

Dedication

Every great accomplishment has been brought into manifestation through self-efforts as well as guidance and support from others especially from those who mean a lot to us and those who are close to our hearts. We would first thank God Almighty and merciful, who gave us the strength and patience to accomplish this modest work. My mother and my father who filled my journey with prayers and words of encouragement. My brothers, my friends and to everyone who had supported and motivated us.

Acknowledgements

At the end of this work, I would like to thank the whole team of Agile 4UX for their encouragement and the pleasant atmosphere that I found during the whole period of this internship and more particularly my supervisor Mr. Khouldoun BOURAOUI for his help and encouragement.

I would like also to express my sincere appreciation to the entire teaching team of the Institute ISET (Institut supérieur des études technologiques).

I would like to extend my sincere thanks and deep respect and high appreciation to my academic supervisor Mrs. Benene FRADI for her support, help and patience throughout the period of the project.

Finally, I want to say a heartfelt thank you to all those who participated in the realization of this work whether directly or indirectly.

TABLE OF CONTENTS

GENERAL INTRODUCTION	1
CHAPTER 1 «GENERAL CONTEXT»	2
INTRODUCTION	3
1. PRESENTATION OF THE HOSTING ORGANIZATION	3
1.1. <i>The services of the company Agile 4UX.....</i>	3
2. PROJECT PRESENTATION	4
2.1. <i>Existing presentation.....</i>	4
2.2. <i>Problematic.....</i>	4
2.3. <i>Proposed solution.....</i>	4
3. WORKING METHODOLOGY	5
3.1. <i>Choice of methodology</i>	5
3.2. <i>Presentation of the SCRUM method</i>	6
3.3. <i>Organizations</i>	7
CONCLUSION.....	8
CHAPTER 2 « LAUNCH OF THE PROJECT »	9
INTRODUCTION	10
1. REQUIREMENT'S SPECIFICATIONS	10
1.1. <i>Identifications of actors.....</i>	10
1.2. <i>Functional requirements</i>	10
1.3. <i>Non-Functional requirements</i>	11
2. PROJECT MANAGEMENT WITH SCRUM	12
3. PRODUCT BACKLOG.....	13
3.1. <i>PB Prioritization Technique</i>	13
3.2. <i>Prioritized Product Backlog</i>	14
3.3. <i>General use case diagram.....</i>	15
4. PRACTICE OF SCRUM PRINCIPLES.....	16
4.1. <i>Sprint structuring</i>	16

4.2.	<i>Sprint planning example with TRELLO</i>	16
5.	TECHNOLOGICAL CHOICE	17
5.1.	<i>Physical architecture</i>	17
5.2.	<i>Logic architecture</i>	18
5.3.	<i>Logical architecture front-end side</i>	18
5.4.	<i>Logical architecture back-end side</i>	19
6.	DEVELOPMENT AND TOOLS ENVIRONMENT	19
6.1.	<i>Hardware environment</i>	19
6.2.	<i>Technical environment</i>	20
	CONCLUSION.....	23
	CHAPTER 3 « STUDY AND REALIZATION OF SPRINT 1».....	24
	INTRODUCTION	25
1.	SPRINT BACKLOG	25
2.	CONCEPTION	25
2.1.	<i>Use case diagram</i>	25
2.2.	<i>textual description of use cases</i>	26
2.3.	<i>Sequence diagram</i>	29
2.4.	<i>Class diagram</i>	34
3.	REALIZATION	34
3.1.	<i>Authentication interface</i>	34
3.2.	<i>Admin Table interface</i>	35
3.3.	<i>Update Admin interface</i>	36
3.4.	<i>Adding user interface</i>	37
3.5.	<i>OTHER INTERFACES</i>	37
•	Authorization	37
•	Delete Admin	38
4.	TESTS.....	38
4.1.	<i>Authentication tests</i>	38
4.2.	<i>Authorization test</i>	40

4.3.	<i>Adding user tests</i>	40
4.4.	<i>updating user tests</i>	42
4.5.	<i>Deleting user tests</i>	44
	CONCLUSION.....	45
CHAPTER 4 « STUDY AND REALIZATION OF SPRINT 2».....		46
	INTRODUCTION	47
1.	SPRINT BACKLOG	47
2.	CONCEPTION	48
2.1.	<i>Use case diagram</i>	48
2.2.	<i>textual description of use cases</i>	49
2.3.	<i>Sequence diagram</i>	53
2.4.	<i>Class diagram</i>	58
3.	REALIZATION	59
3.1.	<i>Add, delete training interface</i>	59
3.2.	<i>View, edit Training interface</i>	60
3.3.	<i>Update Module interface</i>	61
3.4.	<i>Add, delete certification interface</i>	62
3.5.	<i>View, edit certification interface</i>	63
3.6.	<i>Plan, delete session interface</i>	64
3.7.	<i>View session interface</i>	65
3.8.	<i>Plan, delete journey interface</i>	66
3.9.	<i>View journey interface</i>	67
3.10.	<i>Add, delete free workshop interface</i>	68
3.11.	<i>View free workshop interface</i>	69
3.12.	<i>Calendar view interface</i>	70
4.	TESTS.....	71
4.1.	<i>Adding Training test</i>	72
4.2.	<i>Planning session test</i>	72
4.3.	<i>Planning session's journey test</i>	73

4.4. <i>Add empty certification's session test</i>	74
CONCLUSION.....	75
CHAPTER 5 « STUDY AND REALIZATION OF SPRINT 3».....	76
INTRODUCTION.....	77
1. SPRINT BACKLOG	77
2. CONCEPTION	77
2.1. <i>Use case diagram:</i>	78
2.2. <i>Use case diagram</i>	78
2.3. <i>sequence diagram</i>	80
2.4. <i>class diagram</i>	84
3. REALIZATION	84
3.1. <i>add, delete feed Post interface</i>	85
3.2. <i>view, edit feed Post interface</i>	85
3.3. <i>view, edit profile interface</i>	87
3.4. <i>Reset password interfaces</i>	88
3.5. <i>history interface</i>	90
3.6. <i>stats interfaces</i>	91
4. TESTS	93
4.1. <i>Reset password tests</i>	93
CONCLUSION	94
BIBLIOGRAPHY	96

LIST OF FIGURES

FIGURE 1 CYCLE OF THE SCRUM METHODOLOGY	7
FIGURE 2 GENERAL USE CASE DIAGRAM	15
FIGURE 3 SPRINT PLANNING	16
FIGURE 4 EXAMPLE OF SPRINT PLANNING	17
FIGURE 5 PHYSICAL ARCHITECTURE	17
FIGURE 6 LOGIC ARCHITECTURE	18
FIGURE 7 MVMM ARCHITECTURE	18
FIGURE 8 CONTROLLER - SERVICE ARCHITECTURE	19
FIGURE 9 NODEJS LOGO	21
FIGURE 10 MONGODB LOGO	21
FIGURE 11 ANGULAR LOGO	21
FIGURE 12 VISUAL STUDIO CODE LOGO	22
FIGURE 13 INTELLIJ IDEA LOGO	22
FIGURE 14 POSTMAN LOGO	22
FIGURE 15 TRELLO LOGO	23
FIGURE 16 SPRINT 1 USE CASE DIAGRAM	26
FIGURE 17 AUTHENTICATION SEQUENCE DIAGRAM	30
FIGURE 18 AUTHORIZATION SEQUENCE DIAGRAM	31
FIGURE 19 ADDING ADMIN SEQUENCE DIAGRAM	32
FIGURE 20 UPDATING ADMIN SEQUENCE DIAGRAM	33
FIGURE 21 SPRINT 1 CLASS DIAGRAM	34
FIGURE 39 ADDING TRAINING SEQUENCE DIAGRAM	54
FIGURE 40 PLAN SESSION SEQUENCE DIAGRAM	55
FIGURE 41 ADD CERTIFICATION SEQUENCE DIAGRAM	56
FIGURE 42 PLAN JOURNEY SEQUENCE DIAGRAM	57
FIGURE 43 PLAN FREE WORKSHOP SEQUENCE DIAGRAM	58

FIGURE 44 SPRINT 2 CLASS DIAGRAM	59
TABLE 16 SPRINT 3 BACKLOG	77
TABLE 17 SPRINT 3 USE CASE DIAGRAM	78
TABLE 18 DESCRIPTION TEXT ADD FEED POST	79
TABLE 19 DESCRIPTION TEXT UPDATE FEED POST	79
TABLE 20 DESCRIPTION TEXT EDIT PROFILE	80
TABLE 21 DESCRIPTION TEXT RESET PASSWORD	80
FIGURE 63 ADDING FEED POST SEQUENCE DIAGRAM	81
FIGURE 64 EDIT PROFILE SEQUENCE DIAGRAM	82
FIGURE 65 RESET PASSWORD SEQUENCE DIAGRAM	84
FIGURE 66 SPRINT 3 CLASS DIAGRAM	84
FIGURE 67 ADD, DELETE POST FEED INTERFACE	85
FIGURE 68 EDIT, VIEW FEED POST INTERFACE (1).....	86
FIGURE 69 EDIT, VIEW FEED POST INTERFACE (2).....	87
FIGURE 70 VIEW, EDIT PROFILE INTERFACE.....	88
FIGURE 71 RESET PASSWORD INTERFACE (1)	88
FIGURE 72 RESET PASSWORD INTERFACE (2)	89
FIGURE 73 RESET PASSWORD INTERFACE (3)	89
FIGURE 74 RESET PASSWORD INTERFACES (4).....	90
FIGURE 75 HISTORY INTERFACE	90
FIGURE 76 STATS INTERFACE (1)	91
FIGURE 77 STATS INTERFACE (2)	91
FIGURE 78 STATS INTERFACE (3)	92
FIGURE 79 STATS INTERFACE (4)	92
FIGURE 80 STATS INTERFACE (5)	93
FIGURE 81 STATS INTERFACE (6)	93
FIGURE 82 RESET PASSWORD TEST (1)	94
FIGURE 83 RESET PASSWORD TEST (2)	94

LIST OF TABLES

TABLE 1 COMPARISON BETWEEN AGILE AND CLASSIC METHODS.....	6
TABLE 2 PROJECT MANAGEMENT WITH SCRUM.....	13
TABLE 3 PRODUCT BACKLOG	15
TABLE 4 SPRINT 1 BACKLOG	25
TABLE 5 DESCRIPTION TEXT AUTHENTICATION	27
TABLE 6 DESCRIPTION TEXT AUTHORIZATION	27
TABLE 7 DESCRIPTION TEXT ADD ADMIN.....	28
TABLE 8 DESCRIPTION TEXT UPDATE ADMIN.....	29
TABLE 9 SPRINT 1 BACKLOG	48
TABLE 10 DESCRIPTION TEXT ADD TRAINING	49
TABLE 11 DESCRIPTION TEXT ADD CERTIFICATION	50
TABLE 12 DESCRIPTION TEXT MANAGE CERTIFICATION TRAININGS.....	51
TABLE 13 DESCRIPTION TEXT PLAN CERTIFICATION SESSION	51
TABLE 14 DESCRIPTION TEXT PLAN SESSION JOURNEY	52
TABLE 15 DESCRIPTION TEXT PLAN FREE WORKSHOP	53
TABLE 16 SPRINT 3 BACKLOG	77
TABLE 17 SPRINT 3 USE CASE DIAGRAM	78
TABLE 18 DESCRIPTION TEXT ADD FEED Post	79
TABLE 19 DESCRIPTION TEXT UPDATE FEED Post	79
TABLE 20 DESCRIPTION TEXT EDIT PROFILE	80
TABLE 21 DESCRIPTION TEXT RESET PASSWORD	80

GENERAL INTRODUCTION

There is no longer any doubt that computing is the most influential and recent innovation that has marked the life of modern and revolutionary human beings.

Also, one of the most important factors that make an employee in an organization productive, financially, and successful is the way he communicates with his superiors and colleagues, it leads to a productive, comfortable, and optimal working environment. As companies grow, employees know less about their colleagues at work, as well as less share information between them.

So, improving the quality of services is a challenge that any business in the professional field seeks to achieve. To achieve this, it is essential to use new information and communication technologies to improve on the one hand, the functioning and visibility of companies, and on the other hand, to guarantee customer loyalty.

In this context the company Agile 4UX proposes to offer a web application allowing all the members of the team to share and enrich their technical knowledge and do technology monitoring.

This report contains five chapters defining the outline of our project. In a first chapter we are interested in the context of the project in which we present the general framework of the project, the proposed solution that will be developed throughout our internship.

Then, in a second chapter, we will present the architecture of our project, Development, and tools environment and finally we will analyze the Requirements specification. The third, fourth and fifth chapters form the body of our report. These three chapters will be devoted to the development of the three releases of our system respecting the fundamental principles of Scrum. In each of these chapters we start with the sprint specification, then a presentation of the conceptual phase that will be materialized by the elaboration of the use case diagram, data modeling diagram, sequence diagram as well as that of the activity diagrams. and we end by presenting the implementation phase by describing the interfaces of the application.

CHAPTER 1 «GENERAL CONTEXT»

INTRODUCTION

The internship is an opportunity to apply our acquired theoretical knowledge in a professional context. Such work helps us to build professional experience. It is in this context that the currently graduation project took place at Agile 4UX.

This chapter is composed of 3 parts starting with presenting the hosting organization then a description of the project. Next, we will describe the working methodology adopted and finally the project launch.

1. PRESENTATION OF THE HOSTING ORGANIZATION

Agile 4UX is a young, new generation DSC (Digital Services Company), that offers young graduates wishing to undertake an alternative to immigration, by offering them continuous training and professional experience in a multicultural international environment in their countries. After a cycle of around 2 years, each cohort trains a new generation of entrepreneurs capable of innovating with their own startups. Agile 4UX has the ambition to position itself as a breeding ground for entrepreneurs with high potential who can positively impact the entire local entrepreneurial ecosystem.

The implementation of the concept is relatively simple. Agile 4UX relies on experts to provide services to corporate clients. These same experts devote parts of their effort to improving the skills of the other members of the internal team.

1.1. THE SERVICES OF THE COMPANY AGILE 4UX

Agile 4UX offers 3 types of services that include IT development, professional training, and consulting.

- IT development

Agile 4UX specializes in the development of custom cloud and mobile applications.

- Professional training

Agile 4UX trains startups and companies undergoing transformation in innovation professions, Design Thinking, Lean Startup, Lean Product Management, Lean User, Experience, Lean Marketing, Lean HR Management and Agility

- Consulting

Agile 4UX offers a range of services around innovation Startup Coaching, Consulting in Product Management and UX / UI, Lean Management Consulting, Innovation Strategy Audits, Coaching in Digital Transformation, Coaching in Agile Transformation, Marketing Strategy and Digital Marketing Consulting, and IT Strategy Consulting

2. PROJECT PRESENTATION

2.1. EXISTING PRESENTATION

For managing the office website as adding a certification, free workshop, new training, or a session, they needed to contact the developer of the website to add the new data into the database.

2.2. PROBLEMATIC

After studying the existing, it was found that the approach used for management was not practical and not fluid for employees. In addition, we have identified some problems related to:

- The time taken to join and contact developer.
- Time taken for waiting the needed change to be introduced into database.
- How to communicate data to the developer
- lack of flexibility in data management
- Lack of confidentiality of data.

So that, good and rigorous management of recorded data is important.

2.3. PROPOSED SOLUTION

The company Agile 4UX always strives to modernize the realization of its projects while adopting the latest innovations to satisfy its customer.

In this context, Agile 4UX proposed a solution to the training office that will help their team to automate their work by this application the employees will be able to manage their certifications,

session, workshops, modify news feed and be able to manage their employees' team on this back office.

3. WORKING METHODOLOGY

The completion of the project on time is the main concern of each software development team. One of the most frequently encountered problems when building the software is the wrong specification and abrupt change of needs. This can affect not only the development team by creating a stressful environment, but also the time spent on the project and therefore exceeded delivery times. The use of a methodology is important to avoid any disturbance in term of deadlines since it helps to plan the work in an efficient way.

3.1. CHOICE OF METHODOLOGY

Before carrying out an IT project, it is necessary to choose a work methodology and a monitoring process to end up with reliable software. This methodology presents a process which aims to formalize the preliminary stages of the development of a system to make this development more faithful to the needs of the customer.

To fix a particular method, we carried out a comparative study between the classical methods and the agile method. We present in Table 1 a comparison between the classic approach and the agile approach.

	Classic approach	Agile approach
Life cycle	Sequential phases	Iterative and incremental
Planning	Predictive	Adaptive
Change	Resistance to change. Heavy change management processes accepted.	Supportive welcome for change integrated into the process
Risk management	District and rigorous risk management process.	Risk management integrated into the overall process.

Measuring success	Compliance with initial commitments in terms of costs, budget, and quality level.	Customer satisfaction by delivering desired value.
Team	Team with specialized resources led by a project manager	Empowered team, supported by the project manager
Quality	Quality control at the end of the development cycle.	Permanent quality control at product and process level.

Table 1 Comparison between agile and classic methods

According to the comparison made in the previous subsection, we can notice that if the project area is mastered, a life cycle cascaded enough. In cases where we cannot predict everything from the beginning or if the needs are incomplete as in our case, use iterative and incremental methods such as agile methods.

An Agile method guarantees a better quality of communication with the user, a better visibility of the customer on the progress of the work, a better-quality control by the fact that the tests are carried out continuously, which makes it possible to quickly detect the problems. It also incorporates the concept of teamwork.

Among the Agile methods, we can cite "SCRUM" which will be used in the realization of our project.

3.2. PRESENTATION OF THE SCRUM METHOD

Scrum is an agile project management methodology used for software development projects with the purpose of delivering new software capability every 2-4 weeks.

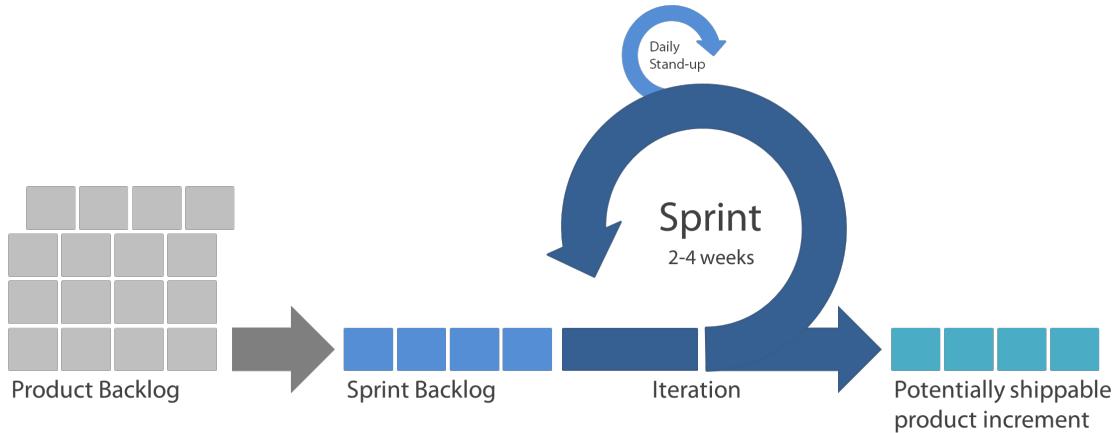


Figure 1 CYCLE OF THE SCRUM METHODOLOGY

3.3. ORGANIZATIONS

The SCRUM methodology is defined by team roles, events, artifacts, and rules below we will detail the role of each:

⇒ SCRUM team:

- The Product Owner: usually an internal or external customer, or a spokesperson for the customer, there is only one Product Owner. He is ultimately accountable or managing the product backlog and accepting completed increments of work.
- The Scrum Master: is the servant leader to the Product Owner, Development Team and Organization. He ensures that the team adheres to Scrum theory, practices, and rules and protects the team by doing anything possible to help the team perform at the highest level.
- The Development Team: The Development Team is a self-organizing, cross- functional group armed with all the skills to deliver shippable increments at the completion of each sprint.

⇒ SCRUM events:

- The Sprint: A sprint is a time-boxed period during which specific work is completed and made ready for review. Sprints are usually 2-4 weeks long but can be as short as one week.
- Sprint Planning: Planning team meetings are time-boxed events that determine which product backlog items will be delivered and how the work will be achieved.
- The Daily Stand-up: The Daily Stand-up is a short communication meeting (no more than 15 minutes) in which each team member quickly and transparently covers progress since

the last stand-up, planned work before the next meeting, and any impediments that may be blocking his or her progress.

- The Sprint Review: The Sprint Review is the "show-and-tell" or demonstration event for the team to present the work completed during the sprint. The Product Owner checks the work against pre-defined acceptance criteria and either accepts or rejects the work. The stakeholders or clients give feedback to ensure that the delivered increment met the business need.
- The Retrospective: The Retrospective, or Retro, is the final team meeting in the Sprint to determine what went well, what didn't go well, and how the team can improve in the next Sprint. Attended by the team and the Scrum Master, the Retrospective is an important opportunity for the team to focus on its overall performance and identify strategies for continuous improvement on its processes.

⇒ SCRUM artifacts:

- Product Backlog: The product backlog is the single most important document that outlines every requirement for a system, project, or product. The product backlog can be thought of as a to-do list consisting of work items, each of which produces a deliverable with business value.
- Sprint Backlog: A sprint backlog is the specific list of items taken from the product backlog which are to be completed in a sprint.
- Increment: An Increment is the sum of all product backlog items that have been completed since the last software release. While it is up to the Product Owner to decide on when an increment is released, it is the team's responsibility to make sure everything that is included in an increment is ready to be released.

⇒ SCRUM rules:

The rules of agile Scrum should be completely up to the team and governed by what works best for their processes. The best agile coaches will tell teams to start with the basic scrum events listed above and then inspect and adapt based on your team's unique needs so there is continuous improvement in the way teams work together.

CONCLUSION

In this chapter, we presented the hosting organization, the general context of the project, the solution that we are going to develop, and we finished with a description of the methodology adopted we quoted the objectives of our project. The next chapter will be the devoted to the preliminary study and the architectural conception of our project.

CHAPTER 2 « LAUNCH OF THE PROJECT »

INTRODUCTION

This chapter is divided into several parts, the first part we will put into practice the actors of the application and present the functional and non-functional requirement of the application. Then we will focus on the methodology selected for our application and present the practice of SCRUM principles. Finally, we will identify the solution architecture.

1. REQUIREMENT'S SPECIFICATIONS

1.1. IDENTIFICATIONS OF ACTORS

In this app we have two actors a super admin and an admin, but every admin has different roles so we can divide the roles as actors, so it's a role-based app.

- Admin manager: this user can manage users (other admins “employees of the office”) and give additional roles to admins.
- Planner: this user can plan and manage certification, training, training modules, free workshop, and plan certification sessions.
- Admin manager: this user can manage posts on the front office website.
- Super admin: this user can do every functionality other admin can do and can check the history of every interaction admins has done.

1.2. FUNCTIONAL REQUIREMENTS

This step is to answer the question of what our system is for. The functional requirements represent the fundamental functionalities that need to be provided by the project. Therefore, the application must offer the following features:

⇒ **Admin manager**

- Authenticate
- Reset password

- Edit profile
- Add admin
- Delete admin
- Edit admin
- Give additional roles to admins
- Check and filter admin table

⇒ **Planner**

- Authenticate
- Reset password
- Edit profile
- Manage certifications
- Manage Trainings and training modules
- Manage and plan Free workshops
- Plan certification sessions
- Plan training journeys

⇒ **Media manager**

- Authenticate
- Reset password
- Edit profile
- Manage feed posts on the office website

⇒ **Super Admin**

- All functionalities that admins got
- Check and filter admin history

1.3. NON-FUNCTIONAL REQUIREMENTS

These are the requirements that characterize the system. They concern the behavioral aspects of the developed application and can be resumed in these main points:

⇒ **The ergonomic constraints**

- Quick access to information
- Simple and understandable interface.

⇒ **The technical constraints**

- Ensure consistency of interfaces.
- The code must be expandable and maintainable to facilitate any improvement or optimization operation.

⇒ **Security**

- The application must guarantee the integrity and confidentiality of the data to the logged-in user.
- The security of the system is ensured by the authentication of members by an encrypted login and password, and this is ensured by the mechanism of the “Json Web Token”.
- Disconnection after idle time (duration, actions).

⇒ **The flexibility**

- Interfaces with tools already used internally.

⇒ **Performance**

- The various offered operations must respond within a reasonable time.

2. PROJECT MANAGEMENT WITH SCRUM

In this subsection, we present all the actors involved in the development of the various phases of the project and the preparation of the internship report. The development team is made up, in our case, of a single person responsible for carrying out the project from design to development. The Product Owner, who is the representative of customers and users, defines needs, priorities

and functionality and directs the activity of the development team. The Scrum Master is the orchestrator; he ensures the smooth running and good atmosphere of the project team.

ROLE	MISSION	ACTOR
Scrum team	<ul style="list-style-type: none"> - Conception - Development - Tests and Validation - Deployment 	JIYED MOHAMED AHMED
Product Owner Scrum Master	<ul style="list-style-type: none"> - Definition of the need and the functionalities to be developed - Approval of the project 	KHOULDOUN BOURAOUI

Table 2 Project management with SCRUM

3. PRODUCT BACKLOG

After identifying the requirements of our application, we detail in this part the product backlog which summarizes all the tasks that need to be performed on the project ranged by priority which leads to a certain order of realization.

3.1. PB PRIORITIZATION TECHNIQUE

the prioritization of the Product Backlog is a primary task, and to do so correctly, it is essential to establish the criteria and the method of prioritization of the products.

There are many prioritization techniques, for our case we have chosen the MoSCoW Method:

- Mo - Must have: must be completed.
- S - Should have: should be completed if possible.
- Co - Could have: could be completed if there is no impact on other ongoing tasks.
- W-Would have: will not be completed instantly but would be preferable for a later version.

3.2. PRIORITIZED PRODUCT BACKLOG

The following table lists the User Stories and their priorities.

- Super Admin
- Admin:
 - Admin manager
 - Planner
 - Media manager

ID	USER STORIES	PRIORITY
1	As a super admin or an admin, I should authenticate to access the app.	Mo
2	As a super admin or an admin, I should reset my password.	Co
3	As a super admin or admin, I should edit my profile image, name, email ...	S
4	As an admin manager, I should add, edit, and delete admins	Mo
5	As an admin manager, I should give additional roles to admins	S
6	As an admin manager, I should check Admin table and filter it by name or role	Mo
7	As a planner, I should add, edit, and delete trainings and training modules.	Mo
8	As a planner, I should add, edit, and delete trainings modules	Mo
9	As a planer I should add, edit, and delete certifications	
10	As a planner, I should plan certification session and manage them	Mo

11	As a planner, I should plan journeys of a session	Mo
12	As a planner, I should plan and manage free workshops	S
13	As a Media manager, I should be able to manage feed post	S
14	As a super admin, I should be able to check history of every admin's interaction done	Co
15	As a super admin or an admin, I should be able to check stats dashboard	Co

Table 3 Product backlog

3.3. GENERAL USE CASE DIAGRAM

In this section we going to transport our product backlog into a use case diagram so we could have a clear view about the system and actors.

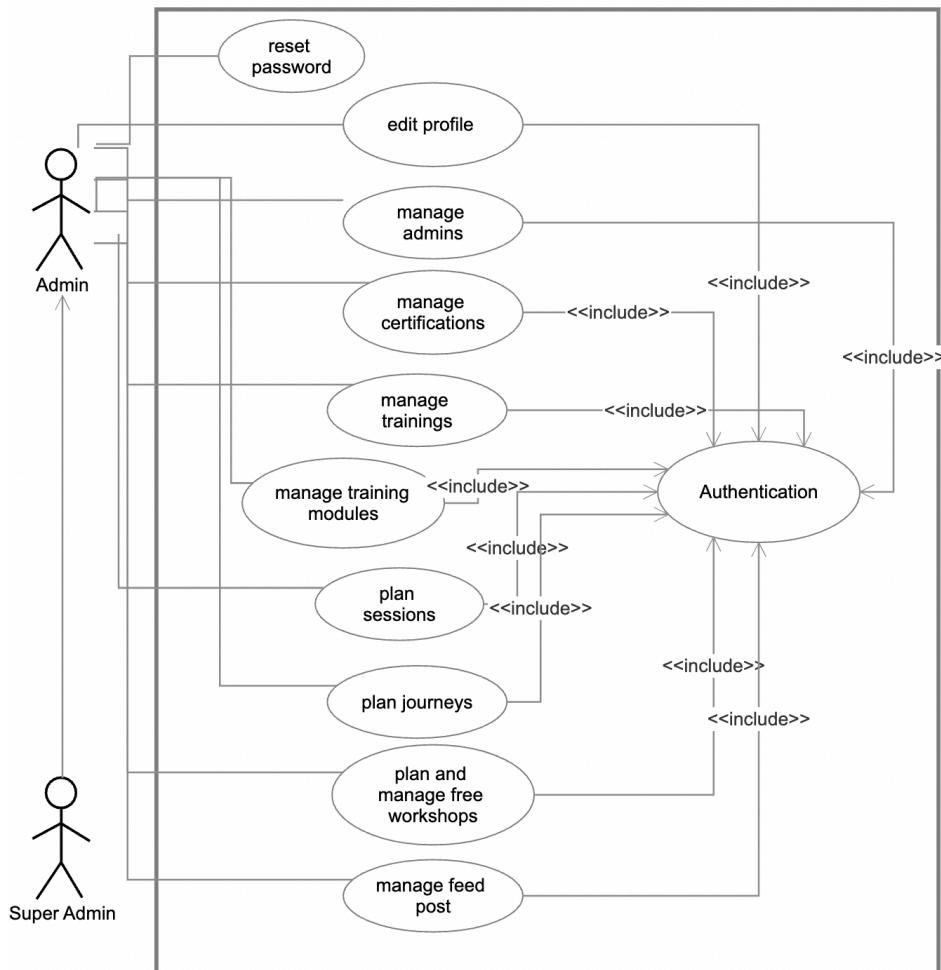


Figure 2 general use case diagram

4. PRACTICE OF SCRUM PRINCIPLES

4.1. SPRINT STRUCTURING

Once we have completed the product backlog, we have established the planning meeting. The purpose of this meeting is to build the sprint backlog based on the product backlog made by the product owner. At the end, we identified the estimated durations of the work to be done during each sprint. For our project we have devised the work on three sprints in one release.

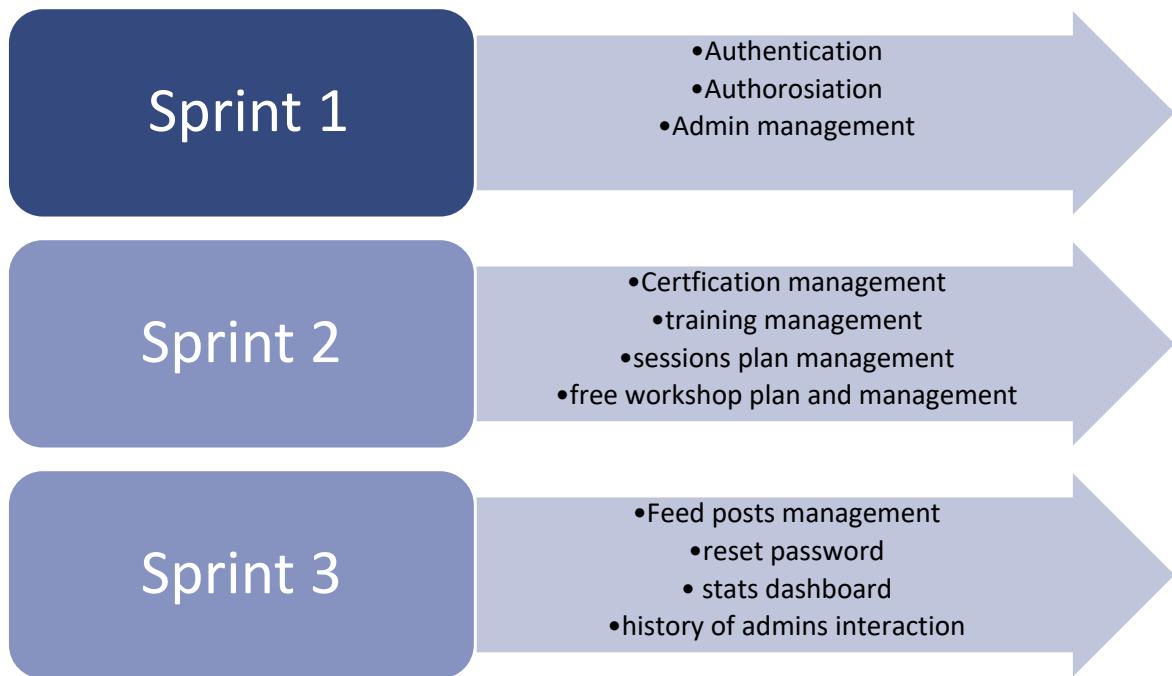


Figure 3 Sprint planning

4.2. SPRINT PLANNING EXAMPLE WITH TRELLO

For planning our project and having the right tools to help us build our project we are using trello application to plan our sprint and in this figure down below we show you how we manage a sprint in trello with our scrum team.

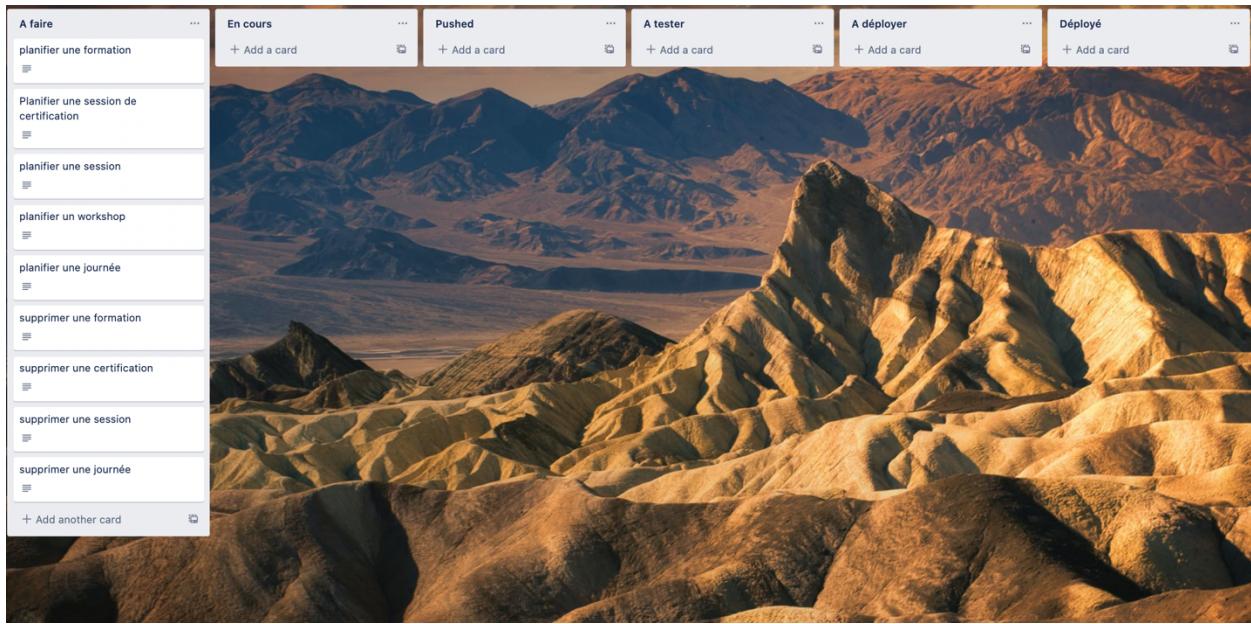


Figure 4 example of sprint planning

5. TECHNOLOGICAL CHOICE

5.1. PHYSICAL ARCHITECTURE

As a physical architecture we have a BACK-OFFICE application created to training office admins, that contains as an architecture, a front-end application server, back-end application server and database server. It is therefore an n-third architecture that is presented as follows:

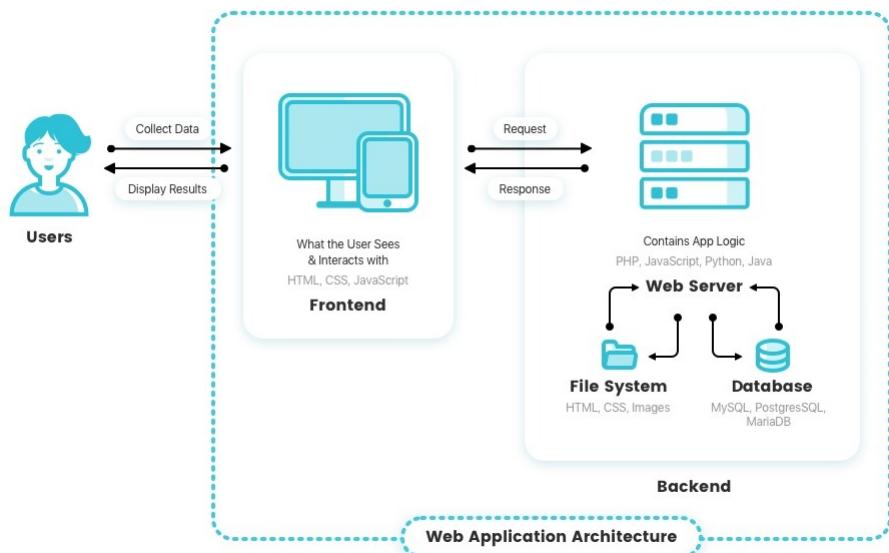


Figure 5 PHYSICAL ARCHITECTURE

5.2. LOGIC ARCHITECTURE

The logical architecture focuses more on the logical decomposition of the application and the grouping of components according to the type of function and the treatments they perform as shown the figure below:

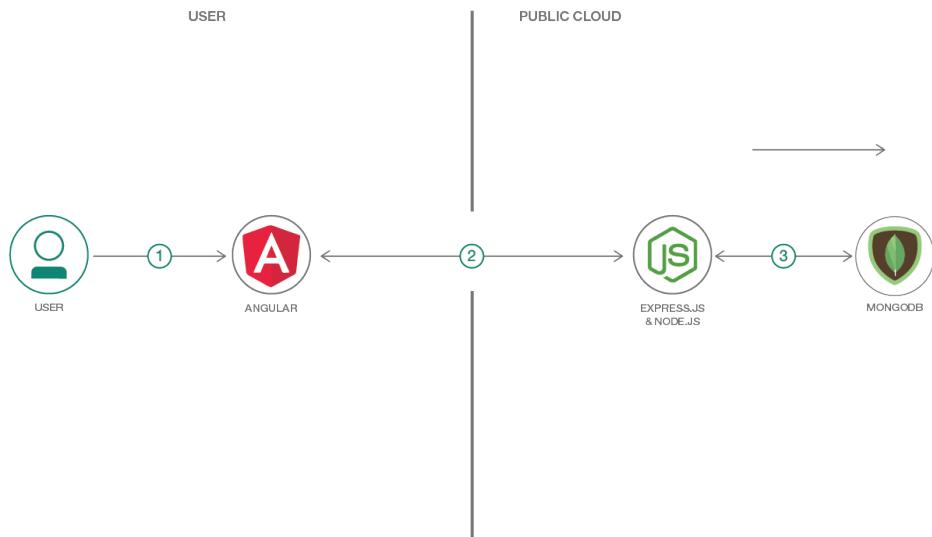


Figure 6 LOGIC ARCHITECTURE

5.3. LOGICAL ARCHITECTURE FRONT-END SIDE

The back-office application is a web application, developed by the angular framework that uses the MVVM architecture.

The Model, View, ViewModel (MVVM pattern) is all about guiding you in how to organize and structure your code to write maintainable, testable, and extensible applications.

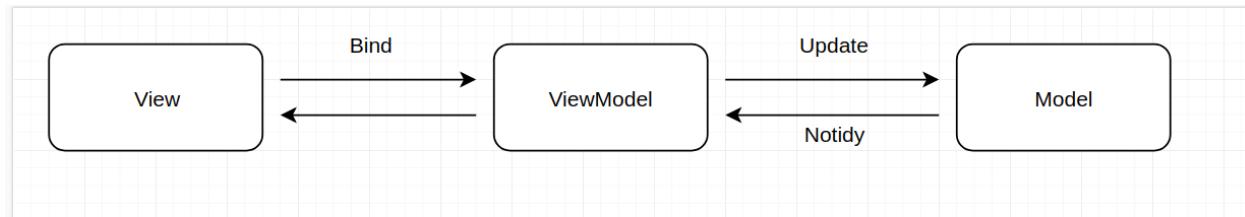


Figure 7 MVVM ARCHITECTURE

Model: It simply holds the data and has nothing to do with any of the business logic.

ViewModel: It acts as the link/connection between the Model and View and makes stuff look

pretty.

View: It simply holds the formatted data and essentially delegates everything to the Model.

5.4. LOGICAL ARCHITECTURE BACK-END SIDE

In our study, we are only interested in NodeJS platforms, uses the Express.js framework Express.js is an excellent framework for creating a REST API, but it doesn't give us any indication on how to organize our project. A correct organization of the project structure will avoid code duplication and improve stability. To make the right choice we will choose the controller - service class architecture as an architecture for our project.

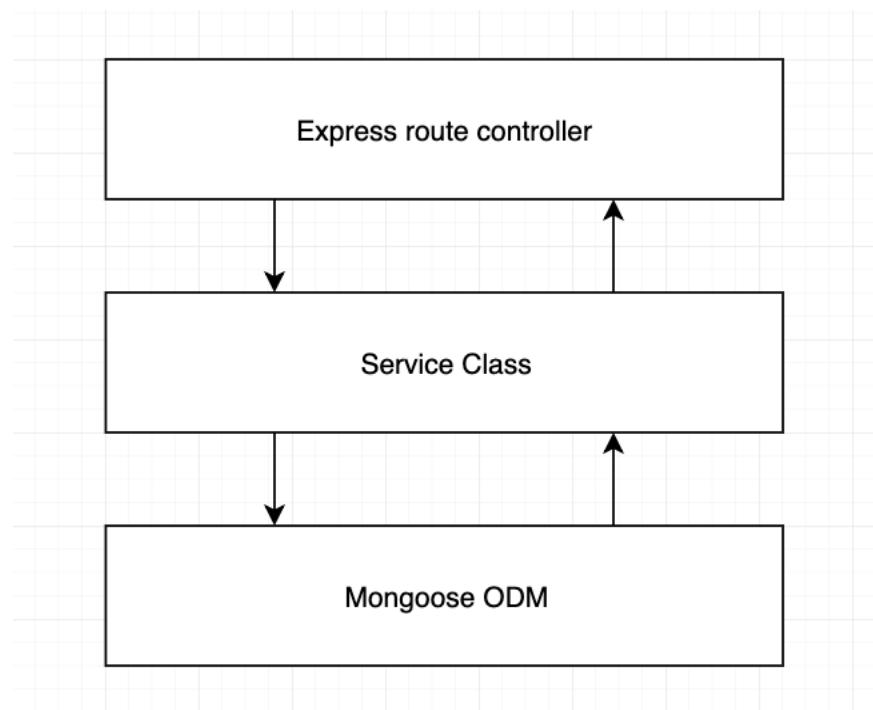


Figure 8 controller - service architecture

6. DEVELOPMENT AND TOOLS ENVIRONMENT

In this section we will present the material and technical environment related to the implementation of the application.

6.1. HARDWARE ENVIRONMENT

During the realization of this work, we used a MacBook pro 2020 with macOS BigSur as operating system and the following features.

- MacBook Pro (13-inch, M1, 2020)
 - Chip: Apple M1
 - Memory: 8GB
 - 13.3-inch (2560 × 1600)
-

6.2. TECHNICAL ENVIRONMENT

The technical choices regarding the development languages are paramount, for this purpose this section will focus on the study of the different language and framework to make a wise choice for the realization of our project.

○ Language of development

The choice of development language will be divided into two parts:

⇒ Back end is invisible to visitors but represents a large part of the development of a web project. The Back end can be divided into two essential parts:

- Server
- Database

⇒ Front end: When we talk about Front end, it is the elements of the website that we see on the screen and with which we can interact.

For the Back end: To make the right choice we study the platform NodeJS and for the database we chose MongoDB for data storage.

- **NodeJS:** is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. [1]



Figure 9 NodeJS Logo

- **MongoDB:** MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. [2]



Figure 10 MongoDB logo

For the Front-End: we choose the framework angular to make our application.

- **Angular:** Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your apps. [3]



Figure 11 Angular Logo

○ Development tools:

In this paragraph, we will talk about the software we used for the conception, development, meeting, testing, and hosting of our web platform.

- **Visual studio code:** is a free source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages. [4]



Figure 12 Visual Studio code logo

- **IntelliJ IDEA:** is an integrated development environment written in Java for developing computer software. It is developed by JetBrains and is available as an Apache 2 Licensed community edition, and in a proprietary commercial edition. Both can be used for commercial development. [5]



Figure 13 IntelliJ IDEA logo

- **Postman:** Postman is an interactive and automatic tool for verifying the APIs of your project. Postman is a Google Chrome app for interacting with HTTP APIs. It presents you with a friendly GUI for constructing requests and reading responses. It works on the backend and makes sure that each API is working as intended. [6]



Figure 14 Postman logo

- **Trello:** is the visual collaboration platform that gives teams perspective on projects. Use Trello to collaborate, communicate and coordinate on all your projects. [7]



Figure 15 Trello logo

CONCLUSION

In this chapter, we have laid out the requirements specifications. This allowed us to have a clearer vision to highlight the different releases of our project as well as the technologies used. The next chapters will be devoted to the presentation of the various sprints.

CHAPTER 3 « STUDY AND REALIZATION OF SPRINT 1»

INTRODUCTION

In this chapter, we present the realization of the first sprint in release 1, by organizing the work on main phases which are the analysis, the realization, and the tests.

1. SPRINT BACKLOG

The sprint is the heart of Scrum. This is a block of time during which an increment of the product will be made. All sprints in a release have a constant duration and never overlap, that is, a sprint cannot start until the previous one is finished. Before embarking on a sprint, the Scrum team must define the goal of the latter, which must be a descriptive table that specifies the workload for each task in number of days.

ID	TASKS	ESTIMATION (days)
1	Authentication	3
2	Authorization	2
3	Adding admin	2
4	Delete admin	1
5	Edit admin	3

Table 4 sprint 1 backlog

2. CONCEPTION

In this section we present the analysis phase that answers the question "what does the system". The answer to this question is reflected in the presentation of the use case diagram and the textual description of each.

2.1. USE CASE DIAGRAM

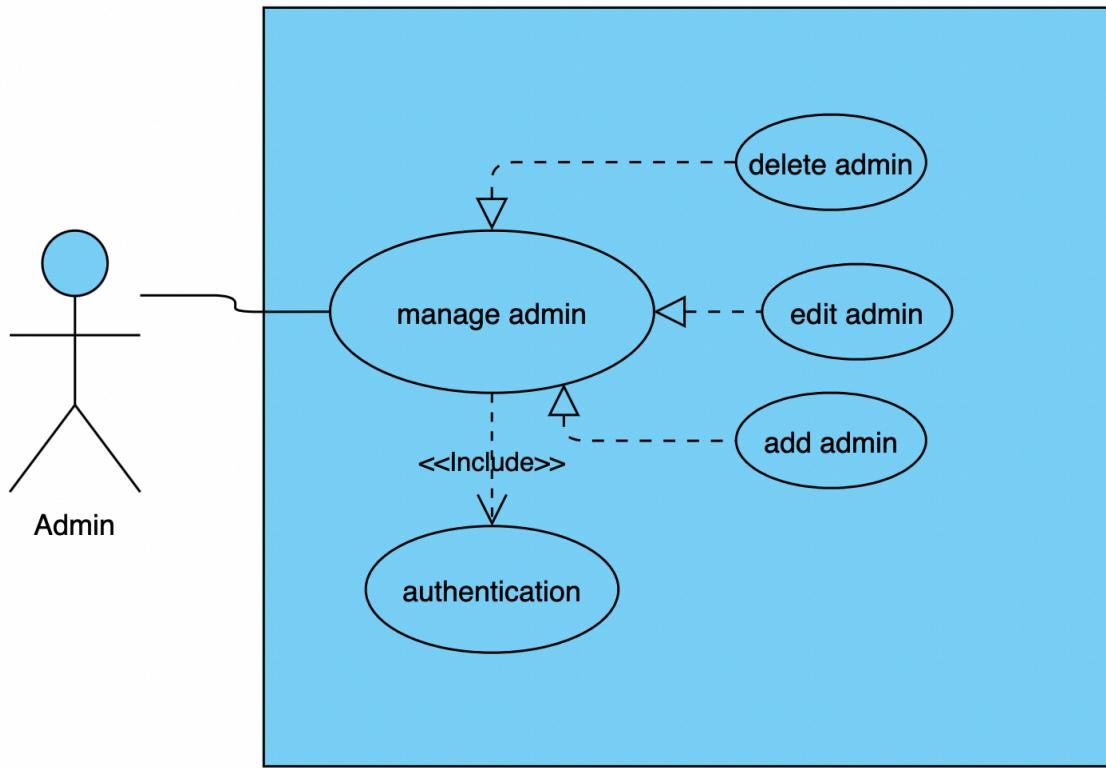


Figure 16 sprint 1 use case diagram

2.2. TEXTUAL DESCRIPTION OF USE CASES

- Text description of use case “Authentication”**

TITLE	Authentication
ACTORS	Super admin and all admins
SUMMARY	This is the first step to do as an admin to access the back office
PRE-CONDITION	----
PRINCIPAL SCENARIO	1- Actor access to the back-office website

	<p>2- Actor enter login credentials</p> <p>3- The system verifies the credentials</p> <p>4- The system save a token with admin information and let the admin access the back office and</p> <p>5- The system displays a success notification to the admin</p> <p>6- The admin now can access the application</p>
POST-CONDITION	Admin access the back-office and can perform actions

Table 5 description text authentication

- **Text description of use case “Authorization”**

TITLE	Authorization
ACTORS	Super admin and all admins
SUMMARY	This step verifies the admin if he got authorization to perform the desired action
PRE-CONDITION	----
PRINCIPAL SCENARIO	<ol style="list-style-type: none"> 1. Actor perform any action 2. Actor token sent to the system to verifies if he got the authorization 3. The system authorizes the action 4. The system displays a success notification to the admin
POST-CONDITION	Admin action desired performed successfully

Table 6 description text authorization

- **Text description of use case “Add admin”**

TITLE	Add admin
ACTORS	Super admin and admin manager
SUMMARY	In this case the super or manager admin can add an admin to the system
PRE-CONDITION	The actor must be authenticated to perform the task and got the authorization to the page
PRINCIPAL SCENARIO	<ol style="list-style-type: none"> 1. Actor access to admin management section and add user page 2. Actor enter admin data 3. The system verifies the data 4. The system sends an email to the admin with his credentials 5. The system displays a success notification to the actor. 6. The admin now can access the application
POST-CONDITION	Admin receive his credentials by mail and can access to the application

Table 7 Description text Add admin

- **Text description of use case “Update admin”**

TITLE	Update admin
ACTORS	Super admin and admin manager
SUMMARY	In this case the super or manager admin can edit an admin and add new roles to him

PRE-CONDITION	The actor must be authenticated to perform the task and have access to the specific page
PRINCIPAL SCENARIO	1- Actor access to admin management section 2- Actor click on edit button for a user in the table 3- Actor enter admin data to be updated 4- The system verifies the data and pass a successfully message 5- The admin now is updated
POST-CONDITION	Admin updated

Table 8 Description text Update admin

2.3. SEQUENCE DIAGRAM

Sequence diagrams are the graphical representation of interactions between actors and the system in chronological order in the UML formulation. We present thereafter the system sequence diagrams of the most important classified use case.

- “authentication” sequence diagram

Authentication

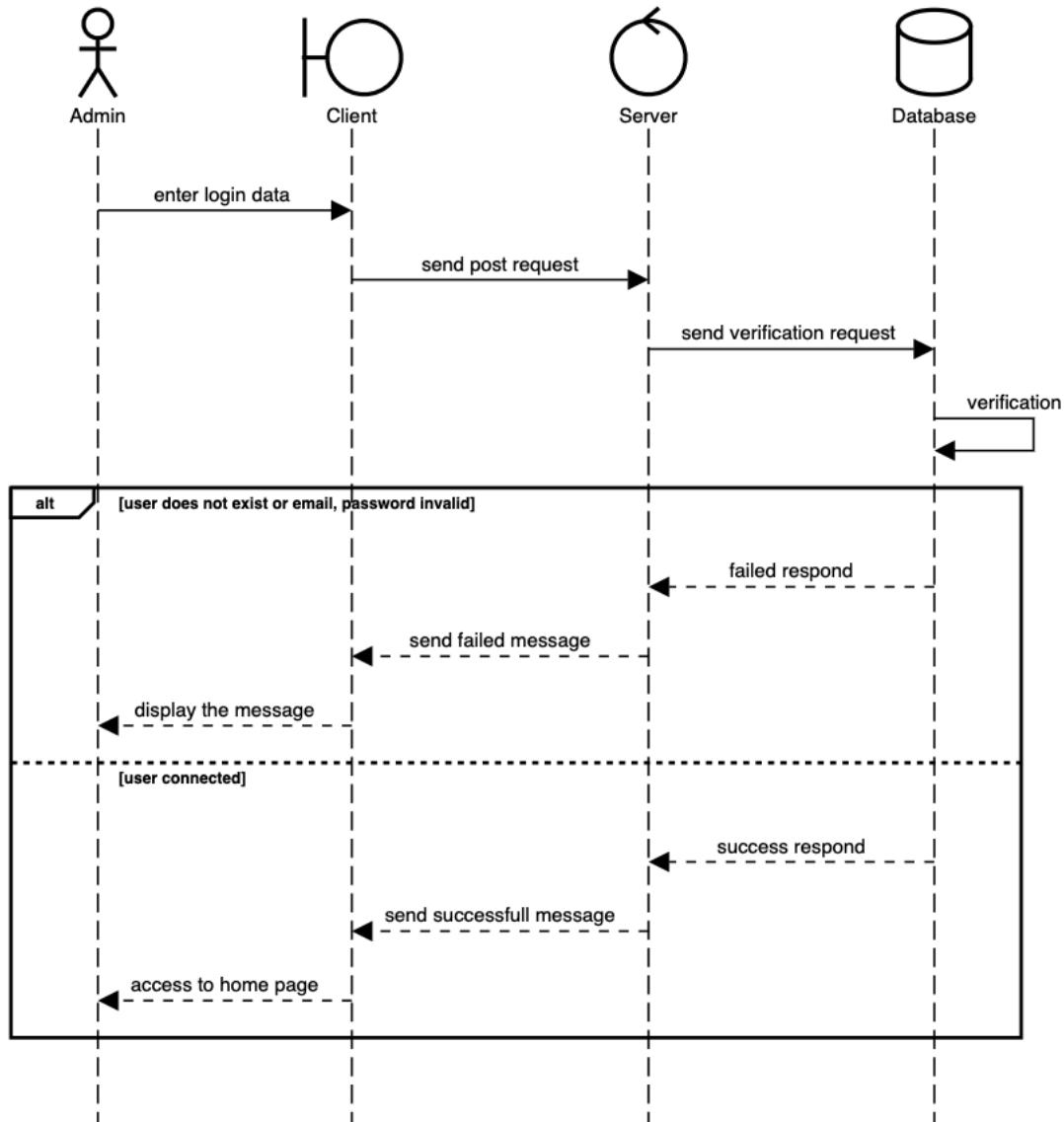


Figure 17 authentication sequence diagram

- “Authorization” sequence diagram

Authorization

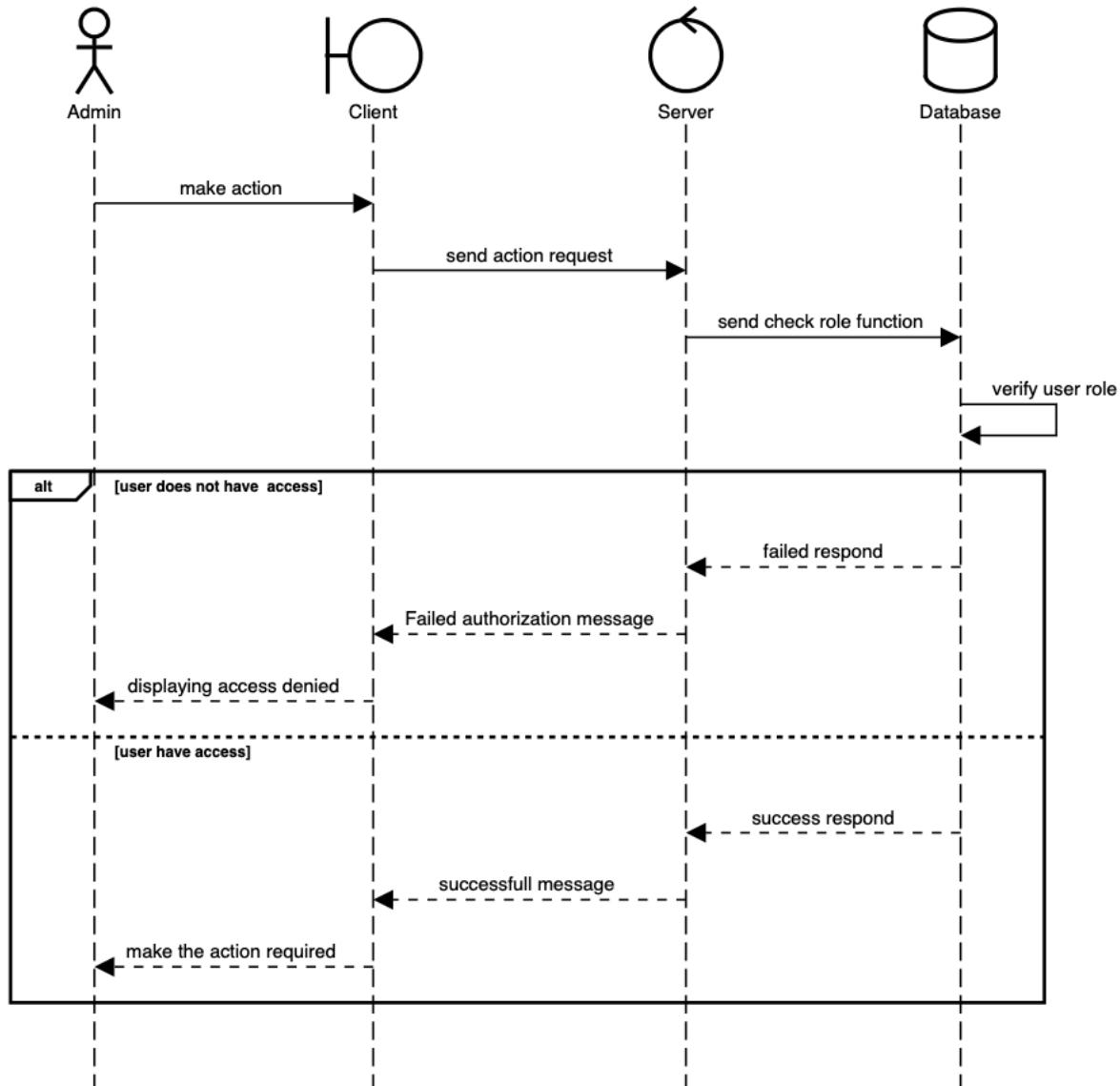


Figure 18 Authorization sequence diagram

- “Adding admin” sequence diagram

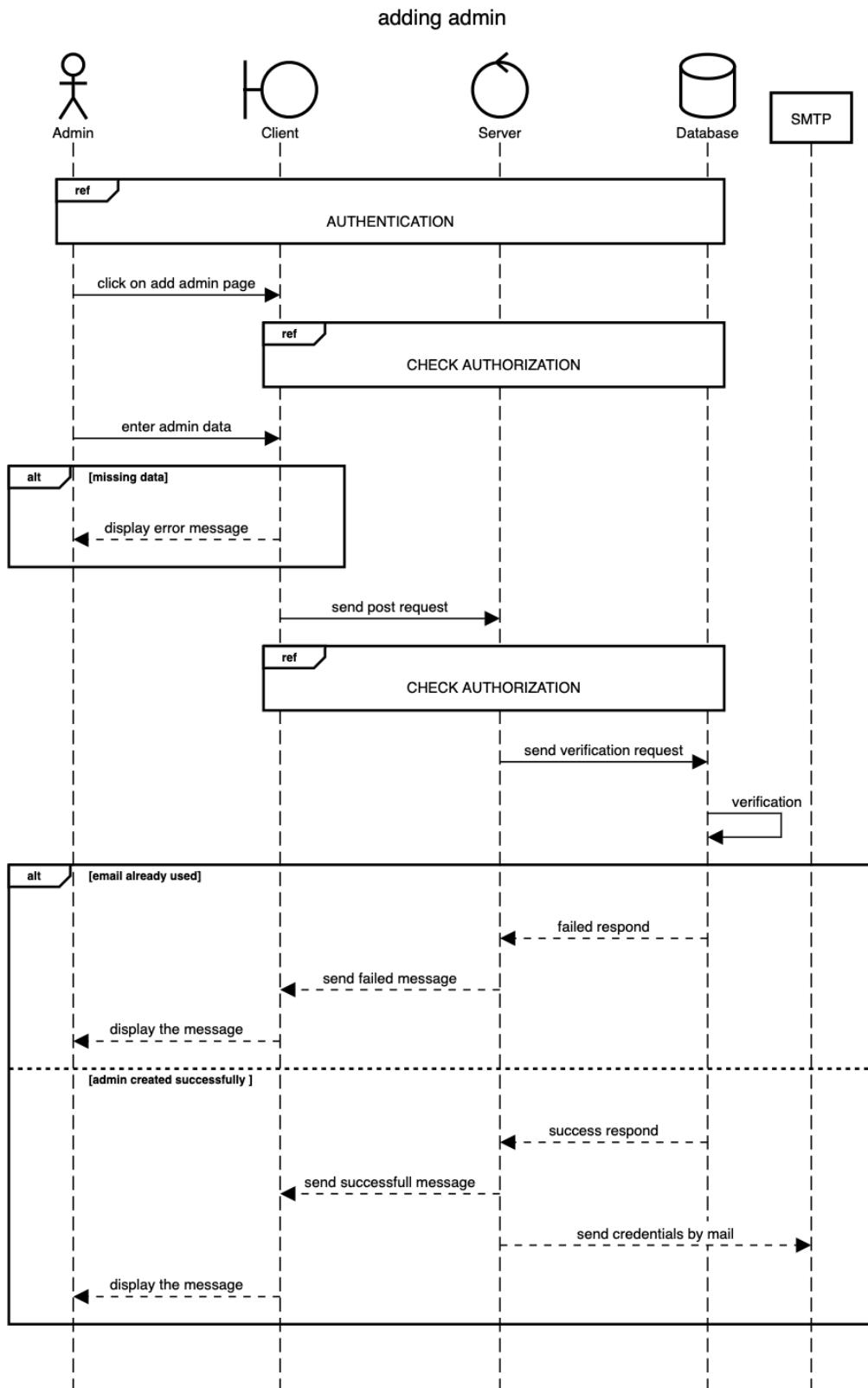


FIGURE 19 Adding Admin Sequence Diagram

- “Updating admin “sequence diagram

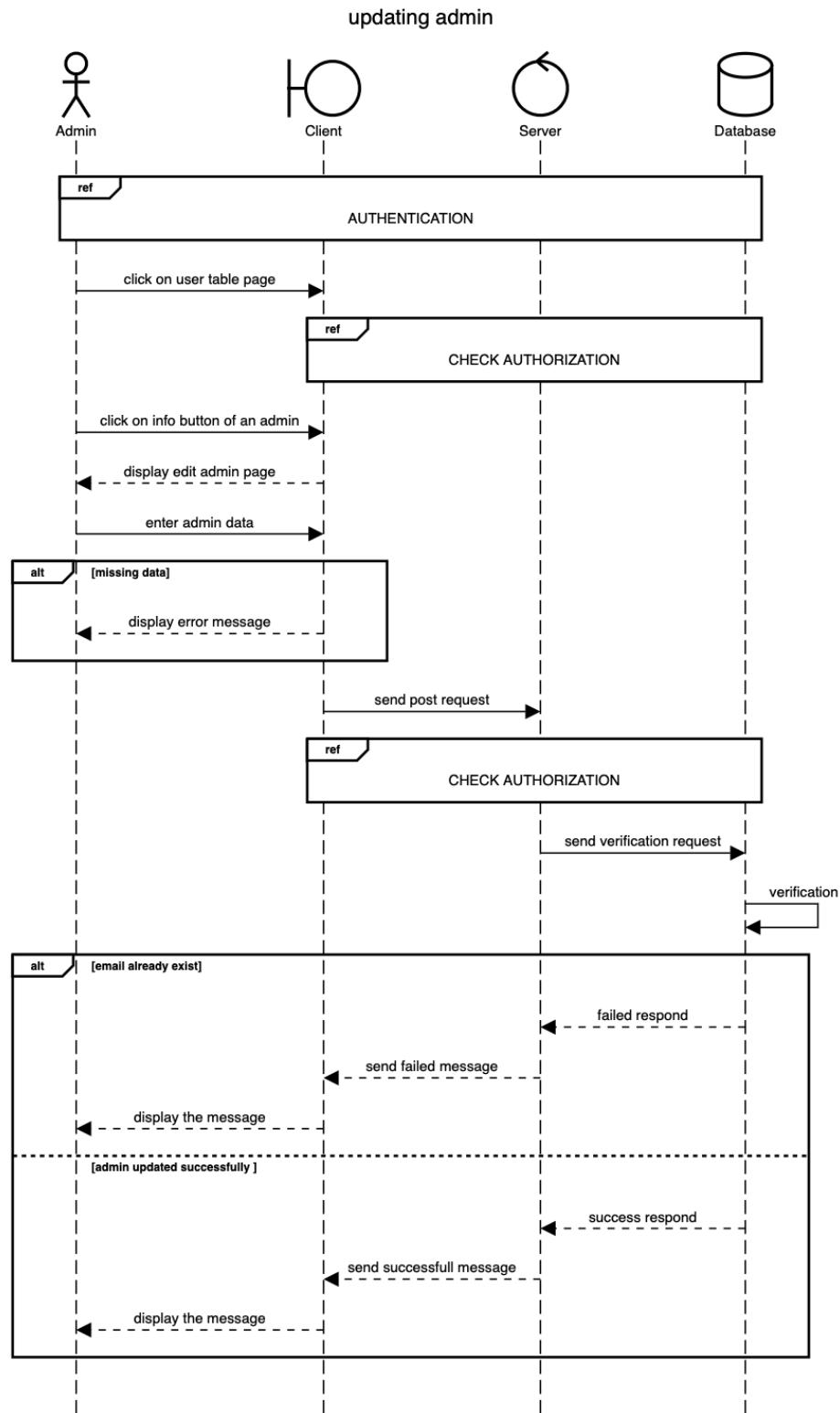


FIGURE 20 Updating Admin Sequence Diagram

2.4. CLASS DIAGRAM

A class diagram allows us to give a general view on the application by the description of the intervening classes and the different relationships between them.

The following figure shows the class diagram for the first sprint:

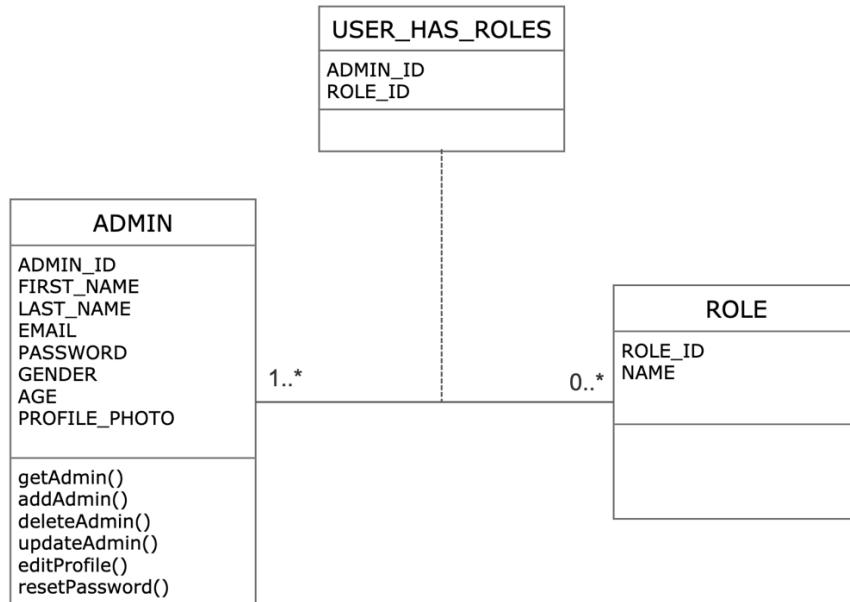


FIGURE 21 Sprint 1 Class Diagram

3. REALIZATION

This part is devoted to the exhibition of work completed through screenshots of different interfaces developed during this sprint.

3.1. AUTHENTICATION INTERFACE

the login page is the first page that it will appear when you access to the web app or redirect to it if your session is over as you can see in this interface, we have the login inputs and the login button also we have redirection to reset password page.

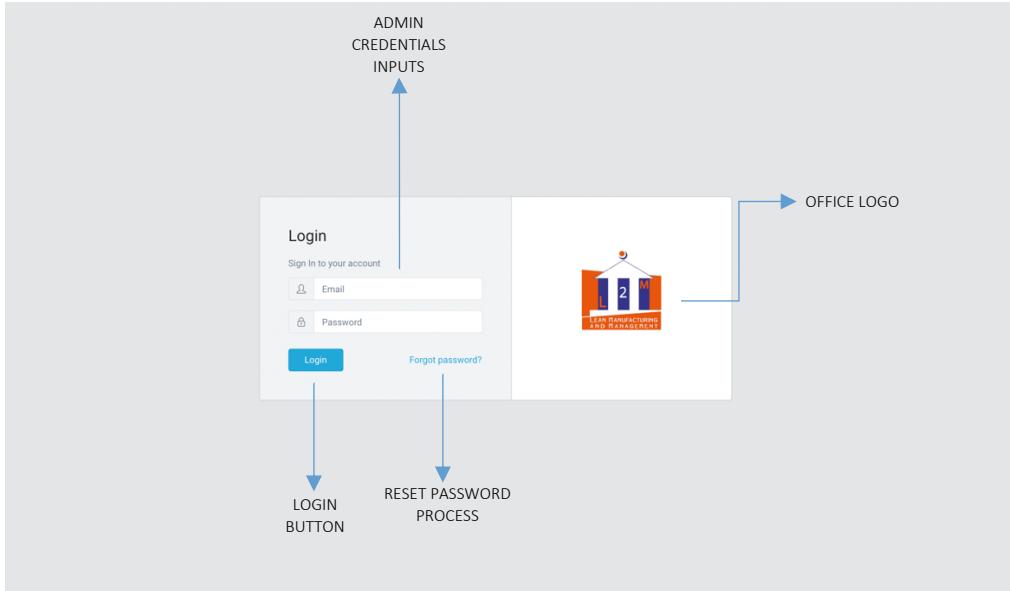


Figure 22 Authentication interface

3.2. ADMIN TABLE INTERFACE

The user table is a page accessed only by super admin and the admin manager in this page we have on top the number of employees of every role we have, also we have a search bar to search the user desired as a button to add new user and in the row of user table we have the roles of the user beside his photo and some info.

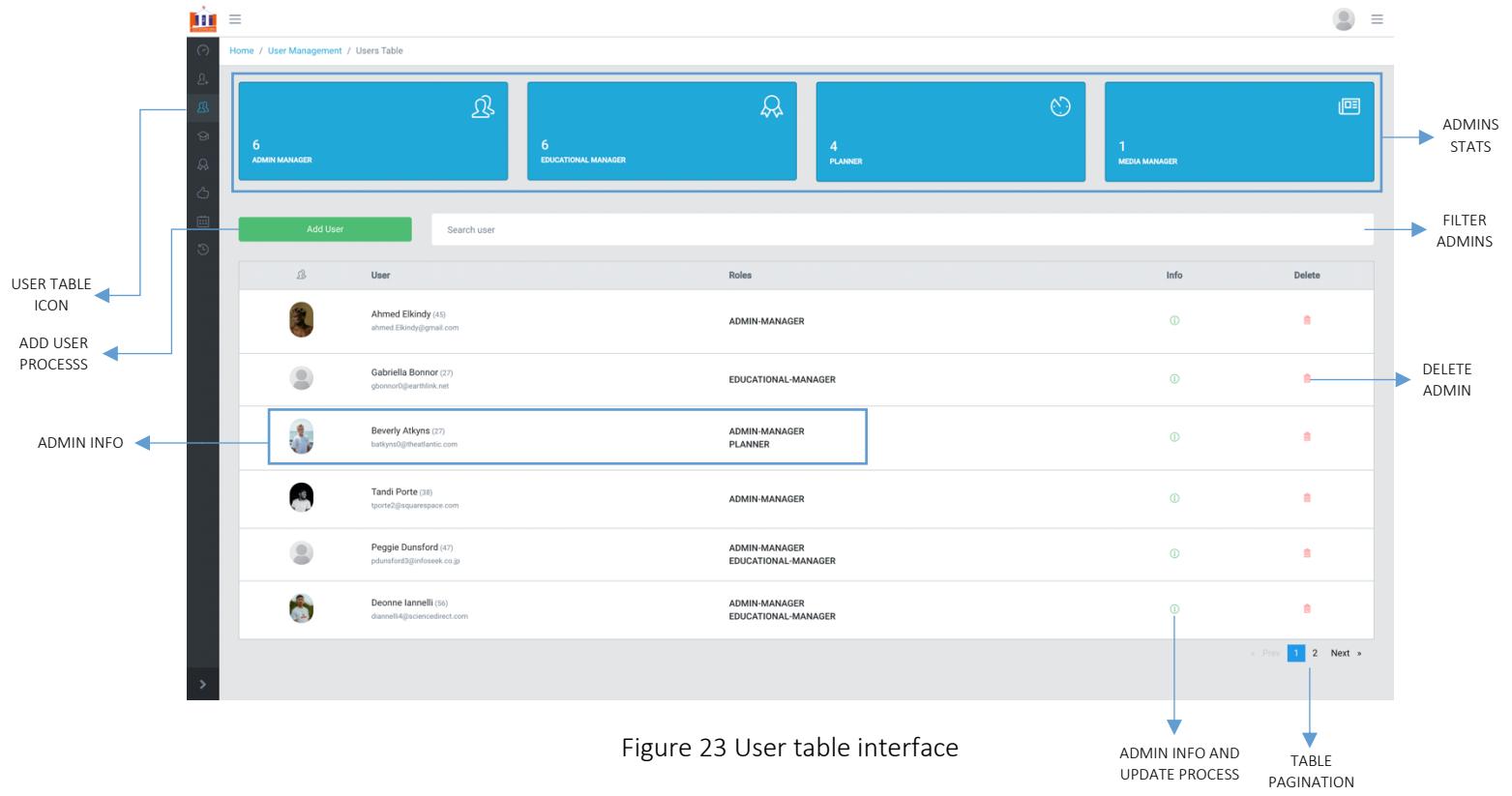


Figure 23 User table interface

3.3. UPDATE ADMIN INTERFACE

The edit user page is the page we can edit user to access this page you have to have authorization in this page we can update user and we can give additional roles to the user as well

The screenshot shows the "Edit user profile" page with the following components:

- Left Sidebar:** Contains icons for Home, User Management, and a search bar.
- Header:** Shows "Home / User Management / Edit user profile".
- User Profile Card:** Displays a user's photo, name (Tandi Porte), email (tporte2@squarespace.com), and roles (Media-Manager, Admin-Manager).
- User Info Form:** Fields for "USER INFO" (Name: Tandi, Email: Porte) and "ROLES" (Media-Manager, Admin-Manager).
- Buttons:** "Save Profile" and "Additional Roles (add role to admin)".

Annotations with arrows point to specific elements:

- LINK BACK TO USER TABLE (points to the back arrow in the sidebar)
- USER INFO (points to the user info card)
- ADDITIONAL ROLES (add role to admin) (points to the "Additional Roles" button)
- UPDATE PROFILE (points to the "Save Profile" button)

Figure 24 Update Admin interface

3.4. ADDING USER INTERFACE

Adding user page is accessed only by an admin manager or super admin as you can see this page have data inputs for adding a user and when the add is completed successfully the admin will be redirected to the user table.

The screenshot shows the 'Add User' interface. On the left, a dark sidebar menu includes 'Dashboard', 'USERS' (with 'Create User' highlighted in blue), 'Users Table', 'PROGRAM' (with 'Certification' and 'Training'), 'PLANIFICATION' (with 'Free Workshop'), 'HISTORY' (with 'History'), and a search bar. The main area shows the 'Add User' form with fields for First Name*, Last Name*, Email*, Password*, Confirm Password*, Age*, Gender (Male/Female), and Role. There is also a placeholder for a profile picture and a blue 'Add User' button.

Figure 25 Add admin interface

3.5. OTHER INTERFACES

• AUTHORIZATION

In our app user with specified roles are the ones that can access some pages and sections for example a planner can't add a user or view user table

The screenshot shows the 'Certification' interface. On the left, a dark sidebar menu includes 'Dashboard', 'USERS' (with 'Create User' highlighted in blue), 'Users Table', 'PROGRAM' (with 'Certification' and 'Training'), and a search bar. The main area shows a form for 'CERTIFICATION' with fields for Title, Price (0, TND), and Description. A red banner in the top right corner displays 'permissions Access denied'.

Figure 26 Authorization pops up

• DELETE ADMIN

When we click on the delete icon in the user table a modal opens for confirmation.

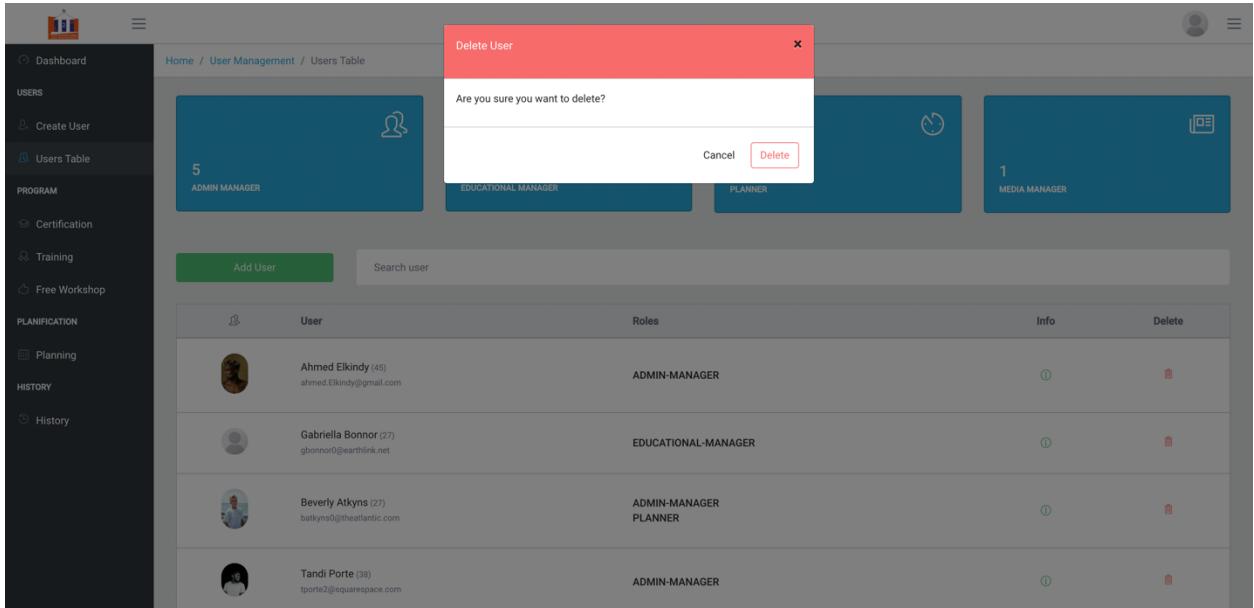


Figure 27 Delete admin modal

4. TESTS

To complete our solution at a very high score we had to do some tests to make sure that all functionalities are working fine and to tests our functionalities we are using “POSTMAN” to send request and do tests.

4.1. AUTHENTICATION TESTS

For authentication we going to apply a login action for invalid email or password,

Then we going to do the same with a valid email and password.

The screenshot shows a POST request to `localhost:3000/auth/signin`. The request body contains:

```

1
2
3 "email": "nejmisidaty@gmail.com",
4 "password": "Zkadpj"
5
6

```

The response status is 401 Unauthorized, with a message: "Invalid email or password!".

Figure 28 Authentication test (1)

The screenshot shows a POST request to `localhost:3000/auth/signin`. The request body contains:

```

1
2
3 "email": "nejmisidaty@gmail.com",
4 "password": "12m1234"
5
6

```

The response status is 200 OK, containing a JSON object with access token, user ID, roles, first name, last name, and profile photo URL.

```

1
2
3 "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYwNWI5Y2NhYjM0NTEyNTIyZWJzM0c5NSIiMhdCI6MTYyMjIzM0I30SwizXhwiJoxNjIyMzIwNjcfQ.42z1L9dq4py1aH6n",
4 "id": "605b9ccab34112522ebc0795",
5 "roles": [
6     {
7         "name": "Admin"
8     }
9 ],
10 "first_name": "Nejmi",
11 "last_name": "Sidaty",
12 "profile_photo": "https://rohsco.rooh.com/wp-content/uploads/sites/9/2019/09/default-profile.png"
13

```

Figure 29 Authentication test (2)

4.2. AUTHORIZATION TEST

To perform any action in this web application you must have authorization to perform desired action to do that we check the role in the token provided with the request decoded and see if his token is valid and if he or she has the role to specified action.

This test is adding user with an admin who have planner role.

The screenshot shows a POST request to `localhost:3000/auth/signup`. The Body tab is selected, displaying the following JSON payload:

```
1 ...
2 ... "first_name" : "ahmed",
3 ... "last_name" : "mohamed",
4 ... "gender" : "Male",
5 ... "age" : "20",
6 ... "email" : "ahmed.mohamed@yahoo.fr",
7 ... "password" : "00000"
8 ...
```

The response status is 500 Internal Server Error, with the message "unAuthorized !".

Figure 30 authorization test

4.3. ADDING USER TESTS

To add an admin his email should not be in use, and he should not be missing any inputs and have at least one role, so we did three test one admin missing a role, admin with a used email and a valid admin to be added

The screenshot shows a POST request to `localhost:3000/auth/signup`. The Body tab contains the following JSON payload:

```
1 {
2   "first_name": "ahmed",
3   "last_name": "mohamed",
4   "gender": "Male",
5   "age": "20",
6   "email": "ahmed.mohamed@yahoo.fr",
7   "password": "00000",
8   "roles": [
9     {"name": "Planner"}
10  ...
11 }
```

The response status is 400 Bad Request, with the message: "Failed! Email is already in use!"

Figure 31 adding admin test (1)

The screenshot shows a POST request to `localhost:3000/auth/signup`. The Body tab contains the same JSON payload as Figure 31.

The response status is 500 Internal Server Error, with the message: "error adding user try to check your inputs"

Figure 32 adding admin test (2)

The screenshot shows the Postman interface with a POST request to `localhost:3000/auth/signup`. The request body is a JSON object:

```

1  {
2   "first_name": "ahmed",
3   "last_name": "mohamed",
4   "gender": "Male",
5   "age": "20",
6   "email": "ahmed.mohamed@yahoo.fr",
7   "password": "000000",
8   "roles": [
9     {
10      "name": "Planner"
11    }
12  ]
13}

```

The response status is 200 OK, with a message: "user was registered successfully". The user object includes profile photo, ID, name, last name, gender, age, email, password, and roles.

Figure 33 adding admin test (3)

4.4. UPDATING USER TESTS

To update a user, we should have the authorization to do it and, we should not update a user with already used email, so we run two test one with a used email and a valid user update.

The screenshot shows a POST request to `localhost:3000/users/updateuser/60b15673a4c65748b95ad6f9`. The request body is a JSON object:

```

1
2   "first_name" : "ahmed",
3   "last_name" : "mohamed",
4   "gender" : "Male",
5   "age" : "20",
6   "email" : "john.doe@gmail.com",
7   "password" : "00000",
8   "roles": [
9     {
10      "name" : "Planner"
11    }
12 ]

```

The response status is `500 Internal Server Error` with a size of `320 B`. The response body is:

```

1
2   "message": "Error updating user"
3

```

Figure 34 updating user test (1)

The screenshot shows a POST request to `localhost:3000/users/updateuser/60b15673a4c65748b95ad6f9`. The request body is a JSON object:

```

1
2   "first_name" : "ahmed",
3   "last_name" : "mohamed",
4   "gender" : "Male",
5   "age" : "25",
6   "email" : "ahmed.mohamed@yahoo.fr",
7   "password" : "00000",
8   "roles": [
9     {
10      "name" : "Planner"
11    }
12 ]

```

The response status is `200 OK` with a size of `665 B`. The response body is:

```

1
2   "message": "User was updated successfully !",
3   "user": {
4     "profile_photo": "https://rohco.roh.com/wp-content/uploads/sites/9/2019/09/default-profile.png",
5     "_id": "60b15673a4c65748b95ad6f9",
6     "first_name": "ahmed",
7     "last_name": "mohamed",
8     "gender": "Male",
9     "age": 25,
10    "email": "ahmed.mohamed@yahoo.fr",
11    "password": "$2b$10$W3dZE4Ub47loJmxh7cmY1eFaTBV7AmuoQPINxZLbGz5JoyBeiSUKC",
12    "roles": [
13      {
14        "name": "Planner"
15      }
16    ],
17    "__v": 0
18  }

```

Figure 35 updating user test (2)

4.5. DELETING USER TESTS

To delete a user, we should have the authorization to do it and, we should not delete the super admin, we already don't display the super admin and the connected admin in the user table but in case of trying to delete the super admin we always have failed respond.

The screenshot shows the Postman interface with a DELETE request to `localhost:3000/users/deleteuser/605b9ccab34112522ebc0795`. The Headers tab is selected, displaying the following configuration:

KEY	VALUE	DESCRIPTION	Bulk Edit	Presets
<input checked="" type="checkbox"/> Postman-Token	<calculated when request is sent>			
<input checked="" type="checkbox"/> Content-Type	application/json			
<input checked="" type="checkbox"/> Content-Length	<calculated when request is sent>			
<input checked="" type="checkbox"/> Host	<calculated when request is sent>			
<input checked="" type="checkbox"/> User-Agent	PostmanRuntime/7.28.0			
<input checked="" type="checkbox"/> Accept	*/*			
<input checked="" type="checkbox"/> Accept-Encoding	gzip, deflate, br			
<input checked="" type="checkbox"/> Connection	keep-alive			
<input checked="" type="checkbox"/> x-access-token	eyJhbGciOiJIUzI1NilsInR5cCI6IkpxVCJ9.eyJpZCI6ijYwNWl5Y2Nh...			

The Body tab shows a JSON response with the following content:

```
1 "message": "Permission denied ! cannot delete Admin user "
```

The status bar at the bottom indicates `Status: 403 Forbidden`, `Time: 187 ms`, and `Size: 333 B`.

Figure 36 deleting user test (1)

The screenshot shows the Postman interface with a DELETE request to `localhost:3000/users/deleteuser/60649e1b9f687c3ebcb215b2`. The Headers tab is selected, showing the following configuration:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Postman-Token	<calculated when request is sent>	
<input checked="" type="checkbox"/> Content-Type	application/json	
<input checked="" type="checkbox"/> Content-Length	<calculated when request is sent>	
<input checked="" type="checkbox"/> Host	<calculated when request is sent>	
<input checked="" type="checkbox"/> User-Agent	PostmanRuntime/7.28.0	
<input checked="" type="checkbox"/> Accept	/*	
<input checked="" type="checkbox"/> Accept-Encoding	gzip, deflate, br	
<input checked="" type="checkbox"/> Connection	keep-alive	
<input checked="" type="checkbox"/> x-access-token	eyJhbGciOiJIUzI1NlslslnR5ccCI6IkpxVCJ9.eyJpZCI6IjYwNWl5Y2Nh..	

The Body tab is selected, showing the JSON response:

```

1
2 "user": {
3   "profile_photo": "https://firebasestorage.googleapis.com/v0/b/storingimages-f1c05.appspot.com/o/images%2Fpixels-blue-bird-7218222.jpg?alt=media&token=eb8d8e",
4   "_id": "60649e1b9f687c3ebcb215b2",
5   "first_name": "Minerva",
6   "last_name": "Gaze",
7   "email": "mgaze1@census.gov",
8   "age": 25,
9   "password": "$2b$10$DbotIdInHuz1DeR/Nr3nz.bBP/DK.tf684eFfbVXh/SNZPvPGryfS",
10  "gender": "Female",
11  "roles": [
12    {
13      "name": "Planner"
14    }
15  ],
16  "__v": 0
17 }
18

```

The status bar at the bottom indicates: Status: 200 OK Time: 273 ms Size: 707 B Save Response.

Figure 37 deleting user test (2)

CONCLUSION

In this chapter, we presented the first sprint in release 1. To do this, we went through analysis, design, and production. In the next chapter we enter the second sprint of release 1.

CHAPTER 4 « STUDY AND REALIZATION OF SPRINT 2»

INTRODUCTION

In this chapter, we present the realization of the second sprint in release one, by organizing the work on main phases which are the analysis, the realization, and the tests.

1. SPRINT BACKLOG

The sprint is the heart of Scrum. This is a block of time during which an increment of the product will be made. All sprints in a release have a constant duration and never overlap, that is, a sprint cannot start until the previous one is finished. Before embarking on a sprint, the Scrum team must define the goal of the latter, which must be a descriptive table that specifies the workload for each task in number of days.

ID	TASKS	ESTIMATION (days)
1	Add training	1
2	Update training	1
3	Delete training	1
4	Add certification	1
5	Update certification	1
6	Delete certification	1
7	manage training in certification	2
8	Plan certification session	2
9	Delete certification session	1
10	Plan session journeys	2
11	Delete session journeys	1

12	Plan and manage free workshops	3
----	--------------------------------	---

Table 9 sprint 1 backlog

2. CONCEPTION

In this section we present the analysis phase that answers the question "what does the system".

The answer to this question is reflected in the presentation of the use case diagram and the textual description of each.

2.1. USE CASE DIAGRAM

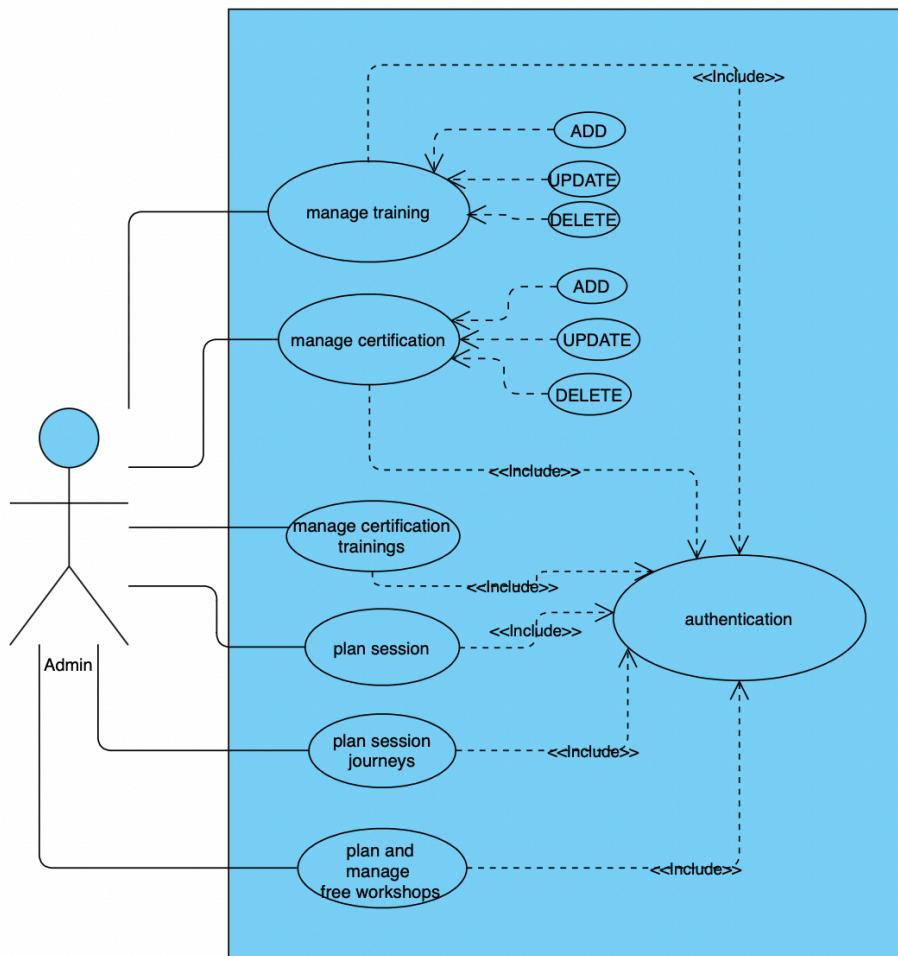


Figure 38 sprint 2 use case diagram

2.2. TEXTUAL DESCRIPTION OF USE CASES

- Text description of use case “Add training”

TITLE	Add training
ACTORS	Super admin and planner
SUMMARY	In this case the super or planner admin can add a new training so it can be added to a certification
PRE-CONDITION	The actor must be authenticated to perform the task and got the authorization to the page
PRINCIPAL SCENARIO	<ol style="list-style-type: none">1. Actor access to training management page2. Actor enter training data3. Actor add modules to training4. The system verifies the data5. The system displays a success notification to the actor.6. The training can now be added to a certification
POST-CONDITION	Training added to training table and can be updated or get deleted

Table 10 Description text Add training

- Text description of use case “Add certification”

TITLE	Add certification
ACTORS	Super admin and planner
SUMMARY	In this case the super or planner admin can add a new certification

PRE-CONDITION	The actor must be authenticated to perform the task and got the authorization to the page
PRINCIPAL SCENARIO	<ol style="list-style-type: none"> 1. Actor access to certification management page 2. Actor enter certification data 3. The system verifies the data 4. The system displays a success notification to the actor 5. The certification now can be planned
POST-CONDITION	Certification added to certification table and can be updated, get deleted or plan session for it

Table 11 Description text Add certification

- Text description of use case “Manage certification trainings”

TITLE	Manage certification trainings
ACTORS	Super admin and planner
SUMMARY	In this case the super or planner admin can add or delete a training from certification
PRE-CONDITION	The actor must be authenticated to perform the task and got the authorization to the page
PRINCIPAL SCENARIO	<ol style="list-style-type: none"> 1. Actor access to info page of one certification 2. Actor view a table of all trainings in the system 3. Actor search for the specific training to be added or deleted 4. Actor choose the action desired “add or delete”

	5. The system displays a success notification to the actor 6. The certification is updated
POST-CONDITION	Certification is updated

Table 12 Description text manage certification trainings

- Text description of use case “Plan certification session”

TITLE	Plan certification session
ACTORS	Super admin and planner
SUMMARY	In this case the super or planner admin can plan session for a certification
PRE-CONDITION	The actor must be authenticated to perform the task and got the authorization to the page
PRINCIPAL SCENARIO	1. Actor access to sessions page of one certification 2. Actor put place and dates for the session 3. The system verifies the data and display the success notification message 4. Session is planned
POST-CONDITION	Session is planned and added to the certification sessions table and now we can add journeys to it

Table 13 Description text plan certification session

- Text description of use case “Plan session journey”

TITLE	Plan session journey
ACTORS	Super admin and planner

SUMMARY	In this case the super or planner admin can add or delete a training from certification
PRE-CONDITION	The actor must be authenticated to perform the task and got the authorization to the page
PRINCIPAL SCENARIO	<ol style="list-style-type: none"> 1. Actor access to training days page of one session 2. Actor put training and modules that will be taught in that day and the date of the journey 3. The system verifies the data and display the success notification message 4. Journey of a session is planned
POST-CONDITION	Journey is planned and added to training days table in the session

Table 14 Description text Plan session journey

- **Text description of use case “Plan free workshop”**

TITLE	Plan free workshop
ACTORS	Super admin and planner
SUMMARY	In this case the super or planner admin can plan free workshop
PRE-CONDITION	The actor must be authenticated to perform the task and got the authorization to the page
PRINCIPAL SCENARIO	<ol style="list-style-type: none"> 1. Actor access to free workshop page 2. Actor enter free workshop data 3. The system verifies the data 4. The system sends notification success message

POST-CONDITION	Free workshop is planned and added to the free workshops table and can now be updated and deleted
----------------	---

Table 15 Description text Plan free workshop

2.3. SEQUENCE DIAGRAM

Sequence diagrams are the graphical representation of interactions between actors and the system in chronological order in the UML formulation. We present thereafter the system sequence diagrams of the most important classified use case.

- “Adding training” sequence diagram

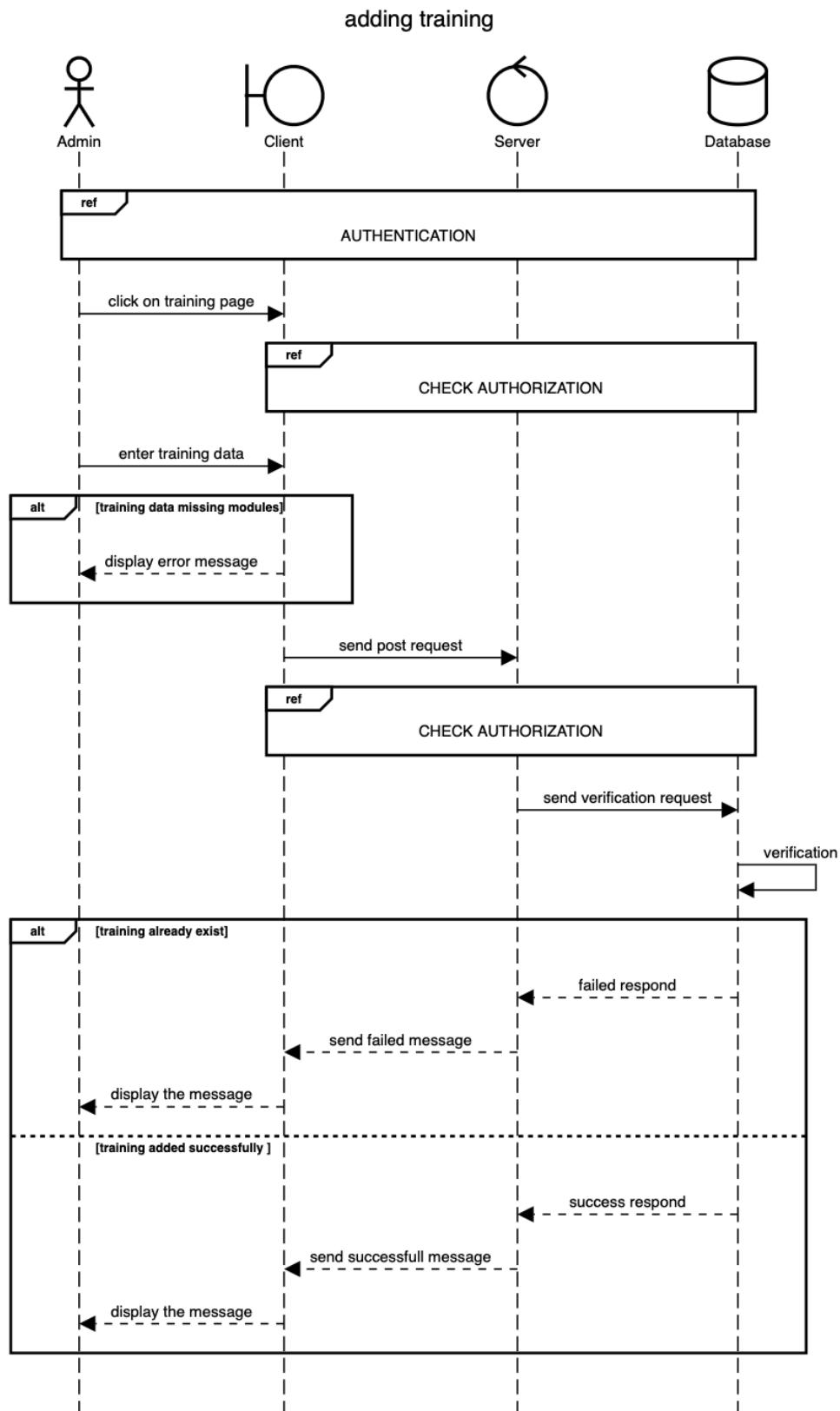


FIGURE 39 Adding Training Sequence Diagram

- “Plan Session” sequence diagram

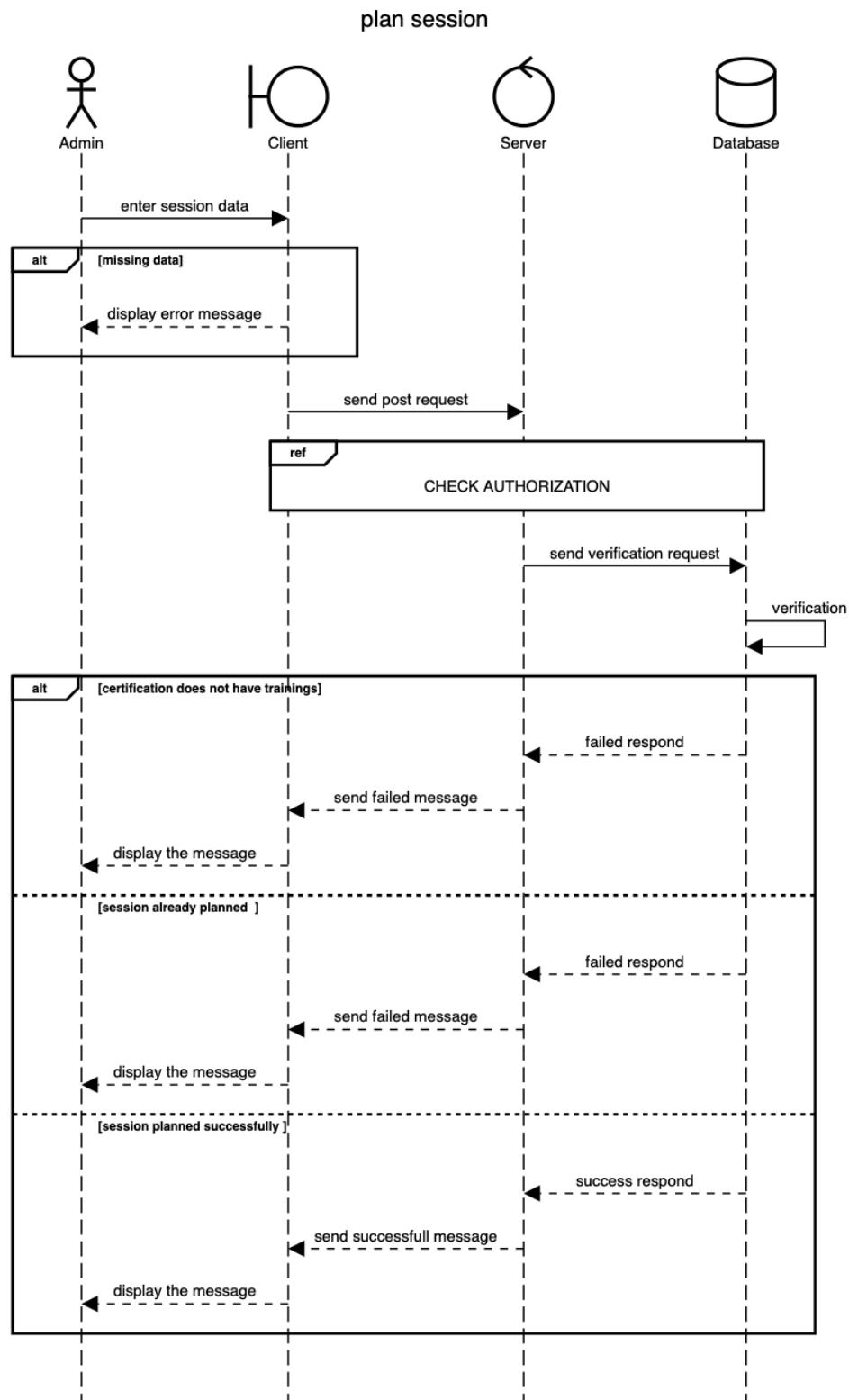


FIGURE 40 Plan Session Sequence Diagram

- “Adding Certification”

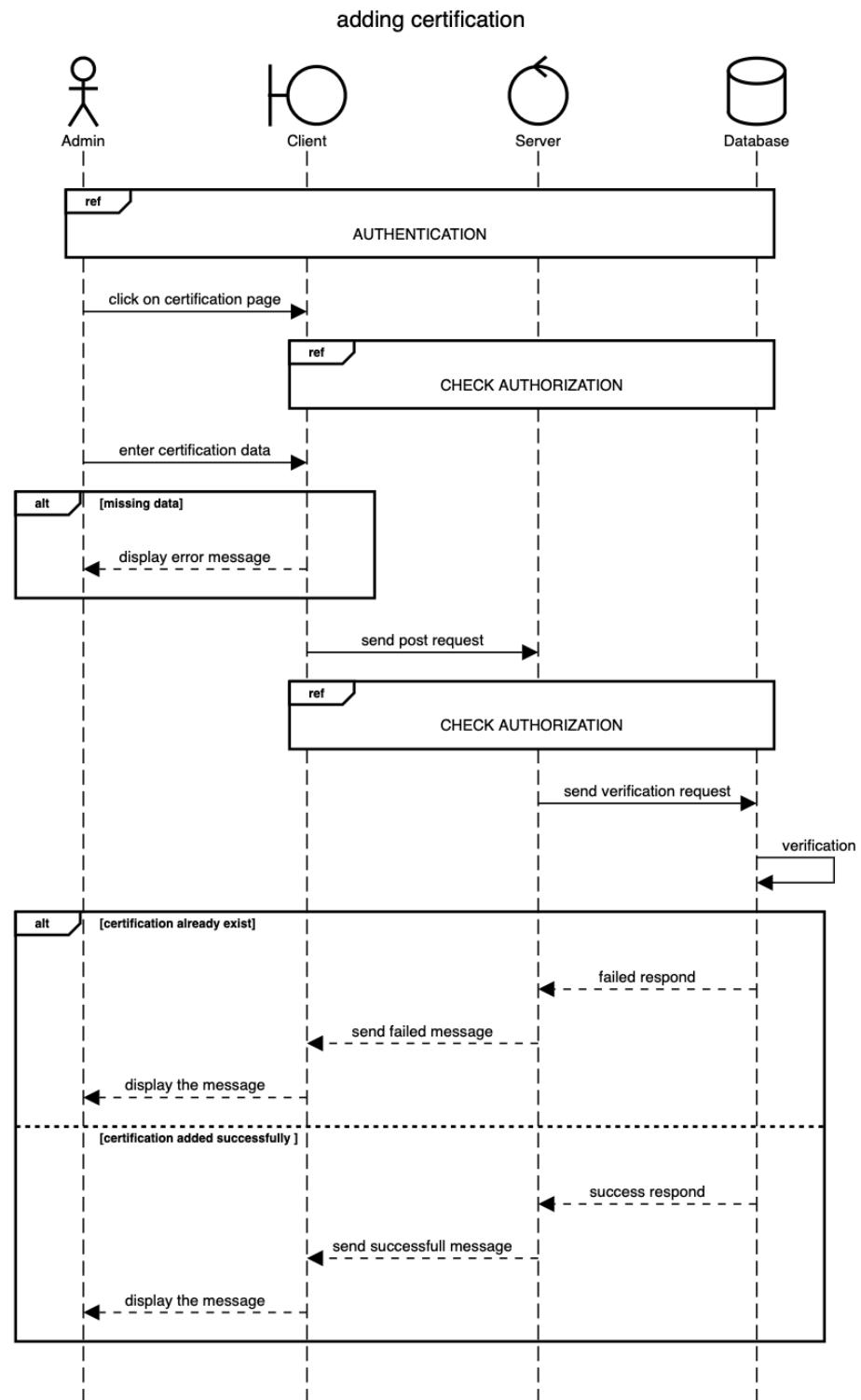


FIGURE 41 Add Certification Sequence Diagram

- “Plan session journey” sequence diagram

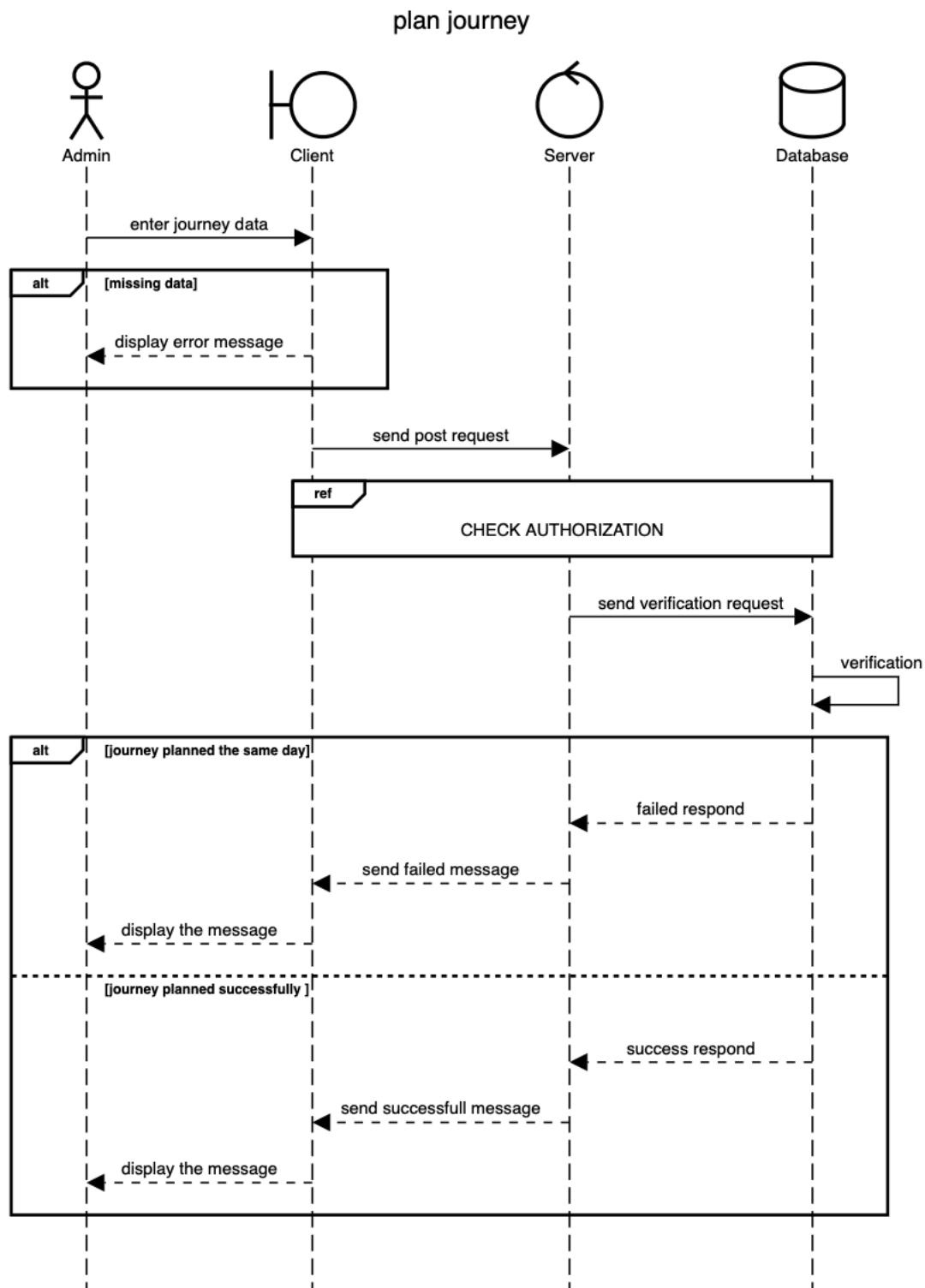


FIGURE 42 Plan Journey Sequence Diagram

- “Plan free workshop” sequence diagram

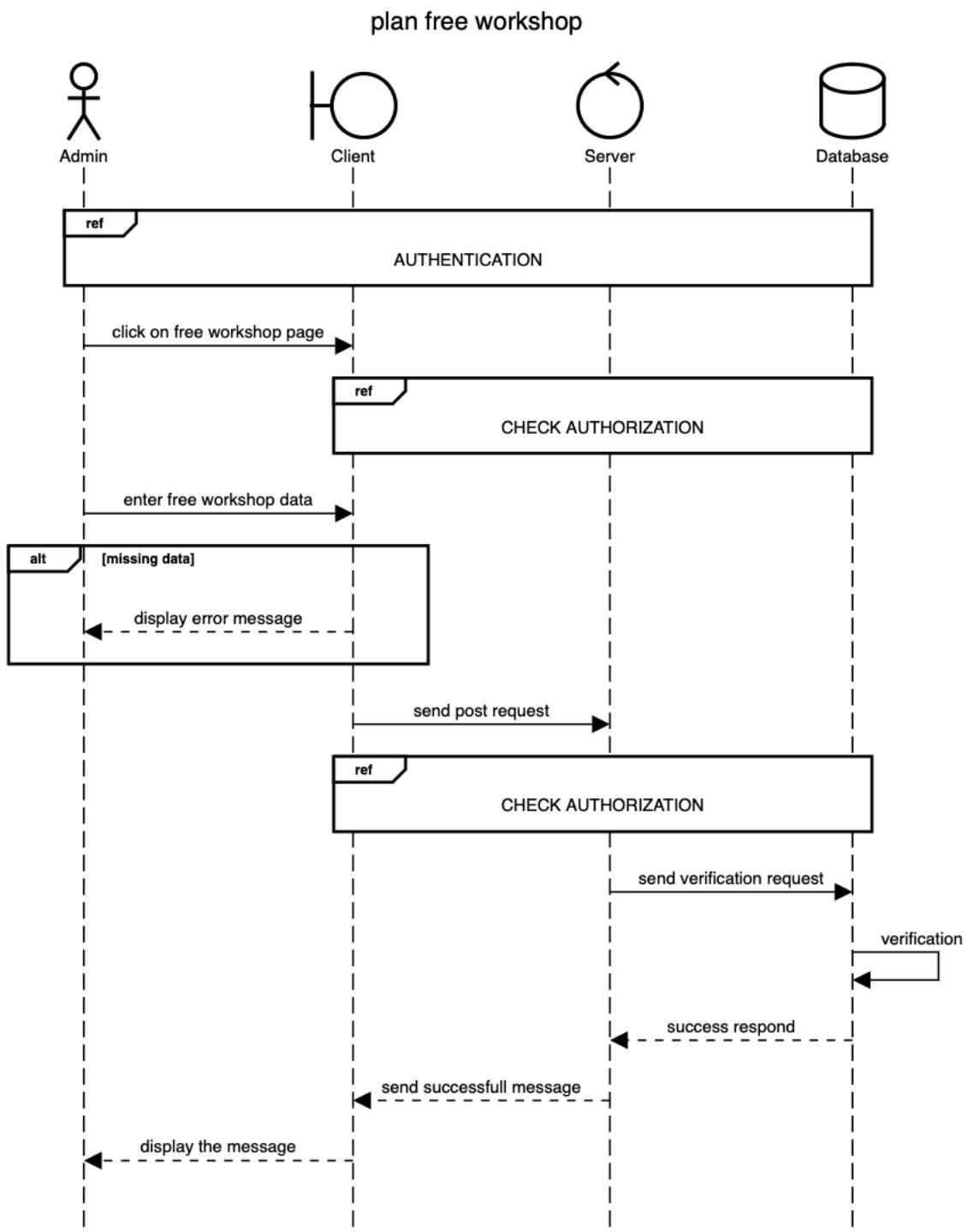


FIGURE 43 Plan Free Workshop Sequence Diagram

2.4. CLASS DIAGRAM

A class diagram allows us to give a general view on the application by the description of the intervening classes and the different relationships between them.

The following figure shows the class diagram for the second sprint:

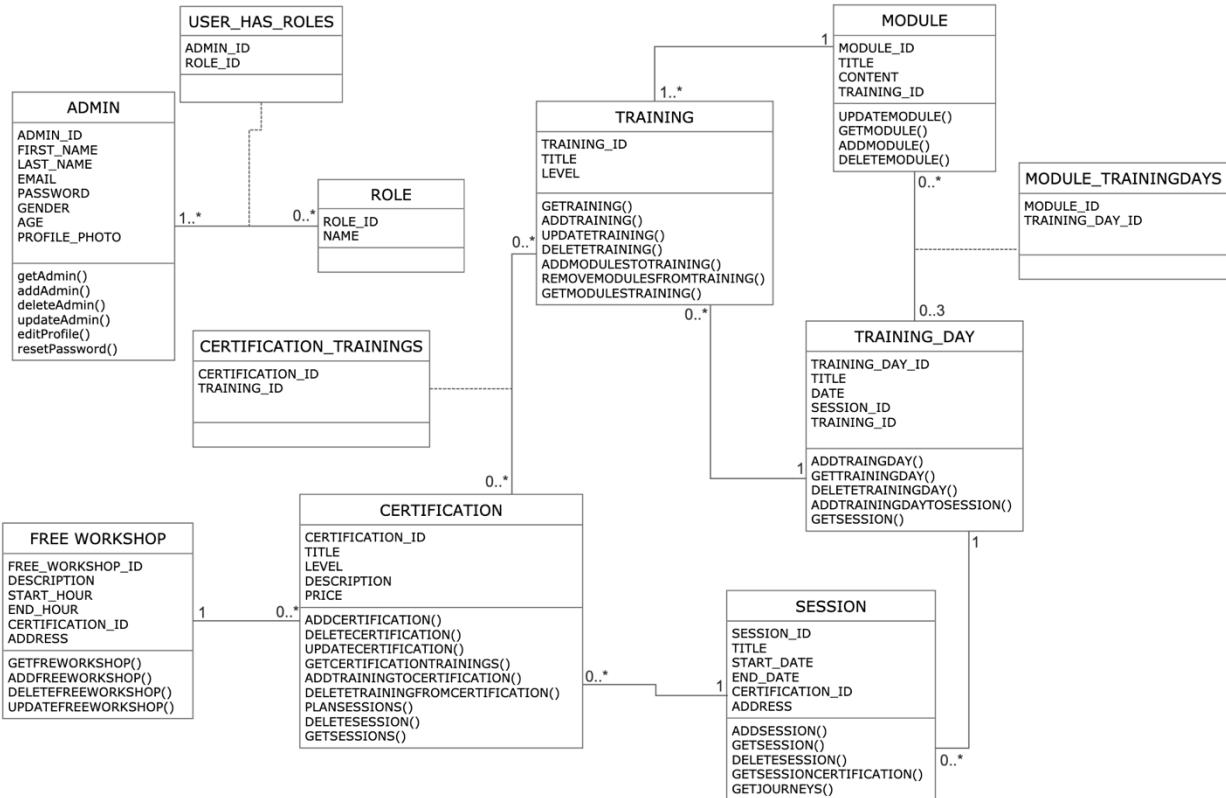


FIGURE 44 Sprint 2 Class Diagram

3. REALIZATION

This part is devoted to the exhibition of work completed through screenshots of different interfaces developed during this sprint.

3.1. ADD, DELETE TRAINING INTERFACE

This interface is only accessed by two actors the super admin and the planner, this page as you see is divided into two section one is adding a training and this other is the training table, in the add training section we have the training info like title and the level also we have box to keep adding modules to the training before it added.

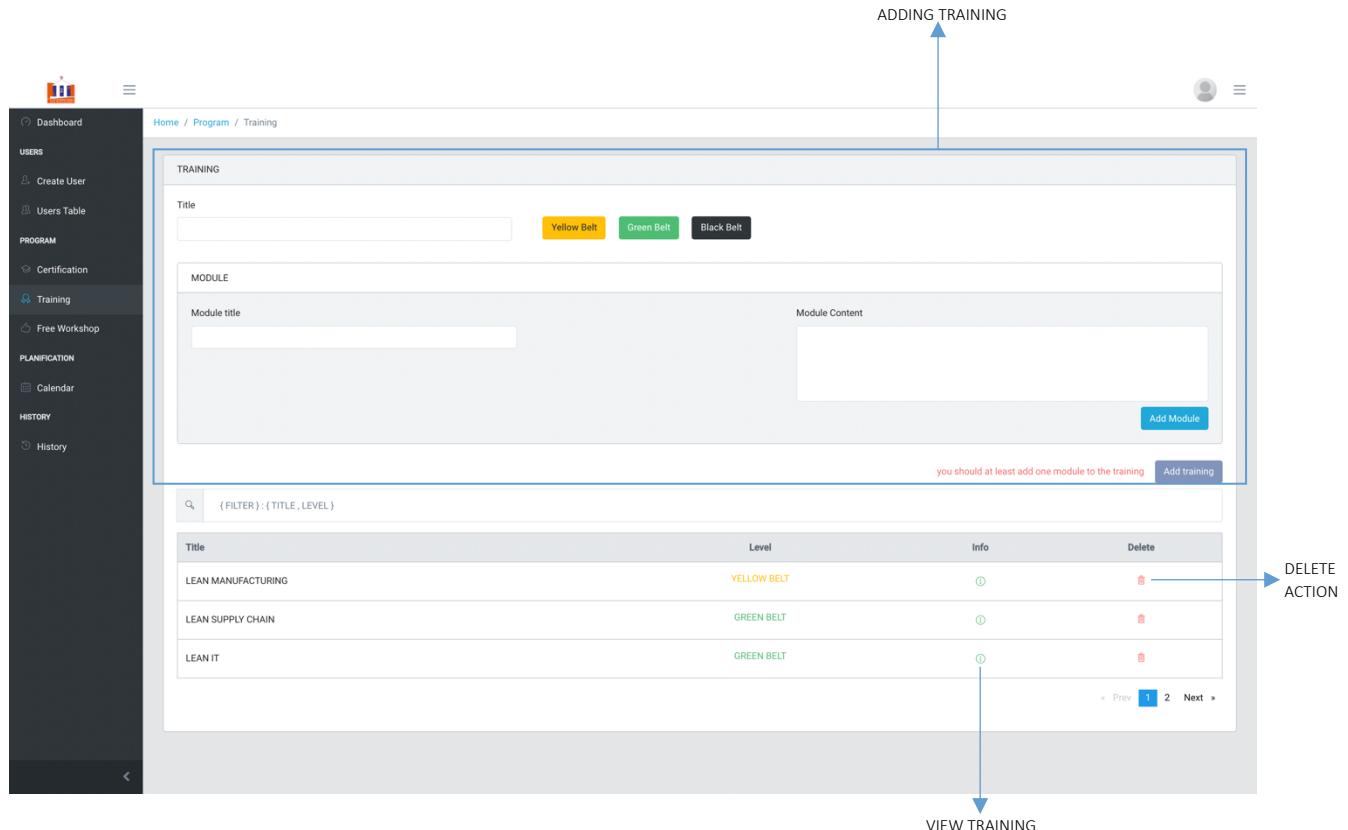


Figure 45 Add, delete training interface

3.2. VIEW, EDIT TRAINING INTERFACE

When we click on the info button of a training in the training table you will be redirected to this page.

This page contains on the left the training title and the level as also the module inside and on the right, we got the update box as we can update the training, and under that we go list of modules that can be deleted and edited by clicking on the action desired.

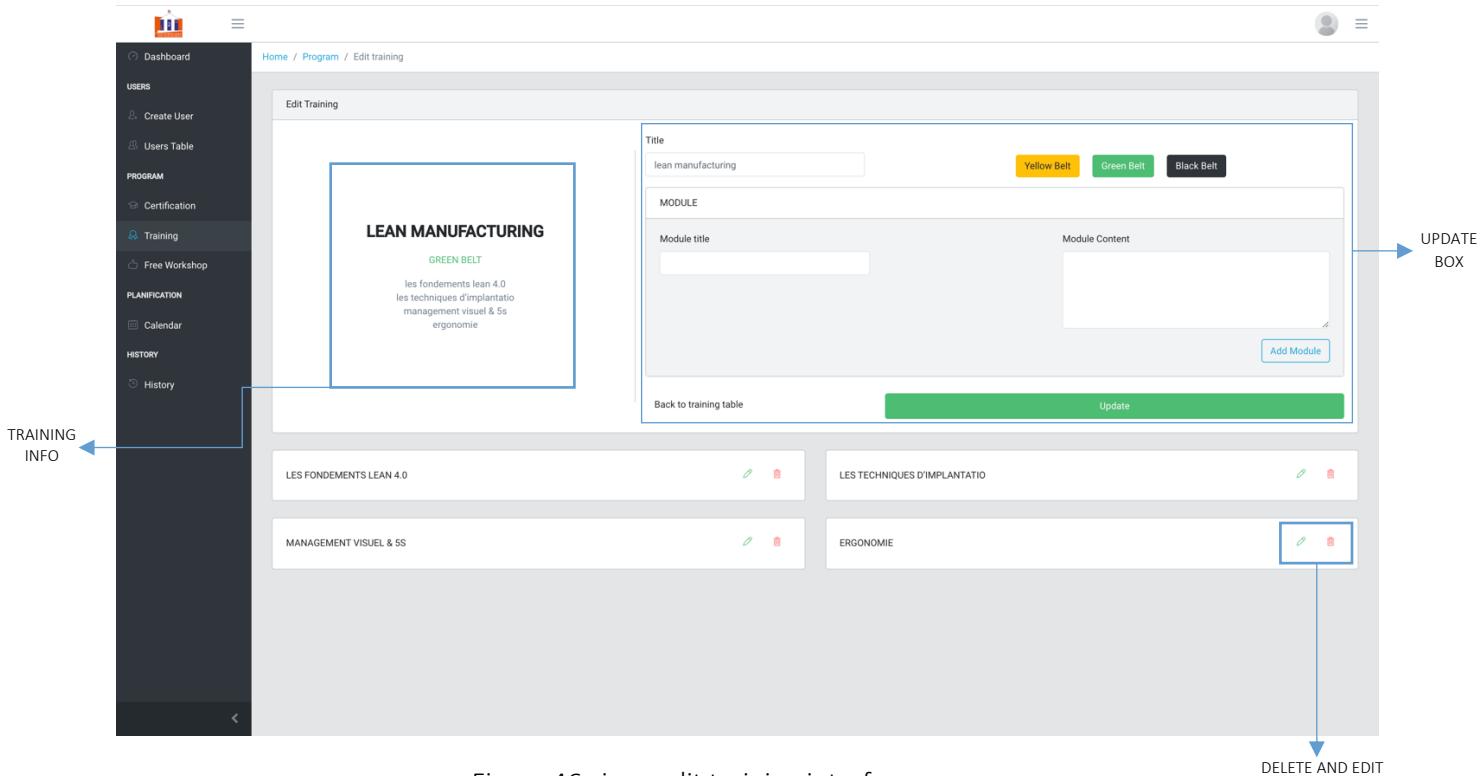


Figure 46 view, edit training interface

3.3. UPDATE MODULE INTERFACE

This a modal that appears when we click on the edit action of a module inside a training its simple modal that contains the title and the content inside the module.

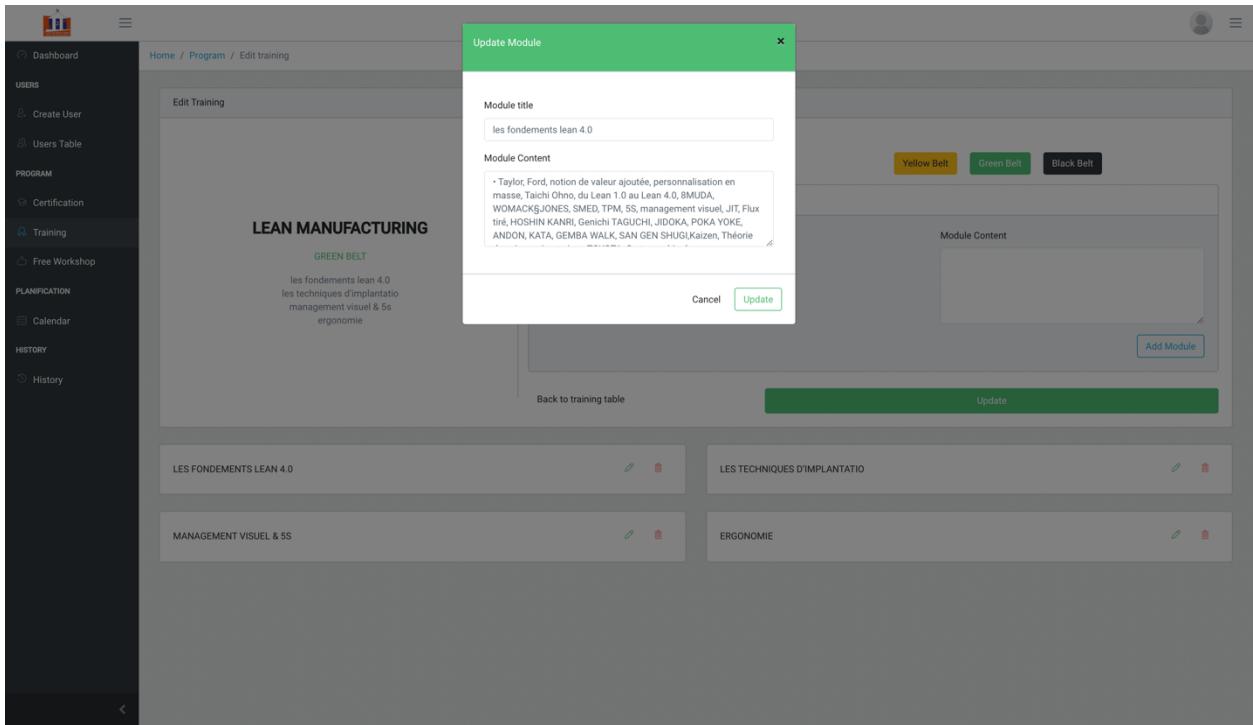


Figure 47 update module interface

3.4. ADD, DELETE CERTIFICATION INTERFACE

This interface is accessed by two actors the super admin and the planner, this page as you see is divided into two section one is adding a certification and this other is the certification table, in the add certification section we have the certification info as like title and the level, and in the certification table we have three actions, deleting certification , info action to edit and see the certification and the add session to add a certification's session.

The screenshot shows a web-based application interface for managing certifications. On the left is a dark sidebar with various menu items: Dashboard, USERS (Create User, Users Table), PROGRAM (Certification, Training, Free Workshop), PLANIFICATION (Calendar), MEDIA (Feed Post), and HISTORY (History). The main content area is titled 'CERTIFICATION'. At the top, there's a form for adding a certification with fields for Title, Price (0 TND), and three buttons: Yellow Belt (highlighted in yellow), Green Belt, and Black Belt. Below this is a table listing existing certifications:

Title	Level	Add sessions	Info	Delete
LEAN SIX SIGMA 3245 TND	YELLOW BELT	<input type="button" value="Add"/>	<input type="button" value="Info"/>	<input type="button" value="Delete"/>
TEST 320 TND	YELLOW BELT	<input type="button" value="Add"/>	<input type="button" value="Info"/>	<input type="button" value="Delete"/>
LEAN 4.0 3340 TND	GREEN BELT	<input type="button" value="Add"/>	<input type="button" value="Info"/>	<input type="button" value="Delete"/>
NEW TEST 3454 TND	YELLOW BELT	<input type="button" value="Add"/>	<input type="button" value="Info"/>	<input type="button" value="Delete"/>

At the bottom of the table, there are two notes: "*Certification in red have 0 trainings" and "*Certification in yellow have 0 sessions". A blue arrow points from the 'Adding Certification Box' label at the top right towards the 'Add sessions' button for the 'GREEN BELT' certification. Another blue arrow points from the 'View and Delete Action' label at the bottom right towards the 'Delete' button for the same certification.

Figure 48 Add, delete certification interface

3.5. VIEW, EDIT CERTIFICATION INTERFACE

The router will take you to this page when you click on the info action in the certification table as you can see is dedicated to one certification we have on the top some info as the price , number of trainings inside the certification and also number of sessions that this certification got we have under that on the left certification info , and on the right the update box , under that we got a table of trainings that are in our system we can add and remove to and from the certification by the actions available.

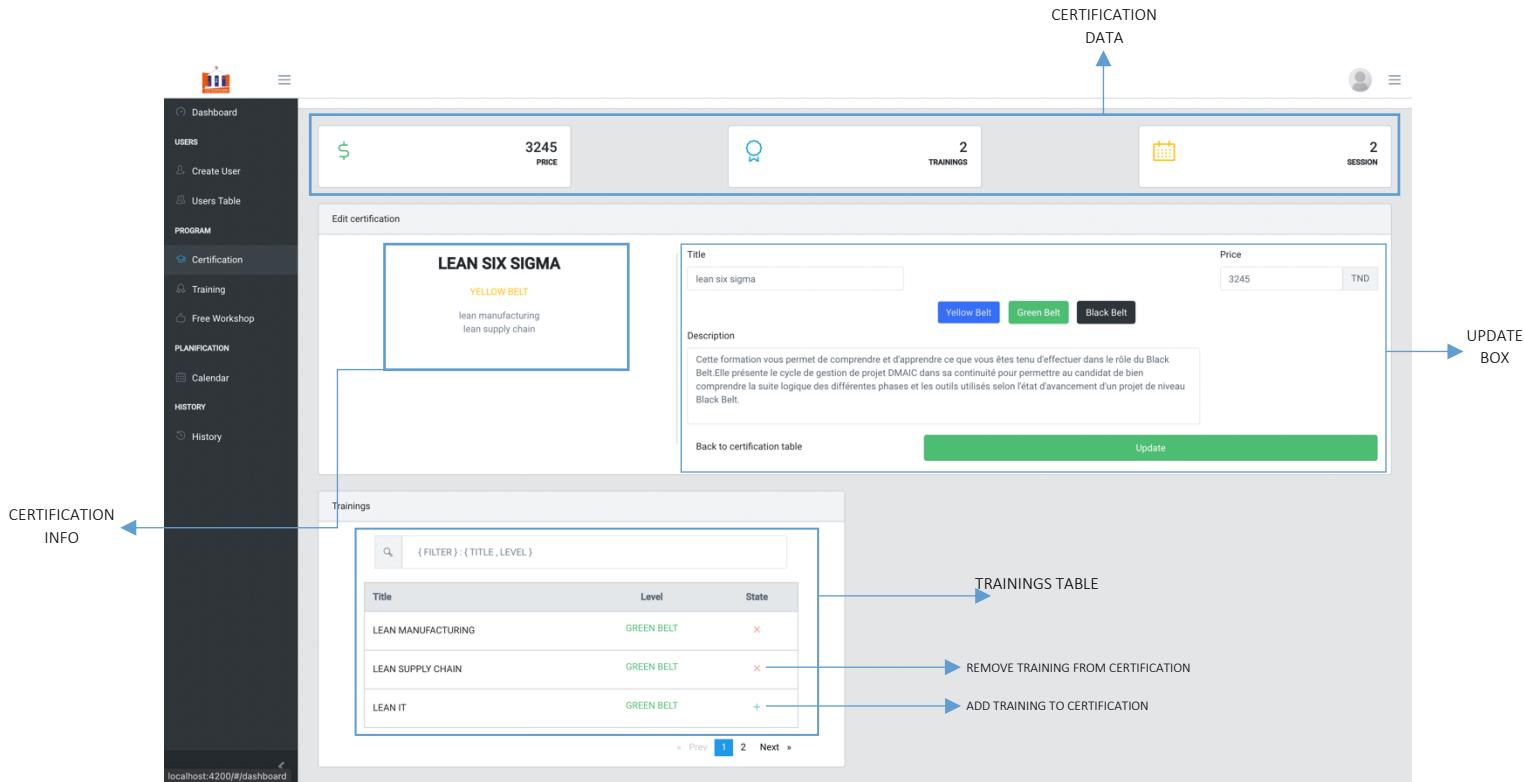


Figure 49 View, edit certification interface

3.6. PLAN, DELETE SESSION INTERFACE

When we click on adding a session to a certification will be relocated here and, in this page, we can add many sessions and also, we have a table of sessions already created and three actions is adding training days "journeys" to the session, session info page or deleting a session

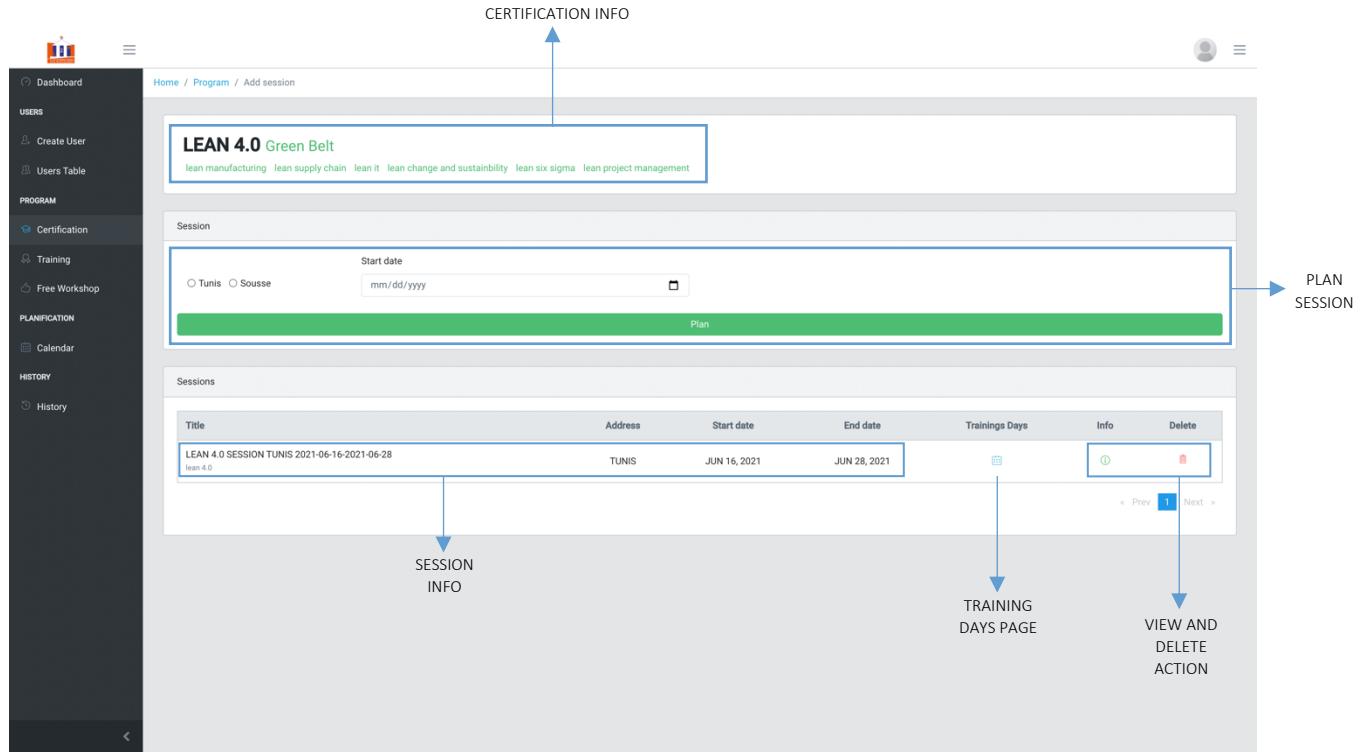


Figure 50 Plan, delete session interface

3.7. VIEW SESSION INTERFACE

This page is the info page of one session we have some information of the certification of this session and the dates of the session, we also have trainings inside the certification and the modules that will be taught through the journeys of the session that why we implemented a calendar, so the employee has clear view of upcoming journeys.

The screenshot shows a web-based application interface for managing training sessions. On the left, a dark sidebar menu includes 'Dashboard', 'USERS' (with 'Create User' and 'Users Table'), 'PROGRAM' (with 'Certification', 'Training', 'Free Workshop'), 'PLANIFICATION' (with 'Calendar'), and 'HISTORY' (with 'History'). The main content area is titled 'Session' and displays a summary for a 'LEAN 4.0 Green Belt' session. The session details are as follows:

- CERTIFICATION:** Lean 4.0
- ADDRESS:** Tunis
- DATE:** Jun 16, 2021 - Jun 28, 2021
- TRAINING DAYS:** 11

Below this, there are sections for 'Certification Trainings' (listing LEAN MANUFACTURING, LEAN SUPPLY CHAIN, LEAN IT, LEAN CHANGE AND SUSTAINABILITY, LEAN SIX SIGMA, LEAN PROJECT MANAGEMENT) and 'Training Days' (a calendar for May 2021). To the right, there are detailed sections for 'CERTIFICATION INFO' (including 'LES FONDEMENTS LEAN 4.0' and 'LES TECHNIQUES D'IMPLANTATION'), 'MANAGEMENT VISUEL & SS' (including 'HOSHIN Classification'), 'ERGONOMIE', and 'INTRODUCTION GÉNÉRALE DES NOTIONS DE LA SUPPLY CHAIN'.

Figure 51 view session interface (1)

This screenshot shows the same application interface as Figure 51, but with a specific history entry highlighted. A blue arrow points from the 'SESSION JOURNEY' label on the left to a green dot on the calendar for June 2021, indicating a past event. The history entry is labeled 'localhost:4200/#users/create'. The rest of the interface is identical to Figure 51, displaying session details, training modules, and various implementation techniques.

Figure 52 view session interface (2)

3.8. PLAN, DELETE JOURNEY INTERFACE

In this page we can plan and delete a journey from a session, we have some information about the session on top and section that we can plan a journey with the modules inside of it and we have a table of journeys inside the session we can view and delete them.

The screenshot shows a web-based application interface for managing training sessions. On the left is a dark sidebar with navigation links for Dashboard, USERS, PROGRAM, and PLANNIFICATION. The main content area is titled 'Edit session' and displays session details: Certification (Lean 4.0), Address (Tunis), Date (Jun 16, 2021 - Jun 28, 2021), and Training Days (11). Below this is a 'Training Journey' section with fields for Date (06/18/2021), Trainings (lean supply chain), and Modules (modules), with a 'plan' button. To the right is a 'Training Days' table listing days from June 16 to 21, 2021, each associated with a session (Lean 4.0) and location (Tunis Jun 16, 2021 - Jun 28, 2021). The table includes columns for Day, Month - Year, Session, Info, and Delete. Arrows on the left point to the sidebar and the table, while arrows on the right point to the 'plan' button and the table's delete column.

Day	Month - Year	Session	Info	Delete
16 Wednesday	Jun 2021	Lean 4.0 Tunis Jun 16, 2021 - Jun 28, 2021	Info	Delete
17 Thursday	Jun 2021	Lean 4.0 Tunis Jun 16, 2021 - Jun 28, 2021	Info	Delete
18 Friday	Jun 2021	Lean 4.0 Tunis Jun 16, 2021 - Jun 28, 2021	Info	Delete
19 Saturday	Jun 2021	Lean 4.0 Tunis Jun 16, 2021 - Jun 28, 2021	Info	Delete
20 Sunday	Jun 2021	Lean 4.0 Tunis Jun 16, 2021 - Jun 28, 2021	Info	Delete
21 Monday	Jun 2021	Lean 4.0 Tunis Jun 16, 2021 - Jun 28, 2021	Info	Delete

Figure 53 plan, delete journey interface

3.9. VIEW JOURNEY INTERFACE

This page shows some info about a journey inside a session we have the big title is the training title that will be taught in that journey and under the certification that belongs to it also we have the modules that will be taught through the day.

The screenshot shows a web-based application interface for managing training programs. On the left, a sidebar menu includes options like Dashboard, USERS, PROGRAM, and HISTORY. The main content area displays a 'Training Day' for 'LEAN MANUFACTURING Green Belt'. This section includes details such as 'Lean 4.0', 'TUNIS - JUN 16, 2021', and 'LES FONDEMENTS LEAN 4.0 LES TECHNIQUES D'IMPLANTATION'. Below this, there are four boxes: 'CERTIFICATION' (Lean 4.0), 'ADDRESS' (Tunis), 'DATE' (Jun 16, 2021), and 'TRAINING' (Lean Manufacturing). A blue arrow points from the text 'TRAINING INFO' at the top right towards the 'TRAINING' box.

Figure 54 view journey interface

3.10. ADD, DELETE FREE WORKSHOP INTERFACE

This interface shows two sections one is for adding new free workshop with the specific information and if its online or presential and the other section is a table with a search bar to filter free workshops that we have in the system and two actions as info and delete actions.

The screenshot displays the 'Free Workshop' section of a software application. On the left, a sidebar menu includes 'Dashboard', 'USERS' (with 'Create User' and 'Users Table'), 'PROGRAM' (with 'Certification' and 'Training'), 'Free Workshop' (selected), 'PLANIFICATION' (with 'Calendar'), and 'HISTORY' (with 'History'). The main area shows a 'Free Workshop List' with a single entry:

Title	Type	Date	Info	Delete
LEAN FREE WORKSHOP 8:00 AM - 9:30 AM	ONLINE	27-05-2021	Edit	Delete

Below the list is a navigation bar with 'Prev' (disabled), '1', and 'Next'. A blue arrow labeled 'FREE WORKSHOP INFO' points from the sidebar to the list. Another blue arrow labeled 'VIEW AND DELETE ACTION' points from the 'Delete' link in the list to the 'Delete' link in the list row.

ADDING FREE WORKSHOP BOX

FREE WORKSHOP INFO

VIEW AND DELETE ACTION

Today is May 26, 2021

Free Workshop List

Search

Title

Type

Date

Info

Delete

**LEAN FREE WORKSHOP
8:00 AM - 9:30 AM**

ONLINE

27-05-2021

Edit

Delete

Prev **1** **Next**

Figure 55 add, delete free workshop interface

3.11. VIEW FREE WORKSHOP INTERFACE

This page shows all the information of a free workshop as also give the possibility to update this free workshop.

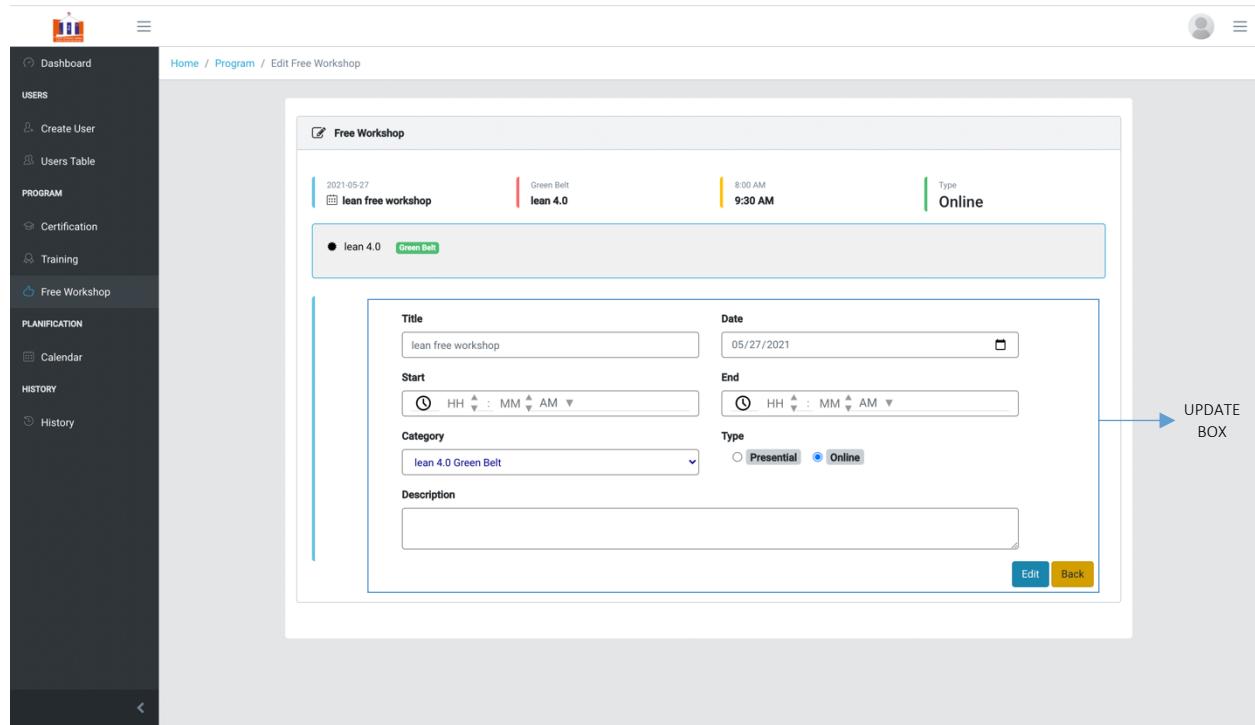


Figure 56 view free workshop interface

3.12. CALENDAR VIEW INTERFACE

This is just a simple page that only accessed by the super admin and the planner, that shows two calendar one for the sessions and the other is for the free workshop and when we click on the required session or the free workshop you will be taken to the view page of it and get some more information about it.

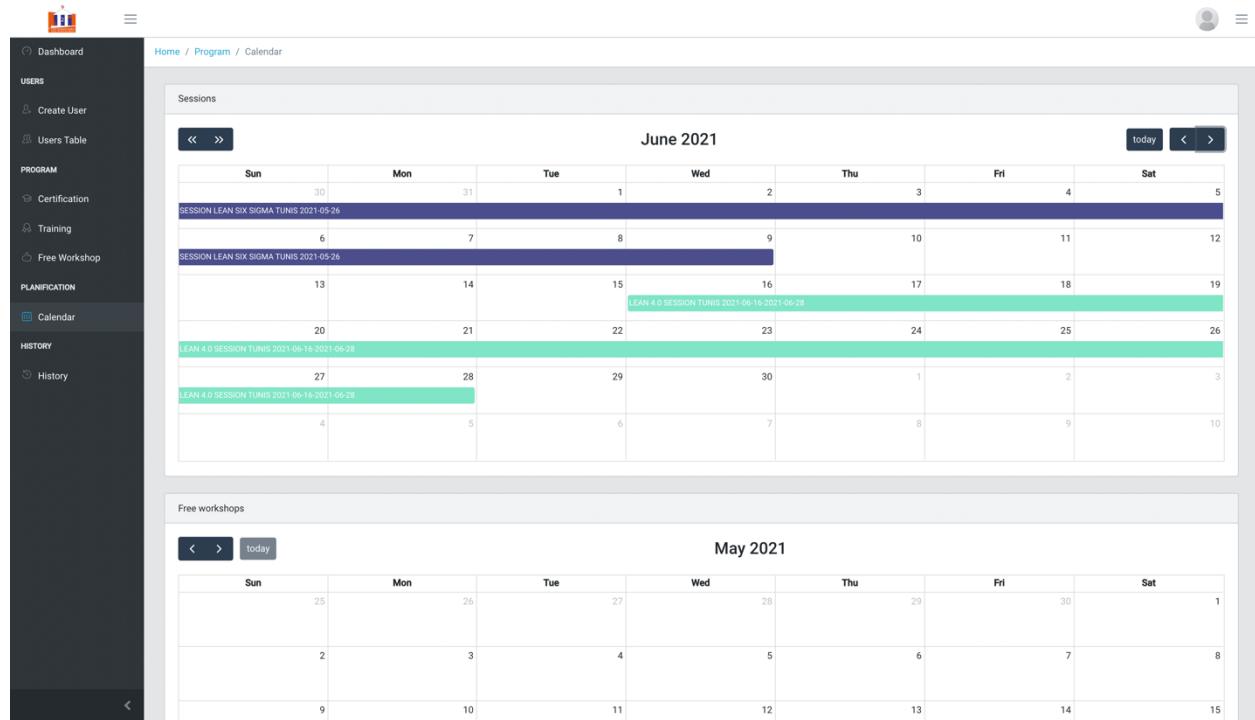


Figure 57 calendar view interface (1)

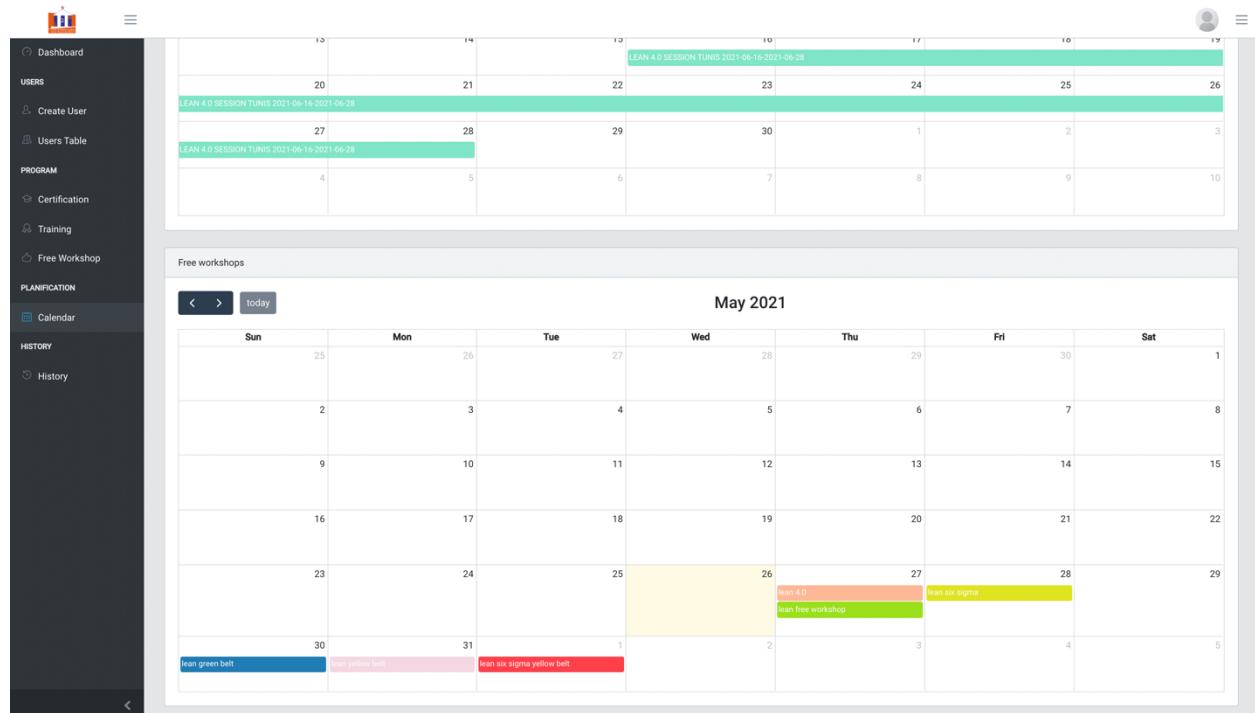


Figure 58 calendar view interface (2)

4. TESTS

To complete our solution at a very high score we had to do some tests to make sure that all functionalities are working fine and to tests our functionalities we are using “POSTMAN” to send request and do tests, in this sprint we had a lot of tests that we done but we going to show you just some of the important test that we had done.

4.1. ADDING TRAINING TEST

When we add a training to the system we must test if we add duplicated training so we should do this test with POSTMAN.

The screenshot shows a POST request to `localhost:3000/program/addTraining`. The request body contains JSON data representing a training with a module titled "test" and content "testt test". Another module with the same title and content is also present. The response status is 400 Bad Request, indicating that the training already exists. The response body is a JSON object with a single key "message": "Failed! training already existed".

```
localhost:3000/program/addTraining
POST localhost:3000/program/addTraining
Params Authorization Headers (9) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
1
2
3
4
5
6
7
8
9
10
11
12
Body Cookies Headers (8) Test Results
Pretty Raw Preview Visualize JSON
1
2
3
Status: 400 Bad Request Time: 287 ms Size: 322 B Save Response
Bootcamp Runner Trash
```

```
modules: [
  {
    "title": "test",
    "content": "testt test"
  },
  {
    "title": "test",
    "content": "testet tyysfcs"
  }
],
"level": "Green Belt"
```

```
{"message": "Failed! training already existed"}
```

Figure 59 adding training test

4.2. PLANNING SESSION TEST

When we plan session, we should make sure that the start date of the session doesn't come after it end date.

The screenshot shows the Postman application interface. At the top, the URL is `localhost:3000/planning/addSession`. Below it, a POST request is being made to the same URL. The 'Body' tab is selected, showing a JSON payload:

```

1
2   ...
3     "title": "Session 4",
4     "start_Date": "2012-05-30",
5     "end_Date": "2012-04-30",
6     "address": "Tunis",
7     "certification": "607d8a4d9d716004a66227cc"

```

Below the body, the status bar indicates `Status: 400 Bad Request`, `Time: 180 ms`, and `Size: 352 B`. The response body shows an error message:

```

1
2   "message": "Failed! end date of the session should be after the start date"
3

```

Figure 60 planning session test

4.3. PLANNING SESSION'S JOURNEY TEST

When we plan a journey inside a session, we should make sure that it doesn't interact with other journey and doesn't plan at the same day.

The screenshot shows a Postman interface for a POST request to `localhost:3000/planning/addTrainingDay`. The request body is JSON:

```
1 "date": "2012-06-27",
2 ... "session": "60a7fd33de0d7c6ebb261b9d",
3 ... "training": "607d8a4d9d16004a66227cc"
4
5
```

The response status is 400 Bad Request, with the message: "Failed! cant have two training in one day try to plan it in other day".

Figure 61 planning session's journey test

4.4. ADD EMPTY CERTIFICATION'S SESSION TEST

When we have a certification without trainings, we can't plan a session for it unless we add new trainings by editing the certification.

The screenshot shows a POST request to `localhost:3000/planning/addSession`. The request body is a JSON object:

```

1
2   "title" : "Session 4",
3   "start_Date": "2012-05-30",
4   "end_Date": "2012-08-30",
5   "address" : "Tunis",
6   "certification" : "6087435c5799171a6aaf47f0"
7

```

The response status is 400 Bad Request, with the message: "Failed! cant add session to a certification without trainings".

Figure 62 Add empty certification's session test

CONCLUSION

In this chapter, we presented the second sprint in release one. To do this, we went through analysis, design, and production. In the next chapter we enter the third and last sprint in release one.

CHAPTER 5 « STUDY AND REALIZATION OF SPRINT 3»

INTRODUCTION

In this chapter, we present the realization of the third and last sprint in release 1, by organizing the work on main phases which are the analysis, the realization, and the tests.

1. SPRINT BACKLOG

The sprint is the heart of Scrum. This is a block of time during which an increment of the product will be made. All sprints in a release have a constant duration and never overlap, that is, a sprint cannot start until the previous one is finished. Before embarking on a sprint, the Scrum team must define the goal of the latter, which must be a descriptive table that specifies the workload for each task in number of days.

ID	TASKS	ESTIMATION (days)
1	Add feed post	2
2	Delete feed post	1
3	Edit feed post	2
4	Reset password	3
5	Edit profile	2
6	Check history	2
7	Check dashboard stats	4

TABLE 16 Sprint 3 backlog

2. CONCEPTION

In this section we present the analysis phase that answers the question "what does the system". The answer to this question is reflected in the presentation of the use case diagram and the textual description of each.

2.1. USE CASE DIAGRAM:

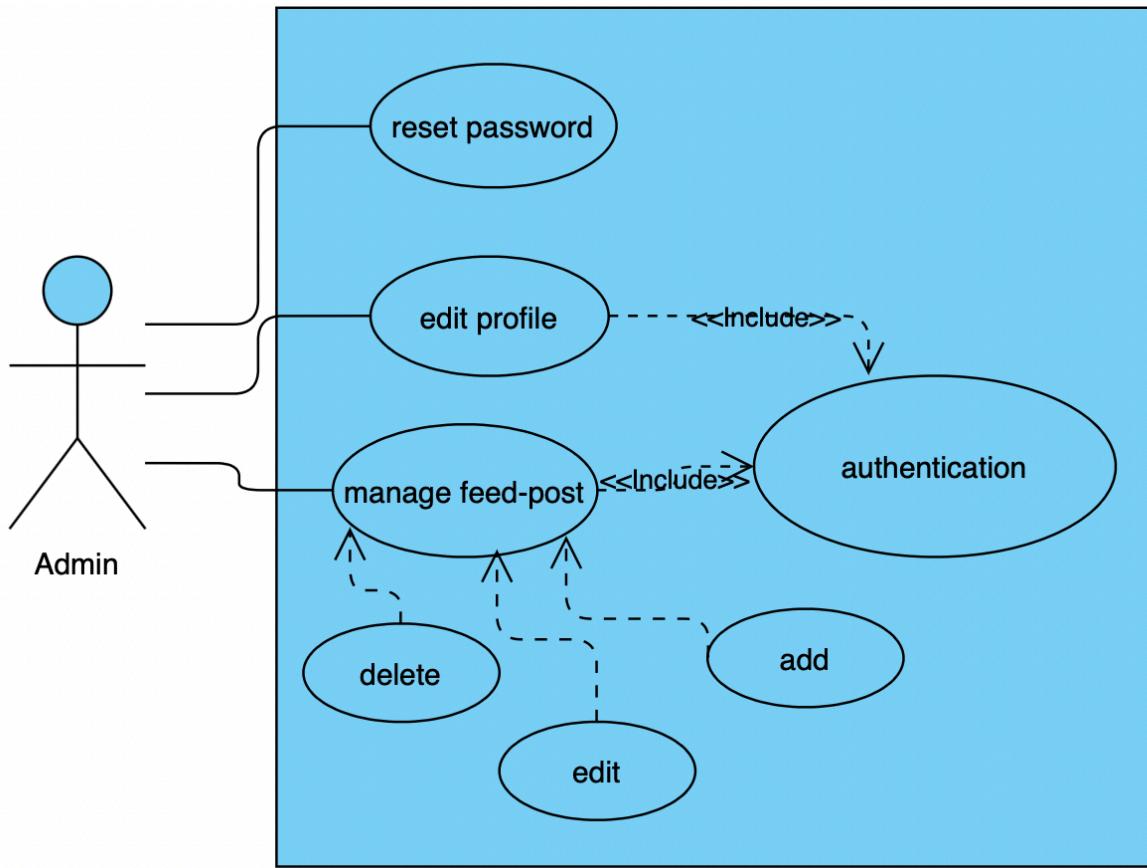


TABLE 17 SPRINT 3 USE CASE DIAGRAM

2.2. USE CASE DIAGRAM

- **Text description of use case “Add feed post”**

TITLE	Add feed post
ACTORS	Super admin and media manager
SUMMARY	In this case the super or media manager admin can add a new feed post to the front office website.
PRE-CONDITION	The actor must be authenticated to perform the task and got the authorization to the page
PRINCIPAL SCENARIO	<ol style="list-style-type: none">1. Actor access to feed post management page2. Actor enter feed post data3. The system verifies the data

	4. The system displays a success notification to the actor. 5. The feed post can now be displayed on the front office website
POST-CONDITION	Feed post added to feed posts table and can be updated or get deleted

TABLE 18 Description Text Add Feed Post

- **Text description of use case “update feed post”**

TITLE	Update feed post
ACTORS	Super admin and media manager
SUMMARY	In this case the super or media manager admin can update a new feed post.
PRE-CONDITION	The actor must be authenticated to perform the task and got the authorization to the page
PRINCIPAL SCENARIO	1. Actor access to feed post management page 2. Actor click on info page of a feed post inside the table data 3. Actor enter the feed post data to be updated 4. The system verifies the data and pass a successfully message 5. The feed post is now updated
POST-CONDITION	Feed post is updated

TABLE 19 Description Text Update Feed Post

- **Text description of use case “edit profile”**

TITLE	Edit profile
ACTORS	Super admin and admins
SUMMARY	1. In this case the super or an admin can update his profile.

PRE-CONDITION	2. The actor must be authenticated to perform the task.
PRINCIPAL SCENARIO	3. Actor access to edit profile page 4. Actor enter the data required to be added 5. Actor reenter his password or new one and confirm it 6. The system verifies the data and pass a successfully message 7. The profile is now updated
POST-CONDITION	The profile is updated

TABLE 20 Description Text Edit Profile

- **Text description of use case “reset password”**

TITLE	Reset password
ACTORS	Super admin and admins
SUMMARY	In this case the super or an admin can rest his password.
PRE-CONDITION	----
PRINCIPAL SCENARIO	1. Actor click on reset password button 2. Actor enter his email 3. System checks his email and send a token to the email. 4. Actor copy and paste token in the token input 5. System verifies the token 6. Actor enter new password 7. Actor password is updated
POST-CONDITION	The actor password is updated

TABLE 21 Description Text Reset Password

2.3. SEQUENCE DIAGRAM

Sequence diagrams are the graphical representation of interactions between actors and the system in chronological order in the UML formulation. We present thereafter the system sequence diagrams of the most important classified use case.

- “Adding feed post” sequence diagram

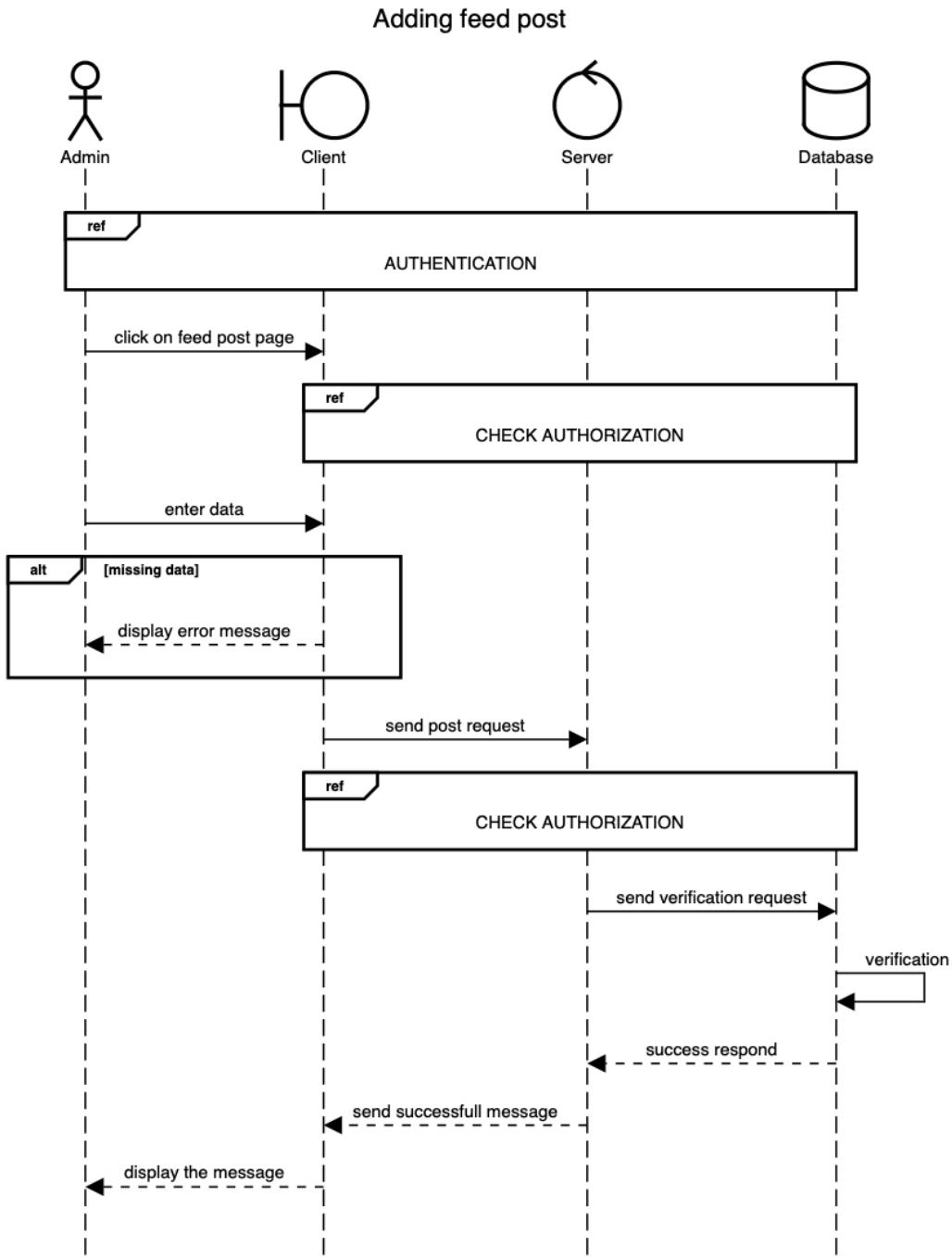


FIGURE 63 Adding Feed Post Sequence Diagram

- “Edit profile” sequence diagram

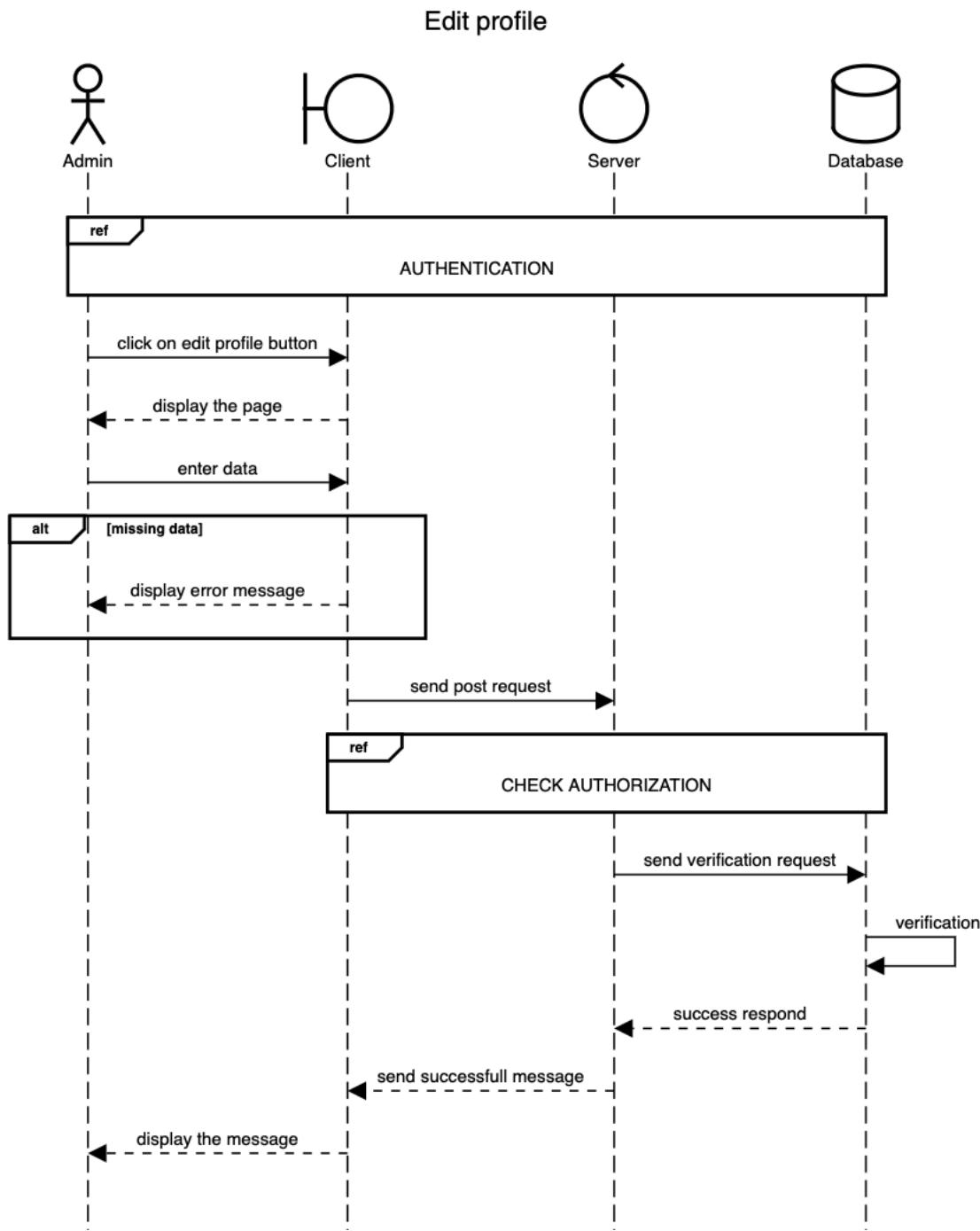


FIGURE 64 Edit Profile Sequence Diagram

- “Reset password” sequence diagram

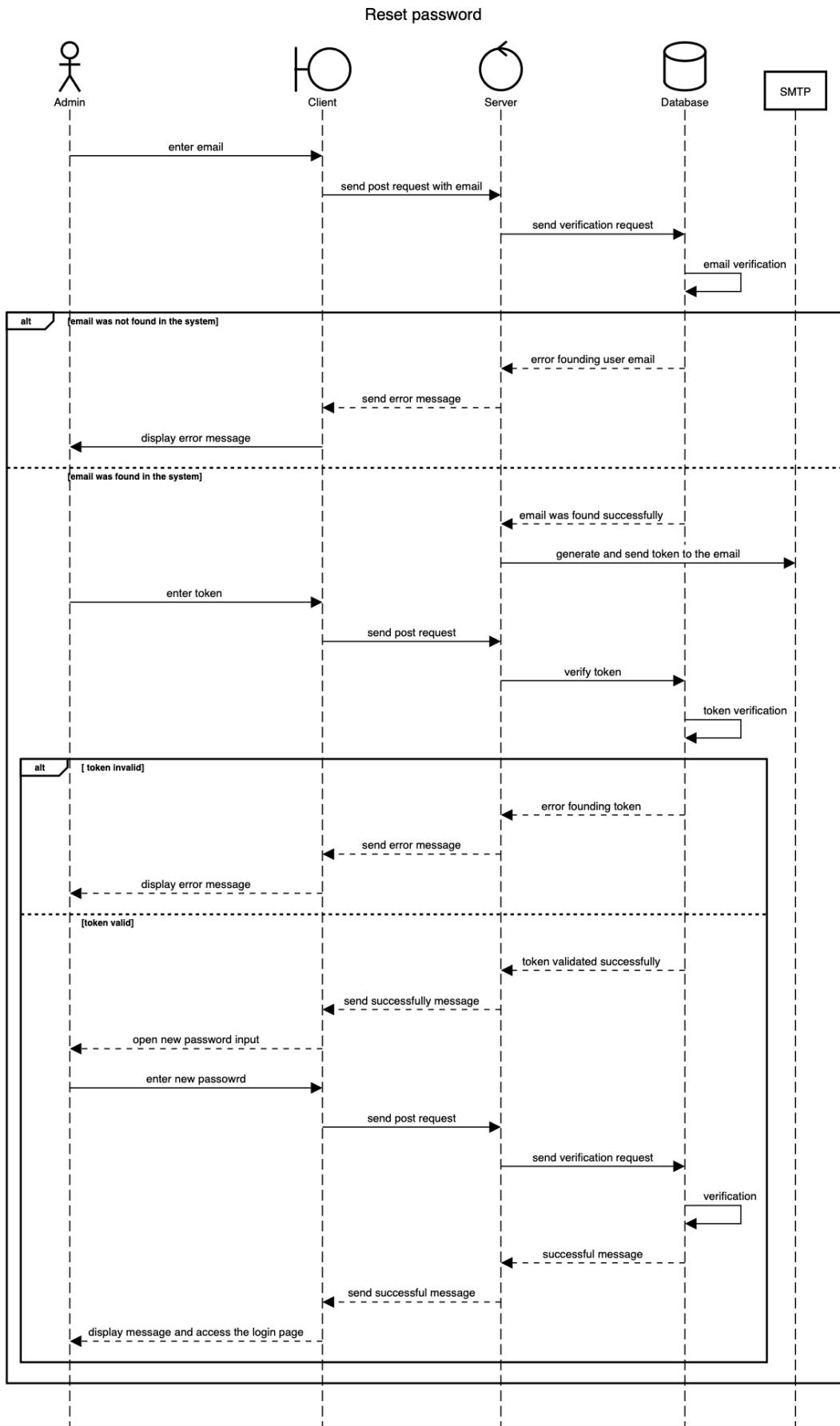


FIGURE 65 RESET PASSWORD SEQUENCE DIAGRAM

2.4. CLASS DIAGRAM

A class diagram allows us to give a general view on the application by the description of the intervening classes and the different relationships between them.

The following figure shows the class diagram for the third sprint:

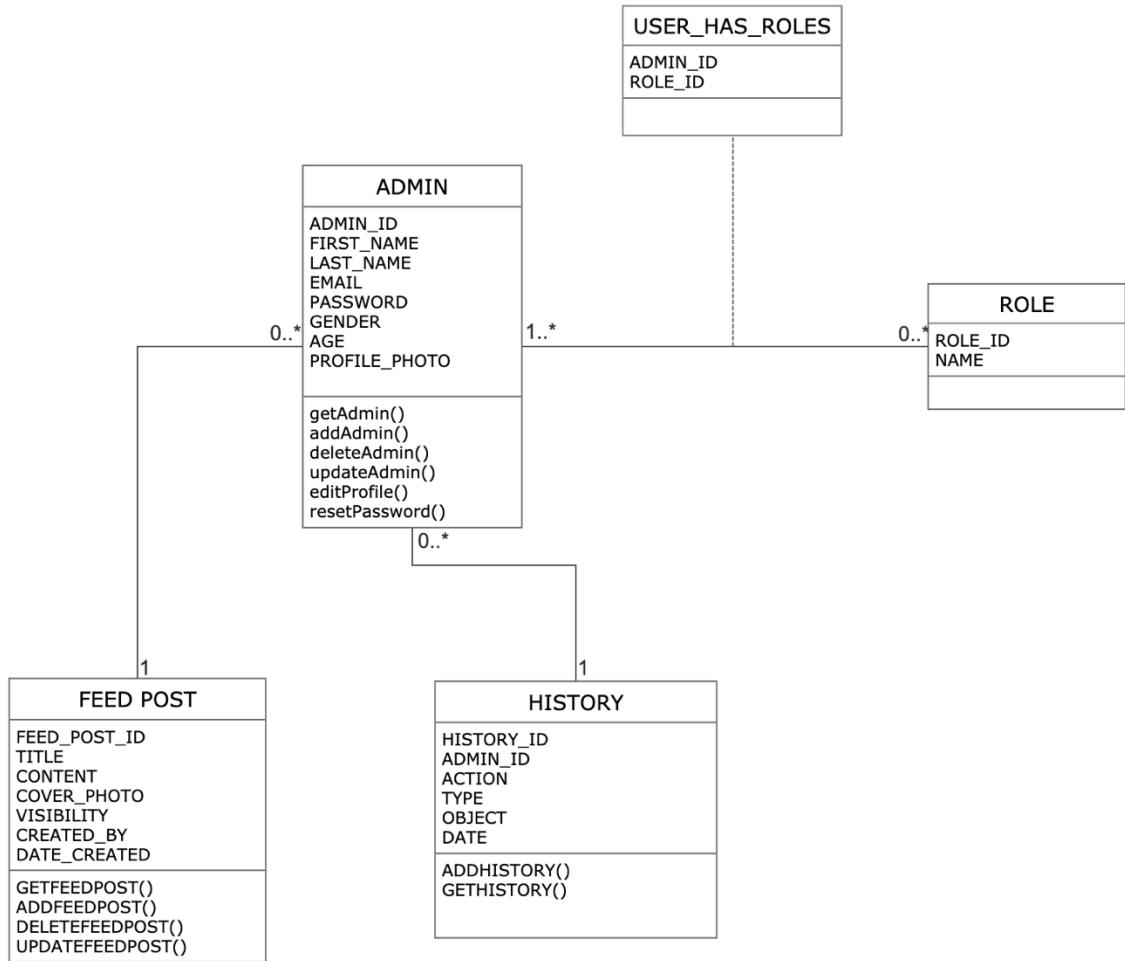


FIGURE 66 Sprint 3 Class Diagram

3. REALIZATION

This part is devoted to the exhibition of work completed through screenshots of different interfaces developed during this sprint.

3.1. ADD, DELETE FEED POST INTERFACE

This interface is only accessed by two actors the super admin and the media manager, this page as you see is divided into two section one is adding a feed post and this other is the feed post table. In the feed post table, we have two actions delete and view (edit feed post).

The screenshot shows a web-based application interface for managing feed posts. On the left is a dark sidebar menu with various options like Dashboard, Create User, Users Table, Certification, Training, Free Workshop, Calendar, Feed Post (which is selected), History, and History. The main content area has a header "Home / Media Management / Feed posts".

The top half of the main area is labeled "ADDING FEED POST BOX" and contains a form for creating a new post. It includes fields for "TITLE" and "CONTENT", a "Choose Cover Photo" button with a "No file chosen" message, and a large green "Publish" button.

The bottom half is labeled "VIEW AND DELETE ACTION" and shows a table of existing feed posts. The table has columns for "Title", "Content", "Visibility", "Info", and "Delete". The first four columns are standard text, while "Visibility" contains icons (green eye, red eye, blue info). The "Info" column for the third row is highlighted with a blue border. The "Delete" column for the fourth row is also highlighted with a blue border. At the bottom of the table are navigation buttons: "Prev", "1", "2", "Next", and "Visibility State".

Annotations with arrows point from the text labels to their corresponding parts in the interface:

- An arrow points from "ADDING FEED POST BOX" to the top form area.
- An arrow points from "VIEW AND DELETE ACTION" to the table area.
- An arrow points from "VISIBILITY STATE" to the "Visibility" column in the table.

FIGURE 67 Add, Delete Post Feed Interface

3.2. VIEW, EDIT FEED POST INTERFACE

When we click on the info button of a feed post in the feed post table you will be redirected to this page. We got two sections in this interface we have the full feed post content with the title and the date it was created alongside the full name of the admin who post it.

Post

World Environment Day FEED POST TITLE

Reviving And Protecting Our Ecosystems Is This Year's World Environment Day Theme And Marks The Beginning Of The United Nations Decade On Ecosystem Restoration. ISO Is Committed To Driving The Movement Towards A #Generationrestoration With New Standards And Initiatives Underway. Poverty, Climate Change And Extreme Weather Events Are Just Some Of The Havoc Caused By The Degradation Of Our Environment, And Now Is The Time To Boost Efforts To Reverse That Trend. Recognizing That Stimulating Biodiversity Is Vital To Ecosystem Regeneration, ISO's Newly Formed Technical Committee ISO/TC 331, Biodiversity, Is Currently Working On Standards That Can Help All Kinds Of Organizations And Governments Contribute To The Effort. The Scope Of Work Covers Standardized Terms And Definitions, Methodologies For Impact Analysis, Frameworks For Defining Strategies And Action Plans, Monitoring And Reporting Tools, And Guidelines On Specific Biodiversity Issues Such As Ecological Engineering And Nature-Based Solutions And Technologies. The International Union For Conservation Of Nature (IUCN) Is A Liaison Organization On ISO's Committee For Biodiversity And Has Developed Key Global Standards For Biodiversity Conservation, Including The IUCN Red List Categories And Criteria, The Global Standard For Identification Of Key Biodiversity Areas And The IUCN Nature-Based Solutions Standard. "We Are Delighted To Work With ISO To Develop International Standards For Biodiversity," Said Dr Thomas Brooks, IUCN Chief Scientist. "The Timing Is Crucial To Support The Post-2020 Global Biodiversity Framework." ISO Secretary-General Sergio Mujica Added That International Collaboration Is Essential. "The Over-Exploitation Of Land And Species Has Eroded Biodiversity, With Consequences Such As Natural Disasters And Unequal Distribution Of Food And Water Across The World." He Said. "Individual Efforts Can Help, But What Will Really Make The Difference Is Collaboration At An International Level. ISO Standards Can Be The Catalyst For That, Providing Agreed Ways Of Working That Organizations And Governments Everywhere Can Benefit From, To Help Move Biodiversity Issues Higher Up The Political Agenda." Restoring The Ecosystem Also Requires Significant Investment And It Is Estimated That Many Trillions Of Dollars Will Be Needed If We Are To Meet Our Net-Zero Carbon Goals. That Is Why ISO Has Also Been Working On A New Family Of Standards To Facilitate A Secure, Transparent And Effective Green Finance Industry. Recently Published, ISO 14097, Greenhouse Gas Management And Related Activities – Framework Including Principles And Requirements For Assessing And Reporting Investments And Financing Activities Related To Climate Change, Helps Financiers Assess And Report On Their Actions And See The Real Value Of Their Contribution To Climate Goals. The Standard Will Be Implemented By Others In The Series, Currently In Development, Including The ISO 14030 Series For The Environmental Performance Evaluation Of Green Debt Instruments, ISO 14100, Green Finance: Assessment Of Green Financial Projects, And ISO 14093, Mechanism For Financing Local Adaptation To Climate Change: Performance-Based Climate Resilience Grants. ISO Has Hundreds Of Standards That Contribute To Sustainable Development And Improving Our Impact On The Environment, Such As ISO 14001 On Environmental Management Systems. Use Of These Standards Will Enable Organizations To Directly Support All Of The United Nations' 17 Sustainable Development Goals (SDGs), Particularly SDG 13 (Climate Action), SDG 14 (Life Below Water) And SDG 15 (Life On Land).

DATE ADDED ← Friday 4 June 2021 20:25

ADMIN FULL NAME ← NEJMI SIDATY

FIGURE 68 Edit, View Feed Post Interface (1)

And the other section is for editing the feed post it's like an update box with the title and changing the visibility stat, the cover photo, or the content.

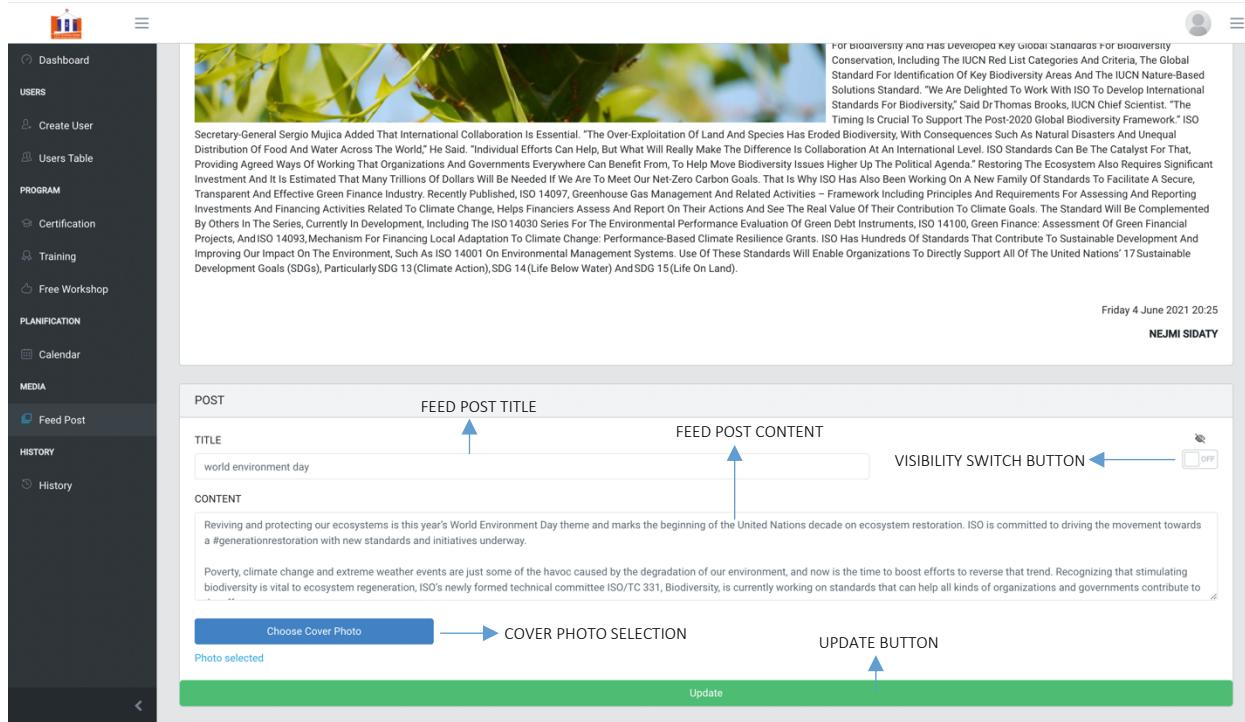


FIGURE 69 Edit, View Feed Post Interface (2)

3.3. VIEW, EDIT PROFILE INTERFACE

When the user clicks on the photo in the top right corner, he will have an edit profile button that will redirect him to this page in this interface the user can change his name, email, the profile photo and, he can change the password, but he should confirm it before updating the profile.

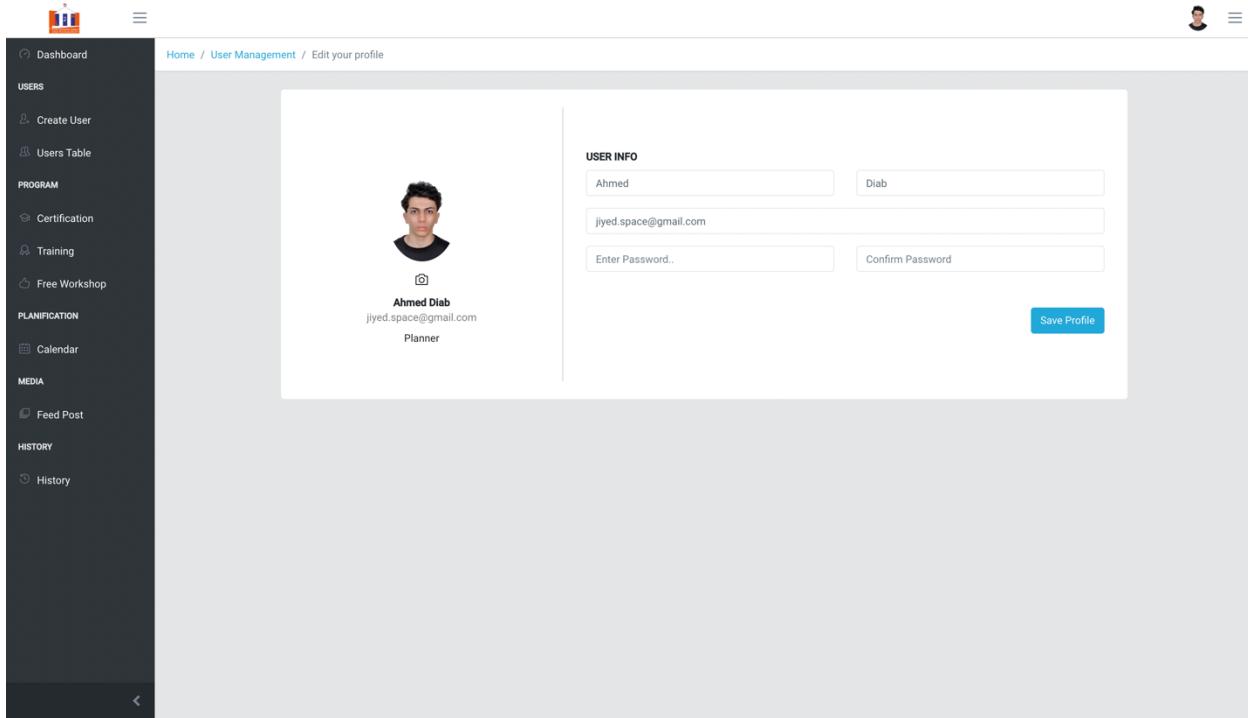


FIGURE 70 View, Edit Profile Interface

3.4. RESET PASSWORD INTERFACES

In this section we going to explain the reset password interfaces and the sequence behind it, first when we click on the forget password, the system will take you to this interface that contain an email input and the user should enter his email to retrieve his password.

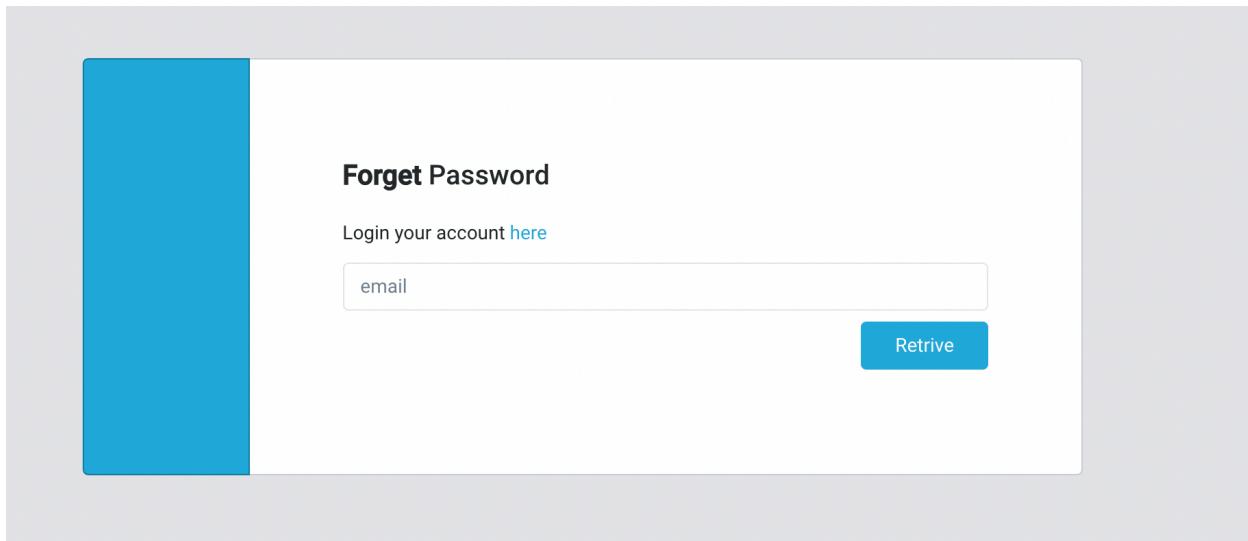


FIGURE 71 Reset Password Interface (1)

Then an email will be sent if the email is registered in the system, the email will contain the token to be pasted and checked by the system to validate it.

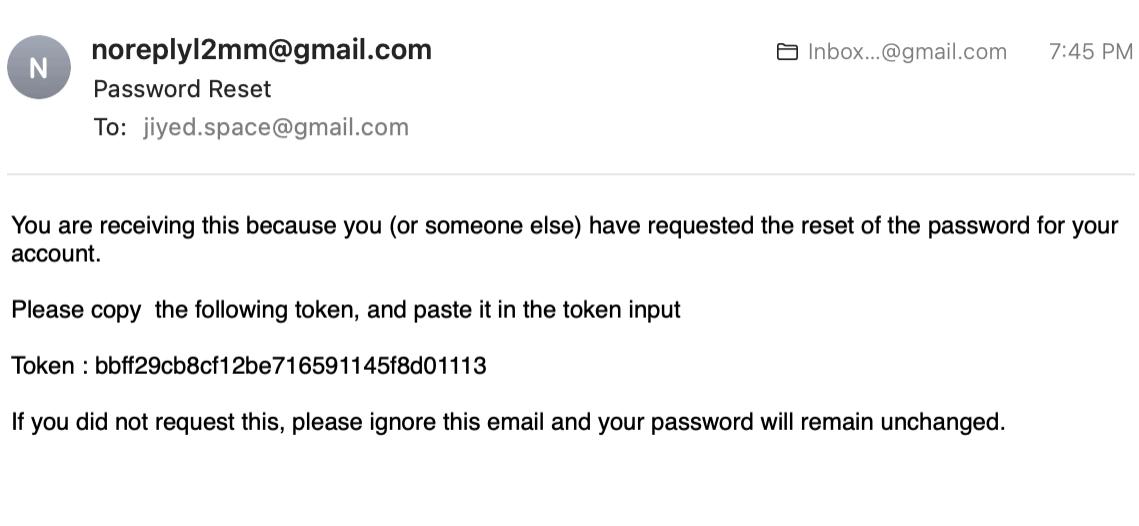


FIGURE 72 Reset Password Interface (2)

The token sent should be pasted in the input and should be validated if it's valid or not expired.

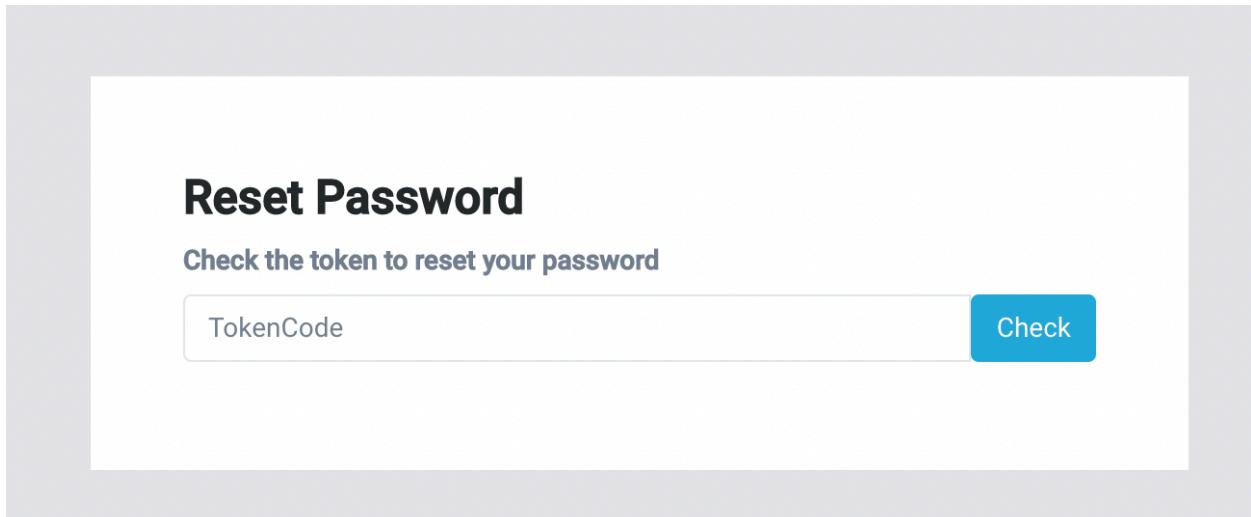


FIGURE 73 Reset Password Interface (3)

When the token is validated, the user will have the possibility to change his password by typing and clicking on the reset button.

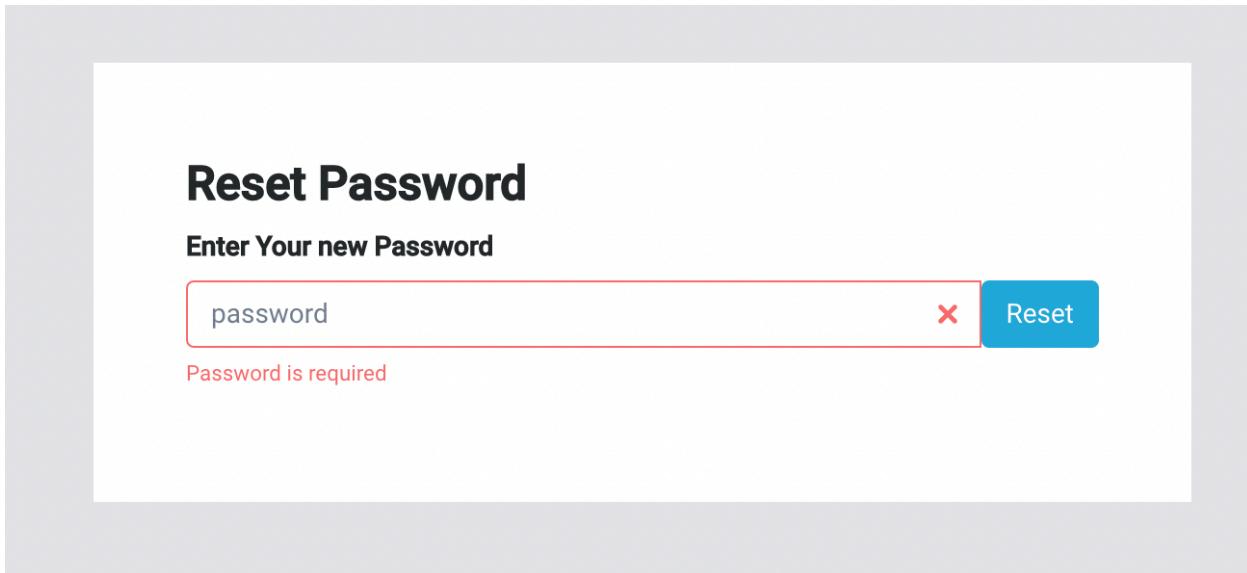


FIGURE 74 Reset Password Interfaces (4)

3.5. HISTORY INTERFACE

This page can only be accessed by the super admin to check all the interactions done by the admins and be able to filter the data table.

The screenshot shows the 'History' interface. On the left is a dark sidebar with navigation links: Dashboard, USERS (Create User, Users Table), PROGRAM (Certification, Training, Free Workshop), PLANIFICATION (Calendar), MEDIA (Feed Post), and HISTORY (History). The 'History' link is highlighted. The main area has a header 'Home / History / History Table' and a search bar with placeholder '(FILTER):(USER, ACTION, TYPE, OBJECT)'. Below is a table with columns: User, Action, Type, Object, and Date. The table contains eight rows of data. At the bottom are navigation links for 'Prev' and 'Next'.

User	Action	Type	Object	Date
Nejmi Sidaty (32) nejmisdaty@gmail.com	UPDATED	FEED POST	Covid-19 variant identified in India may increase risk of hospitalization, UK officials say	Friday 4 June 2021 23:32
Nejmi Sidaty (32) nejmisdaty@gmail.com	UPDATED	FEED POST	Covid-19 variant identified in India may increase risk of hospitalization, UK officials say	Friday 4 June 2021 23:27
Nejmi Sidaty (32) nejmisdaty@gmail.com	UPDATED	FEED POST	Covid-19 variant identified in India may increase risk of hospitalization, UK officials say	Friday 4 June 2021 23:27
Nejmi Sidaty (32) nejmisdaty@gmail.com	PUBLISH	FEED POST	Nigeria bans Twitter after company deletes President Buhari's tweet	Friday 4 June 2021 20:36
Nejmi Sidaty (32) nejmisdaty@gmail.com	PUBLISH	FEED POST	Covid-19 variant identified in India may increase risk of hospitalization, UK officials say	Friday 4 June 2021 20:36
John doe (24) john.doe@gmail.com	PUBLISH	FEED POST	Pakistani court overturns death row Christian couple's blasphemy conviction	Friday 4 June 2021 20:36
John doe (24) john.doe@gmail.com	PUBLISH	FEED POST	Hong Kong's Tiananmen vigil always set the city apart from China. Those days may now be over	Friday 4 June 2021 20:36
Ahmed Diab (24) jyed.space@gmail.com	ADDED	SESSION	lean 4.0 session Tunis 2021-06-23-2021-06-30	Friday 4 June 2021 20:36

FIGURE 75 History Interface

3.6. STATS INTERFACES

This is the home page of our web application this home page is accessed by all the admins, and we have on top some specific pages that only accessed if you got the authorization, the admins could check the news and reload for new ones, also the have a widget for the temperature by the location of the user, also we have some stats on the office.

The screenshot shows a web-based dashboard interface. On the left is a dark sidebar menu with the following items:

- Dashboard
- USERS**
 - Create User
 - Users Table
- PROGRAM**
 - Certification
 - Training
 - Free Workshop
- PLANIFICATION**
 - Calendar
- MEDIA**
 - Feed Post
- HISTORY**
 - History

The main content area has a header "Home / Dashboard". It features two cards: "User Stats" (blue) and "Program Stats" (red). Below these is a news section titled "HONG KONG'S TIANANMEN VIGIL ALWAYS SET THE CITY APART FROM CHINA. THOSE DAYS MAY NOW BE OVER" with a small image of a protest and a detailed text block. To the right of the news is a "REFRESH BUTTON FOR NEW NEWS" icon. Further down are two boxes showing "MAX CERTIFICATION PRICE" (\$4568) and "MIN CERTIFICATION PRICE" (\$320), each with a bar chart. To the right of these is a green box labeled "Hammam Sousse" with a temperature of "74.48°F" and a "TEMPERATURE WIDGET" icon. At the bottom are sections for "USER STATS" (a bar chart showing 4 users for Admin manager, 3 for Planner, and 3 for Media manager) and "CERTIFICATIONS" (listing LEAN SIX SIGMA YELLOW BELT, LEAN SUPPLY CHAIN BLACK BELT, and LEAN 4.0 GREEN BELT with their respective training counts).

FIGURE 76 Stats Interface (1)



FIGURE 77 Stats Interface (2)

When we click on the user stats, we should be a super admin or an admin manager to access that page, in this page we got some stats on the admins that are using this web application. And we have also last interaction done by an admin.

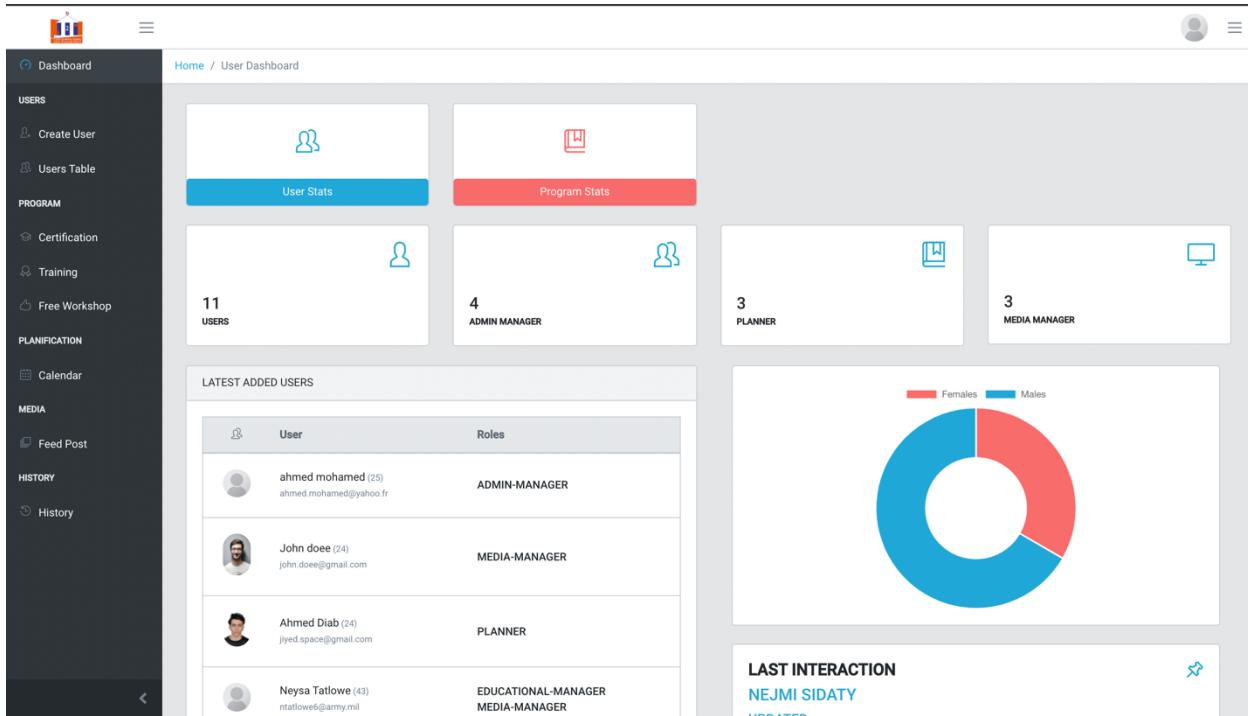


FIGURE 78 Stats Interface (3)

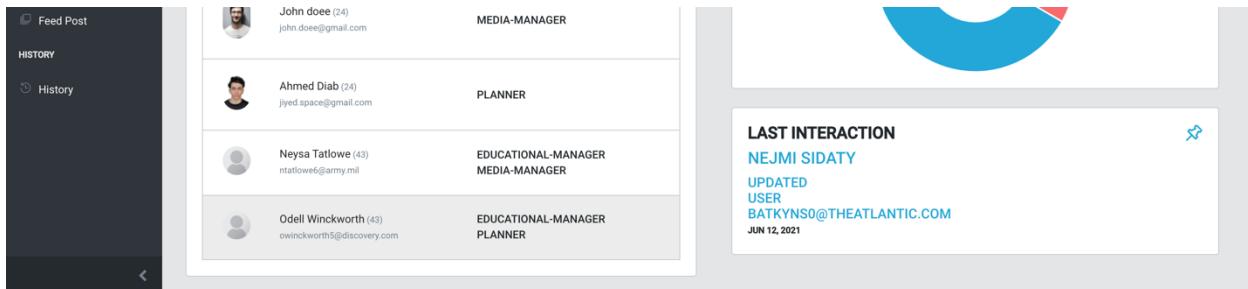


FIGURE 79 Stats Interface (4)

On the right side when we click on the program stats, we should be a super admin or a planner to access that page, in this page we got some overall stats on the program created in the web application if it was a certification, a module or even a planned session.

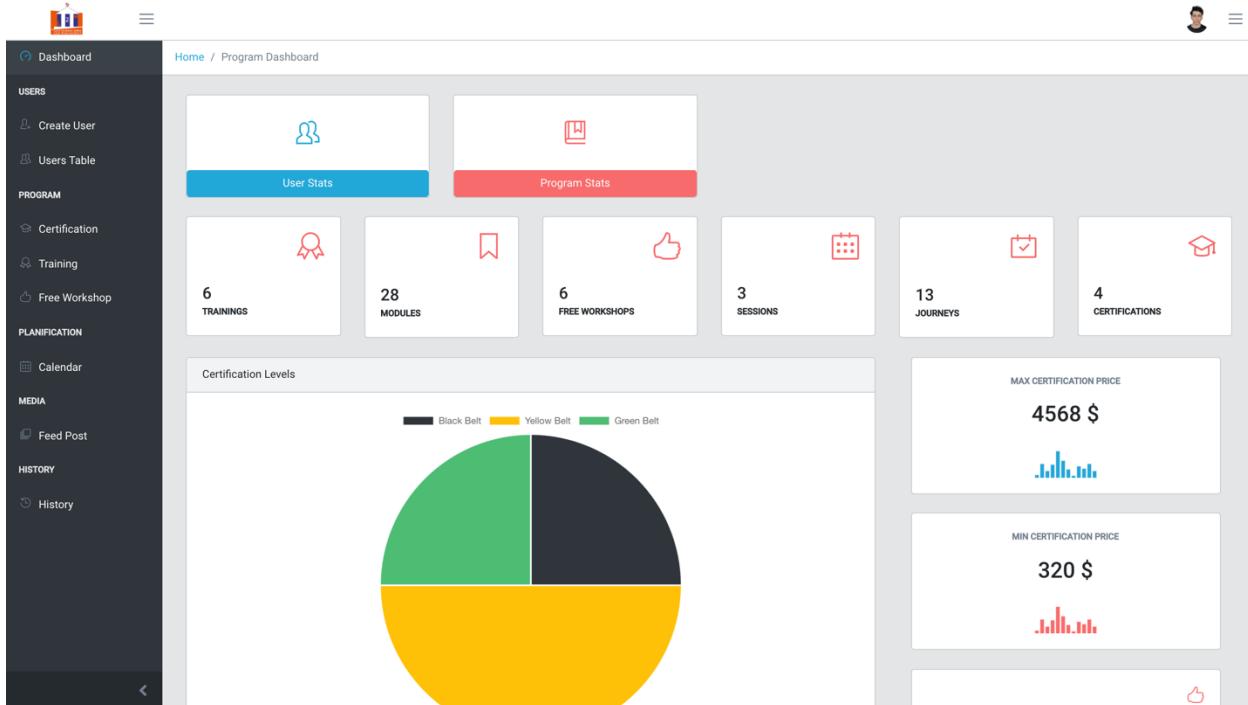


FIGURE 80 Stats Interface (5)

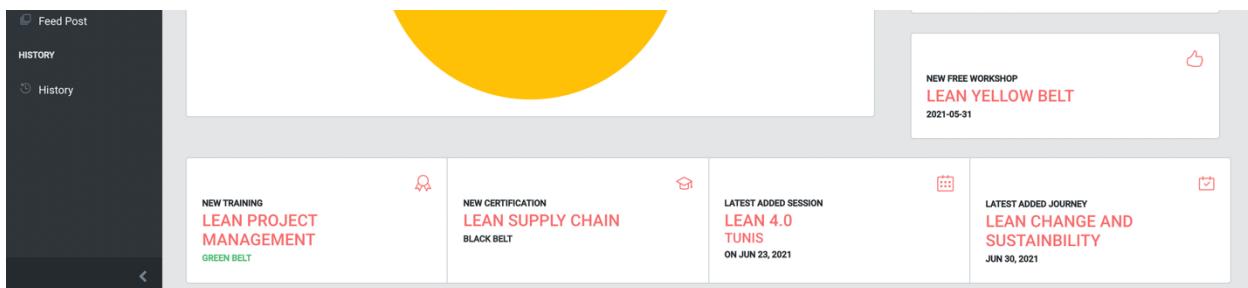


FIGURE 81 Stats Interface (6)

4. TESTS

To complete our solution at a very high score we had to do some tests to make sure that all functionalities are working fine and to test our functionalities we are using “POSTMAN” to send requests and do tests, in this sprint we had a lot of tests that we did but we are going to show you just some of the important tests that we had done.

4.1. RESET PASSWORD TESTS

We are going to test our functionalities by sending incorrect data like for email verification we are going to give it an email that is not registered in the system.

The screenshot shows a POST request to `localhost:3000/auth/req-reset-password`. The request body is JSON with a single key `"email"` set to `"nejmi@gmail.com"`. The response status is 409 Conflict, with the message `"Email does not exist"`.

```

1 {
2   "email": "nejmi@gmail.com"
3 }

```

```

1 {
2   "message": "Email does not exist"
3 }

```

FIGURE 82 Reset Password Test (1)

And when we send a token, we should verify that the token send with data is valid and not expired

The screenshot shows a POST request to `localhost:3000/auth/token-verification`. The request body is JSON with a single key `"resettoken"` set to `"7fa25021d8179b7f3045038897be6277"`. The response status is 409 Conflict, with the message `"Invalid Token"`.

```

1 {
2   "resettoken": "7fa25021d8179b7f3045038897be6277"
3 }

```

```

1 {
2   "message": "Invalid Token"
3 }

```

FIGURE 83 Reset Password Test (2)

CONCLUSION

In this chapter, we presented the third and last sprint in our release. To do this, we went through analysis, design, and production

GENERAL CONCLUSION

The work carried out during this end of study project focused on the design and implementation of a web application to allow all members of the Lmm office team to manage and hold control of their website including different type of management from admin management to planning sessions. The implementation of our project took place in several stages. A first step is to understand the general context of the project. A second with the aim of analyzing and bringing out the real needs of the user. This step is one of the key steps in the success of any project. It is for this reason that we have devoted an important period in the analysis and the specification of needs. We opted for Scrum as development process and for UML as a modelling language. For the realization, we used a set of tools.

Although we encountered some difficulties at the beginning of the project, it was a good opportunity to get out of the theoretical framework and to apply the knowledge acquired during university studies in the world of work. In addition, this project allowed us to put into practice the knowledge acquired regarding the life cycle of a computer project. This achievement was also the opportunity to integrate the professional world and learn several skills and habits social issues such as group work and information gathering, this to extract the needs of system actors to be implemented. This project is a nucleus on which several perspectives can be extended to enrich and develop it. Indeed, the work that we have realized is modular and scalable and need more developing and time to make it close to perfect.

BIBLIOGRAPHY

- [1] [Online]. Available: <https://en.wikipedia.org/wiki/Node.js>.
- [2] "Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/MongoDB>.
- [3] "Angular," [Online]. Available: <https://angular.io/guide/architecture>.
- [4] "Visual Studio Code," [Online]. Available: <https://code.visualstudio.com/docs>.
- [5] "IntelliJ IDEA," [Online]. Available: https://en.wikipedia.org/wiki/IntelliJ_IDEA.
- [6] "Axelerant," [Online]. Available: <https://www.axelerant.com/resources/team-blog/api-testing-with-postman>.
- [7] "Umass," [Online]. Available: <https://blogs.umass.edu/onlinetools/community-centered-tools/trello/>.