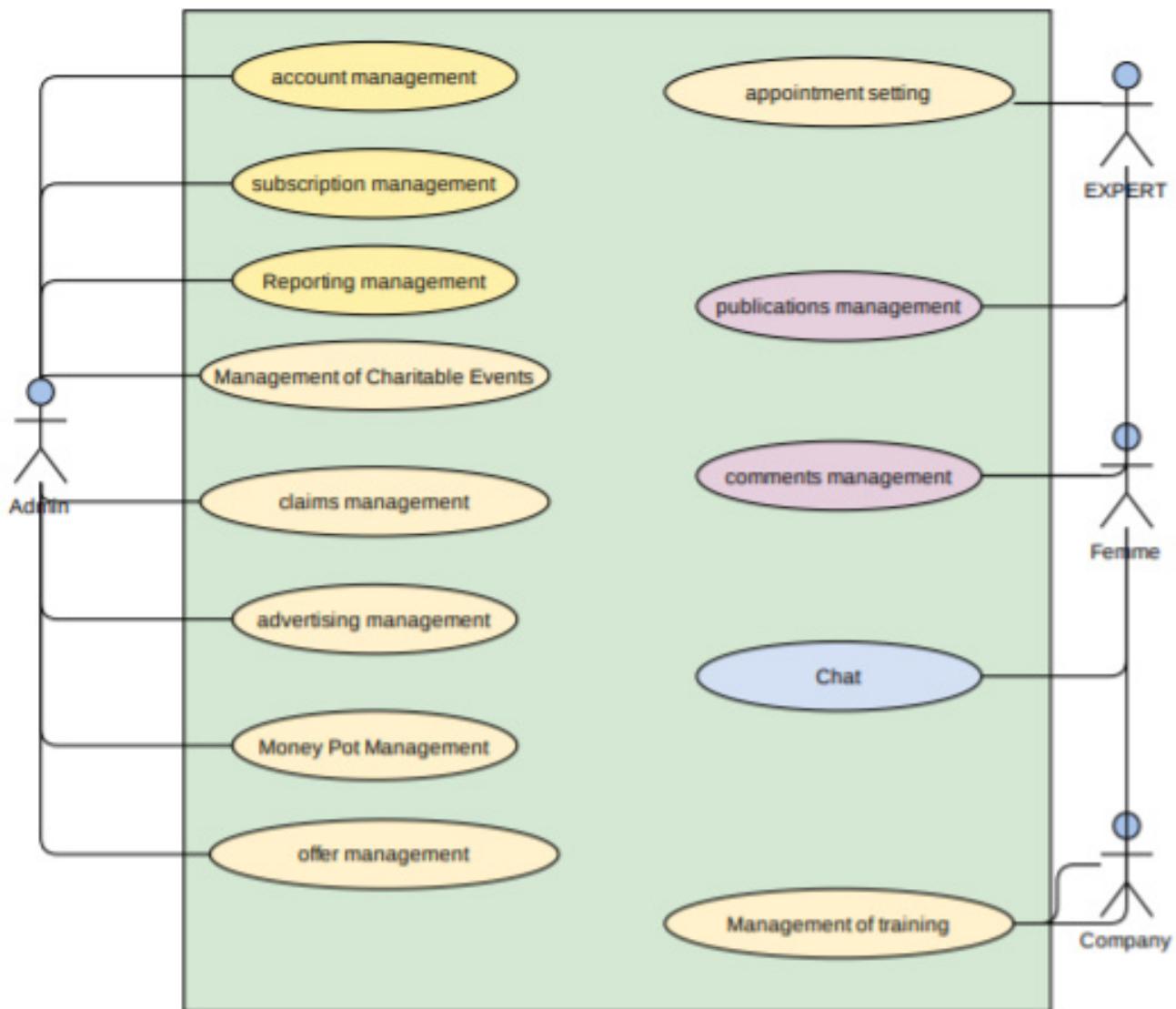


Figure II.6: Use Case Diagram

II.6 Class Diagram

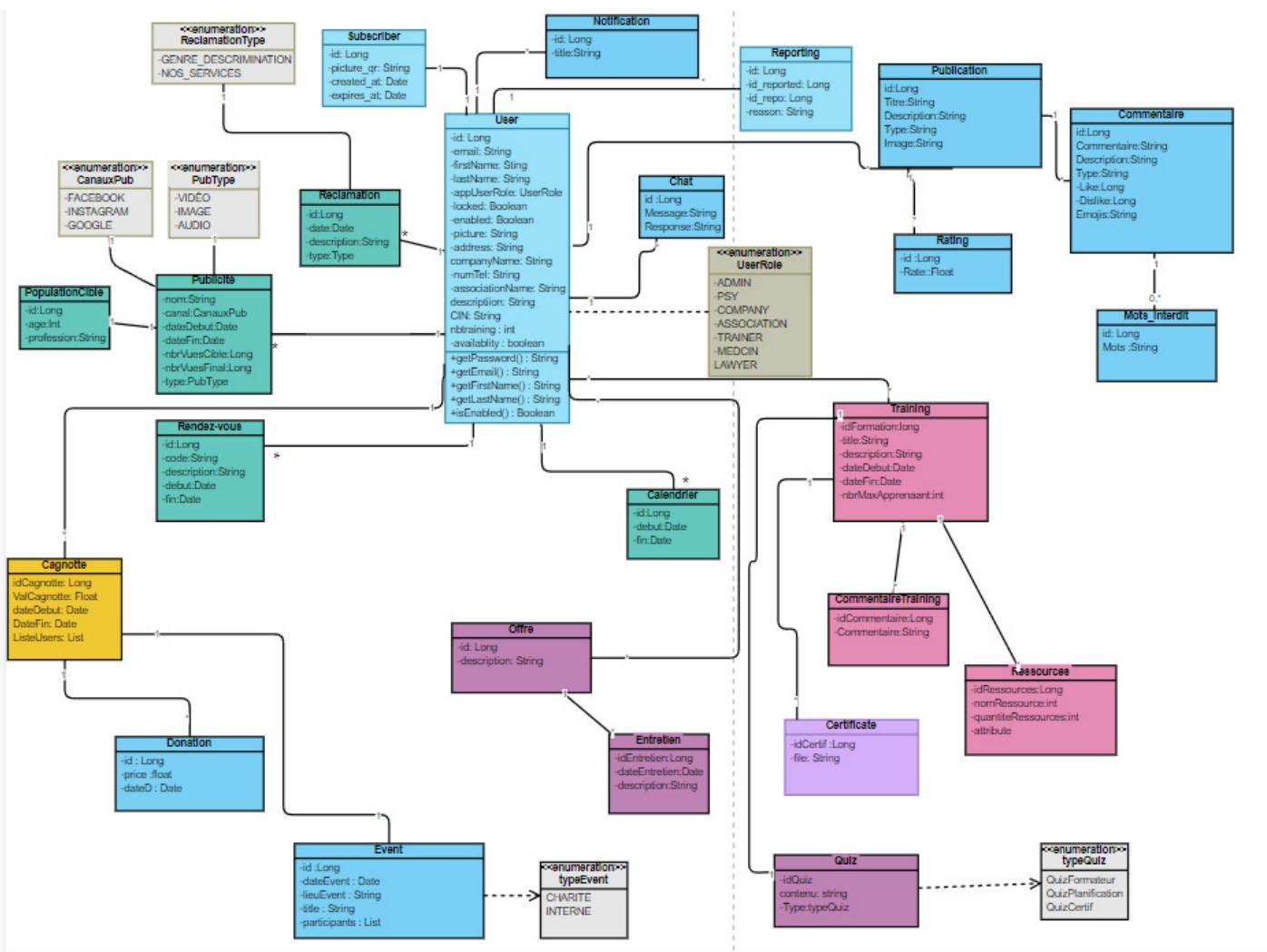


Figure II.7: Class Diagram

II.7 System Sequence Diagram

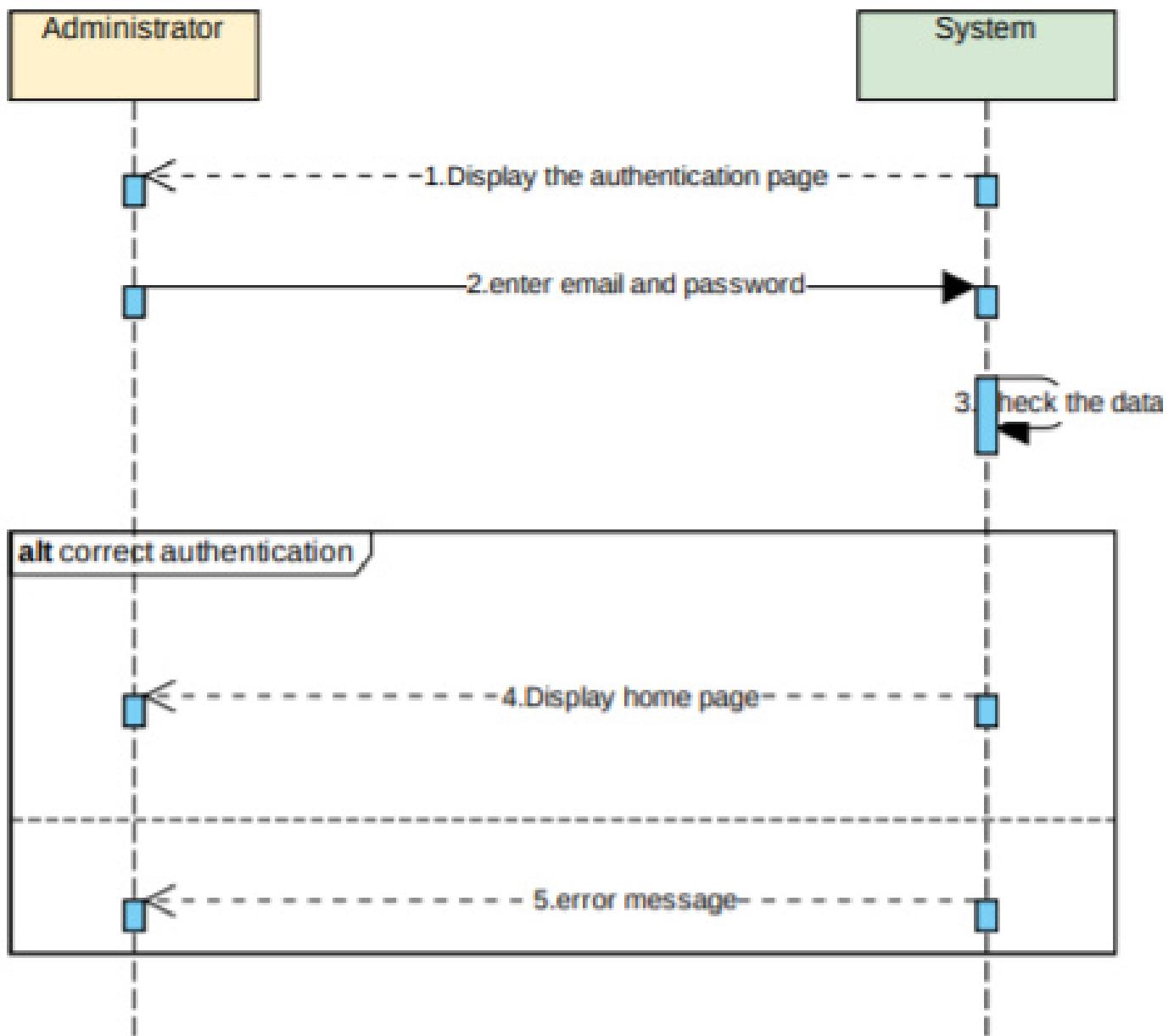


Figure II.8: System Sequence Diagram Authentication

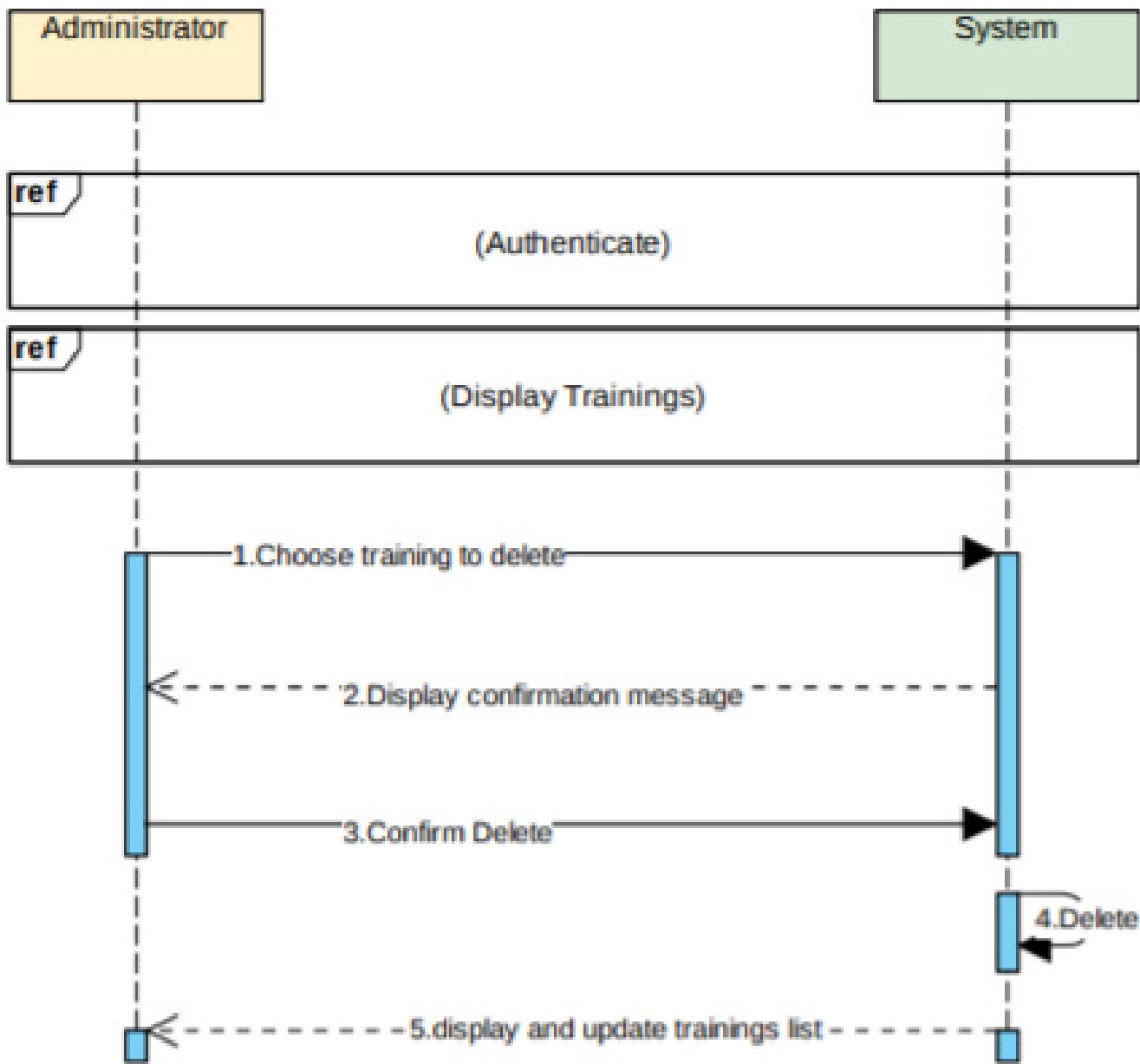


Figure II.9: System Sequence Diagram Delete Training

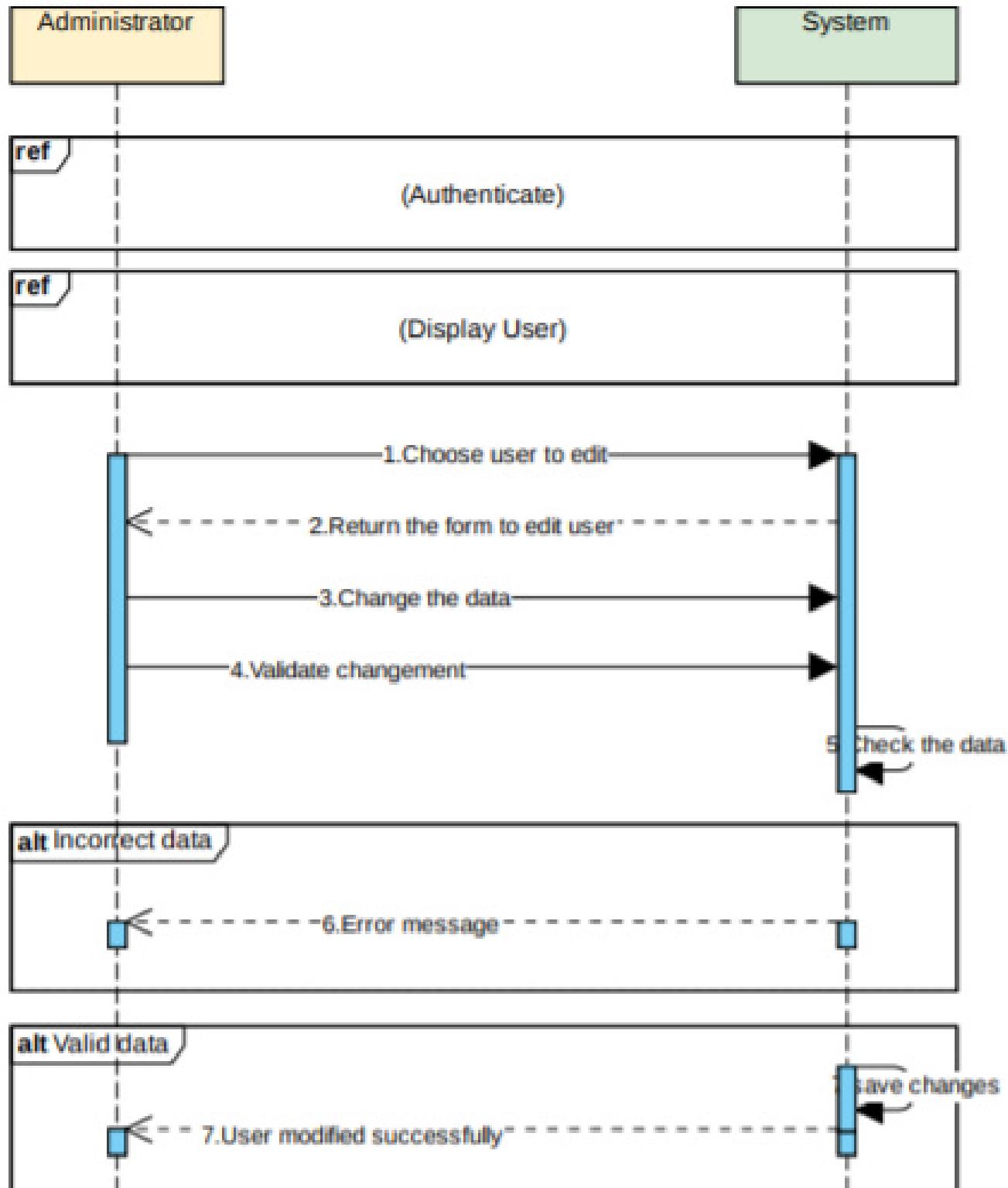


Figure II.10: System Sequence Diagram Edit User

II.8 Physical architecture

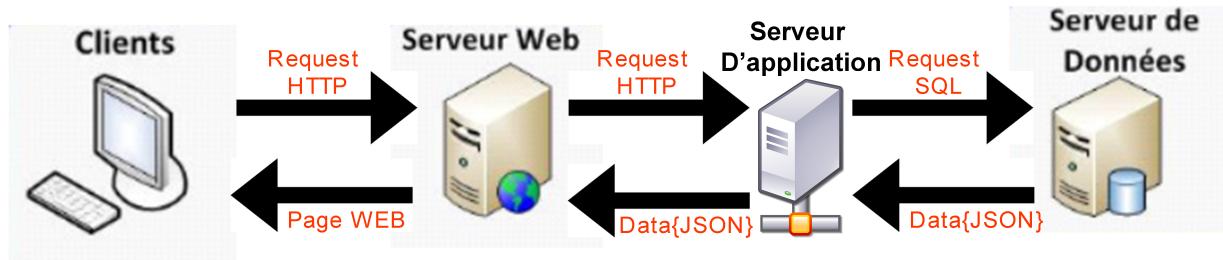


Figure II.11: an overview of the application Physical architecture

II.9 Conclusion :

In this chapter, we defined the functional and the non-functional requirements of our application, the role of the users. Then, we prepared the diagrams as well as the the physic Architect.

Chapter III

Conception

III.1 Introduction

In this chapter, we present a detailed Use case, sequence and package Diagrams with a coherent scenarios of various management. Then we found the Operational and the logic architecture. Finally we finished with the static Aspect of our project.

III.2 Refinement and detailed description of use cases

III.2.1 Text description of use cases

Actor	User
Goals	a User has the right to manage his own publication
Precondition	a user must be authenticated
Post-condition	
Nominal Scenario	<ul style="list-style-type: none"> -The system shows a list of publications -a User can add a new publication -The User selects his publication -The User chooses one publication -The system displays the modification interface <ul style="list-style-type: none"> (a) The User can modify his publication (b) The User writes down the new data (c) The system saves the new data -The User choose one publication <ul style="list-style-type: none"> (a) The User like/dislike the publication (b) The system saves the new data -The User delete his own publication selected
Alternative scenario	Invalid data

Manage Publications

Figure III.1: Manage Publications Scenario

Actor	User
Goals	a User has the right to modify a publication
Precondition	a user must be authentified
Post-condition	Update Publication
Nominal Scenario	<ul style="list-style-type: none"> -The User asks to display the list of publication -The system returns the list -The User selects his publication -The User requests to modify the selected publication -The system displays the modification interface -The User updates the publication and validates the request -The system checks the data entered by the user
Alternative scenario	Invalid data

Manage Publications

Figure III.2: Edit Publications Scenario

Actor	User
Precondition	the former must be connected to the server and authentified
Post-condition	Application sent
Nominal Scenario	<ul style="list-style-type: none"> -The user asks to consult a training -The system shows a list of trainings -The user chooses a train to participate in -The user can : 1-comment the training 2-rate the former
Alternative scenario	Application closed

Figure III.3: Training Scenario

Actor	Admin
Goals	Admin has the right to manage users
Precondition	Administrator must be connected to the server and authentified
Post-condition	
Nominal Scenario	<ul style="list-style-type: none"> -The system shows a list of users -administrator can add a new user -administrator chooses one user <ul style="list-style-type: none"> (a) administrator can modify a user (b) administrator confirms the modification (c) the system saves the new data confirms the modification -administrator can delete user <ul style="list-style-type: none"> (a) the system asks for to confirm the process (b) administrator confirms
Alternative scenario	

Manage Users

Figure III.4: Manage Users Scenario

Actor	User
Goals	a User has the right to report another user
Precondition	a user must be authentified
Post-condition	Report User
Nominal Scenario	<ul style="list-style-type: none"> -The User report another user -The system check the number of report -The system block the publication if the report number is 100 -The system add one to the number report if it's less than 100 -The system save changes
Alternative scenario	Invalid data

Manage Reports

Figure III.5: Manage Reports Scenario

Actor	User
Goals	User has the right to apply to a job
Precondition	A member must be authentified
Post-condition	Application sent
Nominal Scenario	<ul style="list-style-type: none"> -The user asks to consult a job offer -The system shows a list of offers -The user chooses an offer to apply in -The system shows a form of application -The user can : <ul style="list-style-type: none"> 1-Fill the form 2-Cancel the application *Once the user sent her application he couldn't cancel it or delete it -The user submit the application -The system asks for confirmation -The user confirm -The system commits
Alternative scenario	Application closed

JOB APPLICATION

Figure III.6: JOB APPLICATION Scenario

Actor	Admin
Goals	Admin has the right to manage the events And donations
Pre-condition	Administrator must be connected to the server and authenticated
Post-condition	
Nominal scenario	<ul style="list-style-type: none"> -the system shows a list of events and pools -admin can add new event / pool -admin can choose one event / pool <ul style="list-style-type: none"> (a) admin can modify an event / pool (b) admin confirm / deny the modification (c) the system saves the new data -admin can delete an event / pool <ul style="list-style-type: none"> (a) the system asks for confirmation (b) admin confirm / deny
Alternative scenario	

Figure III.7: Manage Events

Actor	User
Goals	User can see an event and make a donation
Pre-condition	User must be authenticated
Post-condition	
Nominal scenario	<ul style="list-style-type: none"> -the system shows a list of events and pools -User can read about the event -User can make a donation (a) system take the user to a payment page (b) user make the payment
Alternative scenario	Invalid data

Figure III.8: Manage Donation

III.2.2 Sequence diagram

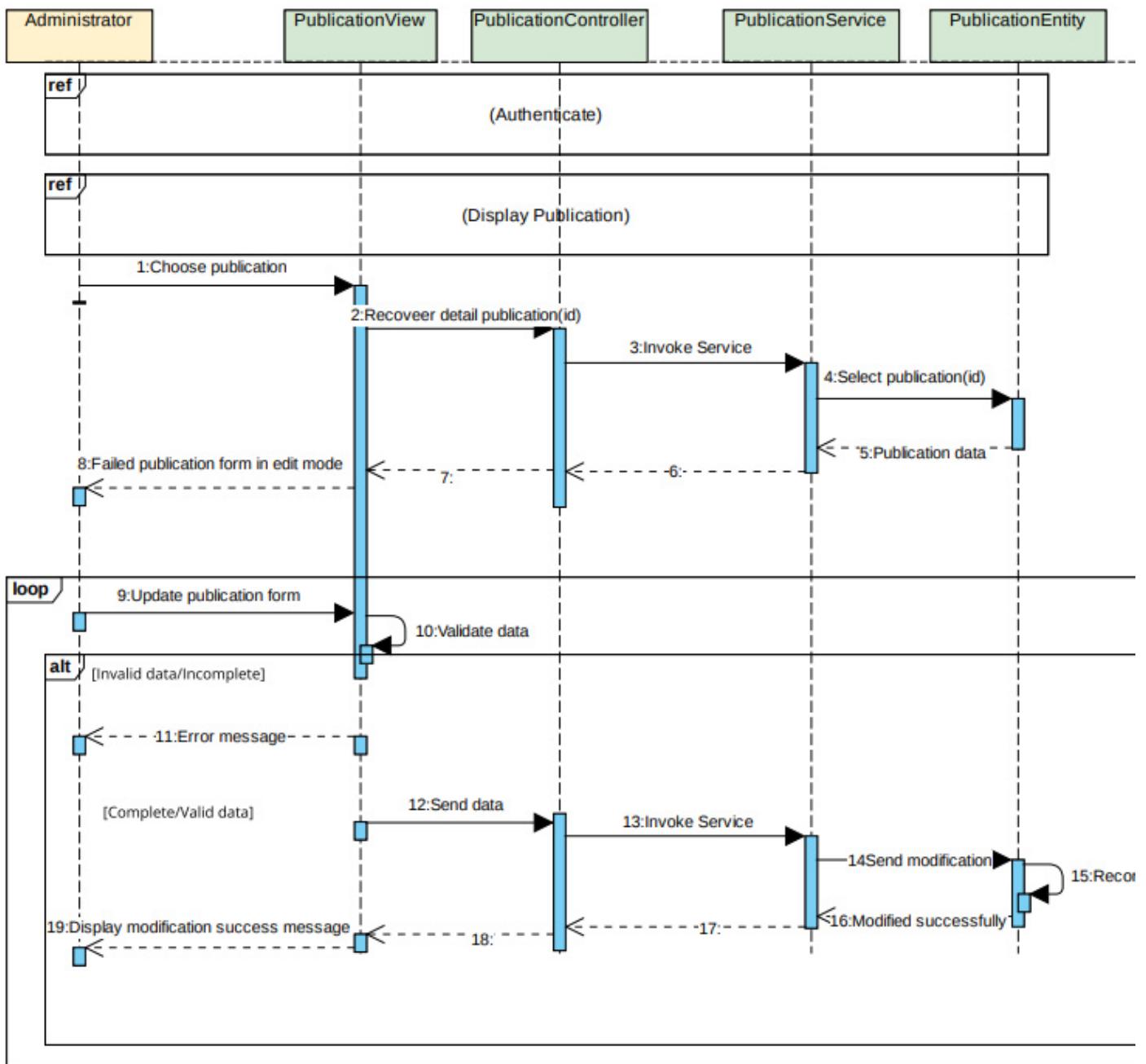


Figure III.9: Sequence diagram Update Publication

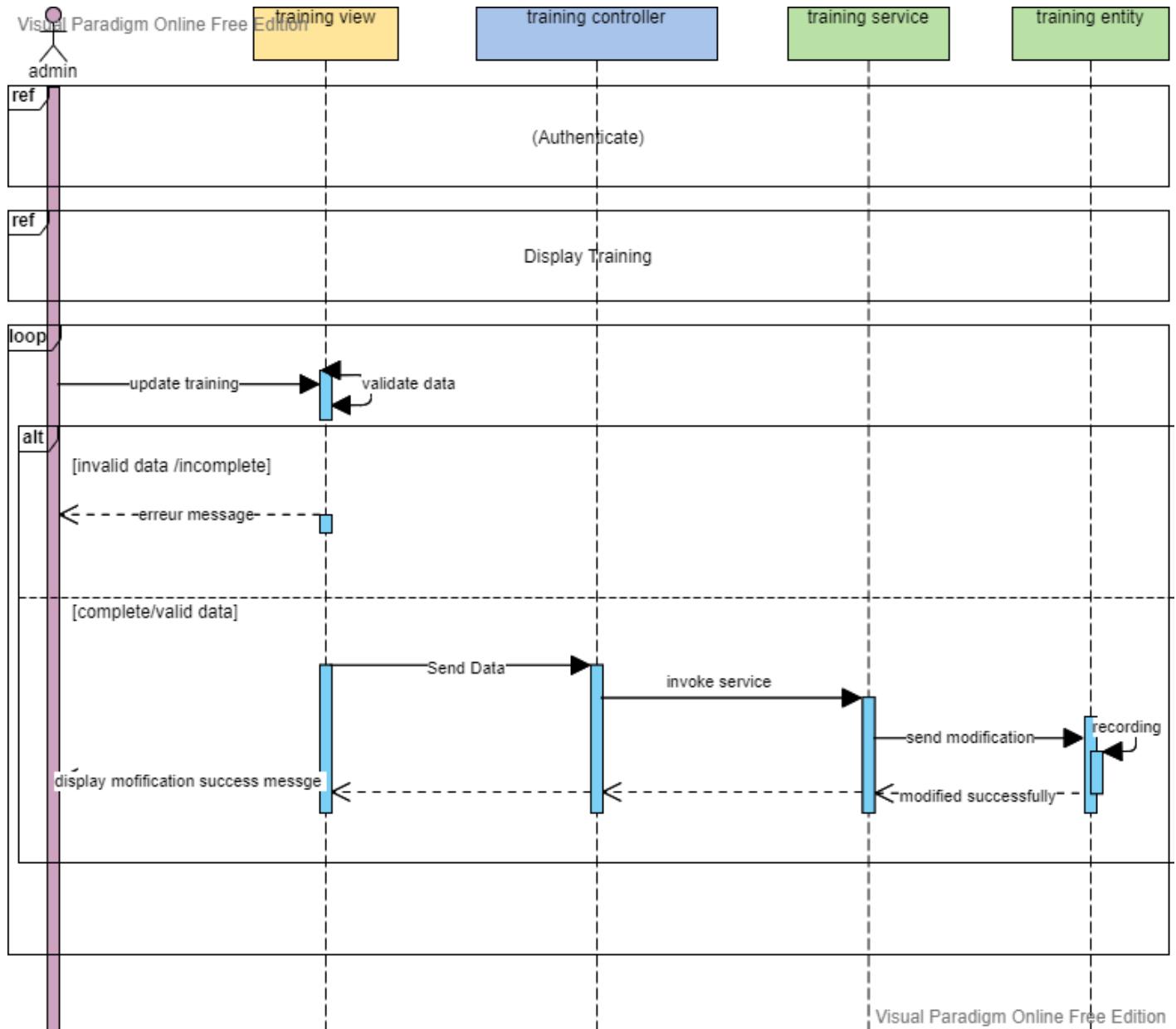


Figure III.10: Sequence diagram Update Training

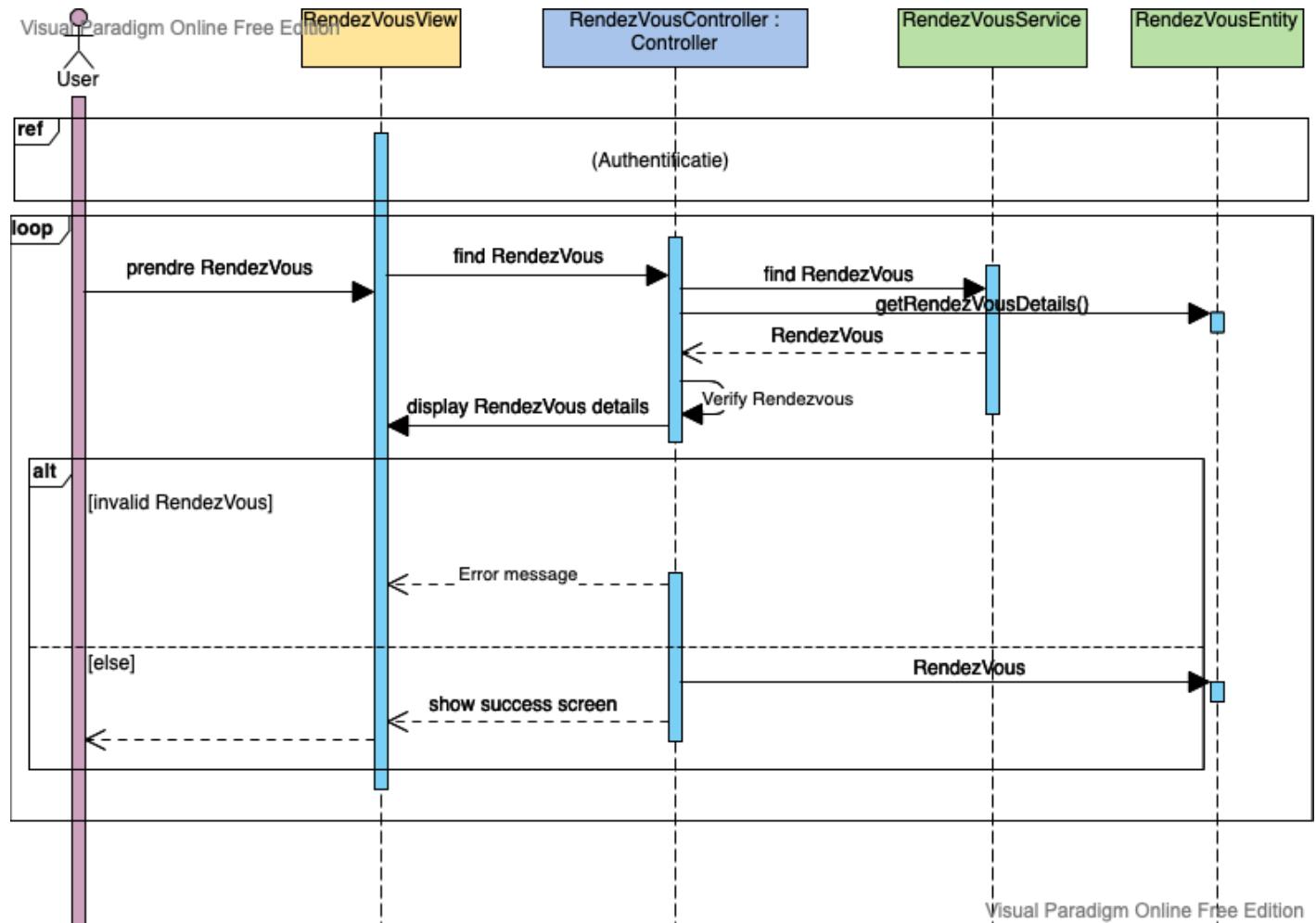


Figure III.11: Sequence diagram RDV

III.3 Conception class diagram

III.3.1 Package class diagram : Training

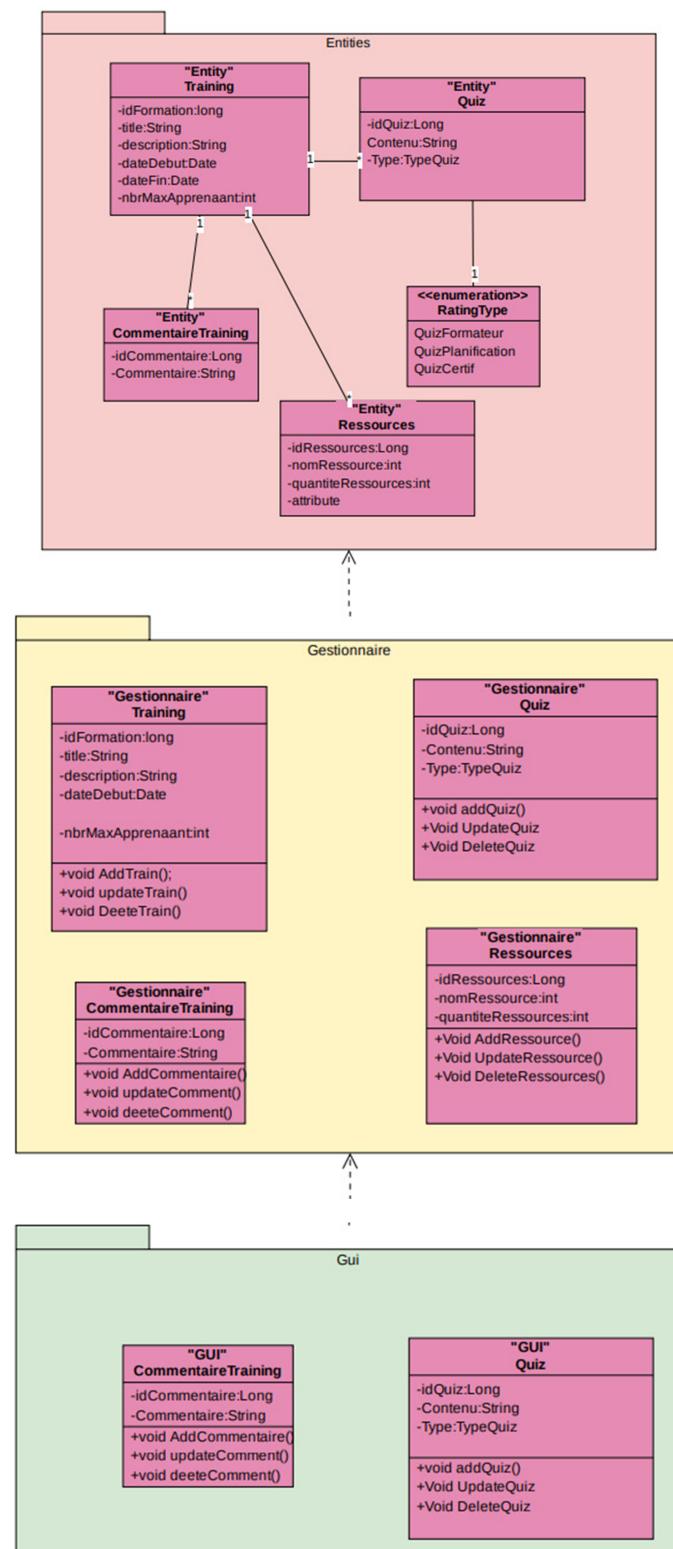


Figure III.12: Package class diagram : Training

III.3.2 Package class diagram : User

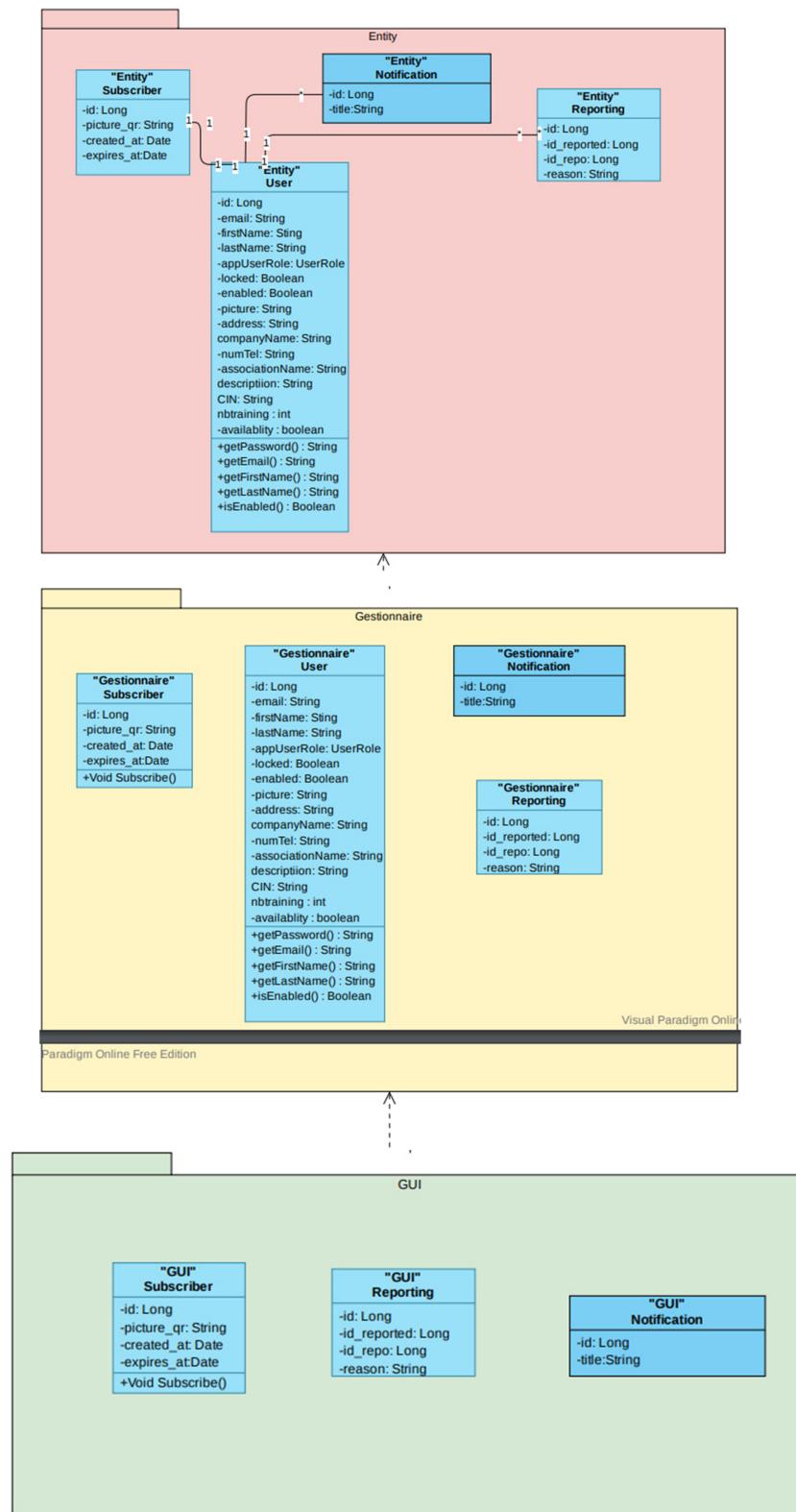


Figure III.13: Package class diagram : User

III.3.3 Package class diagram : Forum

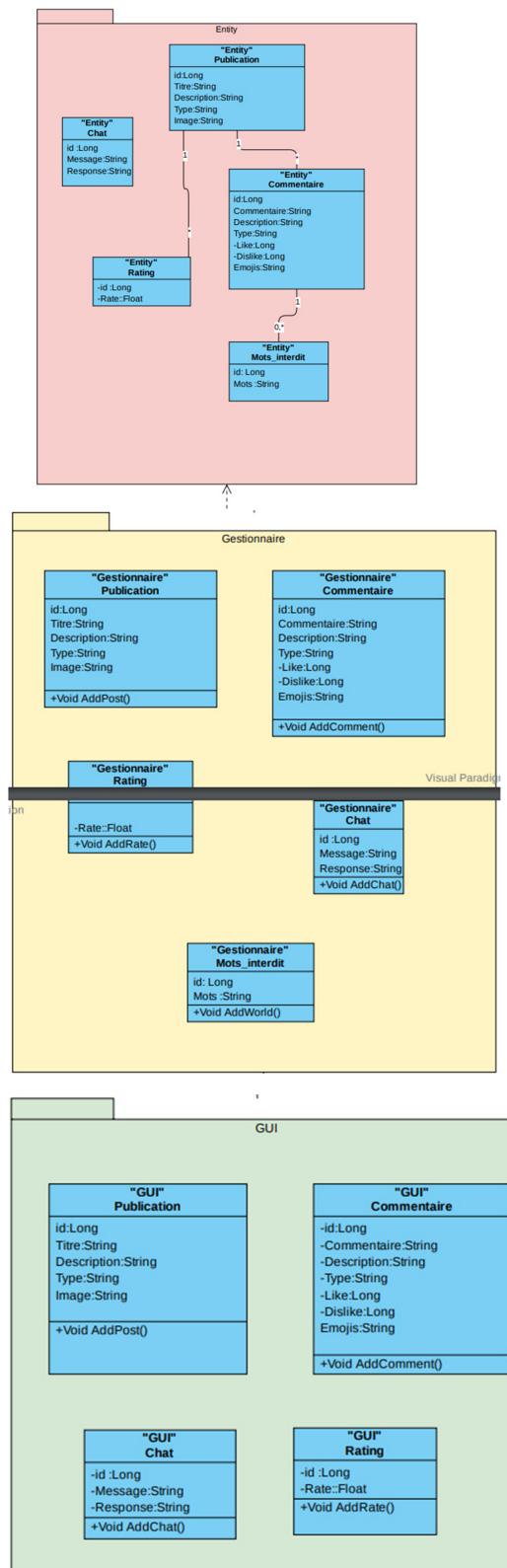


Figure III.14: Package class diagram : Forum

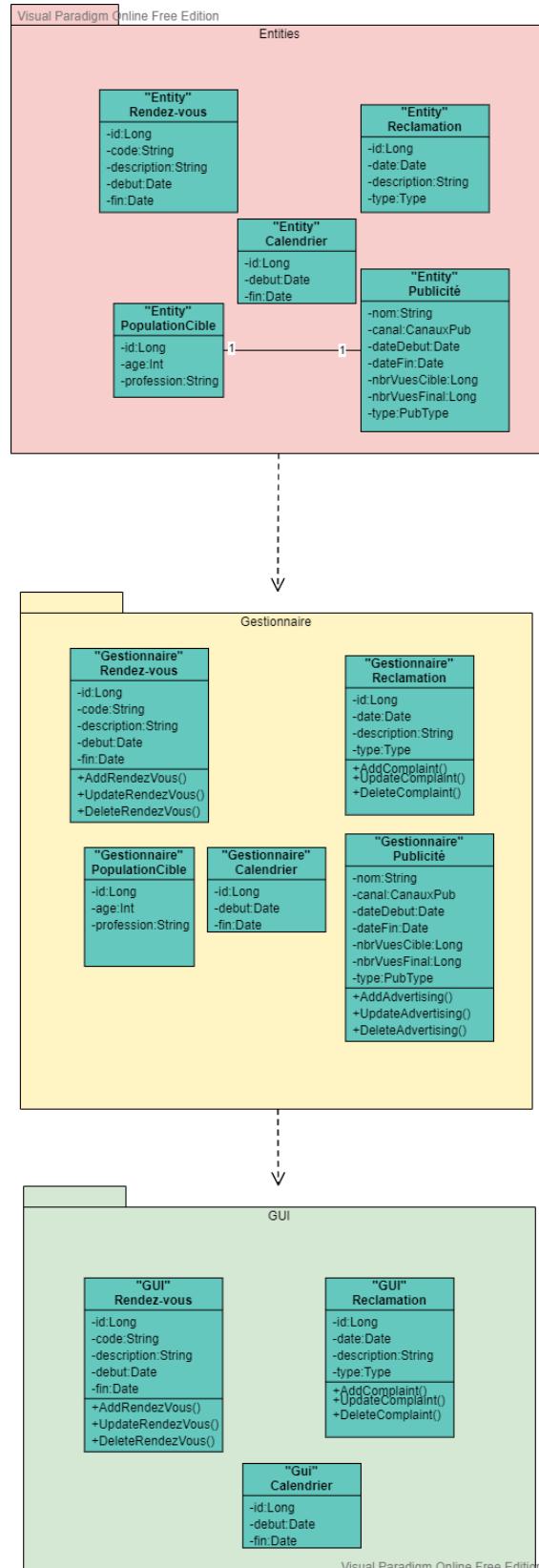


Figure III.15: Package class diagram : RDV

III.4 Global solution architecture

III.4.1 Operational architecture

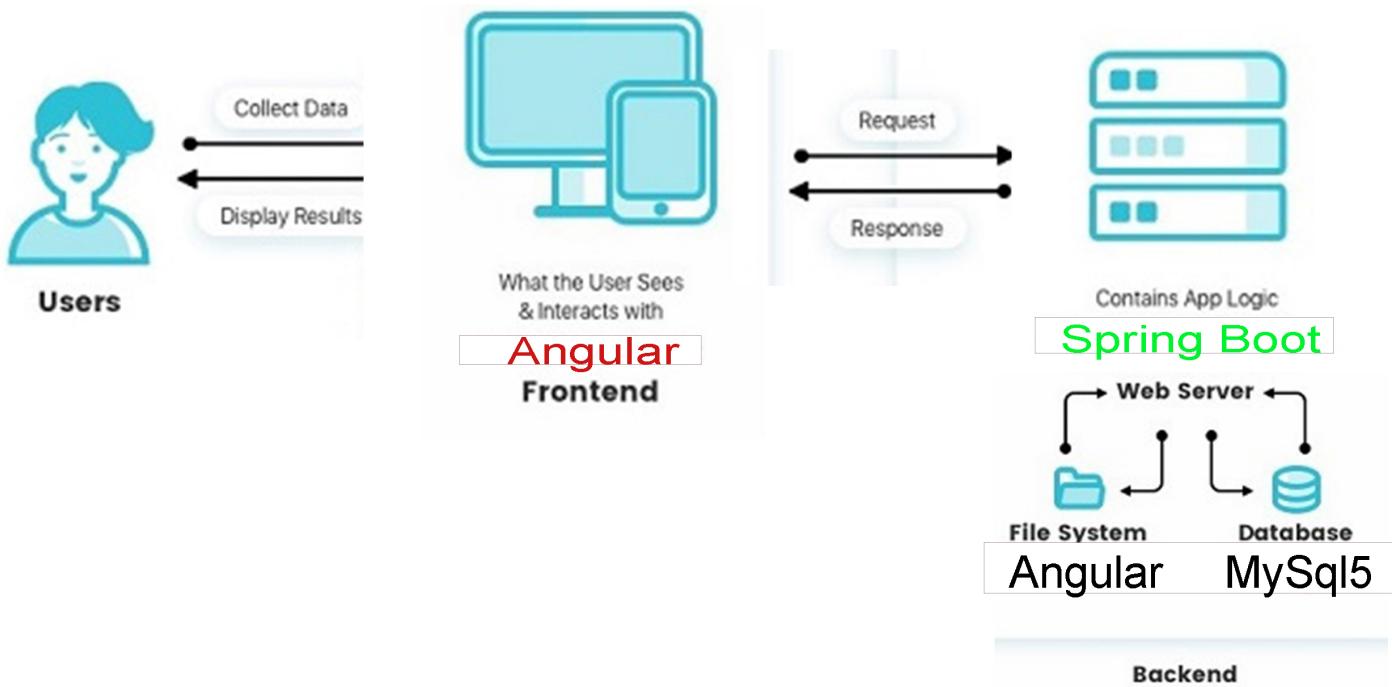


Figure III.16: an overview of the application Operational architecture

III.4.2 Logical architecture

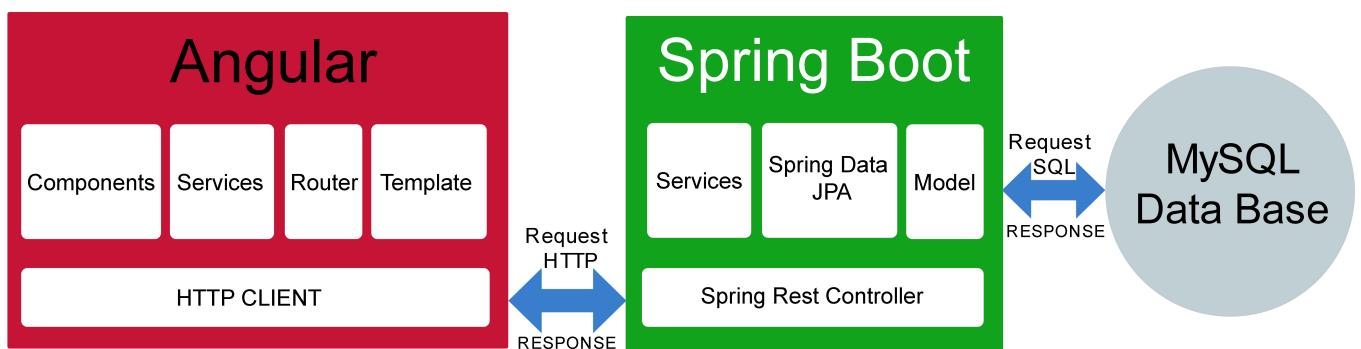


Figure III.17: an overview of the application Logical architecture

III.5 Static Aspect

III.5.1 Deployment diagram

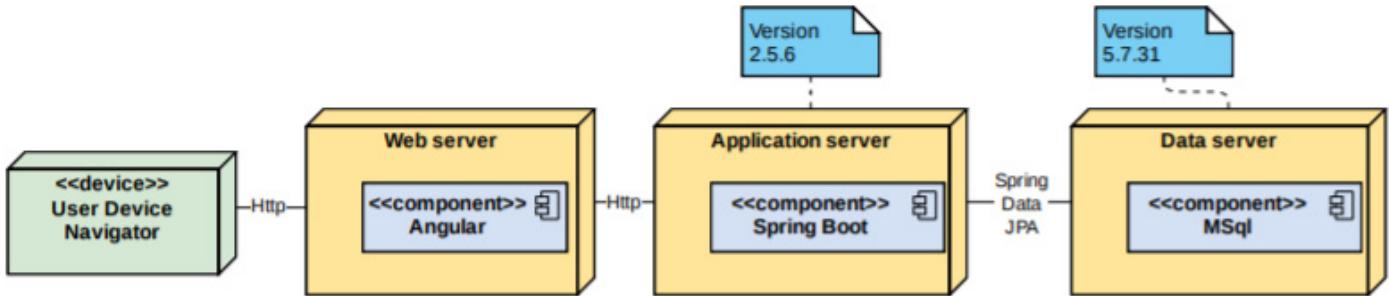


Figure III.18: Deployment diagram

III.6 Conclusion :

In this chapter, we have presented the main diagrams with all the expected scenarios. We have prepared the Operational, the logic architecture and the static Aspect of our project.

Chapter IV

Implementation

IV.1 Introduction

In this chapter we describe the implementation phase. In the first paragraph we describe the hardware and software development environment.

IV.2 Framework and technical tools

IV.2.1 Spring boot framework



Figure IV.1: Spring Boot

Why Spring Boot :

Reduces the time spent on development and increases the overall efficiency of the development team.

Helps to autoconfigure all components for a production-grade Spring app.

Facilitates the creation and testing of Java-based applications by providing a default setup for unit and integration tests.

Helps to avoid all the manual work of writing boilerplate code, annotations, and complex XML configurations.

Comes with embedded HTTP servers like Jetty and Tomcat to test web applications.

The integration of Spring Boot with the Spring ecosystem which includes Spring Data, Spring Security,

Spring ORM, and Spring JDBC is easy.

Provides many plugins that developers can use to work with embedded and in-memory databases smoothly and readily.

Allows for easily connecting with database and queue services like Oracle, PostgreSQL, MySQL, MongoDB, Redis, Solr, ElasticSearch, Rabbit MQ, ActiveMQ, and many more.

Provides admin support – you can manage via remote access to the application.

Eases the dependency and comes with Embedded Servlet Container.

Offers flexibility in configuring XML configurations, Java Beans, and Database Transaction.

Offers easy access to Command Line Interface which makes the development and testing of Spring Boot apps built with Java or Groovy agile.

IV.2.2 Angular framework



Figure IV.2: Angular Advantages

Why Angular :

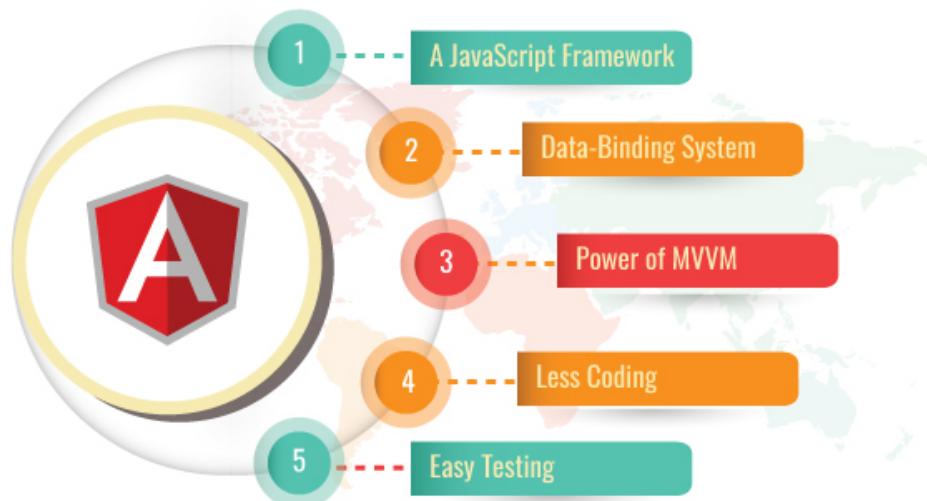


Figure IV.3: Job Market Angular

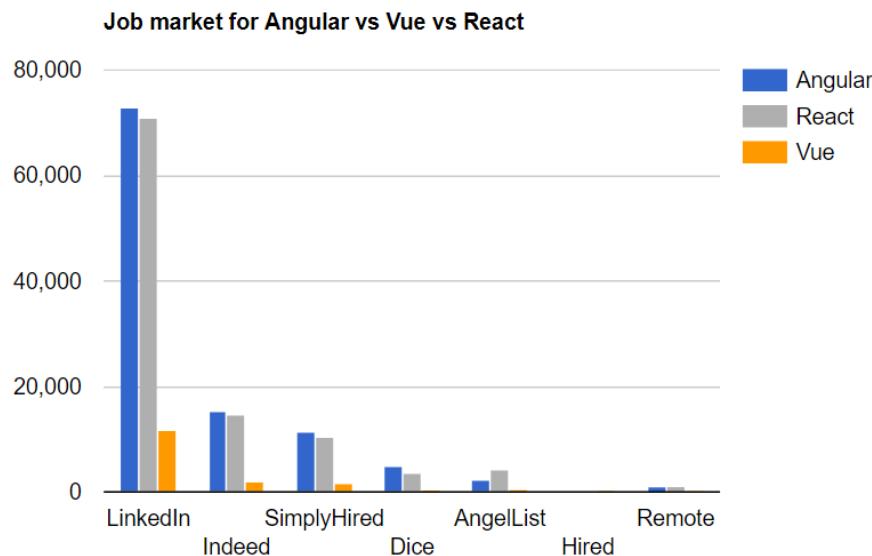


Figure IV.4: Learning Curve Angular

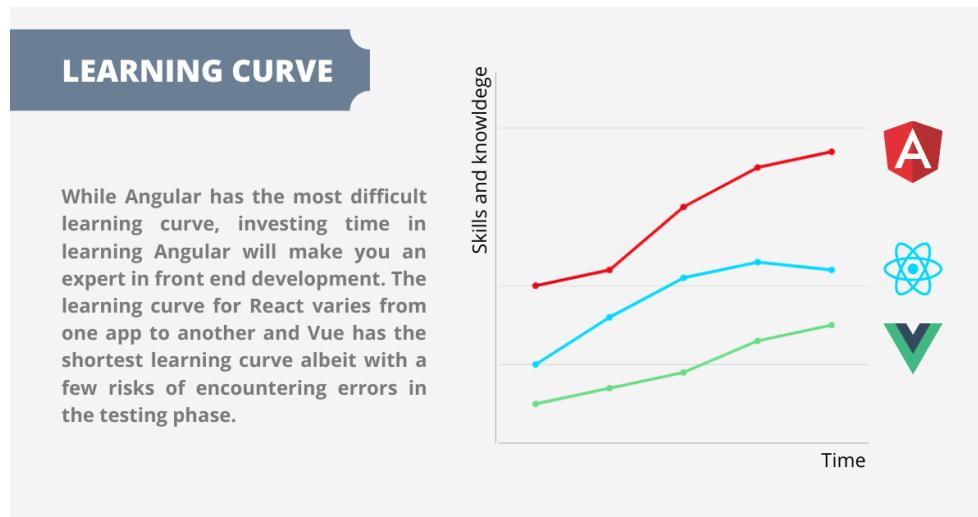


Figure IV.5: Angular

IV.2.3 Data mining

DATA MINING IS THE PROCESS OF SORTING THROUGH LARGE DATA SETS TO IDENTIFY PATTERNS AND RELATIONSHIPS THAT CAN HELP SOLVE BUSINESS PROBLEMS THROUGH DATA ANALYSIS. DATA MINING TECHNIQUES AND TOOLS ENABLE ENTERPRISES TO PREDICT FUTURE TRENDS AND MAKE MORE-INFORMED BUSINESS DECISIONS. DATA MINING IS A KEY PART OF DATA ANALYTICS OVERALL AND ONE OF THE CORE DISCIPLINES IN DATA SCIENCE, WHICH USES ADVANCED ANALYTICS TECHNIQUES TO FIND USEFUL INFORMATION IN DATA SETS. AT A MORE GRANULAR LEVEL, DATA MINING IS A STEP IN THE KNOWLEDGE DISCOVERY IN DATABASES (KDD) PROCESS, A DATA SCIENCE METHODOLOGY FOR GATHERING, PROCESSING AND ANALYZING DATA. DATA MINING AND KDD ARE SOMETIMES REFERRED TO INTERCHANGEABLY, BUT THEY'RE MORE COMMONLY SEEN AS DISTINCT THINGS.

IV.3 Coding and test

IV.3.1 Spring Boot (Postman)

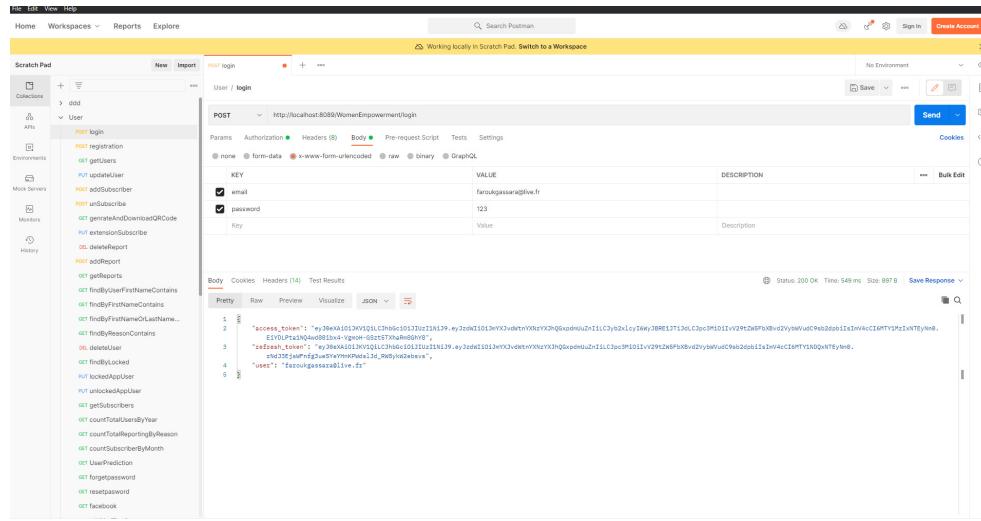


Figure IV.6: Login

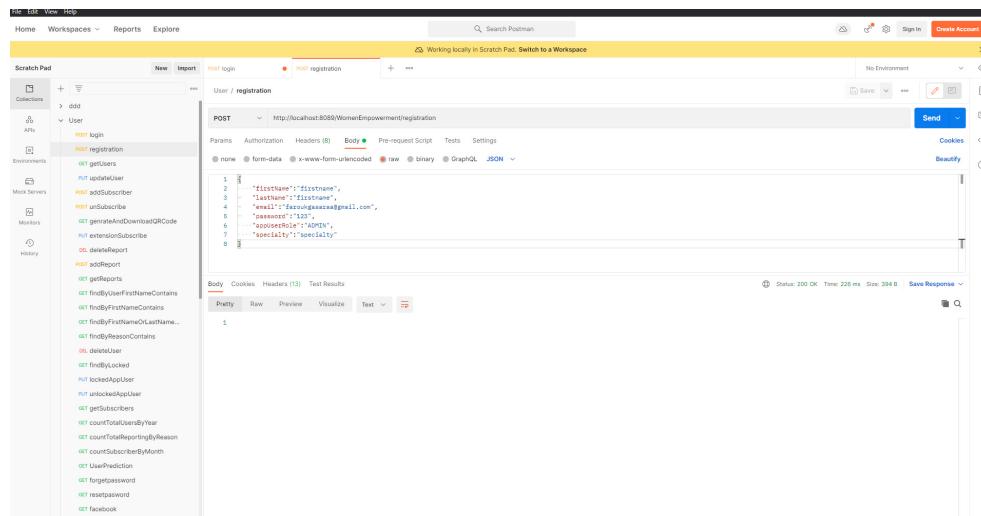


Figure IV.7: Sign up

```

1 {
2   "id": 2,
3   "firstname": "firstname",
4   "lastname": "lastname",
5   "email": "firstname.lastname@gmail.com",
6   "password": "1234567890",
7   "locked": false,
8   "lockedAt": null,
9   "lockedBy": null,
10  "lockedUntil": null,
11  "enabled": true,
12  "enabledAt": null,
13  "enabledBy": null,
14  "associations": null,
15  "reports": null,
16  "reportsAt": null,
17  "reportsBy": null,
18  "age": 0,
19  "reportsCount": null,
20  "reportsRate": null,
21  "birthDate": null,
22  "name": "John Doe"
}

```

Figure IV.8: Users List

```

1 {
2   "count": 2
}

```

Figure IV.9: Stats

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** http://localhost:8089/WomenEmpowerment/Training/add-training/loujein.hragua@esprit.tn
- Body (JSON):**

```

1  {
2   "title": "nouvelle formation",
3   "description": "nouvelle",
4   "dateDebut": "2022-01-01",
5   "dateFin": "2022-02-02",
6   "nbrMaxApprenant": "12",
7   "file": "louja"
8
9

```
- Headers:** (10) - Authorization (green dot), Headers (green dot), Body (green dot), Pre-request Script, Tests, Settings
- Body Options:** Pretty, Raw, Preview, Visualize, Text (dropdown), Copy (button)

Figure IV.10: Events

The screenshot shows a Postman interface with the following details:

- Method:** DELETE
- URL:** http://localhost:8089/WomenEmpowerment/Training/remove-training/21
- Body:** (8) - Authorization (green dot), Headers (green dot), Body (red dot), Pre-request Script, Tests, Settings
- Body Options:** None (radio button), form-data, x-www-form-urlencoded, raw (radio button), binary, GraphQL
- Message:** This request does not have a body
- Status:** Status: 200 OK | Time: 1066 ms | Size: 394 B | Save
- Body Options:** Pretty, Raw, Preview, Visualize, Text (dropdown), Copy (button)

Figure IV.11: Events

IV.3.2 Data mining (R Studio/Python)

WE VISUALIZED THE CLUSTERS ON A SCATTERPLOT THAT DISPLAYS THE FIRST TWO PRINCIPAL COMPONENTS ON THE AXES USING THE FIVZ_CLUSTER() FUNCTION:

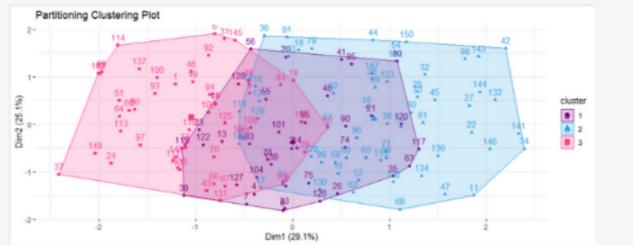


Figure IV.12: Login

```
# Evaluer les performances du modèle
y_pred_svc_1 = svm.predict(X_test)
print(classification_report(y_pred_svc_1, y_test))
fpr_svm, tpr_svm, threshold1 = roc_curve(y_test, y_pred_svc_1)
roc_auc_svm=auc(fpr_svm,tpr_svm)

C>          precision    recall  f1-score   support
          0       0.35      0.47      0.40       74
          1       0.61      0.48      0.53      126
          accuracy                           0.48      200
          macro avg       0.48      0.47      0.47      200
        weighted avg       0.51      0.47      0.48      200
```

Figure IV.13: Sign up

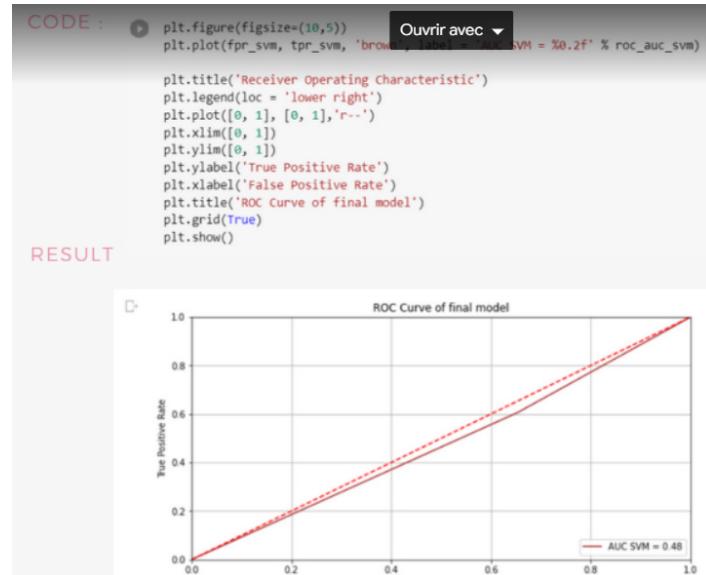


Figure IV.14: Users List



Figure IV.15: Stats

```

CODE: [ ] #Create KNN Classifier
      knn = KNeighborsClassifier(n_neighbors=25)
      #Train the model using the training sets
      knn_model = knn.fit(X_train, y_train)
      #Predict the response for test dataset
      y_pred_knn = knn_model.predict(X_test)

[ ] knn.score(X_test, y_test)

RESULT: 0.44

CODE: [ ] print('Accuracy of K-NN classifier on test set: {:.2f}'
      .format(knn.score(X_test, y_test)))

RESULT: Accuracy of K-NN classifier on test set: 0.44

CODE: [ ] #".format(345.7916732)
      print('Accuracy of K-NN classifier on training set: {:.2f}'
      .format(knn.score(X_train, y_train)))

RESULT: Accuracy of K-NN classifier on training set: 0.58

```

Figure IV.16: Login

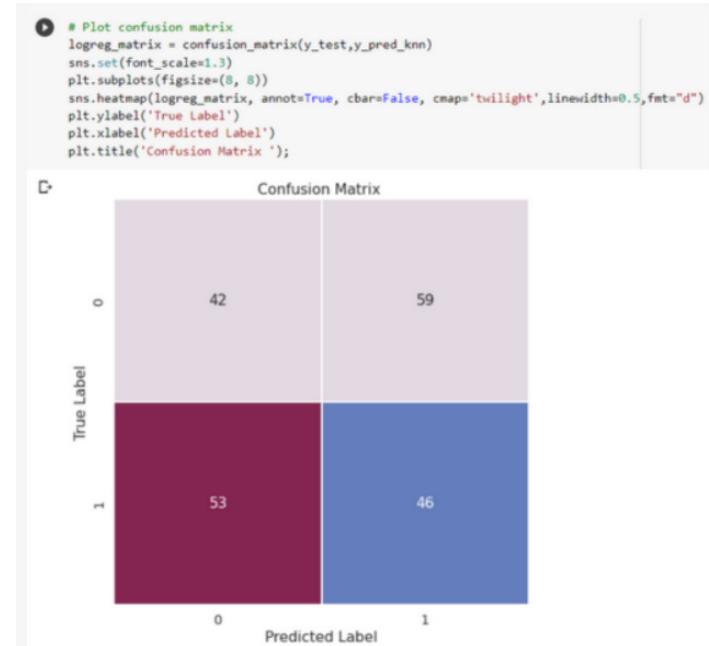


Figure IV.17: Sign up

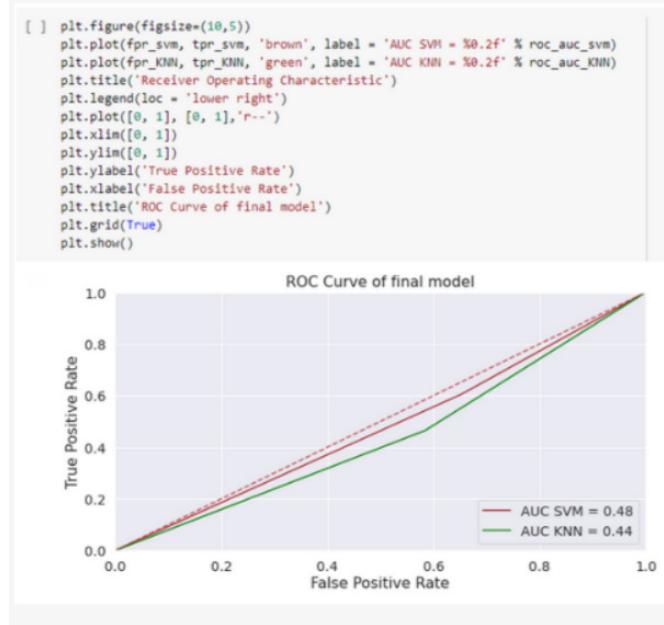


Figure IV.18: Users List

```

684) age< 35. 5 4 0 FALSE (1.00000000 0.00000000) *
685) age>= 35. 5 4 1 TRUE (0.26315789 0.73684211) *
343) violence_frequency< 18 6 0 TRUE (0.00000000 1.00000000) *
344) age>= 59. 5 19 5 TRUE (0.26315789 0.73684211) *
11) happiness< 20 8 0 FALSE (0.50000000 0.50000000) *
22) violence==TRUE 14 4 FALSE (0.50000000 0.50000000) *
44) happiness>=72. 5 4 2 FALSE (0.75000000 0.25000000) *
45) happiness< 72. 5 6 1 TRUE (0.16666667 0.83333333) *
23) violence==FALSE 12 1 TRUE (0.08333333 0.91666667) *
3) niveau_educatif==Excellent_Passable 173 69 TRUE (0.39884393 0.60115607)
6) age< 24 20 2 FALSE (0.60000000 0.40000000) *
12) happiness< 31 5 9 7 TRUE (0.46666667 0.53333333) *
13) happiness<=31 15 7 TRUE (0.46666667 0.53333333) *
26) happiness<=60 4 0 FALSE (1.00000000 0.00000000) *
27) happiness< 60 11 3 TRUE (0.27272727 0.72727273) *
7) age>=24.5 153 57 TRUE (0.37254902 0.62745098)
14) age>= 24.5 146 57 TRUE (0.39041096 0.60958904) *
28) gender==Female 146 57 TRUE (0.39041096 0.60958904) *
26) violence_frequency<=18 5 11 FALSE (0.60000000 0.40000000)
112) happiness>=28 31 11 FALSE (0.64516129 0.35483871) *
113) happiness< 28 4 1 TRUE (0.25000000 0.75000000) *
57) violence_frequency< 18.5 28 9 TRUE (0.32142857 0.67857143)
114) actif==FALSE 14 6 FALSE (0.57142857 0.42857143)
228) age>=38.5 10 2 TRUE (0.00000000 1.00000000) *
229) actif==TRUE 14 1 TRUE (0.00000000 1.00000000) *
115) actif==TRUE 14 1 TRUE (0.07142857 0.92857143) *
29) gender==Agender, Male, Non-binary, Polygender 83 27 TRUE (0.32530120 0.67469880)
58) age< 38.5 20 8 FALSE (0.60000000 0.40000000)

```

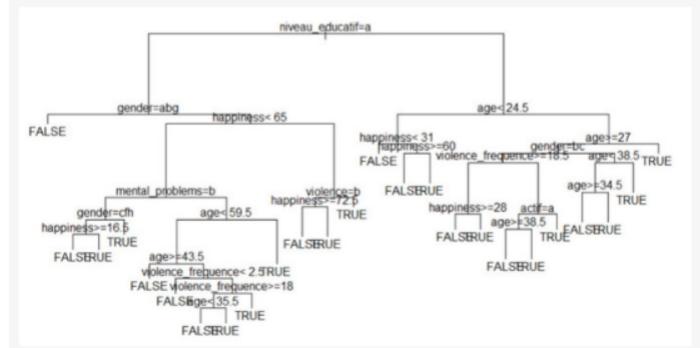


Figure IV.19: Stats

IV.3.3 Angular

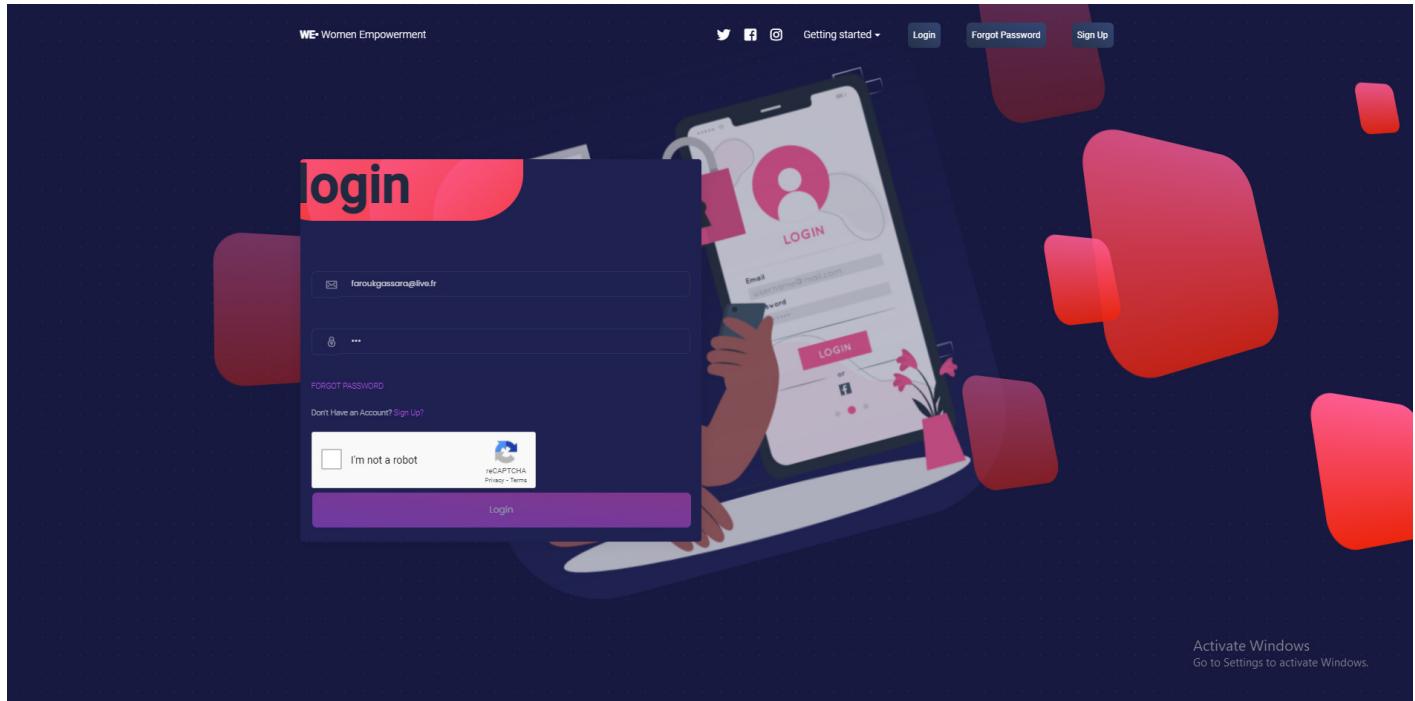


Figure IV.20: Login

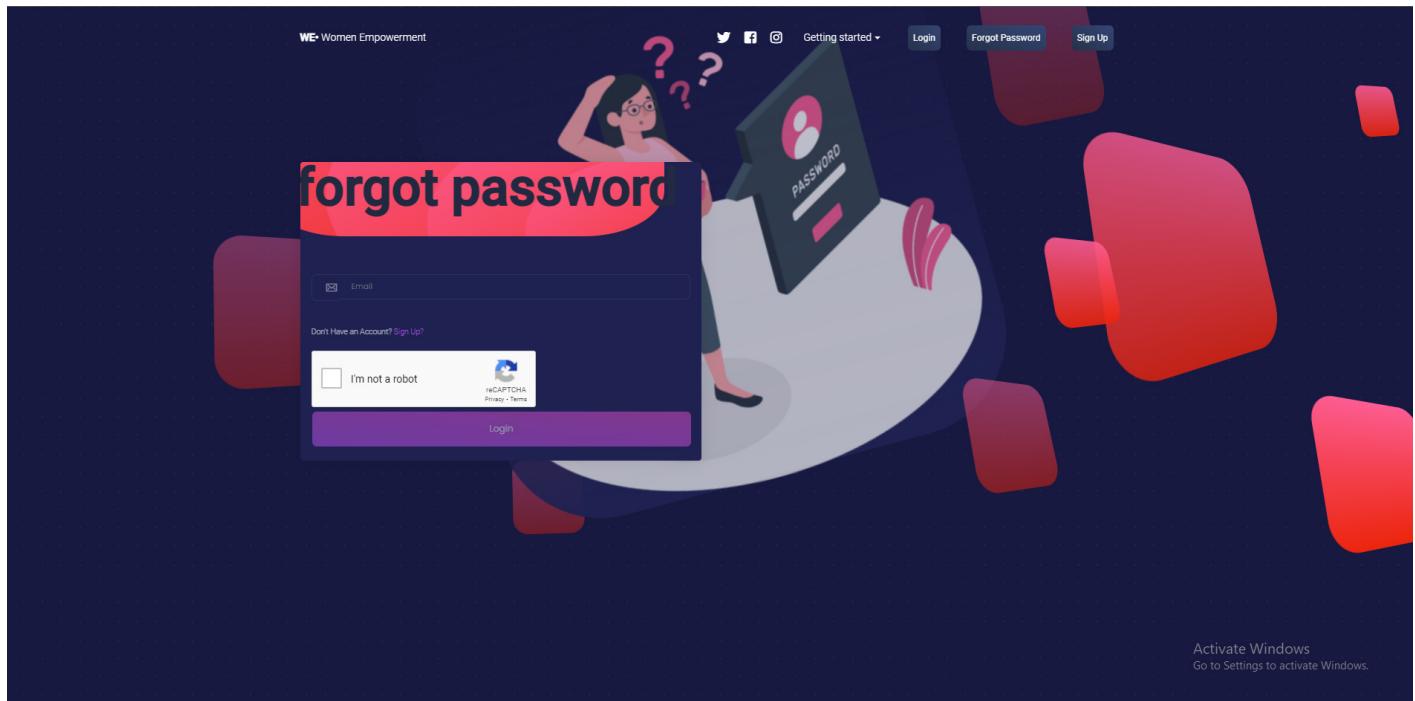


Figure IV.21: Login

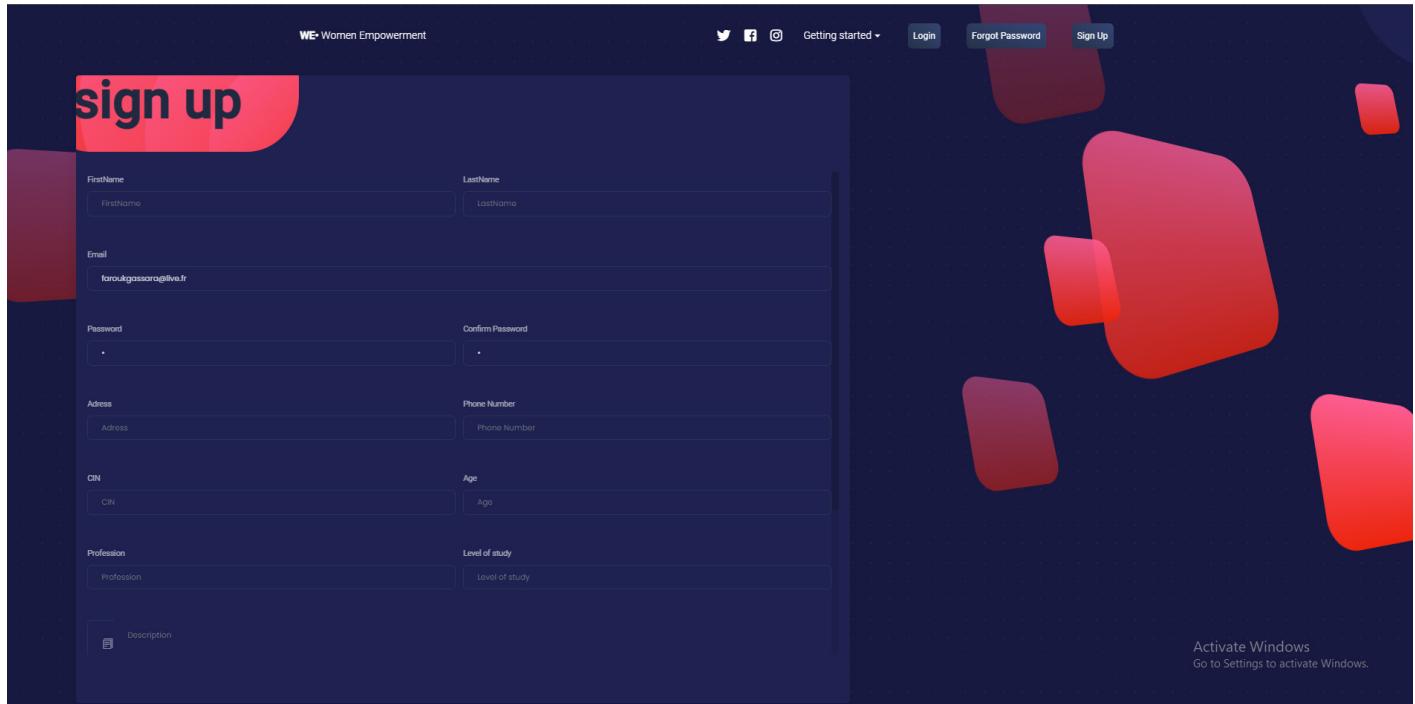


Figure IV.22: Login

The screenshot shows the Cyber Hunter dashboard. On the left, a sidebar lists navigation items: Dashboard, User Management, Events, Cagnotte, Reservation, Reporting, Blacklist, Fakeaccounts, Trainings, and Comments Management. The main area is titled 'DASHBOARD ADMIN' and contains two sections: 'Coronavirus disease tracker:' and 'Coronavirus disease tracker:'. Each section has four colored boxes (red, blue, red, orange) showing statistics: CONFIRMED (288730029, 726706), ACTIVE (0, 0), DEATHS (5436561, 25569), and RECOVERED (0, 0). Below these are four cards: 'Total Users' (364, updated now), 'Inactif Users' (21, updated now), 'Actif Users' (343, updated now), and 'Users Interactions' (49.5%, updated now). At the bottom, there is a 'Management Table' section and a message: 'Activate Windows Go to Settings to activate Windows.' The background features abstract red and orange shapes.

Figure IV.23: Login

The screenshot shows a dark-themed web application interface. On the left, a sidebar lists various management modules: Dashboard, User Management, Events, Cagnotte, Reservation, Reporting, Blacklist, Fakeaccounts, Trainings, and Comments Management. The main content area is titled "Users List". It features a search bar with placeholder "Enter Firstname" and a "Print PDF" button. Below the search is a table with columns: EMAIL, FIRSTNAME, LASTNAME, ENABLED, LOCKED, and ACTIONS. The table contains six rows of user data. At the bottom of the table is a navigation bar with "Previous", a page number "1", and "Next". A watermark at the bottom right reads "Activate Windows Go to Settings to activate Windows."

EMAIL	FIRSTNAME	LASTNAME	ENABLED	LOCKED	ACTIONS
achraf@gmail.com	Achraf	Benzrig	false	false	
loujyan@gmail.com	Loujyan	Hraga	false	false	
faroukgasara@live.fr	Farouk	Gassara	true	false	
marwa@gmail.com	Marwa	Boudellea	false	false	
omridali@gmail.com	Dali	Omri	false	false	
dalicharrad@gmail.com	Dali	Charrad	false	false	

Figure IV.24: Login

This screenshot shows a modal window titled "Edit Profile" overlaid on the "Users List" page. The modal contains fields for Email (disabled), FirstName (Achraf), LastName (Benzrig), CIN (2), Age (5), Address (dd), Profession (sd), Level of study (sd), Phone Number (3), and About Me (sd). At the bottom of the modal are "Save" and "Close" buttons. To the right of the modal is a preview card for a user profile, featuring a circular profile picture of a man with dreadlocks, a purple and pink background, and a quote: "Do not be scared of the truth because we need to restart the human foundation in truth And I love you like Kanye loves Kanye I love Rick Owens bed design but the back is...". The preview card also includes social media sharing icons and a "Print PDF" button. A watermark at the bottom right of the preview card reads "Activate Windows Go to Settings to activate Windows."

Figure IV.25: Login

The screenshot shows a dark-themed web application interface. On the left, a sidebar lists various menu items: Dashboard, User Management, Events, Cagnotte, Reservation, Reporting, Blacklist, Fakeaccounts, Trainings, and Comments Management. The main content area has a title 'Jackpots List' at the top. Below it is a table with columns: VALEUR, DATE DEBUT, DATE FIN, and ACTIONS. Two rows are visible: one with VALEUR 0, DATE DEBUT 2022.04.14, and DATE FIN 2022.04.14; another with VALEUR 0, DATE DEBUT 2020.04.14, and DATE FIN 2022.04.14. At the bottom of the table are navigation links: '< Previous', a central page number '1', and 'Next >'. Below the table is a modal window titled 'Add Jackpot' with two input fields for 'DATE DEBUT' and 'DATE FIN', both labeled 'mm/dd/yyyy'. A pink 'Ajouter' button is at the bottom right of the modal. In the bottom right corner of the main content area, there is a watermark: 'Activate Windows Go to Settings to activate Windows.'

Figure IV.26: Login

The screenshot shows a dark-themed web application interface. On the left, a sidebar lists various menu items: Dashboard, User Management, Events, Cagnotte, Reservation, Reporting, Blacklist, Fakeaccounts, Trainings, and Comments Management. The main content area has a title 'Fake Accounts' at the top. Below it is a chart with a bell-shaped curve, showing data points across categories: ADDRESS, AGE, FULL NAME, REPORTED, PUBLICATIONS, and TOKEN. Below the chart is a table titled 'Fake accounts' with columns: EMAIL, ADDRESS, AGE, FULL NAME, REPORTED, PUBLICATIONS, TOKEN, and PERCENTAGE. Six rows of data are listed:

EMAIL	ADDRESS	AGE	FULL NAME	REPORTED	PUBLICATIONS	TOKEN	PERCENTAGE
faroukgassara@live.fr	Verified(ez)	age does not meet the requirements(5)	Verified(Farouk Gassara)	Verified	Verified	Verified	0.2
achraf@gmail.com	Verified(dd)	age does not meet the requirements(5)	Verified(Achraf Benzrig)	Verified	the Publications number is suspicious	Verified	0.5
loujain@gmail.com	Verified(ze)	age does not meet the requirements(5)	Verified(Loujain Hrage)	Verified	the Publications number is suspicious	Verified	0.8
marwa@gmail.com	Verified(zef)	age does not meet the requirements(1)	Verified(Marwa Boudellea)	Verified	the Publications number is suspicious	Verified	1.1
omridali@gmail.com	Verified(dd)	age does not meet the requirements(1)	Verified(Dali Omri)	Verified	the Publications number is suspicious	Verified	1.4
dalicharrad@gmail.com	Verified(d)	age does not meet the requirements(1)	Verified(Dali Charrad)	Verified	the Publications number is suspicious	Verified	1.7

At the bottom of the table are navigation links: '< Previous', a central page number '1', and 'Next >'. In the bottom right corner of the main content area, there is a watermark: 'Activate Windows Go to Settings to activate Windows.'

Figure IV.27: Login

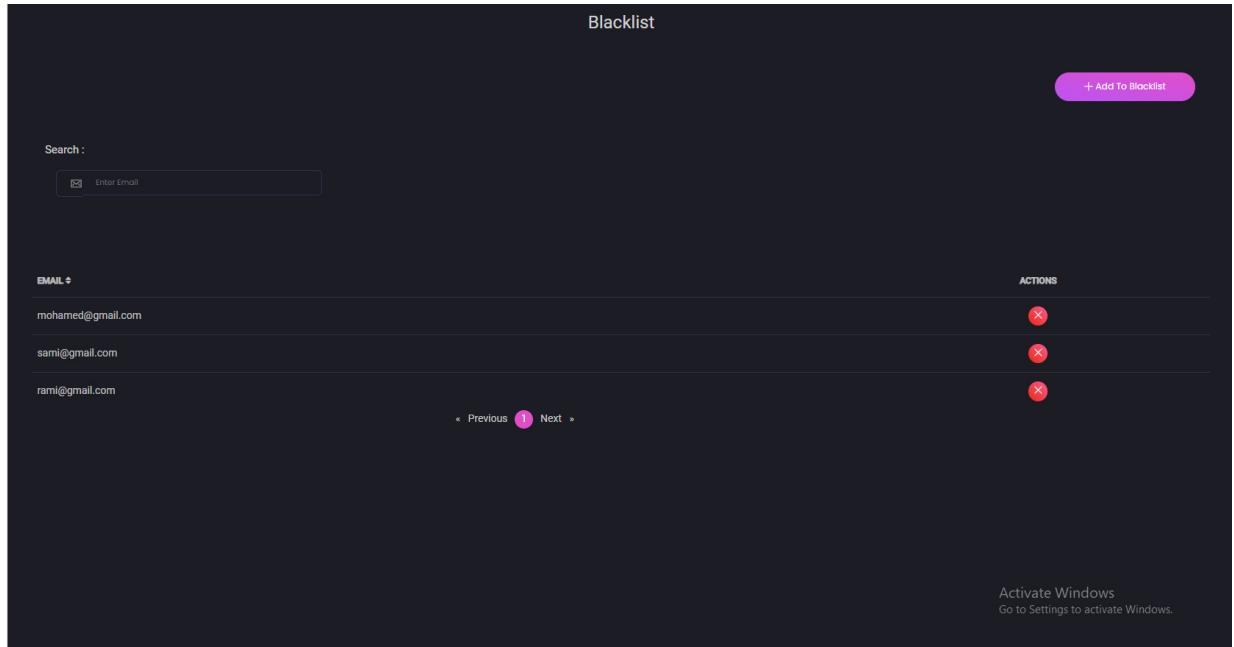


Figure IV.28: Login

Figure IV.29: Login

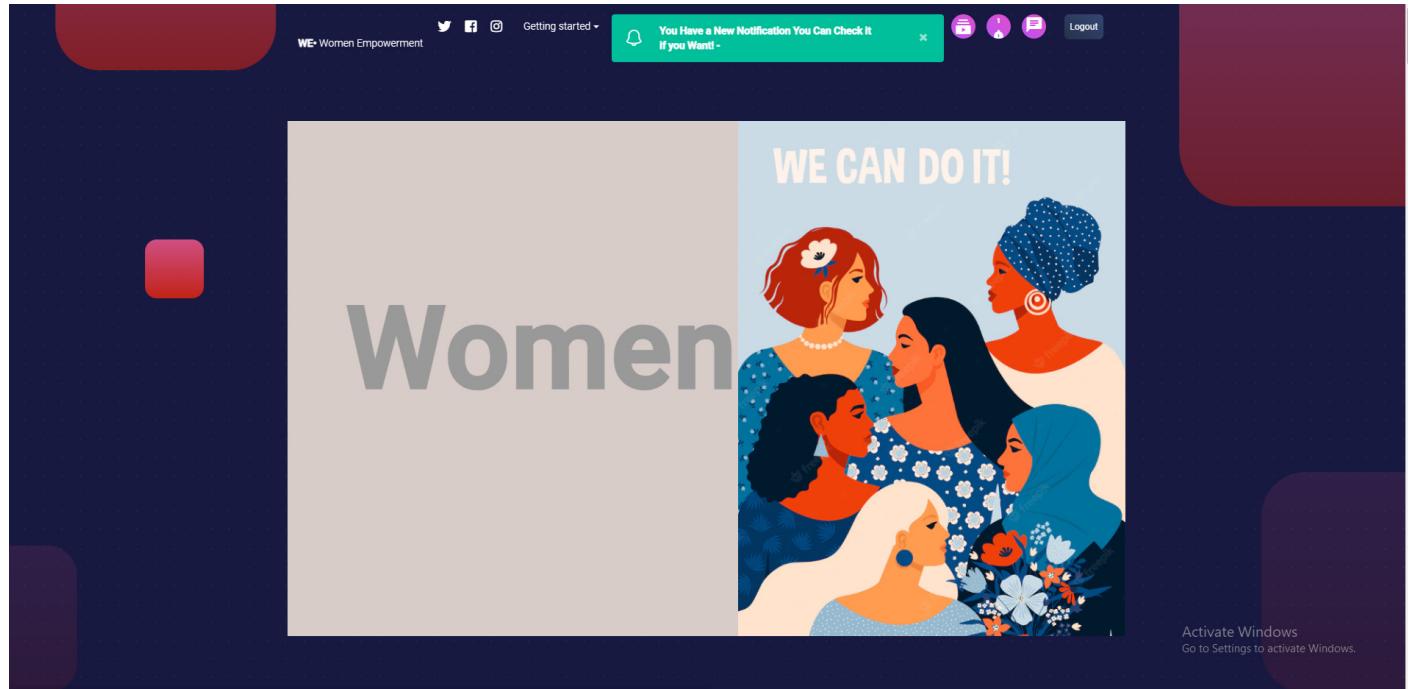


Figure IV.30: Login

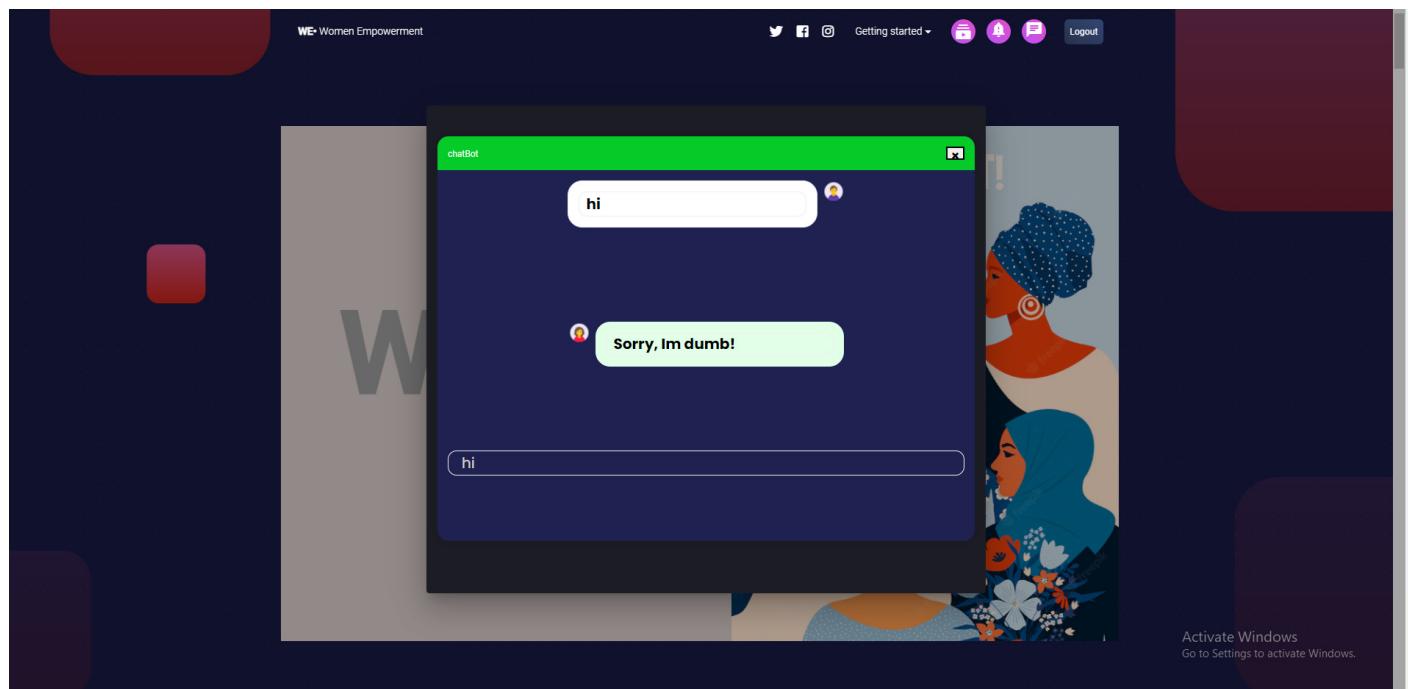


Figure IV.31: Login

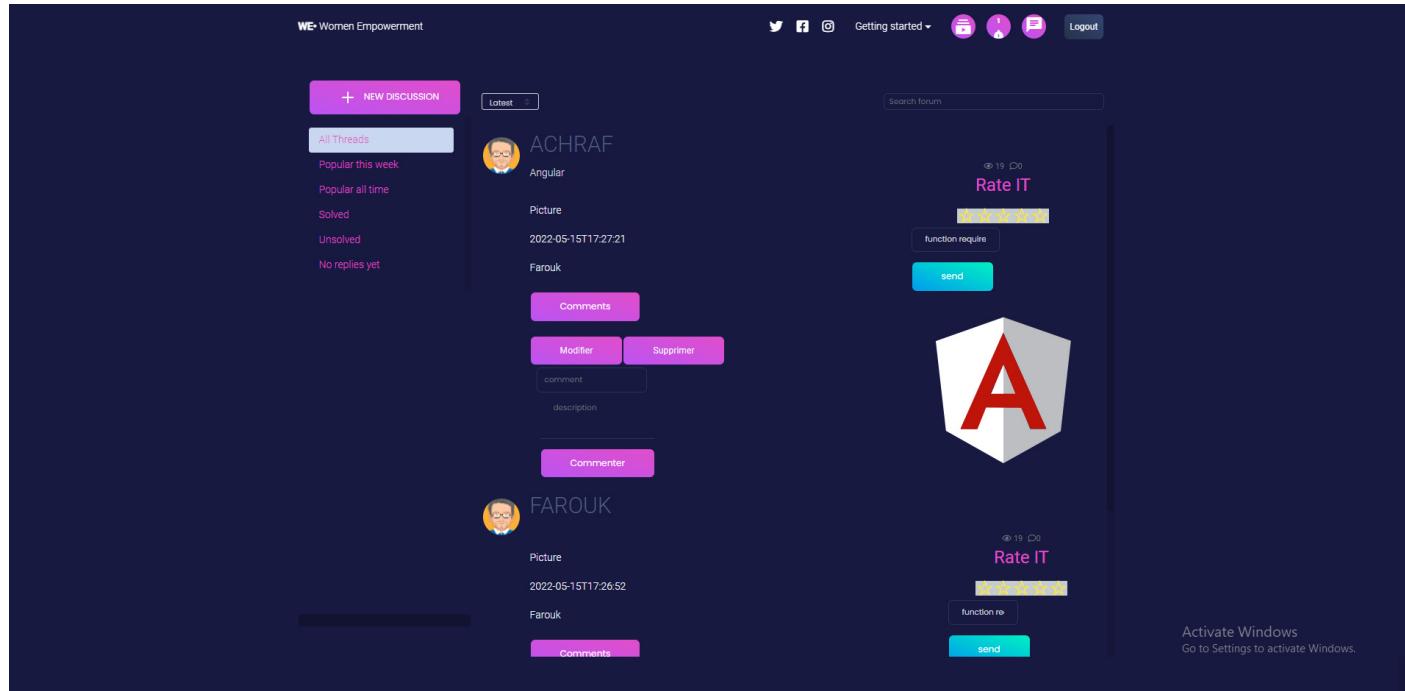


Figure IV.32: Login

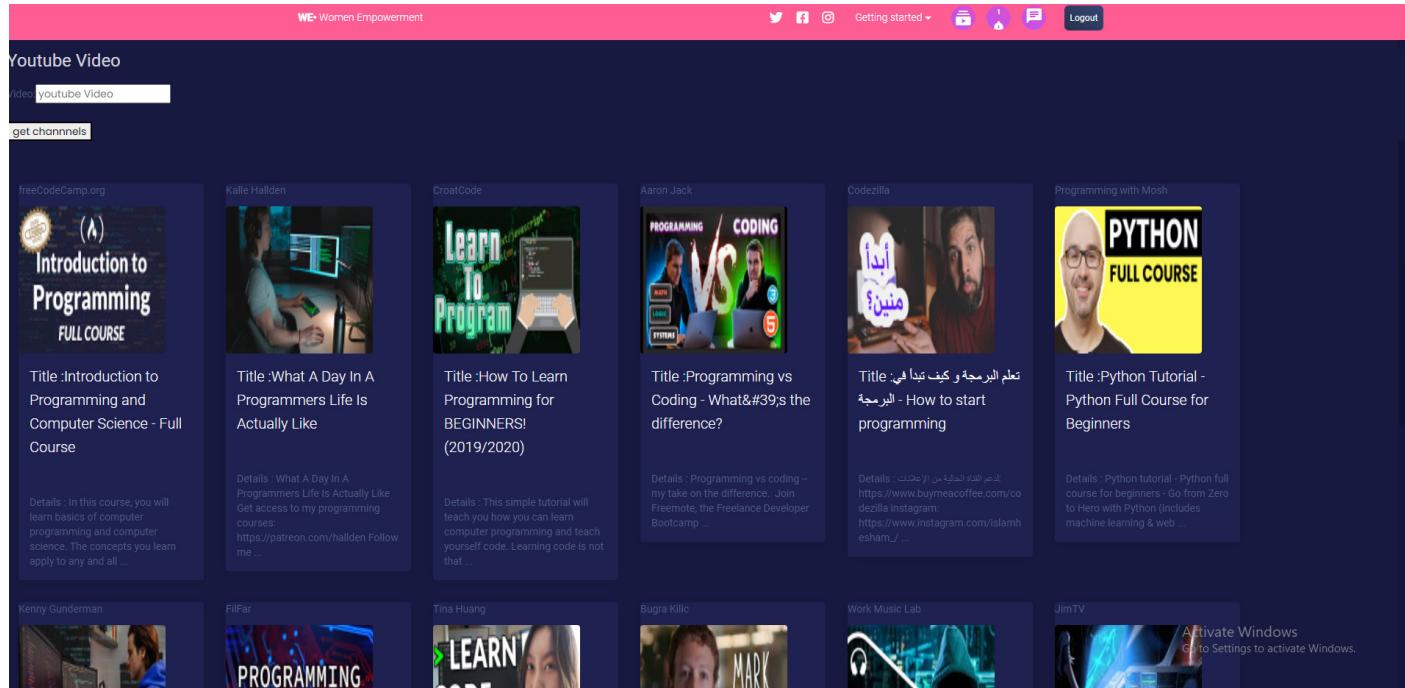


Figure IV.33: Login

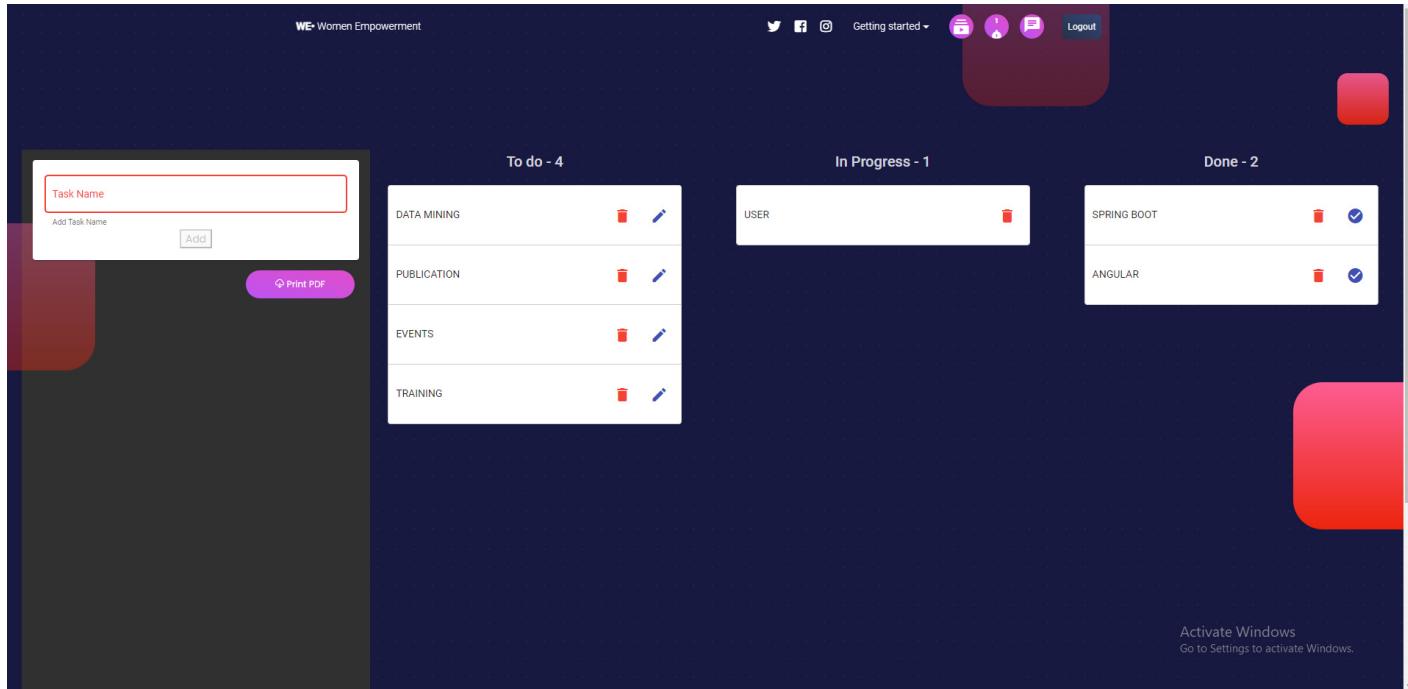


Figure IV.34: Login

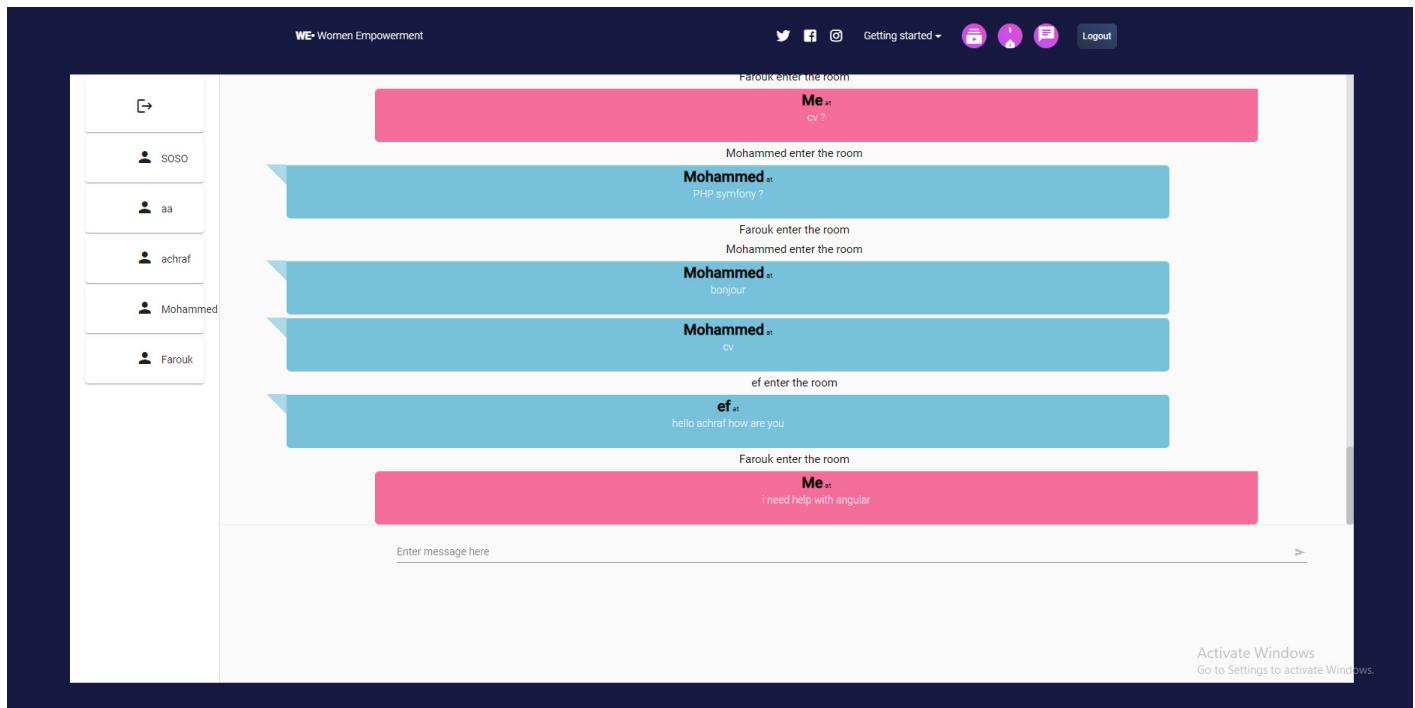


Figure IV.35: Login

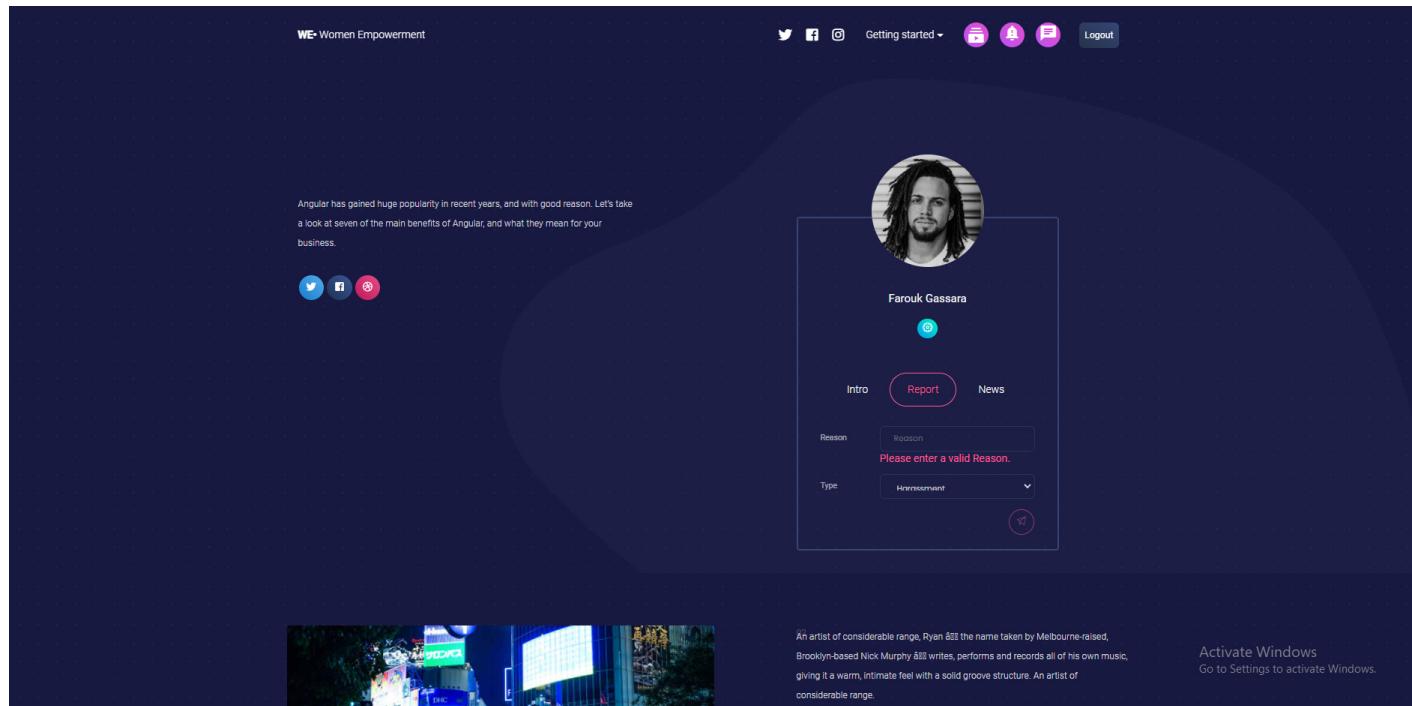


Figure IV.36: Login

IV.4 Working environment

IV.4.1 Materiel Environment

During the project we used PCs with the following characteristics:

- Processeur: Intel(R) Core (TM) i7-7500U CPU up to 3.5GHz
- RAM: 8.00 Go.
- Disque dur: 1 To.
- SSD: 1 To.

- Processeur: Intel(R) Core (TM) i5-7500U CPU up to 3.5GHz
- RAM: 16.00 Go.
- Disque dur: 2 To.
- SSD: 1 To.

- Processeur: Intel(R) Core (TM) i5-7500U CPU up to 3.5GHz
- RAM: 16.00 Go.
- Disque dur: 2 To.
- SSD: 1 To.

- Processeur: Intel(R) Core (TM) i5-7500U CPU up to 3.5GHz
- RAM: 8.00 Go.
- Disque dur: 2 To.
- SSD: 1 To.

- Processeur: Intel(R) Core (TM) i7-7500U CPU up to 3.5GHz
- RAM: 16.00 Go.
- Disque dur: 2 To.
- SSD: 1 To.

- Processeur: Intel(R) Core (TM) i7-7500U CPU up to 3.5GHz
- RAM: 16.00 Go.
- Disque dur: 2 To.
- SSD: 1 To.

IV.4.2 Software Environment



Figure IV.37: STS

- Spring Tool Suite (STS) IDE :Spring Tool Suite is an IDE to develop Spring applications. It is an Eclipse-based development environment. It provides a ready-to-use environment to implement, run, deploy, and debug the application. It validates our application and provides quick fixes for the applications[1].



Figure IV.38: VSC

- Visual Studio Code: Visual Studio Code is an open-source, free, multi-platform code editor (Windows, Mac and Linux), developed by Microsoft. VSC is developed with Electron and uses advanced editing features of the Monaco Editor project. The editor can adapt to other types of languages thanks to a well-provided extension system[2]



POSTMAN

Figure IV.39: Postman

- Postman : Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster[4].



Figure IV.40: GitHub

- GitHub :GitHub is a web-based software development hosting and management service, using Git version management software. GitHub offers paid business accounts, as well as free accounts for free software projects. The site also provides access control and collaboration features such as bug tracking, feature requests, task management and a wiki for each project[5].



Figure IV.41: Adobe XD

- Adobe XD : is a vector-based user experience design tool for web apps and mobile apps, developed and published by Adobe Inc. It is available for macOS and Windows, although there are versions for iOS and Android to help preview the result of work directly on mobile devices[6]



Figure IV.42: Adobe XD

- LaTeX : LaTeX is a language and system for composing documents. This is a collection of macro commands intended to facilitate the use of the TeX “text processor”[7].

IV.5 Conclusion

In this chapter we have presented an overview about the Working environment

Bibliography

- [1] Find Sisterhood [HTTPS://WWW.OURSISTERHOOD.CO.UK/](https://www.oursisterhood.co.uk/).
- [2] Girls Who Code Loop [HTTPS://WWW.OURSISTERHOOD.CO.UK/](https://www.oursisterhood.co.uk/).
- [3] Maven Clinic [HTTPS://WWW.OURSISTERHOOD.CO.UK/](https://www.oursisterhood.co.uk/).
- [4] STS [HTTPS://SPRING.IO/TOOLS](https://spring.io/tools).
- [5] VSC [HTTPS://CODE.VISUALSTUDIO.COM/](https://code.visualstudio.com/)
- [6] GitHub [HTTPS://GITHUB.COM/](https://github.com/)
- [7] Adobe XD [HTTPS://WWW.ADOBE.COM/PRODUCTS/XD.HTML](https://www.adobe.com/products/xd.html)
- [8] Latex [HTTPS://WWW.LATEX-PROJECT.ORG/](https://www.latex-project.org/)