Ministry of Higher Education and Scientific Research

And Information and Communication Technologies

Générale Direction of Technologies studies

**Higher Institute of Technologies studies of Bizerte**

**Information Technology Department**

**INTEGRATION PROJECT REPORT**

Subject: Development of purchasing and sales applications

**Realized By:** Sprint Squad Team

**Supervising teacher:** Miss Afef Gafsi

**Academic year: 2023-2024 DSI31**

**Summary**

[General Introduction 1](#_Toc153442147)

[Chapter 1: Preliminary study 2](#_Toc153442148)

[Introduction 2](#_Toc153442149)

[1.1. Description of existing equipment 2](#_Toc153442150)

[1.2. Existences Criticisms 2](#_Toc153442151)

[1.3. Proposed Solution 3](#_Toc153442152)

[1.4. Adapted working method. 4](#_Toc153442153)

[a SCRUM project management method 4](#_Toc153442154)

[b Done criteria 4](#_Toc153442155)

[c Sprints Time Box 4](#_Toc153442156)

[1.5. Working tools 4](#_Toc153442157)

[a Frameworks used 4](#_Toc153442158)

[b Software development 5](#_Toc153442159)

[Conclusion 8](#_Toc153442160)

[Chapter 2: Product Backlog planning 9](#_Toc153442161)

[Introduction 9](#_Toc153442162)

[2.1. User’s identification 9](#_Toc153442163)

[2.2. User Stories 10](#_Toc153442164)

[2.3. Application Flows 11](#_Toc153442165)

[2.4. Product Backlog 12](#_Toc153442166)

[Conclusion 14](#_Toc153442167)

[Chapter 3: Sprint 1 15](#_Toc153442168)

[Introduction: 15](#_Toc153442169)

[3.1 Sprint goals 15](#_Toc153442170)

[3.2 Sprint backlog 15](#_Toc153442171)

[3.3 Sprint 1 implementation 19](#_Toc153442172)

[a System description phase 19](#_Toc153442173)

[b Analyze 22](#_Toc153442174)

[c Conception phase 24](#_Toc153442175)

[d Interfaces Models 26](#_Toc153442176)

[3.4 Sprint retrospective 29](#_Toc153442177)

[Conclusion 30](#_Toc153442178)

[Chapter 4: Sprint 2 31](#_Toc153442179)

[Introduction 31](#_Toc153442180)

[4.1 Sprint goals 31](#_Toc153442181)

[4.2 Sprint Backlog 31](#_Toc153442182)

[4.3 Sprint implementation: 33](#_Toc153442183)

[a System decription phase 34](#_Toc153442184)

[b Analyze 35](#_Toc153442185)

[c Conception phase 36](#_Toc153442186)

[d Interfaces Models 38](#_Toc153442187)

[4.4 Sprint Retrospective 38](#_Toc153442188)

[Conclusion 38](#_Toc153442189)

[Chapter 5: Sprint 3 39](#_Toc153442190)

[Introduction 39](#_Toc153442191)

[5.1 Sprint goals: 39](#_Toc153442192)

[5.2 Sprint backlog: 39](#_Toc153442193)

[5.3 Sprint implementation 41](#_Toc153442194)

[a System description phase 41](#_Toc153442195)

[b Analyze 42](#_Toc153442196)

[5.4 Sprint retrospective 44](#_Toc153442197)

[Keyclock security implementation 44](#_Toc153442198)

[Implementation of a CI/CD pipeline based on docker 45](#_Toc153442199)

[Implementation : 47](#_Toc153442200)

[Conclusion 48](#_Toc153442201)

[General Conclusion 49](#_Toc153442202)

[Bibliography 50](#_Toc153442203)

**List of figures**

[Figure 1: NodeJS and Express [1] 5](#_Toc153442204)

[Figure 2: Android Native with Koltin [2] 5](#_Toc153442205)

[Figure 3: Spring boot logo [3] 5](#_Toc153442206)

[Figure 4: VsCode logo [4] 6](#_Toc153442207)

[Figure 5: Android Studio logo [5] 6](#_Toc153442208)

[Figure 6: IntelliJ ide logo [6] 6](#_Toc153442209)

[Figure 7: Postman logo [7] 7](#_Toc153442210)

[Figure 8: Github logo [8] 7](#_Toc153442211)

[Figure 9: Github Repository 7](#_Toc153442212)

[Figure 10: Application browsing flow 11](https://mohetn.sharepoint.com/sites/DSI312023-2024-eq3SprintSquad/Shared%20Documents/eq3%20SprintSquad/Rapport/Rapport%20Projet%20Intégration%20(2)%20(1).docx#_Toc153442213)

[Figure 11: Use Case diagram 20](#_Toc153442214)

[Figure 12: System Sequence Diagram 21](#_Toc153442215)

[Figure 13: Domain Model Diagram 22](#_Toc153442216)

[Figure 14: Participant Class 23](#_Toc153442217)

[Figure 15: Sequence Diagram "Crud Products" 24](#_Toc153442218)

[Figure 16: Sequence Diagram "Crud Categories" 25](#_Toc153442219)

[Figure 17: Conception Class 26](#_Toc153442220)

[Figure 18: List Product 27](#_Toc153442221)

[Figure 19: Delete Product Confirmation 28](#_Toc153442222)

[Figure 20: Update Product Confirmation 29](#_Toc153442223)

[Figure 21: Home Page. 33](https://mohetn.sharepoint.com/sites/DSI312023-2024-eq3SprintSquad/Shared%20Documents/eq3%20SprintSquad/Rapport/Rapport%20Projet%20Intégration%20(2)%20(1).docx#_Toc153442224)

[Figure 22: Use case diagram 34](#_Toc153442225)

[Figure 23: Domaine model 35](#_Toc153442226)

[Figure 24: Class Diagram 36](#_Toc153442227)

[Figure 25: Sequence detailed diagram 37](#_Toc153442228)

[Figure 26: Use case diagram 41](#_Toc153442229)

[Figure 27: Domaine model 42](#_Toc153442230)

[Figure 28: Participant diagram for “authentication Use Case” 42](#_Toc153442231)

[Figure 29: participant diagram “update Profile Details Use Case” 43](#_Toc153442232)

[Figure 30: participant diagram "managed Registered User Use Case" 43](#_Toc153442233)

[Figure 31: participant diagram "Manage News Banner Use Case" 43](#_Toc153442234)

[Figure 32: participant diagram "View Order History Use Case" 43](#_Toc153442235)

[Figure 33: participant diagram "View Transaction's History Use Case" 43](#_Toc153442236)

[Figure 34: keycloack interface 44](#_Toc153442237)

[Figure 35: Continuouns integration 45](#_Toc153442238)

[Figure 36 : Delivery’s old process 46](#_Toc153442239)

[Figure 37 :CI/CD architecture 47](#_Toc153442240)

[Figure 38 : Github actions pipline 47](#_Toc153442241)

**List of Tables**

[Table 1: Product Backlog 12](#_Toc153308011)

[Table 2: Sprint 1 backlog 15](#_Toc153308012)

[Table 3: ToDo for amelioration 30](#_Toc153308013)

[Table 3: ToDo for amelioration 38](#_Toc153308014)

[Table 3: ToDo for amelioration 44](#_Toc153308015)

# General Introduction

During a university project based on the SCRUM method, our team collaborated to develop a web and mobile application that aimed to address the functionality gaps observed in existing apps. To effectively present our work, it is essential to provide a comprehensive overview, starting with a preliminary study that examines the problems identified in other existing applications. This study will include a description of their shortcomings, criticisms, and our proposed solutions (Chapter I).

Following the preliminary study, I will outline the product backlog, which will include user stories and various use cases specific to our application (Chapter II). This will help provide a clear understanding of the planned features and functionalities.

Moving on to the implementation phase, It will present the organization and details of Sprint 1, including the sprint backlog, sprint goals, and the actual implementation process (Chapter III). Similarly, It will provide information on Sprint 2, including its organization, sprint backlog, goals, and implementation (Chapter IV). Finally, It will cover Sprint 3, discussing its organization, sprint backlog, goals, and implementation (Chapter V).

By structuring the presentation in this manner, we can effectively showcase our project's initial study, product backlog, and the implementation details of each sprint. This approach will allow this report to give a comprehensive understanding of our project, from identifying problems in existing applications to delivering solutions through an iterative and incremental development process.

# Chapter 1: Preliminary study

## Introduction

In this chapter, we are going to present the different functionalities of our application, as well as its critics, the proposed solution and a detailed description of the final solution.

## Description of existing equipment

In the traditional method of selling and buying products, customers or suppliers typically need to physically visit a store or individual to purchase a product. They would have to engage in direct communication with the seller, and transactions would be conducted through documents such as purchase orders. Payments are often made in person, with money exchanged directly.

However, with the advent of technology and the rise of e-commerce, this traditional approach is evolving. Nowadays, customers and suppliers have alternative options that offer greater convenience and efficiency.

Online platforms and marketplaces enable customers to browse and purchase products from the comfort of their own homes, eliminating the need for physical visits. Transactions are conducted electronically, and payment methods include online payment gateways with credit cards. This allows for secure and convenient financial transactions without the need for physical cash exchange.

## Existences Criticisms

In the current system, to purchase products there is some criticisms like:

* **No availability:** customers need to contact the seller to arrange a meeting when they are available.
* **Wasting time and money:** They must then arrange transportation, and sometimes unforeseen issues may prevent them from completing the transaction, resulting in wasted time and money
* **Scamming and fraudulent activities:** There is no customer service available, which has led to a significant number of scams and fraudulent activities. Customers cannot feel secure because they must carry cash with them to make purchases, putting them at risk of theft.

## Proposed Solution

We are proposing our innovative application for buying and selling products. Our platform empowers users with an array of essential features, enabling seamless transactions and fostering a secure and trustworthy marketplace. With our application, sellers can effortlessly list their products for sale, while buyers can explore, purchase, and even add items to their Wishlist. We've implemented a robust and secure online payment system to ensure a smooth transaction process.

In addition to the core functionalities, we've integrated optional shipping capabilities, providing users with the choice of convenient delivery with tracking. To enhance user trust and reliability, we've introduced verified accounts in fact users must verify their account with their phone number to us our application, with help eliminating fake profiles, spam, and non-serious users from our platform. To enhance user trust and reliability, we've introduced verified accounts in fact users must verify their account with their phone number to us our application, with help eliminating fake profiles, spam, and non-serious users from our platform. Moreover, our application offers badges and a reward system, encouraging sellers to build strong relationships with their customers.

To further enhance user support, our platform now features a real-time chat system, allowing buyers and sellers to communicate seamlessly. Additionally, we've introduced a ticket system, ensuring that customers have a direct channel to resolve any issues and assert their rights. if the problem isn’t solved, buyers can transfer tickets to support team, we take user satisfaction seriously, and in line with this commitment, sellers are held to high standards. With our 'three strikes and you're out' policy, sellers are given three chances to maintain account integrity. After three strikes, their accounts will be subject to removal.

Should any issues or concerns arise, our platform includes a dedicated feature for users to contact either the admin or the seller for prompt assistance. Our application is built using Android with Kotlin, utilizing a microservices architecture with Node.js and Express (part of the MEAN stack technology), as well as Spring Boot, ensuring a high-performance and scalable platform for a seamless buying and selling experience.

## Adapted working method.

### SCRUM project management method

For the development of our projects, we are decided to choose SCRUM like a method of the project management with:

* Miss Afef Gafsi: the Scrum master which she is the responsible to help us to move forward independently, while constantly striving to improve.
* Developers: We are playing the developers roles that we are supposed to developing the different apps and architectures of it.
* Product Owner

### Done criteria

We are supposed that a user story is done, it should be terminated and work fluently.

### Sprints Time Box

We take as an estimation for every sprint 2 weeks.

## Working tools

### Frameworks used

During the development of our project, we are using a variety of Frameworks like:

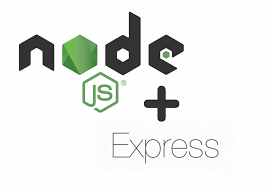


Figure 1: NodeJS and Express [1]

We are using NodeJS and Express for developing micro services, for the web version.



Figure 2: Android Native with Koltin [2]

To developing the mobile version of our application we are using the android native with kotlin as a programming language.



Figure 3: Spring boot logo [3]

We are also using Spring boot for the developing of the micro services for the web version.

### Software development

For the developing of our application, we are using a variety of software programs like:



Figure 4: VsCode logo [4]

For the development of NodeJS and Express, we are decided to use VsCode.



Figure 5: Android Studio logo [5]

For the development of the mobile version we decided to use android studio.

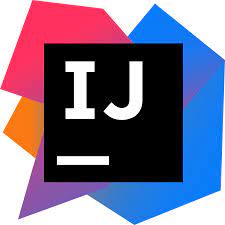


Figure 6: IntelliJ ide logo [6]

For the development for the micro services in spring boot we decided to use IntelliJ ide



Figure 7: Postman logo [7]

We are using Postman for testing the API of the backend which develops with NodeJS express and SpringBoot.



Figure 8: Github logo [8]

We are using Github for colloaborating and for coding as a team, means everyone have a code deploying in github and he can change whenever he wants.

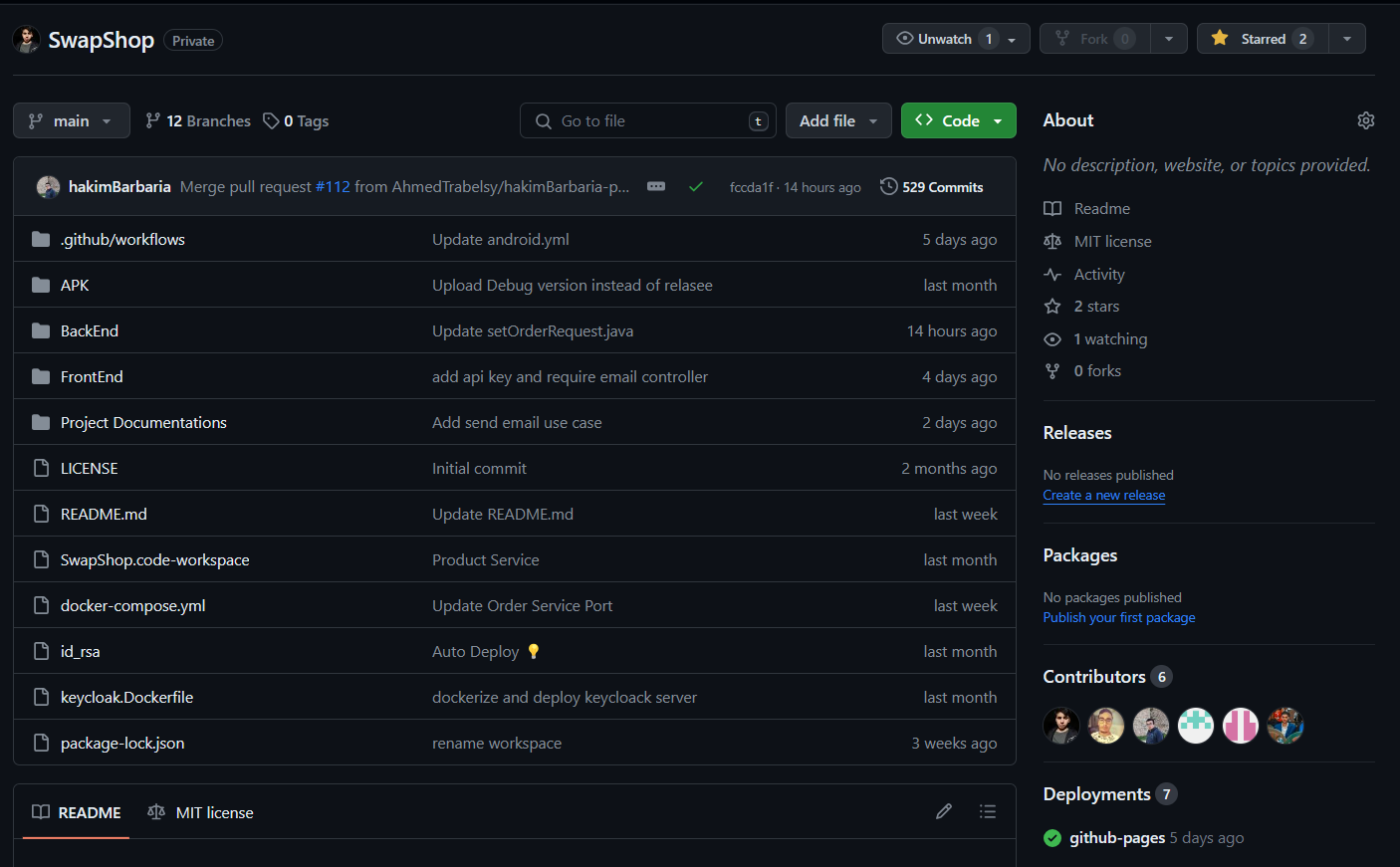
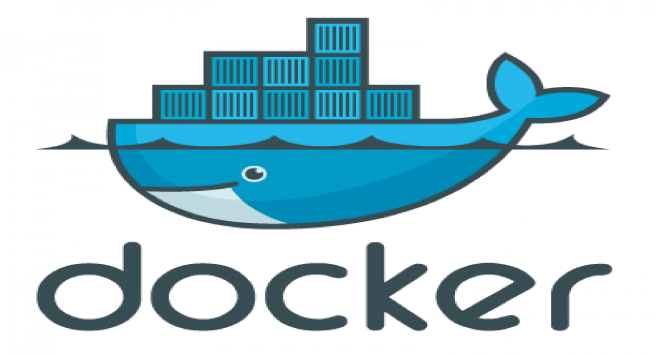


Figure 9: Github Repository



it is a set of platform-as-a-service products that use operating-system-level virtualization to deliver software in packages called containers. Containers are isolated from each other and bundle their own software, libraries, and configuration files, they can communicate with each other through well-defined channels.

## Conclusion

In this chapter we’ve presented a preliminary study of our application, we’ve presented the organization's need for this application as well as its benefits, we’ve also presented all the system's functionalities.

# Chapter 2: Product Backlog planning

## Introduction

In this chapter, we are going to present the different actors and their user’s stories, then we are going to present the product backlog.

## User’s identification

In this application we’re developing different functionalities for 3 actors:

A seller he has functionalities:

* Managing the products, he intends to sell.
* Reviewing annual activities and profits.
* Displaying featured badges on their profile to build trust with customers: badges are given automatically after replying for some conditions (example: sellers with receive a "**first product sold** "badge after they sell their first product)
* Keep in contact with the admin if there is a problem.
* Display a list of a buyer's past orders for reference.

A buyer has functionalities:

* Search for products based on a range of criteria.
* Add & manage products Wishlist.
* Ensuring Secure Purchases, Whether Online or In-Person.
* Evaluate the services provided by sellers
* Place an order.
* Monitor order history and shipment tracking.
* Maintain open communication with sellers or our support team through a chat or a ticket system to address any product, shipping, payment, or other concerns.
* Enhance the user experience with a 'Recently Viewed' list, providing users with easy retrieval of their past interactions.

And an admin has functionalities:

* View and manage all users
* Manage the categories of the products.
* Keep in contact with sellers and buyers through a chat/ticket system and give users strike or ban their accounts if they violate the policies.

## User Stories

## Application Flows

Figure 10: Application browsing flow

## Product Backlog

Table 1: Product Backlog

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N° | Description User story | Type | Priority | Complexity | Estimate | Acceptance criteria | Sprint |
| 1 | As a developer, I initialize the project with the necessary dependencies to properly manage the application. | TS | 1 | Low | 1 day | Functional project | 1  Initialization Project, category and product’s seller Crud |
| 2 | As an admin I would like to add a category | US | 1 | Medium | 2 days |  |
| 3 | As an admin I would like to delete a category | US | 1 | Low | 2 days |  |
| 4 | As an admin I would like to modify a category | US | 1 | Medium | 2days |  |
| 5 | As a seller I would like to add a product in my shop | US | 1 | Medium | 2 days | adding successfully |
| 6 | As a seller I would like to delete a product in my shop | US | 1 | Low | 2 days | deleting successfully |
| 7 | As a seller I would like to modify a product detail in my shop | US | 1 | Medium | 2 days | modifying  successfully |
| 8 | As a user I would like to see all the product in the application | US | 1 | Low | 1 day | Showing products successfully |  |
| 9 | As a customer I would like to check a product detail and it price | US | 2 | Low | 1 day | Showing products details successfully | 2  Crud’s products in customer’s  Wishlist and pass orders |
| 10 | As a customer I would like to search for a product using a various criteria | US | 2 | Medium | 2 days | Searching successfully |
| 11 | As a customer I would like to add a product in my wishlist to buy it later | US | 2 | Medium | 2 days | Adding successfully |
| 12 | As a customer I would like to delete a product in my wishlist | US | 2 | Low | 1 day | Deleting successfully |
| 13 | As a customer, I would like to pass an order to a product | US | 2 | Medium | 2 days | Orders passed successfully |
| 14 | As a customer, I would like to verficate the qualities of products before paying | US | 2 | Medium | 2 days | Verification successfully |
| 15 | As a customer, I would like to pay for the product online | US | 2 | Medium | 2 days | System of payment working successfully |
| 16 | As a customer, I would like to track my shipment | US | 2 | Medium | 2 days | Shipment tracked successfully |
|  | As a developer I developing a system of earnings badges to who’s have send 5 successful tracks | US | 3 | Medium | 1 day |  | 3  Authentication  Reviewing, badges system  And user’s statics |
| 17 | As a customer I would like to review the services of sellers | US | 3 | Medium | 1 day |  |
| 18 | As a customer I would like to add strikes if I don’t like a service of sellers | US | 3 | Medium | 1 day |  |
| 19 | As an admin, I would like to verificate the strikes after user’s add | US | 3 | Medium | 1 day |  |
| 20 | As an admin, I would ban sellers if he got 3 strikes | US | 3 | Medium | 1 day |  |
| 21 | As a seller, I want to showcase earned badges to build trust with buyers. | US | 3 | Medium | 1 day |  |
| 22 | As a user, I want to see badges on the profiles of other users or sellers | US | 3 | Medium | 1 day |  |
| 23 | As a user, I want to view the transaction history with real-time tracking data. | US | 3 | High | 2 days |  |
| 24 | As a seller I would like to check my profits in the year | US | 3 | High | 2 days |  |
| 25 | As a customer I would like to contact the seller or the admin if there is a problem | US | 3 | High | 3 days |  |
| 2 | As a user, I would like to authenticate to get an access to my space in the app | US | 1 | Medium | 4 days | Authentication  successfully |  |

In our project we got 3 sprints, the one of it releasing during 2 weeks:

* Sprint 1: Initialization Project, category and product’s seller Crud.
* Sprint 2: Crud’s products in customer’s Wishlist and pass orders.
* Sprint 3: Authentication Reviewing, badges system And user’s statics

## Conclusion

In this chapter, we’ve presented all the user stories related to our projects, also we’ve presented all the user’s identifications, and finally we’ve presented the product backlog classified by sprints.

# Chapter 3: Sprint 1

## Introduction:

In this chapter, we are going to present the sprint backlog, the implementation, and the different diagrams and conceptions.

## Sprint goals

* Adding a system to let the user to manage their products that he would buy.
* Adding a system to let the user to manage the categories of products

## Sprint backlog

Table 2: Sprint 1 backlog

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | User Story | Tasks | Owner | Estimation | Start | In progress | Done |
| 1 | As a developer, I initialize the project with the necessary dependencies to properly manage the application. | Install the developing environments | All members | 15 hours |  |  | X |
| Install the project and all their dependencies | 9 hours | X |  |  |
| Developing the eureka server service using spring boot |  |  |  |  |  |
| Developing the gatway server service using spring boot |  |  |  |  |  |
| 2 | As an admin I would like to add categories for products | Initialize the category management service using spring boot |  |  |  |  |  |
| Initialize the category management service using NodeJS |  |  |  |  |  |
| Adding the add categories functionality to category service using spring Boot |  |  |  |  |  |
| Adding the add categories functionality to category service using NodeJS | Farouk | 15 hours | X |  |  |
| Create a mobile user interface | Jasser | 5 hours | X |  |  |
| Creating an add category functionality in the category controller in kotlin | Hakim | 4 hours |  |  |  |
| 3 | As an admin I would like to delete categories | Adding the delete category functionality to category service using spring Boot |  |  |  |  |  |
| Adding the delete category functionality to category service using NodeJS |  |  |  |  |  |
| Create a delete button in the  mobile user interface | Haythem | 1/2 hour | X |  |  |
| Create a delete category functionality in the category controller in kotlin |  | 23 hours | X |  |  |
| 4 | As an admin I would like to modify a category | Adding the edit category functionality to category service using spring Boot | Farouk | 15 hours | X |  |  |
| Adding the edit category functionality to category service using NodeJS | Ahmed | 8 hours |  |  |  |
| Create a mobile user interface | Bayrem | 1 jour | X |  |  |
| Creating a modify category functionality in the category controller in kotlin |  |  |  |  |  |
| 5 | As a seller I would like to add a product in my shop | Initialize the products management service using spring boot | Hakim | 15 hours | X |  |  |
| Initialize the products management service using NodeJS | Bayrem | 9 hours | X |  |  |
| Adding the add products functionality to products service using spring Boot | Farouk |  |  |  |  |
| Adding the add products functionality to products service using NodeJS |  |  |  |  |  |
| Create a mobile user interface |  |  |  |  |  |
| Creating an add products functionality in the products controller in kotlin |  |  |  |  |  |
| 6 | As a seller I would like to delete a product in my shop | Adding the delete product functionality to product service using spring Boot | Jasser | 1 hour | X |  |  |
| Adding the delete product functionality to product service using spring Boot | 1 hour | X |  |  |
| Create a delete button in the  mobile user interface | 22 hours | X |  |  |
| Create a delete product functionality in the product controller in kotlin |  |  |  |  |  |
| 7 | As a seller I would like to modify a product detail in my shop | Adding the edit product functionality to product service using spring Boot | Hakim | 15 hours | X |  |  |
| Adding the edit product functionality to product service using NodeJS | Bayrem | 8 hours | X |  |  |
| Create a mobile user interface | Ahmed | 1 day | X |  |  |
| Creating an add products functionality in the product controller in kotlin |  |  |  |  |  |
| 8 | As a user I would like to see all the product in the application | Adding the show products functionality to product service using spring Boot |  |  |  |  |  |
| Adding the show products functionality to product service using NodeJS |  |  |  |  |  |
| Create a mobile user interface |  |  |  |  |  |
| Creating an show all products functionality in the product controller in kotlin |  |  |  |  |  |

## Sprint 1 implementation

### System description phase

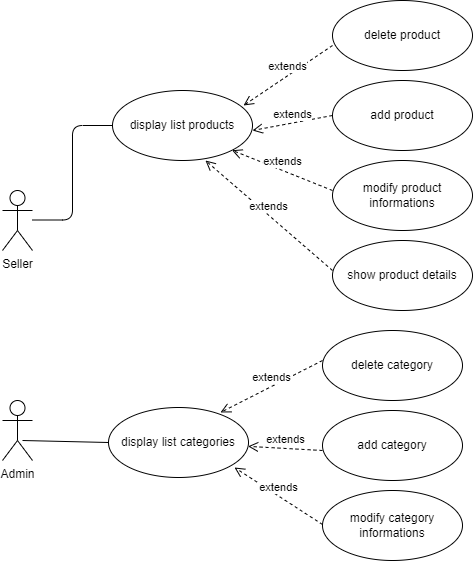


Figure 11: Use Case diagram

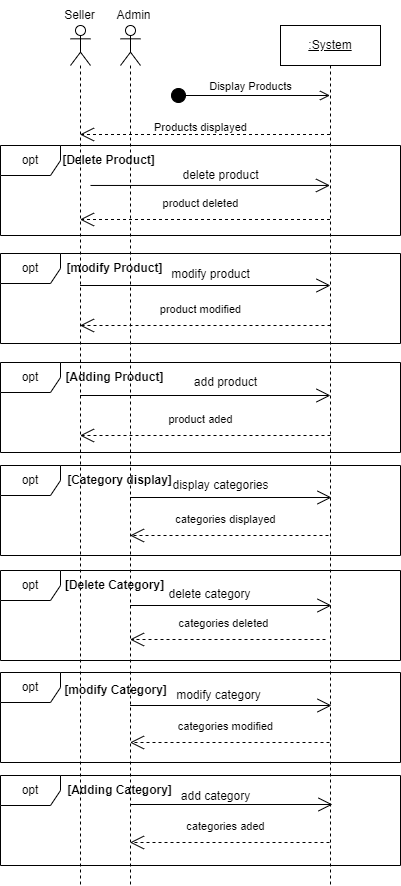


Figure 12: System Sequence Diagram

### b Analyze

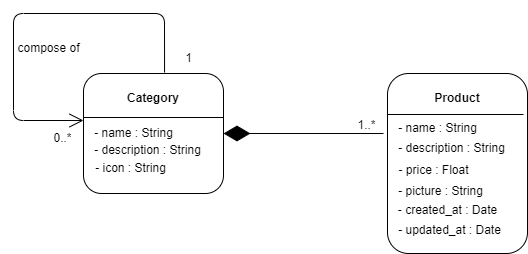


Figure 13: Domain Model Diagram

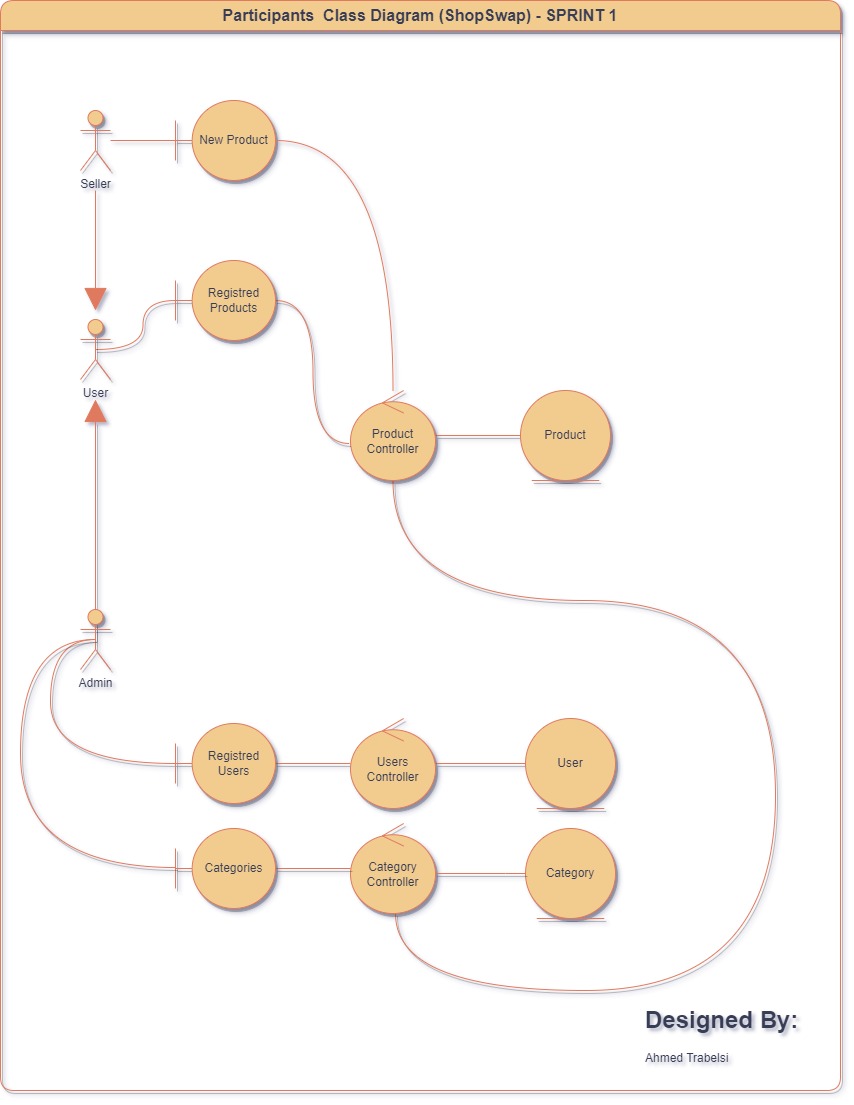


Figure 14: Participant Class

### Conception phase

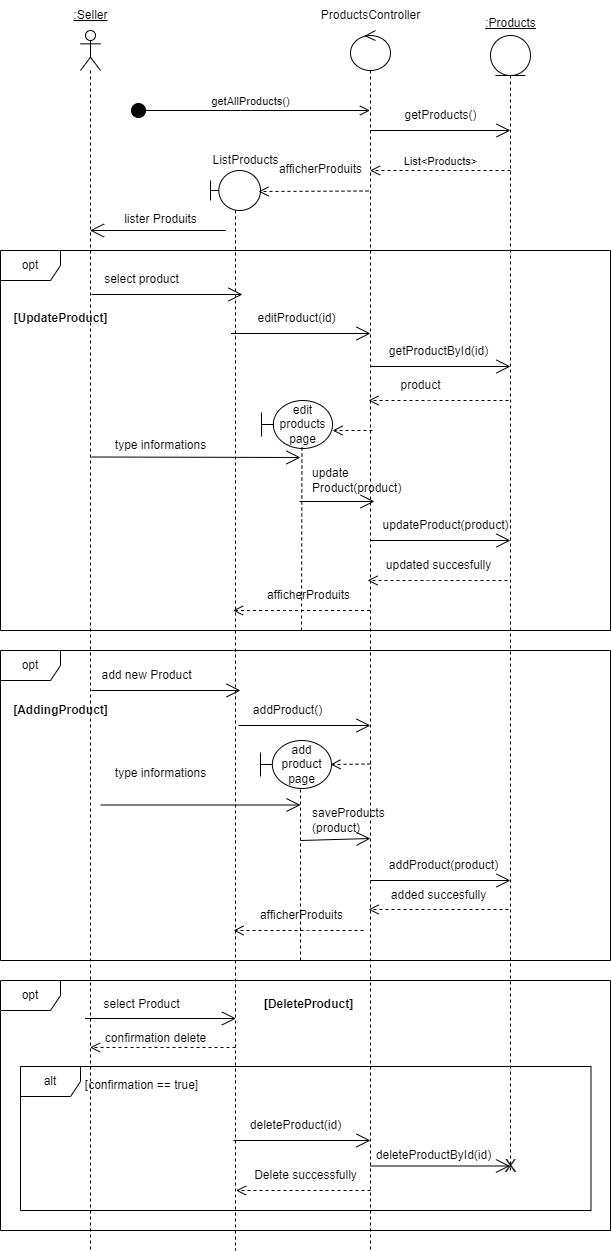


Figure 15: Sequence Diagram "Crud Products"

Une image contenant capture d’écran, noir, noir et blanc, obscurité

Description générée automatiquement

Figure 16: Sequence Diagram "Crud Categories"

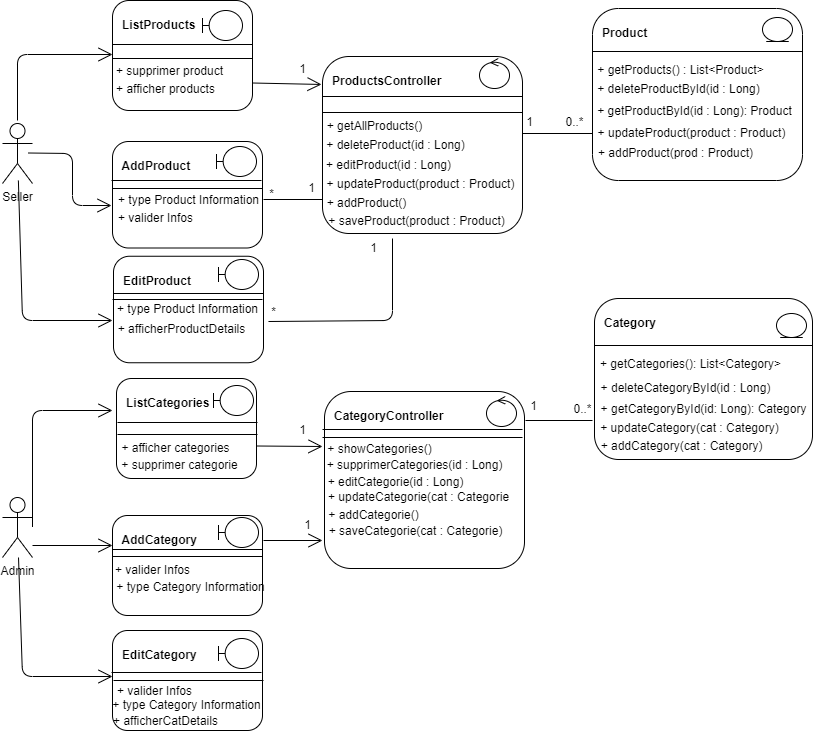


Figure 17: Conception Class

### Interfaces Models

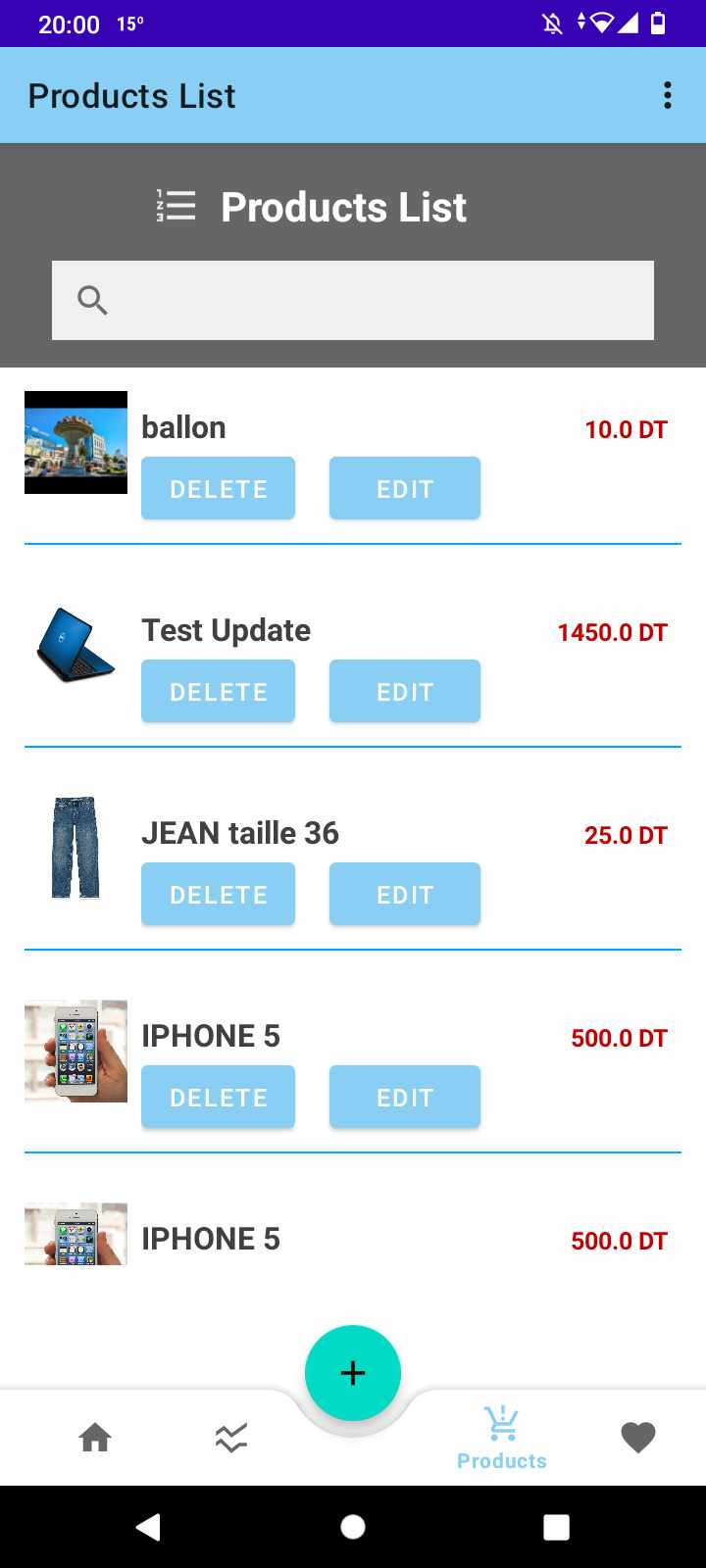


Figure 18: List Product

This interface represent the products list in our application, the seller here can update or delete any product with simple click on button, he can also filter by name also when he click in a product he can see the details.

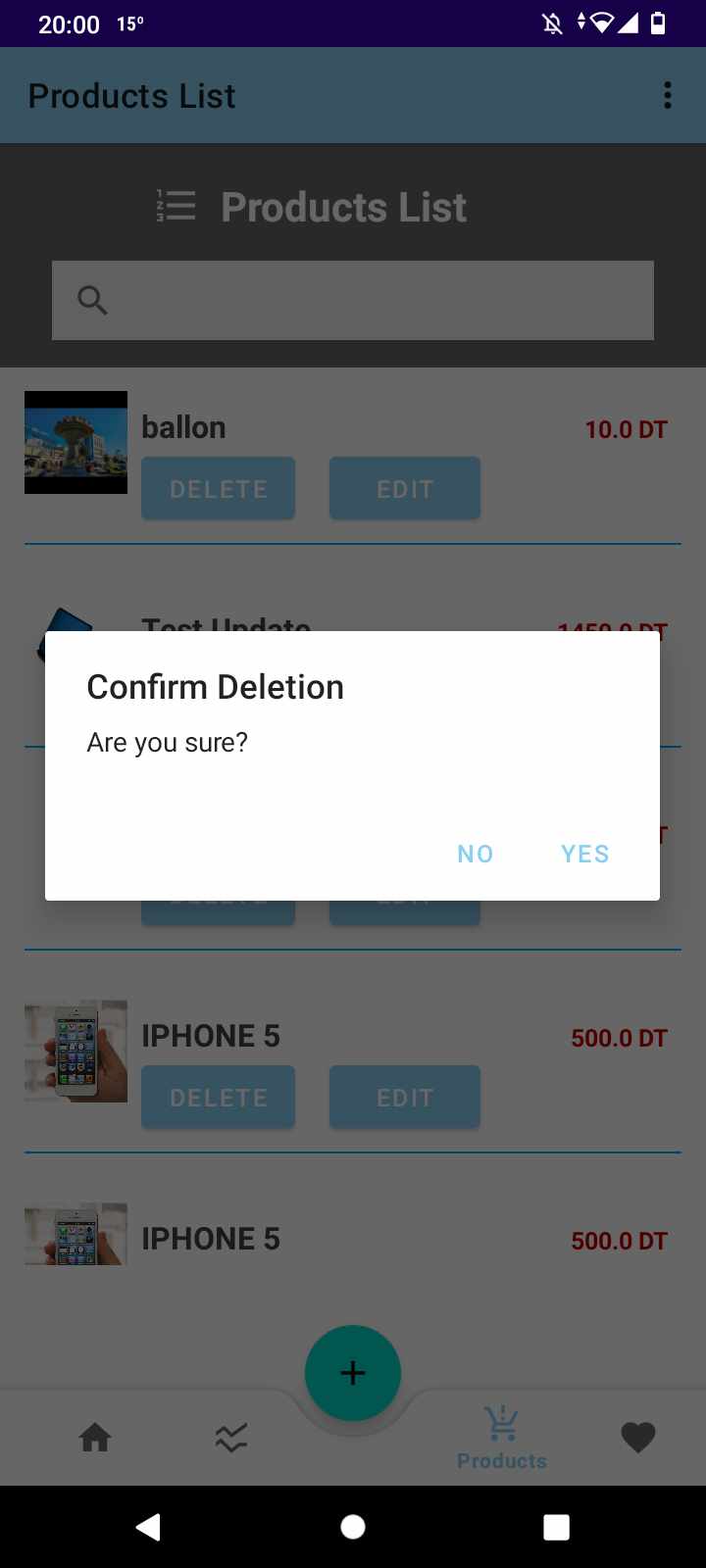


Figure 19: Delete Product Confirmation

By clicking in the delete button in the product list after selecting a product for delete an alert dialog box appear to let the seller choose and confirm the deletion.

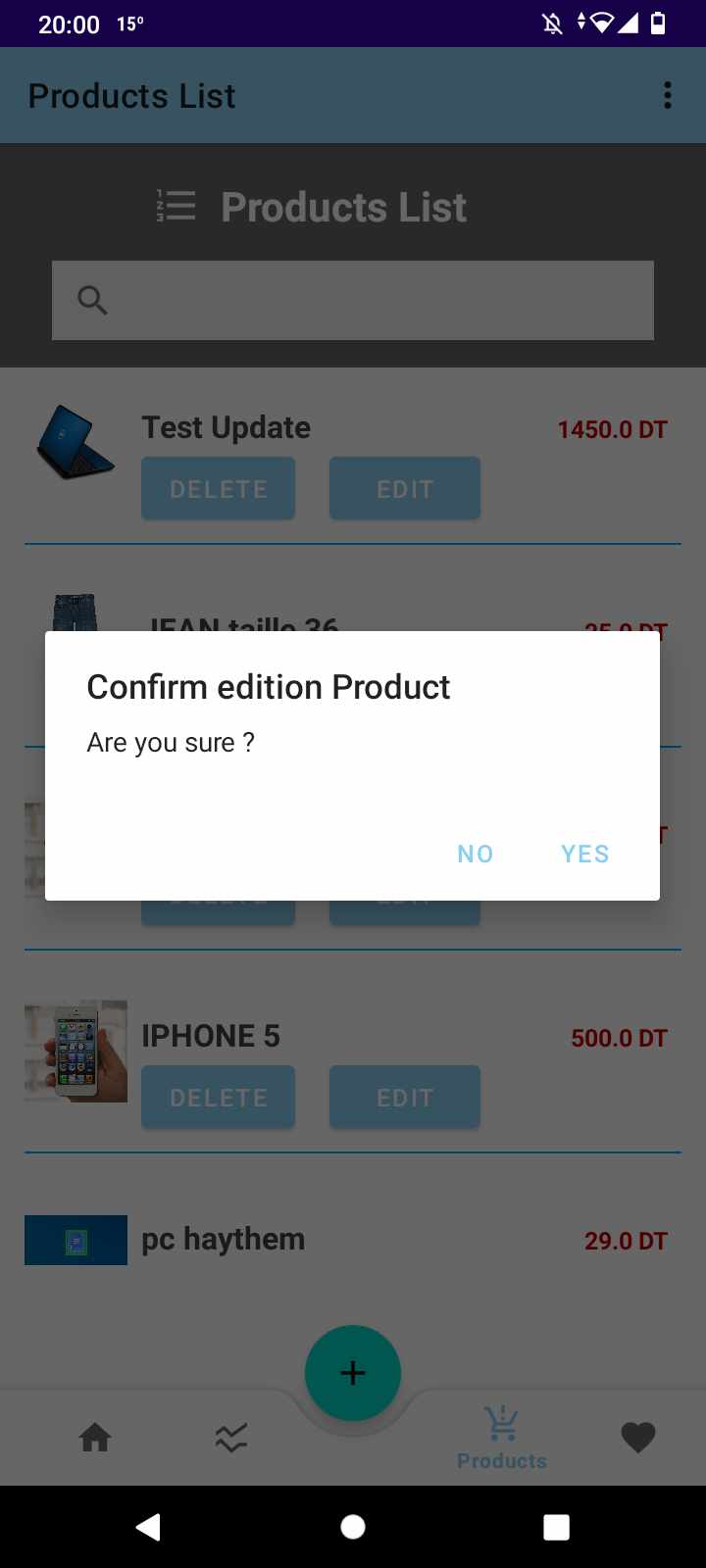


Figure 20: Update Product Confirmation

By clicking in the update button after selecting a product an alert dialog box appear to let the seller choose and confirm the edition of the product’s information.

## Sprint retrospective

As a scrum teams, we are planned this meeting to discuss about the problems during this sprint, what’s going well and what we can do to ameliorate.

Table 3: ToDo for amelioration

|  |  |
| --- | --- |
| What’s going well | Doing to ameliorate |
| Technical developpement | Time estimation for tasks |
|  | Affecting tasks to members |

## Conclusion

In this chapter we’ve presented the sprint 1, means we’ve presented the backlog, the sprint objective and the implementations contains the different UML diagram in the MVC structure, and we’ve also presented all the interfaces in mobile.

# Chapter 4: Sprint 2

## Introduction

In this chapter, we are going to present the sprint 2 backlog, the implementation, and the different diagrams and conceptions also the interfaces.

## Sprint goals

* Adding a system to add a product to a user wishlist from the home page.
* Adding a system to delete a wish Item from the user’s wishlist.
* Adding a system to let the user pass an order to checkout a product from the wish List “payment”.
* Adding a track shipment after the checkout of the wish Item.

## Sprint Backlog

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **User Story** | **Tasks** | **Owner** | **Estimation** | **Start** | **In progress** | **Done** |
| **1** | As a customer I would like to check a product detail and it price | Create a user interfece | **Hakim** | 15 hours |  |  | X |
| Create an order Button in the product details | **Bayrem** | 9 hours |  |  |  |
| Create an order Product Controller | **Ahmed** | 9 hours |  |  |  |
| **2** | As a customer I would like to search for a product using a various criteria | Create a SearchView | **Jasser** | 9 hours |  |  | X |
| Craeate a SearchView Cotroller | **Hakim** | 15 Hours |  |  |  |
| **3** | As a customer I would like to add a product in my wishlist to buy it later | Create a WhishLists interface | **Jasser** | 1 Day |  |  | X |
| **4** | As a customer I would like to delete a product in my wishlist | Create a delete button in my WishList | **Haythem** | 9 hours |  |  | X |
| Create a delete product  Controller | **Hakim** | 9 Hours |  |  |  |
| **5** | As a customer, I would like to pass an order to a product | Create a shippemet interface | **Jasser** | 1 Day |  |  | X |
| **6** | As a customer, I would like to verficate the qualities of products before paying |  | **Bayrem** | 9 hours |  |  | X |
| **7** | As a customer, I would like to pay for the product online | Create a payment Method in the shippement Interface | **Farouk** | 1 Day |  |  | X |
| Create a payment controller | **Haythem** | 1 Day |  |  |  |
| **8** | As a customer, I would like to track my shipment | Create a track shipement interface | **Ahmed** | 9 hours |  |  | X |

## Sprint implementation:

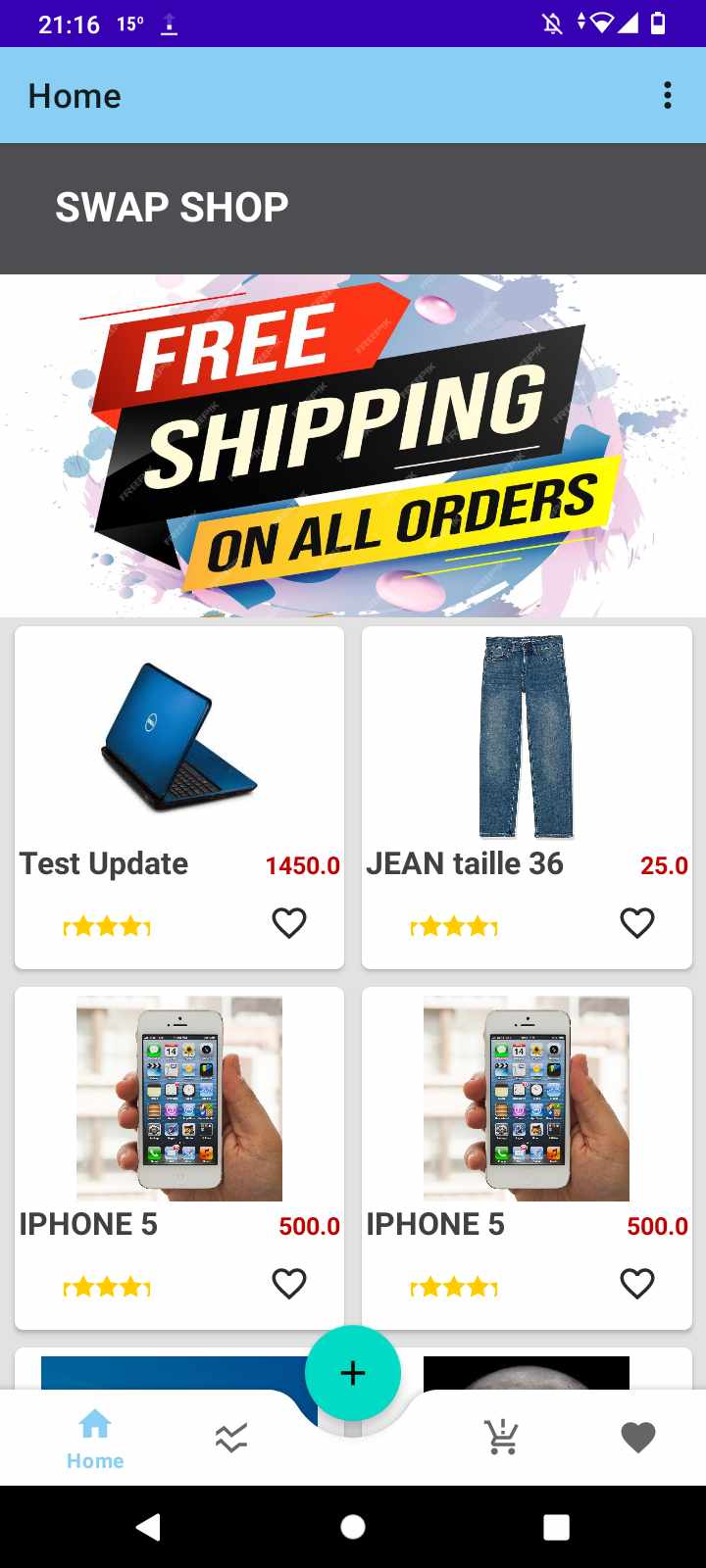


Figure 21: Home Page.

### System decription phase

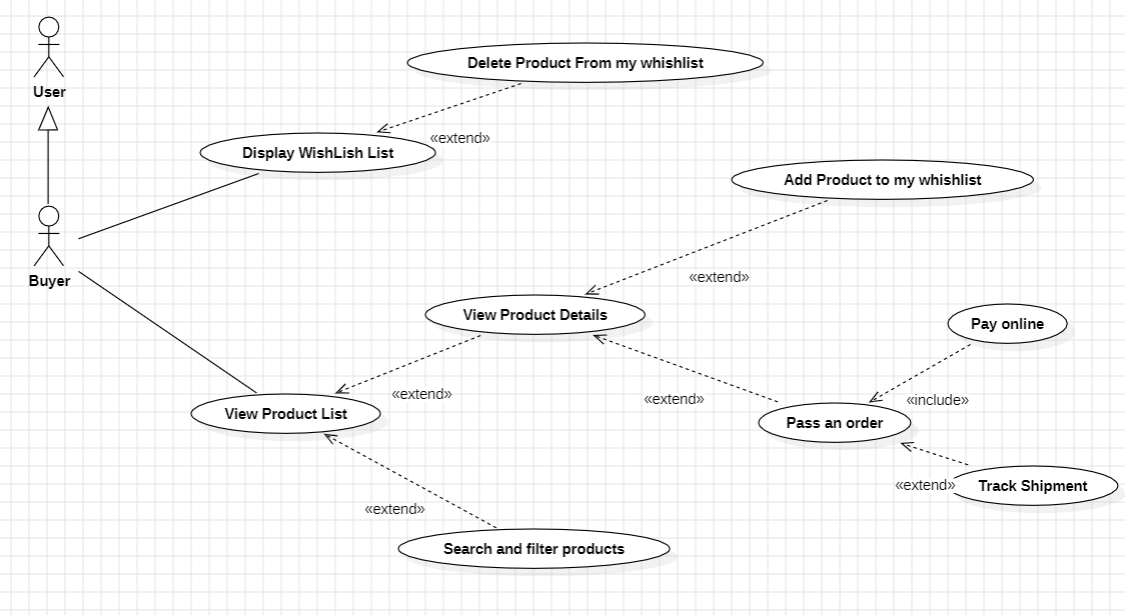


Figure 22: Use case diagram

### b Analyze

Une image contenant texte, capture d’écran, Parallèle, diagramme

Description générée automatiquement

Figure 23: Domaine model

### c Conception phase

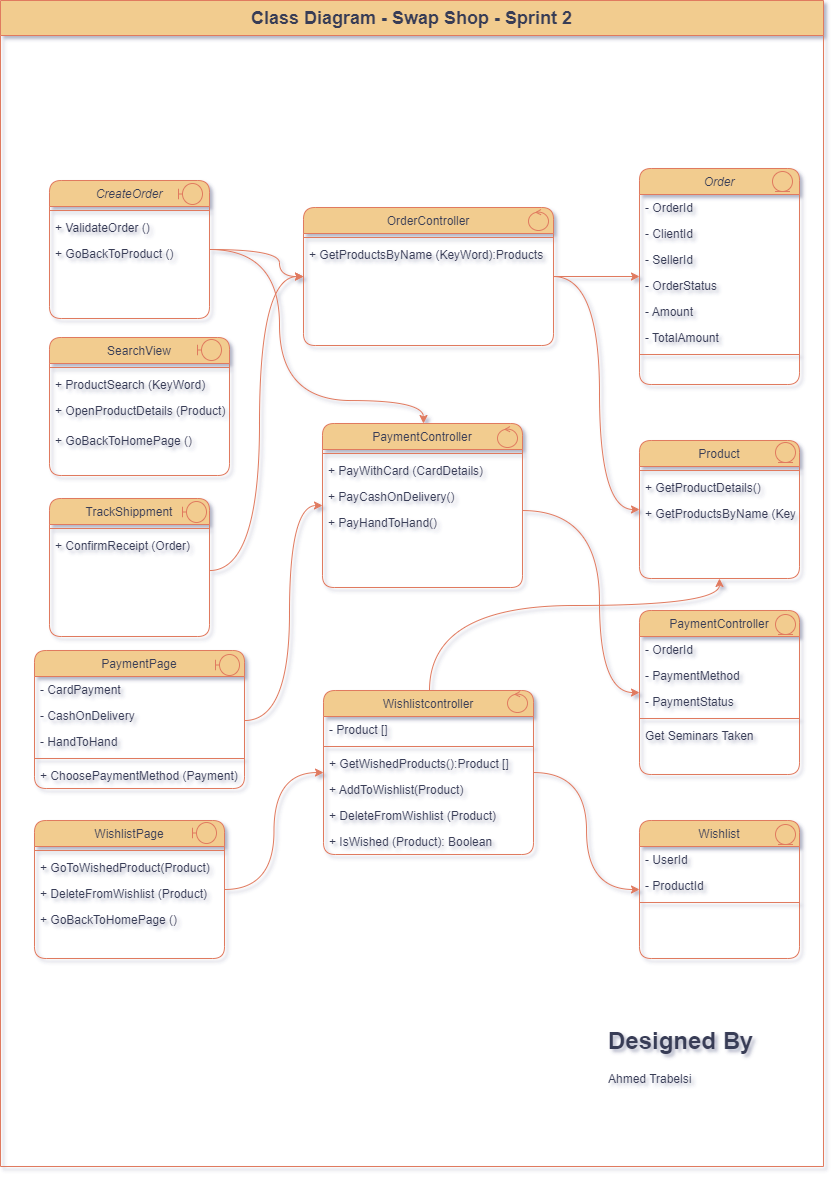


Figure 24: Class Diagram

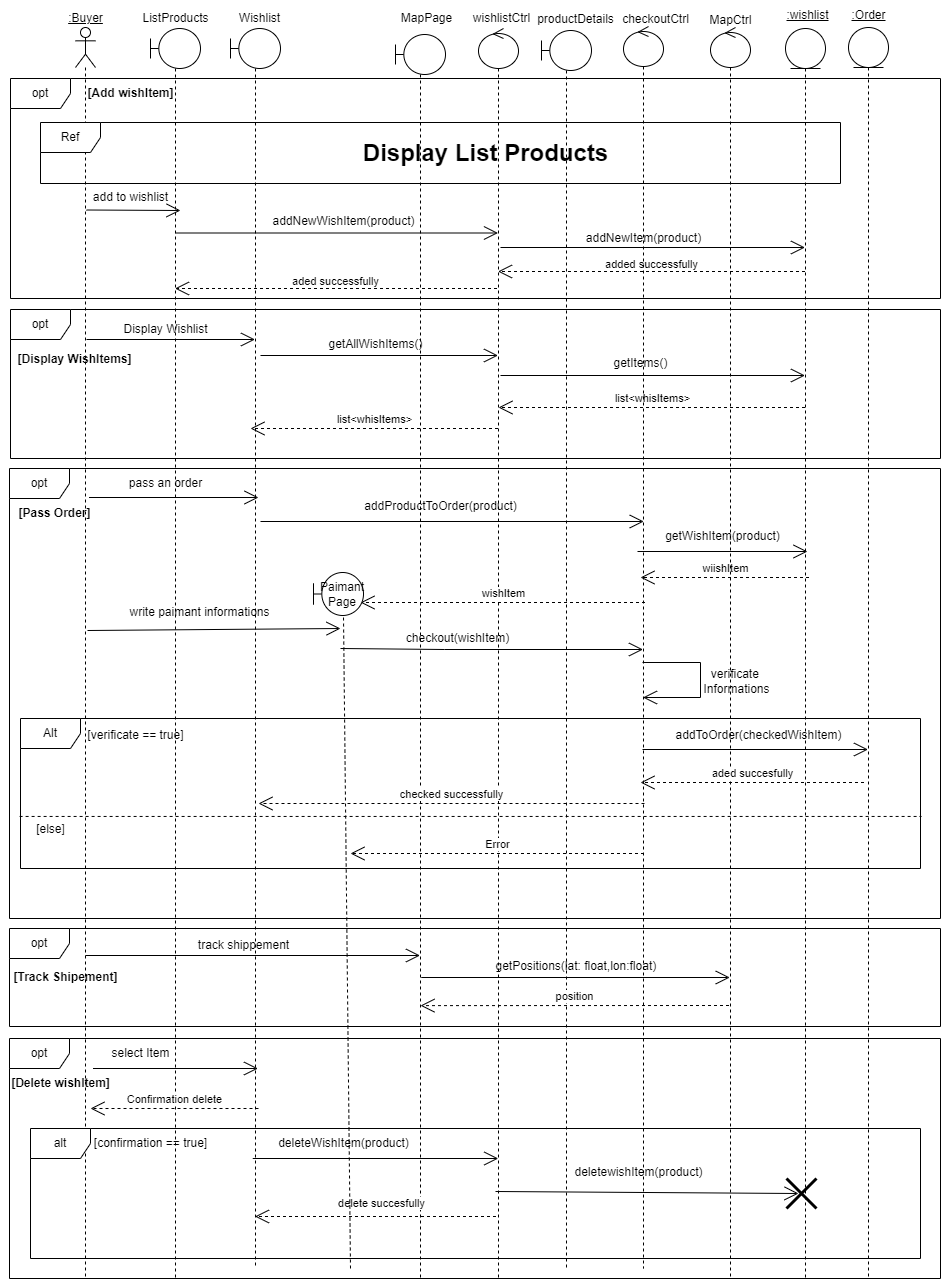


Figure 25: Sequence detailed diagram

### d Interfaces Models

## Sprint Retrospective

As a scrum teams, we are planned this meeting to discuss about the problems during this sprint, what’s going well and what we can do to ameliorate.

Table 4: ToDo for amelioration

|  |  |
| --- | --- |
| What’s going well | Doing to ameliorate |
| Technical developpement | Time estimation for tasks |
| Affecting tasks to members |  |

## Conclusion

In this chapter we’ve presented the sprint 2, means we’ve presented the backlog, the sprint objective and the implementations contains the different UML diagram in the MVC structure, and we’ve also presented all the interfaces in mobile and in the end we are presented the sprint retrospective that we fixed as a team the positive point and things for ameliorating.

# Chapter 5: Sprint 3

## Introduction

In this chapter, we are going to present the sprint 3 backlog, the implementation, and the different diagrams and conceptions also the interfaces.

## Sprint goals:

* Strengthen authentication protocols for enhanced user security.
* Introduce a badges system to recognize user achievements and boost engagement.
* Implement a user statistics feature for valuable insights into user behavior.
* Develop a comprehensive notification system to keep users informed and engaged with platform activities.

## Sprint backlog:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **User Story** | **Tasks** | **Owner** | **Estimation** | **Start** | **In progress** | **Done** |
| **1** | As a user, I would like to authenticate to get an access to my space in the app | Create a Sign in interfece in the mobile version | **Hakim** | 15 hours |  |  | X |
| Create an sign up  interface | **Bayrem** | 9 hours |  |  | X |
| Create an authentication Controller | **Ahmed** | 9 hours |  |  | X |
| **2** | As a seller I would like to check my statistics in the year | Create a Dashboard interface | **Jasser** | 9 hours |  |  | X |
| Implement the dashboard interface with product and order Controller | **Hakim** | 15 Hours |  |  | X |
| **3** | As a customer I would like Receive notifications | Create a WhishLists interface | **Jasser** | 1 Day |  |  | X |
| **4** | As a customer I would like Receive notifications | Create a notification interface | **Haythem** | 9 hours |  |  | X |
| Create a notification controller | **Hakim** | 9 Hours |  |  | X |
| **5** | As a customer, I would like see my lists of orders | Create a order list product | **Farouk** | 1 Day |  |  | X |
| Implement the order Controller | **Haythem** | 1 Day |  |  | X |
| **6** | As a customer after I sign in i receive a mail notification | Create the email interface  Create a email controller | **Ahmed**  **Ahmed** | 1 day  9 hours |  |  | X  X |

## Sprint implementation

### System description phase

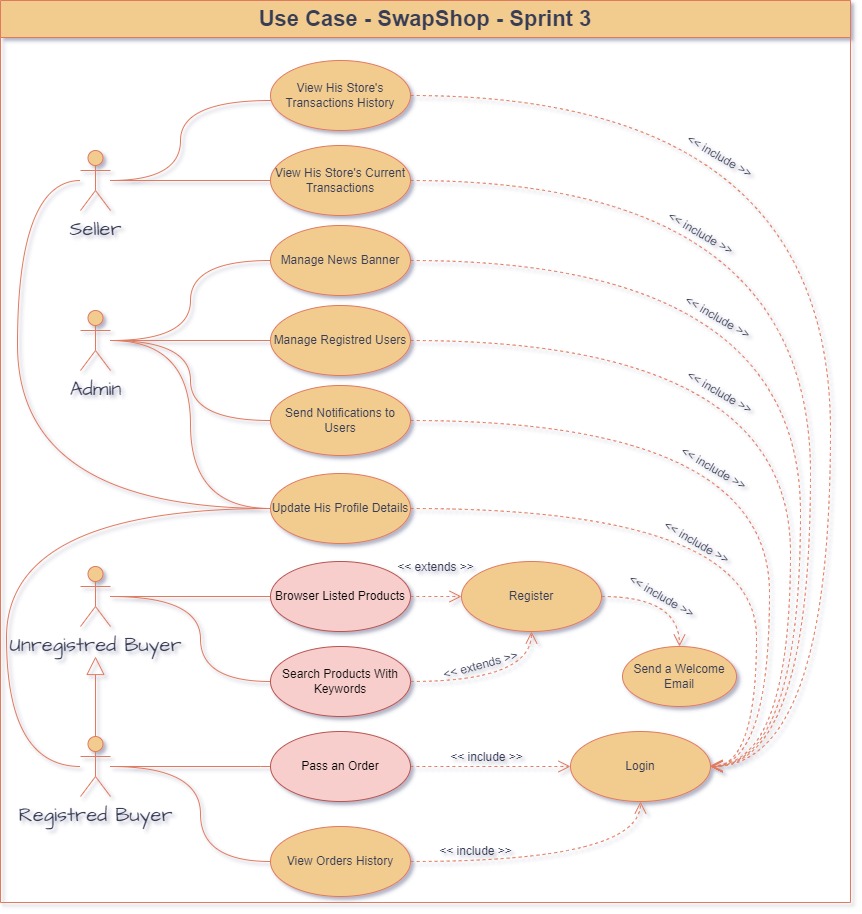


Figure 26: Use case diagram

### Analyze

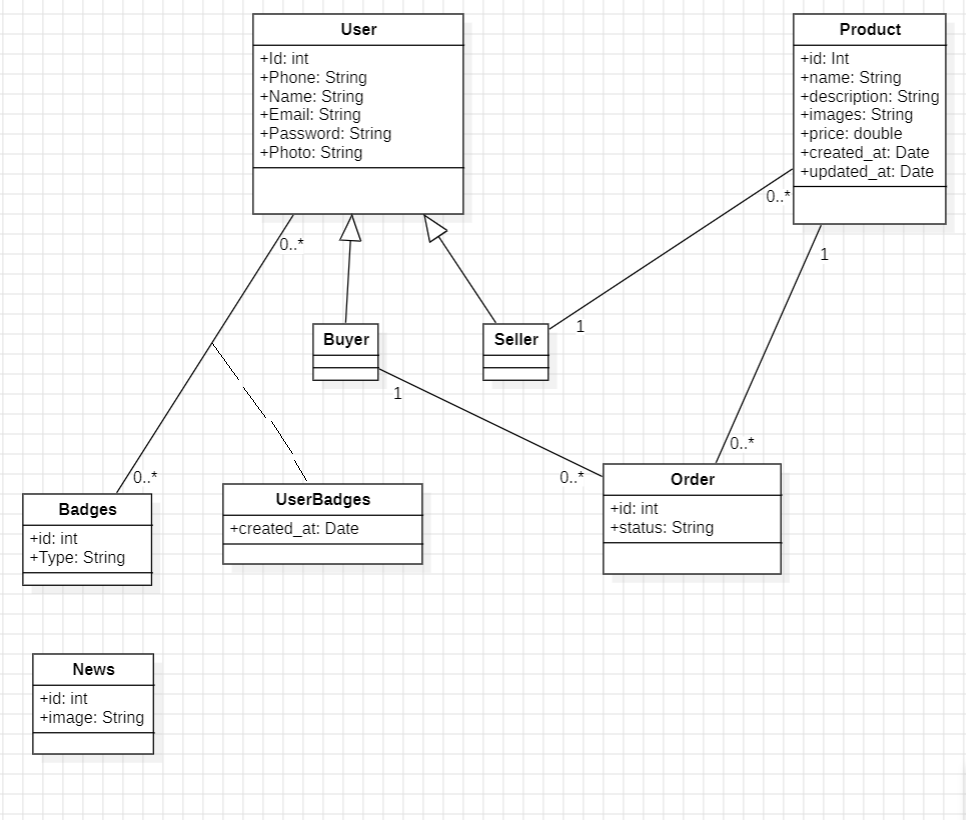


Figure 27: Domaine model

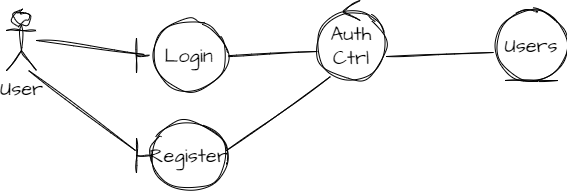


Figure 28: Participant diagram for “authentication Use Case”

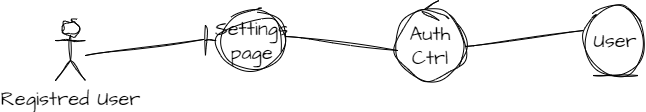


Figure 29: participant diagram “update Profile Details Use Case”

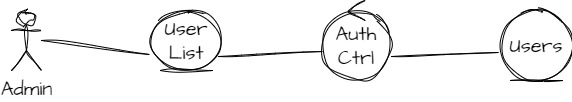


Figure 30: participant diagram "managed Registered User Use Case"



Figure 31: participant diagram "Manage News Banner Use Case"



Figure 32: participant diagram "View Order History Use Case"

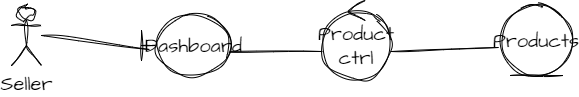


Figure 33: participant diagram "View Transaction's History Use Case"

## Sprint retrospective

As a scrum teams, we are planned this meeting to discuss about the problems during this sprint, what’s going well and what we can do to ameliorate.

Table 5: ToDo for amelioration

|  |  |
| --- | --- |
| What’s going well | Doing to ameliorate |
| Technical developpement |  |
| Time estimation for tasks |  |

## Keyclock security implementation

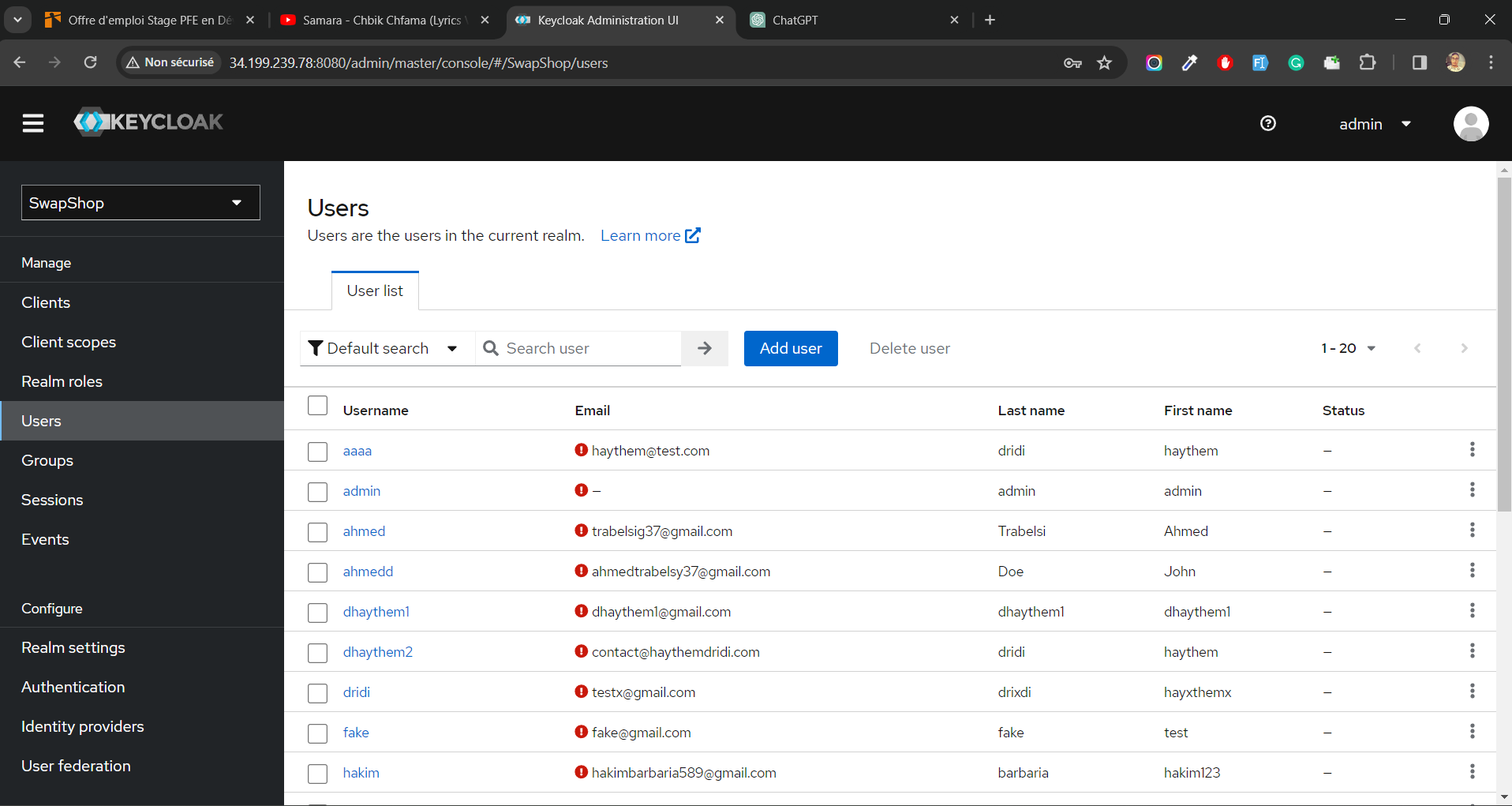


Figure 34: keycloack interface

**Secured authentication service:** Keyloack support secure authentication methods with o Auth2 protocol.  
**Easy User Management:** Keycloak provides a centralized user management system, simplifying the administration of user accounts.  
  
**Easy Role Management** : Users can be assigned roles, and access permissions can be granted based on these roles, enhancing security and manageability.

**High Availability**: It offers features for creating high-availability configurations, ensuring continuous service availability

## Implementation of a CI/CD pipeline based on docker

1. **CI : Continuous Integration Continuous integration**

CI : Continuous Integration Continuous integration : is defined as a set of practices used to ensure that each change in the source code does not produce regression problems in the under development application. The main goal of this approach is to anticipate and quickly identify bugs early before the software is put into production. This makes it possible to gain in reactivity to face the various problems which can be present in the various phases of the project In order to detect these problems, it is necessary to go through several steps, which are shown in the following figure :

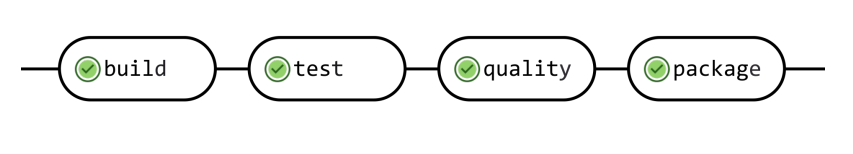


Figure 35: Continuouns integration

1. **CD : Continuous Delivery**

CD : Continuous Delivery

Continuous Delivery refers to regularly releasing our application and automating the delivery process. By doing so, we increase lower risks of traditional release cycles, and speed up the feedback process. Going through the “old-school” release methodology, a release is sent out after a specific period of time, say six months. The release must first be packaged, tested, and the necessary infrastructure must be installed or updated before it can be deployed on the server.

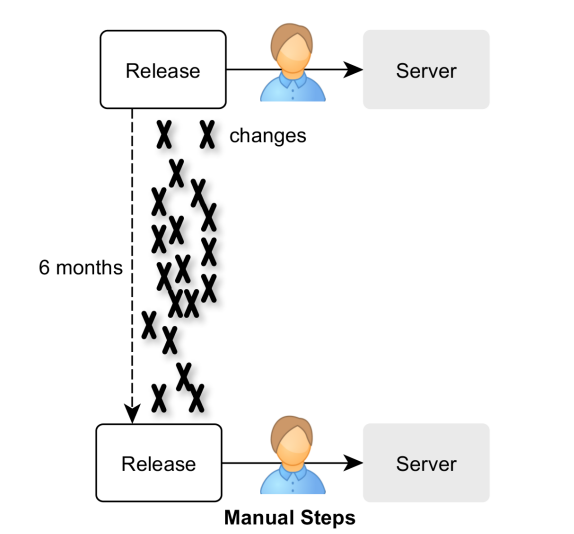


Figure 36 : Delivery’s old process

In this case, the release process is carried out rarely. As a result, we have little experience releasing. Errors can occur more frequently.

In addition to that, There are many manual tasks involved in the release process, including shutdown, infrastructure setup and updating, deployment, restarting, and manual testing

So in order to eliminate all these problems, this is where continuous delivery interferes : We reduce the risks of releasing by releasing more often.

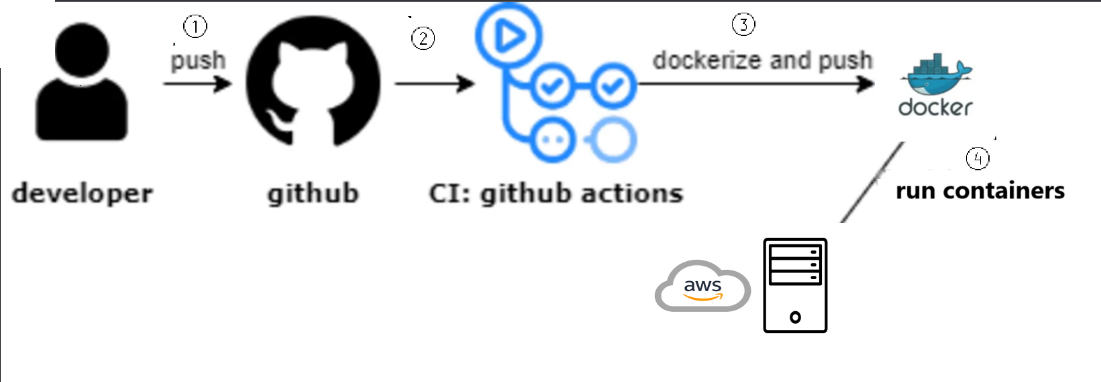


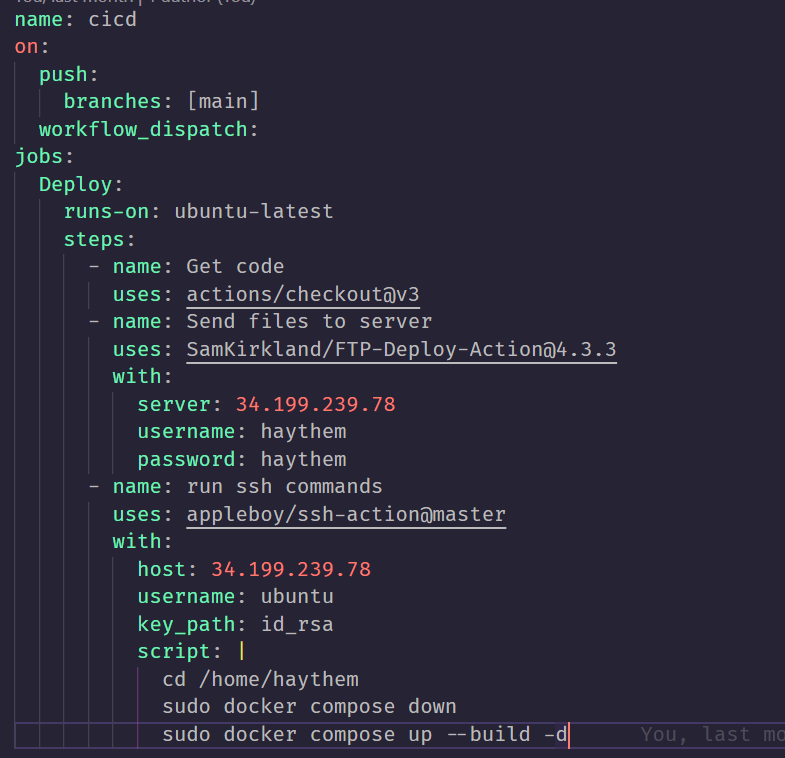
Figure 37 :CI/CD architecture

Figure 38 : Github actions pipline

GitHub Actions, this pipeline allows us, with each push to the main branch, to connect to the servers, send the new changes to the servers, and initiate the Docker compose build to create Docker images and launch the containers. This pipeline ensures that all changes made are deployed instantly on the servers and displayed to users.

### Implementation :

## Conclusion

In this chapter we’ve presented the sprint 3, means we’ve presented the backlog, the sprint objective and the implementations contains the different UML diagram in the MVC structure, and we’ve also presented all the interfaces in mobile and in the end we are presented the sprint retrospective that we fixed as a team the positive point and things for ameliorating.

# General Conclusion

In summary, the culmination of these efforts represents a significant stride forward in enhancing our platform. The successful implementation of features like CRUD operations, seamless order processing, and real-time shipment tracking reflects a commitment to improving the user experience. The introduction of security measures, badges, user statistics, and a comprehensive notification system further demonstrates our dedication to innovation and user engagement.

Looking ahead, the groundwork laid during these presentations sets the stage for continuous growth and refinement. The collaborative spirit of the team positions us for ongoing success, and we anticipate a future marked by even greater advancements, ensuring our platform remains at the forefront of user satisfaction and technological excellence.

# Bibliography

|  |  |
| --- | --- |
| [1] | [En ligne]. Available: https://web-and-mobile-development.medium.com/what-are-the-prime-differences-between-node-js-and-express-js-b560b19b8b33. |
| [2] | [En ligne]. Available: https://www.freecodecamp.org/news/learn-how-to-develop-native-android-apps-with-kotlin-full-tutorial/. |
| [3] | [En ligne]. Available: https://spring.io/trademarks/. |
| [4] | [En ligne]. Available: https://en.m.wikipedia.org/wiki/File:Visual\_Studio\_Code\_1.35\_icon.svg. |
| [5] | [En ligne]. Available: https://techcrunch.com/2017/02/19/why-is-android-studio-still-such-a-gruesome-embarrassment/. |
| [6] | [En ligne]. Available: https://fr.wikipedia.org/wiki/IntelliJ\_IDEA. |
| [7] | [En ligne]. Available: https://www.postman.com/company/press-media/. |
| [8] | [En ligne]. Available: https://1000logos.net/github-logo/. |