



RÉPUBLIQUE  
TUNISIENNE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE

Ministry of Higher Education and Scientific Research  
Directorate-General for Technological Studies

\*\*\*



Higher Institute of Technological Studies of

Kélibia

IT Technology Department

## END-OF-STUDIES INTERNSHIP REPORT

Developed for the graduation of

**Applied Bachelor's Degree in Computer Technology**

Route

Information Systems Development

Subject

---

**Design and implementation of an AI based Mobile  
App for pc builds recommendations**

---

Host Organization

**Mega Pc Plus**

**Directed by :** Grira Ismail

Mohamed Omar Ghilene

Ref :

---

---

**Supervised by :** Saddem Khalil Trabelsi, Manager

Mohamed Anis Mejri, Lecturer Technologist at ISET Kélibia

Année Universitaire : 2023/2024

## **Acknowledgement**

We would like to express our deep gratitude to our supervisors; their support, guidance, help, and wise advice have been invaluable.

We wish to extend our sincere thanks to all the professors in the IT department of ISET Kélibia as well as the entire team at Mega PC Plus. Their expertise, guidance, and responses to our questions have greatly contributed to enriching our reflections.

We also want to express our gratitude to our university supervisor, Mr. Mohamed Anis Mejri, for his availability, invaluable support, and insightful advice. His kind presence has been of immense help throughout our work.

Our warmest thanks also go to our company supervisor, Mr. Saddem Khalil Trabelsi, for his kind supervision and for his wise directions and recommendations. His attentive support has been crucial to the success of our project.

Finally, we wish to express our gratitude to all those who have contributed in any way to the realization of this work. Their participation and support have been precious, and we sincerely thank them.

# SUMMARY

<b>General Introduction.....</b>	<b>1</b>
<b>Chapter 1 Project Overview .....</b>	<b>1</b>
Introduction .....	1
1. Host Organization Introduction.....	1
2. Context and Motivations .....	4
Conclusion .....	6
<b>Chapter 2 Methodological Approach .....</b>	<b>7</b>
Introduction .....	7
1. System Architecture Approach.....	7
2. AI Modal Approach.....	10
Conclusion .....	12
<b>Chapter 3 Analysis and Requirements Specification .....</b>	<b>13</b>
Introduction .....	13
1. Requirements Specification .....	13
2. Functional requirements .....	13
3. Non-functional requirements.....	15
4. Use Cases Overview.....	16
5. User Interface Design .....	43
Conclusion .....	46
<b>Chapter 4 Conception .....</b>	<b>47</b>
Introduction .....	47
1. Architectural Design.....	47
2. Modeling Language .....	48
3. Static Design.....	49
4. Dynamic Design .....	50
Conclusion .....	70
<b>Chapter 5 Implementaion.....</b>	<b>71</b>

Introduction .....	71
1. Technology Stack .....	71
2. Development Environment.....	78
3. Application Interfaces .....	82
4. Admin interfaces .....	100
Conclusion.....	107
<b>Overall conclusion .....</b>	<b>108</b>
<b>Bibliography .....</b>	<b>cix</b>
<b>Webography .....</b>	<b>cx</b>
<b>List of abbreviations .....</b>	<b>cxi</b>

# List of Figures

<b>Figure 1-1</b> Company Organizational Chart .....	2
<b>Figure 1-2</b> Gantt Chart of Project Timeline.....	3
<b>Figure 2-1</b> PC Builder System Architecture .....	8
<b>Figure 2-2</b> Vanilla Retrieval-Augmented Generation Workflow.....	11
<b>Figure 3-1</b> Global Use Case Diagram.....	16
<b>Figure 3-2</b> " Search for products " use case diagrams .....	30
<b>Figure 3-3</b> " Browse new products " use case diagram .....	33
<b>Figure 3-4</b> " Browse Hot Deals " use case diagram.....	34
<b>Figure 3-5</b> " Browse discounts codes " use case diagram .....	36
<b>Figure 3-6</b> " Manage Products " use case diagram .....	37
<b>Figure 3-7</b> " Manage Notifications " use case diagram .....	39
<b>Figure 3-8</b> " Manage Users " use case diagram .....	40
<b>Figure 3-9</b> " Manage Categories " use case diagram.....	42
<b>Figure 3-10</b> PC Builder System Wireframes .....	44
<b>Figure 4-1</b> MVC Architector.....	47
<b>Figure 4-2</b> Unified Modeling Language.....	48
<b>Figure 4-3</b> Global Class Diagram.....	49
<b>Figure 4-4</b> "Authentication" sequence diagram.....	51
<b>Figure 4-5</b> " Sign Up " sequence diagram. ....	51
<b>Figure 4-6</b> " Log Out " sequence diagram. ....	52
<b>Figure 4-7</b> " Update Password " sequence diag diagram.....	52
<b>Figure 4-8</b> " Reset Password " sequence diagram. ....	53
<b>Figure 4-9</b> "Save Build" sequence diagram.....	53
<b>Figure 4-10</b> "Share Build" sequence diagram. ....	54
<b>Figure 4-11</b> "Delete Build" sequence diagram. ....	54
<b>Figure 4-12</b> "CreateManualBuild" sequence diagram.....	55
<b>Figure 4-13</b> " CreateIaBuild " sequence diagram.....	56
<b>Figure 4-14</b> " Get Build " sequence diagram.....	57
<b>Figure 4-15</b> " Get Categories " sequence diagram.....	57
<b>Figure 4-16</b> " Get Porfile " sequence diagram.....	58
<b>Figure 4-17</b> " Get Notifications " sequence diagram.....	58
<b>Figure 4-18</b> " Get Products " sequence diagram.....	59
<b>Figure 4-19</b> " Add Product" sequence diagram.....	59

<b>Figure 4-20</b> "Update Product" sequence diagram.....	60
<b>Figure 4-21</b> Delete Product" sequence diagram.....	60
<b>Figure 4-22</b> "GetProducts" sequence diagram.....	61
<b>Figure 4-23</b> "Add Notification" sequence diagram.....	62
<b>Figure 4-24</b> "Update Notification" sequence diagram.....	62
<b>Figure 4-25</b> "Delete Notification" sequence diagram.....	63
<b>Figure 4-26</b> "GetNotifications" sequence diagram.....	64
<b>Figure 4-27</b> "Add User" sequence diagram.....	65
<b>Figure 4-28</b> "Update User" sequence diagram.....	65
<b>Figure 4-29</b> "Delete User" sequence diagram.....	66
<b>Figure 4-30</b> "GetUsers" sequence diagram.....	67
<b>Figure 4-31</b> "Add Category" sequence diagram.....	68
<b>Figure 4-32</b> " Update Category " sequence diagram.....	68
<b>Figure 4-33</b> "Delete Category" sequence diagram.....	69
<b>Figure 4-34</b> "GetCategories" sequence diagram.....	70
<b>Figure 5-1</b> Splash screen.....	82
<b>Figure 5-2</b> Getting Started Screen .....	83
<b>Figure 5-3</b> Login screen.....	84
<b>Figure 5-4</b> Signup screen .....	85
<b>Figure 5-5</b> Terms and Services Screen .....	86
<b>Figure 5-6</b> Recover Account Screen.....	87
<b>Figure 5-7</b> Reset Password Screen .....	88
<b>Figure 5-8</b> Home screen .....	89
<b>Figure 5-9</b> SideBar Menu .....	90
<b>Figure 5-10</b> "Search" Interface.....	91
<b>Figure 5-11</b> Profile Page .....	92
<b>Figure 5-12</b> Notification detail Page .....	93
<b>Figure 5-13</b> Component details Page.....	94
<b>Figure 5-14</b> AI Builder Page .....	95
<b>Figure 5-15</b> Manual Builder Interface .....	96
<b>Figure 5-16</b> Full Build Page .....	97
<b>Figure 5-17</b> Saving Build Interface .....	98
<b>Figure 5-18</b> sharing Build Interface.....	99
<b>Figure 5-19</b> Saved Builds Interface .....	100
<b>Figure 5-20</b> Login Interface.....	101

<b>Figure 5-21</b> Users List Interface .....	101
<b>Figure 5-22</b> Add User Interface .....	102
<b>Figure 5-23</b> Modify User Interface .....	102
<b>Figure 5-24</b> Products List Interface .....	103
<b>Figure 5-25</b> Add Product Interface .....	103
<b>Figure 5-26</b> Modify Product Interface .....	104
<b>Figure 5-27</b> Categories List Interface .....	104
<b>Figure 5-28</b> Add Category Interface .....	105
<b>Figure 5-29</b> Modify Category Interface .....	105
<b>Figure 5-30</b> Notifications List Interface .....	106
<b>Figure 5-31</b> Add Notification Interface .....	106
<b>Figure 5-32</b> Modify Notification Interface .....	107

# List of Tables

<b>Table 3-1</b> Functional Requirements Table.....	15
<b>Table 3-2</b> Text description of "Register" use cases. ....	18
<b>Table 3-3</b> Text description of "Sign In" use cases.....	19
<b>Table 3-4</b> Table 3 4 Text description of "Sign in Admins" use cases. ....	19
<b>Table 3-5</b> Text description of " Recover password " use cases. ....	20
<b>Table 3-6</b> Text description of "Manage Profile" use cases.....	22
<b>Table 3-7</b> Text description of " Use AI Builder" use cases. ....	24
<b>Table 3-8</b> Text description of "Create Custom PC Build" use cases.....	26
<b>Table 3-9</b> Text description of " View Saved Builds " use cases.....	27
<b>Table 3-10</b> Text description of "Browse Products By Categories " use cases. ....	29
<b>Table 3-11</b> Text description of " Search for products " use cases. ....	31
<b>Table 3-12</b> Text description of " Browse New Events " use cases.....	32
<b>Table 3-13</b> Text description of " Browse new products " use cases.....	34
<b>Table 3-14</b> Text description of " Browse hot deals" use cases. ....	35
<b>Table 3-15</b> Text description of " Browse hot deals" use cases. ....	37
<b>Table 3-16</b> Text description of " Manage products" use cases.....	38
<b>Table 3-17</b> Text description of " Manage notifications" use case. ....	40
<b>Table 3-18</b> Text description of " Manage Users" use case. ....	41
<b>Table 3-19</b> Text description of " Manage Categories" use case. ....	43
<b>Table 5-1</b> Programming Language Table .....	72
<b>Table 5-2</b> Frameworks and Library Table .....	74
<b>Table 5-3</b> Database Technology Table .....	75
<b>Table 5-4</b> Cloud Services Table.....	75
<b>Table 5-5</b> Other Technologies Table .....	77
<b>Table 5-6</b> Hardware Environment Table .....	78
<b>Table 5-7</b> Development Tools Table .....	80
<b>Table 5-8</b> Version Control System Table .....	81

# **General Introduction**

In today's world, businesses are always looking to use new technologies to grow and stay ahead in the market. This means finding smart tech solutions to make their work better, increase productivity, and meet their financial targets. Using new tech helps businesses move forward and reach more customers.

Our final year project with "Mega Pc Plus" focuses on making a mobile app with Flutter and a backend system with Node.js TypeScript. The main goal is to build a platform that can put together computer builds based on "Mega Pc Plus's" own data. We use a special AI language called RAG (Retrieval-Augmented Generation) to make sure our app can work well with the company's data. This helps us offer a service that is both personalized and efficient.

For hosting our Node.js app, we use AWS Lightsail. This ensures our project is safe and works well. By choosing these technologies, we aim to meet "Mega Pc Plus's" needs now and in the future.

This report is structured into four main chapters:

## **Chapter 1: Project Overview**

- Introduces "Mega Pc Plus" and highlights the project's necessity.

## **Chapter 2: Methodological Approach**

- Discusses the methodology used in the project.

## **Chapter 3: Analysis and Requirements Specification**

- Details the project requirements and analysis process.

## **Chapter 4: Conception**

- Explores the project's conception phase.

## **Chapter 5: Implementation**

- Focuses on the practical execution of the project.

# Chapter 1 Project Overview

## Introduction

This first chapter introduces MegaPc, where we did our project. We'll explain the advanced technology and key terms used in our project. Understanding these basics is important for getting the full picture of what we're doing. We'll review what's currently out there, what's lacking, and how our project offers a solution. We end with the work method we chose, shown by a Gantt chart.

## 1. Host Organization Introduction

### 1.1 About the Organization

MegaPc stands out in Tunisia as a premier destination for esports and gaming equipment, emphasizing competitive gaming and player performance. With its headquarters in Tunis (Aouina), the company boasts an inventory exceeding 2000 products, all set for dispatch within 48 hours. MegaPc collaborates with leading brands such as Gigabyte, Aorus, and MSI, offering customers the latest in gaming technology. The organization is committed to excellence in service, managing logistics, and customer support directly or through partnerships with fast delivery services like ARAMEX.



Figure 1-1 Mega Pc Company's logo

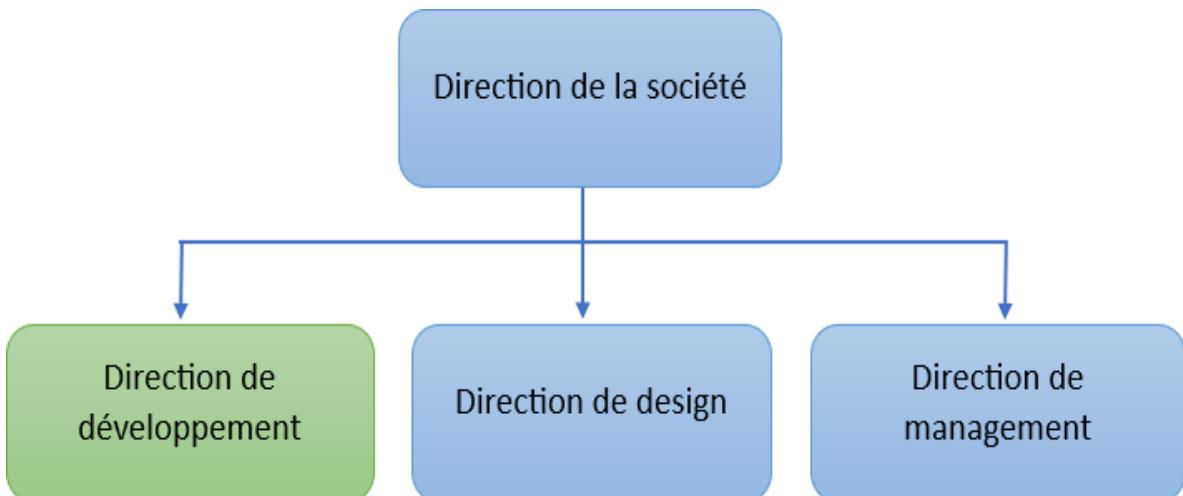
**Services:**

MegaPc.tn offers a wide range of services and products for esports and gaming enthusiasts, including:

- Selling esports and gaming peripherals: Graphics Cards, High-End Processors, Gaming Cases, gaming mice, keyboards, headsets, RGB mouse pads, and gaming controllers for PC or PS5.
- Partnering with leading brands to bring the latest industry innovations.
- Providing quality customer service with internal logistics and fast delivery partnerships.

## 1.2 Organization Chart

Figure 1 shows the organizational chart of Company “Mega Pc Plus” and gives a general overview of its hierarchy.



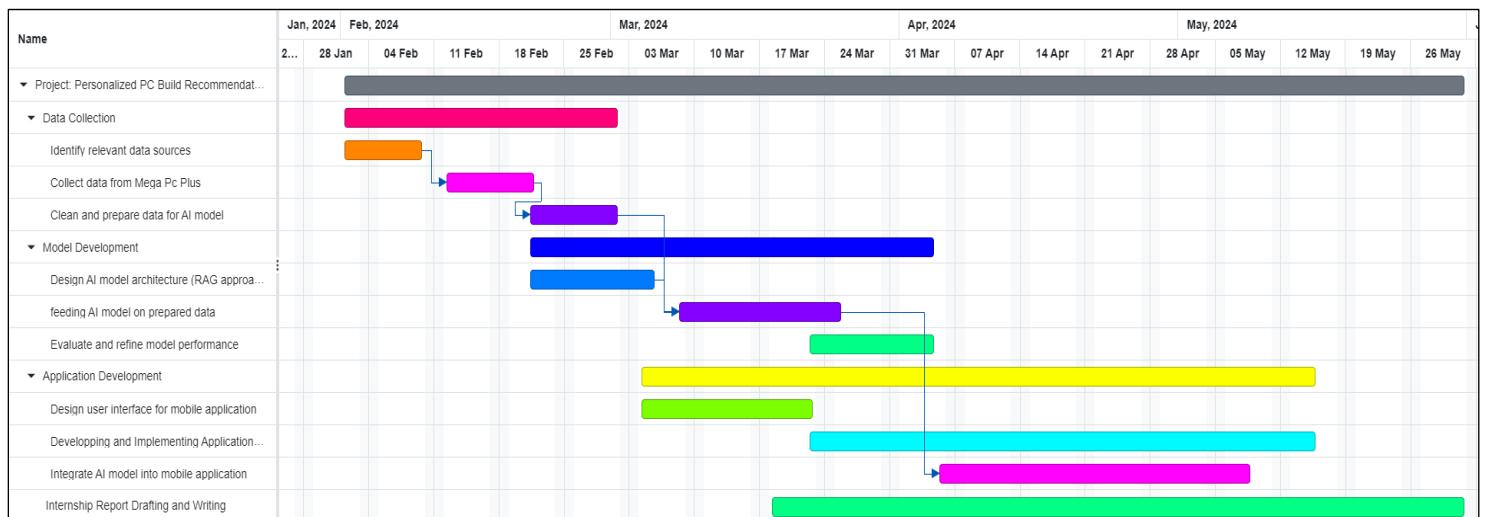
**Figure 1-1 Company Organizational Chart**

## 1.3 Project Overview

Our project enhances "Mega Pc Plus's" customer service by introducing an AI-driven application for personalized PC builds. Recognizing the importance of data in connecting customers with ideal gaming setups, we focused on an innovative solution to simplify the selection and purchasing process. "Mega Pc Plus" has been incorporating advanced technologies, like CRM and ERP systems, to improve customer interactions and sales efficiency. Our application aligns with these advancements, using "Mega Pc Plus's" data to offer customers custom PC builds based on their preferences, budget, and usage. This initiative not only improves the shopping experience but also supports "Mega Pc Plus's" ongoing digital transformation, reinforcing its position in the gaming and esports market..

## 1.4 Project Timeline (Gantt Chart)

The following Gantt chart visually represents the project schedule, including task durations and dependencies, for the Personalized PC Build Recommendation System.



**Figure 1-2 Gantt Chart of Project Timeline**

## 2. Context and Motivations

### 2.1 Market Study

In today's market, personalized shopping experiences are becoming increasingly important, especially in sectors like gaming and PC builds where customer needs and budgets vary widely. A market study was conducted to understand current trends, customer preferences, and gaps in the market. This study helped identify the need for a personalized approach in recommending PC builds.

Currently, there are several solutions available in the market that offer customized PC build recommendations:

- **PC Builder AI:** This platform allows users to input their budget and intended use to receive a recommended PC build.

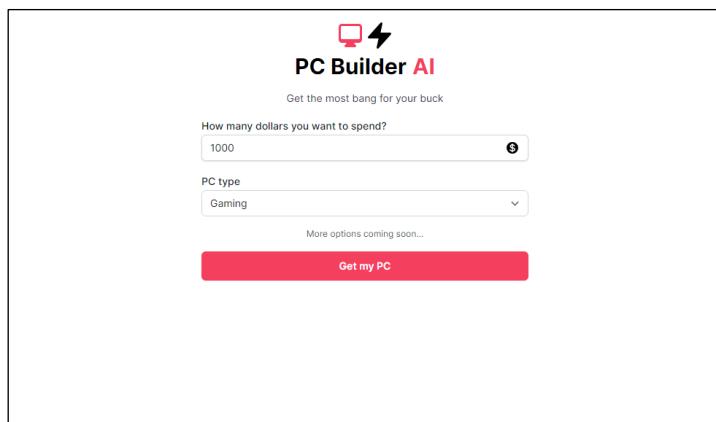
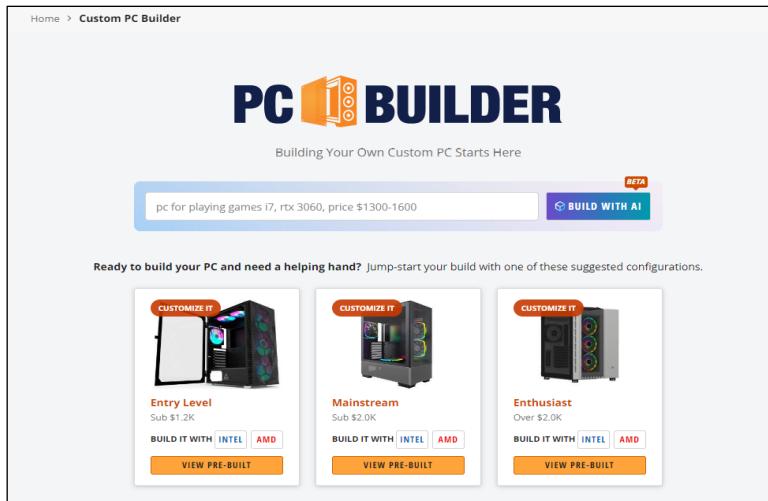


Figure 1-4 PC Builder AI landing page

- **Newegg Custom PC Builder:** Newegg's tool also uses user budget and purpose to suggest PC builds.



**Figure 1-5 Newegg pc builder webpage**

Both of these tools provide a basic level of customization based on user budget and purpose. However, our project adds an additional layer by incorporating user preferences, allowing for further customization and a more tailored build that aligns with individual user needs.

## 2.2 Motivations

"Mega Pc Plus" has recognized the potential of offering customized PC builds to enhance customer satisfaction and loyalty. Traditional methods of product recommendation often fall short in meeting the exact needs of customers, leading to missed opportunities in sales and customer retention. The development of an application that solicits customers' preferences, budget, and intended use, and then suggests a customized PC build, represents a significant step forward. This project is motivated by the need to leverage "Mega Pc Plus's" extensive data on gaming and esports equipment, alongside advanced AI modeling, to deliver highly personalized and relevant product recommendations.

## 2.3 Project Objectives

The main goal of this project is to create a user-friendly application that:

- Gathers input from the customer regarding their preferences, budget, and the purpose of the PC build.
- Utilizes "Mega Pc Plus's" data and AI models to generate a customized PC build that matches the customer's specifications.

- Aims to enhance the shopping experience, thereby increasing customer satisfaction and loyalty.

To achieve this, the project will involve:

- Collecting and analyzing data to understand customer needs and market trends.
- Designing and implementing an AI model capable of making accurate recommendations based on customer inputs.
- Developing a mobile application interface that is easy for customers to use and understand.
- Testing the application to ensure it meets our objectives and refining it based on feedback.

The success of this project will be measured by its ability to accurately predict and meet customer needs, ultimately leading to increased sales and customer retention for "Mega Pc Plus."

## **Conclusion**

In this chapter, we introduced MegaPc, highlighting its role as a premier destination for esports and gaming equipment in Tunisia. We covered the organization's background, services, and structure. We also explored the project's context and motivations, emphasizing the market need for personalized PC build recommendations. Our project aims to enhance customer satisfaction by developing an AI-driven application that provides customized PC builds based on user preferences, budget, and usage. This sets the stage for the detailed exploration of our methodology, tools, and implementation in the subsequent chapters.

# Chapter 2 Methodological Approach

## Introduction

This chapter outlines the methodological approach employed in designing and developing the custom PC build system. A well-defined methodology ensures a structured and efficient development process, fostering the creation of a high-quality and user-centric application.

## 1. System Architecture Approach

### 1.1 Adopted Architecture:

- Client Part: Consists of the creation of the various interfaces of the mobile application as well as the interpretation of the data provided and their displays.
- Server part: This part allows the client part to be fed with data. In addition, it allows the insertion, consultation of data and updating of the client application.

### 1.2 Logical Architecture

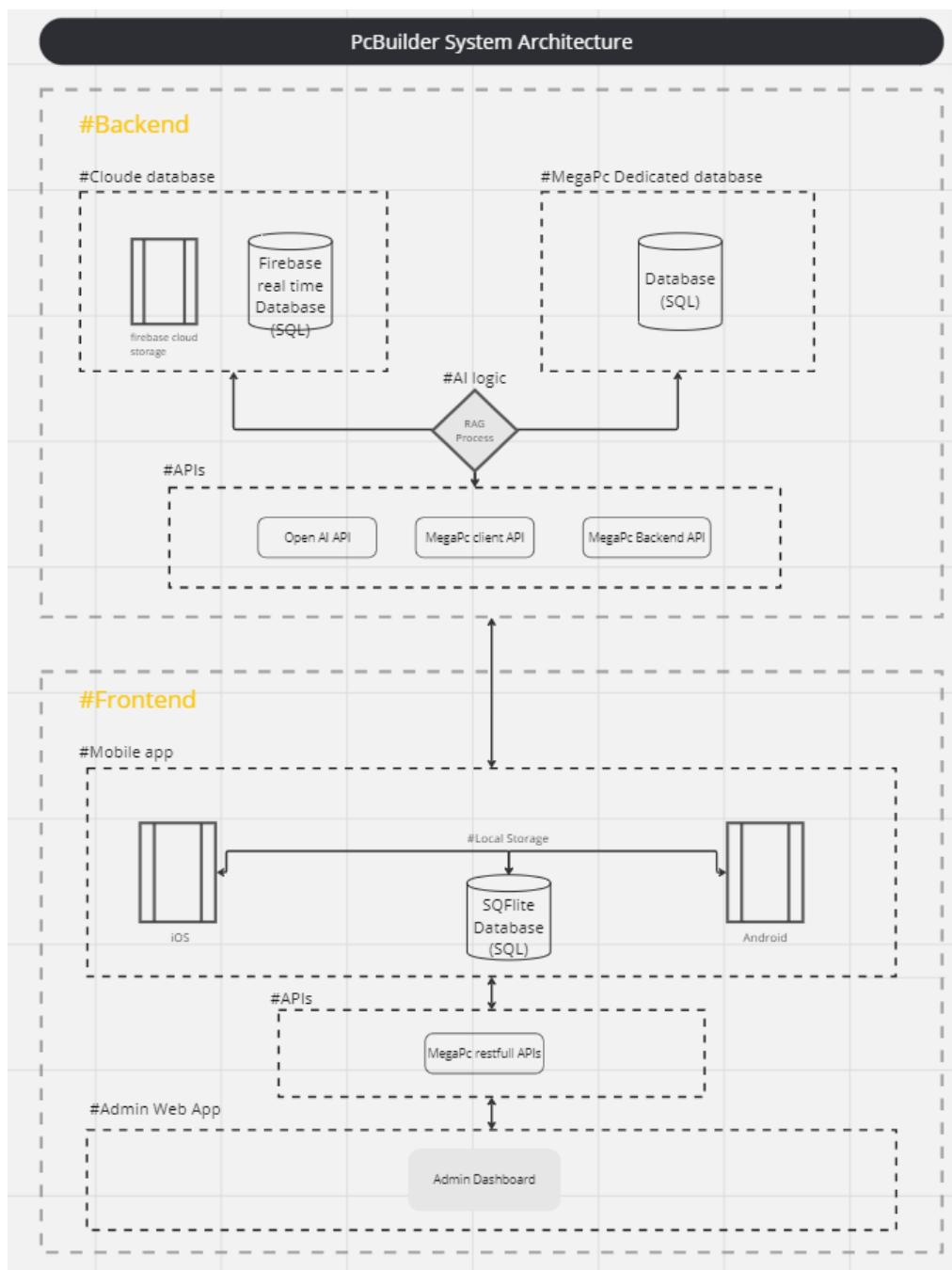
The logical architecture of our system describes how the different parts of the system interact with each other and how they are organized. It is based on a layered model, where each layer has a specific responsibility and communicates only with adjacent layers.

#### 1.2.1 Layers of our logical architecture:

- **Presentation Layer:** This layer is responsible for the user interface of PC Builder. This layer is represented by the interfaces of the mobile application.
- **Business Logic Layer:** This layer is responsible for the application logic and data management. It can be implemented using a server-side programming language. This layer is represented by MegaPc and PcBuilder web services.
- **Data Layer:** This layer is responsible for managing and accessing the application's data. It is implemented using the system's databases.
- **Communication Layer:** This layer is responsible for communicating with other systems, such as web APIs or remote databases. In our system, this is the layer implemented in the mobile application to call and process requests and responses from web services.

### 1.3 Physical Architecture

The physical architecture describes how the different parts of an application or system are implemented and interconnected. Here, we'll explore the physical components and their deployment strategy for the custom PC build system.



**Figure 2-1 PC Builder System Architecture**

## 1.4 Chosen Technologies:

- **Frontend:** The frontend layer will likely utilize a cross platform framework or library (e.g., React, Flutter, etc.) to deliver an interactive user experience. This user interface can be accessed on various devices operating systems.
- **Backend:** The backend layer, responsible for handling complex logic and data processing, uses a server-side programming language (e.g. JavaScript, Java, Node.js) and a web framework (e.g., Spring Boot, Express.js) to handle server-side operations and API requests.
- **Database:** A database management system (DBMS) will be essential for storing product information, user data, and potentially build configurations. The type of database chosen (e.g., SQL or NoSQL) will depend on factors such as data structure, scalability requirements, and desired performance characteristics.
- **Cloud Storage:** A cloud storage solution Amazon S3 will be implemented to store user-uploaded content or product images. This frees up space on the primary server and facilitates scalability.
- **APIs:** APIs (Application Programming Interfaces) act as intermediaries, enabling communication between different parts of the system. RESTful APIs are commonly used architectural styles for APIs, and the system showcases a RESTful API used for communication between the frontend and backend.

## 1.5 Deployment Strategy:

The specific deployment strategy will depend on factors such as project scale, performance requirements, and budget. However, common practices include deploying the frontend to a Content Delivery Network (CDN) for improved performance and global accessibility. The backend components and databases might be deployed on a cloud platform Amazon Web Services to benefit from scalability, reliability, and managed services.

## 1.6 Security Considerations:

Implementing robust security measures is crucial to protect user data and system integrity. This might involve securing API access points with authentication and authorization mechanisms, encrypting sensitive data at rest and in transit, and staying updated on potential security vulnerabilities.

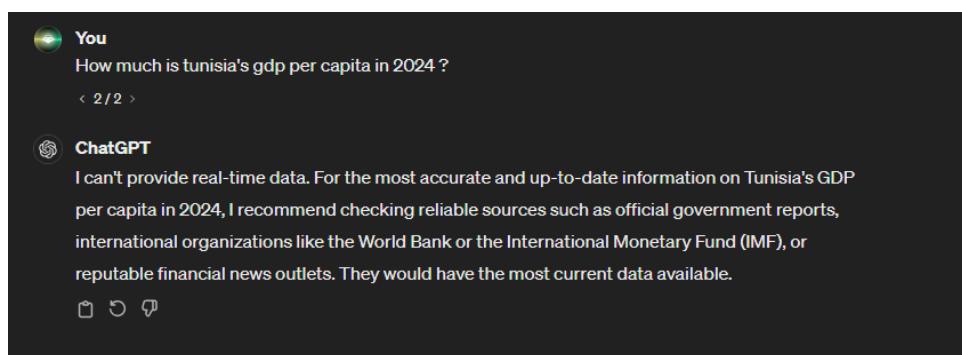
## 2. AI Modal Approach

To make this project come to life there has been a lot of debate around whether fine-tuning or Retrieval-Augmented Generation (RAG) is more suited for this.

- **Fine-tuning** a model is the process of optimizing a pre-trained machine learning model by adjusting its parameters using task-specific data, enabling it to perform a specific task more accurately and efficiently.
- **Retrieval-Augmented Generation (RAG)** is a methodology designed to provide LLMs with additional information from an external knowledge source. This allows them to generate more accurate and contextual answers while reducing hallucinations.

### 2.1 Problem

State-of-the-art LLMs are trained on large amounts of data to achieve a broad spectrum of general knowledge stored in the neural network's weights (parametric memory). However, prompting an LLM to generate a completion that requires knowledge that was not included in its training data, such as newer, proprietary, or domain-specific information, can lead to factual inaccuracies (hallucinations), as illustrated in the following screenshot.



**Fig 2.0.1ChatGPT's answer to the question.**

### 2.2 Solution

Traditionally, neural networks are adapted to domain-specific or proprietary information by fine-tuning the model. Although this technique is effective, it is also compute-intensive, expensive, and requires technical expertise, making it less agile to adapt to evolving information.

In **2020**, there were introduced a more flexible technique called Retrieval-Augmented Generation (RAG)

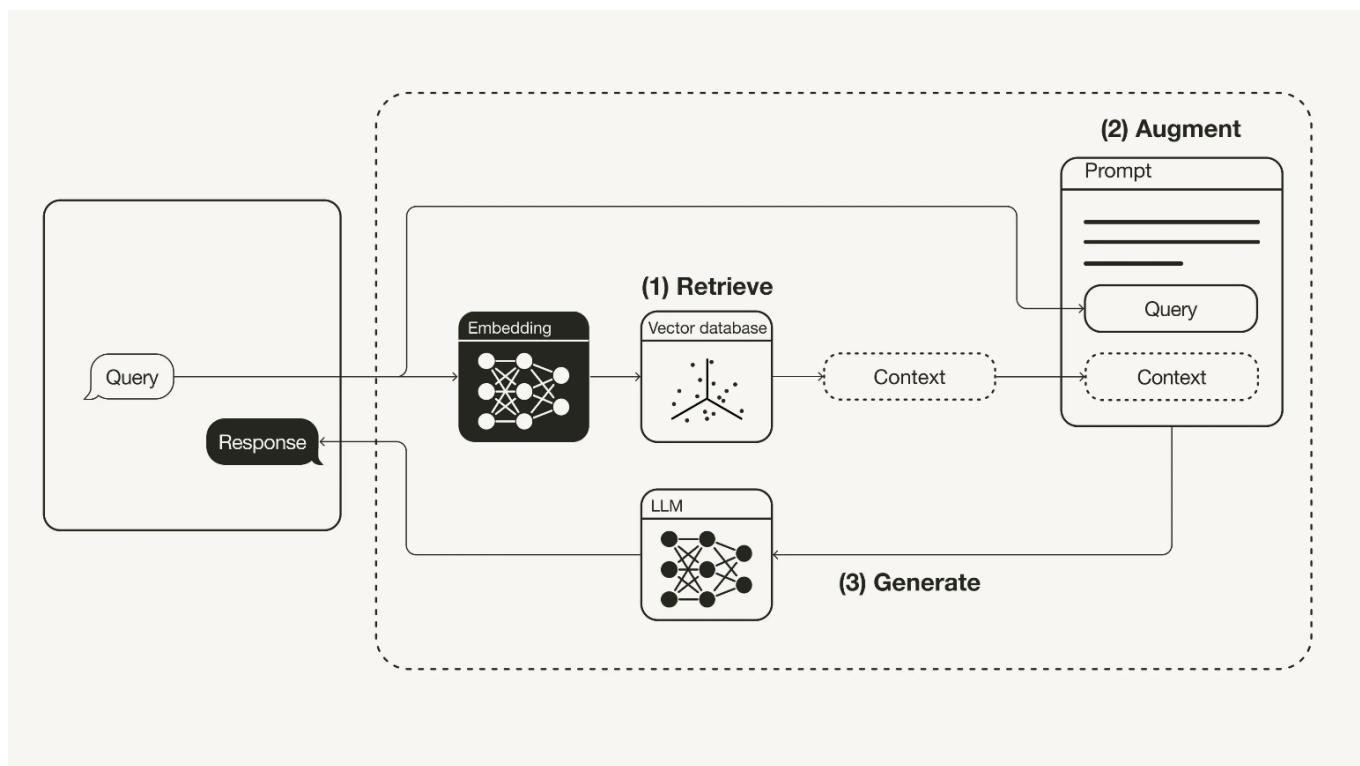
“In simple terms, RAG is to LLMs what an open-book exam is to humans. In an open-book exam, students are allowed to bring reference materials, such as textbooks or notes, which they can use to look up relevant information to answer a question. The idea behind an open-book exam is that the test focuses on the students’ reasoning skills rather than their ability to memorize specific information.”

- Comparison by **JJ** during the Kaggle — LLM Science Exam competition.

Similarly, the knowledge is separated from the LLM’s reasoning capability and stored in an external knowledge source, which can be easily accessed and updated.

### 2.3 Vanilla Retrieval-Augmented Generation Workflow

The vanilla RAG workflow is illustrated below:



**Figure 2-2 Vanilla Retrieval-Augmented Generation Workflow**

- **Retrieve**

The user query is used to retrieve relevant context from an external knowledge source.

Traditionally as in vanilla RAG Workflow the user query is embedded with an embedding model into the same vector space as the additional context in the vector database. This allows to perform a similarity search, and the top k closest data objects from the vector database are returned.

For this step, since we have access to a structured database and no semantic search is required, we adjusted this step of the RAG process and performed a filtering algorithm on the MegaPc database to retrieve pc components based on the user query.

- **Augment**

The user query and the retrieved additional context are stuffed into a prompt template.

- **Generate**

Finally, the retrieval-augmented prompt is fed to the LLM.

To augment the prompt with the additional context, you need to prepare a prompt template. Generative artificial intelligence (AI) systems are designed to generate specific outputs based on the quality of provided prompts. **Prompt engineering** helps generative AI models better comprehend and respond to a wide range of queries, it's the process of structuring an instruction that can be interpreted and understood by a generative AI model.

## Conclusion

In conclusion, this chapter explored fine-tuning vs. RAG for our project. RAG's flexibility and external knowledge source make it ideal. We described the RAG workflow, including our adaptation for retrieving PC components from MegaPc's database. Integrating RAG with an LLM allows for accurate, contextual recommendations for personalized PC builds.

# Chapter 3 Analysis and Requirements Specification

## Introduction

Requirements specification is a crucial step in the project development cycle as it highlights the needs of different users. In this chapter, we will proceed with the analysis of both functional and non-functional requirements.

### 1. Requirements Specification

To gain a detailed and clearer understanding of our application, we will present both the functional and non-functional requirements, as well as the actors within our system.

### 2. Functional requirements

We categorize the functional requirements by actor in Table 3-1.

Actor	functional requirements	Description
User	Authenticate	<ul style="list-style-type: none"> <li>- <b>Sing up:</b> Allows the user to authenticate to access the application's features.</li> </ul>
		<ul style="list-style-type: none"> <li>- <b>Login:</b> Logs the user into the application to access their account.</li> </ul>
		<ul style="list-style-type: none"> <li>- <b>Reset password:</b> Resets the user's password to regain access to the account</li> </ul>
	Manage Profile	<ul style="list-style-type: none"> <li>- <b>Edit password:</b> Enables the user to modify their password.</li> </ul>

Admin		<ul style="list-style-type: none"> <li>- <b>View Profile:</b> Allows the user to view their profile information such as their name, email address, phone number, etc.</li> </ul>
	Create a Pc Build	<ul style="list-style-type: none"> <li>- <b>Create Custom PC Build:</b> Allows the user to manually select components to assemble a custom PC build.</li> </ul>
		<ul style="list-style-type: none"> <li>- <b>Generate AI-Powered Build:</b> Utilizes a pre-trained model to generate a custom build based on user preferences, budget, and intended purpose.</li> </ul>
	Manage Builds	<ul style="list-style-type: none"> <li>- <b>Manage Builds:</b> enables users to effectively organize and control their custom PC builds within the application. Users can save, share, and delete their custom PC builds.</li> </ul>
	Manage Products	<ul style="list-style-type: none"> <li>- <b>Manage Products:</b> enables admins to effectively manage and control the products within the admin panel. Admins can view, add, edit, and delete products.</li> </ul>
	Manage Users	<ul style="list-style-type: none"> <li>- <b>Manage Users:</b> enables admins to effectively manage and control the users within the admin panel. Admins can view, add, edit, and delete users.</li> </ul>

	Manage Categories	<ul style="list-style-type: none"> <li>- <b>Manage Categories:</b> enables admins to effectively manage and control the categories within the admin panel. Admins can view, add, edit, and delete categories.</li> </ul>
	Push Notifications	<ul style="list-style-type: none"> <li>- <b>Push Notifications:</b> enables admins to effectively sends notification within the admin panel. Admins can add notifications.</li> </ul>

**Table 3-1 Functional Requirements Table**

### 3. Non-functional requirements

Non-functional requirements are of great importance because they have an indirect impact on the user's results and performance. It is therefore essential not to neglect them and to meet the following requirements:

1. **Security:** The application will need to be secured by authentication and an encrypted password.
2. **Reliability:** The data provided by the app must be safe and trustworthy.
3. **Ergonomics:** the application must be adapted to the user with a user-friendly interface, i.e. simple and easy to handle
4. **Speed:** the application saves time for the user since it promotes direct access as soon as the ads are made. Minimal response time.
5. **Maintainability:** The source code of this application should be clean with clear comment lines. We should add a log file to monitor the usage of the application and make maintenance easier.
6. **Extensibility:** As part of this work, the application will need to be extensible, i.e. there may be an opportunity to add or modify new features.
7. **Performance:** Ensure through cross-platform flutter and MERN stack (Framework

composed of the stack of MongoDB, Express.js, React.js and NodeJs.)

## 4. Use Cases Overview

### 4.1 Global Use Case Diagram

A use case is a methodology used in system analysis to identify, organize, and clarify system requirements. Figure 3-1 illustrates the general use case diagram that encompasses the basic functionalities of our project.

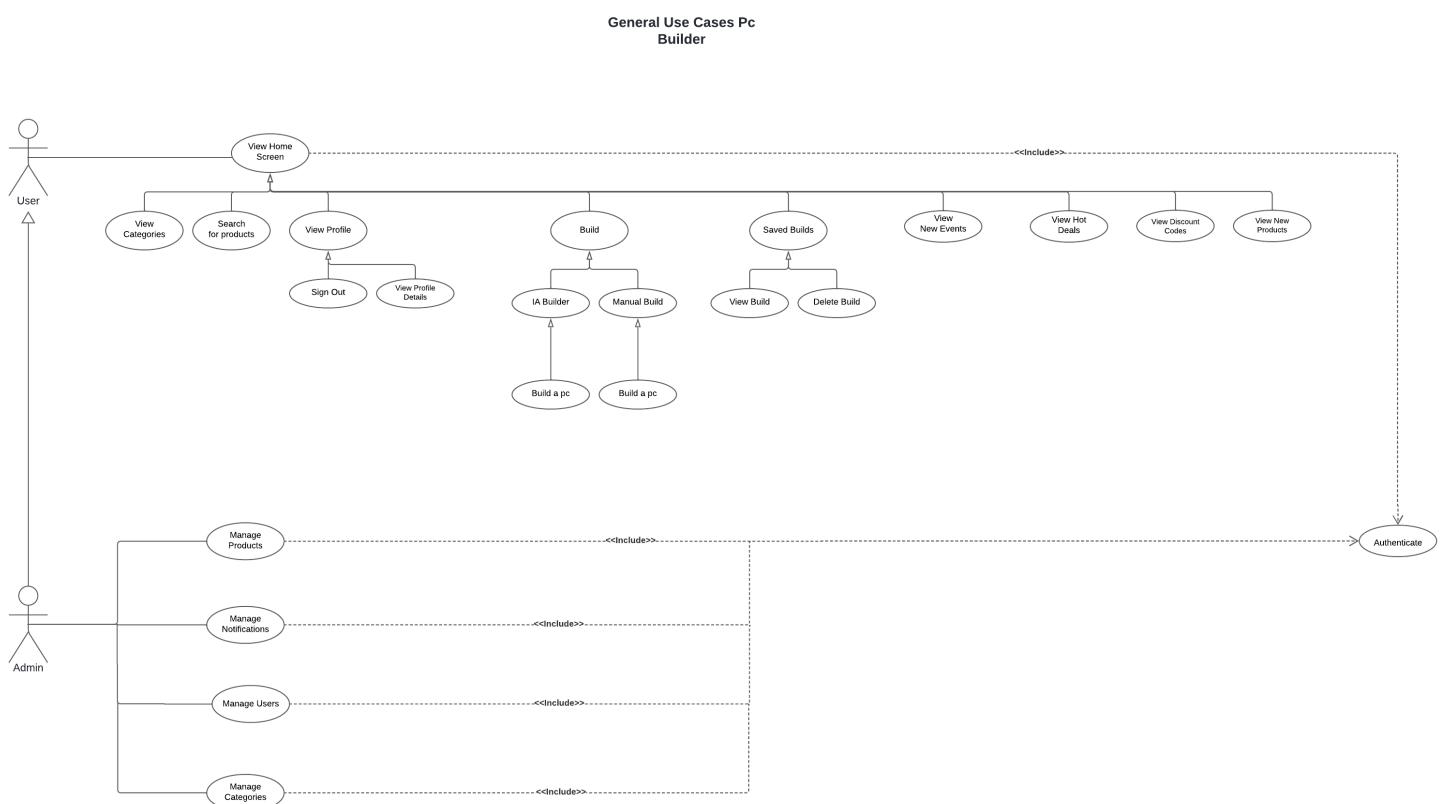
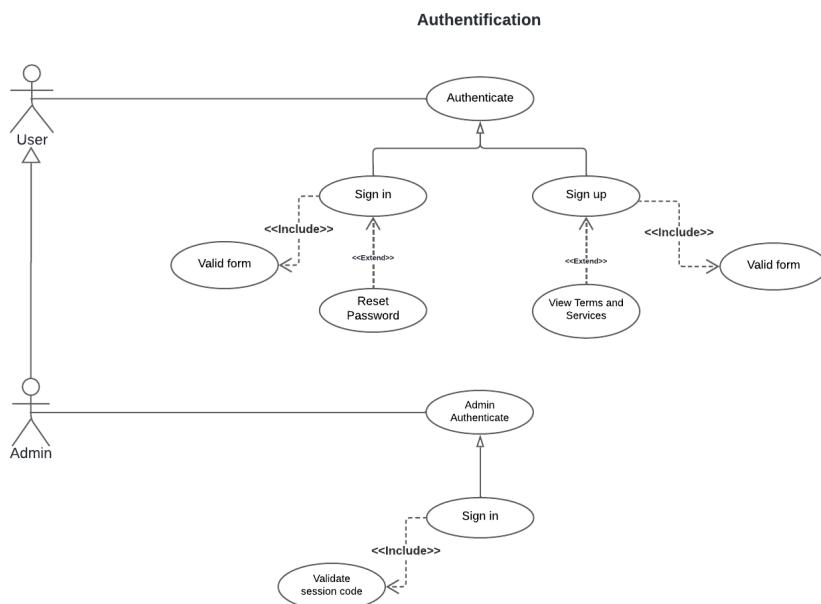


Figure 3-1 Global Use Case Diagram.

## 4.2 Detailed use case diagrams

### - "Authenticate" use case diagrams.

Authentication ensures Maximum data security; no action is allowed without authentication and verification of access rights.



**Figure 3-2 "Authenticate" use case diagrams**

- Textual description of use case « Sign up »

The table 3-2 below provides the textual description of the use case “Sign up”.

Use Case	Sign up
Actor	User
Purpose	Create an account
Pre-condition	Does not have an account
Main scenario	<ul style="list-style-type: none"> <li>-The user requests the Create account interface.</li> <li>-The system displays the Create account form.</li> <li>-The user enters the email address, password, phone number, name, accept the terms and services and validates fields.</li> </ul>

	- The system checks and creates new account.
<b>Exceptional scenario</b>	<ul style="list-style-type: none"> <li>-An error message is displayed if one or two fields are blank or invalid</li> <li>-An error message is displayed if the email address already exists.</li> </ul>
<b>Post-condition</b>	<p>User account created successfully and User directed to login interface.</p>

**Table 3-2 Text description of "Register" use cases.**

- Textual description of use case « Sign In »

The table 3-3 below provides the textual description of the use case “Sign In”.

Use Case	Sign In
<b>Actor</b>	User
<b>Purpose</b>	Sign the user in
<b>Pre-condition</b>	Have an account
<b>Main scenario</b>	<ul style="list-style-type: none"> <li>-The user requests the authentication interface.</li> <li>-The system displays the authentication form.</li> <li>-The user enters the email address and password and validates both fields.</li> <li>- The system checks and grants the user access to the app.</li> </ul>
<b>Exceptional scenario</b>	<ul style="list-style-type: none"> <li>-An error message is displayed if one or two fields are blank or invalid</li> <li>-An error message is displayed if the fields don't match any registered user</li> </ul>

<b>Post-condition</b>	User successfully authenticated.
-----------------------	----------------------------------

**Table 3-3 Text description of "Sign In" use cases.**

- Textual description of use case « Sign in Admin»

The table 3-4 below provides the textual description of the use case “Sign in Admin”.

<b>Use Case</b>	<b>Sign In Admin</b>
<b>Actor</b>	Admin
<b>Purpose</b>	Sign the admin in
<b>Pre-condition</b>	Have an admin account
<b>Main scenario</b>	<ul style="list-style-type: none"> <li>-The admin requests the authentication interface.</li> <li>-The system displays the authentication form.</li> <li>-The admin enters the email address, password and the session code and validates these fields.</li> <li>- The system checks and grants the admin access to the admin panel.</li> </ul>
<b>Exceptional scenario</b>	<ul style="list-style-type: none"> <li>-An error message is displayed if one or two fields are blank or invalid</li> <li>-An error message is displayed if the fields doesn't match any registered admin and the session code is expired.</li> </ul>
<b>Post-condition</b>	Admin successfully authenticated.

**Table 3-4 Table 3 4 Text description of "Sign in Admins" use cases.**

- Textual description of use case « Recover Password »

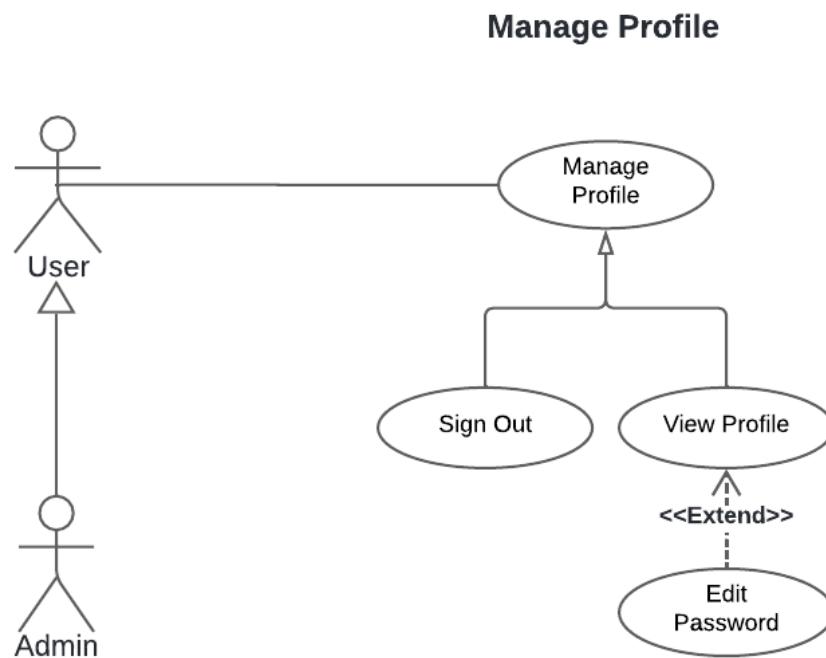
The table 3-5 below provides the textual description of the use case “Recover password”.

<b>Use Case</b>	<b>Recover Password</b>
<b>Actor</b>	User
<b>Purpose</b>	Recover user's account
<b>Pre-condition</b>	Lost or forgot password
<b>Main scenario</b>	<ul style="list-style-type: none"> <li>-The user requests the recover password interface.</li> <li>-The system displays the recover password form.</li> <li>-The user enters the email address and validate.</li> <li>- The system sends a recover link to the user email</li> <li>- The User enters a new valid password.</li> </ul>
<b>Exceptional scenario</b>	<ul style="list-style-type: none"> <li>- An error message is displayed if any of the fields are blank or invalid.</li> <li>- An error message is displayed if the email doesn't match any registered user.</li> <li>- An error message is displayed if the new password is invalid.</li> </ul>
<b>Post-condition</b>	Password successfully recovered.

**Table 3-5 Text description of " Recover password " use cases.**

- **"Manage Profile" use case diagram.**

Manage Profile allows users to view, edit, and delete their profile information, ensuring they have control over their account details and settings



**Figure 3-3 " Manage Profile" use case diagrams**

- Textual description of use case « Manage Profile»

The table 3-6 below provides the textual description of the use case “Manage Profile”.

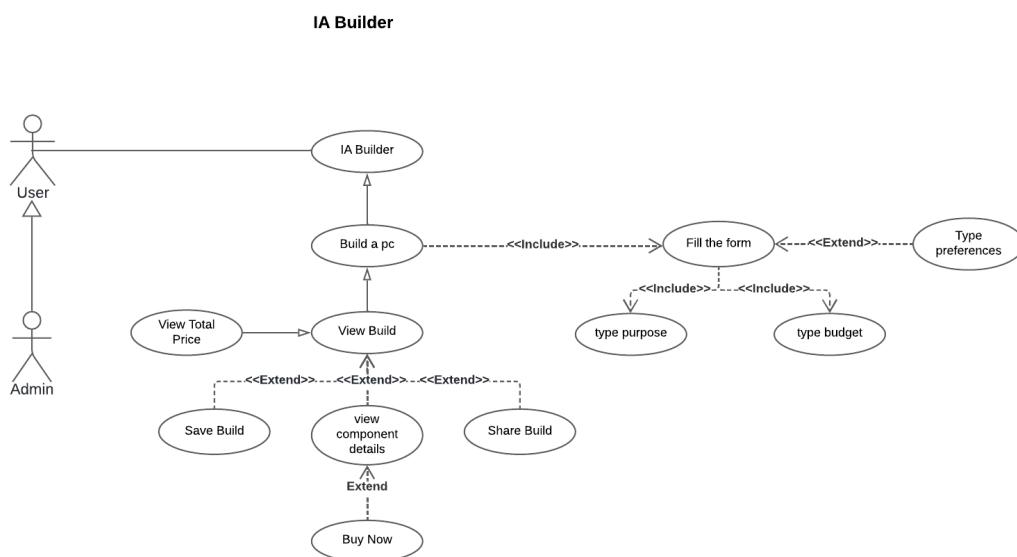
Use Case	Manage Profile
<b>Actor</b>	User
<b>Purpose</b>	Allows the user to manage their profile information.
<b>Pre-condition</b>	User is authenticated and logged into their account.

<b>Main scenario</b>	<ol style="list-style-type: none"> <li>1) The user clicks on the "Profile" icon.</li> <li>2) The system displays the profile management options: View Profile, Sign-out.</li> <li>3) The user selects an option:             <ol style="list-style-type: none"> <li>a) <b>View Profile:</b> The system displays the user's profile information.</li> <li>b) <b>Logout:</b> The system logs the user out and redirects to the login interface.</li> </ol> </li> </ol>
<b>Exceptional scenario</b>	<ul style="list-style-type: none"> <li>• <b>View Profile:</b> If there is an issue retrieving profile data, an error message is displayed.</li> </ul>
<b>Post-condition</b>	<ol style="list-style-type: none"> <li>4) <b>View Profile:</b> The profile information is successfully displayed.</li> <li>5) <b>Logout:</b> The user is successfully logged out.</li> </ol>

**Table 3-6 Text description of "Manage Profile" use cases.**

- **"Use AI Builder" use case diagram.**

Use AI Builder enables users to generate a customized build by filling out a form with their purpose, budget, and preferences, leveraging AI to create tailored recommendations.



**Figure 3-4 "Use AI Builder" use case diagrams**

- Textual description of use case « Use AI Builder»

The table 3-7 below provides the textual description of the use case “Use AI Builder”.

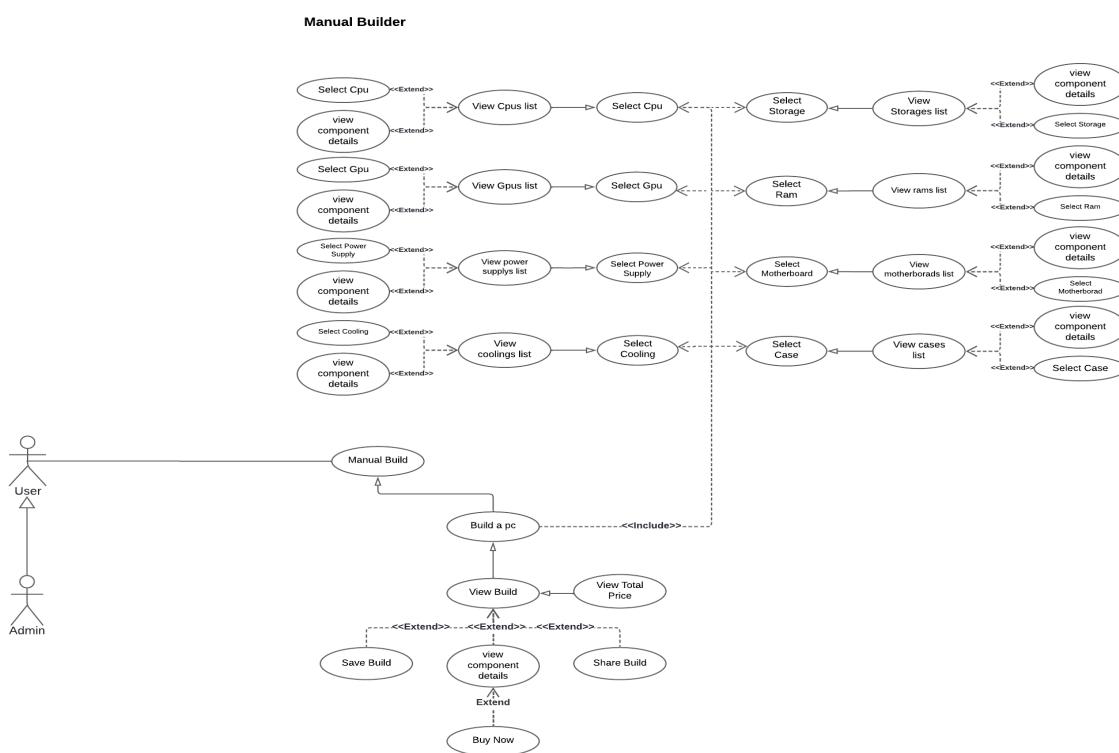
<b>Use Case</b>	<b>Use AI Builder</b>
<b>Actor</b>	User
<b>Purpose</b>	Allows the user to generate a customized build using AI by filling out a form with specific criteria.
<b>Pre-condition</b>	User is authenticated and logged into their account.
<b>Main scenario</b>	<ol style="list-style-type: none"> <li>1) The user navigates to the "Use AI Builder" interface.</li> <li>2) The system displays the form to be filled with fields for Purpose, Budget, and Preferences.</li> <li>3) The user fills in the form:             <ol style="list-style-type: none"> <li>a) Type Purpose: User specifies the purpose of the build (e.g., gaming, work, etc.).</li> <li>b) Type Budget: User specifies the budget for the build.</li> <li>c) Type Preferences: User specifies any additional preferences (e.g., preferred brands, components, etc.).</li> </ol> </li> <li>4) The user submits the form.</li> <li>5) The system processes the input and generates a build using AI based on the provided criteria.</li> <li>6) The system displays the generated build to the user.</li> </ol>
<b>Exceptional scenario</b>	<ol style="list-style-type: none"> <li>1) <b>Type Purpose:</b> If the purpose field is left blank or contains invalid data, an error message is displayed.</li> <li>2) <b>Type Budget:</b> If the budget field is left blank or contains invalid data, an error message is displayed. If the budget is unrealistic, an error message is displayed.</li> <li>3) <b>Generate Build using AI:</b> If there is an error in processing the input or generating the build, an error</li> </ol>

	message is displayed. If the input criteria are too vague or conflicting, an error message is displayed.
<b>Post-condition</b>	<ul style="list-style-type: none"> <li>- The customized build is successfully generated and displayed to the user in the full build interface.</li> <li>- If any errors occurred, appropriate error messages are displayed, and the user is prompted to correct the input.</li> </ul>

**Table 3-7 Text description of " Use AI Builder" use cases.**

- **"Create Custom PC Build" use case diagram.**

Create Custom PC Build allows users to select and assemble compatible components to build a custom PC, ensuring compatibility and meeting user specifications.



**Figure 3-5 "Create Custom PC Build" use case diagrams**

- Textual description of use case « Create Custom PC Build»

The table 3-8 below provides the textual description of the use case “Create Custom PC Build”.

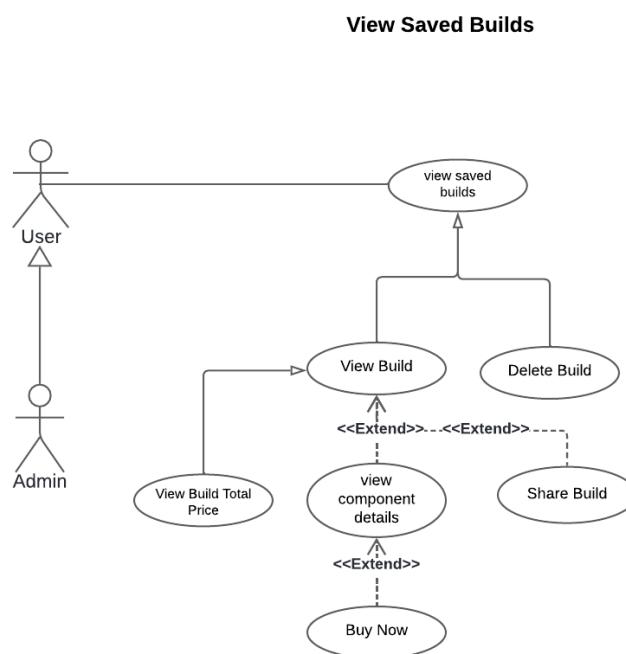
<b>Use Case</b>	<b>Create Custom PC Build</b>
<b>Actor</b>	User
<b>Purpose</b>	Allows the user to create a custom PC build by manually selecting compatible components.
<b>Pre-condition</b>	User is authenticated and logged into their account.
<b>Main scenario</b>	<ol style="list-style-type: none"> <li>1) The user navigates to the "Create Custom PC Build" interface.</li> <li>2) The system displays the options to assemble the PC build.</li> <li>3) The user selects the "Pick PC Component" option to view Compatible Components List: The system displays a list of compatible components. The user can filter the list or search with keywords.</li> <li>4) The user picks the components:             <ol style="list-style-type: none"> <li>a) Pick a CPU: User selects the processor.</li> <li>b) Motherboard: User selects the motherboard from the filtered motherboards based on the CPU socket</li> <li>c) GPU: User selects the graphics card.</li> <li>d) Pick Ram: User selects the RAM based on the filtered rams based on the motherboard.</li> <li>e) Pick Storage: User selects the type and size of storage.</li> <li>f) Pick a PS: User selects the power supply unit from the filtered power supplies based on the CPU tdp and GPU tdp.</li> <li>g) Pick PC Case: User selects the PC case from the filtered cases based on the motherboard format.</li> <li>h) Pick Cooling Unit: User selects the cooling unit</li> </ol> </li> </ol>

	from the filtered coolers based on the CPU socket 5) The system validates selected components.
<b>Exceptional scenario</b>	1) View Compatible Components List: If there is an issue retrieving the list, an error message is displayed. 2) Assemble PC Build: If there is an error during the assembly process, an error message is displayed.
<b>Post-condition</b>	The custom PC build is successfully created and the user is directed to full build interface.

**Table 3-8 Text description of "Create Custom PC Build" use cases.**

- **"View Saved Builds" use case diagram.**

View Saved Builds use case allows users to view a list of their saved custom PC builds and enables them to select a build to view details or delete it.



**Figure 3-6 "View Saved Builds" use case diagrams**

- Textual description of use case « View Saved Builds »

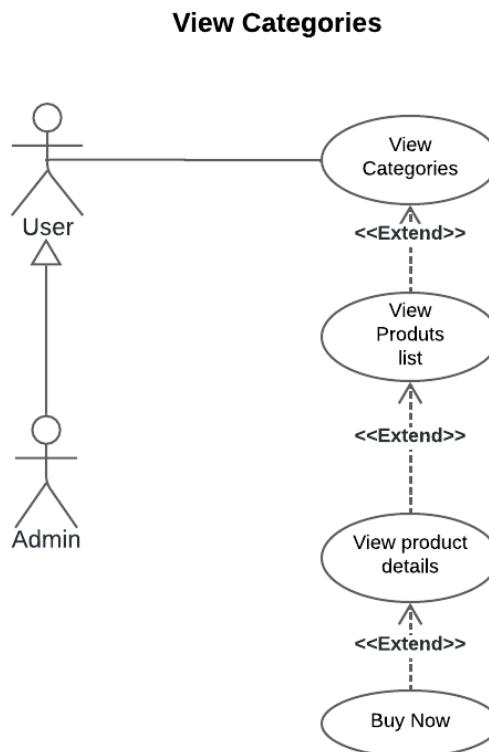
The table 3-9 below provides the textual description of the use case “View Saved Builds”.

<b>Use Case</b>	<b>Vie Saved Builds</b>
<b>Actor</b>	User
<b>Purpose</b>	Allows the user to browse through a list of their saved custom PC builds and view details or delete them.
<b>Pre-condition</b>	The user has created and saved at least one custom PC build.
<b>Main scenario</b>	<ul style="list-style-type: none"> <li>• The user selects the "Saved Builds" option.</li> <li>• The system displays a list of the user's saved custom PC builds.</li> <li>• The list includes basic information about each build, such as name.</li> <li>• The user can select a specific build from the list to view details or delete it.</li> <li>• The system displays the details of the selected build.</li> </ul>
<b>Exceptional scenario</b>	<p>If the user has not saved any builds, the system displays a message indicating no saved builds are available.</p> <p>If there is an error retrieving the list of saved builds, an error message is displayed</p>
<b>Post-condition</b>	The user gets redirected to full build interface to views the full details of their PC build, including component details and total price.

**Table 3-9 Text description of " View Saved Builds " use cases.**

- **"Browse Products by categories" use case diagram.**

Browse Products by categories use case allows users to browse through a categorized list of products.



**Figure 3-7 "Browse Products" use case diagrams**

- Textual description of use case « Browse Products by categories »

The table 3-10 below provides the textual description of the use case “Browse Products by categories”.

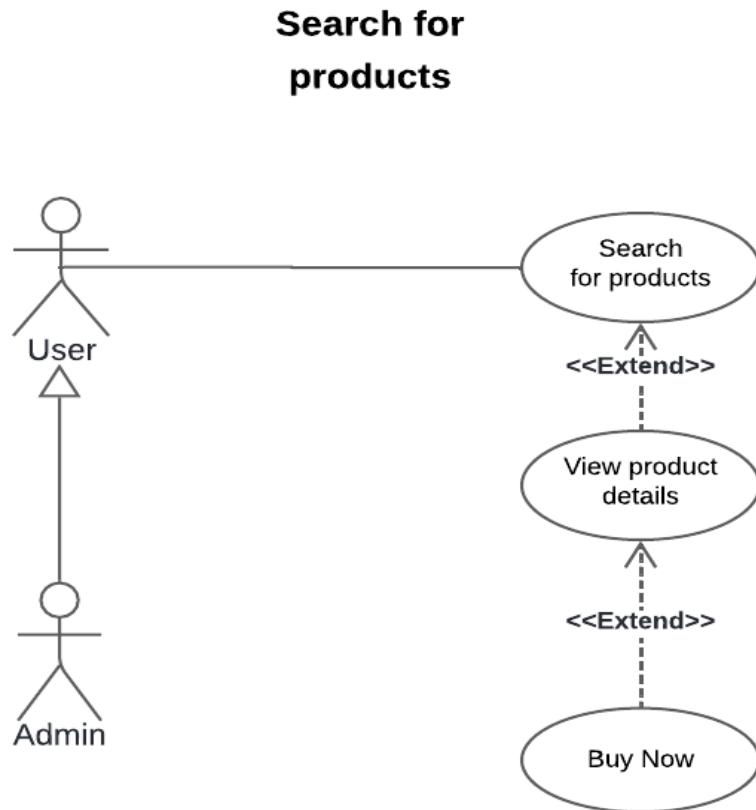
Use Case	Browse Products by categories
Actor	User
Purpose	Allows the user to browse through a categorized list of products offered by the system.
Pre-condition	The user is logged in and on the product browsing interface

<b>Main scenario</b>	<ul style="list-style-type: none"> <li>• The system displays a list of categories.</li> <li>• The user can browse through the product categories.</li> <li>• The user selects a category from the list.</li> <li>• The system displays a list of products from that category           <ul style="list-style-type: none"> <li>• On tapping the product details, the system displays the product details page for the selected product (including information such as product description, specifications, images, and price).</li> </ul> </li> </ul>
<b>Exceptional scenario</b>	<p>If there is an error retrieving the categories list, an error message is displayed.</p> <p>If there is an error retrieving the products list, an error message is displayed.</p>
<b>Post-condition</b>	<ul style="list-style-type: none"> <li>• The user browses through the categorized list of products.</li> </ul>

**Table 3-10 Text description of "Browse Products By Categories" use cases.**

-“Search for products” use case diagram.

Search for products use case allows users to search for a specific product based on its name.



**Figure 3-2 " Search for products " use case diagrams**

- Textual description of use case « Search for products »

The table 3-11 below provides the textual description of the use case “Search for products”.

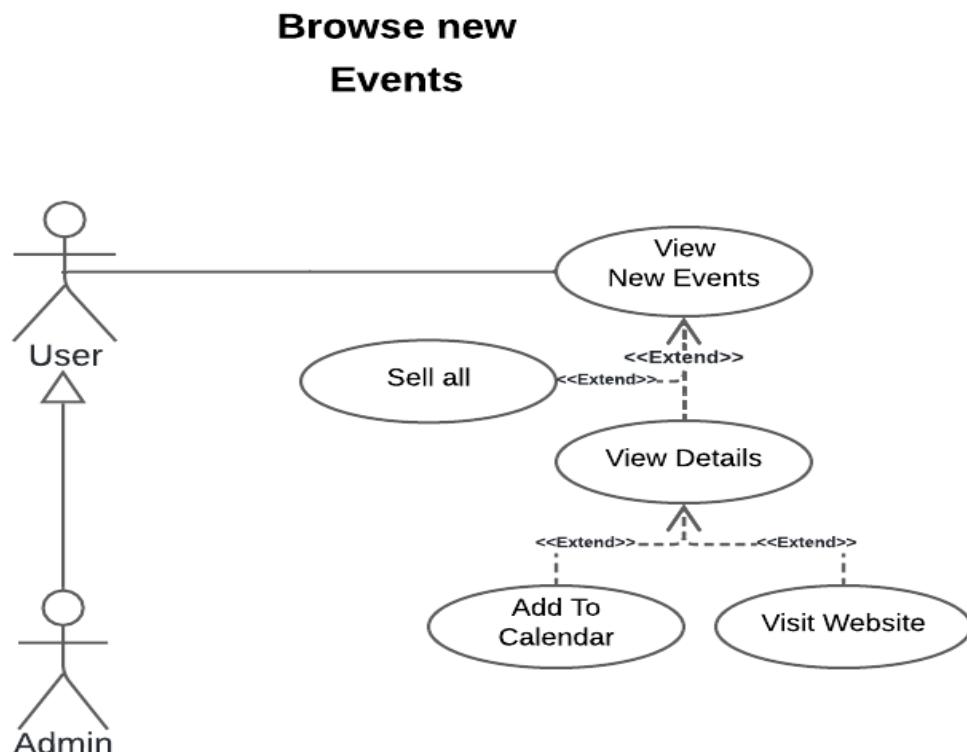
Use Case	Search for products
Actor	User
Purpose	Allows the user to browse search for a specific product based on his prompt.
Pre-condition	The user is logged in

<b>Main scenario</b>	<ul style="list-style-type: none"> <li>The user requests the search interface after clicking on the search icon.</li> <li>The system returns the search interface with an input.</li> <li>The user types the name of the product he's looking for.</li> <li>The system displays the results as list of products.</li> <li>The user can view the product details and buy it from the official website.</li> </ul>
<b>Exceptional scenario</b>	<ul style="list-style-type: none"> <li>If there is an error retrieving the product list, an error message is displayed.</li> <li>If the product is not available a message is displayed.</li> </ul>
<b>Post-condition</b>	The user search for a specific product.

**Table 3-11 Text description of " Search for products " use cases.**

- **"Browse new Events" use case diagram.**

Browse New Events use case allows users to browse through a list of new events concerning MegaPc plus, Users can View and open those events on their calendar.



**Figure 3-9 " Browse New Events " use case diagram**

- Textual description of use case « Browse New Events »

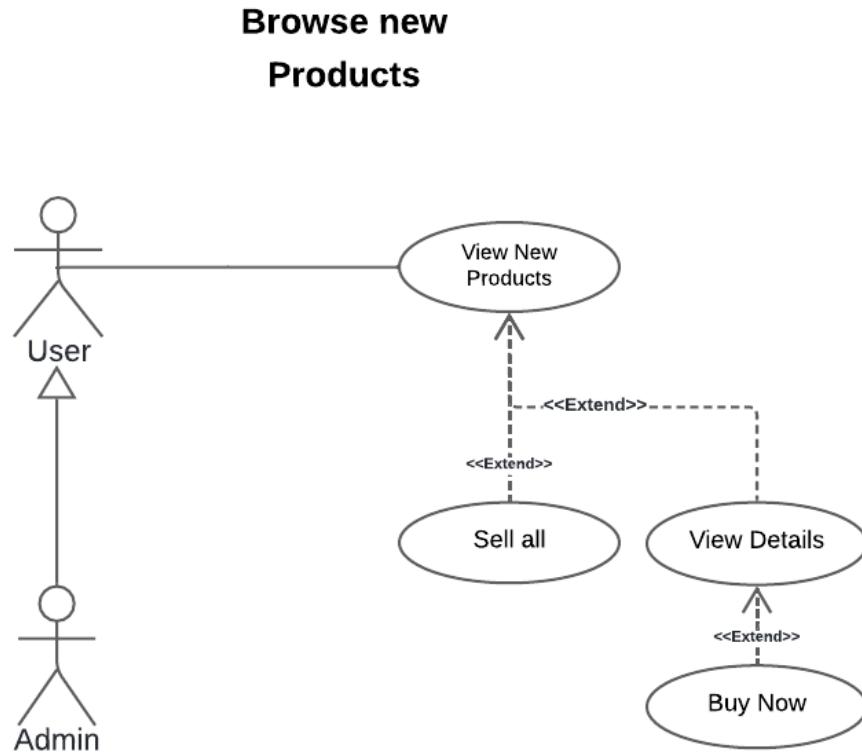
The table 3-12 below provides the textual description of the use case “Browse new Events”.

<b>Use Case</b>	<b>Browse New Events</b>
<b>Actor</b>	User
<b>Purpose</b>	Allows the user to browse the new events and add it to calendar.
<b>Pre-condition</b>	The user is logged in
<b>Main scenario</b>	<ul style="list-style-type: none"> <li>• The user navigates to the home page and view the available new events.</li> <li>• The user can view the event detail by clicking on the event.</li> <li>• The user can view all the available new events by taping “see all” or the arrow at the end of the cards.</li> <li>• The user can and add the event calendar or visit its website from its details page.</li> </ul>
<b>Exceptional scenario</b>	<ul style="list-style-type: none"> <li>• If there is an error retrieving the event list, an error message is displayed.</li> </ul>
<b>Post-condition</b>	The user views the available new events.

**Table 3-12 Text description of " Browse New Events " use cases.**

- "Browse new products" use case diagram.

Browse New Products use case allows users to browse through a list of new products concerning Mega Pc plus,



**Figure 3-3 " Browse new products " use case diagram**

- Textual description of use case « Browse new products »

The table 3-13 below provides the textual description of the use case “Browse new products”.

Use Case	Browse new products
Actor	User
Purpose	Allows the user to browse the new products
Pre-condition	The user is logged in

<b>Main scenario</b>	<ul style="list-style-type: none"> <li>• The user navigates to the home page and view the available new products.</li> <li>• The user can view the new product detail by taping on it.</li> <li>• The user can view all the available new products by taping “see all” or the arrow at the end of the cards.</li> <li>• The user visits the new product website from its details page.</li> </ul>
<b>Exceptional scenario</b>	<ul style="list-style-type: none"> <li>• If there is an error retrieving the new products list, an error message is displayed.</li> </ul>
<b>Post-condition</b>	The user view the available new products.

Table 3-13 Text description of " Browse new products " use cases.

#### -"Browse hot deals" use case diagram.

Browse hot deals use case allows users to browse through a list of the hot deals concerning Mega Pc plus,

#### Browse Hot Deals

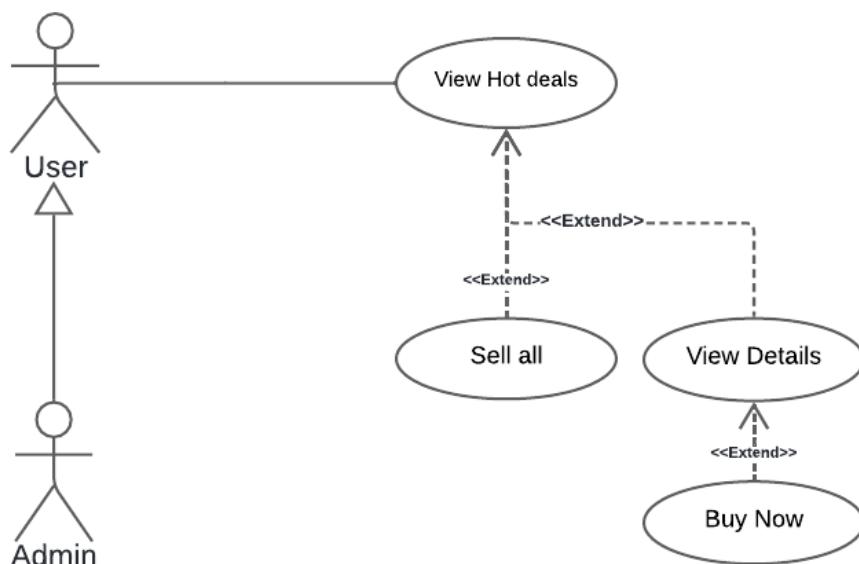


Figure 3-4 " Browse Hot Deals " use case diagram

- Textual description of use case « Browse hot deals »

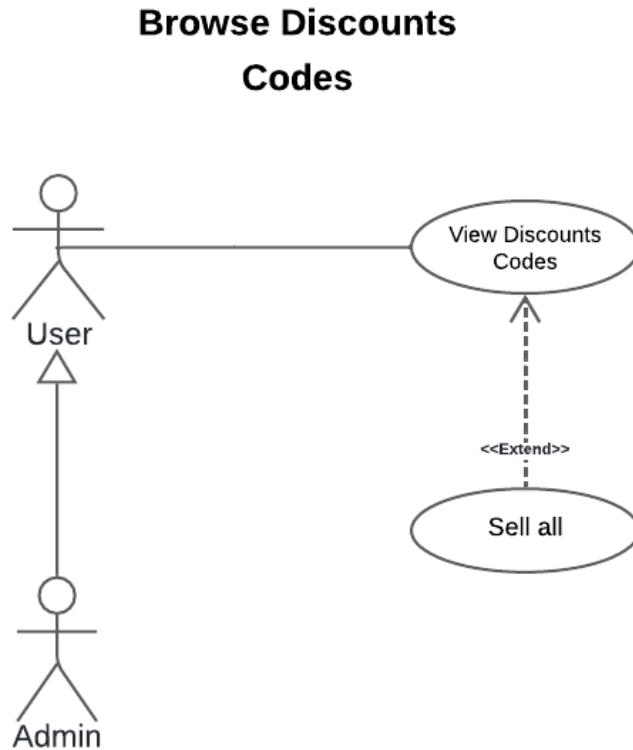
The table 3-14 below provides the textual description of the use case “Browse hot deals”.

<b>Use Case</b>	<b>Browse hot deals</b>
<b>Actor</b>	User
<b>Purpose</b>	Allows the user to browse the hot deals
<b>Pre-condition</b>	The user is logged in
<b>Main scenario</b>	<ul style="list-style-type: none"> <li>• The user navigates to the home page and view the available hot deals.</li> <li>• The user can view the hot deal detail by taping on it.</li> <li>• The user can view all the available hot deals by taping “see all” or the arrow at the end of the cards.</li> <li>• The user visits the hot deal website from its details page.</li> </ul>
<b>Exceptional scenario</b>	<ul style="list-style-type: none"> <li>• If there is an error retrieving the hot delas list, an error message is displayed.</li> </ul>
<b>Post-condition</b>	The user views the available hot deals.

**Table 3-14 Text description of " Browse hot deals" use cases.**

### - "Browse discounts codes" use case diagram.

Browse discounts codes use case allows users to browse through a list of the discount's codes concerning Mega Pc plus,



**Figure 3-5 " Browse discounts codes " use case diagram**

- Textual description of use case « Browse discounts codes »

The table 3-15 below provides the textual description of the use case “Browse discounts codes”.

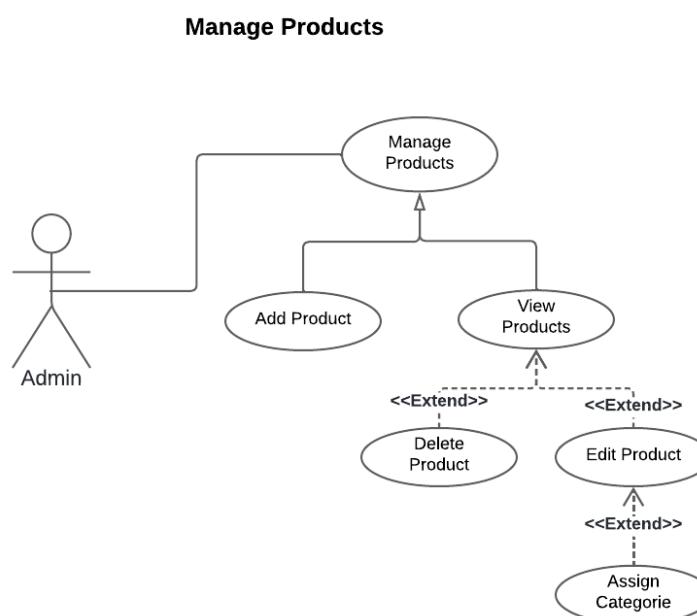
Use Case	Browse discounts codes
Actor	User
Purpose	Allows the user to browse the hot deals
Pre-condition	The user is logged in

<b>Main scenario</b>	<ul style="list-style-type: none"> <li>The user navigates to the home page and view the available discounts codes.</li> <li>The user can view all the available discounts codes by taping “see all” or the arrow at the end of the cards.</li> </ul>
<b>Exceptional scenario</b>	<ul style="list-style-type: none"> <li>If there is an error retrieving the discounts codes list, an error message is displayed.</li> </ul>
<b>Post-condition</b>	The user views the available discounts codes.

**Table 3-15 Text description of " Browse hot deals" use cases.**

#### -"Manage products" use case diagram.

Manage products use case allows admins to manage mega pc inventory products by adding, deleting, updating and displaying the products



**Figure 3-6 " Manage Products " use case diagram**

- Textual description of use case « Manage Products »

The table 3-16 below provides the textual description of the use case “Manage Products”.

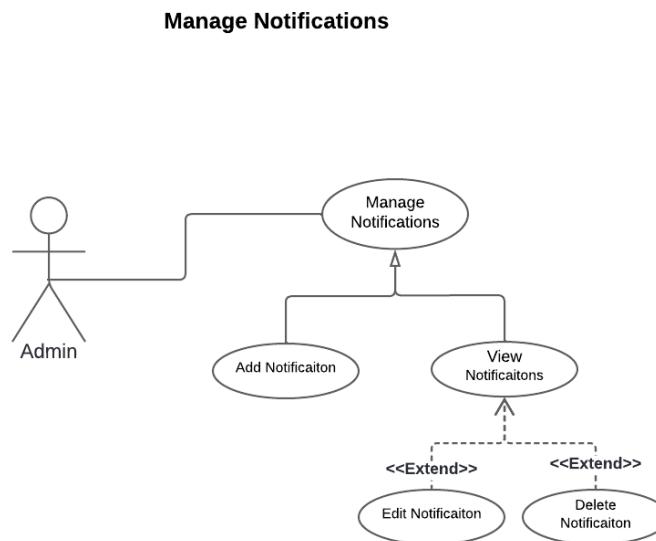
<b>Use Case</b>	<b>Manage Products</b>
-----------------	------------------------

<b>Actor</b>	Admin
<b>Purpose</b>	Allows the admin to manage the products
<b>Pre-condition</b>	The admin is logged in
<b>Main scenario</b>	<p><b><u>Principal scenario 1:</u></b></p> <ul style="list-style-type: none"> <li>• The admin navigates to the products section and view the available products.</li> <li>• The admin clicks on the product he wants to view.</li> <li>• The admin has the option to delete the product or continue editing.</li> <li>• The admin can assign a category while editing the product</li> </ul> <p><b><u>Principal scenario 2:</u></b></p> <ul style="list-style-type: none"> <li>• The admin adds a product.</li> </ul>
<b>Exceptional scenario</b>	<p><b><u>Principal scenario 1:</u></b></p> <ul style="list-style-type: none"> <li>• If there is an error retrieving the products list, an error message is displayed.</li> </ul> <p><b><u>Principal scenario 2:</u></b></p> <ul style="list-style-type: none"> <li>• None</li> </ul>
<b>Post-condition</b>	<p>The admin can:</p> <ul style="list-style-type: none"> <li>-view the products.</li> <li>-add a product and in the meantime assign a category.</li> <li>-delete a product</li> </ul>

**Table 3-16 Text description of " Manage products" use cases.**

**-"Manage notifications" use case diagram.**

Manage notifications use case allows admin to Manage notification.



**Figure 3-7 " Manage Notifications " use case diagram**

- Textual description of use case « Push Notifications »

The table 3-17 below provides the textual description of the use case “Manage notifications”.

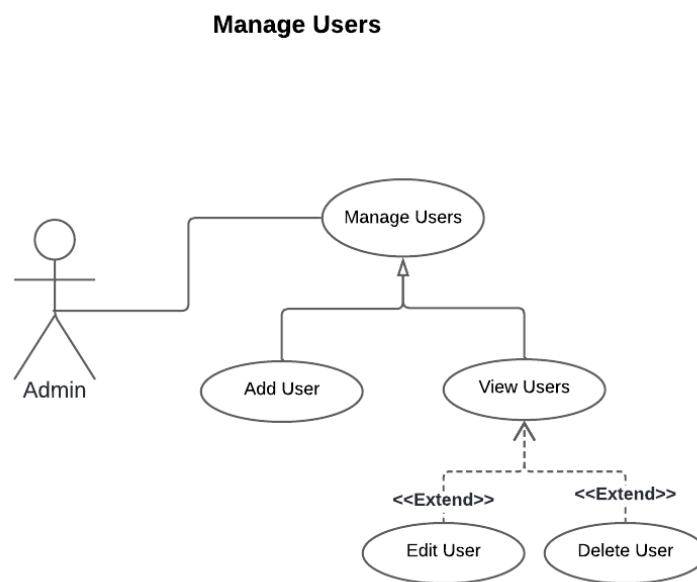
Use Case	Manage notifications
<b>Actor</b>	Admin
<b>Purpose</b>	Allows the admin to Manage notifications.
<b>Pre-condition</b>	The admin is logged in
<b>Main scenario</b>	<p><b>Principal scenario 1:</b></p> <ul style="list-style-type: none"> <li>• The admin navigates to the notifications section and view the active notifications.</li> <li>• The admin clicks on the notification he wants to view.</li> <li>• The admin has the option to delete the notification or continue editing.</li> </ul> <p><b>Principal scenario 2:</b></p> <ul style="list-style-type: none"> <li>• The admin adds a notification.</li> </ul>

<b>Exceptional scenario</b>	<ul style="list-style-type: none"> <li>An error message is displayed if one of fields are blank or invalid</li> <li>If there is an error retrieving the notifications list, an error message is displayed.</li> </ul>
<b>Post-condition</b>	<p>The admin can send notification to all the available users.</p> <ul style="list-style-type: none"> <li>-view the notifications.</li> <li>-add a notification.</li> <li>-edit a notification.</li> <li>-delete a notification.</li> </ul>

**Table 3-17 Text description of " Manage notifications" use case.**

**-"Manage Users" use case diagram.**

Manage users use case allows admins to manage users by adding, deleting, updating and displaying the users



**Figure 3-8 " Manage Users " use case diagram**

- Textual description of use case « Manage Users »

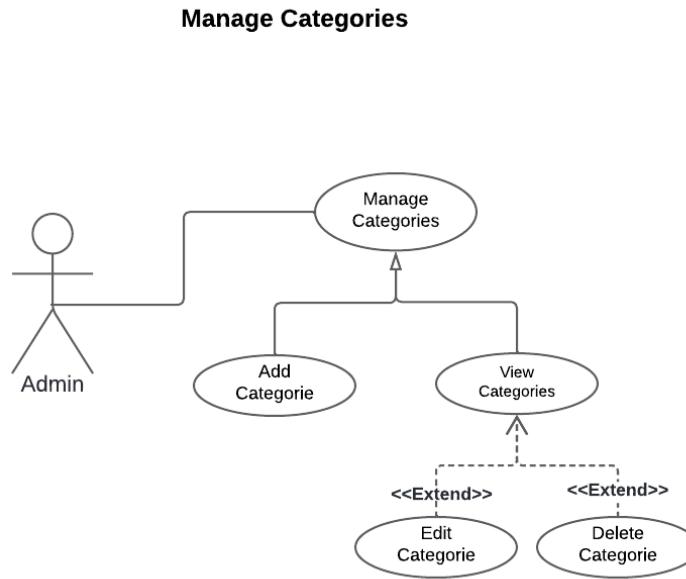
The table 3-18 below provides the textual description of the use case "Manage Users".

<b>Use Case</b>	<b>Manage Users</b>
<b>Actor</b>	Admin
<b>Purpose</b>	Allows the admin to manage the users
<b>Pre-condition</b>	The admin is logged in
<b>Main scenario</b>	<p><b><u>Principal scenario 1:</u></b></p> <ul style="list-style-type: none"> <li>• The admin navigates to the users section and view the available users.</li> <li>• The admin clicks on the user he wants to view.</li> <li>• The admin has the option to delete the user or continue editing.</li> </ul> <p><b><u>Principal scenario 2:</u></b></p> <ul style="list-style-type: none"> <li>• The admin adds a user.</li> </ul>
<b>Exceptional scenario</b>	<p><b><u>Principal scenario 1:</u></b></p> <ul style="list-style-type: none"> <li>• If there is an error retrieving the users list, an error message is displayed.</li> </ul> <p><b><u>Principal scenario 2:</u></b></p> <ul style="list-style-type: none"> <li>• User email already exist.</li> </ul>
<b>Post-condition</b>	The admin can: <ul style="list-style-type: none"> <li>-view the users.</li> <li>-add a user.</li> <li>-edit a user.</li> <li>-delete a user.</li> </ul>

**Table 3-18 Text description of " Manage Users" use case.**

**-"Manage Categories" use case diagram.**

Manage categories use case allows admins to manage categories by adding, deleting, updating and displaying the categories



**Figure 3-9 " Manage Categories " use case diagram**

- Textual description of use case « Manage Categories »

The table 3-19 below provides the textual description of the use case “Manage Categories”.

Use Case	Manage Categories
<b>Actor</b>	Admin
<b>Purpose</b>	Allows the admin to manage the categories
<b>Pre-condition</b>	The admin is logged in
<b>Main scenario</b>	<p><b>Principal scenario 1:</b></p> <ul style="list-style-type: none"> <li>• The admin navigates to the categories section and view the available categories.</li> <li>• The admin clicks on the category he wants to view.</li> <li>• The admin has the option to delete the category or continue editing.</li> </ul> <p><b>Principal scenario 2:</b></p> <ul style="list-style-type: none"> <li>• The admin adds a category.</li> </ul>

<b>Exceptional scenario</b>	<p><b>Principal scenario 1:</b></p> <ul style="list-style-type: none"> <li>If there is an error retrieving the categories list, an error message is displayed.</li> </ul> <p><b>Principal scenario 2:</b></p> <ul style="list-style-type: none"> <li>Category already exists.</li> </ul>
<b>Post-condition</b>	<p>The admin can:</p> <ul style="list-style-type: none"> <li>-view the categories.</li> <li>-add a category.</li> <li>-edit a category.</li> <li>-delete a category.</li> </ul>

Table 3-19 Text description of " Manage Categories" use case.

## 5. User Interface Design

### 5.1 Key UI Screens and Functionalities

In this subsection we showcase the key functionalities of the system and how they are reflected in the UI design. Briefly explaining the purpose of each screen and how users interact with it to achieve their goals.

- Login Screen:** Allows users to enter their credentials and access the application.
- Signup Screen:** Enables users to create a new account for using the application.
- Reset Password Screen:** Assists users in retrieving their forgotten password.
- Profile Screen:** Provides a personalized space for users to manage their account information.
- Home Screen:** Serves as the main landing point for users, offering easy access to key functionalities.
- Product Details:** Presents detailed information about an individual product.
- Manual Builder Screen:** Provides a user interface for selecting components and creating a custom PC build manually.
- Ai Builder Screen:** Offers an AI-powered recommendation system for building a PC based on user preferences.
- Display Full Build:** Shows the complete configuration of a custom PC build, including selected components, total price, and additional details.

- **Saved Builds:** Allows users to access previously created custom PC build configurations.

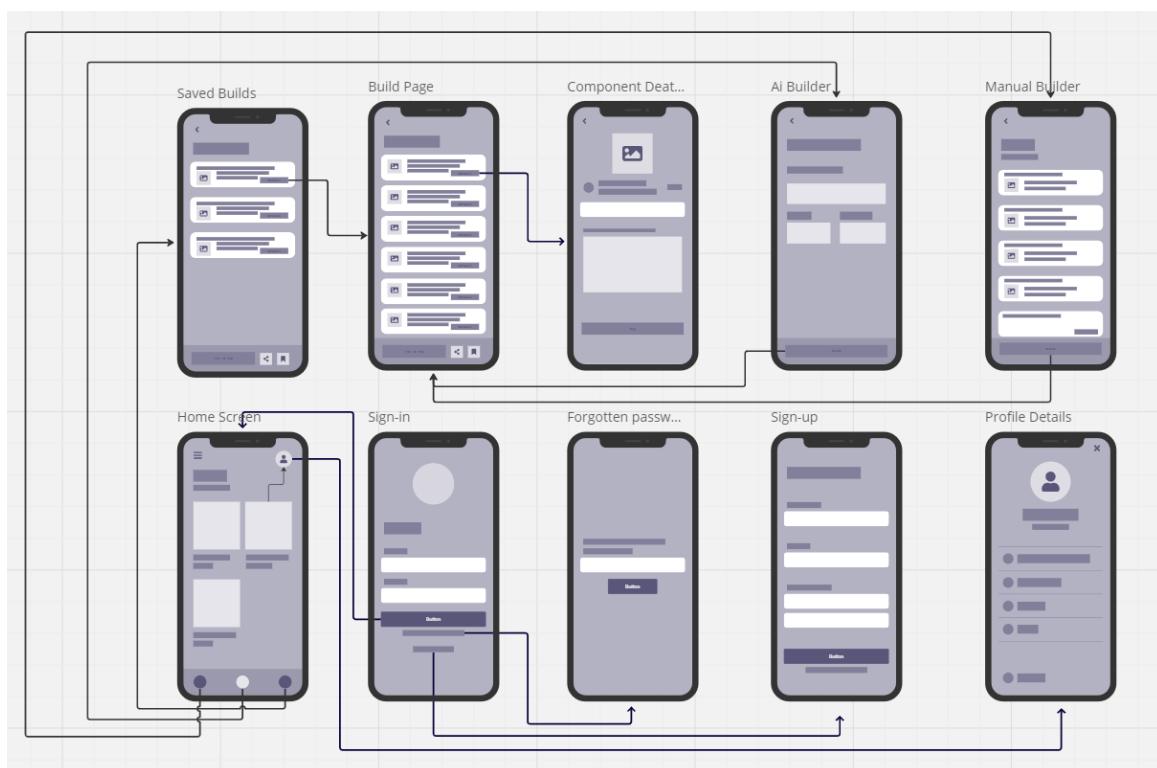
## 5.2 Design Process

This subsection explains the design process followed for creating the user interface (UI) of the custom PC build system.

### 5.2.1 Wireframing

Wireframing is crucial in software development. It creates a visual UI foundation by outlining screen layouts and user flows. This helps prioritize features, gather early feedback, and establish a shared vision for a user-centered design.

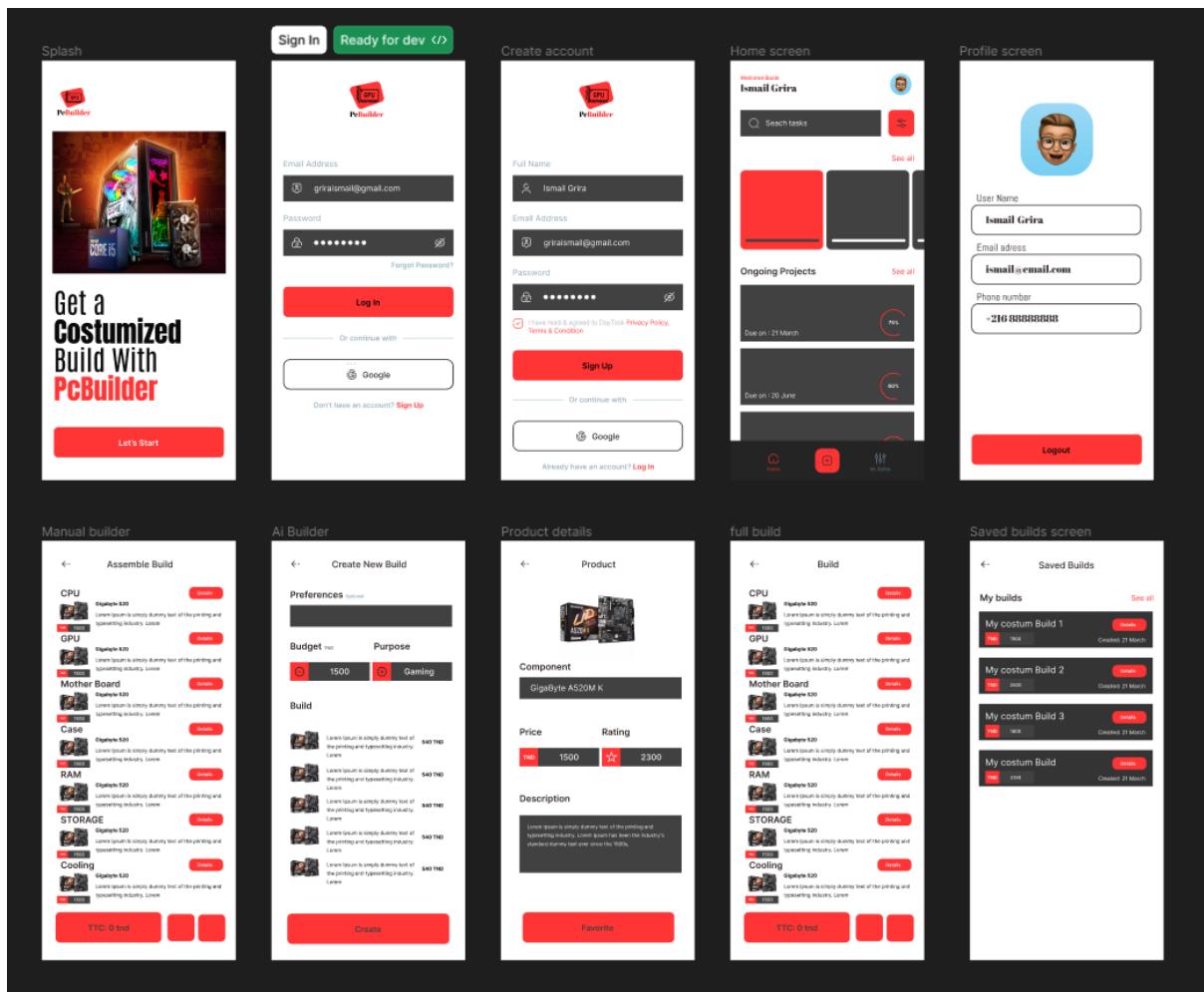
Having discussed the importance of wireframing, here is the breakdown of the key screens covered in our wireframes for our custom PC build system.



**Figure 3-10 PC Builder System Wireframes**

### 5.2.2 UI Design

By leveraging Figma's design capabilities and collaborative features, we were able to transform the initial wireframes into a visually appealing and user-friendly UI for our custom PC build system.



**Figure 3-18 Pc Builder UI Design**

### 5.3 User Experience (UX) Considerations

During the design process, we focused on creating a user experience (UX) that is intuitive, efficient, and user-friendly. Here's a breakdown of some key UX considerations:

- **User Flow:** The UI design prioritizes a clear and straightforward user flow. Users can easily navigate through the app to achieve their goals, whether it's building a custom PC, browsing product options, or saving builds. Consistent placement of menus, functionalities, and indicators to guide users through the process.
- **Intuitive Navigation:** The navigation system utilizes recognizable icons, and a consistent layout across all screens. This allows users to quickly understand how to find the features they need and easily complete tasks.
- **Accessibility Features:** While not an exhaustive list, the UI design incorporates basic accessibility features to ensure wider usability. This includes features like sufficient color contrast for text and background elements, and consideration for future implementation of features.
- **User Interface Consistency:** A consistent visual style is maintained throughout the app using a predefined color scheme, font family, and layout principles. This consistency promotes a sense of familiarity for users as they navigate different sections.
- **Error Handling and Feedback:** The UI provides informative error messages and visual cues to guide users in case of incorrect input or errors. Additionally, the system offers clear feedback for successful actions, such as confirmation messages or visual indicators.

By focusing on these UX considerations, the UI design aims to create a user-friendly and intuitive experience for anyone building their custom PC through this system.

## Conclusion

This chapter covers the analysis and requirements for a system that allows users to create and manage custom PC builds. It discusses functional and non-functional requirements. Use cases are then covered, including creating a custom PC build, displaying the full build, browsing saved builds, and browsing products. Wireframing and Designing process was covered as well to ensures great user experience, efficiency, compatibility between components and meets user specifications throughout the process.

# Chapter 4 Conception

## Introduction

In this chapter, we explain how we designed various use cases for our platform using UML with class and sequence diagrams.

## 1. Architectural Design

### 1.1 Application Architector

The architecture is the fundamental model for the successful implementation of such a mobile application. It mainly ensures the organization and structuring of the application, making it easier for developers or anyone else to understand how it will function. In this context, we chose MVC as the application architecture, with its main advantage being the separation of data (Model), presentation (View), and actions (Controller). This choice is based on the advantages it offers:

- Clear and efficient design through the separation of data from the view and controller.
- Time-saving in application maintenance and evolution.
- Great flexibility in organizing development among different developers.

Functioning of the MVC: Figure 4-1 shows the different interactions between the model, the view, and the controller.

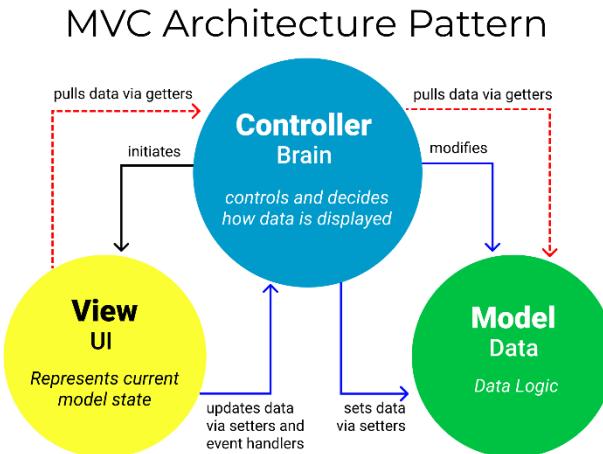


Figure 4-1 MVC Architector.

## 2. Modeling Language

For our design implementation, we use the UML language for graphical modeling, which presents the interactions between the different components of the future software and its users. Unified Modeling Language (UML) is a graphical modeling language designed to provide a standardized method for visualizing the design of a system. It is commonly used in software development and object-oriented design.

Figure 4-2 depicts the UML logo.



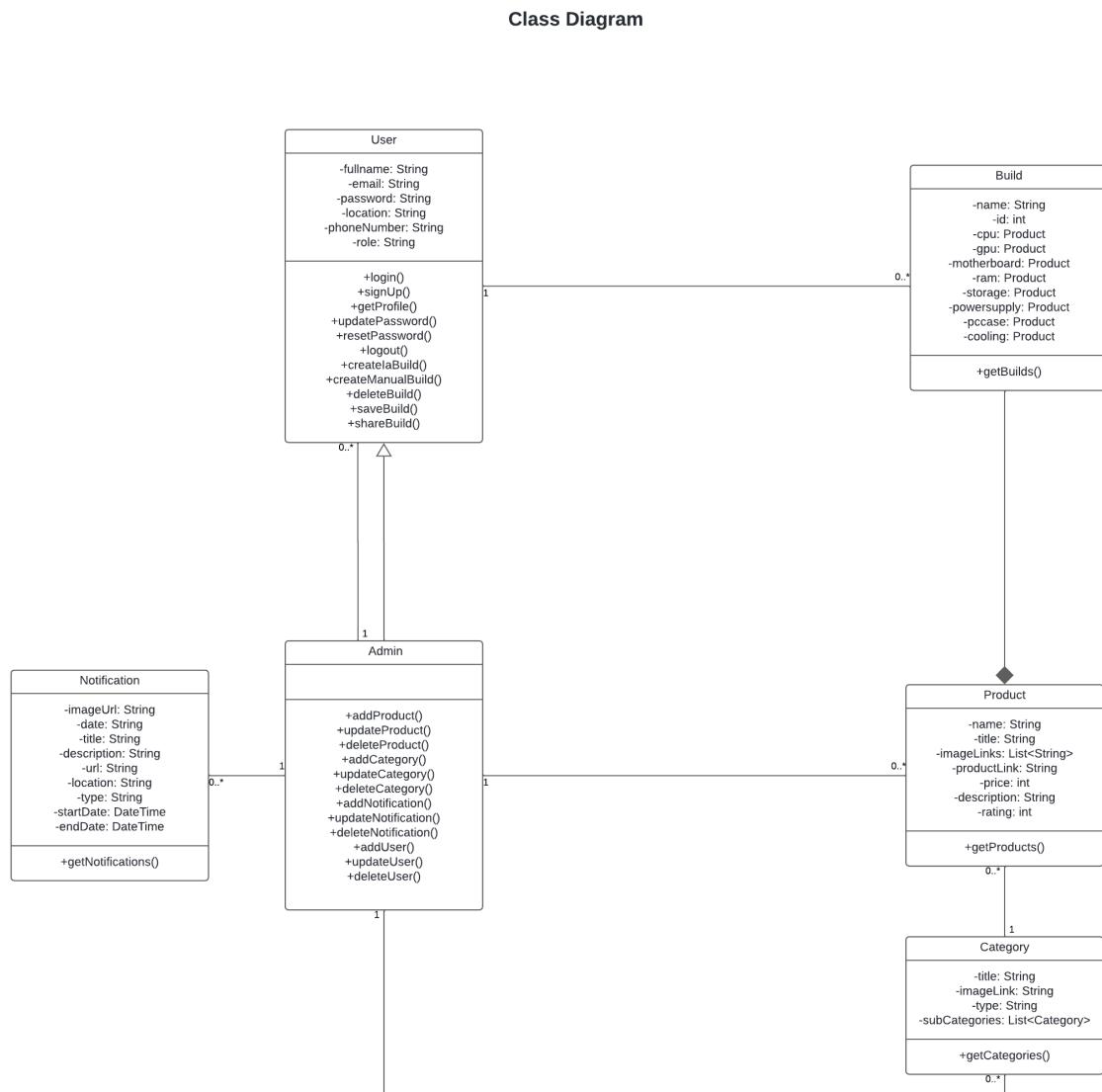
**Figure 4-2 Unified Modeling Language.**

### 3. Static Design

#### 3.1 Class Diagram

The class diagram represents the objects and information structurally in our system for internal interactions, as it presents all possible classes with their relationships.

Figure 4-3 depicts the overall class diagram of our application, which contains 6s classes:



**Figure 4-3 Global Class Diagram.**

- **Class: User**

Represents a user of the application with functionalities for registration, authentication, build management and profile management.

- **Class: Product**

Represents a product within the application, including details such as name, description, price, and ratings.

- **Class: Category**

Represents a category of products, supporting hierarchical categorization with potential subcategories.

- **Class: Build**

Represents a custom PC build configuration, allowing users to assemble and save custom PC setups.

- **Class: Notification**

Represents a notification within the application, used to inform users about new products, deals, discounts, events, and other relevant information.

- **Class: Admin**

Represents an administrator of the application, with privileges to manage user accounts, products, notifications, and other administrative tasks.

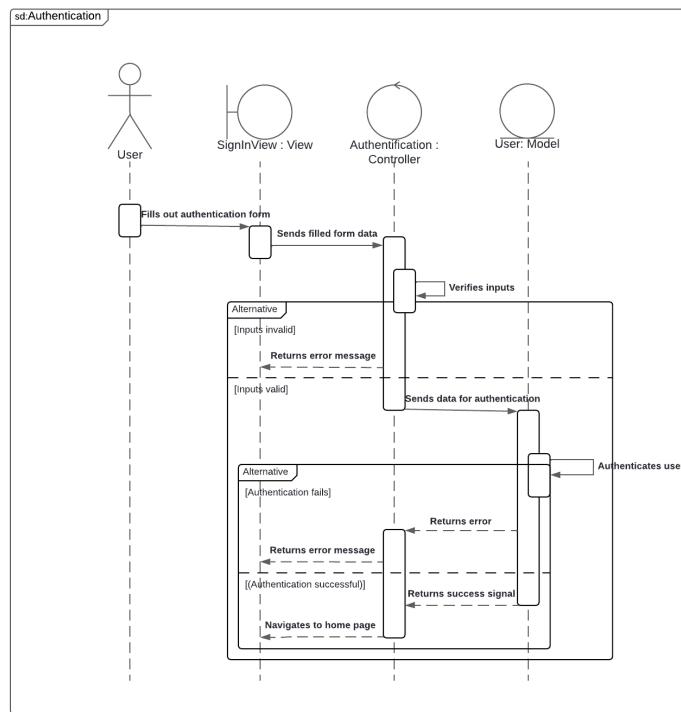
## 4. Dynamic Design

In this section, we present the sequence diagrams of our application, which are used to model the interactions between objects.

### 4.1 Sequence Diagrams

#### 4.1.1 Sequence Diagram: Authentication

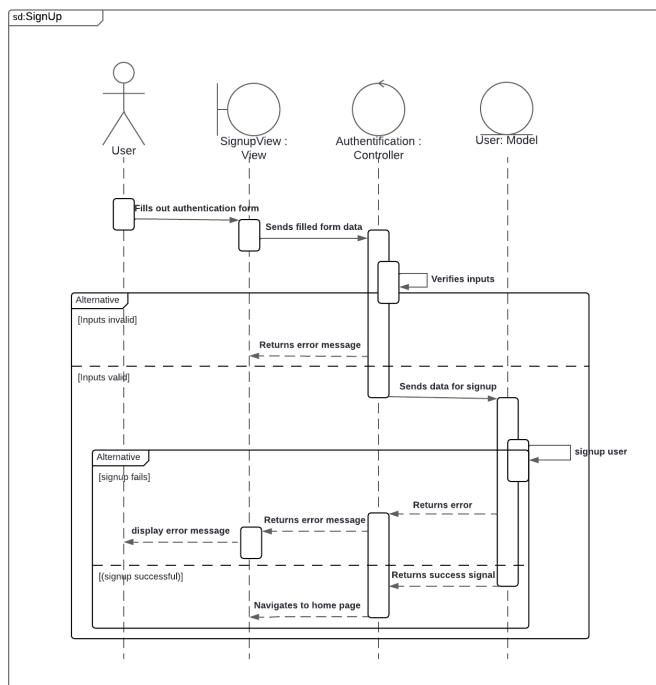
Figure 4-4 represents the "Authentication" sequence diagram.



**Figure 4-4 "Authentication" sequence diagram.**

#### 4.1.2 Sequence Diagram: Sign Up

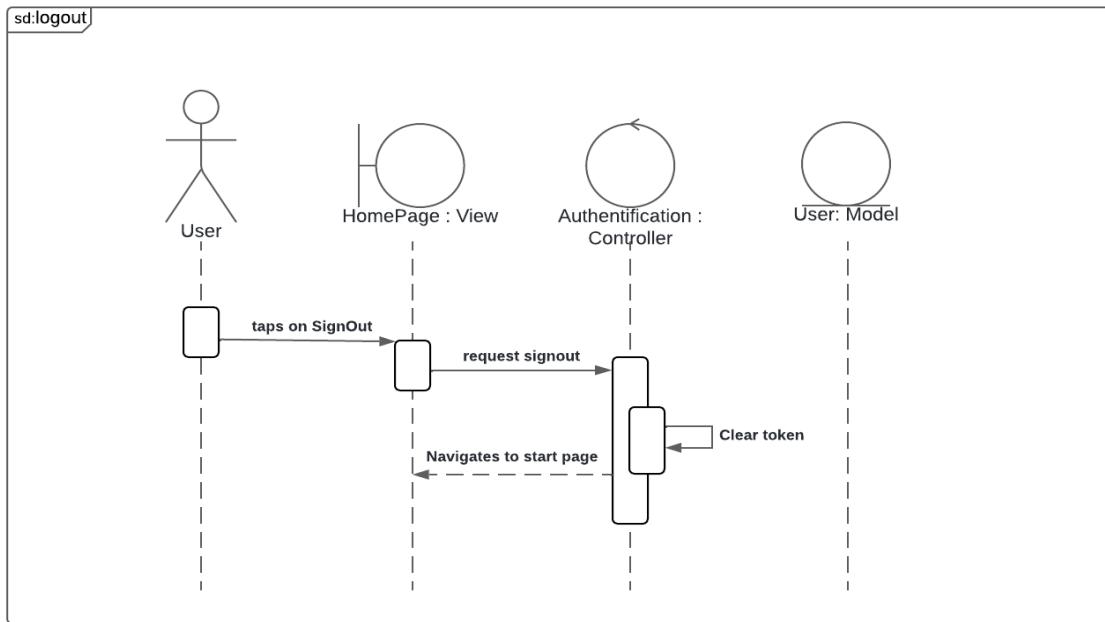
Figure 4-5 represents the " Sign Up " sequence diagram.



**Figure 4-5 " Sign Up " sequence diagram.**

### 4.1.3 Sequence Diagram: Log Out

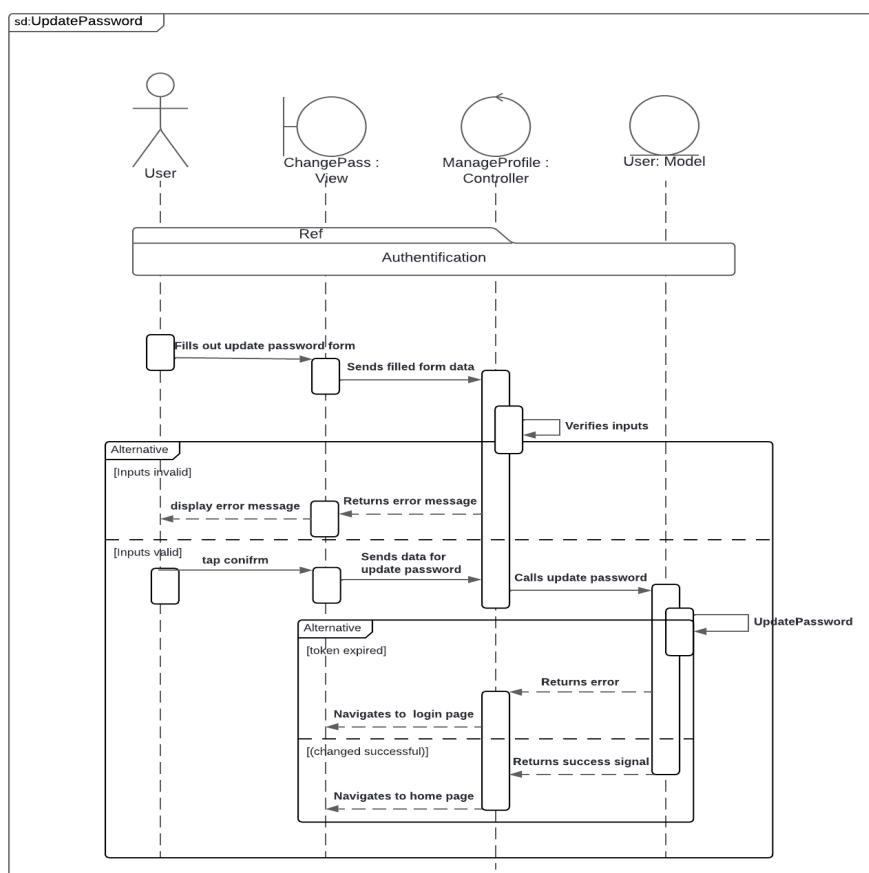
Figure 4-6 represents the " Log Out " sequence diagram.



**Figure 4-6 " Log Out " sequence diagram.**

### 4.1.4 Sequence Diagram: Update Password

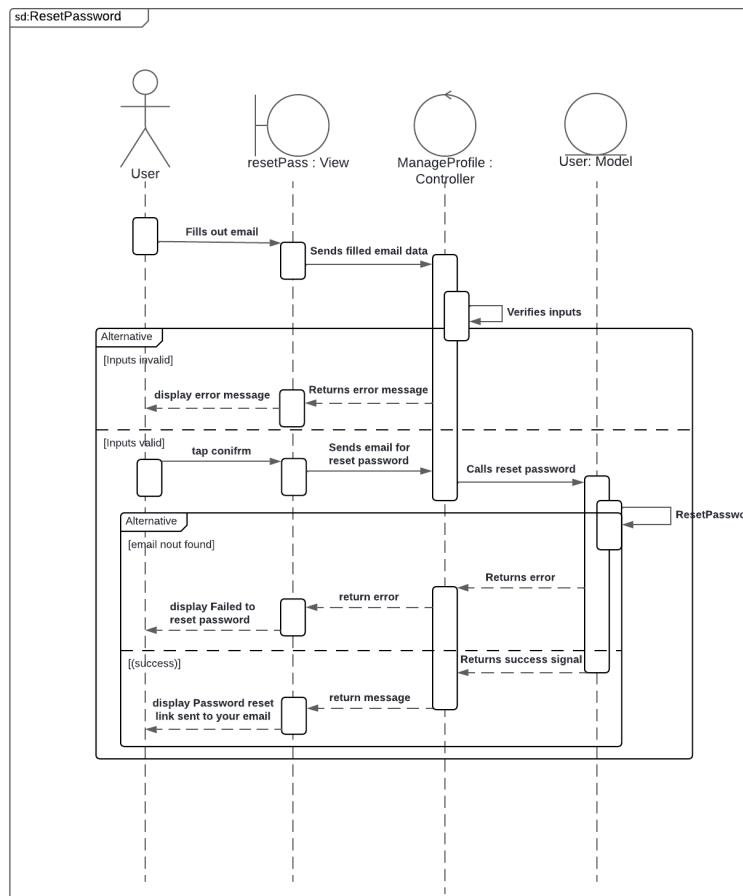
Figure 4-7 represents the " Update Password " sequence diagram.



**Figure 4-7 " Update Password " sequence diag diagram.**

#### 4.1.5 Sequence Diagram: Reset Password

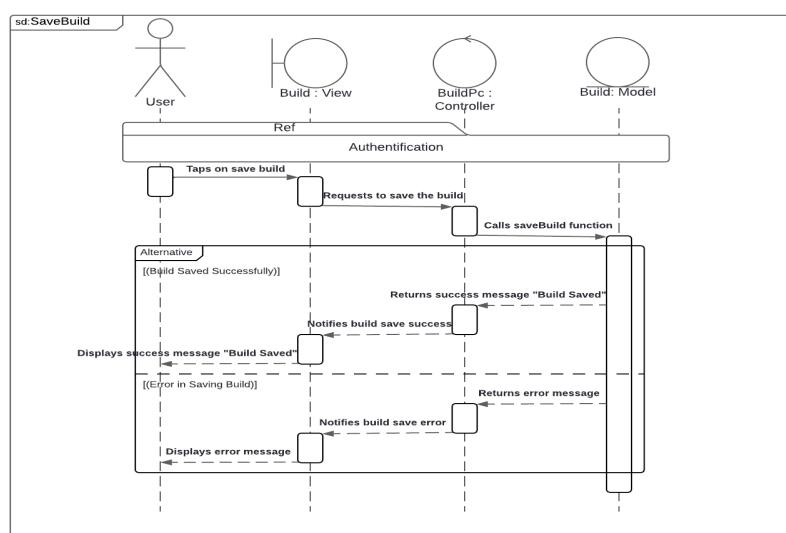
Figure 4-8 represents the "Reset Password" sequence diagram.



**Figure 4-8 "Reset Password" sequence diagram.**

#### 4.1.6 Sequence Diagram: Save Build

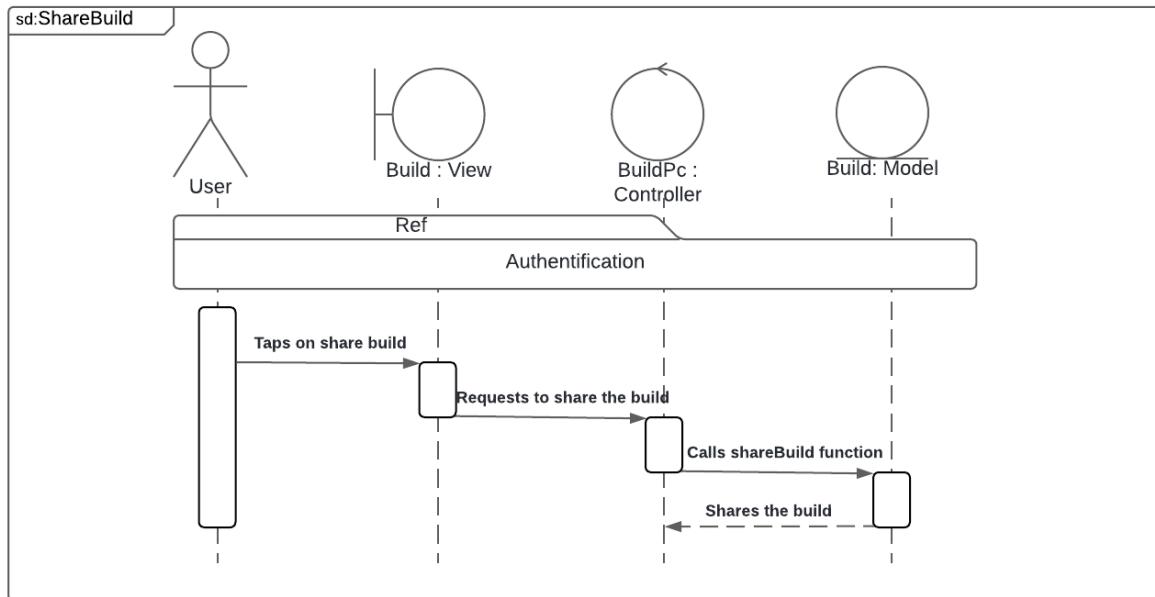
Figure 4-9 represents the "Save Build" sequence diagram.



**Figure 4-9 "Save Build" sequence diagram.**

#### 4.1.7 Sequence Diagram: Share Build

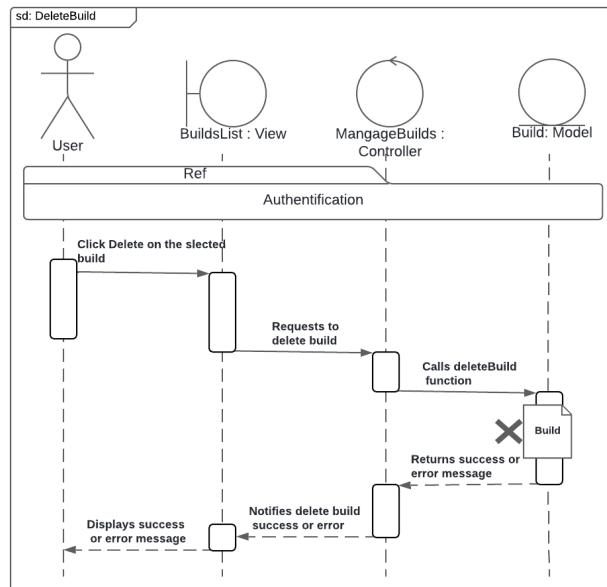
Figure 4-10 represents the "Share Build" sequence diagram.



**Figure 4-10 "Share Build" sequence diagram.**

#### 4.1.8 Sequence Diagram: Delete Build

Figure 4-11 represents the "Delete Build" sequence diagram.



**Figure 4-11 "Delete Build" sequence diagram.**

#### 4.1.9 Sequence Diagram: CreateManualBuild

Figure 4-12 represents the "CreateManualBuild" sequence diagram.

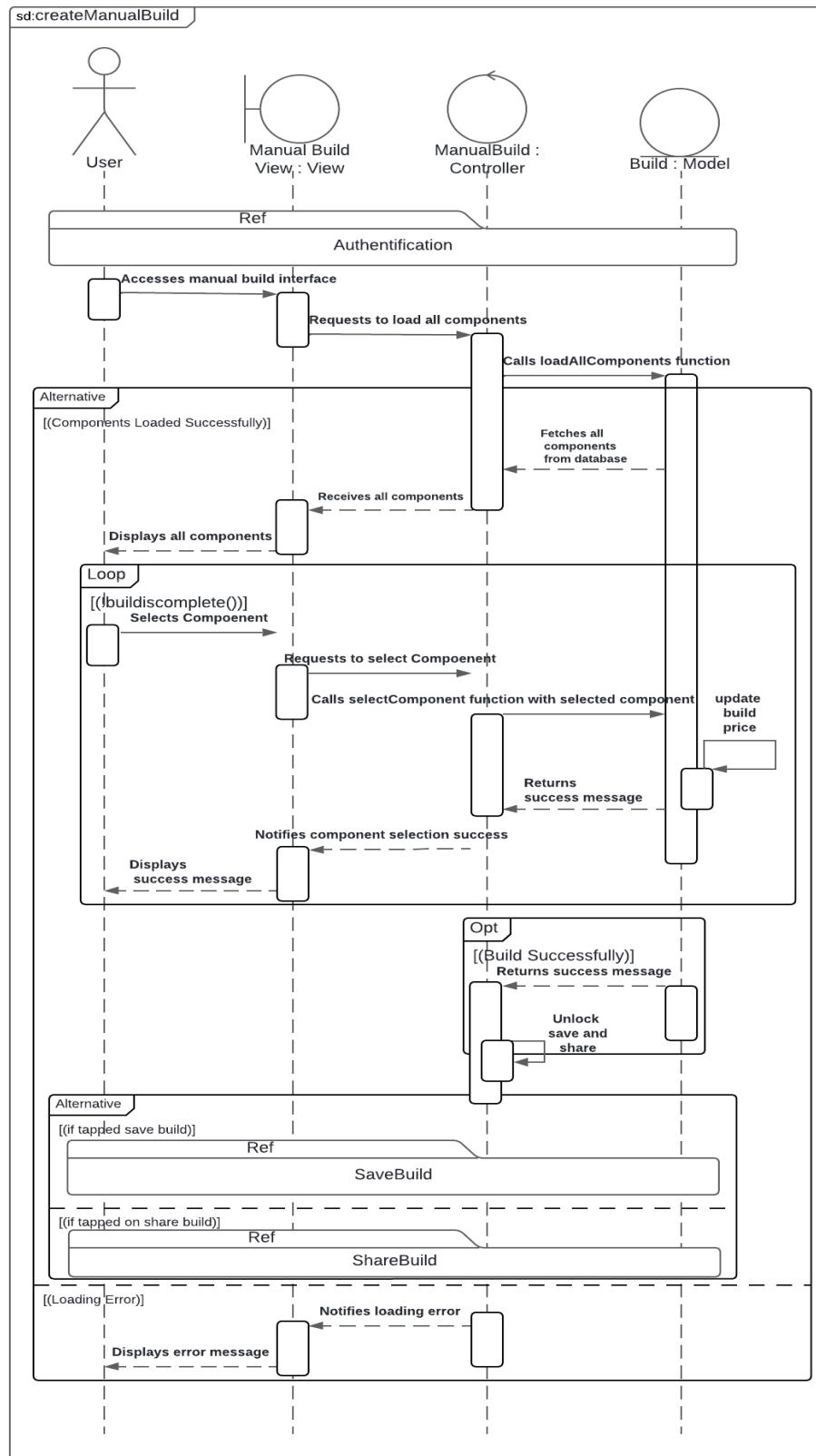


Figure 4-12 "CreateManualBuild" sequence diagram.

#### 4.1.10 Sequence Diagram: CreateIaBuild

Figure 4-13 represents the "CreateIaBuild" sequence diagram.

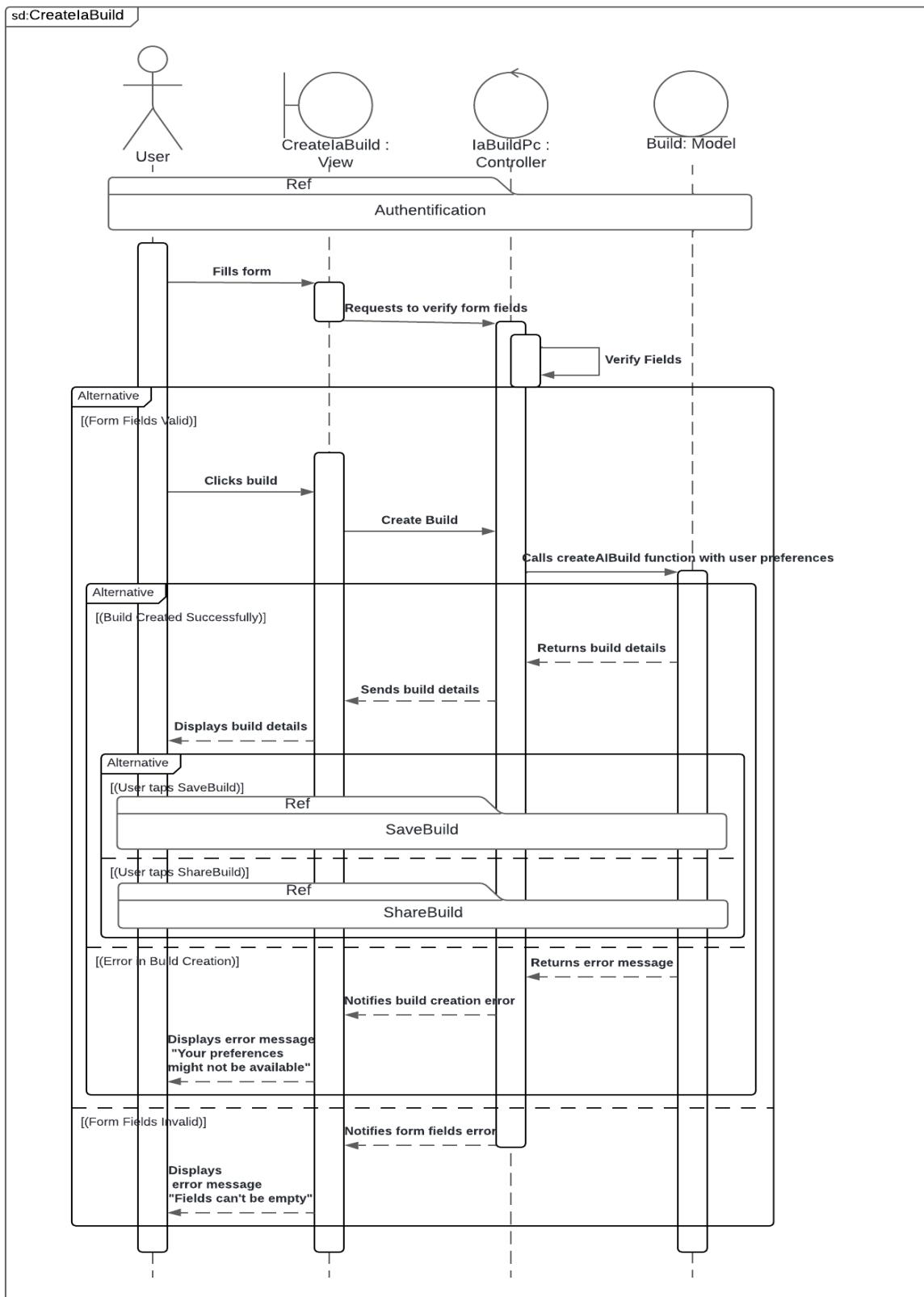
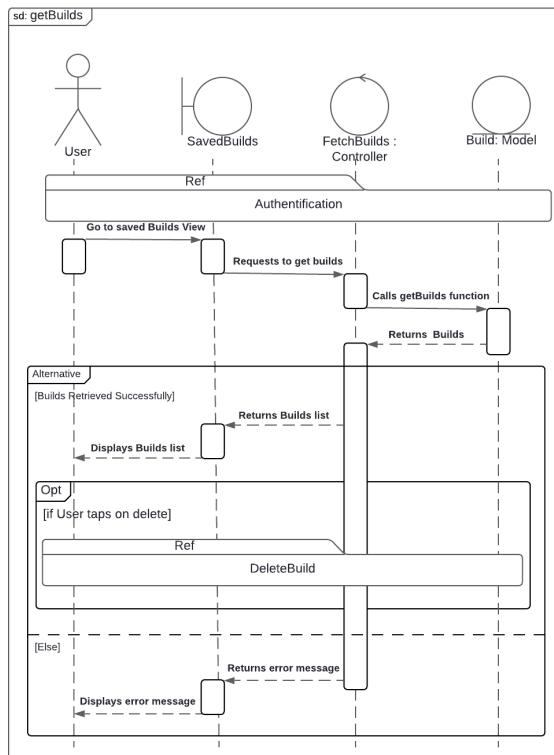


Figure 4-13 "CreateIaBuild" sequence diagram.

#### 4.1.11 Sequence Diagram: Get Build

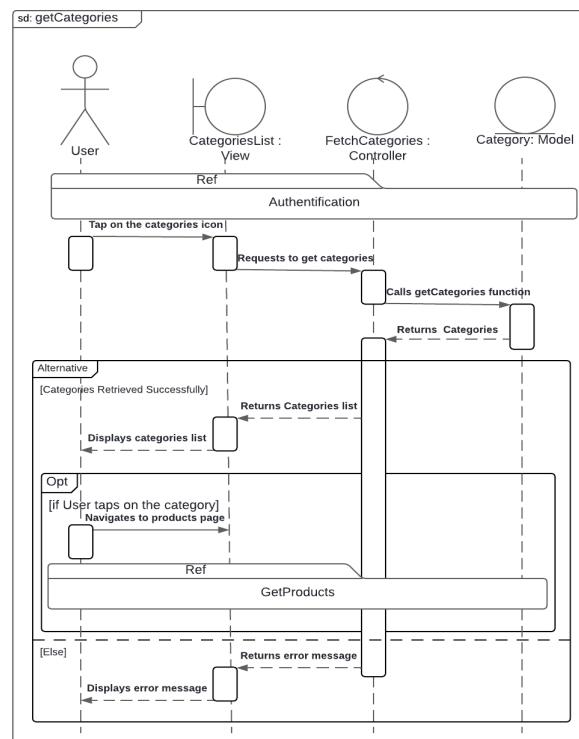
Figure 4-14 represents the " Get Build " sequence diagram.



**Figure 4-14 " Get Build " sequence diagram.**

#### 4.1.12 Sequence Diagram: Get Categories

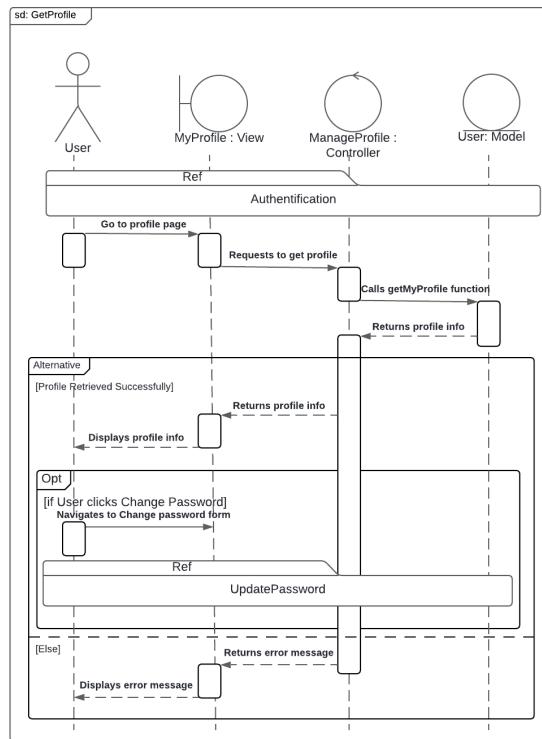
Figure 4-15 represents the " Get Categories " sequence diagram.



**Figure 4-15 " Get Categories " sequence diagram.**

#### 4.1.13 Sequence Diagram: Get Porfile

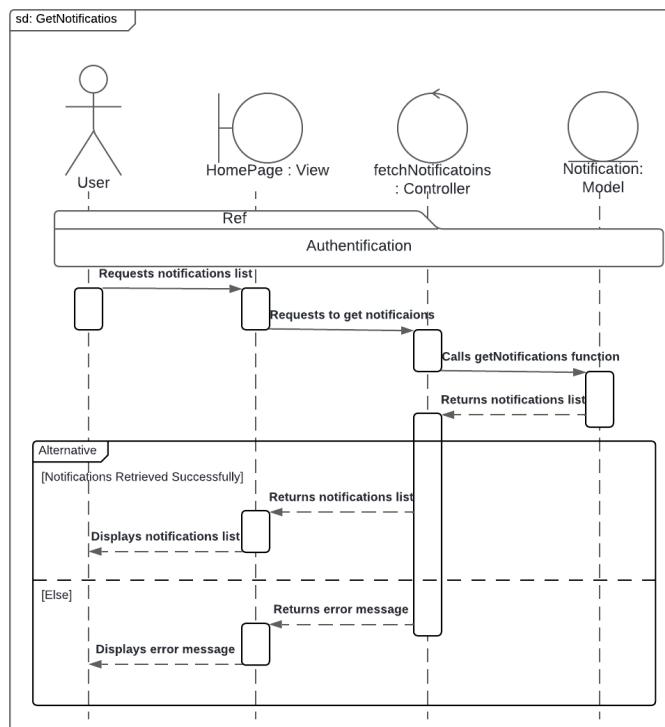
Figure 4-16 represents the " Get Porfile " sequence diagram



**Figure 4-16 " Get Porfile " sequence diagram.**

#### 4.1.14 Sequence Diagram: Get Notifications

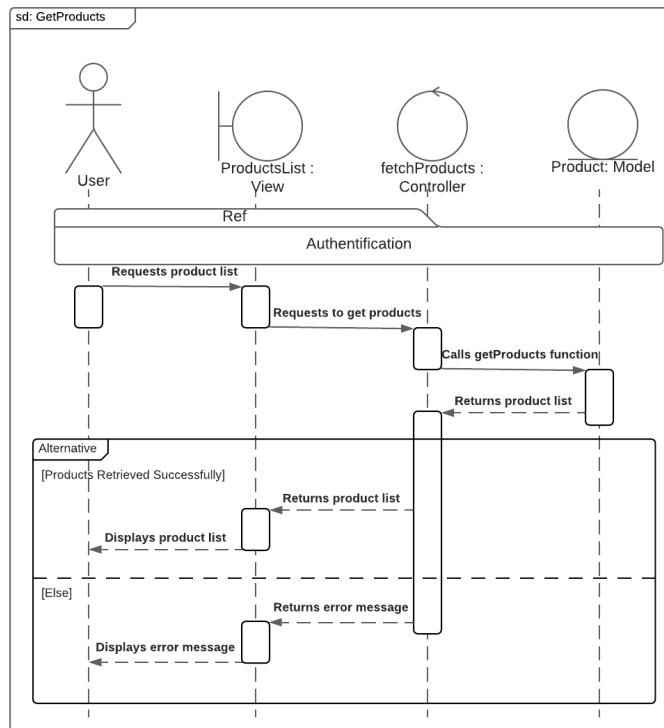
Figure 4-17 represents the " Get Notifications " sequence diagram.



**Figure 4-17 " Get Notifications " sequence diagram.**

#### 4.1.15 Sequence Diagram: Get Products

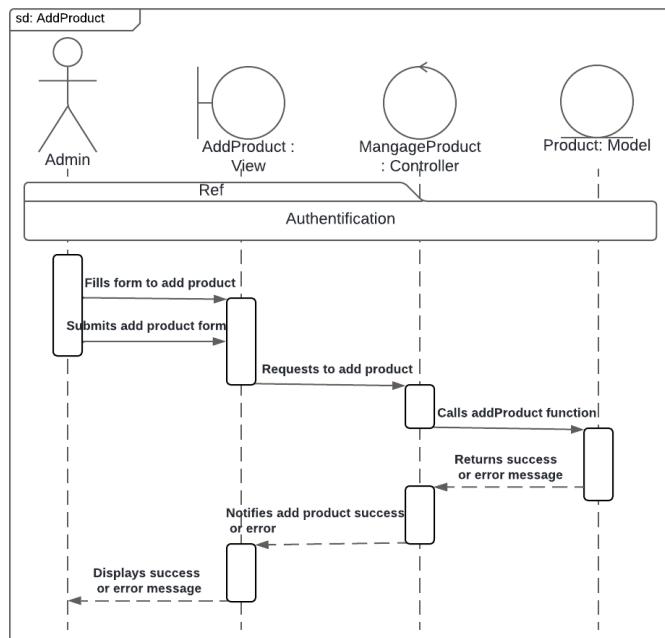
Figure 4-18 represents the " Get Products " sequence diagram.



**Figure 4-18 " Get Products " sequence diagram.**

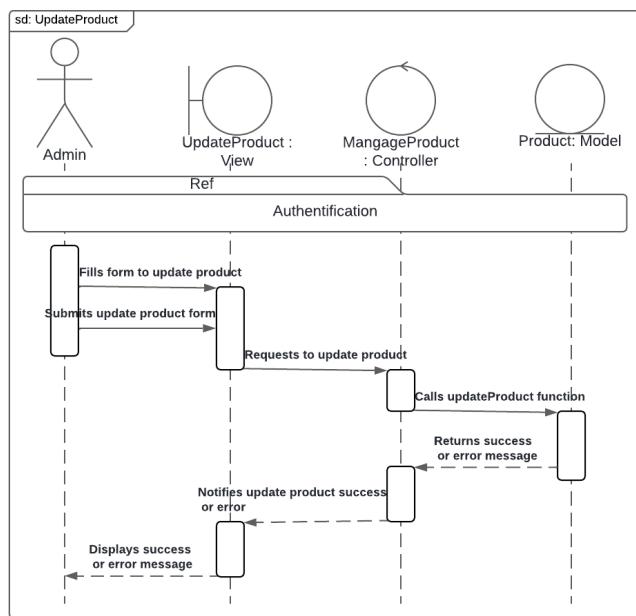
#### 4.1.16 Sequence Diagram: MangeProducts

Figure 4-19 represents the "Add Product" sequence diagram.



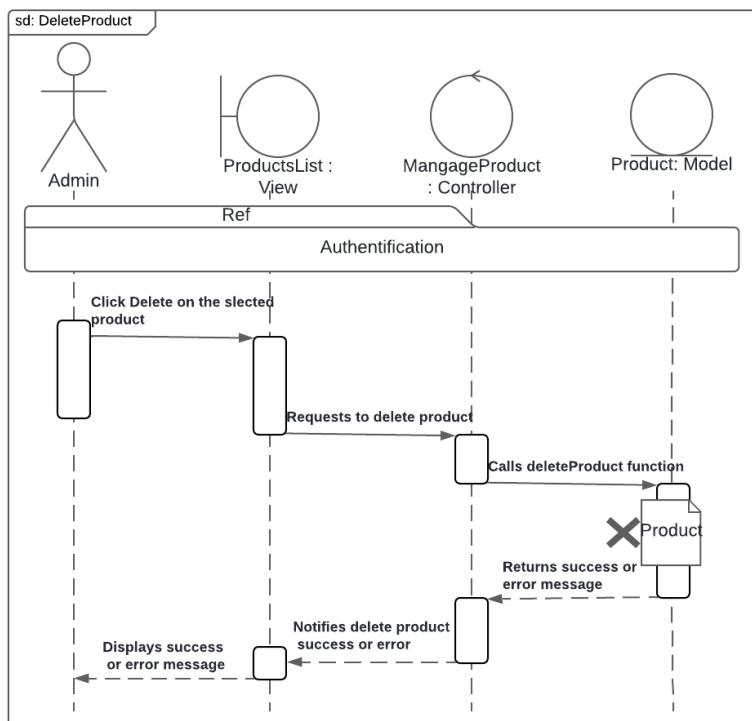
**Figure 4-19 " Add Product" sequence diagram.**

Figure 4-20 represents the "Update Product" sequence diagram.



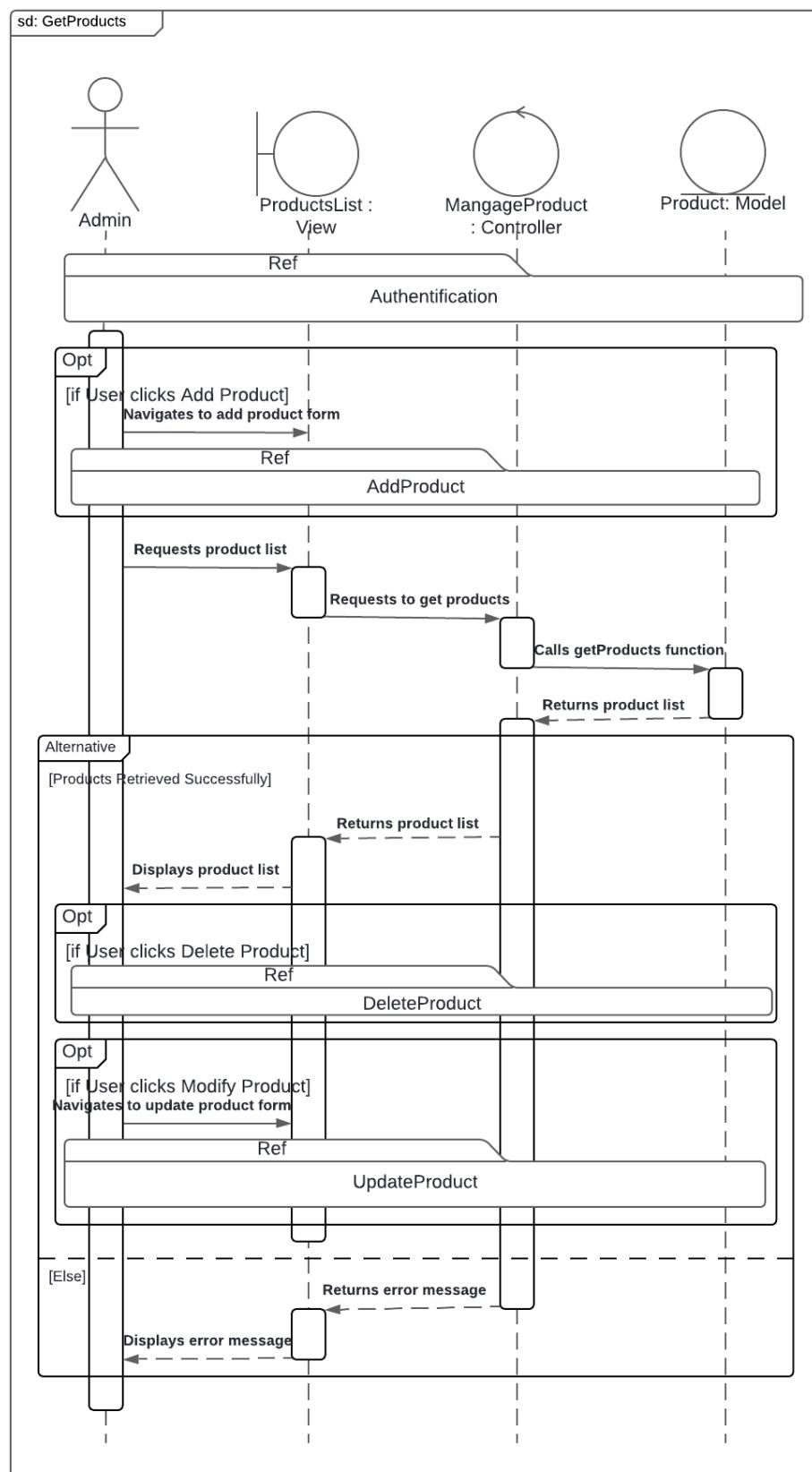
**Figure 4-20 "Update Product" sequence diagram.**

Figure 4-21 represents the "Delete Product" sequence diagram.



**Figure 4-21 "Delete Product" sequence diagram.**

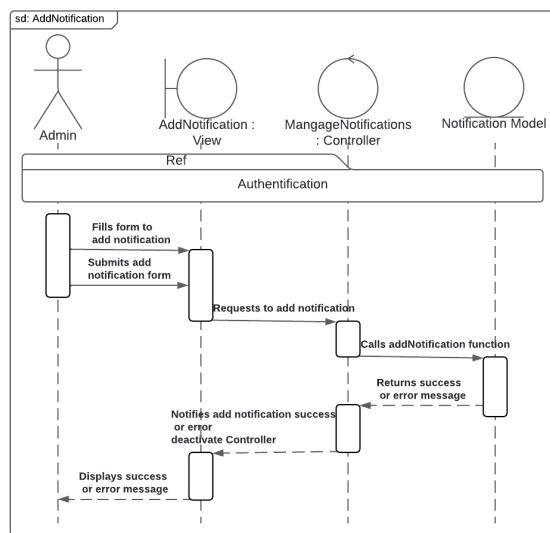
Figure 4-22 represents the "GetProducts" sequence diagram.



**Figure 4-22 "GetProducts" sequence diagram.**

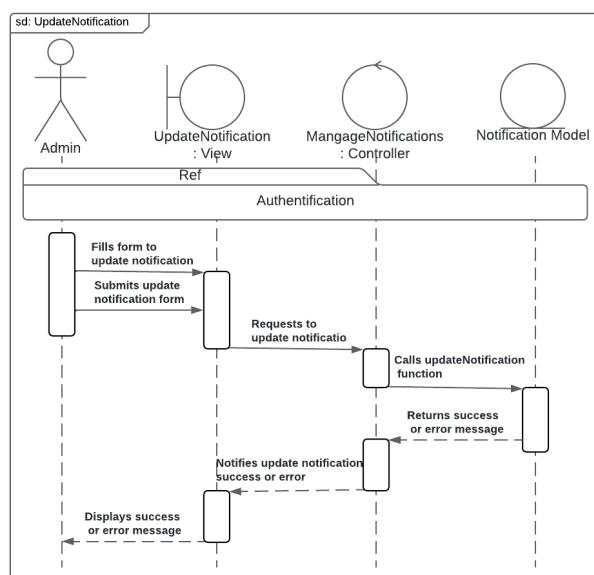
#### 4.1.17 Sequence Diagram: ManageNotifications

Figure 4-23 represents the "Add Notification" sequence diagram.



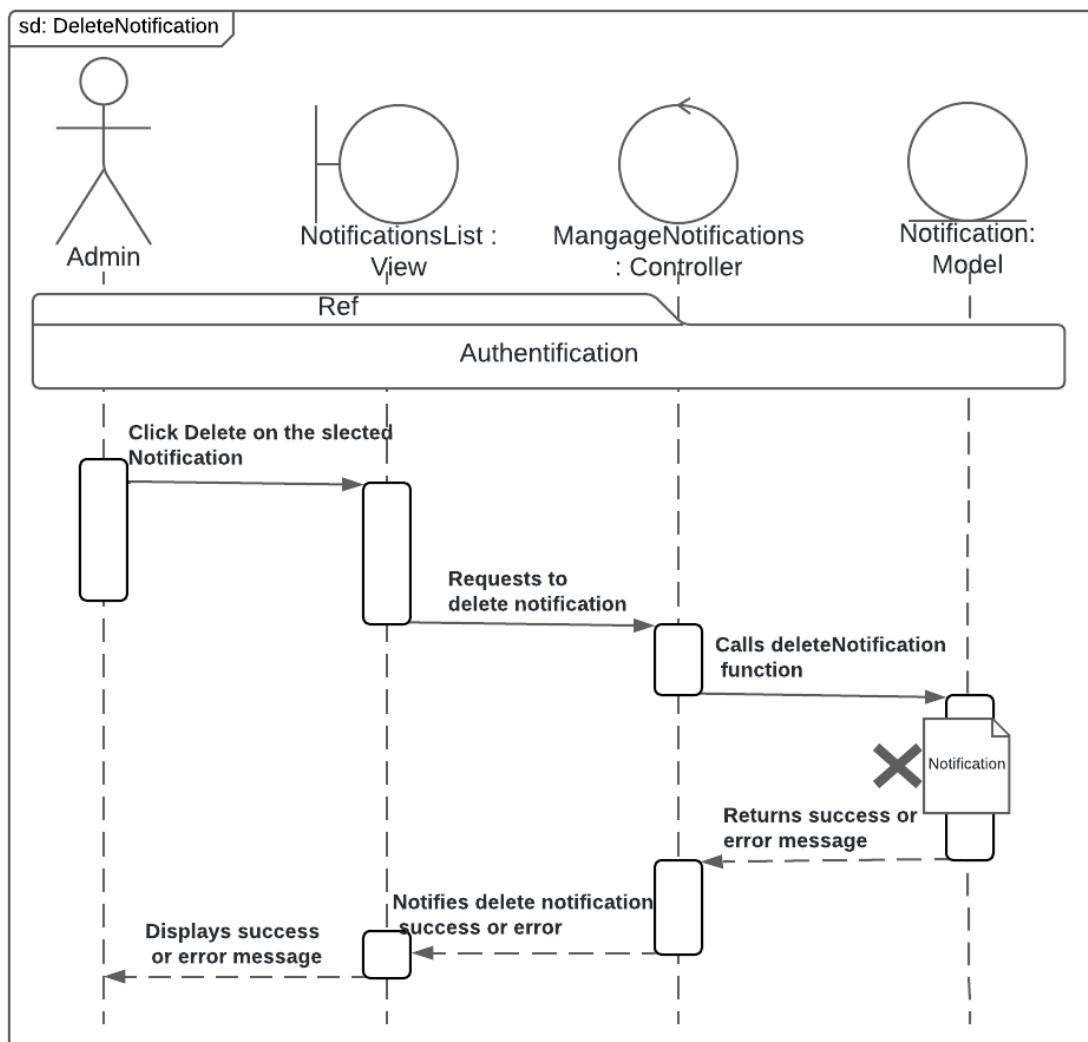
**Figure 4-23 "Add Notification" sequence diagram.**

Figure 4-24 represents the "Update Notification" sequence diagram.



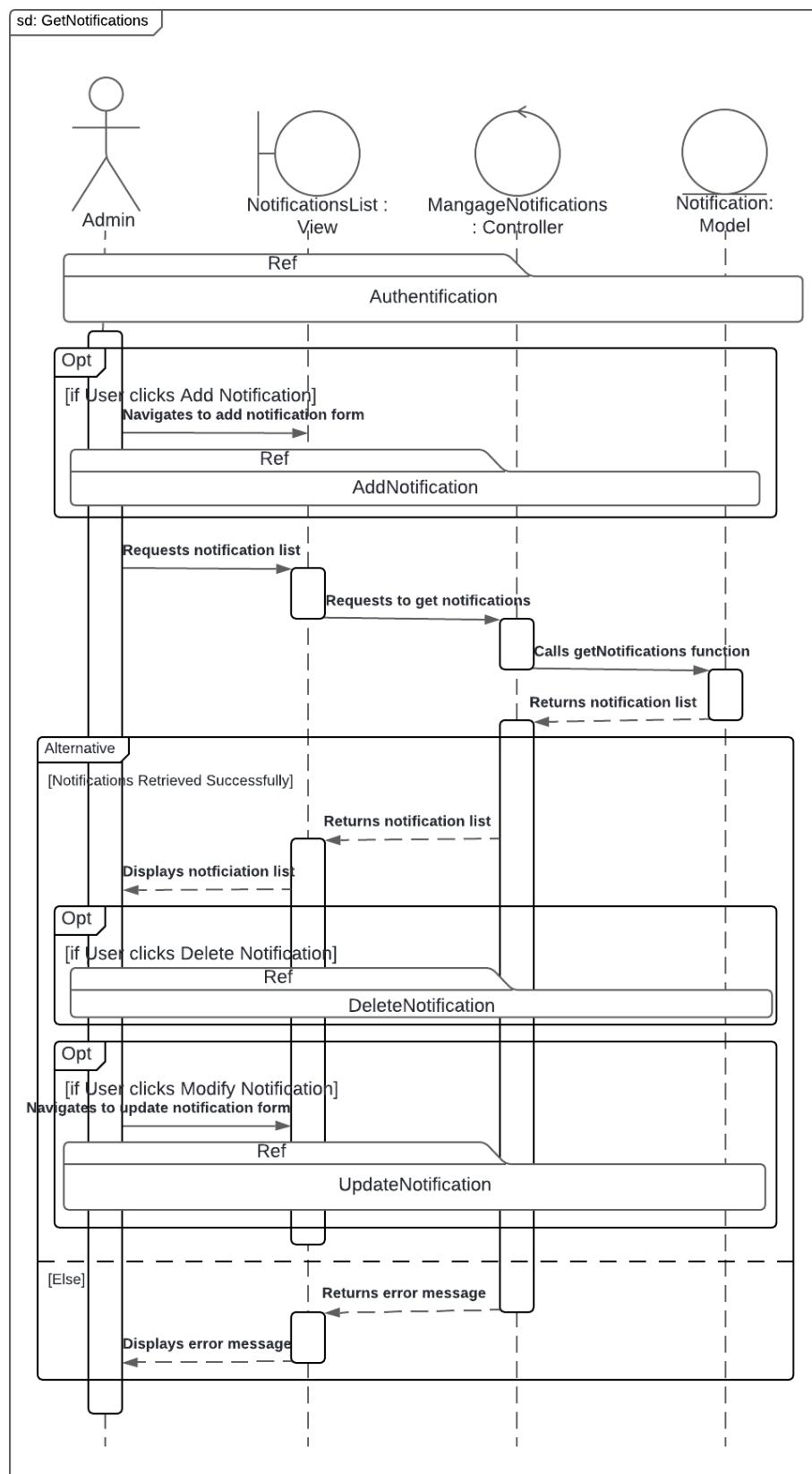
**Figure 4-24 "Update Notification" sequence diagram.**

Figure 4-25 represents the "Delete Notification" sequence diagram.



**Figure 4-25 "Delete Notification" sequence diagram.**

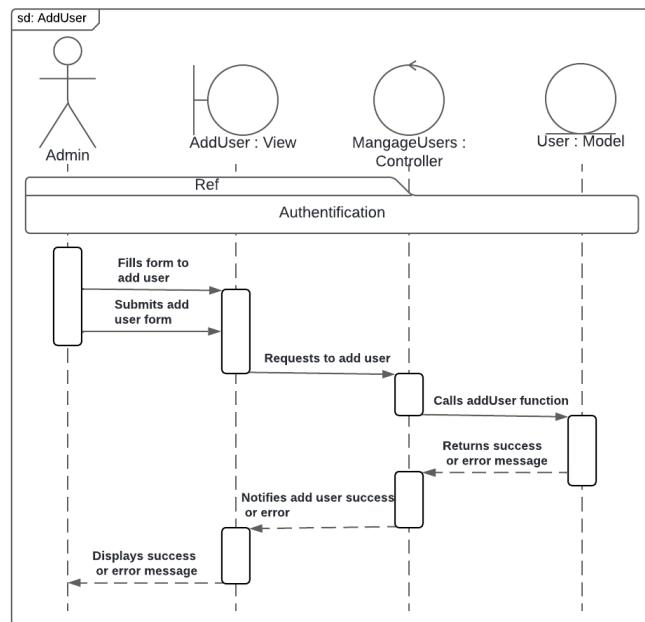
Figure 4-26 represents the "GetNotifications" sequence diagram.



**Figure 4-26 "GetNotifications" sequence diagram.**

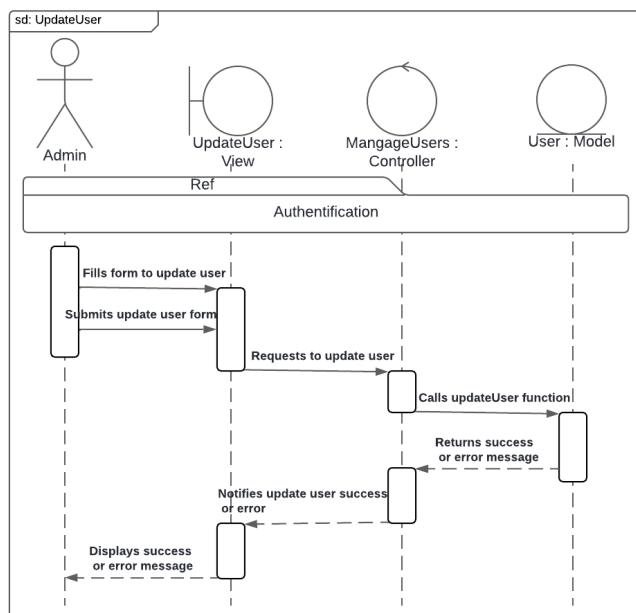
#### 4.1.18 Sequence Diagram: ManageUsers

Figure 4-27 represents the "Add User" sequence diagram.



**Figure 4-27 "Add User" sequence diagram.**

Figure 4-28 represents the "Update User" sequence diagram.



**Figure 4-28 "Update User" sequence diagram.**

Figure 4-29 represents the "Delete User" sequence diagram.

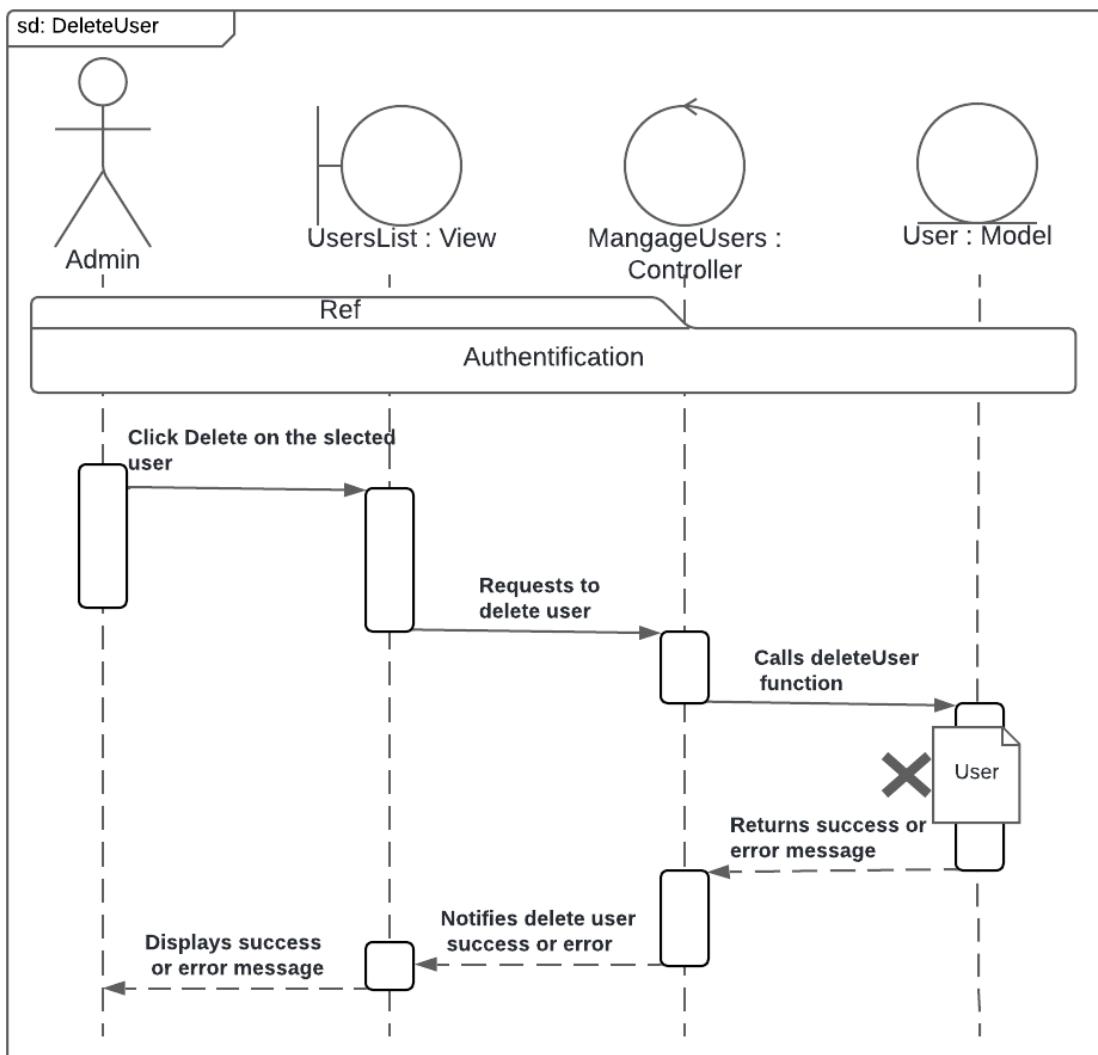


Figure 4-29 "Delete User" sequence diagram.

Figure 4-30 represents the "GetUsers" sequence diagram.

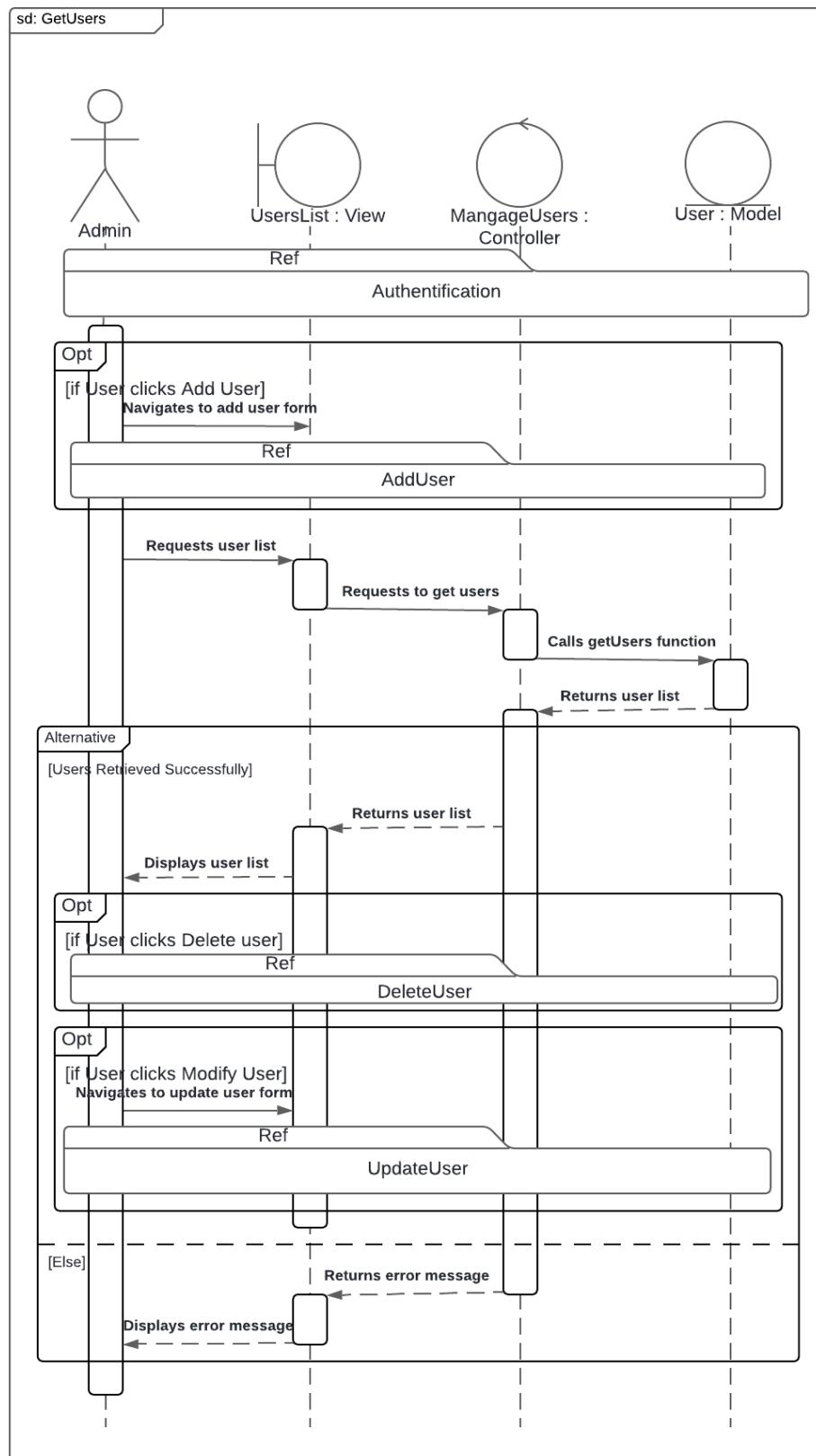
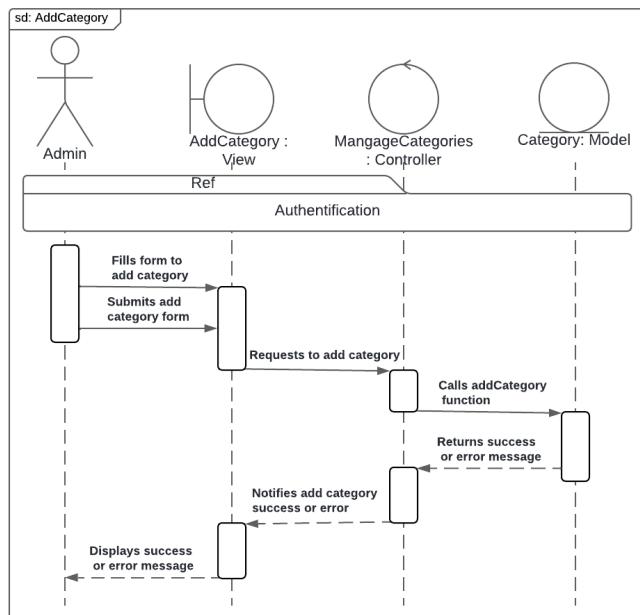


Figure 4-30 "GetUsers" sequence diagram.

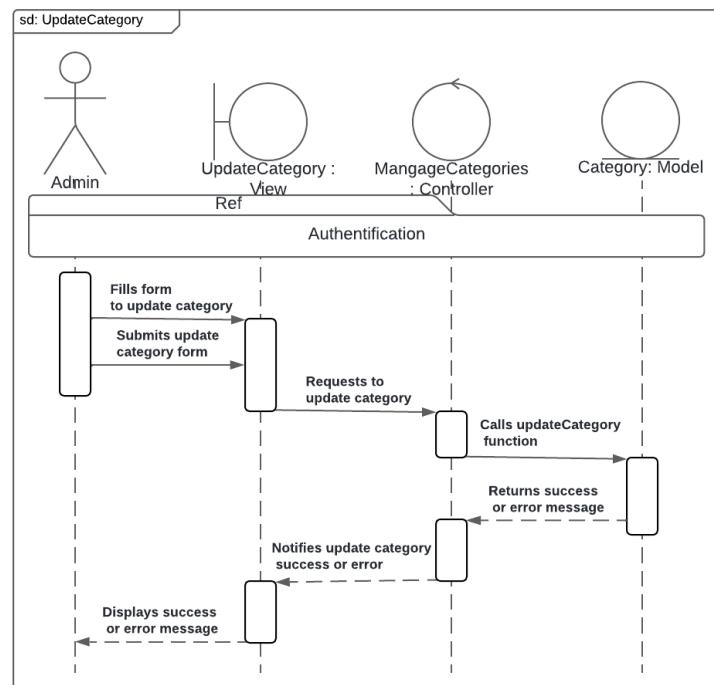
#### 4.1.19 Sequence Diagram: ManageCategories

Figure 4-31 represents the "Add Category" sequence diagram.



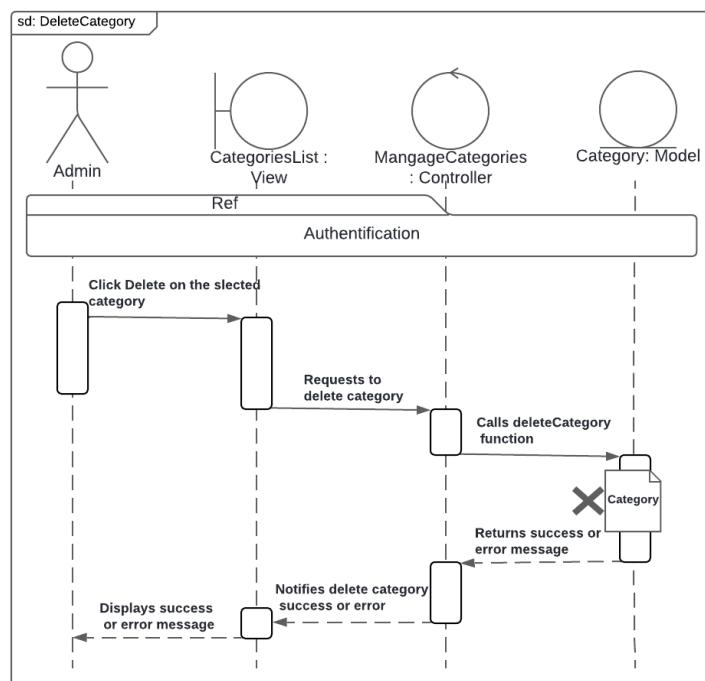
**Figure 4-31 "Add Category" sequence diagram.**

Figure 4-32 represents the "Update Category" sequence diagram.



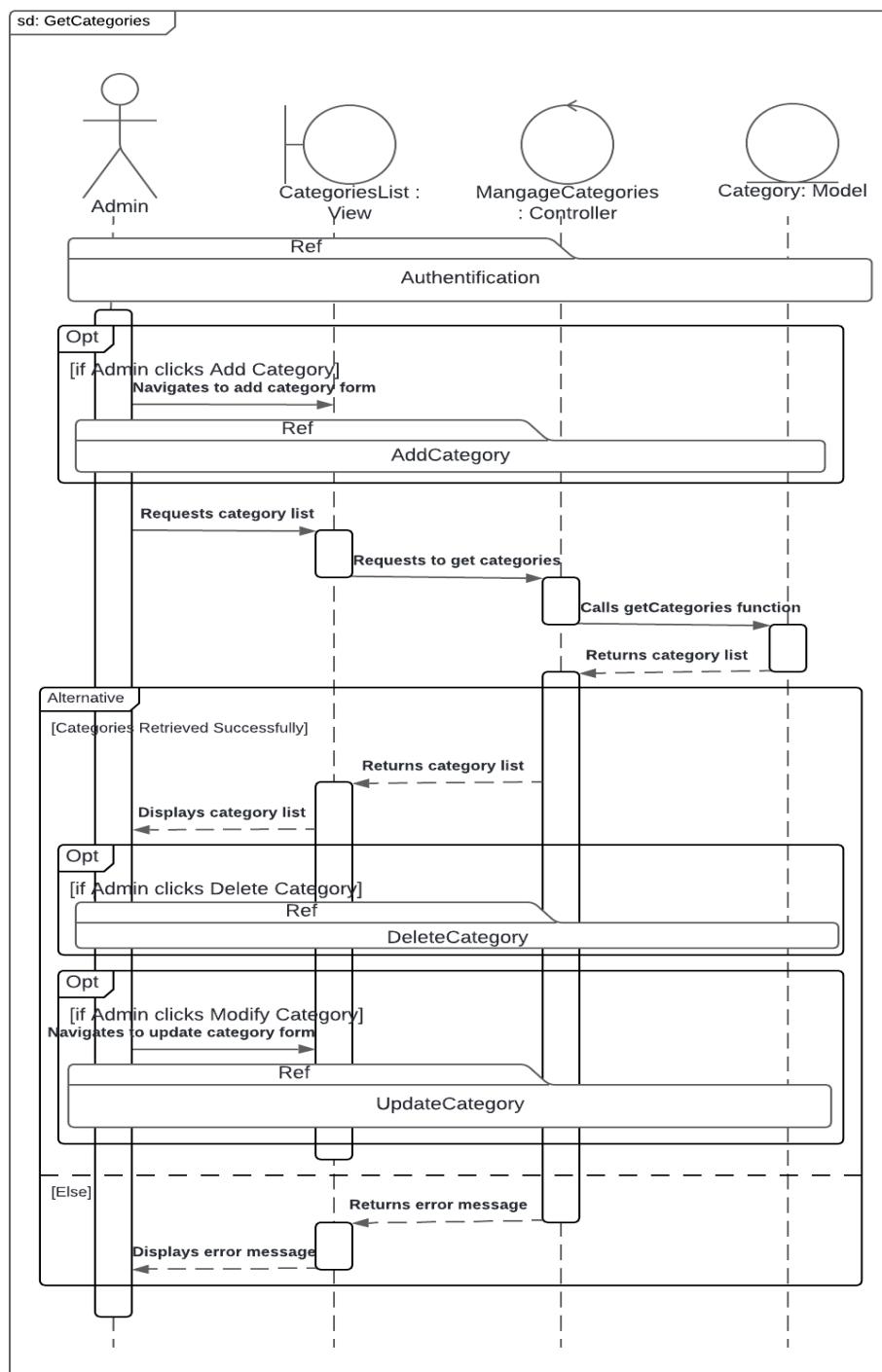
**Figure 4-32 " Update Category " sequence diagram.**

Figure 4-33 represents the "Delete Category" sequence diagram.



**Figure 4-33 "Delete Category" sequence diagram.**

Figure 4-34 represents the "GetCategories" sequence diagram.



**Figure 4-34 "GetCategories" sequence diagram.**

## Conclusion

Chapter 4 detailed the architectural design, including class and sequence diagrams for each function. These diagrams illustrate the system's structure and interactions, ensuring the project is well-organized and meets the specified requirements.

# Chapter 5 Implementaion

## Introduction

This chapter details the implementation of the custom PC build system. It describes the chosen technologies, development tools, and the implementation process. It also covers testing strategies to ensure the system functions as intended.

## 1. Technology Stack

This section outlines the programming languages, frameworks, libraries, and database technologies used to develop the application.

### 1.1 Programming Language

This subsection specifies the programming languages used for both the backend and mobile app development.

Logo	Language Name	Description
	TypeScript	<p><b>TypeScript</b> is a free and open-source high-level programming language developed by Microsoft that adds static typing with optional type annotations to JavaScript. It is designed for the development of large applications and transpiles to JavaScript. (<a href="#">Wikipedia</a>)</p> <p><b>Developer:</b> <a href="#">Microsoft</a></p> <p><b>Filename extensions:</b> ts,.tsx,.mts,.cts</p> <p><b>First appeared:</b> 1 October 2012;</p>

		12 years ago
	Dart	<p><b>Dart</b> is a programming language designed by Lars Bak and Kasper Lund and developed by Google. It can be used to develop web and mobile apps as well as server and desktop applications. Dart is an object-oriented, class-based, garbage-collected language with C-style syntax. (<a href="#">Wikipedia</a>)</p> <p><b>Developer:</b> <a href="#">Google</a></p> <p><b>Filename extensions:</b> dart</p> <p><b>First appeared:</b> October 10, 2011; 13 years ago</p>

Table 5-1 Programming Language Table

## 1.2 Frameworks and Library

This subsection details the frameworks and libraries employed for building the backend and mobile app components.

Logo	Framework Name	Description
	Express	<p><b>Express.js</b>, or simply <b>Express</b>, is a back end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js. (<a href="#">Wikipedia</a>)</p> <p><b>Developer:</b> <a href="#">Strongloop</a>, <a href="#">IBM</a>  <b>Initial release date:</b> November 16, 2010; 14 years ago  <b>Programming language:</b> <a href="#">JavaScript</a> (<a href="#">TypeScript</a> configurable)  <b>Runtime Environment:</b> <a href="#">Node.js</a></p>
	Flutter	<p><b>Flutter</b> is an open-source UI software development kit created by Google. It can be used to develop cross platform applications from a single codebase for the web, Fuchsia, Android, iOS, Linux, macOS, and Windows. First described in 2015, Flutter was released in May 2017. (<a href="#">Wikipedia</a>)</p> <p><b>Initial release date:</b> May 2017 ; 7 years ago  <b>Programming languages:</b> <a href="#">Dart</a>, <a href="#">C</a>, <a href="#">C++</a>  <b>Platforms:</b> <a href="#">Android</a>, <a href="#">iOS</a>, <a href="#">Fuchsia</a>, <a href="#">Web platform</a>, <a href="#">macOS</a>, <a href="#">Microsoft Windows</a>, <a href="#">Linux</a>  <b>Developer(s):</b> <a href="#">Google</a> and community</p>

**Table 5-2 Frameworks and Library Table**

### 1.3 Database Technology

This subsection identifies the database technology used to store and manage the application's data.

Logo	Database Name	Description
	Firebase	<p><b>Firebase</b> is a set of backend cloud computing services and application development platforms provided by Google. It hosts databases, services, authentication, and integration for a variety of applications, including Android, iOS, JavaScript, Node.js, Java, Unity, PHP, and C++. (<a href="#">Wikipedia</a>)</p> <p><b>Founders:</b> James Tamplin, Andrew Lee</p> <p><b>Founded:</b> 2011; 13 years ago</p> <p><b>Parent:</b> <a href="#">Google</a></p>
	SQFlite	<p><b>SQFlite</b> is a Flutter plugin for <b>SQLite</b> databases utilised for local storage management. It provides a simple, easy-to-use interface to perform standard database operations like CRUD (Create, Read, Update, Delete)</p> <p><b>SQLite</b> is a database engine written in the C programming language. It is not a standalone app; rather, it is a library that software developers embed in their apps. As such, it belongs to the family of embedded databases. (<a href="#">Wikipedia</a>)</p>

		<p><b>Developer:</b> <u>D. Richard Hipp</u></p> <p><b>License:</b> <u>Public domain</u></p> <p><b>Size:</b> 699 KiB</p> <p><b>Written in:</b> <u>C</u></p>
--	--	--

**Table 5-3 Database Technology Table**

## 1.4 Cloud Services

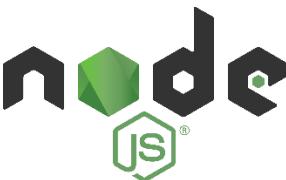
In this section, we will discuss the cloud services utilized in our project,

Logo	Service Name	Description
	AWS Lightsail	<ul style="list-style-type: none"> <li>- AWS is a comprehensive cloud platform offering over 200 fully-featured services from data centers globally. It provides scalable, flexible, and cost-effective solutions for a wide range of applications.</li> <li>- AWS Lightsail is a simplified cloud platform designed for developers to easily deploy and manage virtual private servers (VPS). It offers predictable pricing, scalability, and integration with the AWS ecosystem, making it ideal for hosting applications, websites, and databases.</li> </ul>

**Table 5-4 Cloud Services Table**

## 1.5 Other Technologies

This subsection lists any additional technologies or tools used in the application's development.

Logo	Tech Name	Description
	Node.js	<p><b>Node.js</b> is a cross-platform, open-source JavaScript runtime environment that can run on Windows, Linux, Unix, macOS, and more. Node.js runs on the V8 JavaScript engine, and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting. (<a href="#">Wikipedia</a>)</p> <p><b>Programming languages:</b></p> <p><a href="#">JavaScript</a>, <a href="#">Python</a>, <a href="#">V</a>, <a href="#">C++</a>, <a href="#">CoffeeScript</a></p> <p><b>Developer:</b> Ryan Dahl, OpenJS Foundation, Bryan Cantrill</p> <p><b>Initial release:</b> May 27, 2009; 14 years ago</p>
	Postman	<p><b>Postman</b> is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster. (<a href="#">Wikipedia</a>)</p> <p><b>Initial release:</b> 2014; 12 years ago</p>

	<b>QEMU</b>	<p><b>QEMU</b> is a free and open-source emulator. It emulates a computer's processor through dynamic binary translation and provides a set of different hardware and device models for the machine, enabling it to run a variety of guest operating systems. (<a href="#">Wikipedia</a>)</p> <p><b>Developer:</b> Fabrice Bellard</p> <p><b>Operating system:</b> <a href="#">Linux</a>, <a href="#">Microsoft Windows</a>, <a href="#">macOS</a> and some other <a href="#">UNIX</a> platforms</p> <p><b>Release date:</b> January 2007; 14 years ago</p> <p><b>Written in:</b> <a href="#">C</a></p>
	<b>Miro</b>	<p><b>Miro</b> is a web-based collaborative whiteboard that simplifies wireframing UI designs. It offers a large workspace, pre-built templates, and a shape library for building wireframes quickly. Real-time collaboration features and commenting tools enable easy feedback sharing within the team.</p>
	<b>Figma</b>	<p><b>Figma</b>, a web-based design powerhouse, simplifies the process of creating high-fidelity user interfaces (UIs). Its user-friendly interface and extensive tools allow for crafting visually rich interfaces. Collaboration features enable real-time feedback and iteration.</p>

**Table 5-5 Other Technologies Table**

## 2. Development Environment

This section provides an overview of the hardware and software tools used for developing the application.

### 2.1 Hardware Environment

This subsection specifies the hardware specifications of the computers used for development, including processor, RAM, storage, and operating system.

Computer	Specs	
Model	MSI GF63 Thin	MSI GF63 Thin
Memory	8 GB Ram	16 GB Ram
CPU	Intel Core i5-10300H @ 2.50GHz	Intel Core i7-10750H @ 2.60GHz
GPU	Nvidia GeForce GTX 1650 TI with max-Q design	Nvidia GeForce GTX 1650 TI with max-Q design
Storage	1 To HDD + 512 Gb SSD	512 Gb SSD
Operating System	Windows 10 + Kali Linux ( <u>wsl</u> )	Windows 10
Mobile	Specs	
Model	Samsung Galaxy Z Fold 3	
Memory	12 GB RAM	
CPU	Qualcomm Snapdragon 888	
GPU	Adreno 660	
GPU	256 GB / 512 GB	
Operating System	Android 11, upgradable to Android 13 with One UI 5.1	

**Table 5-6 Hardware Environment Table**

## 2.2 Development Tools

This subsection lists the software tools, such as code editors, IDEs, and command-line tools, used for development

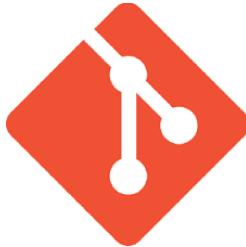
Logo	Tool	Description
	Visual Studio Code	<p><b>Visual Studio Code</b>, also commonly referred to as VS Code, is a source-code editor developed by Microsoft for Windows, Linux, macOS and web browsers. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded version control with Git. (<a href="#">Wikipedia</a>)</p> <p><b>Operating system:</b>  <u>Windows</u>, <u>macOS</u>, <u>Linux</u> (including ChromeOS)</p> <p><b>Initial release:</b> April 29, 2015; 9 years ago.</p>
	Android Studio	<p><b>Android Studio</b> is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. (<a href="#">Wikipedia</a>)</p> <p><b>Programming languages:</b>  <u>Kotlin</u>, <u>JavaScript</u>, <u>Dart</u>, <u>C++</u></p> <p><b>Developer:</b> <u>Google</u>, <u>JetBrains</u></p> <p><b>Operating system:</b></p>

		<u>Windows</u> , <u>macOS</u> , <u>Linux</u> (including ChromeOS) <b>Initial release:</b> May 16, 2013; 11 years ago.
--	--	--

**Table 5-7 Development Tools Table**

### 2.3 Version Control System

This subsection identifies the version control system used to track and manage changes to the application's codebase to ensure collaborative workflow.

Logo	Tool	Description
	Git	<b>Git</b> is a distributed version control system that tracks versions of files. It is often used to control source code by programmers collaboratively developing software. ( <a href="#">Wikipedia</a> ) <b>Programming languages:</b> <u>Python</u> , <u>C</u> , <u>C++</u> , <u>Perl</u> , <u>Shell script</u> , <u>Tcl</u> <b>Operating system:</b> <u>POSIX</u> ( <u>Linux</u> , <u>macOS</u> , <u>Solaris</u> , <u>AIX</u> ), <u>Windows</u> <b>Initial release:</b> 7 April 2005; 19 years ago

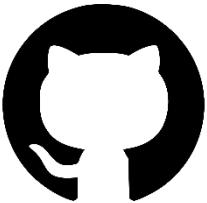
	GitHub	<p><b>GitHub</b> is a developer platform that allows developers to create, store, manage and share their code. It uses Git software, providing the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project. (<a href="#">Wikipedia</a>)</p> <p><b>Parent:</b> Microsoft</p> <p><b>Founded:</b> February 2008; 16 years ago.</p>
---	--------	---

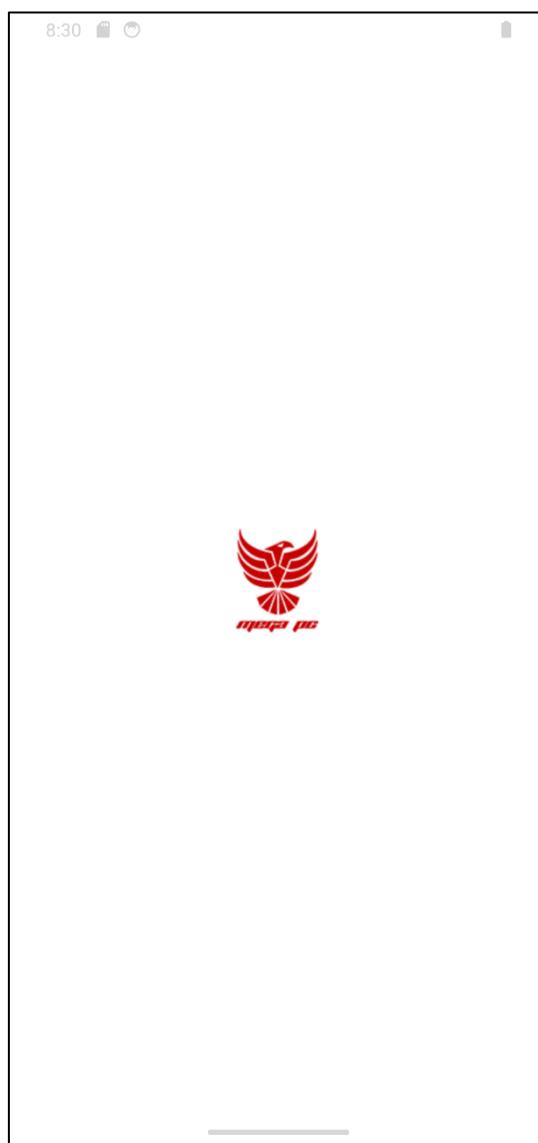
Table 5-8 Version Control System Table

### 3. Application Interfaces

This section will explore the user interfaces (UIs) of the custom PC build system, highlighting the interactive elements that guide users through the application.

#### 3.1 Splash Screen

When launching the application, a splash screen displaying the application's name and logo appears. This brief visual introduction serves as a brand reminder and sets the stage for the user experience.



**Figure 5-1** Splash screen



Figure 5-2 Getting Started Screen

### 3.2 Authentication

Following the splash screen, the user encounters the authentication interface. This section prompts users to either log in with existing credentials or create a new account. The interface prioritizes clarity and intuitiveness, ensuring a smooth and secure login/signup process.

### 3.2.1 “Login” Interface

The login interface provides the initial gateway to the custom PC build system. Here, users with existing accounts can enter their credentials to access the application's functionalities.

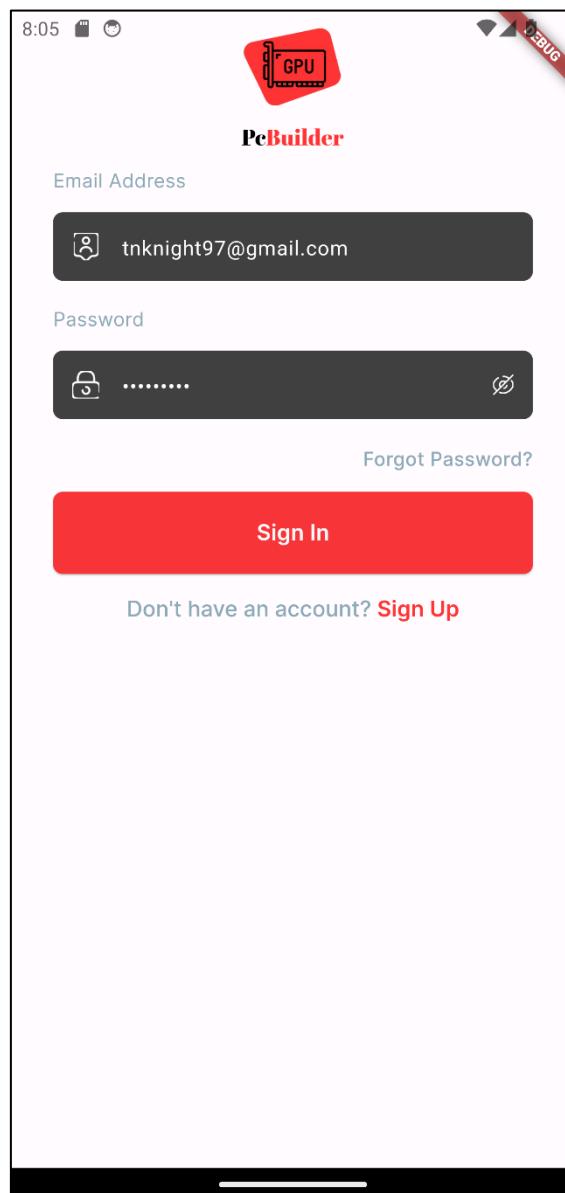


Figure 5-3 Login screen

### 3.2.2 “SignUp” Interface

The signup interface enables new users to create an account within the PC builder system. Here, users can establish their login credentials by entering an email address, choosing a secure password and filling out the rest of the forum with their informations.

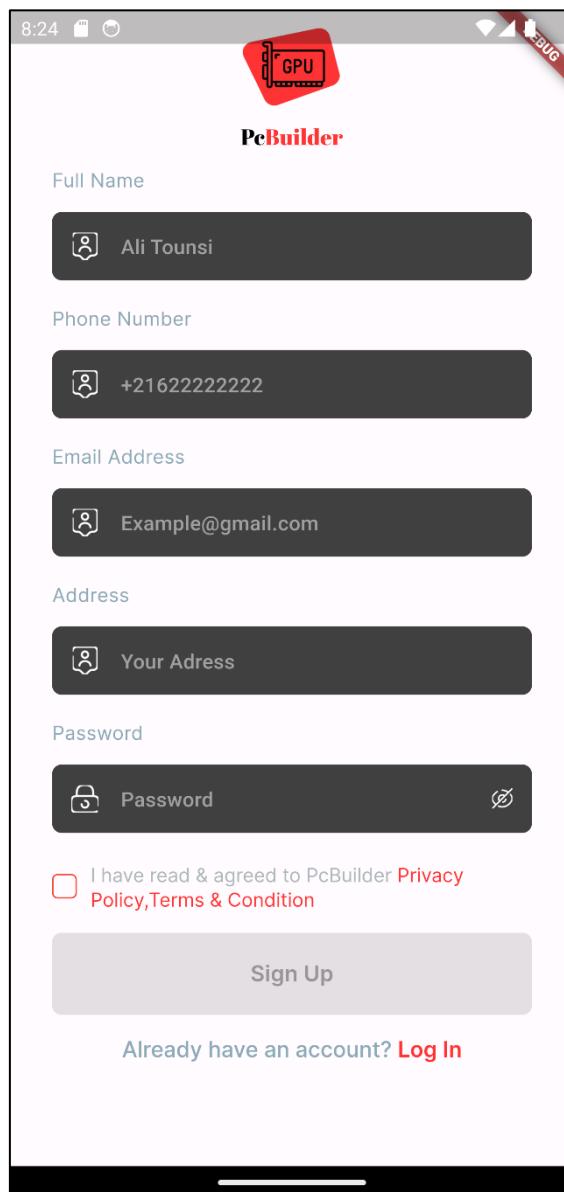
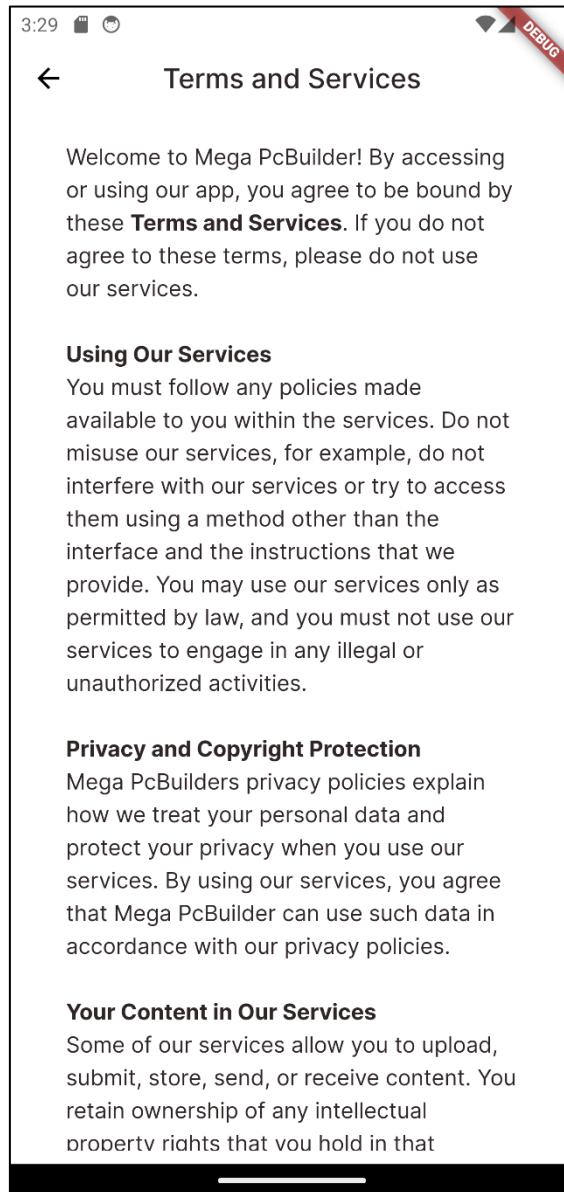


Figure 5-4 Signup screen

### 3.2.3 “Terms and Services” Interface

The Terms and Services interface presents the legal agreement outlining the conditions of use for the custom PC build system.

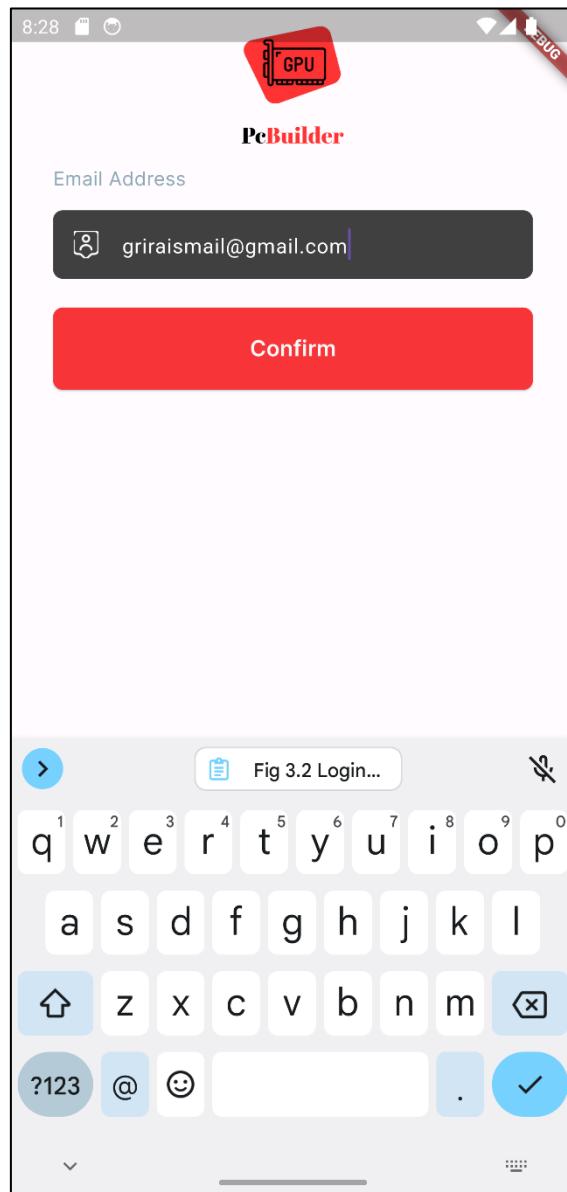


**Figure 5-5 Terms and Services Screen**

This interface allows users to carefully review the terms and conditions before agreeing to them. By doing so, users acknowledge their understanding and acceptance of the system's guidelines and limitations. It's recommended to read these terms thoroughly before proceeding.

### 3.2.4 “Recover Account” Interface

The Recover Account interface serves as a security measure in case users forget their login password. Upon selecting "forgot password," users are directed here. This interface prompts users to enter the email address associated with their account.



**Figure 5-6 Recover Account Screen**

By providing the registered email, users can initiate the password recovery process. The system will then send a password reset link to the entered email address.

### 3.2.5 “Reset Password” Interface

The Reset Password interface allows users to establish a new password for their account. This interface is typically accessed after a user initiates password recovery through the "Recover Account" function.

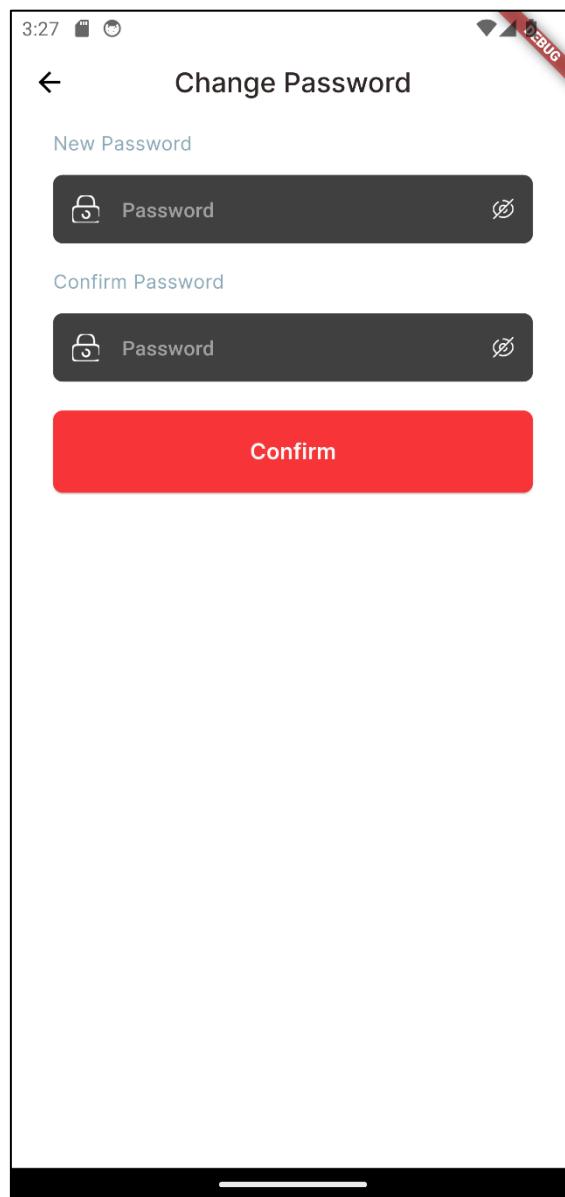


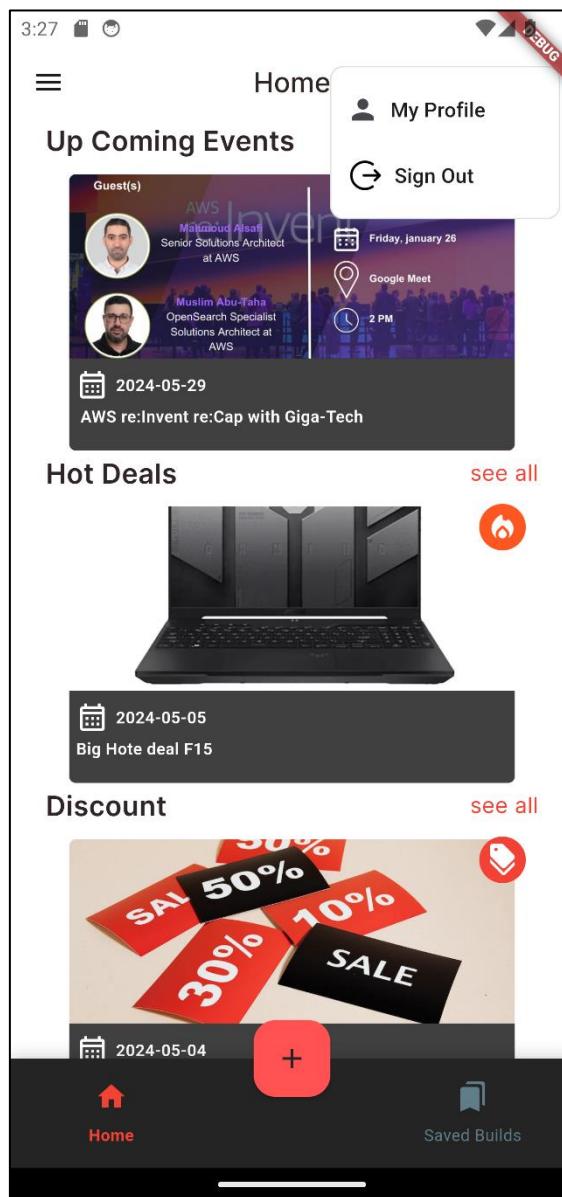
Figure 5-7 Reset Password Screen

### 3.3 Home Screen

The Home Screen serves as the central hub of the PC builder experience. This central hub empowers users to easily access the various functionalities offered by the custom PC build system.

#### 3.3.1 “Landing Page” Interface

This interface enables users to access the main news and deals offered by MegaPc through the mobile app.



**Figure 5-8 Home screen**

### 3.3.2 “SideBar Menu” Interface

The Sidebar menu enables users easily navigate and access categorized products offered by MegaPc.

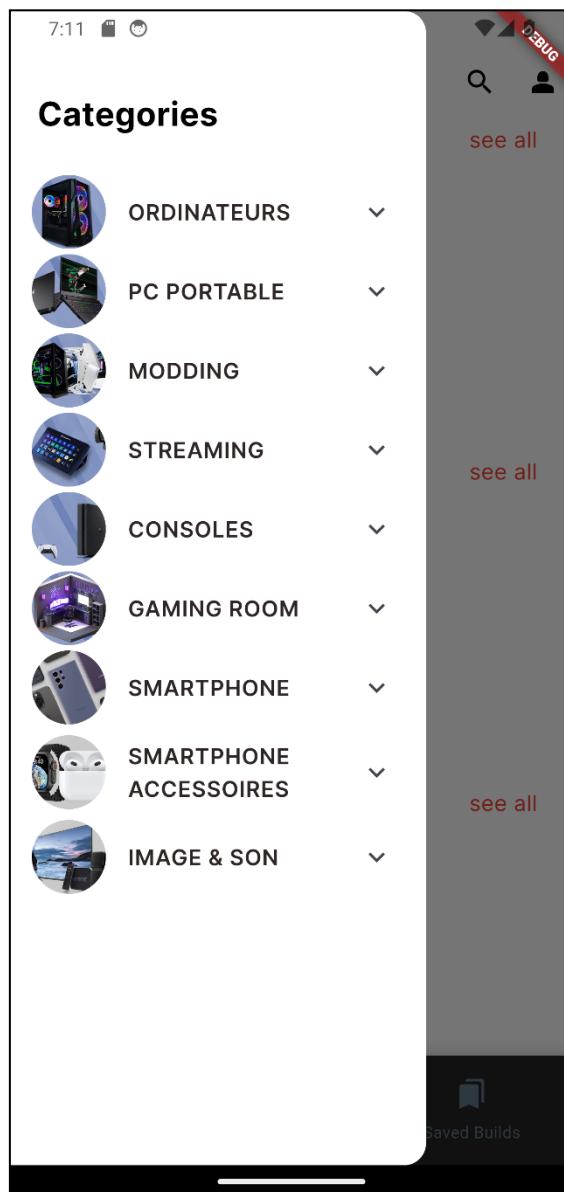
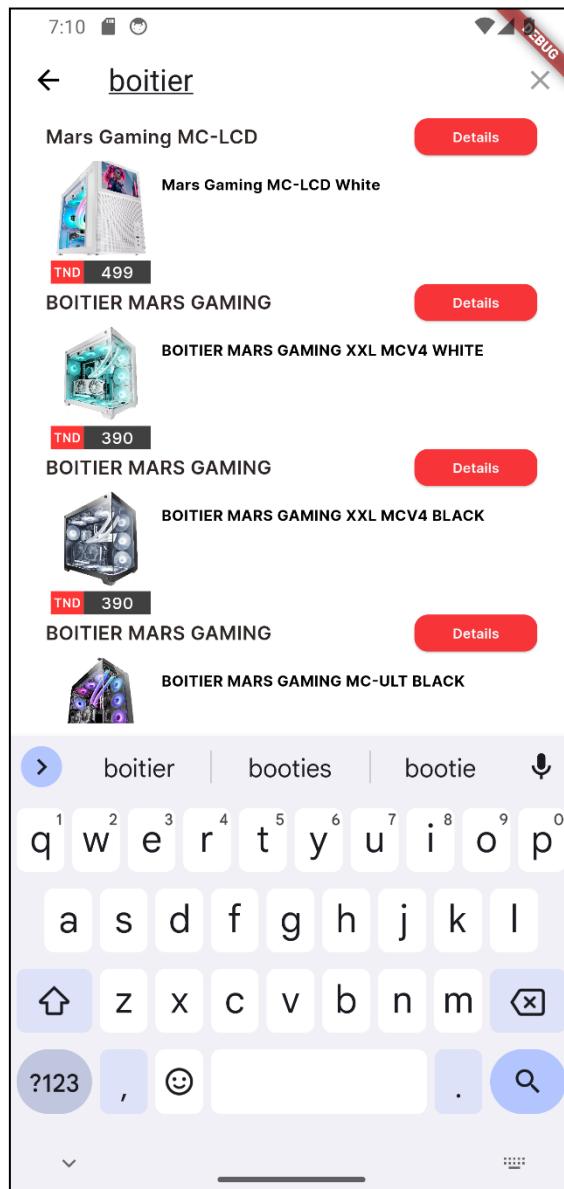


Figure 5-9 SideBar Menu

### 3.3.3 “Search” Interface

The Search interface enables users easily search for products by name offered by MegaPc.



**Figure 5-10 “Search” Interface**

### 3.4 Profile interface

By providing a dedicated profile interface, the custom PC build system empowers users to manage account details.

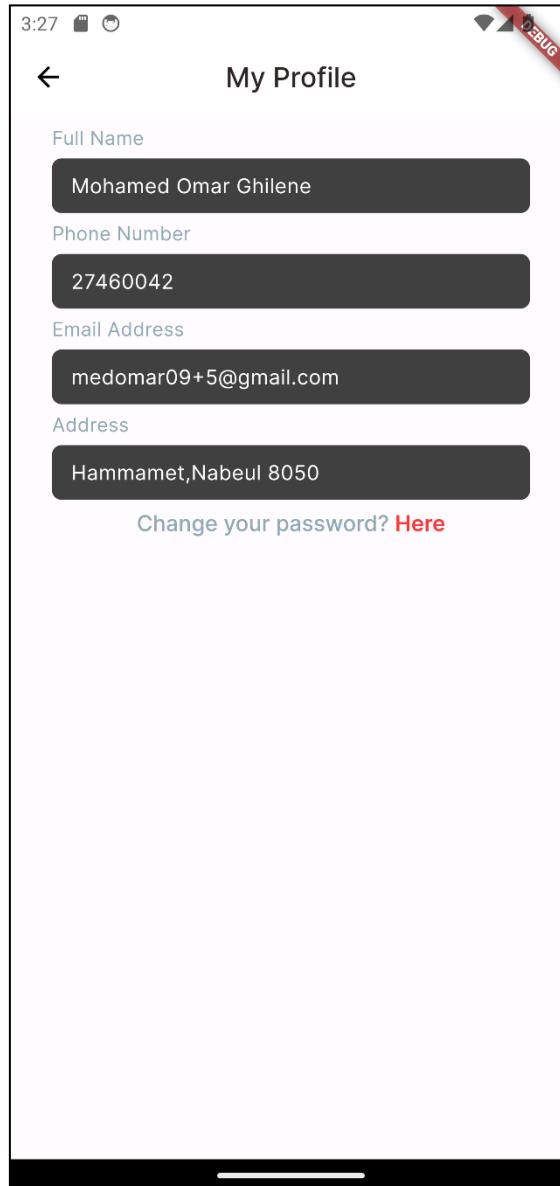


Figure 5-11 Profile Page

### 3.5 Product Details Page

#### 3.5.1 “Notification detail Page” Interface

This interface provides for users all necessary information's about the Notification offered by MegaPc.

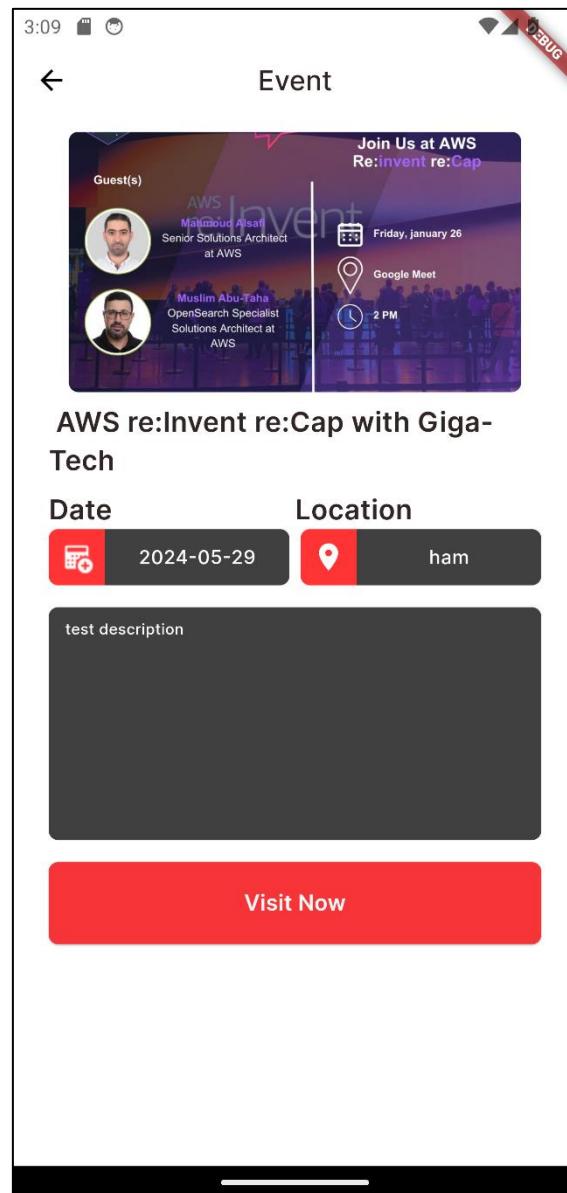
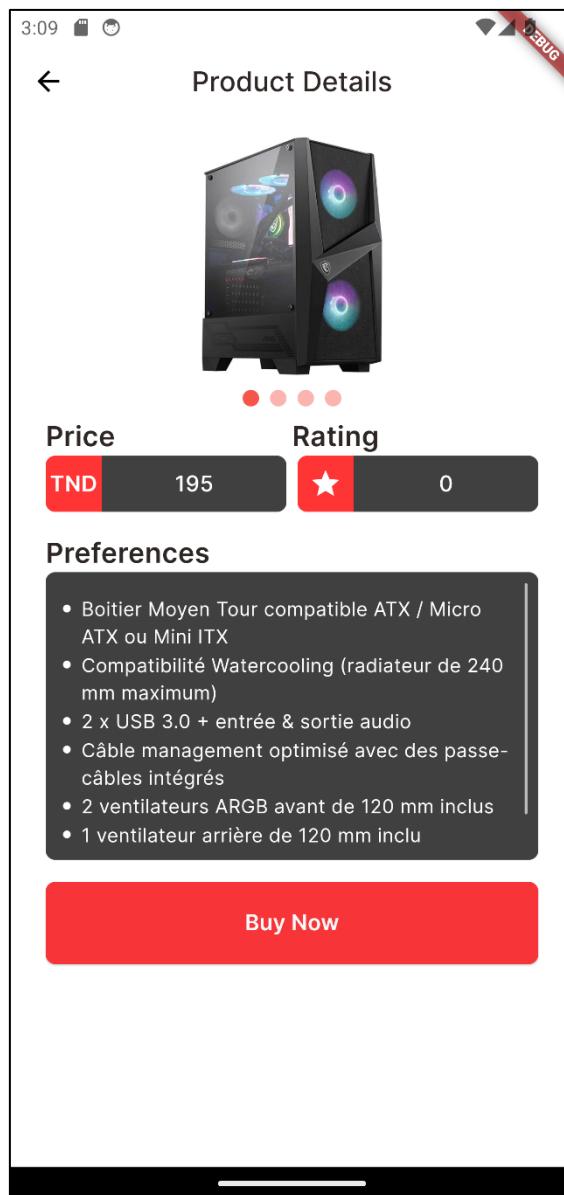


Figure 5-12 Notification detail Page

### 3.5.2 “Component details Page” Interface

This interface provides for users all necessary informations about the Component offered by MegaPc.



**Figure 5-13 Component details Page**

This Interface can be accessed either through searching a product or while displaying a full pc build.

### 3.6 PC Builder Interface

The PC builder system serves a wider range of users, from beginners seeking guidance to experienced users who desire full control over the building process by offering both AI Builder and Manual Builder interfaces.

#### 3.6.1 “AI Builder” Interface

Here, users can specify their preferences, budget, or specific tasks (e.g., gaming, video editing). The AI Builder then analyzes this information and recommends a fitted PC configuration.

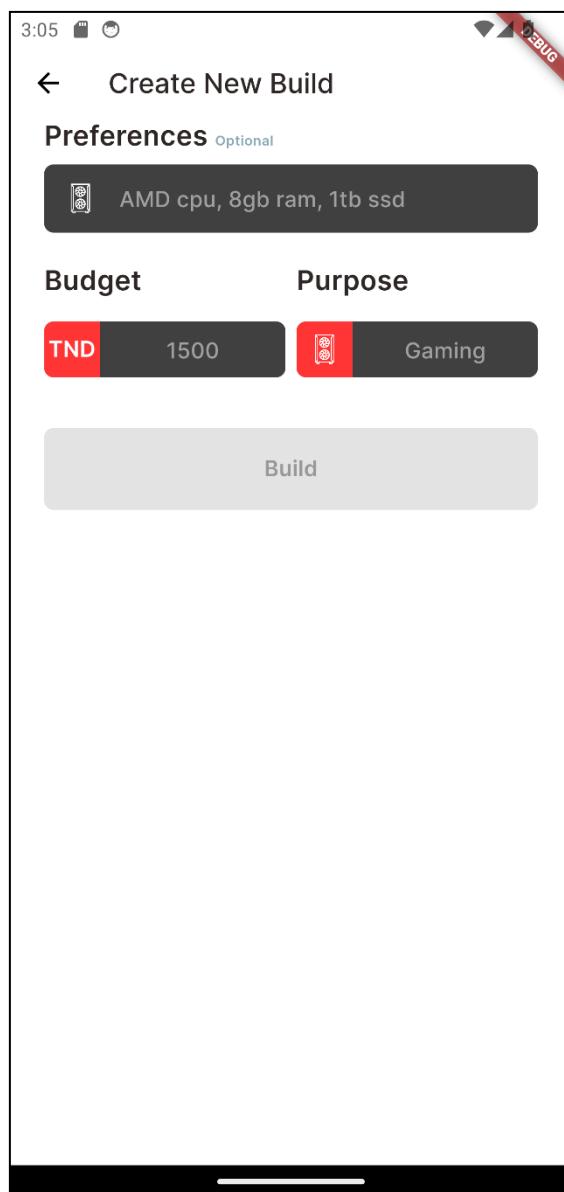
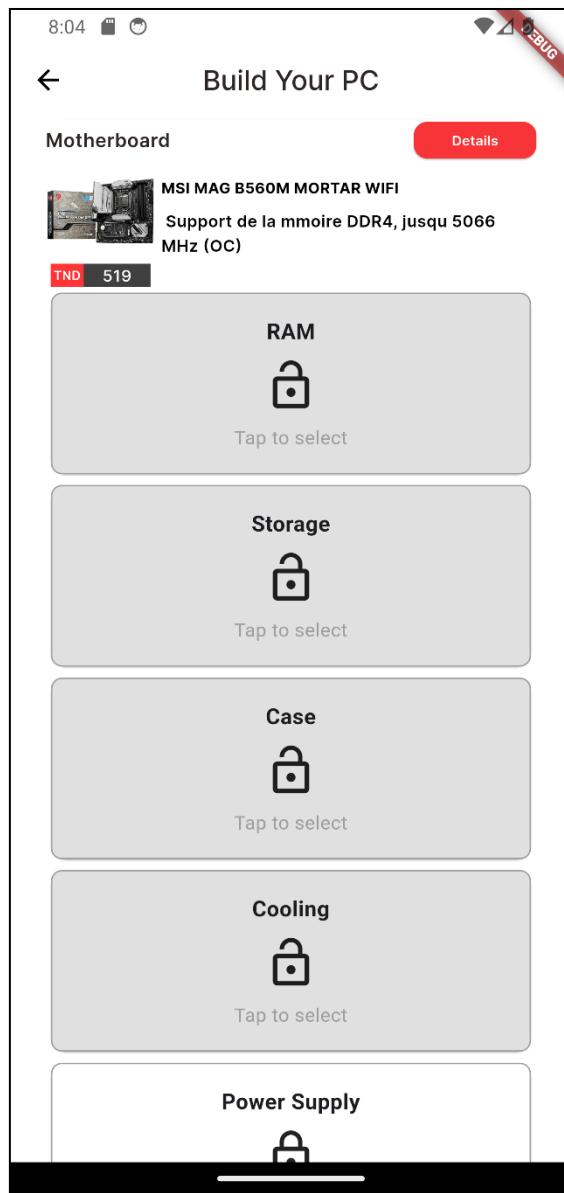


Figure 5-14 AI Builder Page

### 3.6.2 “Manual Builder” Interface

The Manual Builder interface serves experienced users who desire granular control over their custom PC configuration. This interface provides a comprehensive view of all available components categorized by type (processor, motherboard, graphics card, etc.).



**Figure 5-15 Manual Builder Interface**

Users can browse these categories, examine detailed specifications of each component, and select the ones that perfectly match their requirements. The Manual Builder interface enables users to accurately craft their PC, ensuring compatibility between all chosen components.

## 3.7 Build Interface

### 3.7.1 “Full Build” Interface

The Full Build Interface provides a comprehensive overview of the custom PC configuration, serving as a final checkpoint before potentially making a purchase. This interface combines all the components the user accurately chosen for his PC build.

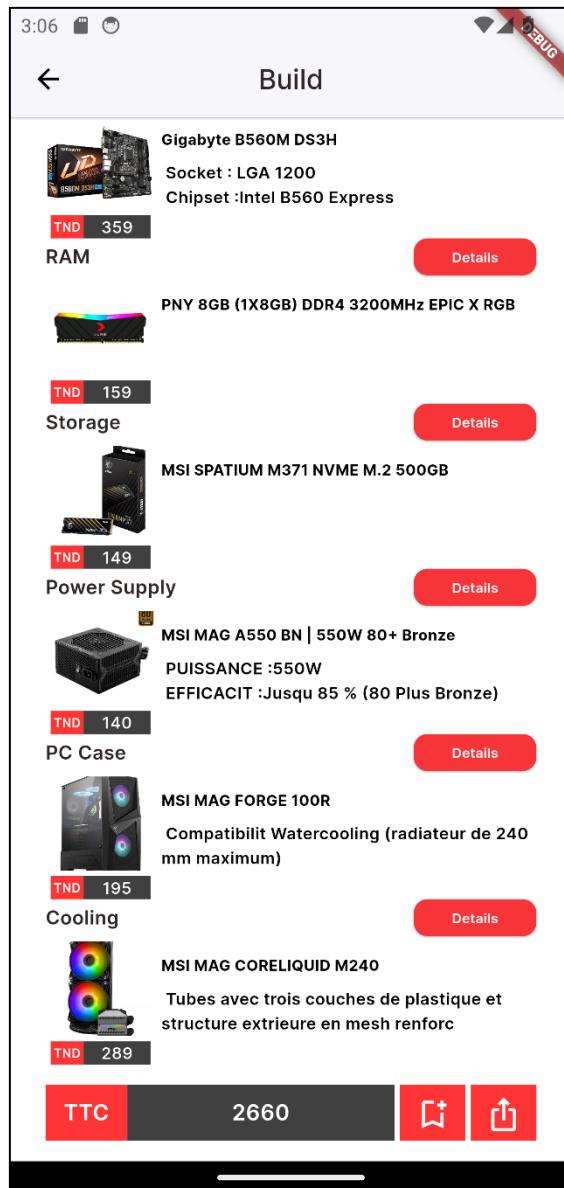
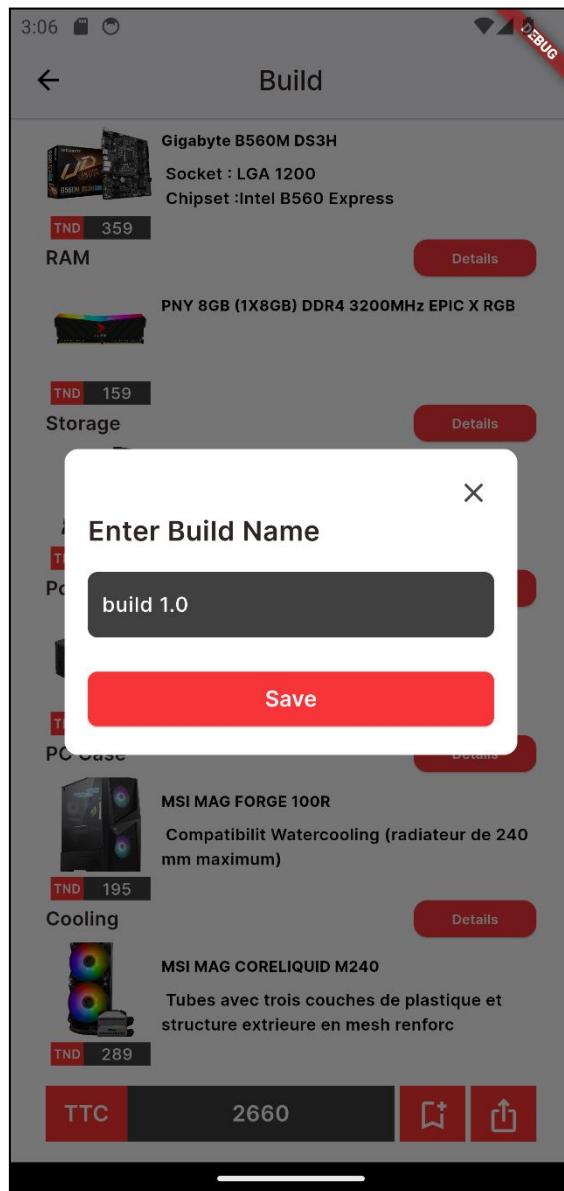


Figure 5-16 Full Build Page

Here, the user can review his full build, this section provides a clear breakdown of the pricing and details for each component. Also, he can either save or share his build.

### 3.7.2 “Saving Build” Interface

The Saving Builds interface allows you to effortlessly preserve your custom PC configurations for later revisiting or potential purchase.



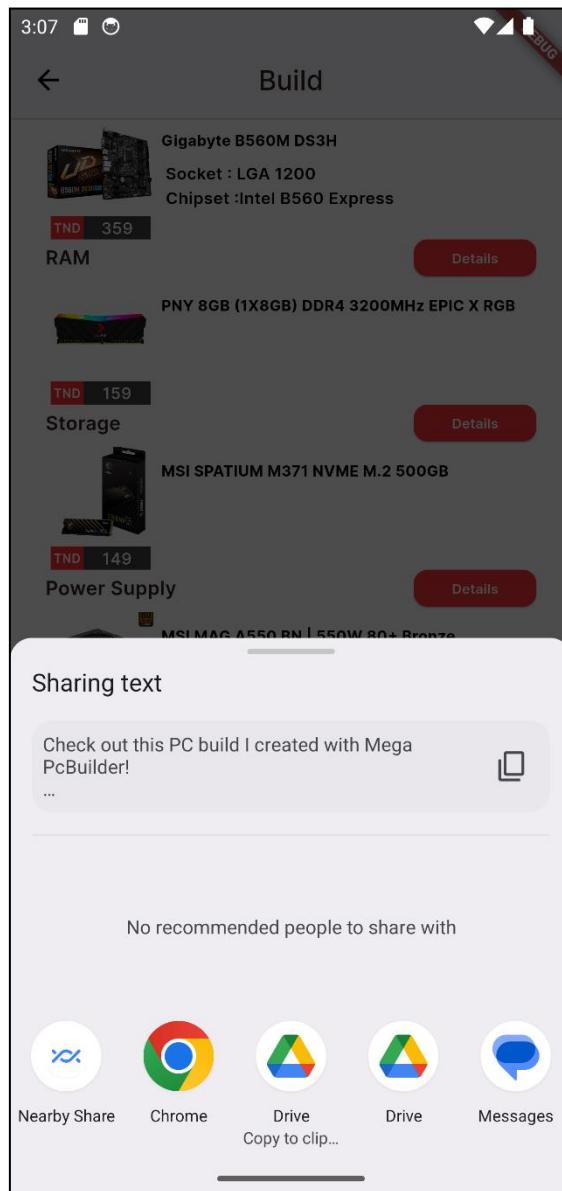
**Figure 5-17 Saving Build Interface**

This functionality is particularly useful for:

- Comparing Builds
- Resuming Progress
- Future order and potential purchases.

### 3.7.3 “sharing Build” Interface

The Sharing Builds functionality allows you to share your custom PC configurations with friends or online communities.



**Figure 5-18 sharing Build Interface**

This can be beneficial for:

- Seeking Feedback
- Inspiring Others
- Social Interaction

### 3.7.4 “Saved Builds” Interface

The Saved Builds List Interface serves as the user's central hub for accessing and managing all his previously saved PC configurations.



Figure 5-19 Saved Builds Interface

## 4. Admin interfaces

This section presents the interfaces related to the admin functionalities. The admin can manage users, products, categories, and notifications through various interfaces.

### 4.1 Login Interface

This interface allows the admin to securely log in to the system by entering their credentials and the generated token code.

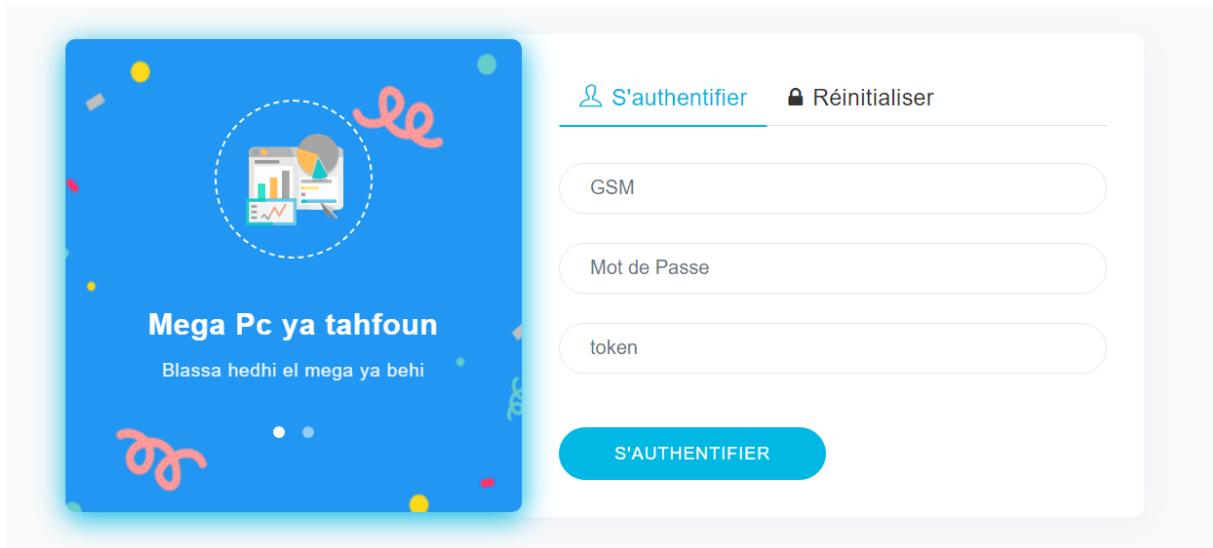


Figure 5-20 Login Interface

## 4.2 Manage Users

The admin has the ability to manage users, including adding new users, deleting and modifying existing ones.

### 4.2.1 Users List Interface

This interface displays a list of all users in the system, allowing the admin to view and manage them.

Manage Users										
<span style="background-color: green; color: white; padding: 2px 10px;">+ Nouveau</span> <span style="background-color: red; color: white; padding: 2px 10px;">Effacer</span>										
<span>Gérer</span> <input type="text" value="Search..."/>										
m ↑	Prénom ↑	Email ↑	Gsm ↑↓	Date De Naissance ↑↓	Adresse ↑↓	Province ↑↓	role ↑↓	Vente privé <input type="checkbox"/>	Date de creation ↑↓	Dernière mise à jour ↑↓
1	sam	sam@gmail.com	(26) 400 099	12/11/2020	test	test	Collaborateur	<span style="color: red;">trash</span>	03/11/2020 11:29 AM	<span style="color: green;">edit</span> <span style="color: orange;">trash</span>
iled	Najib	trapolien@gmail.com	(51) 732 197	10/10/1990	Rades	Rades	Client	<span style="color: red;">trash</span>	18/10/2020 10:13 AM	<span style="color: green;">edit</span> <span style="color: orange;">trash</span>
ntasar	Benyoussef	mohamed.montasar123@gmail.com	(23) 459	06/09/2002	Route saltnia	Sfax	Client	<span style="color: red;">trash</span>	31/10/2020 15:10 PM	<span style="color: green;">edit</span> <span style="color: orange;">trash</span>

Figure 5-21 Users List Interface

#### 4.2.2 Add User

This interface allows the admin to add a new user to the system by entering the required information.

The screenshot shows a modal window titled "Add User". At the top left is a "Role" dropdown menu set to "Client". Below it are five input fields: "Full name", "Password", "Phone number", "Email", and "Locaiotn". At the bottom right are "Cancel" and "Save" buttons.

**Figure 5-22 Add User Interface**

#### 4.2.3 Modify User

This interface enables the admin to modify the details of an existing user.

The screenshot shows a modal window titled "User Details". It includes a "Role" dropdown set to "Collaborateur", a "Code Client Sage" input field containing "Code Client Sage", a "can login?:" toggle switch set to "true", and several other input fields for "Prénom" (sam), "Nom" (sam), "Email" (sam@gmail.com), and "Adresse" (test). At the bottom right are "Cancel" and "Save" buttons.

**Figure 5-23 Modify User Interface**

## 4.3 Manage Products

The admin can manage products by adding new products, deleting and modifying existing ones.

### 4.3.1 Products List Interface

This interface displays a list of all products, allowing the admin to view and manage them.

Manage Products																
		<b>+ Nouveau</b>		<b>Effacer</b>												
		Gérer		Clear		Search...										
		titre	↑↓	Marque	↓	Nbr de Vue	↑↓	price	discount	Stock	Visible	P. spécial	strict QTE	P.privé	Date de creation	Dernière mise à jour
<input type="checkbox"/>	SIMPHONE	INFINIX NOTE	40 PRO+ 5G	12GO 256GO		0	1299		10	⊗	⊗	⊗	⊗	31/05/2024 12:23	31/05/2024 12:23	
<input type="checkbox"/>	Smartphone	Infinix Note	40 Pro 5G			SIMPHONE	INFINIX	0	999	10	⊗	⊗	⊗	31/05/2024 12:03	31/05/2024 12:07	

Figure 5-24 Products List Interface

### 4.3.2 Add Product

This interface allows the admin to add a new product by entering the required details.

Add Product

Name	<input type="text"/>
Title	<input type="text"/>
Rating	<input type="text"/>
Price	<input type="text"/>
Image links	<input type="text"/>
Product link	<input type="text"/>
Description	<input type="text"/>
Categories	<input type="text"/> Select Category

Figure 5-25 Add Product Interface

### 4.3.3 Modify Product

This interface enables the admin to modify the details of an existing product.

The screenshot shows a form titled "Modify Product". It contains the following fields:

- title:** SMARTPHONE INFINIX NOTE 40 PRO+ 5G 12GO 256GO
- Name:** SMARTPHONE INFINIX NOTE 40 PRO+ 5G 12GO 256GO
- Description:** BU-X6851B-N40PRO
- Price:** 200
- ratingValue 5/5:** 2
- Image Links:** (empty)
- Product Link:** (empty)
- Categories:** SmartPhones

**Figure 5-26 Modify Product Interface**

### 4.4 Manage Categories

The admin can manage product categories by adding new categories, deleting and modifying existing ones.

#### 4.4.1 Categories List Interface

This interface displays a list of all categories, allowing the admin to view and manage them.

The screenshot shows a table titled "Manage Categories". The columns are:

- Image 1
- Image 2
- Titre ↑↓
- Description ↑↓
- Order ↑↓
- Visible
- Référence ↑↓
- Date de creation ↑↓
- Dernière mise à jour ↑↓

A single row is visible, showing:

- Image 1: PROMO
- Image 2: (two small images)
- Titre: PROMO
- Description: 0
- Order: 0
- Visible:
- Référence: (empty)
- Date de creation: 27/05/2024 12:27
- Dernière mise à jour: 27/05/2024 12:30

Action buttons include: + Nouveau, Effacer, Cl Browse, UPLOAD FOR 1, Cl Browse, UPLOAD FOR 2, and a green circular icon with a pen and trash can.

**Figure 5-27 Categories List Interface**

#### 4.4.2 Add Category

This interface allows the admin to add a new category by entering the required information.

The screenshot shows the 'Add Category' interface. The 'titre categorie' field contains 'Target Mobile'. The 'target Giga' switch is checked, while 'target Mega' is unchecked. The 'order' field is empty. The 'visible' switch is checked. The 'Référence' and 'Contenu' fields are present. The 'Contenu' field contains a rich text editor with various formatting options like bold, italic, underline, etc. At the bottom, there are 'Cancel' and 'Save' buttons.

**Figure 5-28 Add Category Interface**

#### 4.4.3 Modify Category

This interface enables the admin to modify the details of an existing category.

The screenshot shows the 'Modify Category' interface. The 'titre categorie' field contains 'PROMO'. The 'target Giga' switch is checked, while 'target Mobile' and 'target Mega' are unchecked. The 'order' field has the value '0'. The 'visible' switch is checked. The 'Référence' and 'Contenu' fields are present. The 'Contenu' field contains a rich text editor. At the bottom, there are 'Cancel' and 'Save' buttons.

**Figure 5-29 Modify Category Interface**

## 4.5 Manage Notifications

The admin can manage notifications by adding new notifications, deleting and modifying existing ones.

### 4.5.1 Notifications List Interface

This interface displays a list of all notifications, allowing the admin to view and manage them.

Manage Notifications								
	+ Nouveau	Effacer						
Gérer	<input type="text"/> Search...							
	Image	Titre ↑↓	Type event ↑↓	Valeur% ↑↓	Max Used ↑↓	Expire le ↑↓	Date de creation ↑↓	Dernière mise à jour ↑↓
<input type="checkbox"/>		Big Hot deal F15	hotdeal	0	0	07/05/2024 13:05	05/05/2024 13:05	05/05/2024 17:38
<input type="checkbox"/>		GET20%OFF	discount	0	0	24/05/2024 13:02	05/05/2024 13:02	05/05/2024 17:38

Figure 5-30 Notifications List Interface

### 4.5.2 Add Notification

This interface allows the admin to add a new notification by entering the required details.

**Add Details**

Select a event

titre Event

mini Description

Description FR

Link reservation

X Cancel    ✓ Save

Figure 5-31 Add Notification Interface

### 4.5.3 Modify Notification

This interface enables the admin to modify the details of an existing notification.

The screenshot shows a form titled 'Edit Details' for modifying a notification. It includes fields for selecting an event (dropdown menu), event title ('Big Hote deal F15'), mini description ('ASUS F15 | 15.6" | I7-13620H | 8GB RAM | RTX 4070 | 512GB SSD'), description in French ('Description FR: 6634af6c54c8258b90ad4cab'), and a link reservation ('Link reservation: https://megapc.tn/shop/product/Asus-F15-i7-13620H-8GB-RAM-RTX-4070-512GB-SSD'). At the bottom are 'Cancel' and 'Save' buttons.

**Figure 5-32 Modify Notification Interface**

## Conclusion

This chapter covers implementing the PC build system. We explored the tech stack and development environment used to bring it to life. Following that, we dove into the user interfaces (UIs) guiding users through the app. From login to building their dream PC with AI or manual selection, each UI prioritizes user-friendliness. These elements, along with core functionalities of the system ensures a smooth and engaging PC building journey.

## Overall conclusion

The development of the custom PC build system has been a collaborative and rewarding experience. It provided valuable insights into the practical aspects of software development. This project bridged the gap between theoretical knowledge and real-world application. We designed and implemented complex algorithms while ensuring the system meets user needs and delivers exceptional performance.

Through this project, we gained hands-on experience with various technologies, including programming languages, frameworks, databases, and cloud platforms. We honed our ability to identify and solve complex technical challenges, often requiring creative and innovative approaches. Additionally, we developed effective project management skills, including planning, organizing, executing, and monitoring tasks to ensure the system's timely completion. Our communication skills, both written and verbal, were strengthened through collaboration with team members and stakeholders throughout the project lifecycle.

The custom PC build system serves as a foundation for further enhancements and potential expansion. Future directions include expanding the AI capabilities to provide more personalized and intelligent product recommendations for users. Integrating social features would allow users to share their builds, compare configurations, and engage with the PC enthusiast community. Refining the mobile app for optimal performance, usability, and cross-platform compatibility is another promising avenue. Finally, leveraging cloud services could enhance scalability, data storage, and accessibility of the system.

The development of this custom PC build system not only allowed us to apply theoretical knowledge to a practical project but also instilled in us a passion for software development and a drive to continuously learn and improve our skillset. We are confident that the experience gained from this project will be invaluable as we pursue our careers in the ever-evolving field of technology.

## Bibliography

- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal†, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela; Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, 12 Apr 2021, Facebook AI Research; University College London; New York University;

# Webography

[web1]: <https://megapc.tn/>, Feb 05, 2024.

[web2]: <https://www.onlinegantt.com/#/gantt>, Feb 14, 2024.

[web3]: <https://www.newegg.com/tools/custom-pc-builder>, Feb 21, 2024.

[web4]: <https://pcbuilderai.com/>, Feb 23, 2024.

[web5]: <https://platform.openai.com/docs/api-reference/introduction>, March 3, 2024.

[web6]: <https://js.langchain.com/v0.2/docs/introduction/>, March 21, 2024.

[web7]: <https://github.com/>, March 21, 2024.

[web8]: <https://docs.flutter.dev/>, March 24, 2024.

[web9]: <https://miro.com/app/board/uXjVKBxZZgs=/>, March 28, 2024.

[web10]: [https://www.figma.com/design/77r3b0XX73b04vrHvCUav8/PcBuilder-Design-\(Stage-PFE\)?node-id=0-1&t=UInoOVcCFzsX0mN1-0](https://www.figma.com/design/77r3b0XX73b04vrHvCUav8/PcBuilder-Design-(Stage-PFE)?node-id=0-1&t=UInoOVcCFzsX0mN1-0), April 1, 2024.

[web11]: <https://en.wikipedia.org/wiki>, April 10, 2024.

## List of abbreviations

- **UML:** Unified Modeling Language.
- **DB:** Data Base.
- **MVC:** Model View Controller.
- **XML:** Extensible Markup Language.
- **AI:** Artificial Intelligence.
- **RAG:** Retrieval Augmented Generation.
- **LLM:** Large Language Modal.
- **SOTA:** State of the art.
- **UX:** User Experience.
- **UI:** User Interface.