SESAME TECHNOLOGY

المدرسة العليا الخاصة للعلوم التطبيقية و إدارة الأعمال
ECOLE SUPÉRIEURE PRIVÉE DES SCIENCES APPLIQUÉES ET DE MANAGEMENT

SUMMER INTERNSHIP REPORT

COMPUTER SCIENCE ENGINEERING CYCLE

Development of an E-Commerce Platform for IPTV Subscription Management

Prepared by
Sameh Abdelmajid Kouki

Host Company

K-technologie

K-technologie

Company Supervisor
Mohamed Karim Lahbib

Academic Year
2023-2024

Table of Contents

Figures List

Table of contents

Tables List

General Introduction

The internship period serves as a pivotal phase in bridging the gap between academic knowledge and practical experience, providing an opportunity to apply theoretical concepts in a real-world context. Over the course of one month, I had the privilege of undertaking an internship at K-Technologie, a dynamic and innovative company specializing in technological solutions.

The primary objective of this internship was to gain hands-on experience in the development of an **e-commerce platform for selling IPTV subscriptions**, a project that aligned perfectly with my academic background and career aspirations. This project not only allowed me to refine my technical skills but also provided valuable insights into the workings of the e-commerce domain, client requirements, and the collaborative processes essential for successful project execution.

K-Technologie, known for its forward-thinking approach and emphasis on cutting-edge technologies, offered an ideal environment for learning and professional growth. With guidance from experienced professionals and exposure to real-world challenges, this internship has been an enriching journey, equipping me with essential skills and a deeper understanding of industry practices.

This report documents my internship experience, focusing on the objectives, methodologies, challenges, outcomes, and the key lessons learned during this period.

Chapter 1 : Project Presentation

Introduction

This opening chapter provides an overview of the chosen subject area and its context within the broader field. I will delve into the project's specific goals and objectives, outlining the key challenges identified and the solutions proposed to address them. I will conclude with a discussion of the methodologies employed for project management and design.

1. Presentation of the host company

K-Technologie is a company specializing in the trade and maintenance of computer hardware, as well as in the development of software and computer applications.

K Technologie was created on May 20, 2001 by Mr. Mohamed Karim Lahbib. Their activities are diverse: sale of computer hardware and accessories, maintenance and repair of computer hardware and electronic parts, software development, creation of WEB sites, assistance of companies.

The internship was carried out by specifying its vocation of the development of a modern, efficient website, and in exact period.

Its address is Av Habib Bourguiba El alia, Bizerte 7016.

Figure 1:K-technologie Logo

2. Project context

This project marks the culmination of a focused effort to design and develop a scalable, user-friendly e-commerce platform tailored to selling IPTV subscriptions. In an era where streaming services dominate the entertainment industry, IPTV subscriptions have emerged as a sought-after product, demanding efficient and accessible distribution platforms. The project was initiated to address the gap in streamlined, customer-centric solutions for purchasing and managing IPTV services.

K-Technologie identified the need for a specialized e-commerce platform to cater to their growing customer base and improve the user experience. The project aimed to integrate modern web technologies, optimize the purchase process, and ensure secure transactions while adhering to industry best practices. As an intern, my role was instrumental in contributing to the development and implementation phases of the platform.

3. Description of the existing

The practice of managing and supervising IPTV subscriptions before this project was largely manual or reliant on outdated systems. Existing solutions often lacked user-friendly interfaces, seamless subscription management features, and the ability to handle secure online transactions.

Customers had to rely on fragmented processes involving direct communication with service providers, offline payment methods, or generic e-commerce platforms not specifically designed for IPTV services. These approaches were not only inefficient but also prone to errors, leading to delays and customer dissatisfaction. Moreover, these systems often failed to provide personalized experiences or streamline subscription renewals, making them less appealing to modern customers accustomed to convenience and automation.

4. Criticism of the existing

The existing system, prior to the development of the new platform, suffered from several shortcomings:

Inefficient Subscription Management: The lack of automation and centralized control made it difficult for providers to manage large-scale subscription data and for customers to track their plans.

Poor User Experience: Existing platforms were not optimized for user-friendliness, resulting in complicated navigation and inefficient purchase workflows.

Limited Payment Options: Many of the earlier methods lacked integration with modern payment gateways, reducing accessibility for customers.

Security Concerns: With no robust encryption or compliance measures, the old systems posed potential risks to customer data and payment information.

Lack of Scalability: The systems were not designed to accommodate growing demand or adapt to changing technology trends, limiting their long-term viability.

5. Suggested Solution

To address the challenges and requirements of the project, the following solution was proposed and implemented:

1. **Authentication System**
   - Implemented a secure authentication system to manage Admin access.
   - Incorporated password encryption to safeguard user credentials.
2. **Subscription Management**
   - Developed a module for Admins to add, update, and delete subscriptions.
3. **Client Purchase Process**
   - Streamlined the subscription purchasing process by enabling Clients to buy directly without requiring account creation.
   - Collected Client information (e.g., name, email, and payment details) at the time of purchase.
4. **Communication Module**
   - Enabled Clients to send inquiries or feedback emails directly to Admins.

   - Simplified the process by not requiring Clients to log in for this functionality.

5. **Statistics and Reporting**
   - Implemented a dashboard for Admins to consult key statistics, such as the number of subscriptions sold, active Clients, and revenue trends.
   - Included visual charts and graphs for better data representation.
6. **Database Design**
   - Designed a normalized database structure to manage Admins, Clients, subscriptions, and orders efficiently.
   - Ensured data consistency and optimized queries for better performance.
7. **User-Friendly Interface**
   - Developed an intuitive user interface using modern frameworks (e.g., Angular and Tailwind CSS).
   - Focused on responsiveness and accessibility to ensure usability across different devices.

6. Adopted methodology and formalism.

6.1. Conception Methodology (UML)

In my project, i have decided to use the Unified Modeling Language (UML) as a conceptual approach. UML is a standardized modeling language that allows developers to specify, visualize, construct, and document various aspects of a software system. By using UML, developers can ensure that their software artifacts are scalable, secure, and reliable when executed. UML plays a crucial role in object-oriented software development, and it uses graphical notation to create visual models of software systems.

Figure 2: UML logo

6.2. Work Methodology

To ensure the best work quality and efficiency, it was necessary to choose a Work Methodology that can manage all the project requirements and guarantee stability and flexibility with all the clients' changes.

6.3.Comparison of classic and agile approaches

Characteristics	Agile approach	Traditional approach

Development model	Flexible	Fixed
Testing	Every Iteration	Once
Changes	Easy to integrate new changes	Resistance to changes, it can cost much time, money
Life cycle	Linear	Iterative
Team	Autonomous empowered team	The team is led by a project manager
Success Measurements	Client Satisfaction	Respect budget and expenses
Escalation management	When problems occur, the entire team works together to resolve it	Escalation to managers when problem arise
Risk management	Risk management integrated into the overall process	Strict and rigorous risk management process

Tableau 1: Comparison of classic and agile approaches

A. Agile method: SCRUM

As it’s obvious the agile approach is way more efficient than the classic approach, that’s why i choose to work with SCRUM which is one of the most common and famous agile methods.

Scrum is a framework utilizing an agile mindset for developing, delivering, and sustaining complex products, with an initial emphasis on software development, although it has been used in other fields including research, sales, marketing and advanced technologies.

It is designed for teams of ten or fewer members, who break their work into goals that can be completed within time-boxed iterations, called sprints, no longer than one month and most commonly two weeks.

The Scrum Team assesses progress in time-boxed daily meetings of 15 minutes or less, called daily scrums. At the end of the sprint, the team holds two further meetings :

The sprint review which demonstrates the work done to stakeholders to elicit feedback, and sprint retrospective which enables the team to reflect and improve.

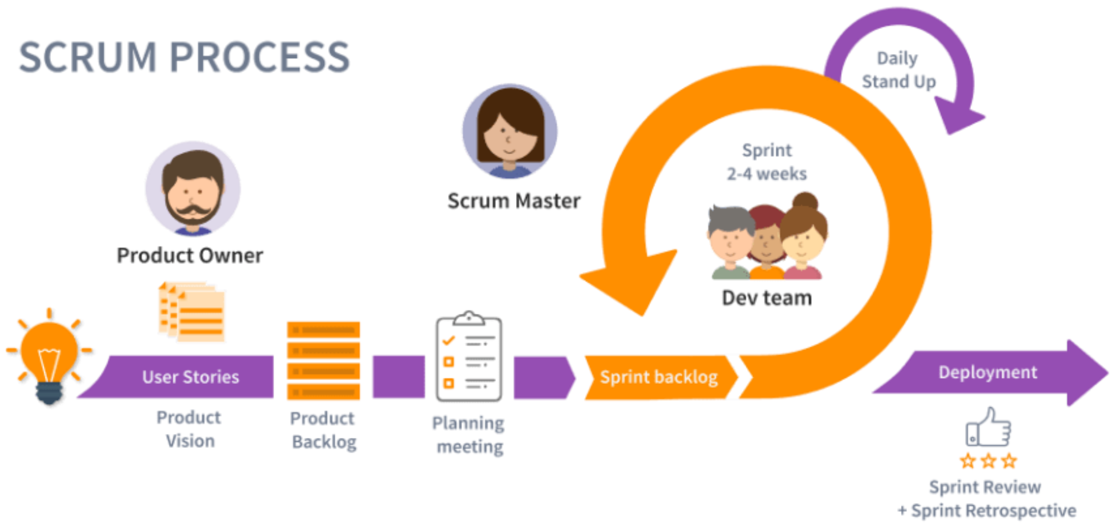


Figure 3: Scrum process.

Conclusion

In this chapter, i get an overview of the host company and the project context, besides presenting the remaining problems and the suggested solutions for that, to finish by describing the different methodologies adopted by this project.

The following chapter will be devoted to the preliminary study

Chapter 2: Preliminary study

Introduction

In this chapter i are going to introduce the different specification of my project, starting by describing the actors and their roles, moving on to the functional and non-functional needs of the application, which lead us directly to the project management method "SCRUM", where i will have an overview about the scrum team in addition to all the user stories, the product backlog, the sprint planning, and the global use case diagram.

In another part, i will represent the architectural conception for both logical and application architecture, to end up covering the work environment and the used technologies.

1. Needs specification

1.1. Actors Identification

An actor represents an external entity that can interact with the system, depending on his role, in order to get a specific service.

In this application, i have 2 actors:

Actor	Roles
Admin	This actor is capable to use the
Client	This actor is capable to use the

Tableau 2: Actors Identification

1.2. Functional needs

In order to build the **application**, the application must have the following functionalities:

Functionality	Description
Admin	**• Add subscriptions** **• Login** **• Manage subscriptions purchased by clients** **• Manage subscriptions** **• Consult statistics**

Client	• **Manage cart** • **Purchase subscriptions** • **Send emails**

Tableau 3: Functional needs

1.3. Non-Functional needs

The non-functional requirements are the restrictions that the system is committed to in order for it to be realized and perform properly, for example:

- **Security:** The application must ensure the confidentiality of the stored data, for that authentication is required in addition to restricting access to users depending on their roles and permissions.
- **Reliability:** The application should perform perfectly without any bugs or failure.
- **Ergonomics:** The front-end side should be user-friendly and respect all the UX principles, in order to facilitate application usage.
- **Performance:** The code must be highly optimized and organized in such a way that boosts the loading speed to give a good impression to users.
- **Extensibility:** One of the best programming practices is to write flexible code that can be easily changed (add new features or update existing functionality).

2. Project Management with Scrum

2.1. User Stories

A. Definition

A user story is an informal, simple phrase that describes an application feature, written from the end-user perspective. Its goal is to represent how this functionality will benefit the customer.

A user story consists of three main elements:

As a [Who], I want to [What] so that I can [Why]

- **Who**: Refer to a user with a unique role who wants to consume an application functionality (Exp: Admin, Manager, etc).
- **What**: Describe the requested functionality from the user.
- **Why**: Describe the user goal in which he requested this functionality.

2.2.Scrum team

❖ **Product Owner:**

He represents the customer requirements to the scrum team and he is responsible for following up during the project progress to ensure the efficiency of the final product since he knows the business domain and context.

❖ **Scram Master:**

He is the scrum expert who's charged to help the scrum team to stay committed to their duties and tasks by respecting the scrum roles. He can be considered as a coach to motivate and support, but not to force his colleagues.

❖ **Development Team:**

It's an auto-organized team composed of 3-9 members who don't occupy a specific role, and they can adapt according to the work situation. Their mission is to transform the product owner's needs into functional and useable features.

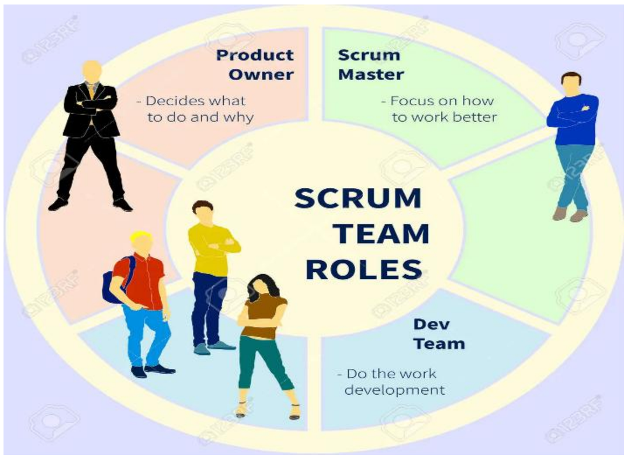

Product Owner
- Decides what to do and why
Scrum Master
- Focus on how to work better
SCRUM TEAM ROLES
Dev Team
- Do the work development

Figure 4: Scrum Team

Product backlog

The product backlog is a prioritized list of functionalities (user stories) that need to be done by the development team, which is considered as the primary source of information for all the team since it describes the project specifications.

For this project, i defined the product backlog with those specifications:

❖ **ID**: Unique number for each theme.

❖ **Theme**: Collection of user stories that have the same context.

❖ **User Stories**: Describe the requested functionality from a user.

❖ **Priority**: Depends on the client's needs, each functionality is rated with priority specification so the scrum team can manage their work.

For that, i can implement the **MoSCoW** method that signifies to:

- **M (Must Have):** functionalities that need to be done in which the project can't be realized without them (vital).
- **S (Should Have):** functionalities less important than "must-have" and it's unvital.
- **C (Could Have):** functionalities desirable to have.
- **W (Won't Have):** unimportant functionality, it can be done later.

❖ **Complexity**: it has three levels; low, medium, high.

ID	Theme	User Stories	Priority	Complexity
1	**Admin**	Admin login	Must	High
		Admin can manage subscriptions	Must	High
		Admin can get subscriptions that the client bought	Must	High
		Admin can consult statistics	Could	Medium
2	**Client**	Client can manage cart	Must	High
		Client can buy subscriptions	Must	Medium
		Client can send emails	Must	Medium

Tableau 4: Product backlog

2.3. Sprint planning

Release	Sprints	Sprint Name	Period

Release	**Sprint 1**	**Core Functionality (Admin and Client Basic Features)**	**30jr**
	Sprint 2	**Enhancements and Admin Features**	

Tableau 5:Sprints planning

3. Global use case diagram

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors).

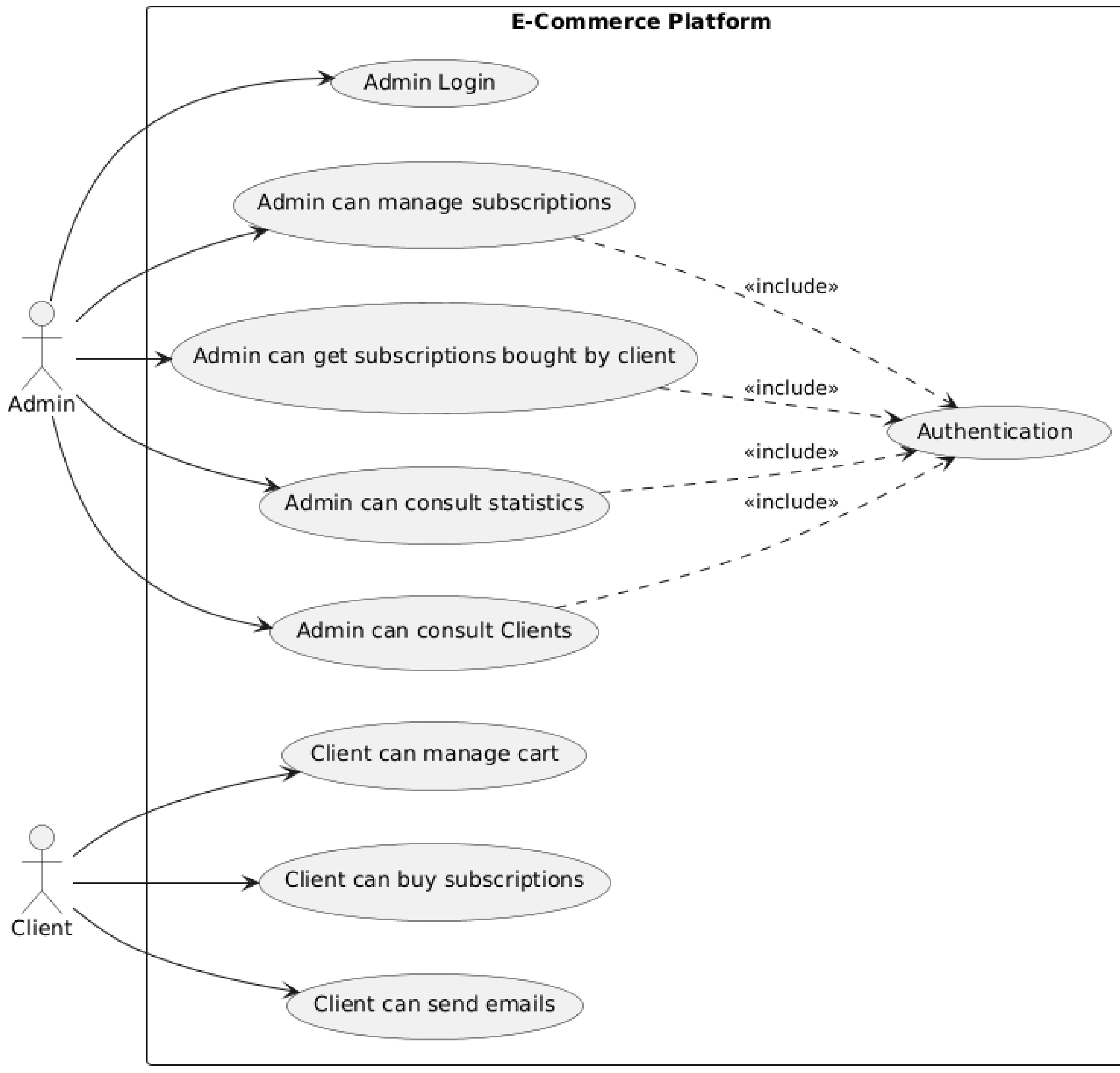

E-Commerce Platform
Admin Login
Admin can manage subscriptions
Admin can get subscriptions bought by client
Admin can consult statistics
Admin can consult Clients
Client can manage cart
Client can buy subscriptions
Client can send emails
Authentication
«include»
«include»
«include»
«include»
Admin
Client

Figure 5: Global use case diagram

4. Architectural conception

4.1. Logical architecture

The architecture that has been adapted for the realization of my solution is a 3-tier type. It is a logical architecture model, which consists of a separation into 3 software layers within the same system.

- **The data presentation layer**:

Corresponds to the display, restitution on the client's workstation and the dialogue with the user.

- **The business data processing layer:**

Corresponds to the implementation of all the business rules and application logic..

- **The data access layer:**

Corresponds to the data that is intended to be kept over time..

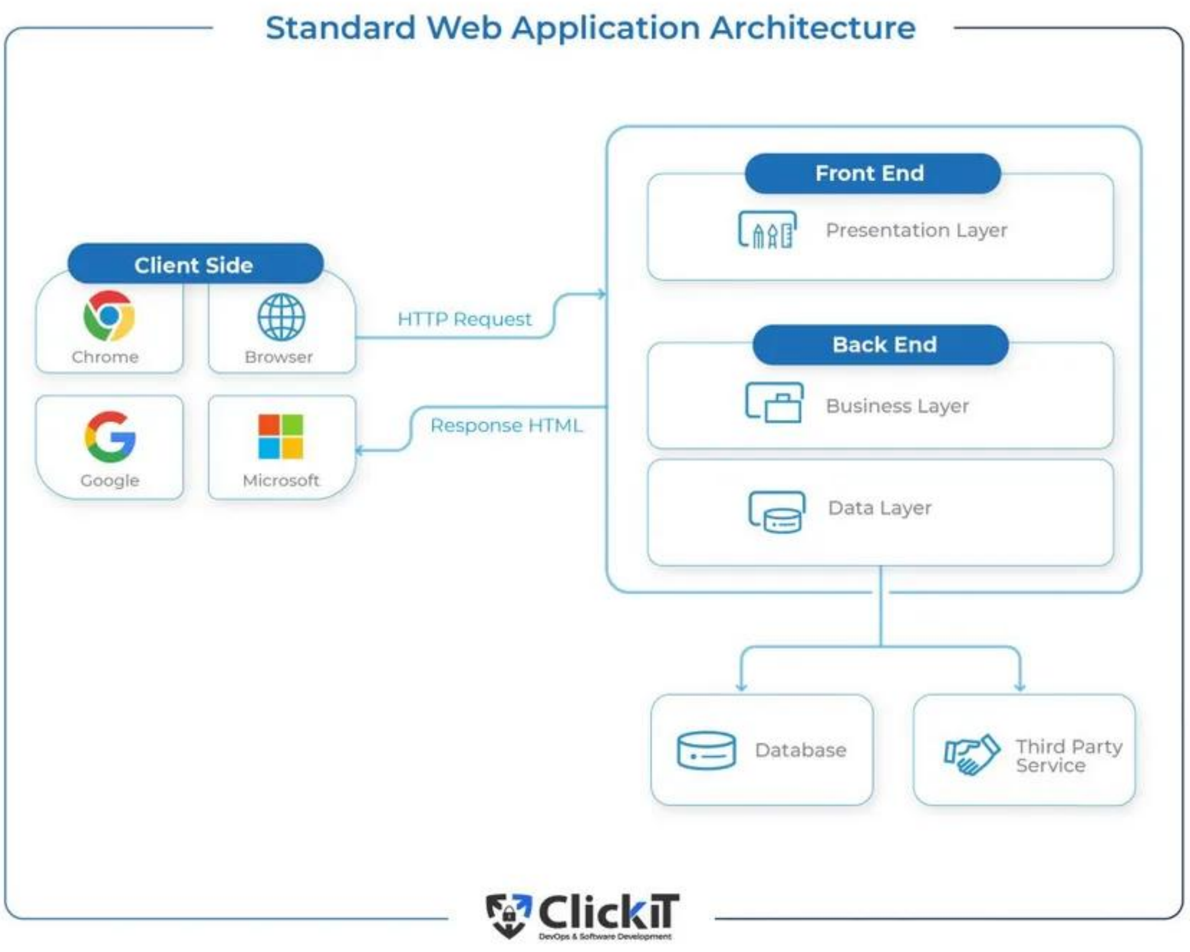


Figure 6: architecture

4.2. Application architecture

The application architecture is one of the most important conception steps since it defines the interaction between the different tiers.

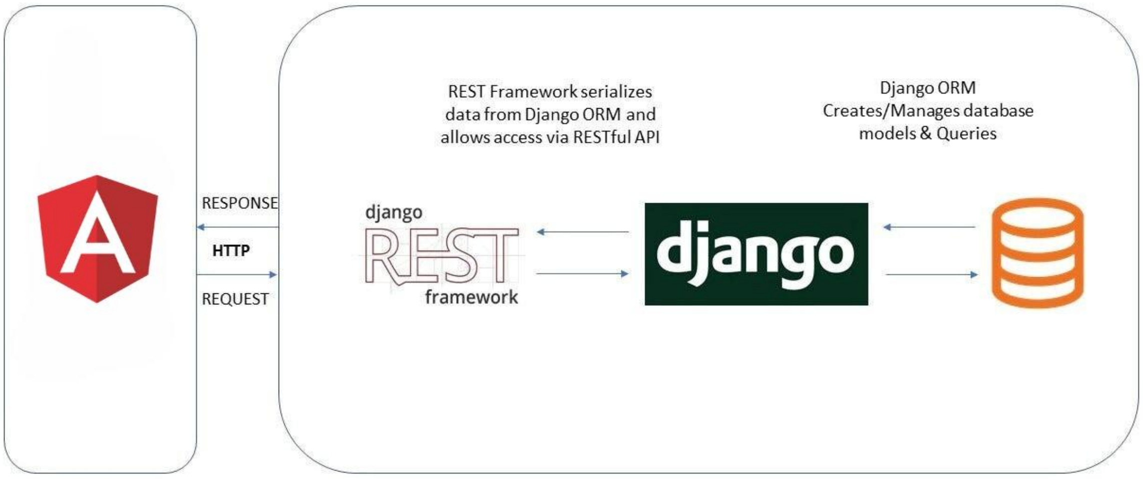

RESPONSE
HTTP
REQUEST
REST Framework serializes data from Django ORM and allows access via RESTful API
Django ORM Creates/Manages database models & Queries
django REST framework
django

Figure 7:Application architecture

❖ **Presentation Layer:**

The front-end, built with **Angular** framework, handles the user interface, presentation logic, and user interactions. It communicates with the backend application layer through the API using RESTful APIs, fetching data, performing actions, and updating the user interface accordingly.

❖ **Application Layer:**

In this layer, the logical processing is performed using the Spring framework, specifically **Django**.

The application layer, handles the business logic and data processing. These services communicate with the frontend through the API, which acts as a mediator between the presentation tier and the backend.

❖ **Data Layer:**

In the storage layer, user-related data is stored in the **SQLite3** database.

5. Work environment

5.1. Hardware environment

This application was developed using a laptop with this hardware setup:

Owner	**Sameh Abdelmajid Kouki**
Brand & Model	Asus

Processor	Intel Core i5-10300H
Ram	8 GO
Storage	500GB SSD
Operating System	Windows 10 Professional 64X

Tableau 6 :Hardware environment

5.2. Software environment

❖ **Visual Studio Code:**

VS code is a free open-source code editor developed by Microsoft for Windows, Linux, and MacOS with an enormous number of tools such as Git, Code Debugging, code snippets, etc.

❖ **Postman:**

Postman is a popular API client that makes it easy for developers to create, share, test and document APIs. This is done by allowing users to create and save simple and complex HTTP/s requests, as well as read their responses.

❖ **GitHub:**

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

5.3.Programming languages

❖ **Python:**

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically typed and garbage-collected..

❖ **JavaScript:**

JavaScript, often abbreviated as JS, is a programming language and core technology of the Web, alongside HTML and CSS.

5.4.Frameworks

❖ **DjangoJS:**

Django is a free and open-source, Python-based web framework that runs on a web server. It follows the model–template–views (MTV) architectural pattern.

- **Angular:**

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on components.

Conclusion

In this chapter, i get an overview of the different system functionalities requested by my product owner, in which i briefly described the essential features of the application, in addition to represent all the technologies and methods used to organize and develop this project which results in having a work plan (sprints and releases)

Chapter 3: Release

Introduction

This chapter presents the release of the project, consisting of two sprints. Each sprint includes analysis, design, implementation, and functional testing

❖ Sprint 1 «Core Functionality (Admin and Client Basic Features)»

1. Sprint introduction

This sprint aims to develop the first part of my project, which includes:

- Authentication.
- Admin can manage subscriptions
- The client can buy subscriptions
- Admin consult Clients

2. Sprint goal

The goal for this sprint is to develop the core functionalities of my project, focusing on authentication ,subscriptions and client management.

3. Sprint backlog

The sprint backlog is a set of functional tasks that the scrum team needs to accomplish during a sprint, for example in this sprint my goal is to manage the external contexts, so i list all the functional demands required to realize this sprint in the table below.

ID	User Story	Task ID	Task	Estimation
1	**Authentication.**	1.1	Prepare interface authentication for Login	4 days
		1.2	Implement authentication API.	
2	**manage subscriptions**	2.1	Prepare interface Admin dashboard	5 days
		2.2	Implement authentication API.	
3	**client can buy subscriptions**	3.1	Prepare interface shop for Client	3 days
		3.2	Implement authentication API.	

4	**Admin consult Clients**	4.1	Prepare interface Client table for Admin	3days
		4.2	Implement authentication API.	

Tableau 7:Backlog of sprint 1

4. Analyze

This section presents the sprint use case diagram in addition to a textual description of the main functionalities.

Use case diagram.

A UML use case diagram is a type of diagram used to represent the functional requirements of a system or application. It provides a high-level view of the interactions between a system or application and its users, actors, or other systems.

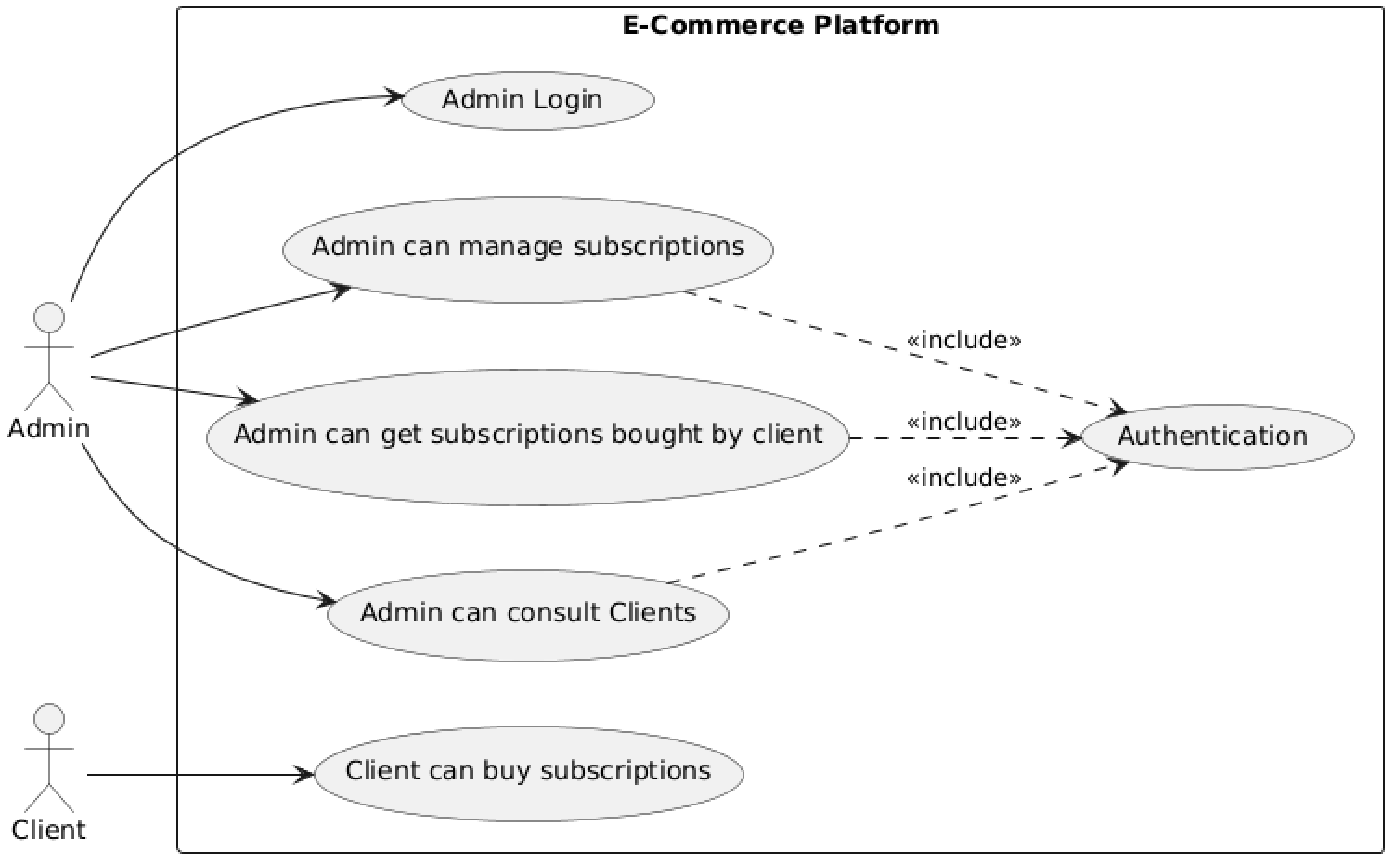

E-Commerce Platform
Admin Login
Admin can manage subscriptions
Admin can get subscriptions bought by client
Admin can consult Clients
Client can buy subscriptions
Authentication
«include»
«include»
«include»
Admin
Client

Figure 8: Use case sprint1

A. Textual Description

The textual description explains the interaction between actors and system, by documenting the use cases and the different scenarios that can happen during the execution.

❖ **Textual Description of use case «Authentication»:**

Use Case	Authentication.
Actors	Admin
Pre-condition	Actor previously created.
Post-condition	Access to the actor's personal space.
Nominal scenario	1. The actor requests the connection. 2. The system displays the authentication form. 3. The actor fills in the required fields and validates. 4. The system verifies the entered data. 5. The system opens the personal space of the actor.
Alternative scenario	4-a. The actor enters incorrect data. 4-a-1. The system displays error messages. 4-a-2. Return to step 3 of the nominal scenario.

Tableau 8: Textual description of use case «Authentication»

5. Conception

A. Class diagram

Class diagram is a type of UML diagram that depicts the structure of a system or application by showing the classes, attributes, methods, and relationships between them.

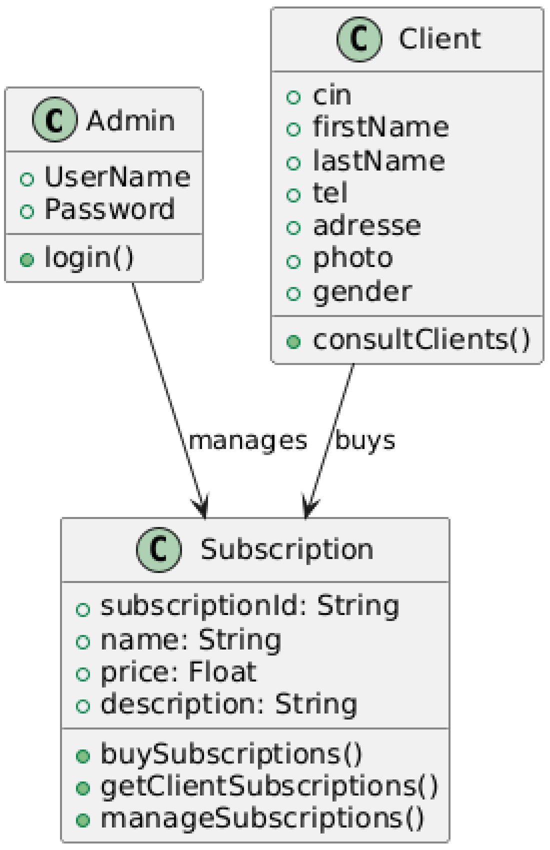


Figure 9: class diagram sprint1

B. Sequence diagram

A sequence diagram is a type of UML diagram that illustrates the flow of messages and interactions between objects or components of a system in a specific scenario or use case.

It shows the sequence of events or steps that occur during the execution of a particular process, including the order in which messages are exchanged between objects and the time at which they occur.

❖ **Authentication sequence diagram.**

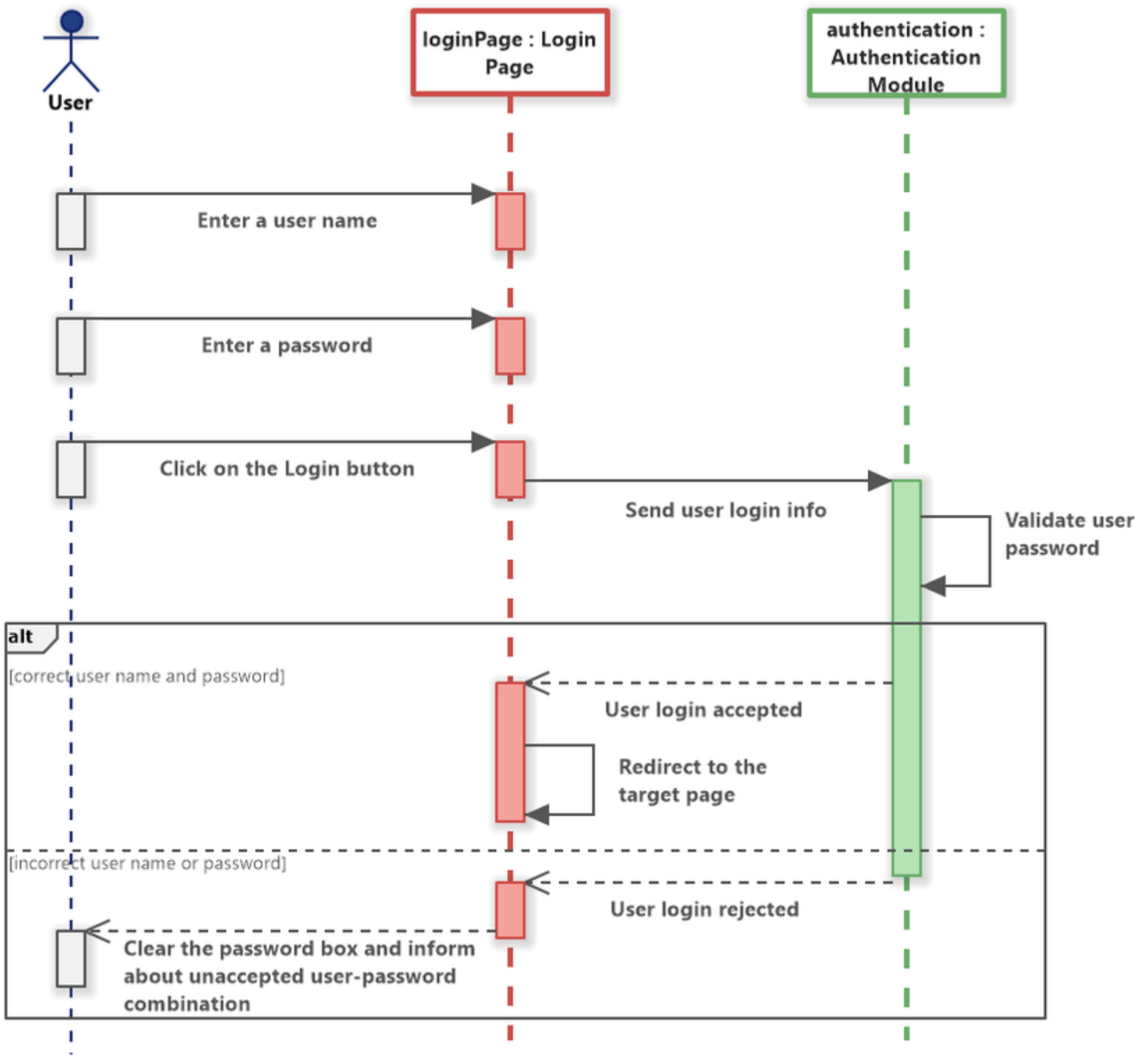

User
loginPage : Login Page
authentication : Authentication Module
Enter a user name
Enter a password
Click on the Login button
Send user login info
Validate user password
alt
[correct user name and password]
User login accepted
Redirect to the target page
[incorrect user name or password]
User login rejected
Clear the password box and inform about unaccepted user-password combination

Figure 10:Authentication sequence diagram.

❖ **Add subscriptions sequence diagram.**

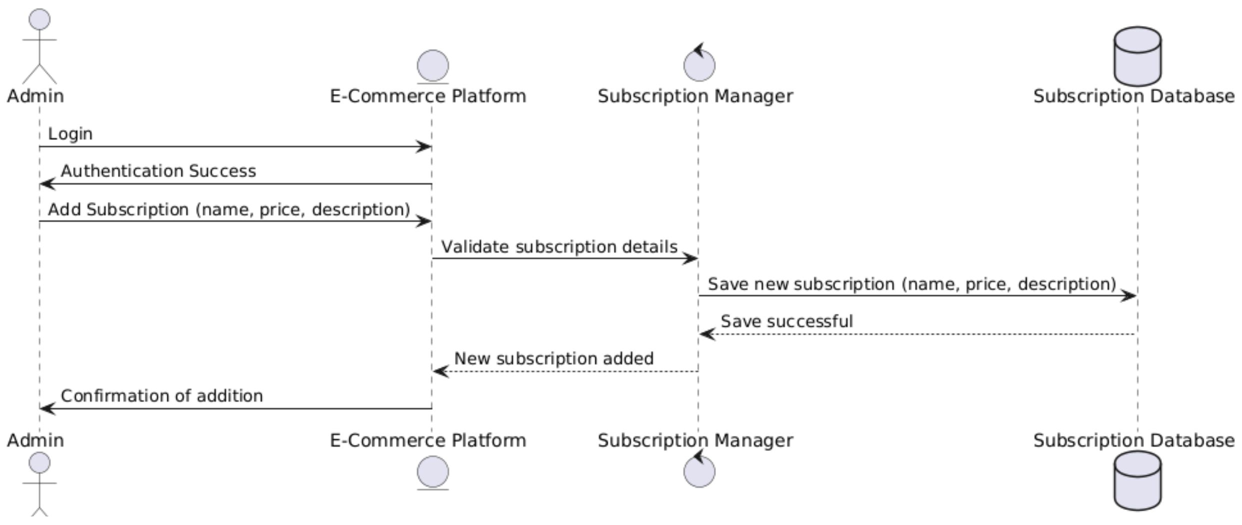


Figure 11:Add subscriptions sequence diagram

❖ Update subscriptions sequence diagram.

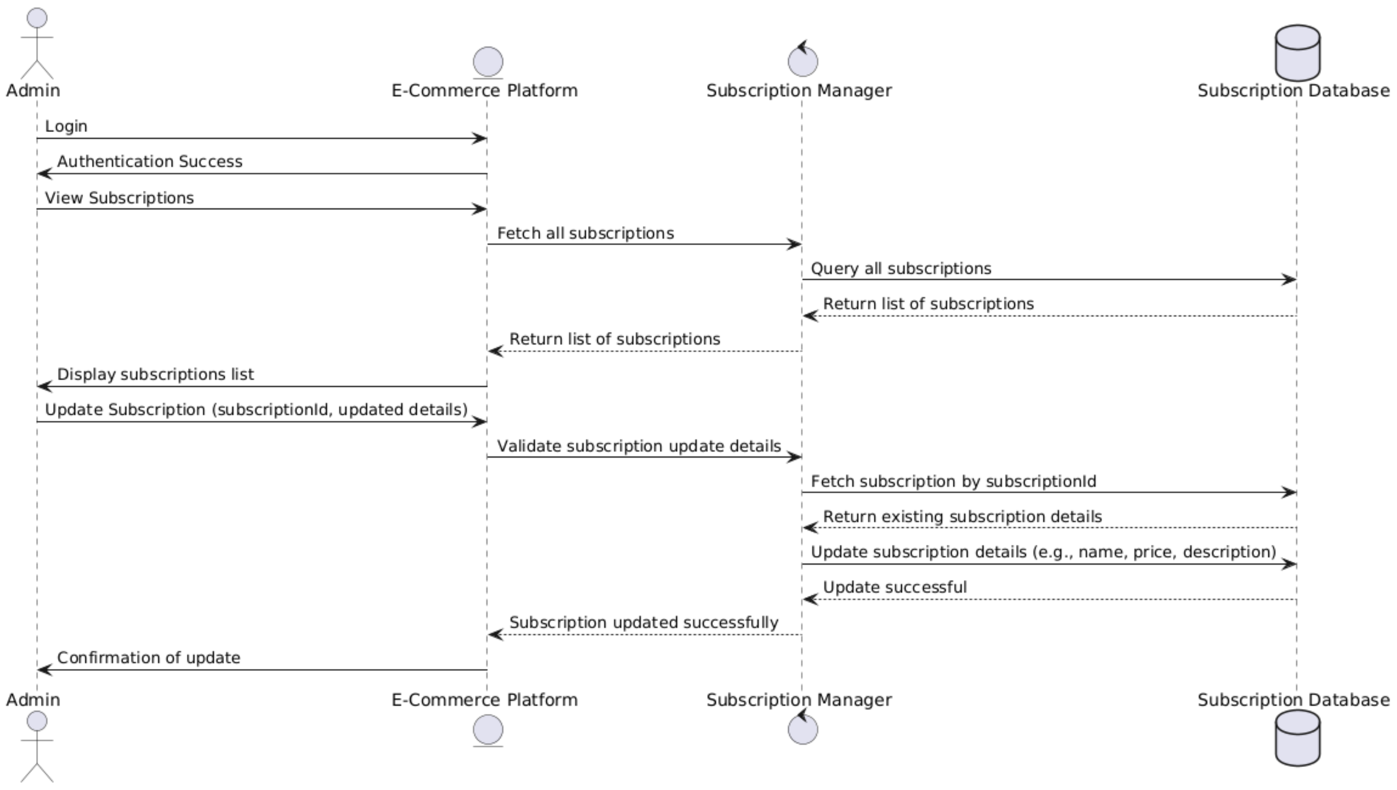

Admin
E-Commerce Platform
Subscription Manager
Subscription Database
Login
Authentication Success
View Subscriptions
Fetch all subscriptions
Query all subscriptions
Return list of subscriptions
Return list of subscriptions
Display subscriptions list
Update Subscription (subscriptionId, updated details)
Validate subscription update details
Fetch subscription by subscriptionId
Return existing subscription details
Update subscription details (e.g., name, price, description)
Update successful
Subscription updated successfully
Confirmation of update

Figure 12:Update subscriptions sequence diagram

❖ Delete subscriptions sequence diagram.

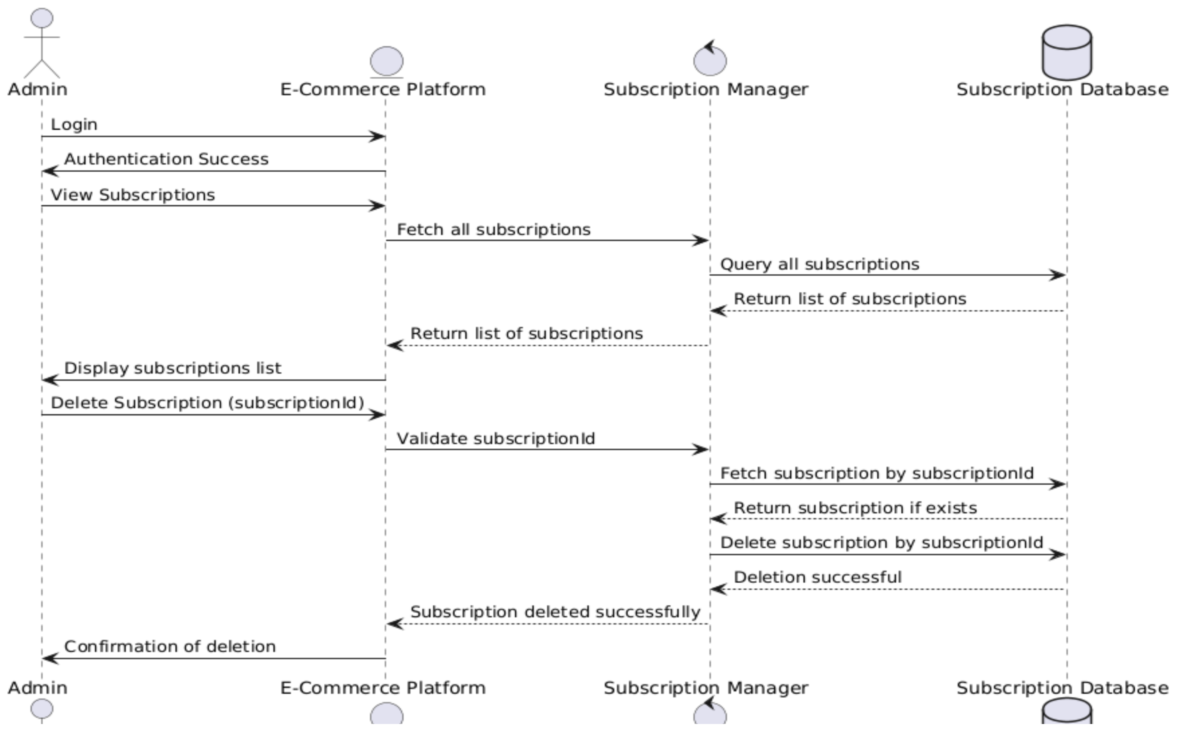

Admin
E-Commerce Platform
Subscription Manager
Subscription Database
Login
Authentication Success
View Subscriptions
Fetch all subscriptions
Query all subscriptions
Return list of subscriptions
Return list of subscriptions
Display subscriptions list
Delete Subscription (subscriptionId)
Validate subscriptionId
Fetch subscription by subscriptionId
Return subscription if exists
Delete subscription by subscriptionId
Deletion successful
Subscription deleted successfully
Confirmation of deletion

Figure 13:Delete subscriptions sequence diagram

❖ **Buy subscriptions sequence diagram.**

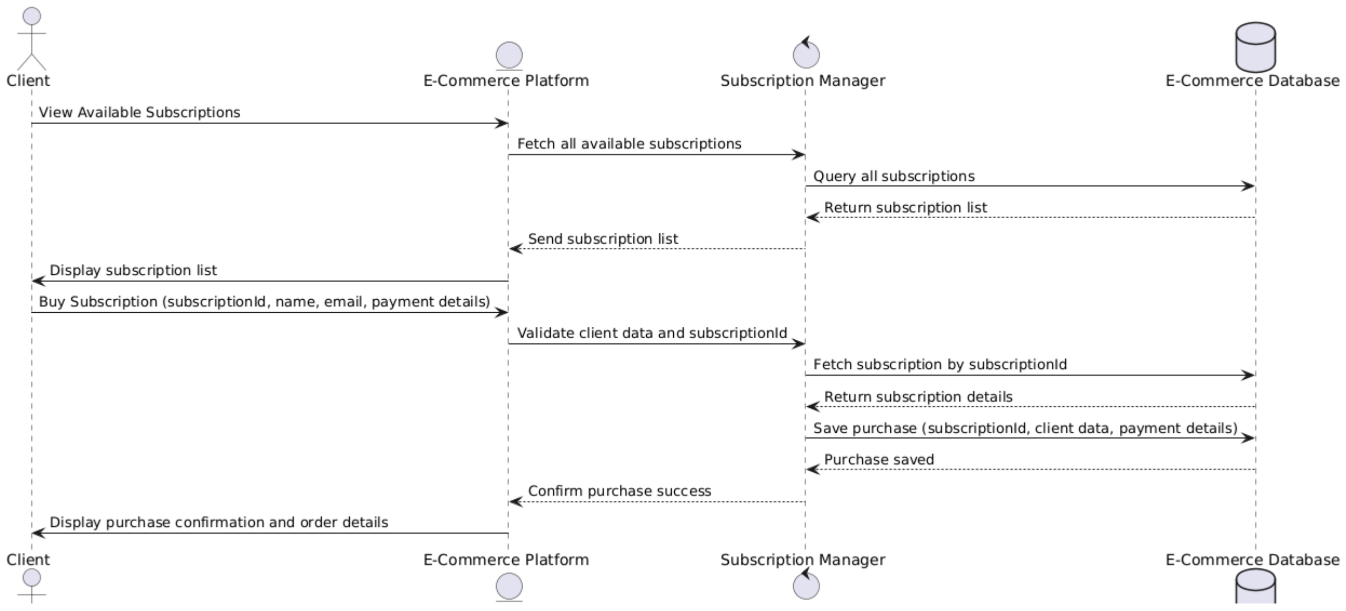

Client
E-Commerce Platform
Subscription Manager
E-Commerce Database
View Available Subscriptions
Fetch all available subscriptions
Query all subscriptions
Return subscription list
Send subscription list
Display subscription list
Buy Subscription (subscriptionId, name, email, payment details)
Validate client data and subscriptionId
Fetch subscription by subscriptionId
Return subscription details
Save purchase (subscriptionId, client data, payment details)
Purchase saved
Confirm purchase success
Display purchase confirmation and order details

Figure 14:Buy subscriptions sequence diagram

C. Activity diagram

The activity diagram is a behavioral diagram in UML that represents the flow of activities triggered by events based on the system's states. It is used to model parallelizable behaviors and describe workflow processes.

❖ Consult orders activity diagram.

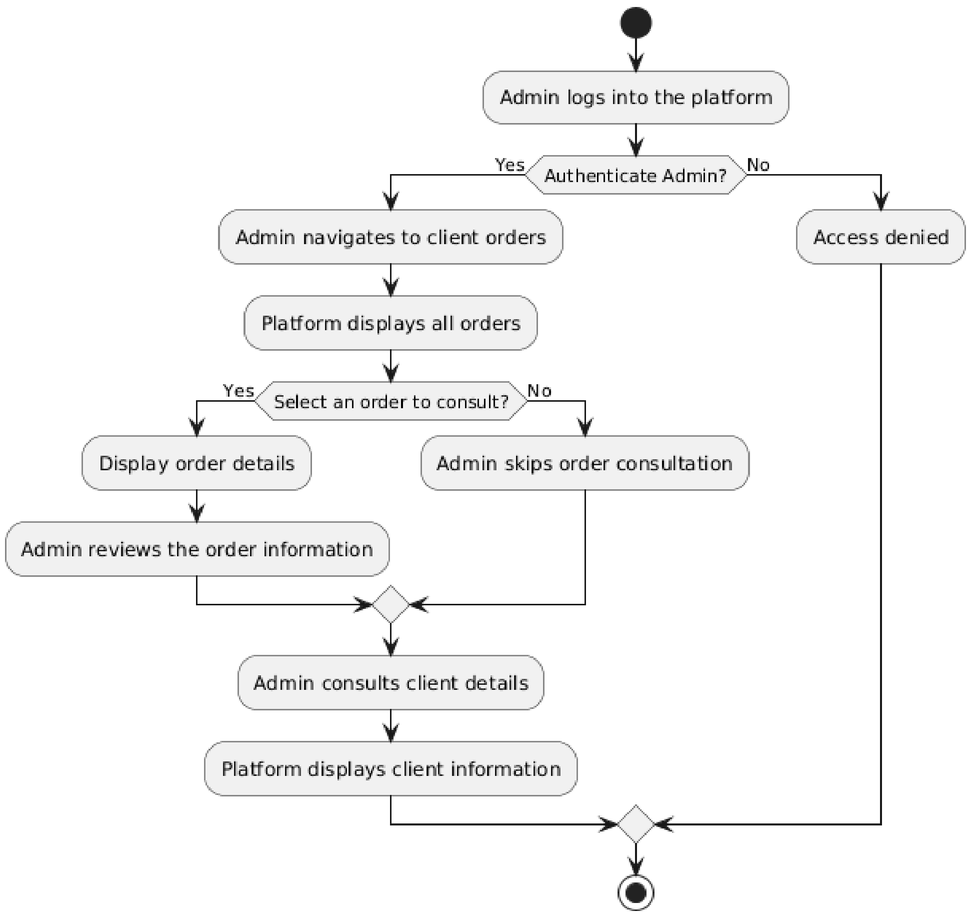

Admin logs into the platform
Yes
Authenticate Admin?
No
Admin navigates to client orders
Access denied
Platform displays all orders
Yes
Select an order to consult?
No
Display order details
Admin skips order consultation
Admin reviews the order information
Admin consults client details
Platform displays client information

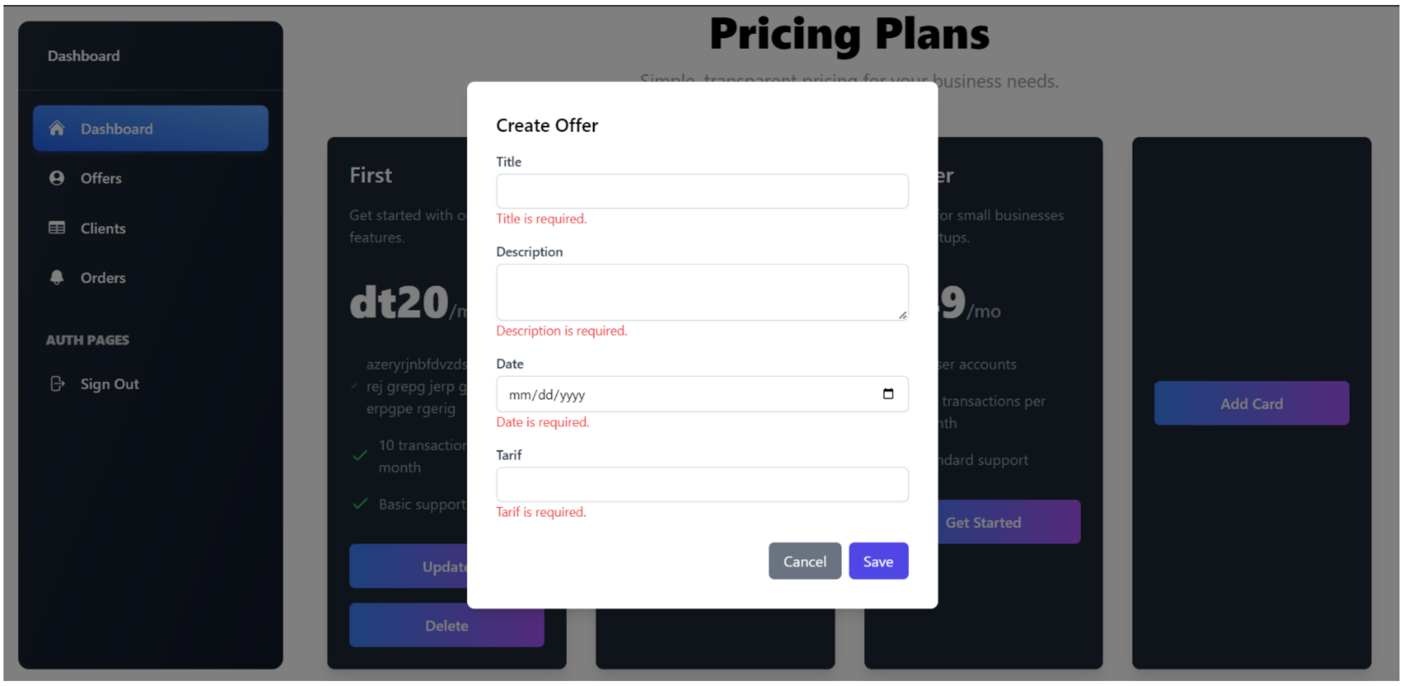
Figure 15:Consult orders activity diagram

6. Realization

This section is dedicated to showcasing the completed work through screenshots of different interfaces developed during this sprint.

❖ Manage Subscriptions

From this dashboard admin can Manage Subscriptions. This picture show the **ADD Subscription**



Create Offer

Title

Title is required.

Description

Description is required.

Date

mm/dd/yyyy

Date is required.

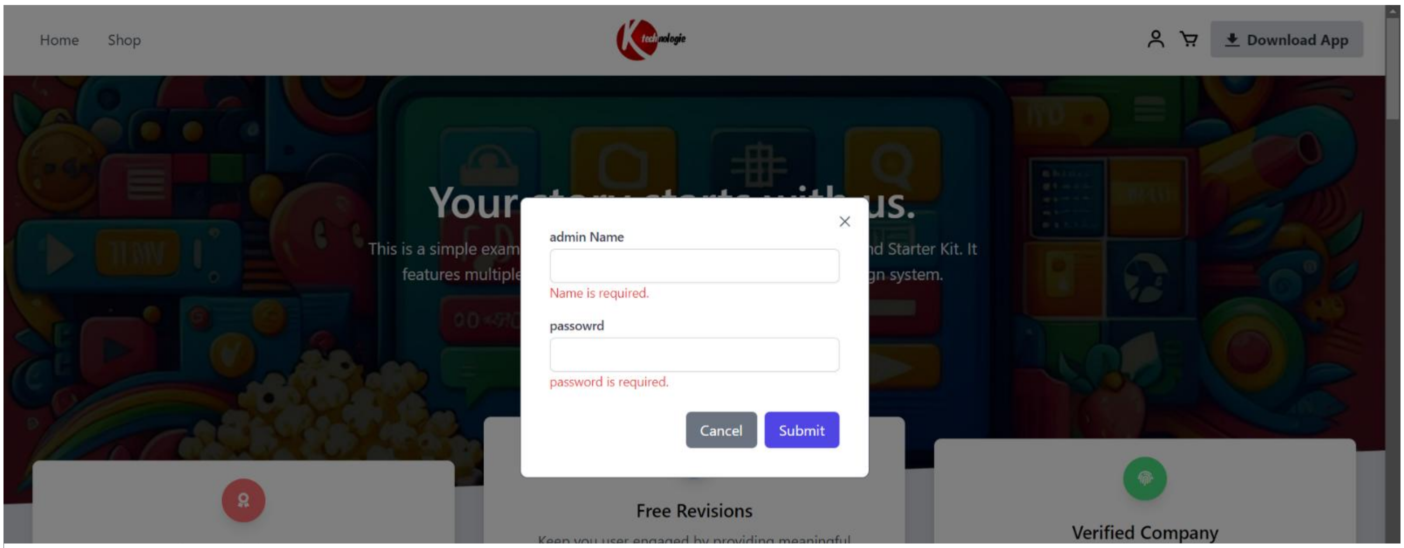
Tarif

Tarif is required.

Figure 16: Manage Offers release

❖ Login

From here the admin can Login.



admin Name

Name is required.

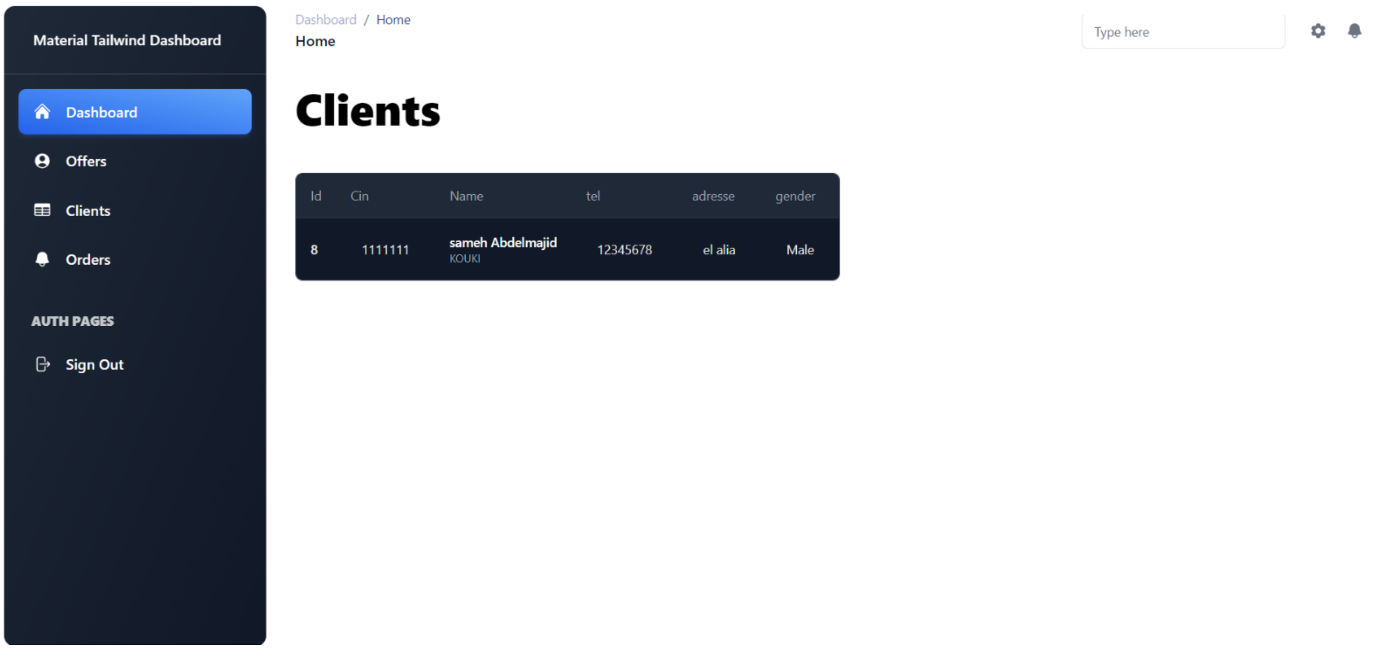
passowrd

password is required.

Figure 17: Login interface

❖ Consult Clients

Admin can Consult Clients here.



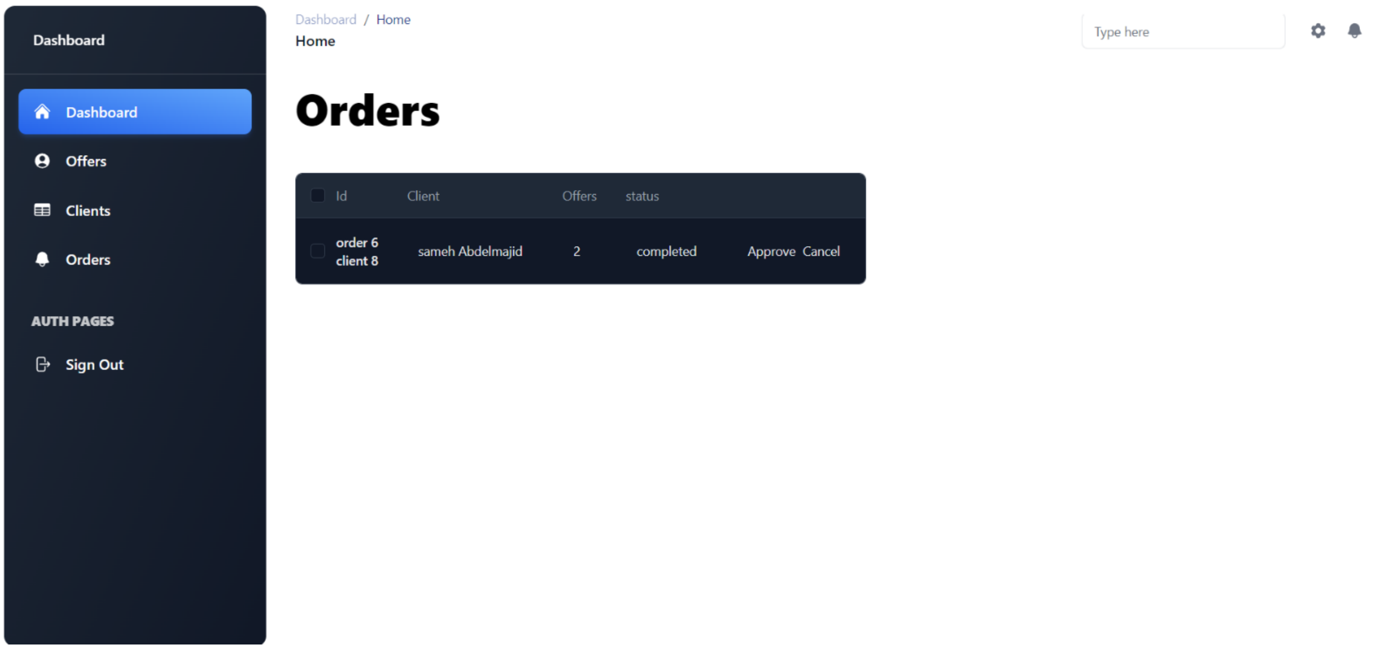
Clients

Id	Cin	Name	tel	adresse	gender
8	1111111	sameh Abdelmajid KOUKI	12345678	el alia	Male

Figure 18:clients interface

❖ Orders

Here the admin can valide or cancel the order.



Orders

Id	Client	Offers	status	
order 6 client 8	sameh Abdelmajid	2	completed	Approve Cancel

Figure 19:orders interface

7. Testing

The testing of a software product is a systematic process aimed at ensuring the proper functioning of the system by comparing expected behaviors with actual results.

Before the end of each sprint, i tested the features of the module. Afterwards, i validated all the functionalities with the Product Owner.

To do so, i have prepared a set of test scenario cases related to Sprint 1 in the table below.

Test Scenario	Steps	Expected behavior	Result
Test the authentication of Admin	1. Navigate to the login page. 2. Enter valid credentials for Admin. 3. Log in as Admin.	Full functional authentication	In accordance with the expected behavior.
Test managing subscriptions as Admin	1. Log in as Admin. 2. Navigate to the subscription management page. 3. Add a subscription. 4. Update the subscription. 5. Delete the subscription.	Subscriptions should be successfully added, updated, and deleted.	In accordance with the expected behavior.
Test the ability for the Client to buy subscriptions	1. Navigate to the subscriptions page. 2. Select a subscription to buy. 3. Fill in required details (e.g., name, email). 4. Confirm the purchase.	The subscription is successfully purchased, and the order is saved in the database.	In accordance with the expected behavior.
Test Admin consulting Clients	1. Log in as Admin. 2. Navigate to the Clients consultation page. 3. View the list of registered Clients. 4. Verify that all client details are displayed correctly.	Admin can view a complete and accurate list of Clients.	In accordance with the expected behavior.

Tableau 9:Testing table of sprint1

Table 1: Testing table of sprint 1-Release 1

❖ Sprint 2 « Enhancements and Admin Features»

1. Sprint introduction

This sprint focuses on services and profile management. In this sprint, i will be building upon the foundation laid in Sprint 1 and adding new features to my system.

- Client can send emails
- Admin can consult statistics
- Admin manage subscriptions that the client bought
- The client can manage cart

2. Sprint goal

The sprint goal for this iteration is to develop the necessary functionalities manage bought subscriptions and cart management.

Additionally, the goal is to provide Admin with tools to Consult statistics.

3. Sprint backlog

ID	User Story	Task ID	Task	Estimation
1	Bought subscriptions management	1.1	Design and develop service publishing interface.	4 days
		1.2	Implement API for uploading user picture.	
2	Cart management	2.1	Prepare interface user profile	5 days
		2.2	Implement API for uploading user picture.	
3	Consult statistics	3.1	Prepare interface user profile	3 days
		3.2	Implement API for uploading user picture.	

4	Send mails	4.1	Prepare interface user profile	3 days
		4.2	Implement API for uploading user picture.	

Tableau 10: Sprint 2 Backlog

4. Analyze

A. Use case diagram.

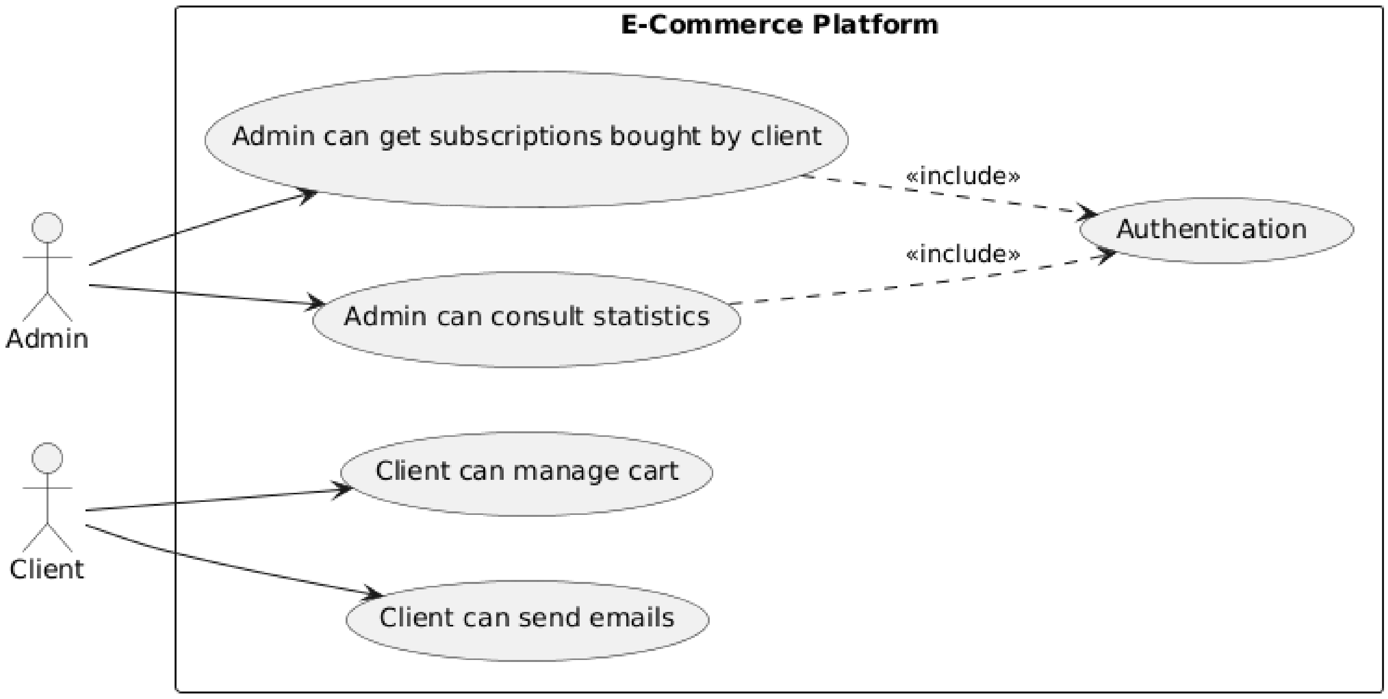


Figure 20 : Sprint2 use case

B. Textual Description

❖ **Textual Description of use case « Client Sends Mail»:**

Use Case	Client Sends Mail
Actors	Client
Pre-condition	
Post-condition	The email is successfully sent to the admin or support team, and the client is notified of the successful sending

Nominal scenario	1. Client navigates to the email section or the support page of the platform. 2. Client fills in the necessary details of the email, including the recipient (admin/support), subject, and body of the message. 3. Client clicks the "Send" button to submit the email. 4. The system verifies that the email contains all necessary information (recipient, subject, and message body). 5. The system sends the email to the designated recipient (admin/support). 6. The system notifies the client that the email has been sent successfully.
Alternative scenario	1a. If the email is missing required fields (such as recipient or message body), the system prompts the client to complete the missing information. 1b. If there is a technical issue (e.g., connectivity problems or invalid email), the system notifies the client of the error and suggests retrying.

Tableau 11 : Textual Description of use case « Client Sends Mail»

❖ **Textual Description of use case « Admin Consults Statistics»:**

Use Case	**Admin Consults Statistics**
Actors	Admin
Pre-condition	The admin must be authenticated and logged into the platform. The platform must have statistical data available (e.g., sales, subscriptions, client activity).
Post-condition	The admin successfully views the statistics, including any relevant data (e.g., total sales, active subscriptions, user engagement)
Nominal scenario	1. Admin logs into the platform. 2. Admin navigates to the "Statistics" section.

	3. The system retrieves and displays the relevant statistical data, such as total sales, subscriptions, active clients, etc. 4. Admin reviews the displayed statistics. 5. Admin may choose to filter or adjust the displayed statistics (e.g., by date range, subscription type, or region). 6. The system updates and shows the filtered or adjusted statistics. 7. The admin exits or continues with other tasks.
Alternative scenario	1a. If the statistical data is unavailable or incomplete, the system notifies the admin with an error message (e.g., "No data available" or "Unable to load statistics"). 1b. If the admin attempts to filter by a non-existing category, the system prompts the admin to adjust the filter criteria. 1c. If there is a technical issue (e.g., database connection problem), the system notifies the admin of the failure and suggests retrying.

Tableau 12:Textual Description of use case « Admin Consults Statistics»

5. Conception

A. Class diagram

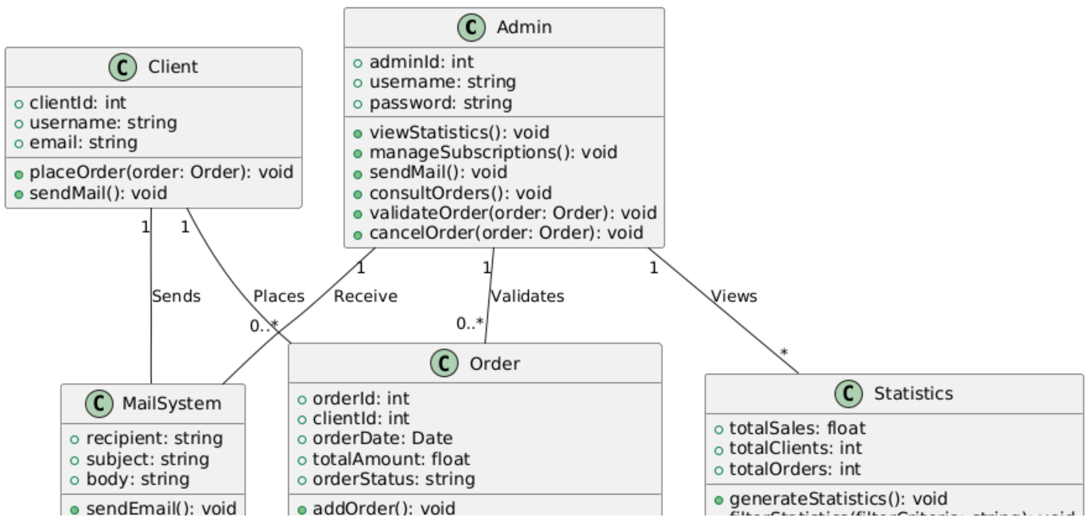


Figure 21: Sprint2 Class diagram

B. Activity diagram

The activity diagram is a behavioral diagram in UML that represents the flow of activities triggered by events based on the system's states. It is used to model parallelizable behaviors and describe workflow processes.

❖ **Buy Subscription activity diagram.**

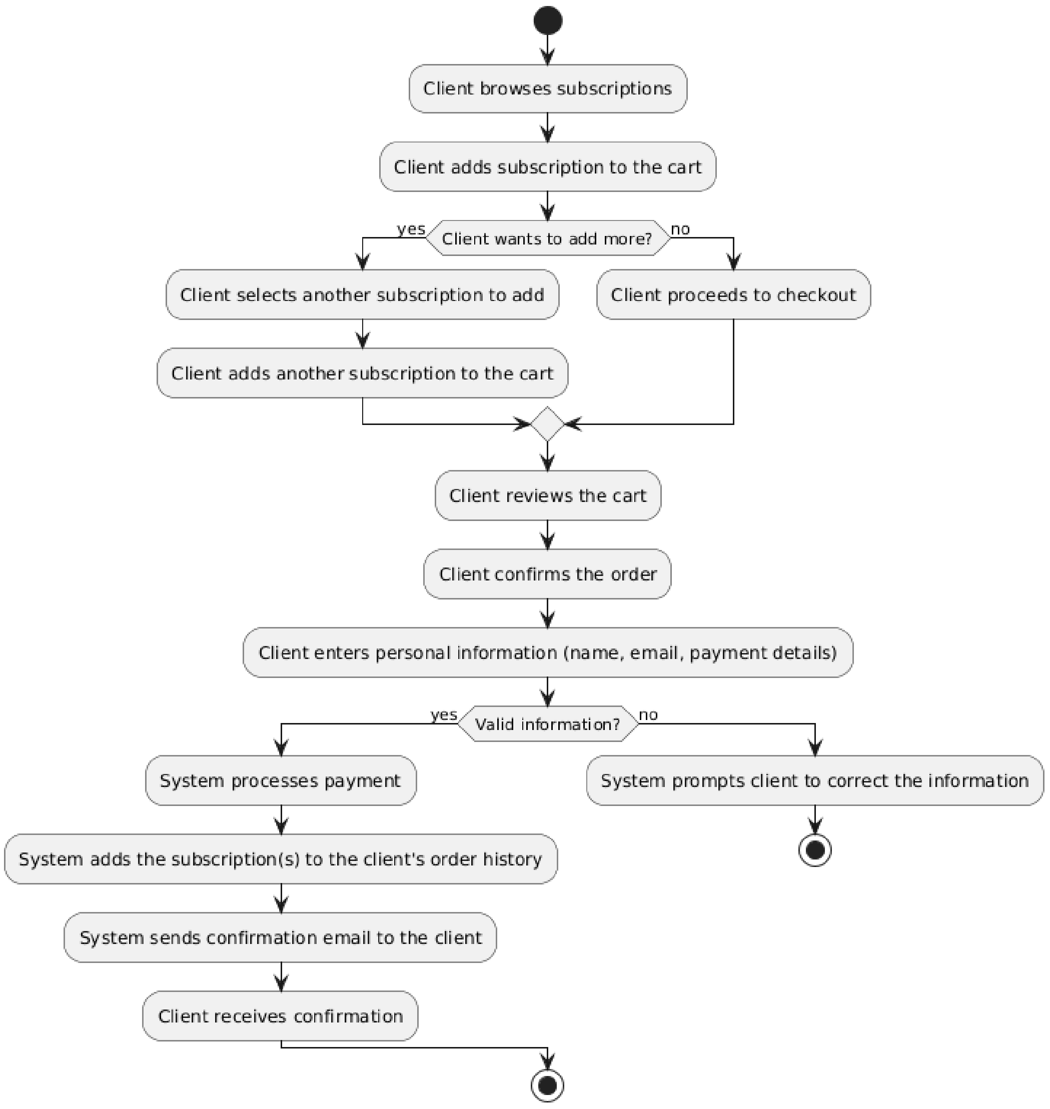

Client browses subscriptions
Client adds subscription to the cart
yes
Client wants to add more?
no
Client selects another subscription to add
Client proceeds to checkout
Client adds another subscription to the cart
Client reviews the cart
Client confirms the order
Client enters personal information (name, email, payment details)
yes
Valid information?
no
System processes payment
System prompts client to correct the information
System adds the subscription(s) to the client's order history
System sends confirmation email to the client
Client receives confirmation

Figure 22: Buy Subscription activity diagram.

C. Sequence diagram

❖ Manage cart sequence diagram.

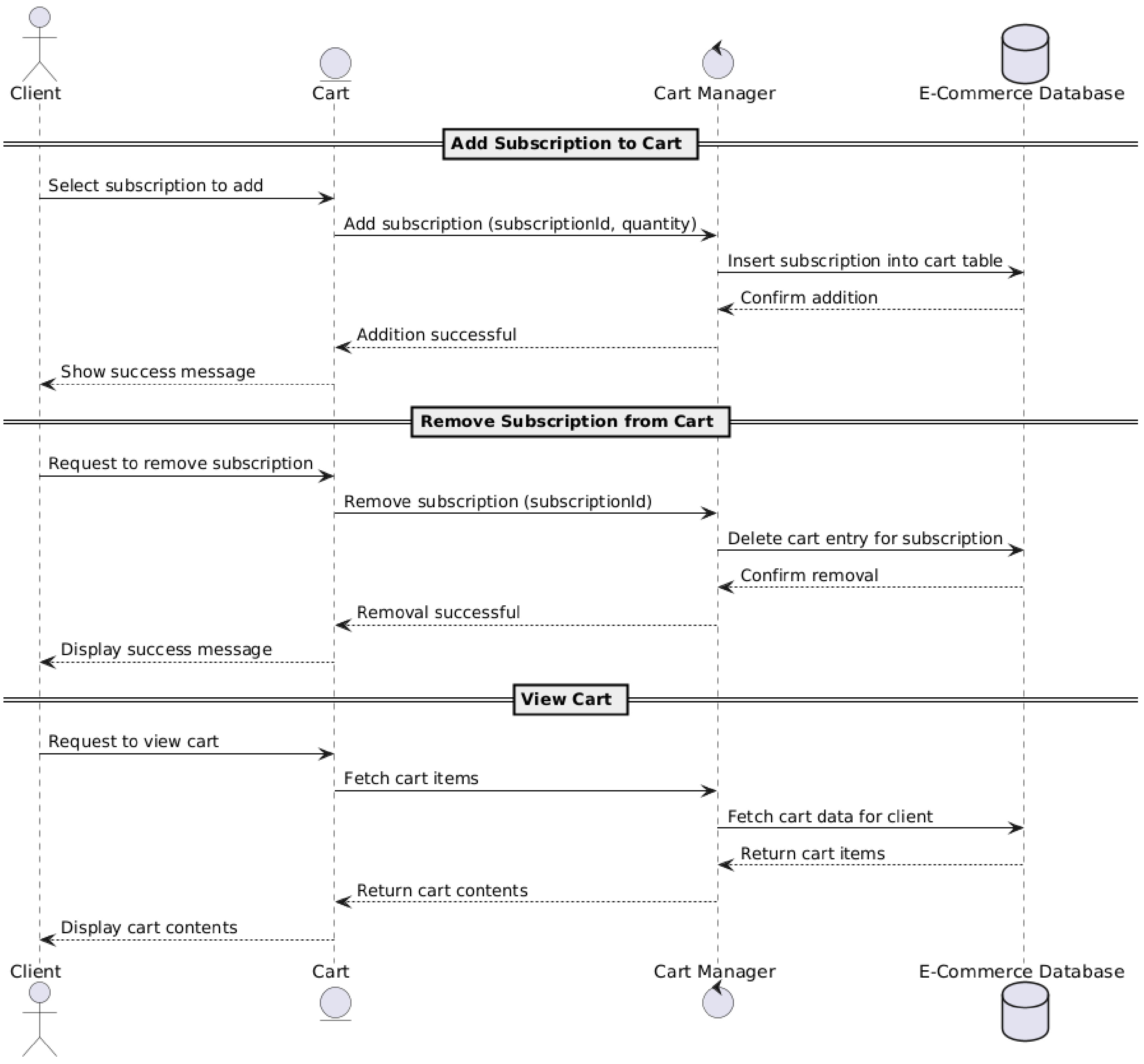


Figure 23: Manage cart sequence diagram.

1. Realization

❖ Statistics

This interface shows the statistics.

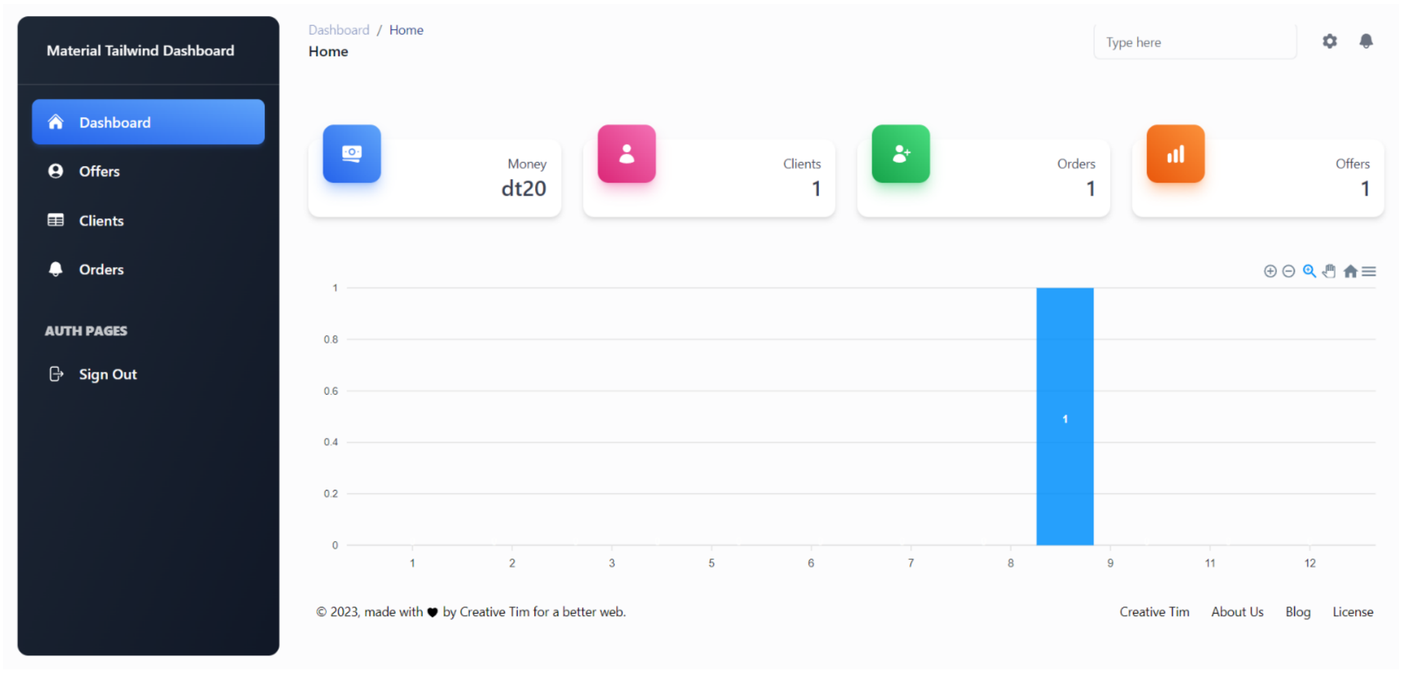
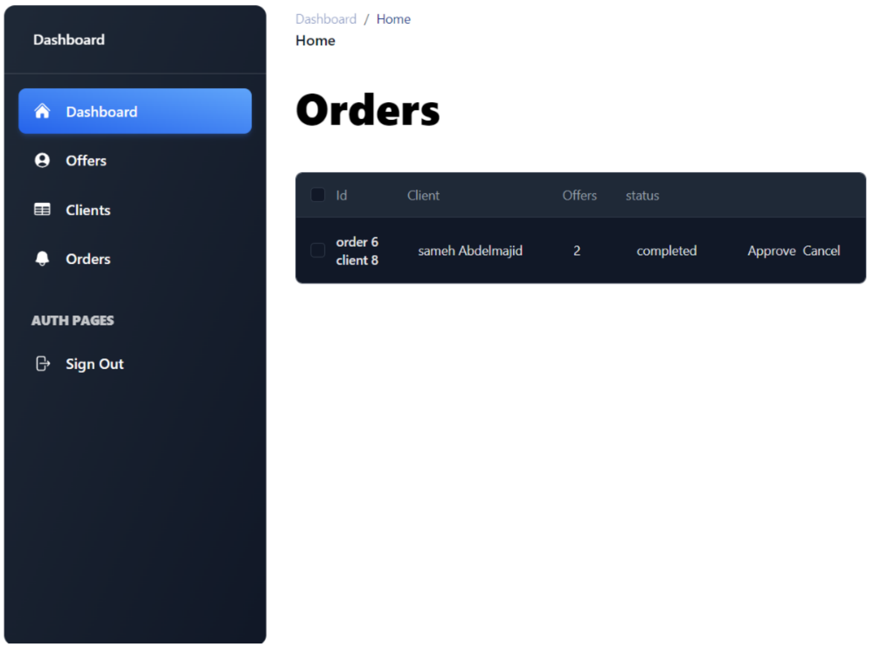


Figure 24: Statistics interface

❖ Manage orders

From this page the admin can manage orders.

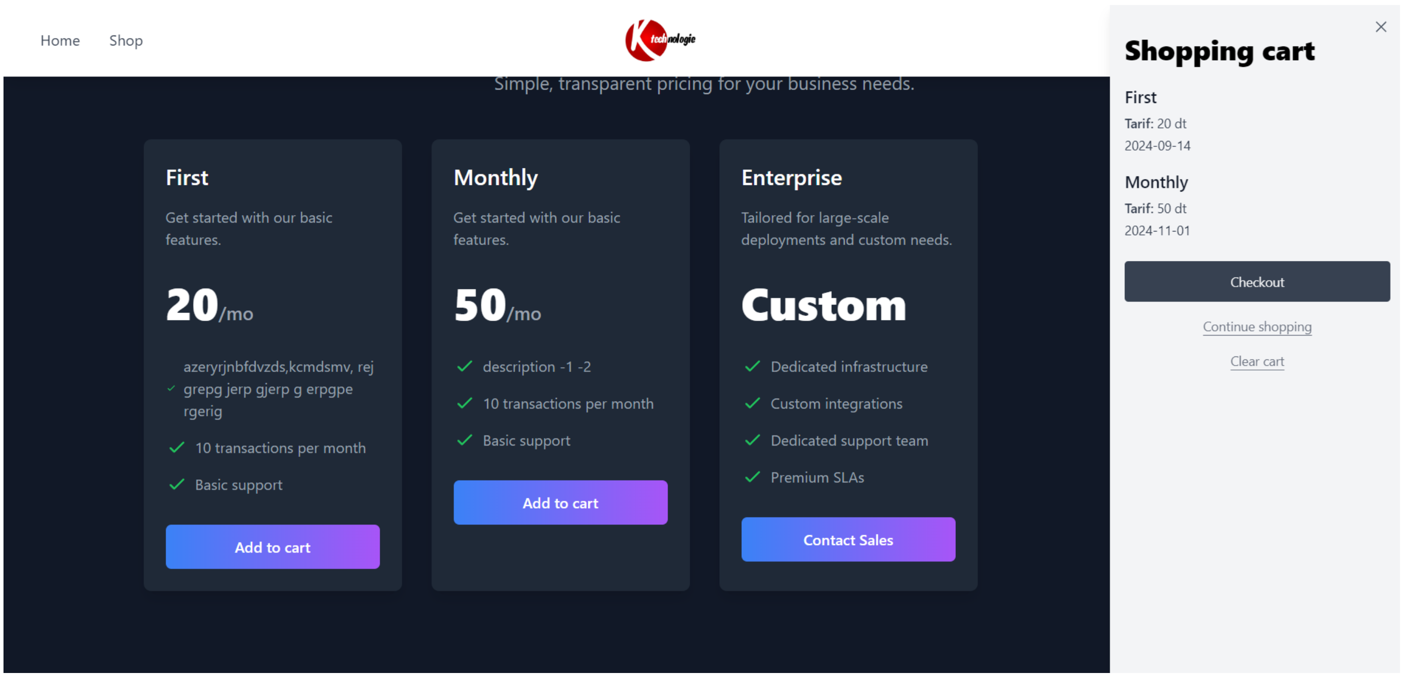


Id	Client	Offers	status	
order 6 client 8	sameh Abdelmajid	2	completed	Approve Cancel

Figure 25: manage orders interface

❖ Cart

The offers that client choose appear in the cart


Home
Shop
Simple, transparent pricing for your business needs.
First
Get started with our basic features.
20/mo
azeryrjnbfdvzds,kcmdsmv, rej grepg jerp gjerp g erpgpe rgerig
10 transactions per month
Basic support
Add to cart
Monthly
Get started with our basic features.
50/mo
description -1 -2
10 transactions per month
Basic support
Add to cart
Enterprise
Tailored for large-scale deployments and custom needs.
Custom
Dedicated infrastructure
Custom integrations
Dedicated support team
Premium SLAs
Contact Sales

Shopping cart

First
Tarif: 20 dt
2024-09-14

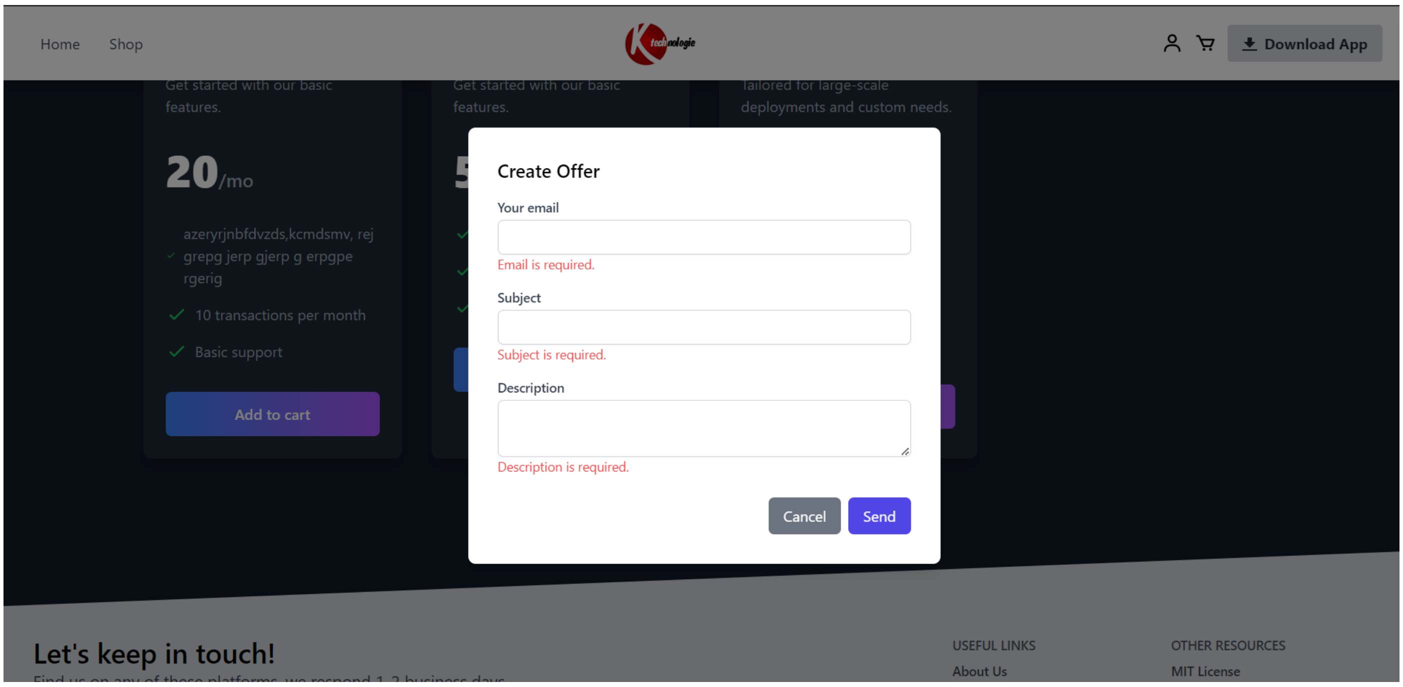
Monthly
Tarif: 50 dt
2024-11-01

Checkout
Continue shopping
Clear cart

Figure 26: Cart interface

❖ Emails

From here the client can send mail to the admin



Create Offer

Your email

Email is required.

Subject

Subject is required.

Description

Description is required.

Figure 27: mailing interface

2. Testing

Test Scenario	Steps	Expected behavior	Result
Test the management of bought subscriptions	1. Log in as Admin. 2. Navigate to the page listing all purchased subscriptions. 3. Verify the details of each subscription (e.g., buyer, date, price).	Admin can view and manage (validate or cancel) all purchased subscriptions accurately.	In accordance with the expected behavior.
Test the management of the Client's cart	1. Add items to the cart. 2. View the cart page. 3. Remove an item from the cart. 4. Update the quantity of an item in the cart. 5. Proceed to checkout.	The cart is updated correctly (items added, removed, or updated), and the checkout works as expected.	In accordance with the expected behavior.
Test Admin consulting statistics	1. Log in as Admin. 2. Navigate to the statistics page. 3. View sales, orders, and subscription data.	Admin can view accurate and updated statistics regarding the platform's activity	In accordance with the expected behavior.
Test Client sending mails to Admin	1. Navigate to the "Contact Admin" page. 2. Fill in the email form (e.g., subject, message). 3. Submit the email.	The email is sent successfully, and the Admin receives it in their inbox.	In accordance with the expected behavior.

Tableau 13: Testing table of Sprint 2

Conclusion

In this chapter, i introduced the only release. To accomplish this, i went through the analysis, design, implementation, and testing phases for each sprint.

General conclusion

The development of the e-commerce platform for selling IPTV subscriptions was a valuable and enriching experience. This project allowed me to delve into various aspects of software development, including front-end and back-end integration, database management, and the implementation of essential e-commerce functionalities such as subscription management, user authentication, and order processing.

Throughout the project, I applied theoretical knowledge to practical scenarios, improving my technical skills in web development frameworks and tools. Collaborating with team members and adhering to best practices like Agile methodologies enabled me to better understand project management processes and the importance of iterative development.

Moreover, the challenges faced during the implementation—such as ensuring secure authentication, handling complex subscription logic, and maintaining a seamless user experience—helped me develop problem-solving skills and resilience. These challenges provided insights into how real-world systems are built, optimized, and maintained.

This internship at **K-Technologie** not only enhanced my technical expertise but also allowed me to grow professionally by working in a dynamic environment. It reinforced the importance of effective communication, time management, and continuous learning in the field of software engineering.

The final product, a functional and scalable e-commerce platform, stands as a testament to the efforts and learning gained throughout the internship. I am confident that the knowledge and experience acquired during this project will serve as a strong foundation for future endeavors in software development.