burkina institute of technology
Educating a New Generation of Leaders

GICOME TECHNOLOGIES

Ministry of Higher Education, Research and Innovation (MESRSI)

Thesis for the award of the Bachelor of Science in computer Science Option: Programming

Topic: Design of a mobile application for ordering computer consumables (EasyOrder) on behalf of the company GICOME TECHNOLOGIES

Presented and publicly defended by
Tarbwenda Nafissatou YAMEOGO

Supervised by: Mohamadou KONATE
Teacher at Burkina Institute of Technology

Jury:
President:
Members: ……
……..

Promotion: [2020/2023]

Dedication

I would like

To dedicate this memoir

To my dear mother DIANE ALINE

Who has been a source of inspiration, support and encouragement

throughout my academic career.

May God bless her and grant her a long life so that she can reap the

Fructs of her suffering and sacrifice.

She's more than a mother to me, she's also like a father to me.

Mom, thank you for your bravery.

I love you.

Aknowlegments

I would like to express my sincere thanks to all those who contributed in any way to the production of this memoir.

First of all, I'd like to thank the founder of BIT and her colleagues for giving me the opportunity to benefit from quality teaching.

To my co-director Mr. Mohamadou KONATE, I thank him from the bottom of my heart for the commitment he made to me. His rigorous supervision enabled me to carefully write this dissertation. His suggestions were extremely helpful in the completion of this work. I am deeply grateful to you, Mr KONATE.

I am extremely grateful to my supervisors and teachers, whose expert advice, expertise and availability greatly facilitated my research work. Their constant support, valuable suggestions and encouragement have been invaluable throughout this process.

I'd like to thank my family and friends for their unconditional support. Their confidence in me, constant encouragement and understanding in times of stress have been essential in keeping me motivated and focused.

In short, I'm deeply grateful to each and every one of the people who have been at my side throughout this academic adventure. Their support and commitment have been invaluable, and without them, this thesis would not have been possible.

Thank you from the bottom of my heart.

YAMEOGO Tarbwenda Nafissatou

Abstract

This document presents the design of an ordering application for Gicome Technologies. The app will allow users to place orders for computer consumables and have them delivered as needed. The app will be designed for use on iOS and Android devices. The app will be divided into three main sections: the home screen, the product catalog and the shopping cart. The home screen will display a list of available products from the Gicome company. Users can browse the product catalog by category, or search for specific products. Once a user has found a product they wish to order, they can add it to their basket. The basket will keep track of all the products the user has added to their order. When the user is ready to checkout, they will be able to enter their shipping and billing information. The application will be built using a user-centered design approach. This means that the application will be designed with the user in mind, and that the user's needs and desires will be taken into account throughout the design process. The application will be tested on a variety of devices to ensure that it works properly. The app will also be tested with a group of users to get feedback on the app's design and usability.

Keywords :

- Computers consumables
- Catalog
- Basket
- Approach
- Usability

Résumé

Ce document présente la conception d'une application de commande pour l'entreprise Gicome Technologies. L'application permettra aux utilisateurs de passer des commandes de consommables informatiques et de se faire livrer au besoin. L'application sera conçue pour être utilisée sur les appareils iOS et Android. L'application sera divisée en trois sections principales : l'écran d'accueil, le catalogue de produits et le panier. L'écran d'accueil affichera une liste des produits disponibles de l'entreprise Gicome. Les utilisateurs pourront parcourir le catalogue de produits par catégorie ou rechercher des produits spécifiques. Une fois qu'un utilisateur a trouvé un produit qu'il souhaite commander, il peut l'ajouter à son panier. Le panier gardera une trace de tous les produits que l'utilisateur a ajoutés à sa commande. Lorsque l'utilisateur est prêt à passer à la caisse, il pourra entrer ses informations d'expédition et de facturation. L'application sera conçue selon une approche de conception centrée sur l'utilisateur. Cela signifie que l'application sera conçue en pensant à l'utilisateur et que les besoins et les désirs de l'utilisateur seront pris en compte tout au long du processus de conception. L'application sera testée sur une variété d'appareils pour s'assurer qu'elle fonctionne correctement. L'application sera également testée avec un groupe d'utilisateurs pour obtenir des commentaires sur la conception et la convivialité de l'application.

Mots clés :

- Consommables informatiques
- Catalogue
- Panier
- Approche
- Convivialité

Table of contents

List of acronyms

Acronyms	Meanings
BIT	Burkina Institute of Technology
BF	Burkina Faso
UML	Unified Modeling Language
UP	Unified Process
2TUP	Two Track Unified Process
IT	Information Technology
IOS	Iphone Operating system
DBMS	Database Management System
RAM	Random Access Memory
SSD	Solid State Drive
HMI	Human Machine Interface
IDEF	Integrated Definition
HTTPS	HyperText Transfert Protocol Secure
SQL	Structured Query Language
XSS	Cross-site Scripting
CSRF	Cross-site Request Forgery
PCI	Peripheral Component Interconnect
DSS	Decision Support System
GDPR	General Data Protection Regulation
COCOMO	COnstructive COst MOdel

List of figures

List of tables

INTRODUCTION

This research paper focuses on the development of a computer consumables ordering application specifically designed to meet the needs of GICOME TECHNOLOGIES. In a professional environment, the efficient management of IT consumables such as ink cartridges, toners, cables and peripherals is crucial to ensure the smooth running of IT activities and employee productivity.

The aim of this thesis is to study GICOME TECHNOLOGIES' requirements for ordering IT consumables, and to develop a tailor-made application that will optimize this process. This application will aim to streamline inventory management, facilitate consumable ordering, automate replenishment processes and provide accurate information on stock status.

Based on this in-depth analysis, a tailor-made application will be designed and developed using software development tools and techniques. The application will enable employees to place orders, check stock levels, access detailed product information and track deliveries. A robust database management system will also be implemented to ensure data accuracy and consistency.

This dissertation is divided into two main parts : the first deals with theoretical studies, and the second with design and implementation. Each part comprises two distinct sections. For the first part, the first chapter will present the context of the GICOME TECHNOLOGIES company, its activities and its specific needs in terms of ordering computer consumables. The second chapter will focus on needs analysis and information gathering. As for the second part, in the first chapter we will describe the design, modeling and conceptual approach. The second chapter covers the development and implementation of the application, focusing on key functionalities and technical challenges. Finally, the general conclusion will summarize the achievements of the dissertation and discuss prospects for improving and extending the application.

The aim of this dissertation is to make a practical contribution to GICOME TECHNOLOGIES by developing a tailor-made solution for ordering computer consumables. By automating and simplifying this process, the application will enable GICOME TECHNOLOGIES to save time, improve operational efficiency and reduce costs associated with inventory management. It is hoped that this work will provide useful recommendations for other companies wishing to implement similar solutions for their specific IT consumables needs.

PARTIE I : THEORETICAL STUDY

CHAPTER 1 : GENERAL INFORMATION

I. Context, Company presentation, Issues, Objectives, Survey of existing facilities

1. Context

The present study is a research dissertation looking at the field of computer consumables ordering application for the GICOME TECHNOLOGIES company. In today's digital age, IT plays a central role in our society, and companies and individuals alike need fast, efficient access to IT items such as computers, electronic components and peripherals. In this context, the development of an innovative solution to facilitate the ordering of these items becomes a necessity.

2. Company presentation

GICOME TECHNOLOGIES is a company created in 2018 and born of the desire to make quality IT services accessible to all. It is a company that offers IT and communication services to other businesses. It operates in the fields of supply, installation, maintenance, security, IT system design and in the drafting of communication and promotional strategies. GICOME TECHNOLOGIES offers its customers monitoring of their IT assets and perfect control of their network to optimize their business strategy. GICOME TECHNOLOGIES has a dynamic sales team ready to advise customers on their IT consumables needs, and to offer them reliable, high-performance products.

3. Issues

The absence of a computer consumables ordering application can lead to inventory management problems, inefficient ordering and delivery delays for a company. It is therefore important to solve this problem by implementing an application tailored to the needs of the company's customers. So how do you design and implement a computer consumables ordering application for a company that doesn't yet have a system in place, guaranteeing order management, ease of ordering, and improved delivery times?

4. Objectives

a. General objective

The main objective of this dissertation is to study and develop a computerized article ordering application for GICOME TECHNOLOGIES. This application will

offer an optimal user experience, while meeting the specific needs of the company's customers.

b. Specific objectives

The application aims to simplify the ordering process, provide detailed product information, manage transactions securely and guarantee fast, reliable delivery.

II. Survey of existing facilities

Since its creation, GICOME TECHNOLOGIES has not had an application for ordering its products. For this reason, our study focuses on the design of a mobile application for ordering computer consumables on behalf of the company in question.

There are similar applications to Easyorder, although these tend to focus on the delivery of food, fruit, honey and many other products.

Table 1 : Some applications similar to easyorder

Application name	Description	Functions
Okalm	Application for ordering meals, spices, fruit, drinks from restaurants near customers' homes. Online farming	Place an order Get delivered Order tracking
Farmigo	Application for ordering dairy products, snacks, meat.	Order a product Community section Get delivered
GrubHub	Food delivery application	Express restocking Map search Pre-order Online tracking Comment option

III. Description and features

1. Solution description

This solution, called EasyOrder, is a mobile application that enables Gicome customers to easily order all the computer consumables they need, such as printers, ink cartridges, toners, laptops and computer accessories. The application will enable users to browse available products, add desired items to their shopping cart, place an order in just a few clicks, and create company profiles for personalized orders.

2. Solution features

- Place an order: EasyOrder's user-friendly, intuitive interface makes it easy for customers to place their orders. This includes searching for products, selecting quantities and adding to the basket.
- Track orders: The application will enable Gicome customers to track the status of their orders, from confirmation to delivery. This will provide customers with a clearer picture of the order process, reducing order-related questions and concerns.
- Track order history: The application will enable customers to view their order history, including previously ordered products, order dates and delivery details. This will avoid the repetition of orders and the management of past purchases.
- Notifications and alerts: EasyOrder will send notifications and alerts to customers about the status of their orders, special promotions, special offers and out-of-stock products. This will keep customers informed and engaged.
- Comment and review: Customers will have the opportunity to leave comments and reviews on the products they have purchased. This will help other customers make informed purchasing decisions and provide Gicome with valuable information about the quality of its products.
- Customer assistance: EasyOrder will offer a customer support function via direct chat. This will enable customers to ask questions, report a problem or request help when needed.

CHAPTER 2 : METHODOLOGY AND CONCEPTUAL APPROACH

I. Modeling methodology

1. Concept of information system modeling

An information system is an organized set of resources such as people, technologies, processes and data that interact to collect, store, process and distribute information to support operations, decision-making, coordination and control within an organization.

Information system modeling is the process of representing and describing an organization's information system using models. These models are used to represent different aspects of the information system, such as processes, data, information flows, interfaces and users.

Information system modeling is essential, as it enables the structure and operation of the information system to be clearly and precisely understood and communicated.

2. The modeling process

The modeling process is an essential step in many fields, such as science and engineering. For the management of any IT project, it is important to choose an appropriate development process in advance, which will meet the functional, technical and qualitative requirements of the project. Although there are several modeling processes available, it is necessary to make a choice based on the different needs of the project. An appropriate development method is essential for the efficient and successful realization of an IT development project. The table 2 below shows the comparative study between the UP and 2TUP development processes.

Table 2 : Comparative study of two fairly familiar modeling processes.

Process	Description	Strengths	Weaknesses
UP	• The unified process is an iterative, incremental	• Iterative • Specifies exchanges : deliverables,	• Expensive to customize

	software development framework. • Methodology and tools • Projects involving more than 10 people	schedules, prototypes • Standard documents	• Process-oriented, little development
2TUP	• 2TUP is a software development process that implements the Unified Process methodology. • Y-shaped development cycle • Dissociates technical and functional aspects • Any project	• Iterative • Broad scope for technology and risk management • Defines stakeholder profiles, deliverables, schedules, prototypes.	• Cumbersome formalism • No standard documentation

For this project, 2TUP seems to be the most appropriate approach, as it enables us to effectively manage the changing requirements and architectural problems that can arise during project development. Indeed, 2TUP follows a Y-shaped development cycle that separates technical from functional aspects. This process is structured around three branches, as illustrated in figure 1.

- The functional branch capitalizes on the company's business knowledge. This branch captures functional requirements, producing a model focused on the end-user's business.

- The technical branch capitalizes on technical know-how and/or technical constraints. The techniques developed for the system are independent of the functions to be realized.
- The realization phase brings the two branches together, enabling us to carry out an application design and, finally, to deliver a solution adapted to requirements.

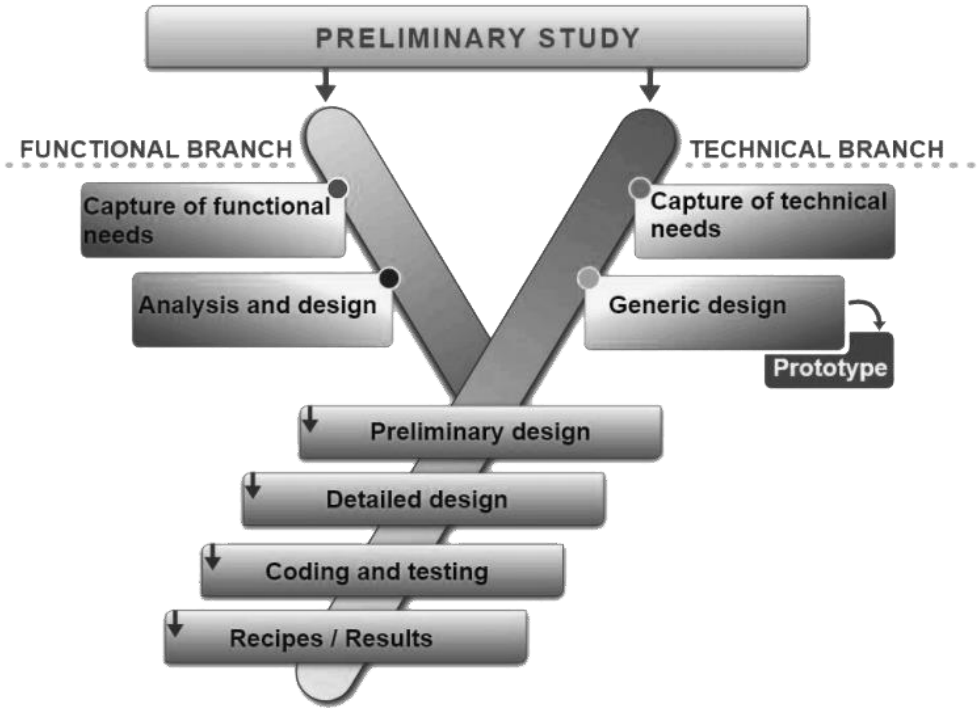


Figure 1: The Y development process and its strops

3. The modeling language

A modeling language is an artificial language that can be used to express information, knowledge or systems in a structure defined by a coherent set of rules. There are several modeling languages, such as EXPRESS and EXPRESS-G, SysML, Business Process Modeling Language, IDEF, Unified Modeling Language.

To model this project, we'll be using the Unified Modeling Language. The UML was designed to be a common, semantically and syntactically rich visual modeling language. Its aim is to facilitate the design of the documents needed to develop object-oriented software, as a standard for modeling software architecture.

Figure 2 shows the 14 UML diagrams

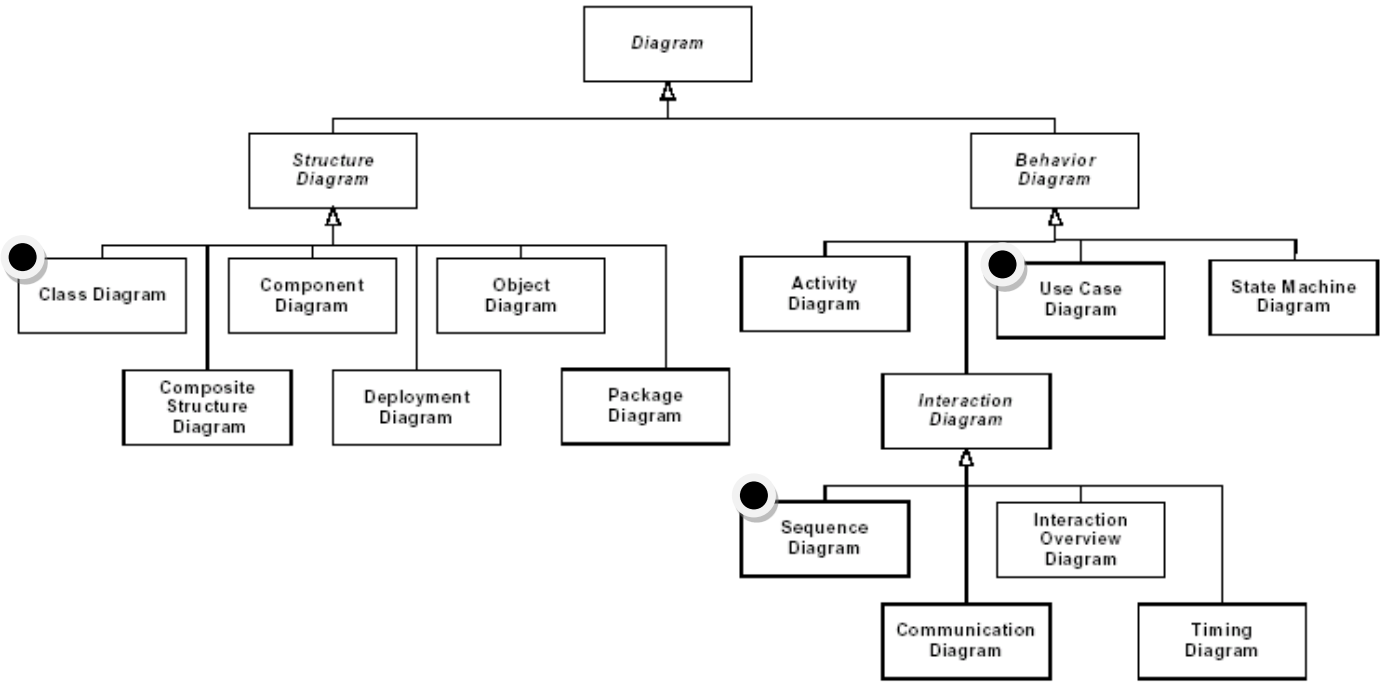


Figure 2: UML diagram

Only class, use case and sequence diagrams have been selected for this study. They will be used to model the IS to which the project relates, presenting the different activities that a user can perform on the system and the interactions between the user and the system.

II. Mobile development

1. Programming languages

A programming language is an algorithm-based computer language used to develop programs. Programming languages have evolved in parallel with web development techniques. Some are complementary, but all serve different purposes. There are almost 700 programming languages, each with its own specific features, advantages and disadvantages.

The following figure 3 presents a ranking of the 20 most popular programming languages in May 2023.

4	4		C++	11.96%	+3.13%
5	5		C#	7.43%	+1.04%
6	6		Visual Basic	3.84%	-2.02%
7	7		JavaScript	2.44%	+0.32%
8	10	^	PHP	1.59%	+0.07%
9	9		SQL	1.48%	-0.39%
10	8	v	Assembly language	1.20%	-0.72%
11	11		Delphi/Object Pascal	1.01%	-0.41%
12	14	^	Go	0.99%	-0.12%
13	24	^^	Scratch	0.95%	+0.29%
14	12	v	Swift	0.91%	-0.31%
15	20	^^	MATLAB	0.88%	+0.06%
16	13	v	R	0.82%	-0.39%

Figure 3: Ranking of the 20 most popular programming languages in May 2023

Choosing a programming language for a project depends on a number of factors. Here are a few steps to help you decide:

- Understand project requirements: i.e. needs and objectives, functionalities, type of application to be developed and technical constraints.
- Consider the strengths and weaknesses of the languages: some languages are better suited to web development, while others are better suited to

scientific applications. Find out about the key features of different languages to assess their relevance to your project.

- Evaluate the availability of resources: take into account the skills and experience of the development team. If there are developers already familiar with a specific language, it may make sense to choose that language.
- For this project, we'll be using a mobile development language to meet the project's needs.

2. State of the art in mobile development

A mobile application is a piece of software. More precisely, it's a program containing a file that can be downloaded from a cell phone or tablet. There are two types of mobile application: native and hybrid. The native application is a mobile application developed specifically for one of the operating systems used by smartphones and tablets (IOS and Android). A hybrid application, on the other hand, is a mobile application capable of running on all the operating systems used by smartphones. Hybrid applications are faster and easier to develop than native applications, and require less maintenance. The speed of a hybrid application depends entirely on the speed of the user's browser. A hybrid application can be created on a single basis, enabling new functionalities to be added to several versions of the application. Whereas in a native application, to introduce a new feature, it must be replicated across all versions of the application.

For this project, we'll be developing a hybrid application, because it's easier and faster to develop than a native application. It's also easier to maintain, as there's only one version to review for multiple platforms.

3. Programming languages for hybrid development

Several programming languages are used to develop hybrid applications, but the best-known are React Native and Flutter.

React Native is an open-source mobile application framework created by Facebook. Its first version was released on March 26, 2015. It is used to develop applications for Android and iOS, enabling developers to use React with the native features of these platforms.

Flutter is an Open Source Framework developed by Google. Its first alpha version was released in 2017. Since its creation, Flutter has been increasingly used by

developers, and today ranks among the technologies of first choice for mobile application development.

The following table 3 gives a general description of each of these two hybrid development technologies.

Table 3 : Description of React Native and Flutter

Technology	React Native	Flutter
Programming Langage	Javascript	Dart
Main architecture	Flux and Redux	Block
Created by	Facebook	Google
Learning curve	Easy to use, especially with prior knowledge of React or JavaScript	Quite fast, but learning the Dart language is mandatory as there's a strong link.
Ecosystem	Quite mature, used in production by several companies, with several packages.	Not yet mature, fewer packages
Benefits	Easy to master, optimized performance, saves time	Create beautiful interfaces, well-done documentation, fast and efficient debugging, fast compilation
Limits	Slow adoption of latest features, difficult to debug, relatively complex user interface.	Technology is still young, evolving very rapidly: packages are quickly obsolete and must be updated.

III. Databases

1. The concept of a database

A database is a collection of information organized in such a way that it can be easily accessed, managed and updated. There are many database options for an application, but they generally fall into one of two categories:

Relational: a relational database is a type of database management system that stores and organizes data in a structured way. It is based on the relational model, which defines the relationships between the different entities in the database. An example of a relational database is SQLite.

NoSQL : a NoSQL database is a type of database that uses a flexible data model, called a document model. MongoDB is an example of a NoSQL database.

A database is generally controlled by a database management system (DBMS).

2. Database for a Flutter application

The backend is an essential element of any application platform ; it's the engine that processes requests for information and delivers the results to client terminals. It is the backend that powers the application's functions. For a Flutter application, there are several databases that can be used. Figure 4 below shows the different databases that can be used with a mobile application developed by Flutter.

For this project, we'll be using Firebase as our database, as it enables us to develop a base of committed users. It also enables us to create high-quality applications.

Figure 4:Some backend technologies with Flutter

IV. Project planning

1. Planning concept

Project planning is an essential phase in the life cycle of a project. It is the activity that consists in determining and ordering the project tasks, estimating their load and giving the necessary resources to carry them out. Planning allows the project to succeed, especially under the constraints of time and deadline.

2. Gantt chart

The following figure 5 is a Gantt chart that summarizes the schedule of the project's stages, thus simplifying the monitoring of the progress of the defined tasks.

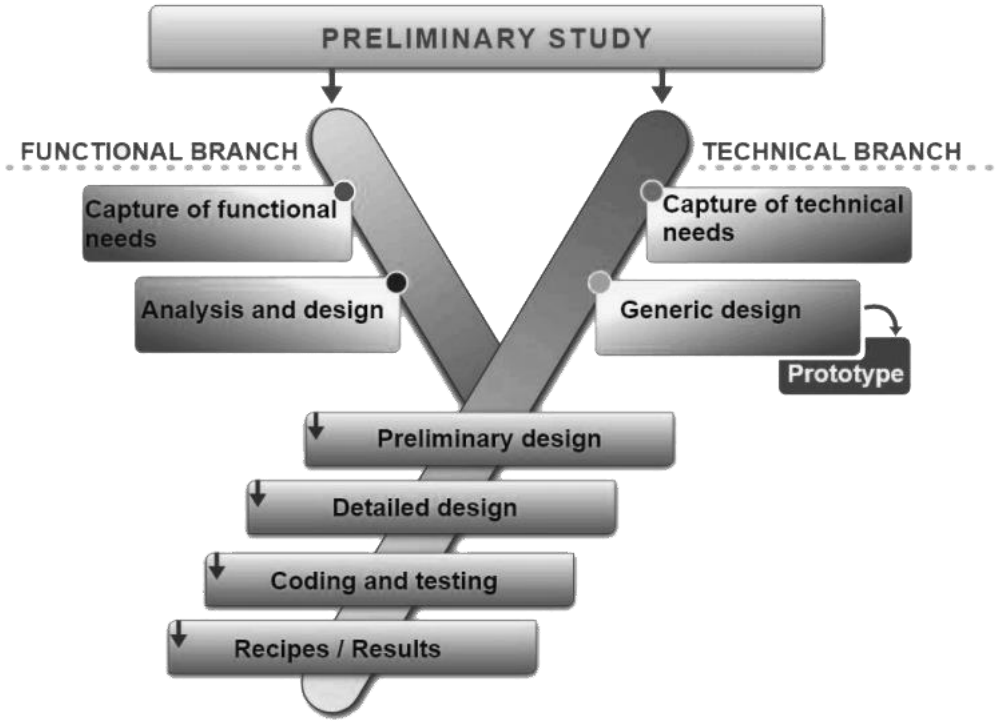


Figure 5: Gantt chart

PARTIE II : DESIGN AND IMPLEMENTATION

CHAPTER 1 : MODELING AND DESIGN

I. Preliminary study

1. The preliminary study in the Y process

This marks the initial phase of the 2TUP process and holds significant importance. It starts by clearly defining the primary directions of the project. Subsequently, it undertakes the initial identification of functional and operational needs primarily using text or basic diagrams. Following these steps, it involves pinpointing the involved actors and constructing the context diagram. This preliminary stage equips project developers to adeptly approach the activities within the two branches of the 2TUP process – the capture of functional requirements and technical requirements.

2. Presentation of the project

EasyOrder is an innovative solution developed to address the order and sales management needs within Gicome Technologies. This hybrid application, compatible with both Android and iOS platforms, aims to optimize and streamline the order and sales process while providing an exceptional user experience.

From a technical perspective, the following outcomes are expected at the end of the EasyOrder project:

- **Comprehensive Database** : To have a centralized database representing the storage part of information related to products, customers, orders, and registered managers for efficient management and quick data access.

- **Order Process Optimization** : Simplify the order process by enabling customers to place orders quickly and intuitively through the application, thereby reducing delays and errors.

- **Real-time Order Tracking** : Provide customers and managers with the ability to track the status of orders in real-time, from validation to delivery, enhancing transparency and trust.

- **Complete Product Catalog** : Present a detailed product catalog with comprehensive information about products offered by Gicome Technologies, including descriptions, images, and prices.

- **Efficient Inventory Management** : Allow the company to maintain an accurate overview of real-time stock levels, facilitating supply management and preventing stockouts.

- **Enhanced Security** : Ensure application security by implementing mandatory authentication for all users, ensuring that only authorized users can access application features.

- **Administration Module** : Set up an administration module that enables managers to efficiently manage products, orders, customers, and application settings.

- **Improved Operational Efficiency** : Simplify and automate processes related to ordering, selling, and inventory management, reducing manual efforts and enhancing overall operational efficiency.

- **Enhanced User Experience** : Offer a smooth and user-friendly experience that encourages customers to use the application regularly, while increasing their engagement with Gicome Technologies.

- **Digital Transformation** : Prepare Gicome Technologies for digital transformation by adopting modern technological tools to manage business operations, remain competitive in the market, and meet changing customer expectations.

By achieving these outcomes, EasyOrder aims to enhance internal management, bolster customer satisfaction, and position Gicome Technologies as a technology-forward company in the order and sales sector.The following illustration (Figure 5) shows the different steps taken to achieve these expected results.

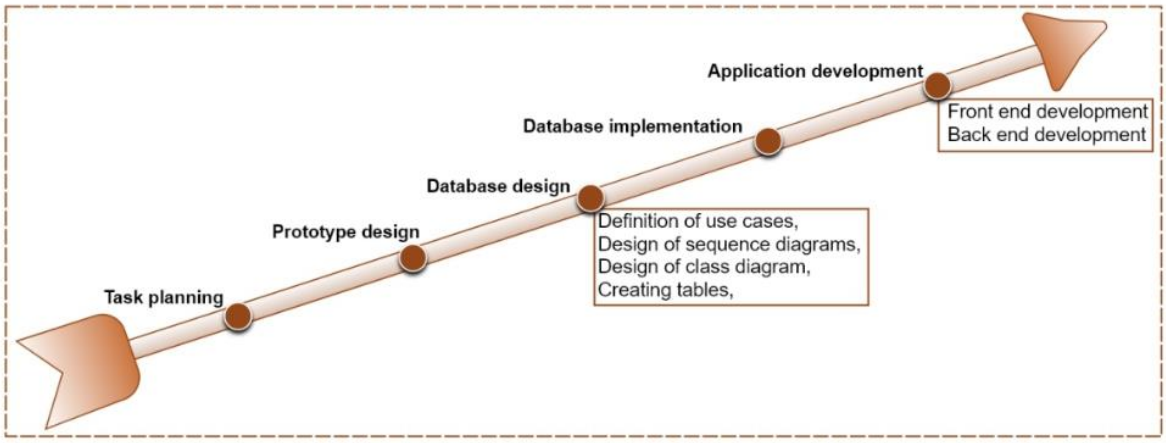


Figure 6: The different steps of development of the project

3. Initial collection of functional and technical requirements

In the stage of collecting functional requirements, the emphasis is primarily placed on identifying and describing the various planned computer processes. The key functionalities that have been selected are as follows :

In the functional requirements collection phase, the essential features that have been identified for the EasyOrder project are as follows:

- **Order and Sales Management** : Users will be able to place orders and manage sales efficiently and intuitively through the application.
- **Product Management** : Administrators will have the ability to add, update, and delete products, while managing associated information.
- **Comprehensive Product Catalog** : Customers will have access to a complete catalog of products with detailed descriptions, images, and prices.
- **Integrated Payment System** : The application will feature a secure payment system, allowing customers to make purchases securely.
- **User Account Management** : Users can create accounts, view their orders histories, and manage their personal information.

- **Administration Module** : Administrators will have a dedicated module to oversee orders, manage inventory, and access sales reports.
- **Management Module** : Managers will have a dedicated module to oversee orders and access sales reports.
- **Real-time Notifications** : Users will receive notifications to stay informed about the status of their orders and special offers.
- **Integrated Customer Support** : Customers will have a means to contact customer support for questions or issues.

These essential features have been selected to address the order and sales management needs within Gicome Technologies through the EasyOrder application.

With regard to the collection of operational needs, these factors pertain to the functioning of the application. They ensure the effectiveness of the system and contribute to enhancing the application's quality. The non-functional requirements identified for this project are as follows:

- **Security** : The system will be secure and safeguarded against unauthorized access. Each user is required to have a username and password.
- **Performance** : The system will possess the capability to swiftly respond to requests, ensuring efficient and timely interactions.
- **Availability** : The system will maintain constant accessibility to its users, ensuring uninterrupted service.
- **Reliability** : The system will execute its intended functions with precision and accuracy, maintaining consistent performance.

These non-functional requirements are crucial for ensuring the operational excellence and user satisfaction of the EasyOrder application.

4. Description of the context

This step consists of three (03) successive stages : the identification of system actors, the enumeration of messages, and the creation of context diagrams.

Identification of actors : An actor represents the abstract representation of a role played by external entities (user, hardware device, or other system) that interact directly with the system under study.

Enumeration of messages : A message represents the specification of one-way communication between objects with the intent of triggering an activity in the receiver.

Creation of context diagrams : Context modeling involves determining the functionalities that the system provides to each user.

The following tables (Table 4, Table 5, and Table 6) present the various actors and the different permissions that the system grants to each of them.

Table 4 : Presentation of the actor 1 and his messages

Actor 1	Type	Description of the functional requirements
Customer	Human	The system offers a customer the possibility to: ▪ Authenticate yourself ▪ Add favorites ▪ Consults promotions ▪ Search products ▪ Consult categories ▪ View product details ▪ Add to card ▪ Place order ▪ Manage profile ▪ Make payment ▪ Make feedback ▪ Contact support

Table 5 : Presentation of the actor 2 and his messages

Actor 2	Type	Description of the functional requirements
		The system offers a customer the possibility to: ▪ Authenticate yourself ▪ Consult orders ▪ Validate orders

Manager	Human	▪ Manage delivery ▪ Manage feedback ▪ Consults statistics ▪ Respond to support requests ▪ Manage promotions ▪ Manages urgent deliveries ▪ Manages notifications

Tableau 6 : Presentation of the actor 3 and his messages

Actor 2	Type	Description of the functional requirements
Administrator	Human	The system offers a customer the possibility to: ▪ Authenticate yourself ▪ Consult orders ▪ Validate orders ▪ Manage delivery ▪ Manage feedback ▪ Consults statistics ▪ Respond to support requests ▪ Manage promotions ▪ Manages urgent deliveries ▪ Manages notifications ▪ Manage stocks ▪ Manage account

II. Functional study

1. The functional study in the Y process

The functional branch (or left branch) of the 2TUP process aims to define the constraints, gather the functional requirements, and analyze the functional specifications in order to determine what the system will actually achieve. It is in this branch that the project developers strive to identify and extract all the functionalities of the system to be implemented.

2. Capture of functional requirements

The objective of the functional requirements capture is to identify and describe the various usage scenarios of the system.

In other words, a usage scenario describes how the system behaves in different situations in response to a user's (or actor's) request to achieve a specific goal. The tables below provide a detailed description of some identified usage scenarios.

Table 7 : Description of the use case "Authentication"

Use case	Authentication
Actors	Customer, manager, administrator
Description	The user at the launch of the application is subjected to this step of authentication. He fills in a form and submits it. The system then checks the conformity of the data, gives him a message and assigns him his role.
Messages sent by the actors	User name and password to be filled in the form
Messages received by actors / system behavior	▪ Authorized Access / User Role Assignment ▪ Access denied / awaiting re-verification ▪ Access denied / Proposal to recover the password ▪ Account not found / Authorization of an entry

Table 8 : Description of the use case "Place an order"

Use case	Place an order
Actors	Customer,manager,administrator

Description	The action of placing an order allows customers to select and purchase products through the EasyOrder application. The process starts when the customer logs into their account and accesses the ordering section. There, they can browse the product catalog, choose the items they want to purchase, and specify the desired quantities. Once the customer has added all items to their cart, they can proceed to the payment step. The system will provide them with available payment options and request necessary information, such as credit card details. After completing the payment, the system will generate an order receipt and send a confirmation to the customer. On the administrators' or managers' side, they will have the ability to manage the received orders. They can view order details, track their status, and take actions if needed, such as managing inventory.
Messages sent by the actors	Selection of items, choice of quantities, payment information.
Messages received by actors / system behavior	▪ Payment validation and receipt generation ▪ Update of the order status ▪ Order management on the Manager's/Administrator's side

Tableau 9 : Description of the use case "Manage stocks"

Use case	Manage stocks
Actors	Administrator

Description	This use case allows the administrator to monitor and modify the stock levels of various products, ensuring that the system accurately represents the availability of products for purchase by customers. Once logged in, the administrator accesses the "Stock Management" section where they can add, delete, or modify the quantities or details of products.
Messages sent by the actors	▪ Modify informations ▪ Add products ▪ Delete products
Messages received by actors / system behavior	▪ Update successed / error ▪ Cancel successed/error

3. Analysis and design

This final phase of the left branch of the Y process holds significant importance. The analysis stage comprises several steps, with the key ones outlined in this context being the creation of the system's use case diagram and the development of sequence diagrams for each individual use case.

a. The use cases diagram

In UML, use case diagrams model the behavior of a system, describe its general functions, scope, and allow the capturing of the system's requirements. In a use case diagram, users are referred to as "actors," and they interact with various "use cases" within the system. However, an association between an actor and a use case is established through a "association" represented by a simple line. Figure 6 below illustrates the system's use case diagram. This diagram is composed by three (03) main actors as indicated in the preliminary study.

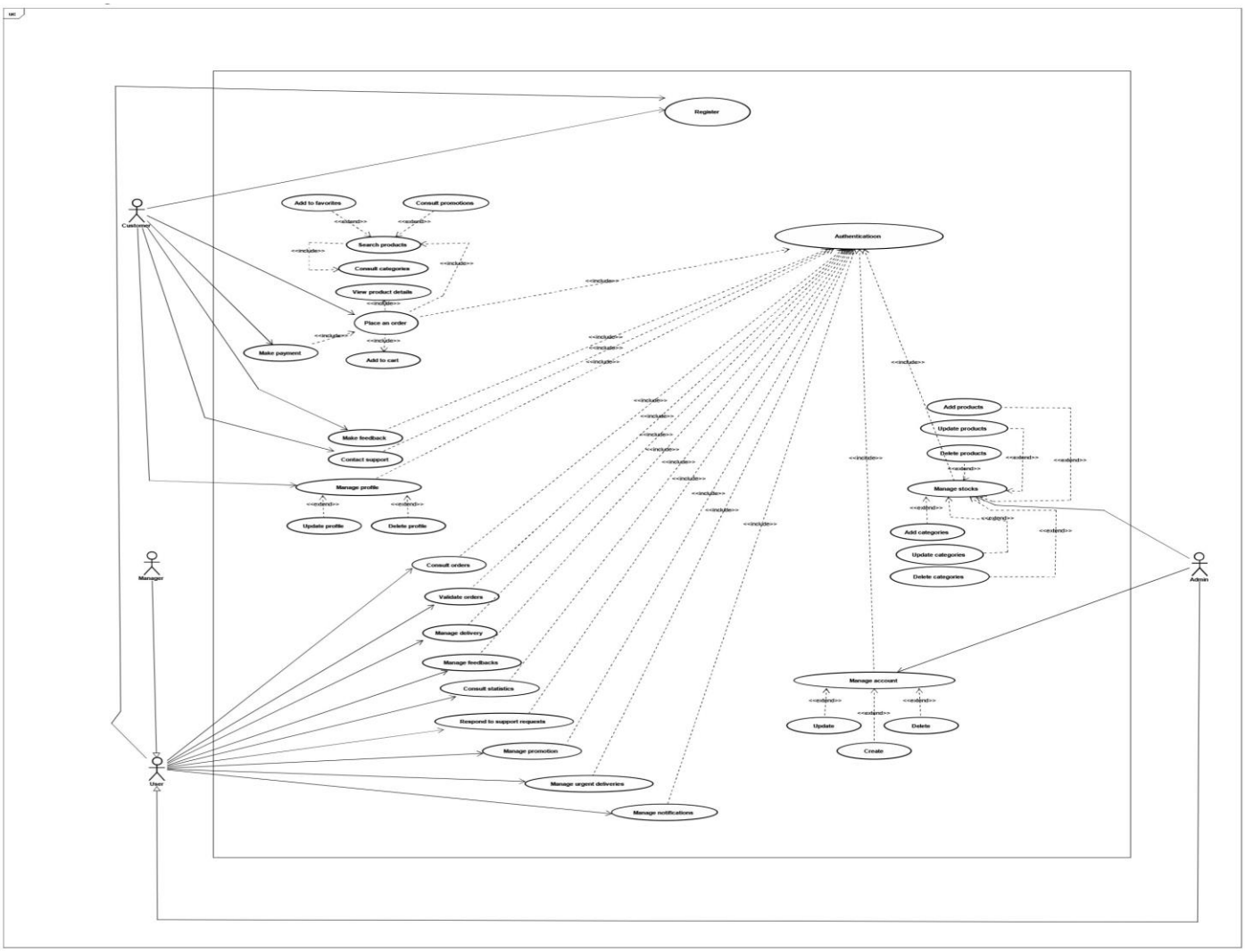


Figure 7: System Use case diagram

b. Sequence diagrams

Sequence diagrams provide a visual depiction of interactions between actors (entities) and the system, arranged chronologically based on their occurrence. The following paragraphs will showcase sequence diagrams illustrating various use cases of the system.

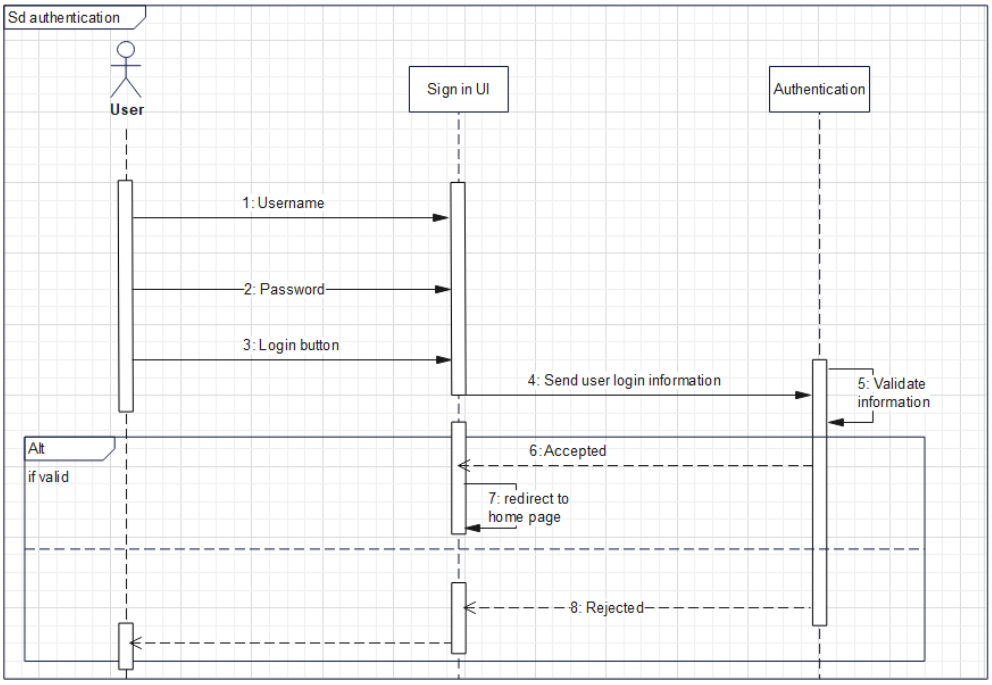


Figure 9: Sequence diagram " Authentication "

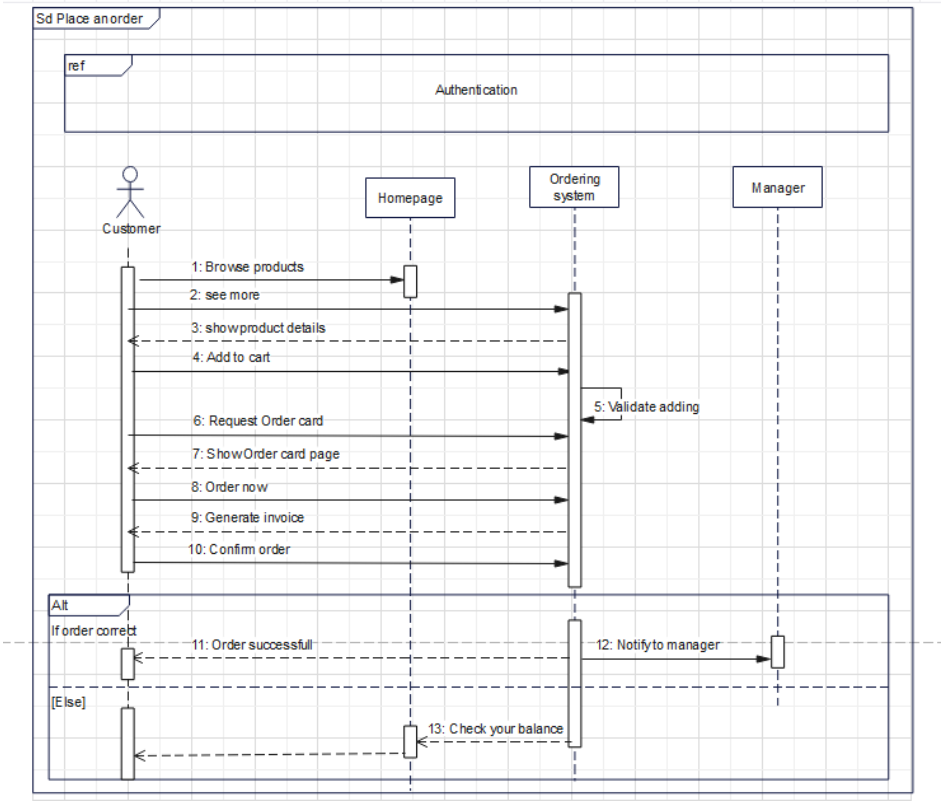


Figure 8: Sequence diagram " Place an order "

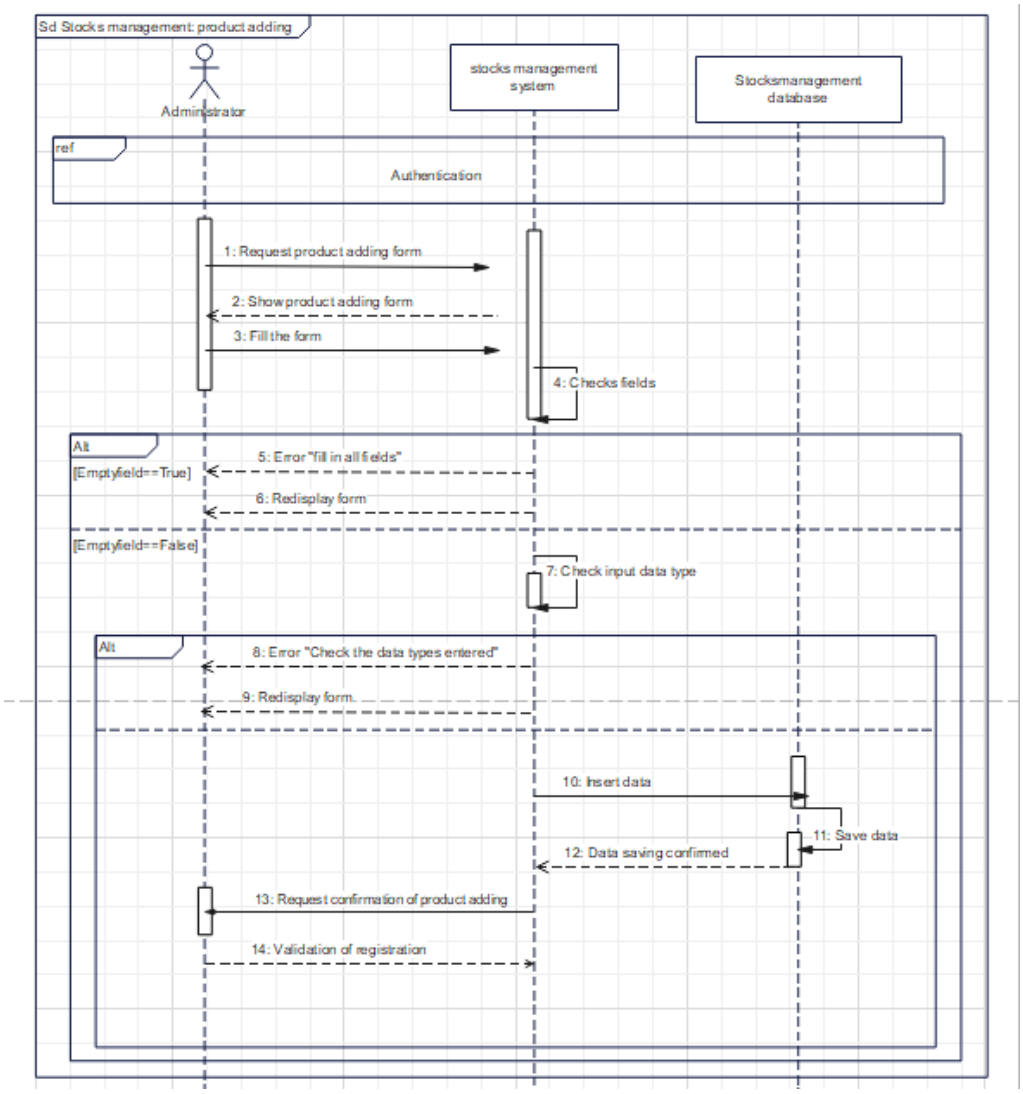


Figure 10:Sequence diagram " stock management product adding "

III. Technical study

1. The technical study in the Y process

The technical branch (or right branch) of the 2TUP process focuses on capturing the non-functional requirements of the system. This primarily involves identifying technical decisions, hardware choices, and constraints that will shape the application's design (including integration, development, and performance constraints). This branch is divided into two main sections : technical requirements capture and high-level design.

2. Capturing technical requirements

In order to satisfy the functional and technical requirements that were listed in the preliminary study, several tools and technologies were used and the objective of this first step of the right branch in the 2TUP process is to bring out all these different technologies that are :

- Lenovo ThinkPad: Laptop equipped with 8GB RAM, 512GB SSD, and an Intel Core i5 processor.
- Visual Studio Code : Employed for code editing purposes.
- Android Studio : Utilized to create an Android Virtual Device.
- Flutter & Dart : Employed for frontend programming of the application.
- Firebase : Utilized to establish and manage the application's database.
- Astah UML : Used for modeling UML system diagrams.
- Adobe Photoshop/Illustrator : Applied for image processing for the application.
- Figma : Employed for designing the application prototype.

In simpler terms, a Lenovo ThinkPad laptop with specific specifications is used alongside software tools such as Visual Studio Code, Android Studio, Flutter & Dart, Firebase, Astah UML, Adobe Photoshop/Illustrator, and Figma. These tools collectively contribute to various aspects of the application development process, including coding, design, database management, and more.

3. Generic design

In the final phase of the right branch within the 2TUP process, we reach the stage of generic design. During this step, it is advisable to create a prototype as a means to grasp the application's interfaces and functionalities. The primary emphasis lies on the visual and design aspects of the application. It becomes crucial to determine key elements such as color schemes, font styles, and shapes, as well as to craft templates for the primary pages of the application.

The provided Figure 7 showcases the initial prototype of the application, meticulously designed using Figma, at the project's inception.

Figure 11: Overview on some interfaces of the prototype(1)

Figure 12: Overview on some interfaces of the prototype(2)

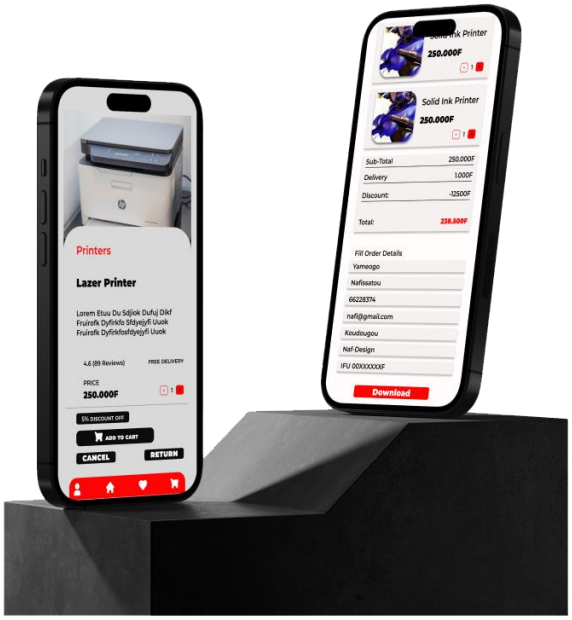


Figure 13: Overview on some interfaces of the prototype(3)

CHAPTER 2 : IMPLEMENTATION AND RESULTS

I. Implementation of the solution

1. Implementation in the Y process

The implementation phase constitutes the final branch (the middle branch) of the 2TUP development process and involves integrating the outcomes of both the functional and technical branches. The objectives of this concluding branch within the 2TUP process are twofold: first, to solidify the design model into application layers and human/machine interfaces (HMIs); and second, to carry out testing on the end product at the conclusion of the project to assess goals and juxtapose achieved results against anticipated outcomes. This segment encompasses five (5) primary steps, namely preliminary design, detailed design, coding, testing, and deployment.

2. The preliminary design

This constitutes the initial phase within the middle branch of the 2TUP process. Precisely, it is at this stage that the integration of the functional and technical analyses starts. A range of tasks needs to be undertaken, offering the opportunity to outline the distinct Human-Machine Interfaces (HMIs) that will facilitate users' interactions with system components. The diverse HMIs of the application, derived post this examination, are detailed in table below.

Table 10 : List of interfaces that will be implemented in the application

HMI	Description
The authentication interface	This interface allows a user to connect each user (administrator, specialist, breeder) and redirects him to his interface according to his privilege.
The homepage interface	This interface offers users access to the catalog of available products within the easyorder application. Users can browse through various product categories, view detailed product information, including images and descriptions, and select

	items to add to their cart for purchase. The homepage interface is designed to make it easy for users to explore and choose products they are interested in buying.
The stocks management interface	This interface provides administrators with the tools to monitor and control the inventory of products available within the EasyOrder application. Administrators can view a comprehensive list of products along with their current stock levels. They have the capability to adjust stock quantities, update product information, and ensure accurate representation of product availability for customers. This interface plays a crucial role in maintaining the correct inventory information and ensuring seamless order fulfillment.
The account management interface	This interface provides administrators with the tools to effectively manage user accounts within the EasyOrder application. Administrators can access a comprehensive list of registered users, including breeders, specialists, and other relevant roles. Through this interface, administrators have the ability to perform various actions, such as creating new accounts, modifying existing account details, or deactivating accounts as needed.
The account profile management interface	It is in this interface that the users (customer and manager) can edit the information concerning them on their profile (Name, first names, password, number).
The order list page interface	This interface serves as a centralized hub for managing and tracking customer orders within the EasyOrder application. It provides an organized view of all incoming orders, allowing users to efficiently monitor their status and take appropriate actions
The order overview page interface	This interface presents a summary of each order, including essential details such as order number, date, customer name, customer city and total amount.

The profile page interface	This enhanced profile page interface includes a menu that provides users with convenient access to various sections and features of the EasyOrder application. The menu complements the profile page by offering quick navigation and interaction options.

3. The detailed design

The detailed design phase, which follows the preliminary design, holds significant importance within the 2TUP process. In this phase, the focus shifts to precisely defining and elaborating on all the classes, subsequently constructing the comprehensive class diagram of the system.

a. Classes identification

This task primarily involves the design of classes, associations, and attributes. A comprehensive analysis has pinpointed a total of eight (9) classes for this project, namely : "User", "customer","Product", "Manager", "Administrator", "Account", "Order", "Catalog" and "Cart". The subsequent tables furnish descriptions of selected classes within this set.

Table11: Description of class ‘’Product’’

Class : ‘’Product’’				
Attributes	Visibility	Name	Type	Description
	Public	idProduct	int	Unique identifier of the product
	Public	name	String	Product name
	Public	description	String	Product description
	Public	imageUrl	String	Product image URL
	Public	price	double	Product price
Methods	Visibility	Name	Description	
	Public	getProductDetail()	Get detail of product	

Table 12: Description of class ''Order''

Class : ''Order''				
Attributes	Visibility	Name	Type	Description
	Public	idOrder	int	Unique identifier of the order
	Public	user	User	The user who placed the order
	Public	product	List<Product>	list of products included in the order
	Public	date	Date Time	Date and time of the order
	Public	status	String	Current status of the order, e.g., "Pending," "Shipped,"
Methods	Visibility	Name	Description	
	Public	addProduct()	Adds a product to the order	
	Public	removeProduct()	Removes a product from the order	
	Public	placeOrder()	Places the order	

Table 13: Description of class ''Cart''

Class : ''Cart''				
Attributes	Visibility	Name	Type	Description
	Public	idcart	int	Unique identifier of the order
	Public	user	User	The user who placed the order
	Public	items	List<cartItem>	List of products included in the order
	Public	total	double	Date and time of the order
	Visibility	Name	Description	

Methods	Public	addItems()	Adds a item to the cart
	Public	removeItems()	Removes a item from cart
	Public	clearCart()	Empty cart
	Public	getCartTotal()	Get the total amount of the cart
	Public	placeOrder()	Creates an order from the cart items

b. The classes diagram

A class diagram is a form of UML diagram which portrays a system's essence by providing a visual depiction of the diverse object types within that system, as well as the static relationships that interconnect them. The creation of class diagrams bears significant importance, essentially being obligatory for object-oriented systems. These diagrams offer a lucid illustration of a specific system's framework by representing its classes, attributes, functionalities, and the fundamental relationships that tie its objects together. The illustration in figure 8 below showcases the resulting class diagram for the system.

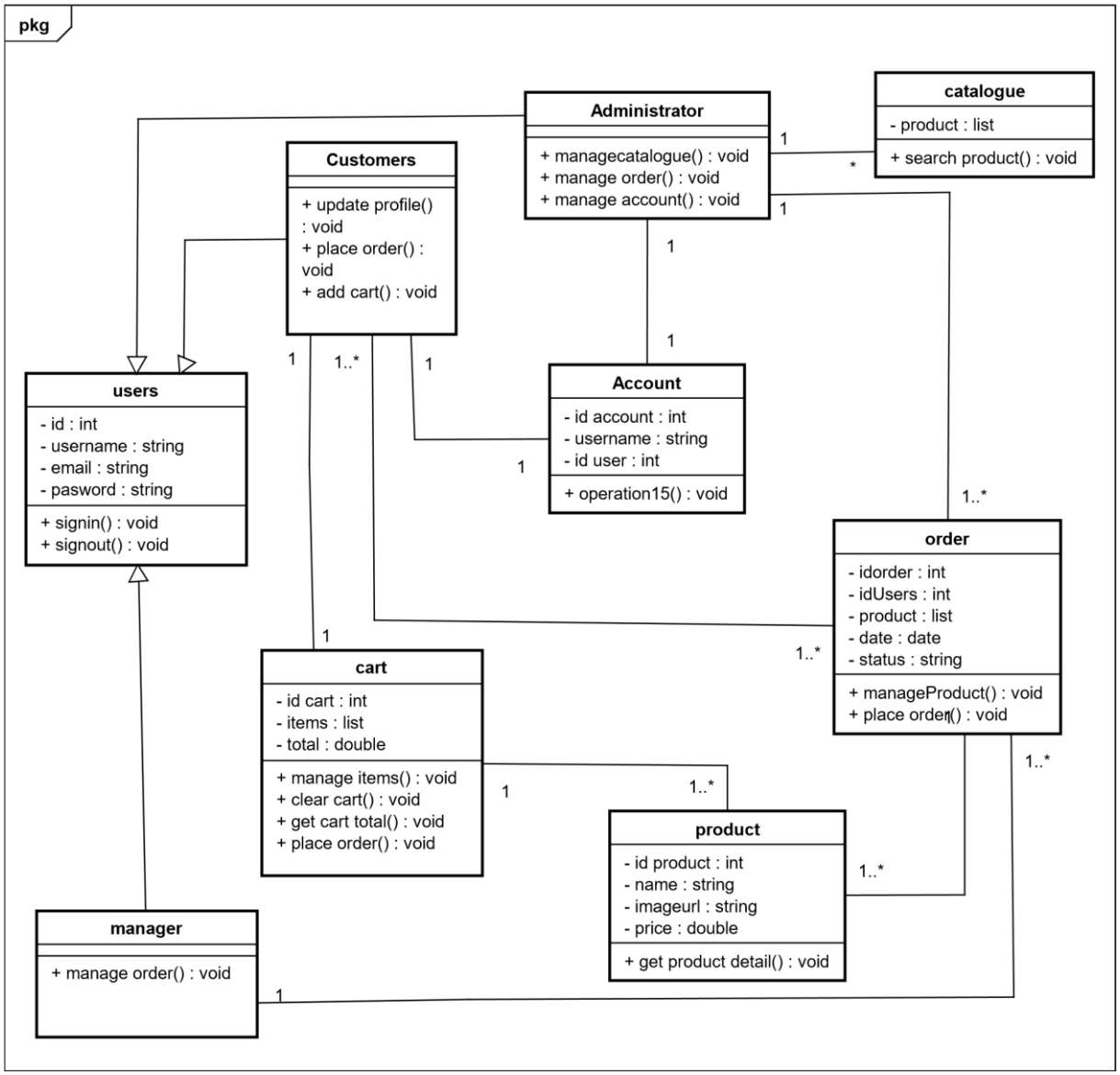


Figure 14: Classes diagram of the system

II. Coding and test

1. Explanation of how the code works

After generating the project, the main folder named "lib" contains the Dart code of the application, with a file named "main.dart" inside. This is the main file that is called when the application starts, and the objective of this section is to analyze the code and the structure of the "main.dart" file. The figure 9 below presents a view of the code contained in this file, in other words.

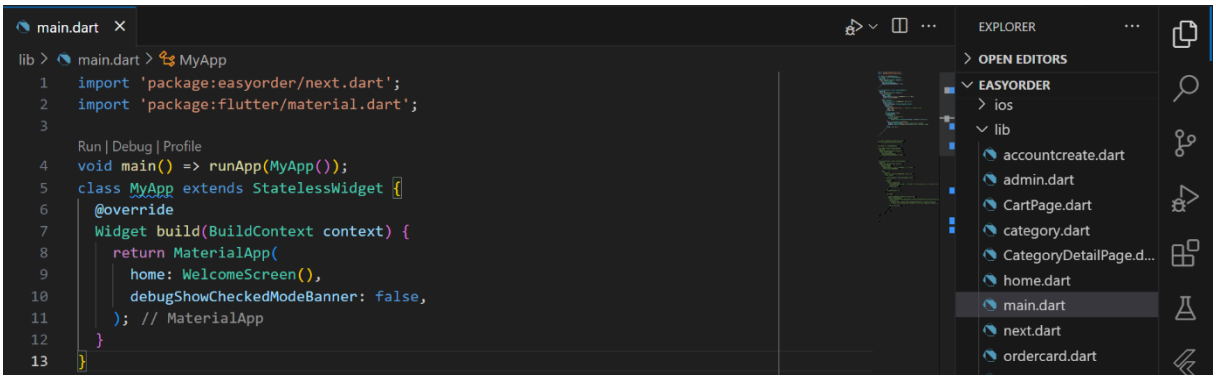


Figure 15 : Preview on the code of the "main.dart" file

```
Line1 import 'package:easyorder/next.dart';
```

- This line imports another Dart file, ‘’next.dart’’, located in the same package (package "easyorder"). This allows using classes and functions defined in the ‘’next.dart’’ file within this file.

```
Line2 import 'package:flutter/material.dart';
```

- This line imports the Flutter package ‘’material.dart’’, which contains widgets and features for creating user interfaces based on Material Design.

```
Line4 void main() => runApp(MyApp());
```

- This is the entry point of the application. The ‘’main’’ function is executed when the application starts. It calls the ‘’runApp’’ function, passing an instance of the ‘’MyApp’’ class as an argument. ‘’runApp’’ is a Flutter function that launches the application using the provided widget.

```
Line5 class MyApp extends StatelessWidget {
```

- Definition of the ‘’MyApp’’ class, which inherits from ‘’StatelessWidget’’. This means that ‘’MyApp’’ is a widget whose interface doesn't change over time (i.e., it's "stateless").

```
Line6 @override
```

- The ‘’@override’’ annotation is used to indicate that the following method (in this case, ‘’build’’) is overriding a method from the parent class (‘’StatelessWidget’’).

```
Line7 Widget build(BuildContext context) {
```

- This ‘’build’’ method is responsible for constructing the widget's user interface. It receives a ‘’BuildContext’’ object as a parameter, which provides information about the current build context.

```
Line8 return MaterialApp(
```

- The ‘’build’’ method returns a ‘’MaterialApp’’ widget, which is a wrapper for a Material Design-based application.

```
Line9 home: WelcomeScreen(),
```

- The ‘’home’’ argument of ‘’MaterialApp’’ sets the application's home screen. Here, it's set to an instance of the ‘’WelcomeScreen’’ class. This means that when the application starts, the ‘’WelcomeScreen’’ will be displayed to the user.

```
Line10 debugShowCheckedModeBanner: false,
```

- This line disables the display of the debug banner in development mode. In other words, the "Debug" banner in the top-right corner of the user interface won't be shown when set to ‘’true’’.

```
Line12 }
```

- Closes the ‘’build’’ method.

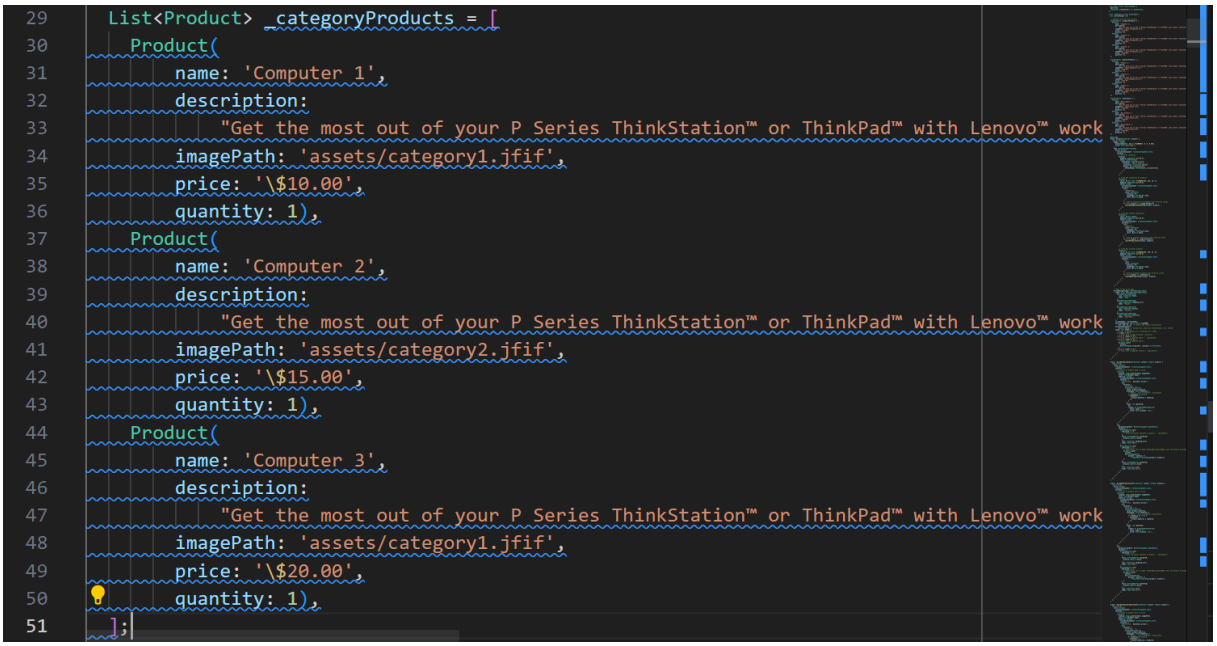
```
Line13 }
```

- Closes the ‘’MyApp’’ class.

In summary, this code imports necessary libraries, defines a ‘’MyApp’’ class that inherits from ‘’StatelessWidget’’, creates an instance of ‘’MaterialApp’’ with ‘’WelcomeScreen’’ as the home screen, and disables the debug banner in development mode. The ‘’main’’ function launches the application using ‘’runApp(MyApp())’’.

2. Dart collections

In Dart programming, a collection is a group of values or objects that can be stored, manipulated, and accessed together. Dart provides several built-in collection types that serve different purposes and offer various functionalities. These collection types help developers manage and work with data efficiently. We have list, map, set, queue.



```
List<Product> _categoryProducts = [
  Product(
      name: 'Computer 1',
      description:
          "Get the most out of your P Series ThinkStation™ or ThinkPad™ with Lenovo™ work
      imagePath: 'assets/category1.jfif',
      price: '\$10.00',
      quantity: 1),
  Product(
      name: 'Computer 2',
      description:
          "Get the most out of your P Series ThinkStation™ or ThinkPad™ with Lenovo™ work
      imagePath: 'assets/category2.jfif',
      price: '\$15.00',
      quantity: 1),
  Product(
      name: 'Computer 3',
      description:
          "Get the most out of your P Series ThinkStation™ or ThinkPad™ with Lenovo™ work
      imagePath: 'assets/category1.jfif',
      price: '\$20.00',
      quantity: 1),
];
```

Figure 16: List of product category

The code initializes _categoryProducts as a list that holds instances of the Product class.

Three instances of Product are created using the constructor. Each instance represents a product with the following attributes:

name: The name of the product (e.g., "Computer 1").

description: A detailed description of the product.

imagePath: The path to the product's image (e.g., 'assets/category1.jfif').

price: The price of the product as a string (e.g., '$10.00').

quantity: The quantity of the product (e.g., 1).

These product instances are added to the _categoryProducts list.

In summary, this code creates a list of products with different attributes and their associated values. These products can be used in the application to display and manage information about different categories of products.

3. The database with Firebase

Firebase was used to manage our application database. It is a mobile and web application development platform offered by Google. It provides a variety of

tools and services that simplify the creation, management, and scaling of applications, particularly those requiring real-time features, cloud databases, authentication services, and much more. In order to use these services, you must first configure and connect your application to Firebase. Figure 10 below shows some prerequisites and steps to configure and connect the application to

the database.

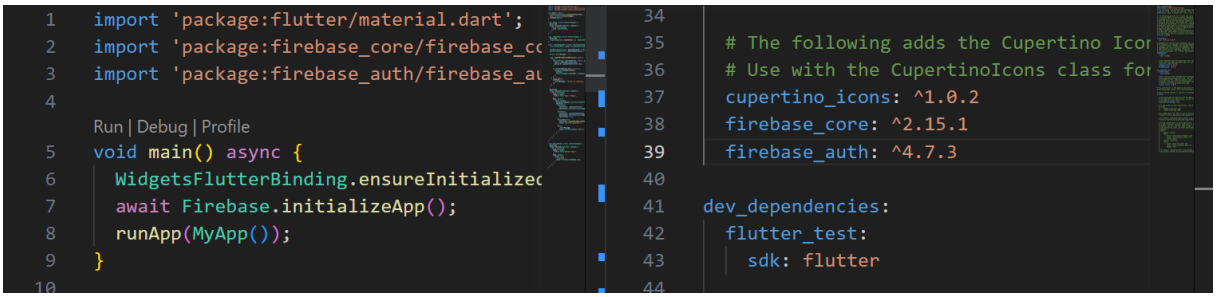


Figure 17: Firebase initialization and connection to the database steps

```
cupertino_icons: ^1.0.2
firebase_core: ^2.15.1
firebase_auth: ^4.7.3
```

These three lines provide the versions of the ‘’cupertino_icons’’, ‘’firebase_core’’, and ‘’firebase_auth’’ packages used in the Flutter project. This code is located in the ‘’pubspec.yaml’’ file, where project dependencies are managed.

```
cupertino_icons: ^1.0.2
```

This specifies the version of the ‘’cupertino_icons’’ package being used. This package provides Cupertino icons designed for iOS-style applications. The version ‘’^1.0.2’’ indicates that we're using version 1.0.2 or higher, allowing for backward-compatible updates.

```
firebase_core: ^2.15.1
```

This specifies the version of the ‘’firebase_core’’ package we're using. This package is the core Firebase SDK, necessary for initializing and configuring Firebase services in our application. The version ‘’^2.15.1’’ indicates that we're using version 2.15.1 or higher, allowing for backward-compatible updates.

```
firebase_auth: ^4.7.3
```

This specifies the version of the ‘’firebase_auth’’ package we're using. This package provides Firebase authentication services, allowing us to manage user authentication in our application. The version ‘’^4.7.3’’ indicates that we're using version 4.7.3 or higher, allowing for backward-compatible updates.

These version specifications ensure that our project uses the specified package versions or higher while maintaining compatibility with the code. It's a good practice to regularly check for updates to these packages to take advantage of bug fixes, new features, and improvements. These updates can be applied by running the ‘’flutter pub upgrade’’ command in the project directory.

```
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_auth/firebase_auth.dart';
```

These two code snippets include two import statements from Flutter packages related to Firebase: ‘’firebase_core’’ and ‘’firebase_auth’’. Let's explain what each of these lines does:

```
import 'package:firebase_core/firebase_core.dart';
```

This line imports the necessary components from the ‘’firebase_core’’ package. The ‘’firebase_core’’ package is responsible for initializing and configuring Firebase services in our Flutter application. It provides a way to set up Firebase before we can use other Firebase services like authentication, database, and more. This line makes the classes and functions from the `firebase_core` package available for use in our code.

```
import 'package:firebase_auth/firebase_auth.dart';
```

This line imports the components from the ‘’firebase_auth’’ package. The ‘’firebase_auth’’ package offers Firebase authentication services, allowing us to handle user authentication in our application. It provides functionalities to manage user registration, login, password reset, and other authentication-related tasks. By importing this package, we gain access to the classes and methods needed to implement Firebase-based authentication in our application.

In summary, these import statements have enabled us to obtain the necessary classes and methods from the ‘’firebase_core’’ and ‘’firebase_auth'' packages, allowing us to configure and use Firebase services, including authentication, in our Flutter application.

```
WidgetsFlutterBinding.ensureInitialized();
```

Cette ligne ensures that Flutter's bindings for widgets are initialized. This needs to be done before initializing Firebase.

```
await Firebase.initializeApp();
```

This line initializes Firebase in the application. The call to ‘’Firebase.initializeApp()’’ is marked with ‘’await’’, which means the program will wait for initialization to complete before continuing. This is important as other

parts of the application might depend on Firebase. Initializing Firebase prepares Firebase services for use, such as authentication, database, storage.

III. Result of the interfaces

The following figures depict the user interfaces of the application that are ready at this stage.

Figure 18: first interface

Figure 19: Welcome interface

Figure 20: Get start interface

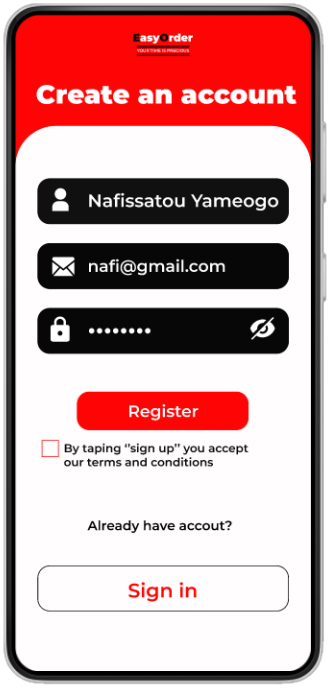


Figure 25: Register interface

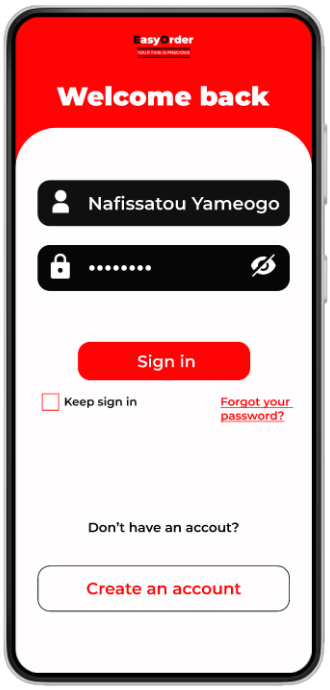


Figure 24: Sign in interface

Figure 26: Home page interface

Figure 23: Categories interface

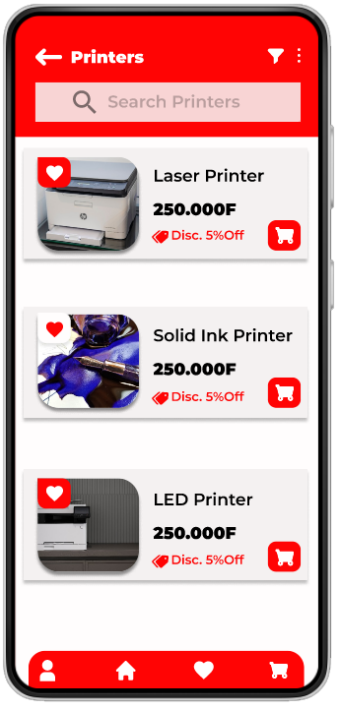


Figure 22: Printers category page interface

Figure 21: Product details interface

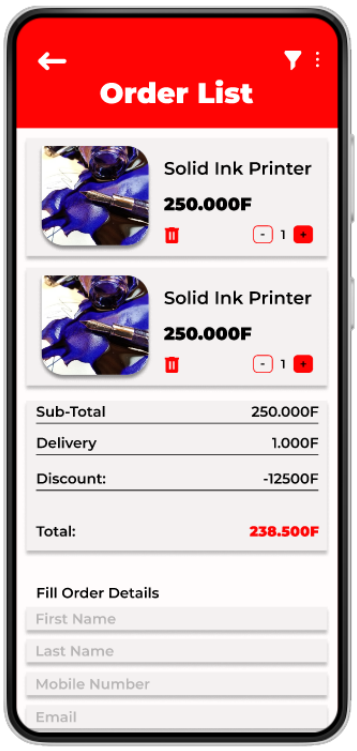


Figure 30: Details for an order(1)

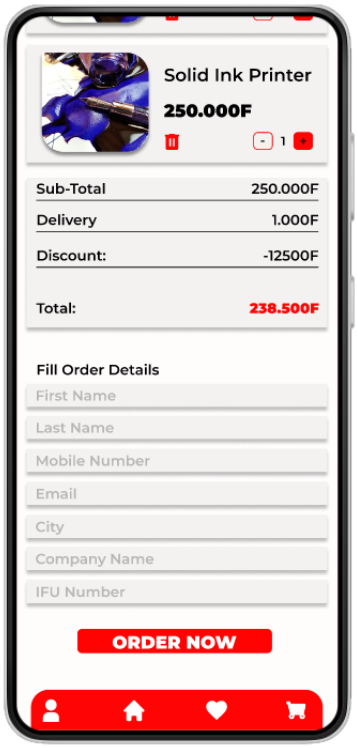


Figure 31: Details for an order(2)

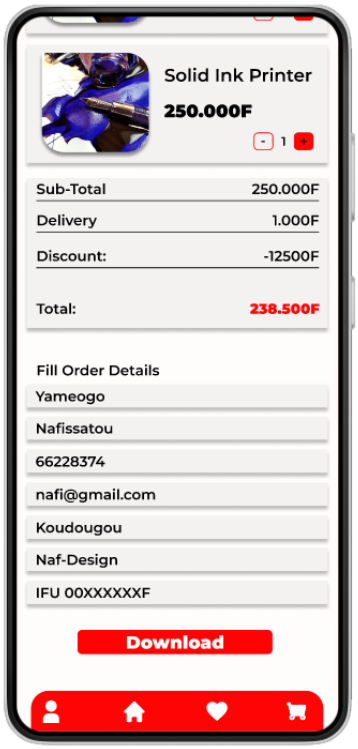


Figure 29: Details for an order(3)

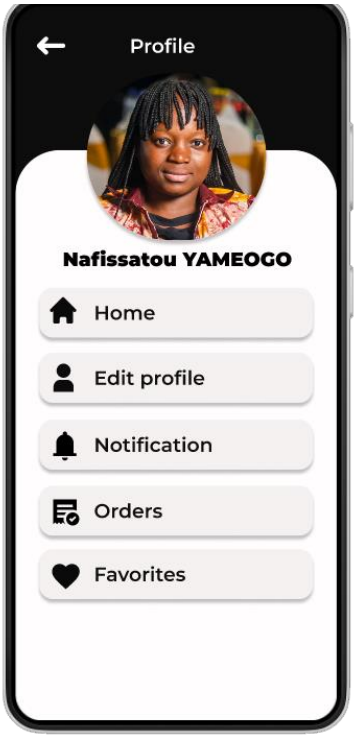


Figure 28: Profile interface

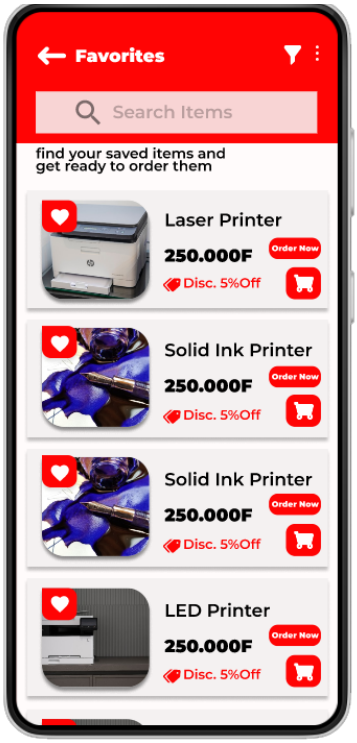


Figure 27: Favorites page

CONCLUSION

In conclusion, the development of the EasyOrder application holds significant importance for GICOME TECHNOLOGY, which seeks to modernize and enhance its process of selling computer consumables. Throughout this project, we have taken into account the specific needs of the company and designed a tailor-made solution to address these needs.

The EasyOrder application offers GICOME TECHNOLOGY the opportunity to establish an online ordering system that simplifies and speeds up the purchasing process for its customers. Users can now browse the product catalog, add items to their cart, and place orders with just a few clicks. This digital solution enables the company to reach a broader audience, enhance its customers' shopping experience, and stand out in a competitive market.

By developing this application, we have also paved the way for new perspectives and opportunities for GICOME TECHNOLOGY. Here are some future prospects that could be considered:

- Online Payment Integration: Add the ability for customers to make online payments, providing more convenience and speed in the purchasing process.
- Real-Time Order Tracking: Integrate a real-time order tracking system to allow customers to monitor the status of their orders from within the application.
- Loyalty Program: Implement a loyalty program to reward regular customers and encourage brand loyalty.
- Data Analysis: Utilize data collected through the application to analyze customer preferences, sales trends, and optimize inventory.
- Collaboration with Suppliers: Establish partnerships with suppliers to improve product sourcing and expand the range of products offered.

In summary, the EasyOrder application represents an innovative solution for GICOME TECHNOLOGY, enabling them to enter the era of e-commerce while providing an enhanced shopping experience to their customers. The future prospects are numerous and offer significant potential to expand the application's functionalities and increase the operational efficiency of the company.

BIBLIOGRAPHY

APPENDICES

Appendix I : Security of application

Security is a critical element in the development and deployment of applications, especially when dealing with sensitive data such as user information and payments, as in the case of the EasyOrder application. For the security of our application, several aspects will be considered:

Authentication and Access Management : Ensure that your application has a strong authentication system. Users should have unique credentials (strong usernames and passwords) to access their accounts. For administrator users, consider implementing multi-factor authentication if possible.

Communication Security : Use security protocols like HTTPS to encrypt communications between users and the server. This protects data in transit from interception.

Data Management : Store data securely. Use secure databases and implement encryption mechanisms for sensitive data such as passwords.

Protection Against Attacks : Implement safeguards against common attacks such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF) attacks.

Payment Security : As our application handles online payment, we will use secure payment gateways that comply with Payment Card Industry Data Security Standard (PCI DSS) standards.

Session Security : Use secure session management mechanisms to prevent unauthorized access to user accounts.

Security Testing : Regularly perform security testing, such as penetration testing, to identify and address vulnerabilities in your application.

Staff Training : Educate your developers and team members about security best practices and data protection.

Legal Compliance : Ensure that your application complies with data protection regulations, such as the General Data Protection Regulation (GDPR) in Europe.

Appendix II : Cost evaluation

The COCOMO (**COnstructive COst MOdel**) estimation model was introduced in 1981 by Barry Boehm. This project cost estimation model is considered "constructive" as it aims to better account for software complexity and thus provide a more accurate project estimation. The model's goal is to mitigate budget errors and delivery delays, which are common challenges in the software development industry.

Appendix III : Working folder on VS Code

This figure shows the project working folder on VS Code

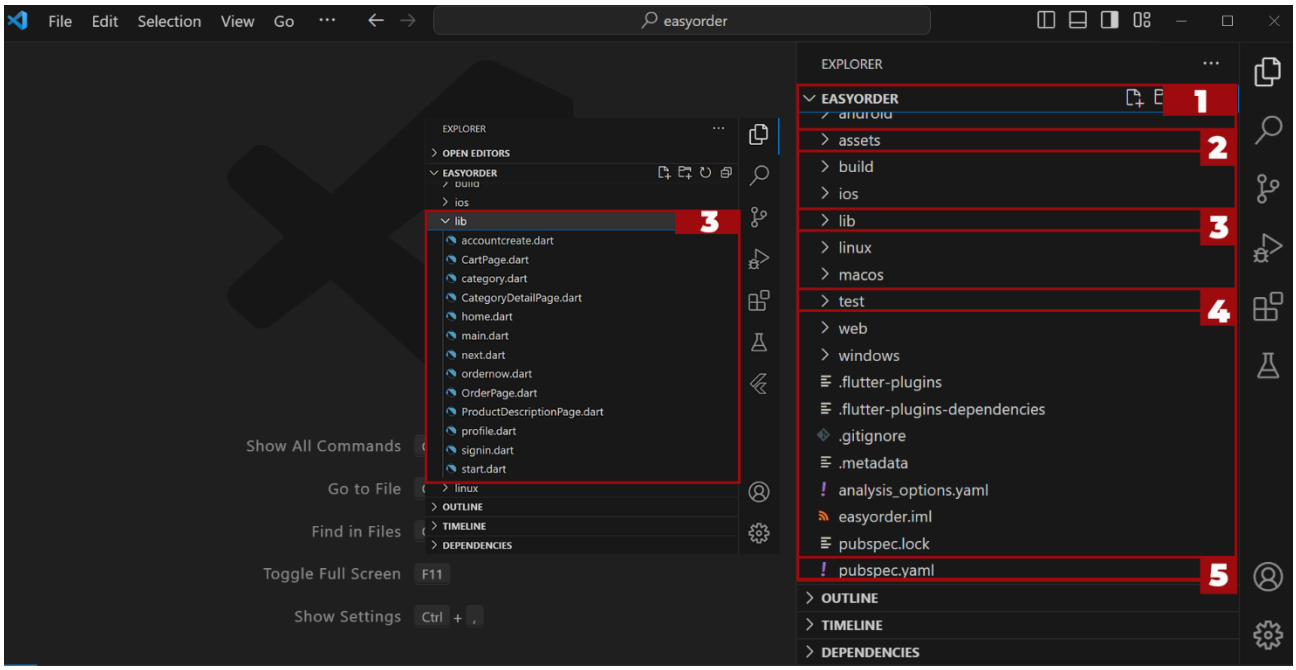


Figure 32: project working folder on Vs Code

- Panel 1 : This is the folder that contains the created project
- Panel 2 : The folder that contains all the resources (images, icons) of the project
- Panel 3 : This is where the files created throughout the project are stored. It is the heart of our application. It contains files such as main.dart, which is used to start the Dart code of the application, home.dart, which holds the source code of the home page, the sign.dart file, which contains the login code, and many other files.
- Panel 4 : It contains the files needed for running automated tests
- Panel 5 : It containd dependencies and resources of project.

Appendix IV : The project's value proposition

The value proposition of the EasyOrder project for GICOME TECHNOLOGIES revolves around modernizing and optimizing the process of ordering computer consumables. Here's how our project idea is innovative and what its value proposition is :

Innovation :

The EasyOrder application represents a significant innovation for GICOME TECHNOLOGIES, as it introduces a modern technological solution where there was no digital platform previously to manage orders for computer consumables. Digitizing this process brings efficiency and simplicity, offering an improved experience for both the company and its customers.

Value Proposition :

- **Simplified Orders** : EasyOrder radically simplifies the process of ordering computer consumables by allowing customers to browse the online catalog, select desired products, and place orders quickly and conveniently, 24/7.
- **Time and Effort Saving** : The application eliminates the need for physical visits or complicated communications. Customers can now order what they need with just a few clicks, saving them time and effort.
- **Enhanced Product Visibility** : Computer consumables offered by GICOME TECHNOLOGIES are showcased in a structured manner within the application, enabling customers to easily explore available options and make informed decisions.
- **Order Accuracy**: By reducing manual entry and enabling customers to specify their needs digitally, EasyOrder helps minimize order errors, ensuring that customers receive exactly what they are looking for.
- **Growth Opportunity** : The application offers GICOME TECHNOLOGIES the opportunity to expand its geographical reach and reach a broader audience by providing an online ordering service accessible to a wide range of customers.

- **Competitive Differentiation** : By offering a modern and efficient ordering method, the company stands out in a competitive market by showcasing its commitment to innovation and customer satisfaction.

In summary, the value proposition of EasyOrder for GICOME TECHNOLOGIES lies in its ability to modernize the process of ordering computer consumables, offering simplicity, speed, accuracy, and user-friendliness for customers, while allowing the company to stay competitive and seize new growth opportunities.