ANALYSIS, DESIGN, AND IMPLEMENTATION OF AN E-COMMERCE SOFTWARE

FABIOLA VEDRA

EPOKA UNIVERSITY

FACULTY OF ECONOMICS AND ADMINISTRATIVE SCIENCES

DEPARTMENT OF BUSINESS ADMINISTRATION

BUSINESS INFORMATICS

TIRANA, ALBANIA

May, 2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATION

SEO	Search Engine Optimization
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
SQL	Structured Query Language
ACID	Atomicity, Consistency, Isolation, Durability
ORM	Object-Relational Mapping
SSL/TLS	Secure Sockets Layer/Transport Layer Security
JSON	JavaScript Object Notation
UML	Unified Modeling Language
ERD	Entity Relationship Diagram
RBAC	Role-Based Authorization Control
DFD	Dataflow Diagram
CPU	Central Processing Unit
API	Application Programming Interface
RDBMS	Relational Database Management System

CHAPTER 1:
INTRODUCTION

Technology has become a key component of the e-commerce industry, providing new opportunities for development and innovation. (Advent Biztech Solution Pvt Ltd, 2023) This is because with the growth of technology, the possibility of people having easier access to what they want increases. In our case, it is about creating the possibility to buy skin care products more easily, not necessarily by showing up at the physical address. It is enough to know what you need, and where to order it, and with just one click, you will bring what you want to your home in a few hours. But like everything, this also has its negative sides, and this is because today people are not yet fully familiar with the idea of buying on an online platform and constantly seek help from different operators. For this reason, they mostly turn to online businesses, which are not always transparent with customers and give them not what they need, but what gives them more profit. Many reviews studied have shown the disappointment of customers towards online businesses, regarding the service and products received and this is for the only reason above, profit-making and not customer satisfaction as an objective.

A survey conducted in November 2019 by a student of Savonia University about people's experiences with misleading advertising in this category of the market, to which 205 people responded, set the conclusion that the majority of the respondents had purchased products that did not give the results promised in the advertisements. This led to their dissatisfaction with the product and the company, causing hesitation to continue as a customer. (Heli, 2019)

This was one of the main reasons that led us to create SkinBliss, a commerce platform very easy to use by any customer group and offers the opportunity to buy a variety of skincare products of different brands, prices, and functions.

The special feature of SkinBliss lies in the fact that it is one of the only E-commerce platforms that will not offer interaction with online operators for product suggestions, and this is for the sole reason of minimizing misunderstandings between the customer and the business that subsequently results in his dissatisfaction. SkinBliss will work most efficiently and will offer the customer everything they need to understand their skin type and the products they need.

It doesn't matter where you are or your knowledge of the skincare field because the application will offer you the possibility of a skin test where you can find out your skin type and based on the situation you have, it will automatically generate the products needed for a complete routine.

1.1 Objectives

SkinBliss's website application is a reflection of our mission to redefine the skincare shopping experience by providing transparency, customization, and reliability to the customer. The main objectives of this web application would include:

Quick Solution Access: Developing a reliable website where people can easily and quickly find solutions to their skin problems, without external influences having information available for every possible scenario.

User-friendly software: enhance the skincare shopping experience by offering a user-friendly interface, easily and usable by all customers as well as through an optimal organization where product discovery will be very easy and efficient.

Customization of routines: One of the distinguishing characteristics of SkinBliss is the personalization of routine products that will be based on references and personal problems. The client would enter the characteristics and problems of the skin, while the application will be based on these inputs and will automatically generate the necessary skin products.

This part will be very important for our web application because it increases customer satisfaction and loyalty.

Transparency and trust: This objective is going to be achieved through the part of not being influenced by external factors in the purchase process. We aim to increase trust in customers not only by using the personalization of products but also by creating a review section as transparent as possible, where customers will have the opportunity to express their opinions about their experience with the products received.

User privacy and security: Today, in the era where online platforms dominate, one of the most common fears of people is the theft of personal data, both by hackers and businesses. We have tried to make our web application as safe as possible by implementing the authentication of the actions of each party that will use the web application, encrypting passwords to make them as secure as possible, and removing the option of card payments, and instead the addition of payments on product delivery, except in special cases.

A diverse range of products: The rule of the market is that to satisfy as many customers as possible, a business should offer as wide a range of products or services as possible. SkinBlis was created with a high diversity of products of different prices, functions, and brands to be able to serve all types of skin or problems that can be encountered.

CHAPTER 2:
LITERATURE REVIEW

Beauty sales are driven mainly by eCommerce with the intervention of technology, as new beauty businesses find it easier to form and evolve through online or social purchasing platforms. The beauty market consists of products related to makeup, cosmetics, skincare haircare, etc. There is no age, gender, or cultural barrier to any of this, which leads to the significant evolution of this industry. According to Common Thread Co., the personal care market grew almost 6% from $483 billion to $511 billion in a year and it is estimated that will exceed $716B by 2025, an indicator that beauty eCommerce will become more competitive in the upcoming years. (S., 2023)

The primary advantage of e-commerce is the convenience it offers. Customers can search for what they desire from a wide range of products and purchase what they are looking for from their actual location without needing to go to the physical location of the store. This saves them time, money, and effort making online purchasing more desirable than traditional shopping. However, everything comes with its challenges, and the biggest concern related to online shopping is the lack of physical interaction with the product before buying it. Customers cannot physically touch, feel, or see the product, an issue that most of the time leads to dissatisfaction because the purchase does not meet people's expectations. (AppCurators, 2023)

This issue gets more challenging when it comes to face products. This is the main reason why we thought of bringing the innovation of online product recommendations to skincare retailing in Albania. Personalized recommendations play a crucial role in boosting the importance and effectiveness of skincare shopping experiences. Based on studies made, 82% of consumers are interested in having a skincare routine created specifically for them; 78%

are interested in a device that would let them formulate skin care in their own home; 76% are interested in an enhanced reality/smart mirror (in-home or at a store); and 72% are interested in an app or camera that would let them track the health of their skin. (Herich, 2020)

But this is not enough. Technology is rapidly developing but at the same time is still something new and a lot of people find it difficult to use technology to purchase goods. To overcome this disadvantage, we will focus on making our software as user-friendly as possible.

User-friendly web design has a variety of benefits. Firstly, web apps that provide a good user experience have higher reachability by customers than poorly designed ones, because quick load times result in better Search Engine Optimization rankings, making the web app ranked in first places and at the same time easier for people to find it.

Once they've landed on our site, we can foster trust in them through the software' organized design, features, performance, content, and security. (Panfili, 2020) SkinBliss is designed in such a way to make it easy for visitors to find the products they are looking for navigate through the pages, and interact with the website's features easily.

User experience is a crucial part of every website as it has a high influence on the success of the web application and also on the business that it belongs to. A web application with a focus on the user experience has a high potential to create a positive reputation which necessarily will attract new and repeat visitors. A positive user experience can boost the probability of visitors, either making a purchase or taking other actions toward the website. Conclusively, a better user experience guarantees a win for both the visitor and the business. (Mehta, 2023)

In conclusion, while the movement of the beauty industry to e-commerce offers significant growth prospects, it offers also challenges that must be carefully considered. SkinBliss tends to overcome these challenges by designing software that will deliver value to customers through innovative features like a well-designed skin diagnosis tool product personalization software, and most importantly a user-friendly website. In this way, we aim to provide the required service to consumers seeking personalized solutions for their skincare needs. The thesis will explore the development of such a website, considering the findings from this literature review.

CHAPTER 3:
ANALYSIS AND DESIGN

3.1 Product Description

SkinBliss is designed to make the process of purchasing skin care goods online easier in both cases: when the client knows what product he wants and when he needs help determining what product he needs.

SkinBliss is built on a large database of common skin problems and skin types, which is closely linked to the process of developing the essential goods for each condition. SkinBliss consists of a user-friendly interface that includes easy navigation of its features and the selection and completion of orders. A user of SkinBliss can do the virtual skin test, see the categories of products he is interested in, add products to his card, place orders, or check them through a few quick clicks. SkinBliss was created to take a big step in the online shopping of the skincare industry, to increase customer confidence and improve the skincare service in our country.

3.2 Product Perspective

SkinBliss will be created to be an all-inclusive platform both for those who are satisfied with their skin and for skincare enthusiasts who are looking for not only quality products but also a quality and transparent service for unique skin types and problems. It will be created with the idea of offering innovative features that are rarely found in our country, such as virtual skin tests and personalized routines for the most transparent interaction and the highest customer satisfaction. SkinBliss is based on a large database equipped not only with a large number of products of different brands, functions, and prices but also based on all types of skin or problems that can be encountered by anyone who uses the website.

3.3 Product Function

SkinBliss will have two main roles, the one of the user and the one of the admin.

The Administrator can view and manage the accounts created on this web application, and if needed delete them. The same goes for the orders too, for example, if a customer does not want his/her order, the administrator can cancel it. Furthermore, the Administrator can create, view, update, or delete products from the page, and update their prices and information as needed. The Administrator will have higher permissions than the user, but some permissions are performed by the user only, and this works for the administrator part too. This will ensure the welfare of the website and will maintain its transparency.

The Users on the other hand as we mentioned previously will have more limited access to the website. Apart from the registration part (which will be required if someone wants to make an order), the Users can edit their account information by adding new information or deleting the previous one. Furthermore, users can view the listed products and add them to their cart, manage the products on the cart, and set orders. The previous orders can only be viewed by the users and not by the administrator to maintain their privacy. Users can leave reviews on the products they have already bought, not only to help the administrators improve the software but also to help the new customers in their choices.

3.4 Software Development Model

For the development of the SkinBliss web application, we decided to use the Agile development model due to its flexibility and adaptability to changes in project requirements. The Agile method aims to break a project into phases and continuously improve the project through continuous interaction with the customer. (ATLASSIAN, n.d.)

Since the application we have decided to develop has a very dynamic nature due to the continuous changes that may be required over time, the Agile development model would be the most suitable to answer these requirements, because it enables us to engage in repetitive development cycles and receive ongoing feedback from the customers regarding the continuation of our work, and the ability to respond to the changes effectively. Furthermore, since one of our main goals is to make the application user-friendly, it will need to be constantly modified while it is being developed to produce the desired results and fulfill our earlier point. So to conclude, one of the biggest advantages of the Agile Method is the constant cooperation with the client, which provides us the ability to deliver a product that will meet the requirements of the client, minimize the costs of development, and most importantly ensure customer satisfaction.

3.5 Software Requirements

Skinbliss application frontend will be assembled through HTML, CSS, JavaScript, and React.js library providing a dynamic and interactive user interface. The backend, on the other hand, will be built using Node.js and Express.js to handle the server-side logic and REST API endpoints, ensuring efficient data processing between the client and server. The database for the data management will be created using MySQL, which is utilized to store and manage product information, user data, and order details, complying with ACID requirements. We will be using Prisma ORM for composing the programming code with database structures. (Awati, n.d.)

For the security of the web application will be used, SSL/TLS protocols for encrypting data transmitted between the client and server, and JSON Web Tokens for authentication between the two parties. For the Code Editor, the Visual Studio Code. For the creation of the diagrams, flowcharts, and UML diagrams will be used DrawIO.

3.5.1 Functional Requirements

User Registration and Authentication: Users will have the ability to register to their account by using their credentials such as name, email, and password and securely log in or log out of the account.

Product Browsing and filtering: Users should be able to browse the product they need by looking at the relevant category such as price range, purpose, and skin condition, or by using the available search bar.

Shopping Cart Management: Users will have the ability to add new products to their cart, view cart content, and update the product quantities.

User Product Feedback: Users should be able to write feedback for the products they have already purchased, which cannot be managed by the administrator to ensure the website's transparency.

Admin Dashboard: Administrators will have access to a dashboard to manage user accounts (deactivating accounts), and user orders by approving or canceling them.

Product Management: Administrators should be able to see the products available, delete them, update their prices, pictures, and descriptions, and add new products to the web application.

3.5.2 Non-Functional Requirements

Performance: All pages and features of the software must have a fast response time and should be able to handle high numbers of users without downgrading performance. Furthermore, the software must not be intersected by ads or cookies.

Usability: The user interface should be uncomplicated and easy to navigate (consistent layout and clear menus).

Scalability: The application should be built to assist in increasing user traffic, product catalog expansion, and activity volume.

Security: The system must guarantee the privacy of customers' personal information and security. The user passwords will be stored by using strong encryption techniques and authentication procedures.

Compatibility: The software must function consistently with all kinds of browsers (Chrome, Firefox, Safari, Edge) so client usage won't be restricted by their browser.

3.5.3 Organizational Requirements

3.5.3.1 Environment Requirements

Internet Connection Quality: The software should optimize the loading times so it can help the users with both high-speed and low-speed connections, giving a seamless browsing experience for all users.

Browser Compatibility: The software should function consistently with a wide range of web browsers and their different versions and ensure its functionality through all different platforms.

Device and screen size variations: The software should be responsive on all kinds of devices, and screen sizes (laptop, computer, smartphone, tablet) providing a smooth experience, regardless of the gadget.

Accessibility standards: The software must be based on accessibility standards to ensure easy usage for disabled users by applying elements like screen reader compatibility and keyboard navigation.

Energy Consumption Concerns: The software must minimize as much as possible the energy consumption in its design and development process via code optimization, server request reduction, and the usage of energy-efficient solutions to narrow environmental impact.

3.5.3.2 Operational Requirements

Website Maintenance: The software must be regularly updated and maintained to remain high-performance, and be functional and secure. The maintenance includes fixing bugs, adding new required features, applying security patches, etc.

Order Processing: Creating an efficient order processing management system, which will handle the incoming orders, verify the information related to each order and manage the inventory after order confirmation or cancelation. Providing accurate order information about the customers during this process.

Security Measures: Creating strong security measures including firewalls, encryption, and systems that detect unauthorized access to protect the data on the website from cyber threats or unauthorized access.

Compliance: Making sure that the software complies with the relevant laws, regulations, industry standards, and data protection. Updating these regularly to make sure the software's procedures and policies remain compliant with any changes made.

3.5.3.3 Development Requirements

Understanding the Requirements: Discussing with the customers to understand what they are expecting from the software, and to have a better image of what they are looking for. During this phase, we get information related to the customers' requests.

Product Design: Making a design/prototype of the product in its final phase to present to the clients.

Product Implementation: Creating the ERD diagrams and database of the system, and then starting the development of frontend and backend through the abovementioned technologies such as HTML, CSS, and JavaScript for frontend and Node.js, Express, js, Prisma ORM, etc. for backend.

User Interface Development: Developing a user-friendly interface that enhances the shopping experience for the users through easy navigation on all its pages and other features.

Testing Phase: Testing the product and checking for errors that need to be fixed to prevent any dissatisfaction from the client, and also to deliver a functional and efficient product.

3.5.4 External Requirements

3.5.4.1 Ethical Requirements

Ethical coding means ensuring that code instructions do not lead to harm, injustice, or inequity. It requires adhering to a set of principles that guide responsible behavior when developing and using the software. The principles mentioned concern respect for privacy, honesty, fairness, and a commitment to not harm. (Addo, 2023) Some ethical requirements include:

User Consent and Transparency: Users must be aware of what and how their information is being used and have the right to accept or deny the usage or collection of their personal information. Furthermore, we should be transparent by providing a privacy policy accessible to the user that defines what data are we collecting, why are we gathering that data, and how will we use it. (Addo, 2023)

Data Minimization: We must understand the importance of the data collected and how much of the data we need. Ethical requirements stand for minimizing as much as possible the data collected, collecting only the data needed for the functionality of the software, and ensuring simplicity.

Data Security: The data's security is one of our biggest priorities, and is going to be achieved via implementing strong measures to secure the user's data. We must regularly update these security measures and improve them where needed.

Accessibility and Inclusivity: The software must be accessible to everyone, including people with disabilities. For this, we must implement features like screen readers, keyboard navigation, and high-contrast modes to make SkinBliss comprehensive. (Addo, 2023)

Age Restriction: The website won't allow services to anyone under the age of 18 years old, nor will require information from this group of people to ensure their data protection.

3.5.4.2 Legislative Requirements

Legislative Requirement means a legal requirement placed under any existing or future legislation, including laws, ordinances, regulations, licenses, permits, authorizations, codes, or any other legally mandatory requirement of a government. (Legislative Requirements, 2023)

Based on Article 35 of the Constitution of Albania (only available in Albania) it is a fundamental right to protect personal information.

1. No one may be obliged, except when the law requires it, to make public the data connected with his person.

2. The collection, use, and making of public data about a person is done with his consent,

except for the cases provided by law. (Albanian Constitution, 2016)

Based on these laws, our software is obligated to ensure the safety of the private information gathered by our clients in the registration process. Thus, the information of our clients is accessed by them and only them. Through our security measures mentioned above, one client's sensitive data won't be visible to another client, or the admin.

Additionally, before registering the user must consent to the Terms of Condition of the software where is stated that he identifies himself as a customer, accepts the use of his information for the following processes, and the affirmation that the information given is true and accurate.

3.5.5 Security Requirements

This platform requires its users to create an account to benefit from its features. The safety of the data used by the users will be our priority and its safety is going to be achieved through some processes. All of the web applications' information will be stored securely in a database, to prevent its loss.

Authentication methods verify the identity of customers, protect their sensitive information, and are responsible for securing online transactions. In this way, they can prevent fraud and increase customer trust. (Barney, 2023) Authentication will be used in the login section to verify if a user is whom he/she claims to be.

Furthermore, as we have mentioned previously the actions and data of the parties included in this software will be limited, via role-based authorization control. In this way, the admin won't have access to the users' private information and vice versa.

3.5.6 Safety Requirements

Encryption stands as the basis of our software's data safety, ensuring protection for the user's data. The users' passwords are encrypted and stored as a hash in the database. Also, the passwords will be compared as hashes and not as decrypted strings. In this way, we will be assuring security for all the data stored in the system. The encryption will be conducted through SSL/TLS protocols. SSL/TLS protocols encrypt data between servers, applications, users, and systems, to exchange the data securely. (AWS, n.d.)

3.6 UML Diagrams

A UML diagram is used to visualize systems and software using Unified Modeling Language. UML diagrams audit the relationships and hierarchies between important pieces of code. (MIRO, n.d.) UML diagram creation is part of the system modeling phase of the development of software, which except for the importance of the documentation part plays a key role in ensuring that the software doesn't have any bugs that prevent it from properly working. The diagrams I have decided to use for the design process include Data Flow Diagrams, Component Diagrams, Class Diagrams, Sequence Diagrams, Database Diagrams, Case Diagrams, Activity Diagrams, and State diagrams.

3.5.1 Data Flow Diagram

A data flow diagram maps out the flow of information for any process or system. It presents data inputs, outputs, storage points, and the routes between each destination, through the use of different symbols including rectangles, circles, and arrows. (LucidChart, 2024) Data flow diagrams usually have no control flow and do not follow a specified loop. They can range from simple to multilevel. We have generated a two-level DFD.

3.5.1.1 DFD 0 Level

The 0 Level is the highest-level DFD, which provides an overview of the entire system. It generally shows the most important processes, data flows, and data stores in the system. (GeeksforGeeks, 2023)

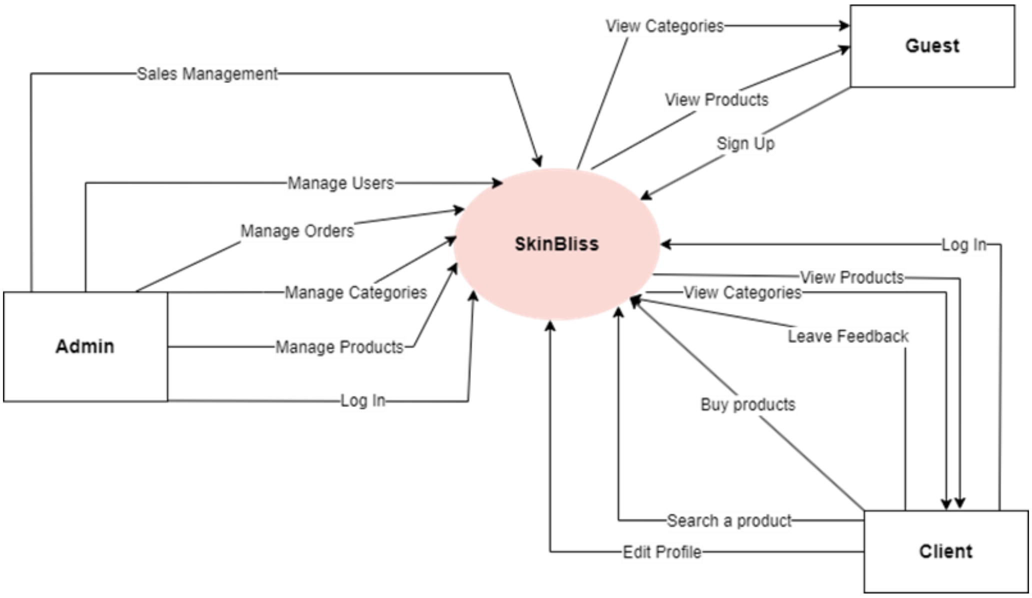


Figure 1 DFD 0 Level

3.5.1.2 DFD 1 Level

It provides a detailed view of the system by breaking down the processes identified in the level 0 DFD into sub-processes. (GeeksforGeeks, 2023) This diagram shows in detail what processes perform all the parties of the software, and helps us better understand the system.

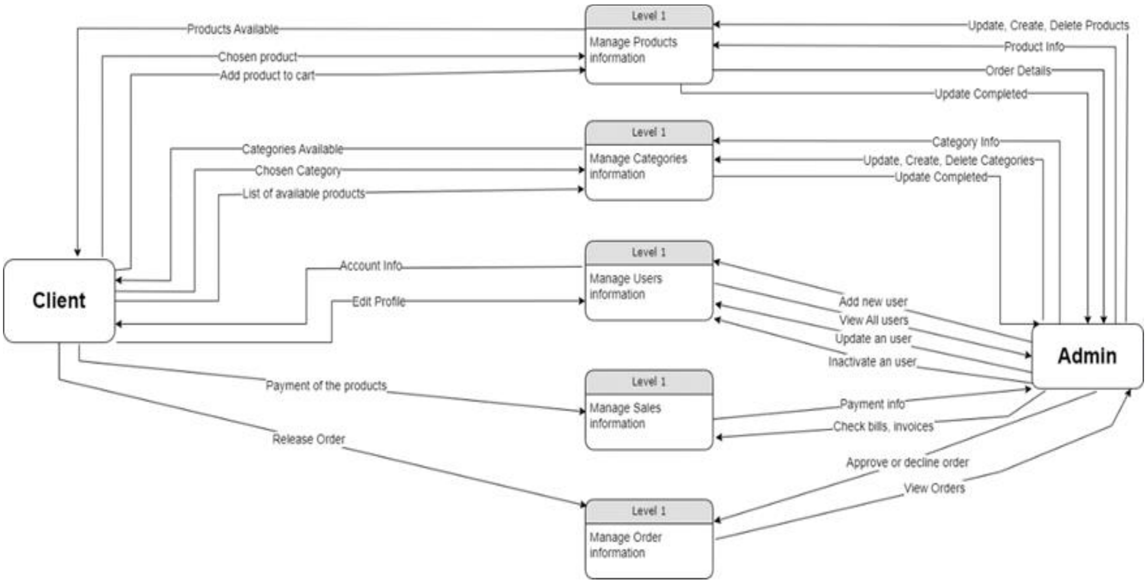


Figure 2 DFD 1 Level

3.5.2 Component Diagrams

The component diagram presents how small components of the software are related together to form the entire software. The diagrams provided show the components during the signup, login, and overall admin management processes.

As presented the signup diagram includes the entry of the user information in the system to create the user account by which one can access the software's services. The admin diagram represents how the admin can manage the accounts, orders, products, and their category to ensure the functionality of the system. The component diagram for order management provides us with e visual view of all of the components participating in this process, starting from the selection of the product from the customer to its checking in the inventory and the placing of the order.

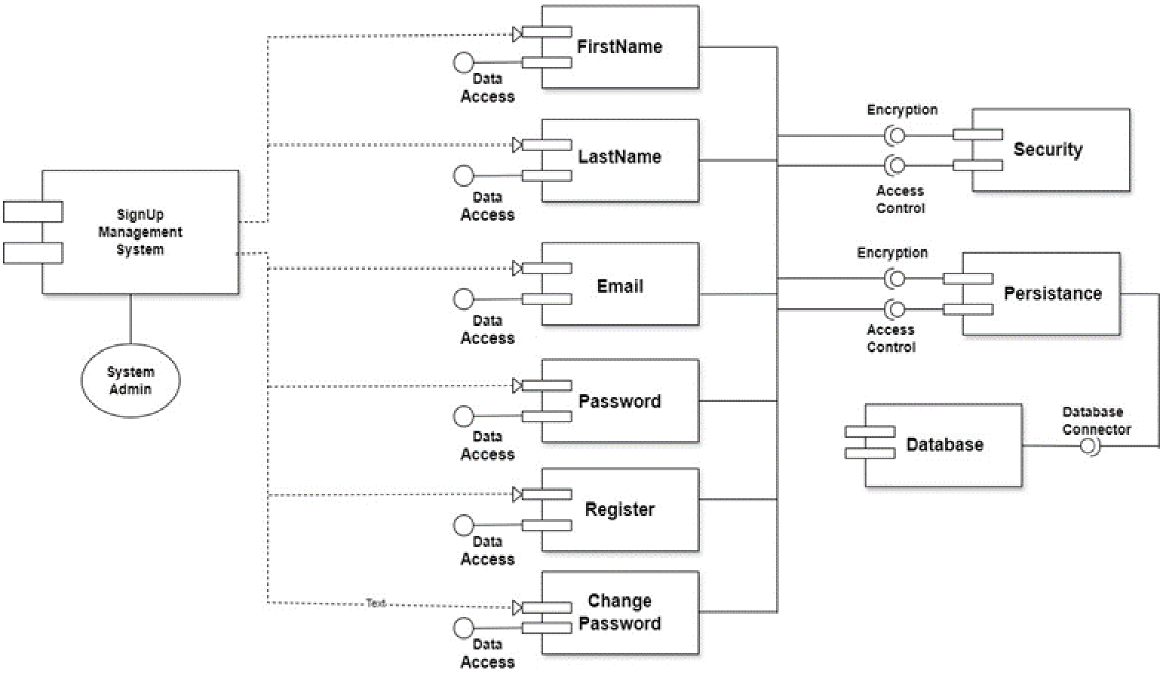


Figure 3 SignUp Component Diagram

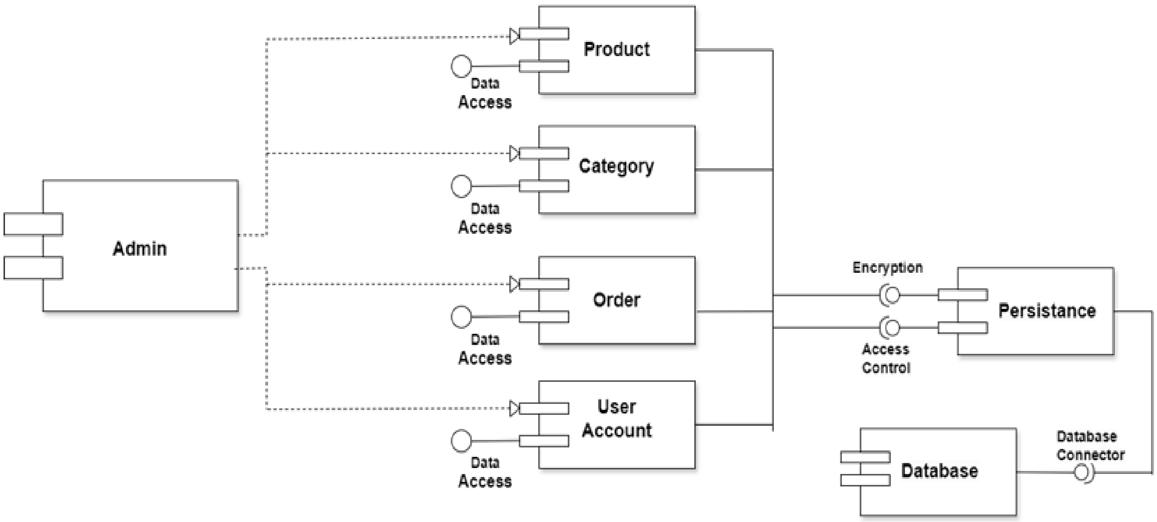


Figure 4 Admin Actions Component Diagram

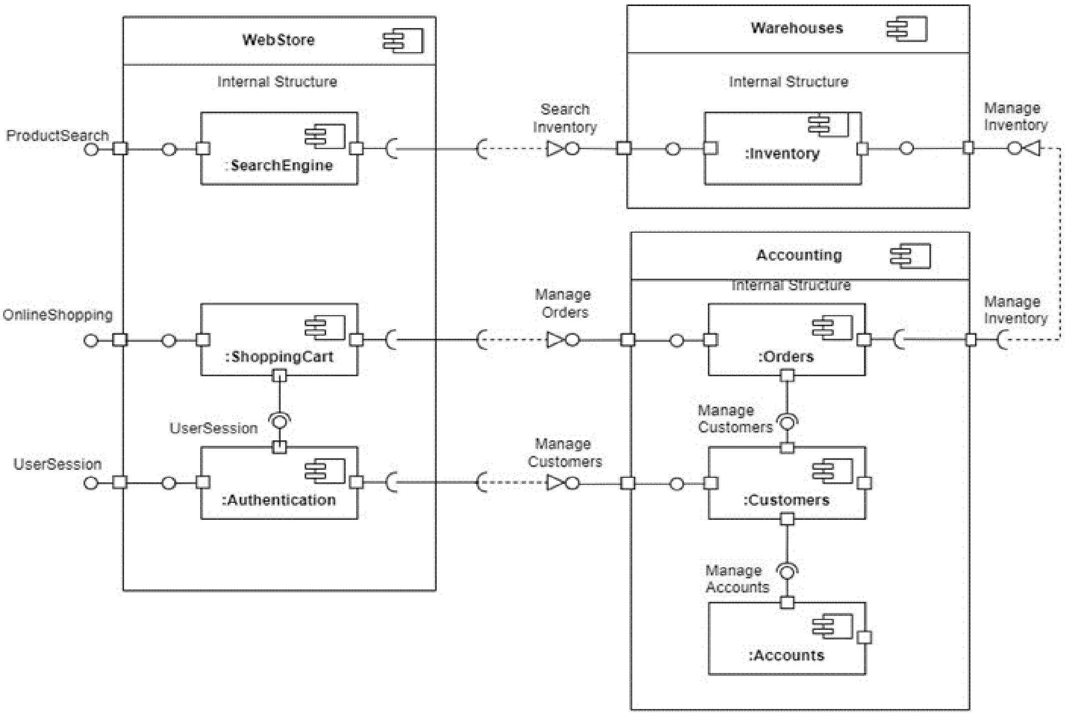


Figure 5 Order Placement Component Diagram

3.5.3 Class Diagram

Class Diagrams are detailed diagrams used when developing an object-oriented system and present all the classes used in the development of our software and the connections and interactions between these classes. This diagram presents all of the functions a specific class can perform, the other entities that it is related and what attributes contains.

This class diagram shows the relationships between the main classes of SkinBliss including User, Client, Company, Product, Role, Admin, Category, Brand, and Order. The User class is the most general in the diagram. It has two child classes, Client and Admin. The Admin class has several functions that manage the other classes in the diagram such as products, categories, brands, and users.

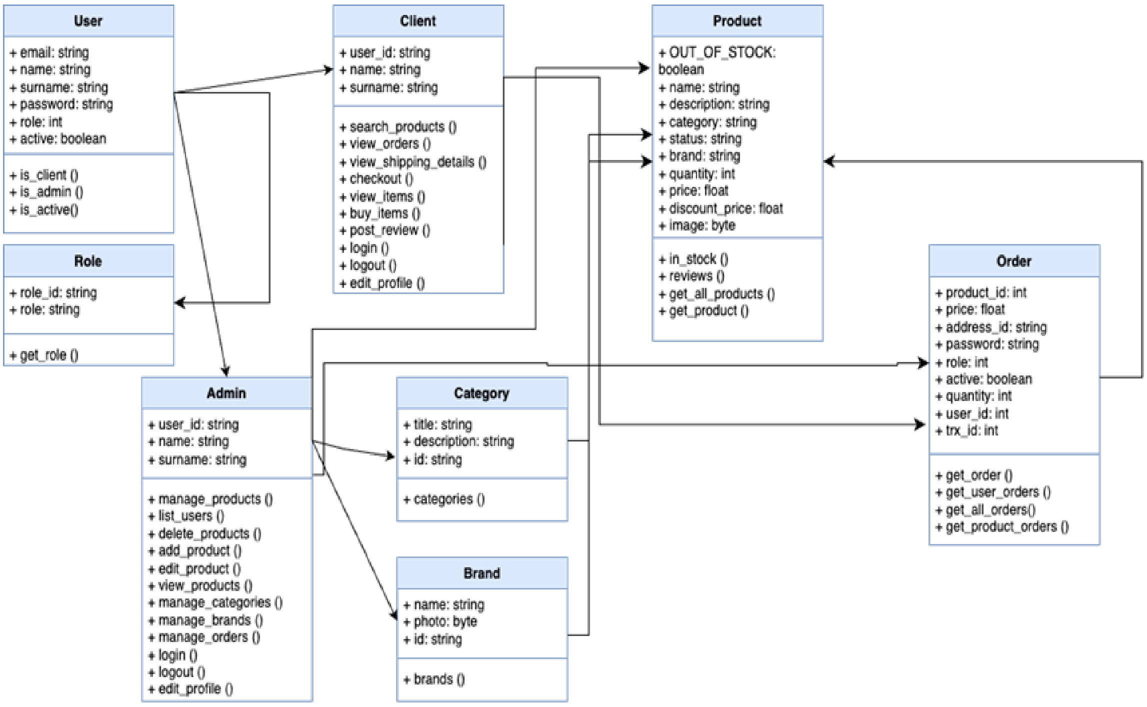


Figure 6 Class Diagram

3.5.4 Sequence Diagrams

Sequence Diagrams are designs that illustrate specific process interactions organized in the time frame they occur in. Below is detailed a sequence diagram of the users' actions on the website, as well as product and shopping cart control from both the admin and client sides. Sequence diagrams are composed of the classes or objects presented at the top of the diagram, the messages that depict the activities performed, the interactions involved presented by the arrows, the user who performs these tasks, and the lifeline that indicates the presence of the product in the sequence of the actions.

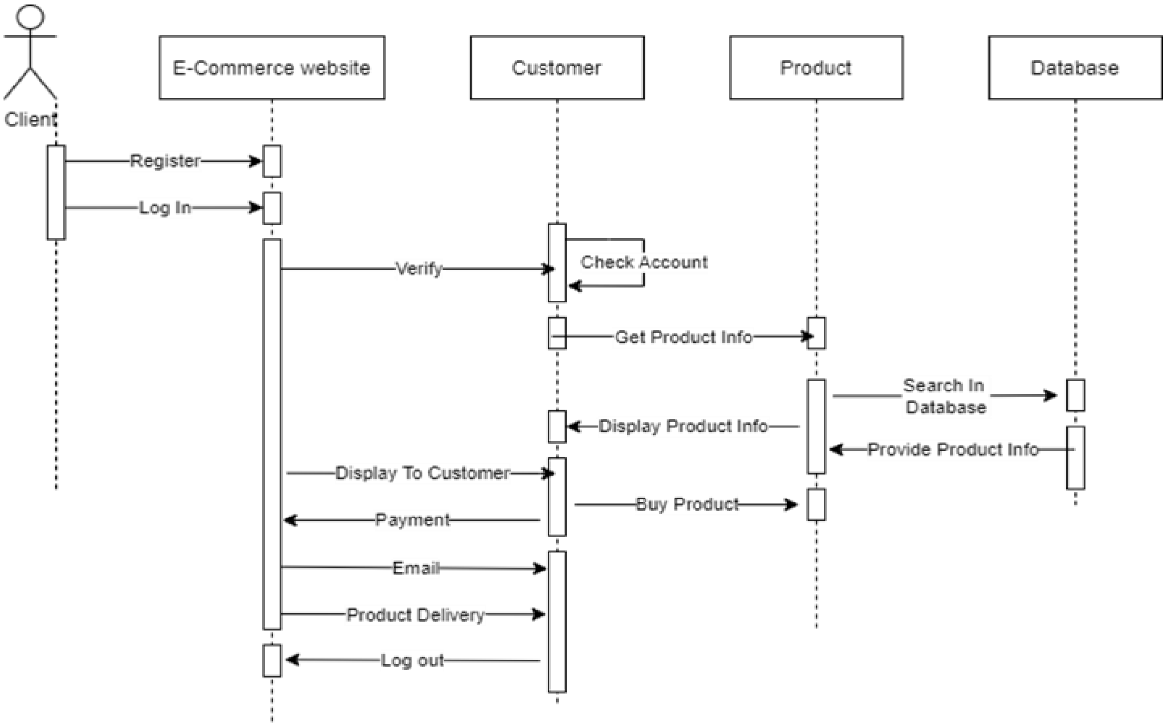


Figure 7 Client Actions Seq. Diagram

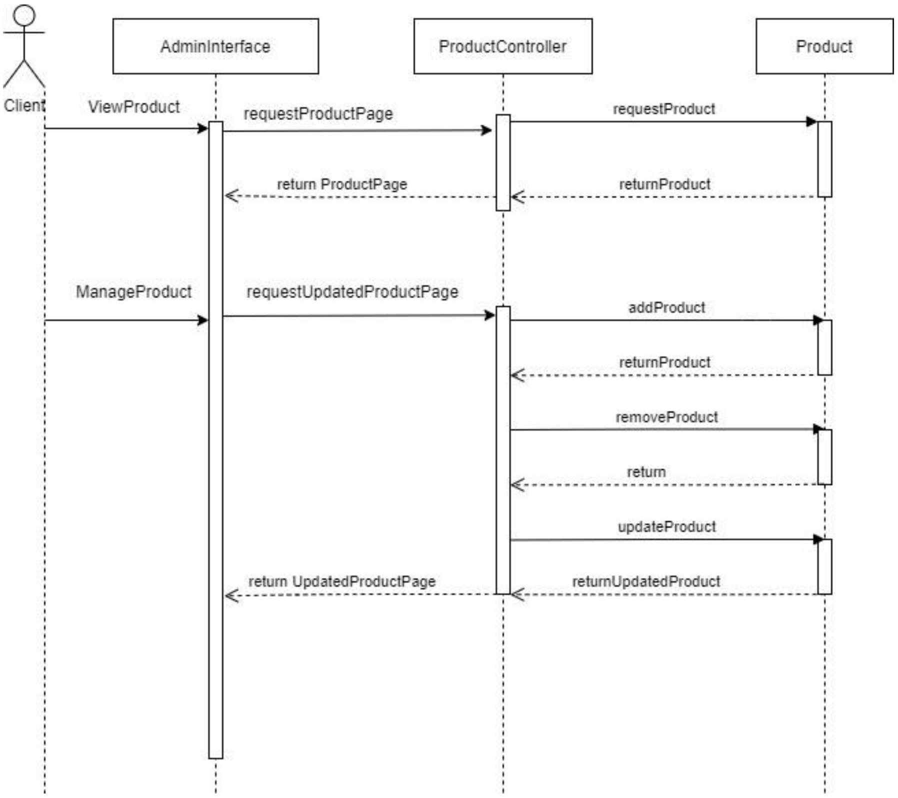


Figure 8 Admin Product Management Seq. Diagram

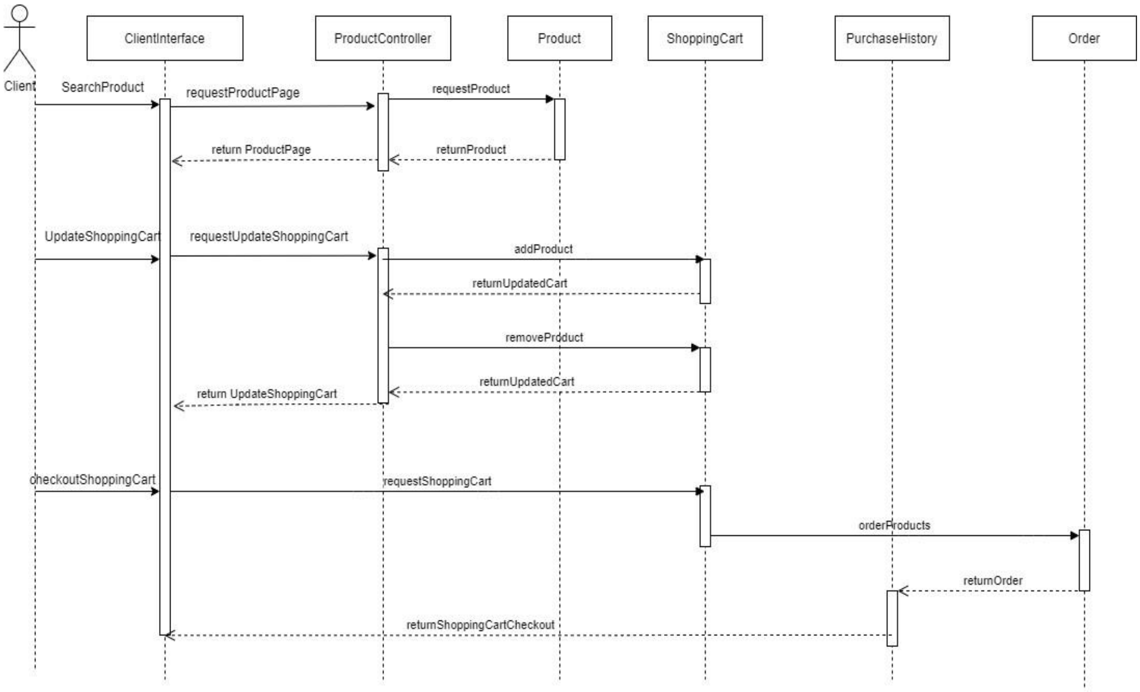


Figure 9 Action on Shopping Cart Seq. Diagram

3.5.5 Use Case Diagrams

Use case diagrams represent tasks of how the users interact with the system. On the below use, case is presented all the tasks a client, a guest, or the admin of the software can perform. As seen below the client and admin can both login, log out from the system, view the products, and check order details and the previous order data. The other actions shown are the individual actions each of the parties can perform.

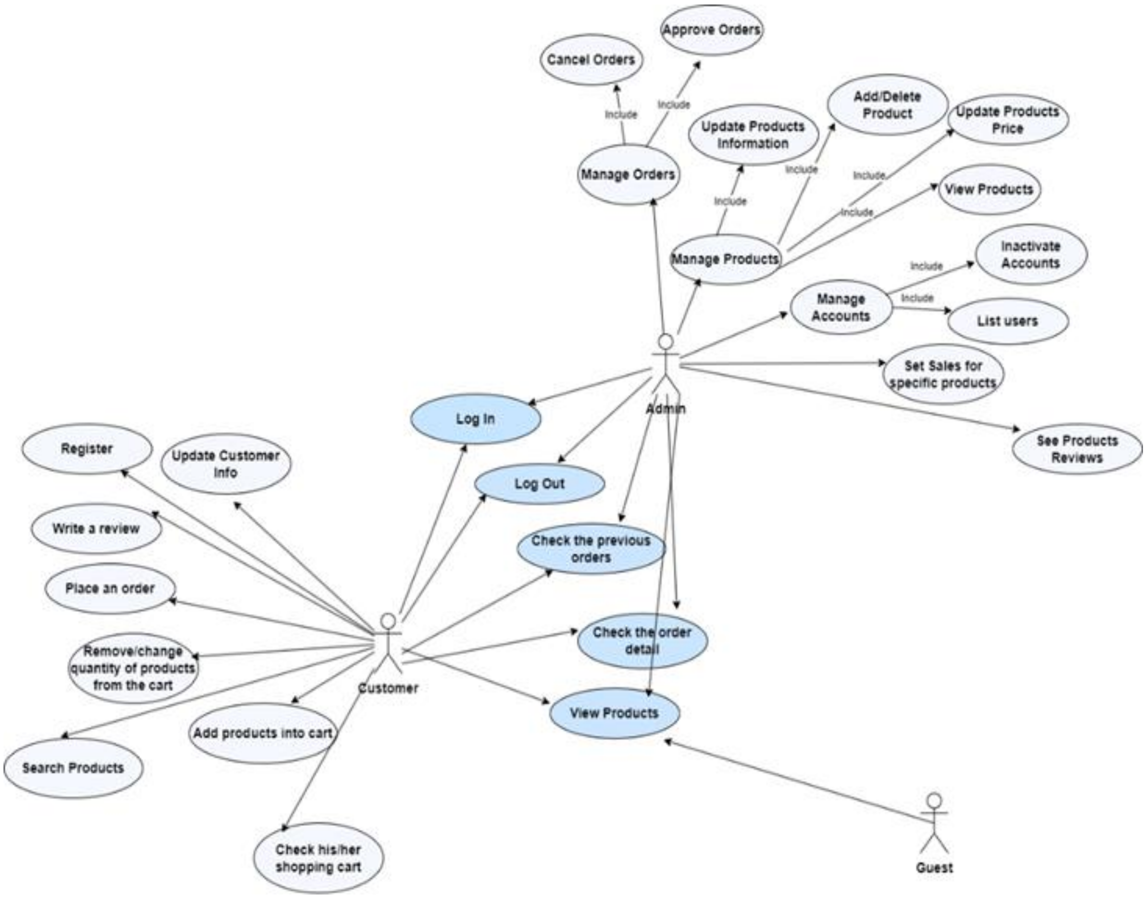


Figure 10 Use Case Diagram

3.5.6 Database Design

The database diagram shows in a detailed way the relations between the entities, which form the structure of the software. It presents how all the entities (user, admin, product, order, shopping cart) are related and affect each other. The database diagram provides a better view of how the system in general will work. The data types for each of the entities of the system are chosen appropriately to ensure efficient storage and retrieval of data and use proper normalization techniques to minimize data redundancy and ensure data integrity.

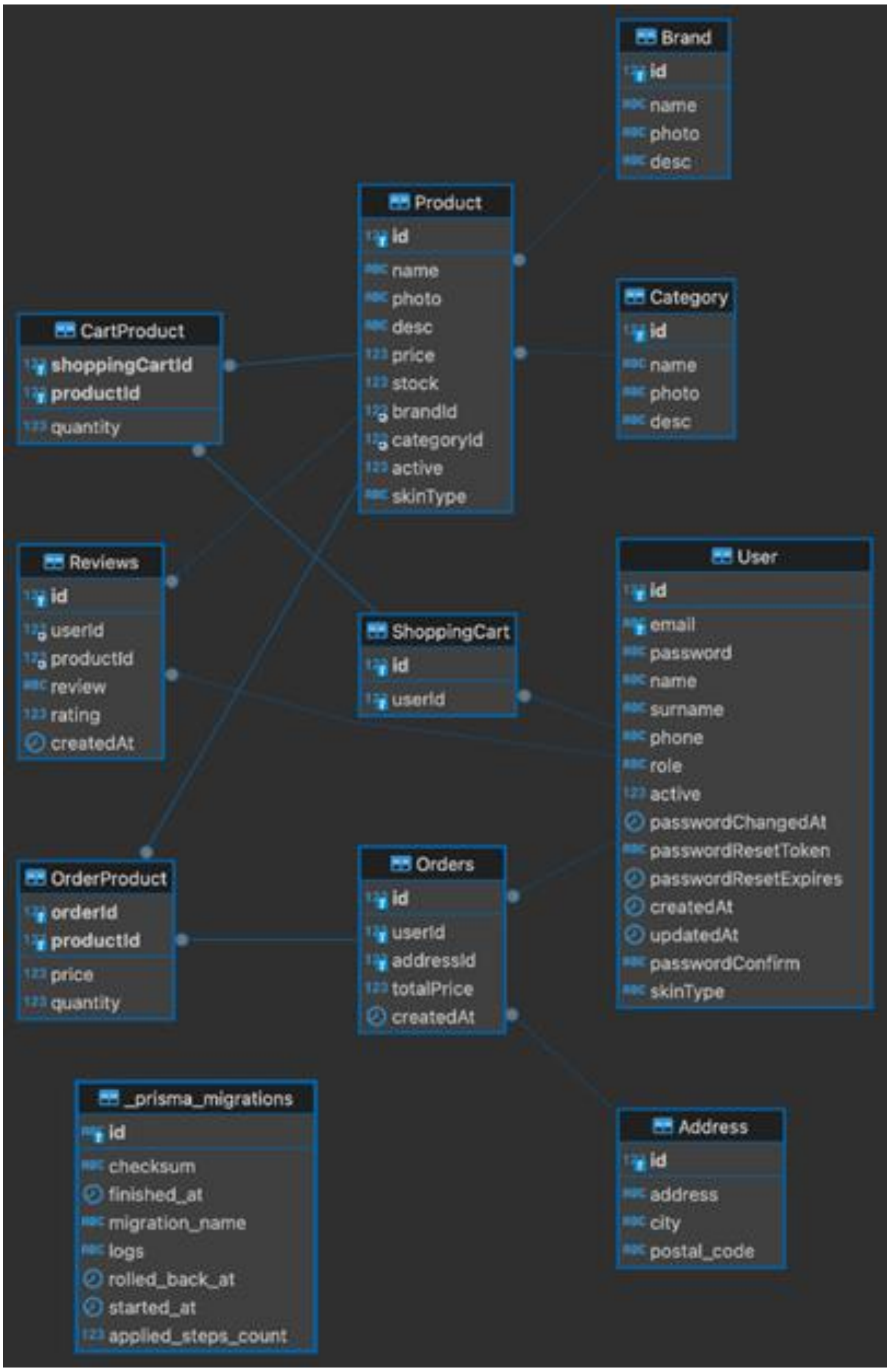


Figure 11 Software Database Schema

3.5.7 Activity Diagrams

Activity diagrams show the process from a start point to the finish point in detail including all the decision paths that can occur on the performing of the process.   These diagrams show a detailed visual form of every case that can happen during the performing of the activities from the clients or the admin. For example, they show every case possible for the process of order approval, if the information is correct or the following actions if there is something wrong with the order or clients’ information, etc. It helps us to better understand all the possibilities that can occur and helps us to deliver a more efficient product with fewer bugs.

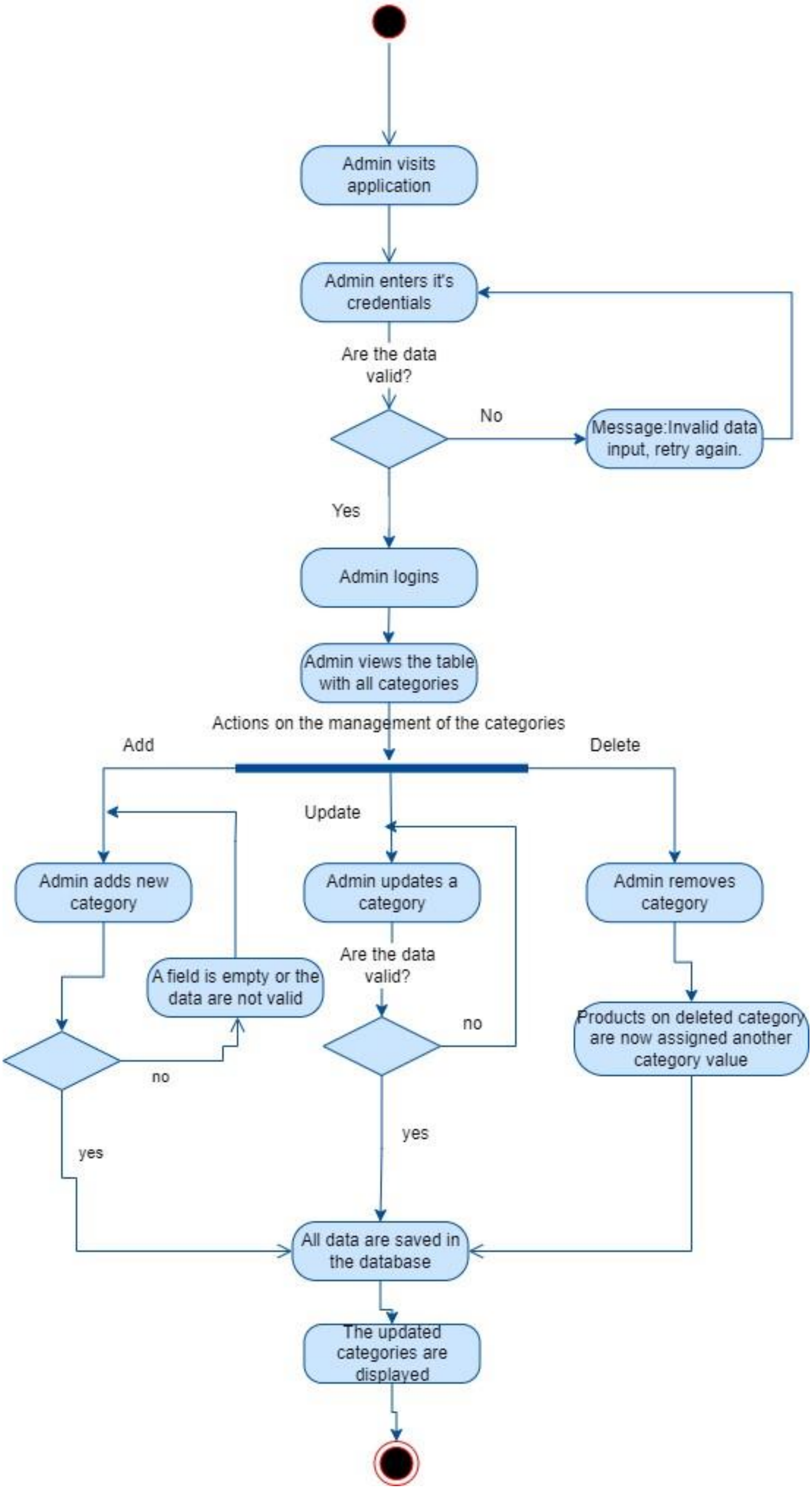

Admin visits application
Admin enters it's credentials
Are the data valid?
No
Message:Invalid data input, retry again.
Yes
Admin logins
Admin views the table with all categories
Actions on the management of the categories
Add
Delete
Update
Admin adds new category
Admin updates a category
Admin removes category
A field is empty or the data are not valid
Are the data valid?
no
Products on deleted category are now assigned another category value
no
yes
yes
All data are saved in the database
The updated categories are displayed

Figure 12 Admin Managing Categories Activity D.

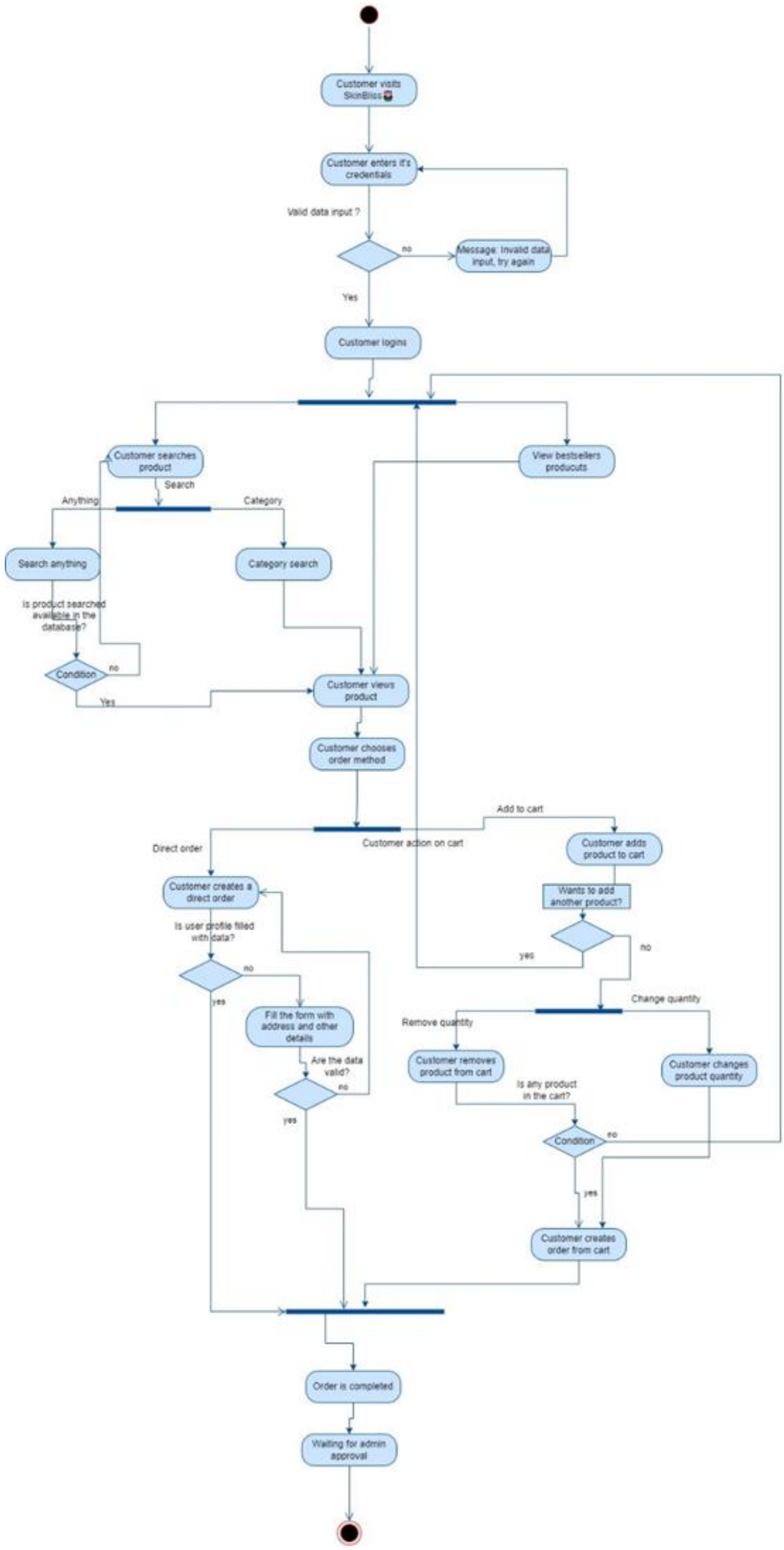

Customer visits SkinBliss
Customer enters it's credentials
Valid data input ?
no
Message: Invalid data input, try again
Yes
Customer logins
Customer searches product
View bestsellers producuts
Search
Anything
Category
Search anything
Category search
Is product searched available in the database?
Condition
no
Yes
Customer views product
Customer chooses order method
Add to cart
Direct order
Customer action on cart
Customer adds product to cart
Wants to add another product?
yes
no
Customer creates a direct order
Is user profile filled with data?
no
yes
Fill the form with address and other details
Are the data valid?
no
yes
Change quantity
Remove quantity
Customer removes product from cart
Customer changes product quantity
Is any product in the cart?
Condition
no
yes
Customer creates order from cart
Order is completed
Waiting for admin approval

Figure 13 Customer Setting Order Activity D.

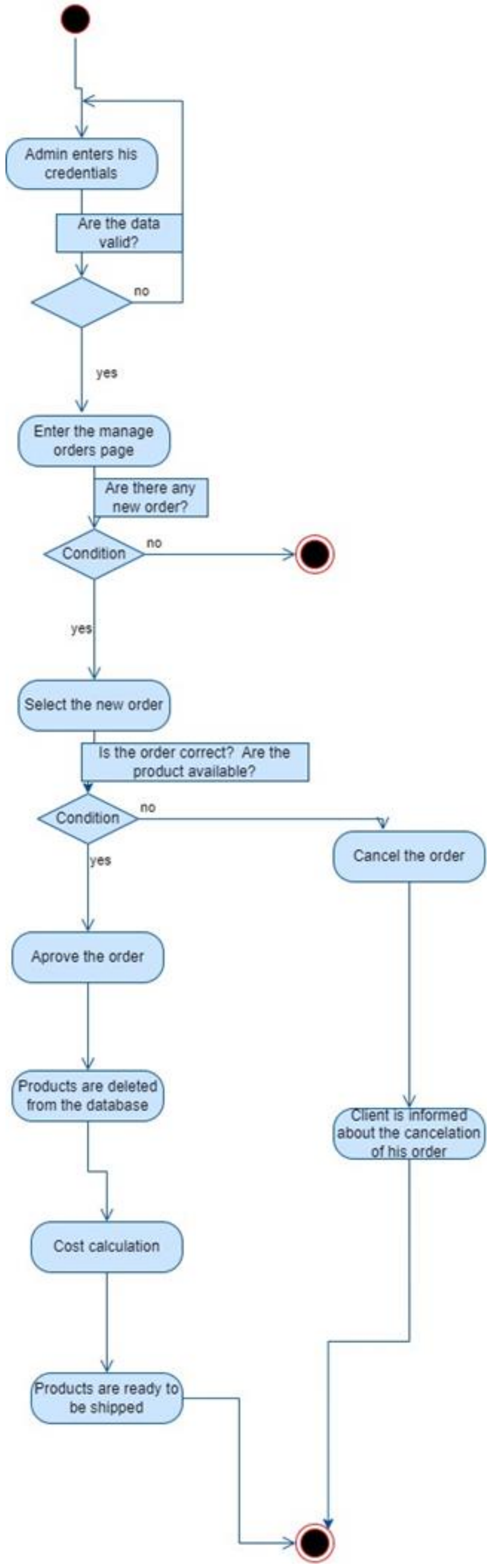

Admin enters his credentials
Are the data valid?
no
yes
Enter the manage orders page
Are there any new order?
Condition
no
yes
Select the new order
Is the order correct? Are the product available?
Condition
no
yes
Cancel the order
Aprove the order
Products are deleted from the database
Client is informed about the cancelation of his order
Cost calculation
Products are ready to be shipped

Figure 14Admin Action on Order Activity D.

3.5.8 State Diagram

State diagrams in contrast to the earlier diagrams are behavioral diagrams, that show the different behaviors the system gives in response to different internal and external events. The state diagram presents each state as a node that changes after a specific event occurs. These diagrams play a crucial role in determining whether our software is not functioning effectively, particularly for tasks with several steps, such as admin management processes, or the client interface with the software.

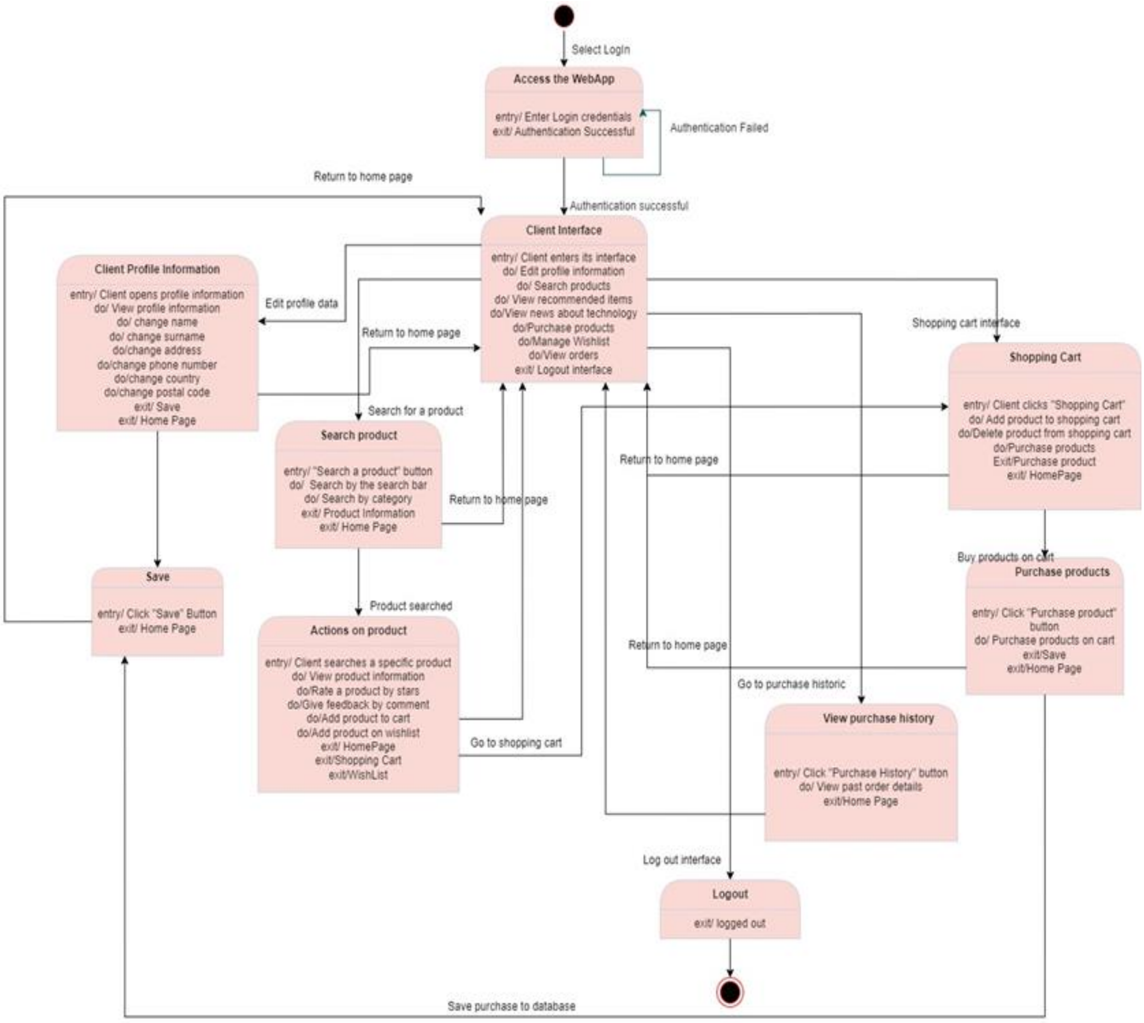


Figure 15 User Interface State Diagram

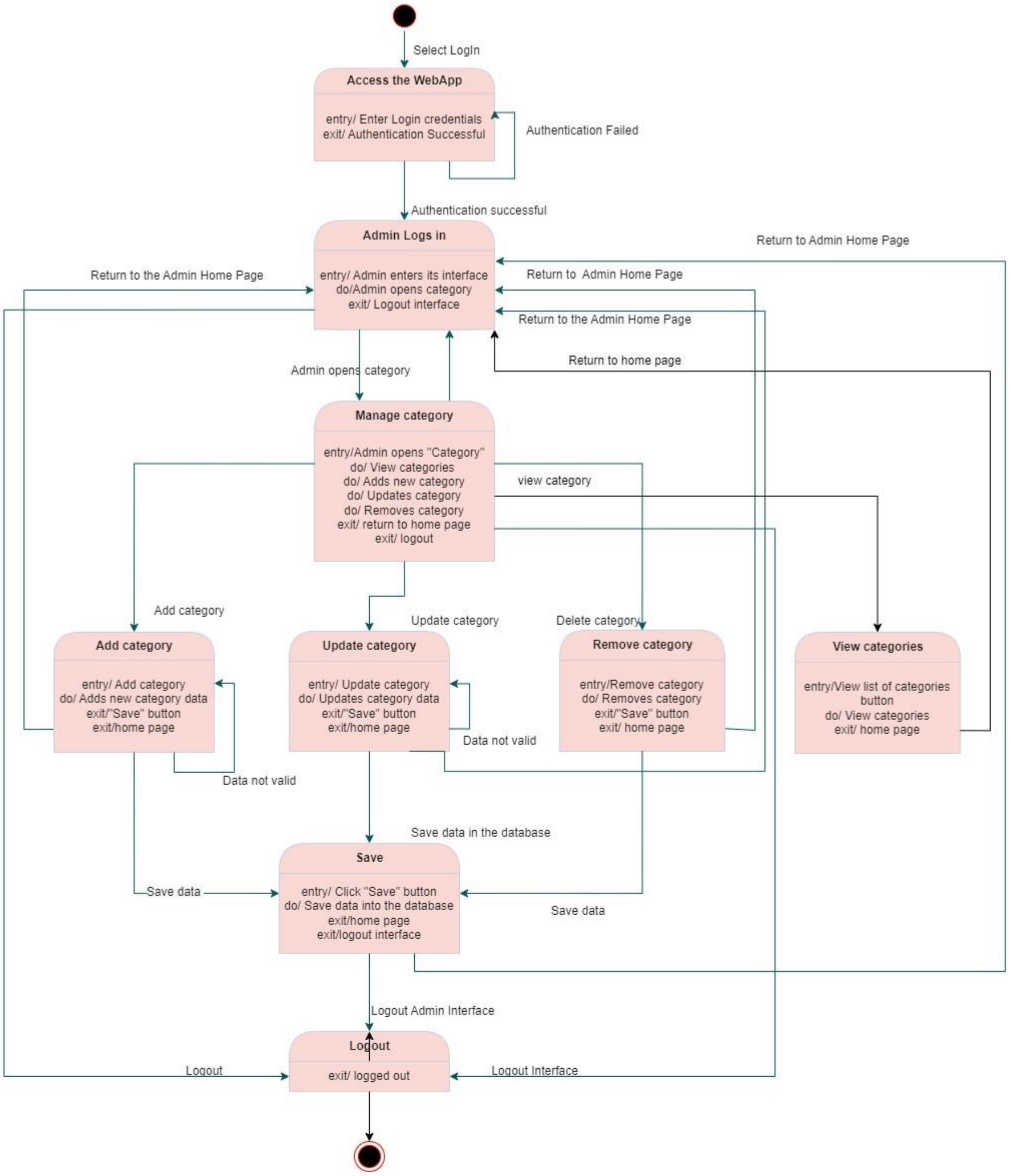


Figure 16 Admin Manage Category State D.

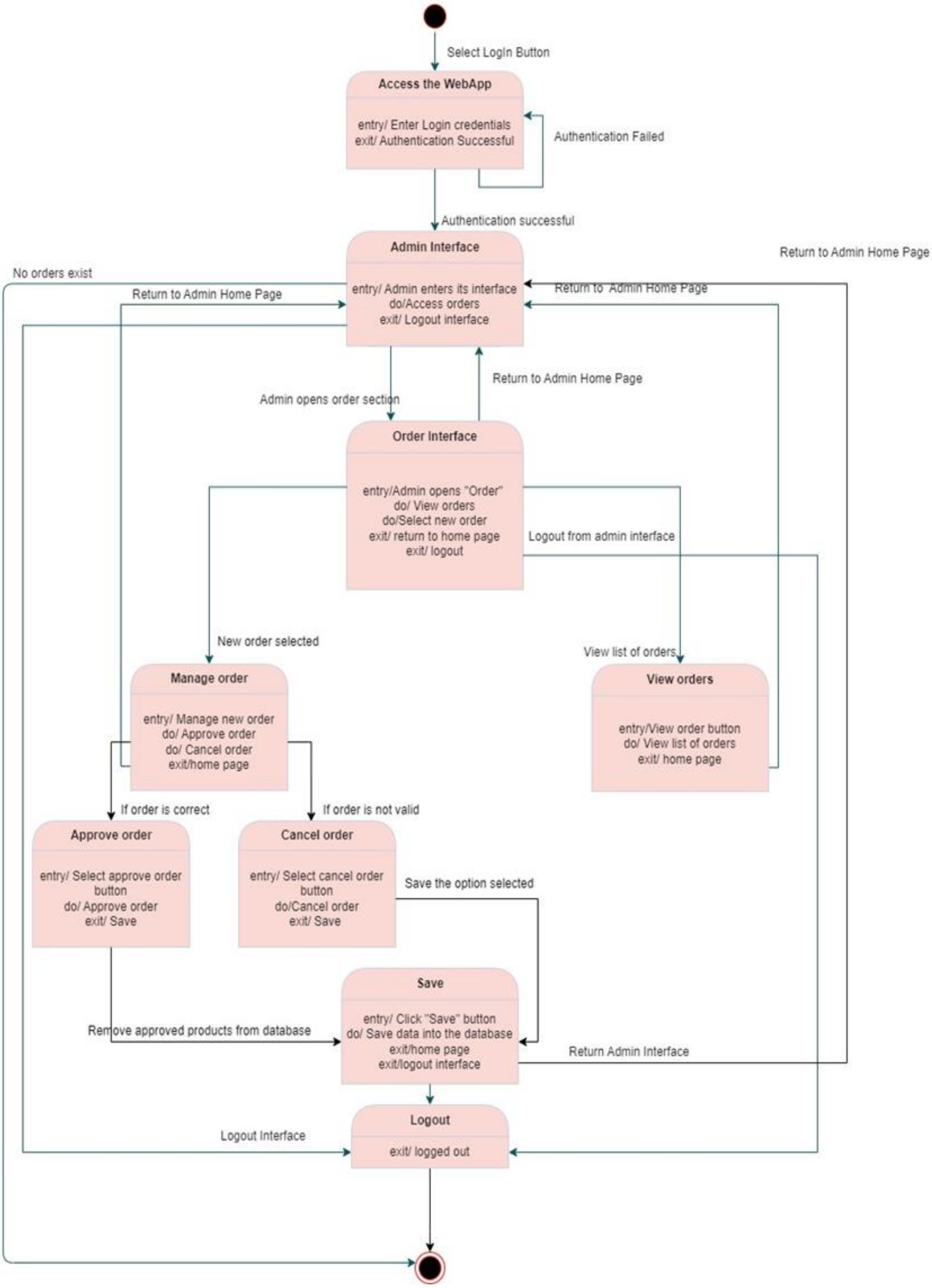


Figure 17 Admin Manage Order State D.

CHAPTER 4:
IMPLEMENTATION

This chapter is going to explain in detail the development of the software, the technologies used, the features of the website, and other specifics.

4.1 Technologies Used

We have decided to utilize the following technologies to achieve the delivery of a secure, responsive, fast web application:

Node.js is created to help developers create both front-end and back-end applications by using JavaScript. It was first introduced in 2009 by Ryan Dahl. Node.js is an open-source, independent, JavaScript runtime environment, built on Chrome V8 JavaScript engine. (Semah, 2022) Node.js is more efficient than other technologies because it is non-blocking I/O as the execution of the programs continues without waiting for the I/O operation to finish. Later when the operation is finished, a callback is triggered to handle the operation result, which will improve the performance of the CPU. (Zulfiqar, 2024) Node.js can handle large numbers of connections to a single server without needing to manage string simultaneity, which is a major source of errors. Node.js has the distinct advantage that many front-end programmers who write JavaScript for the application can already write both server-side and customer-side code without acquiring anything new.

Express.js is a user-friendly library of Node.js that facilitates the process of software development. It is a flexible framework which makes it easier to work with. Since it is a framework of Node.js most of the code is already developed, which makes it easier for the developers to create web applications and APIs. With Express.js the handling of routes, requests, and responses is easier which leads to an easier process of developing flexible applications. (GeeksforGeeks, 2024)

Prisma is an open-source ORM for Node.js. (NestJs, n.d.) It is used for composing the program code with the database structures. In this way, we don't need to write database queries because it integrates the program code into database structures.

MySQL is one of the most known open-source RDBMS. It is used to organize the data of a system in one or more tables related to each other, whose relations form the structure of the system. SQL stands for Structured Query Language and is used to create, adjust, and extract data from the database, as well controlling the users' actions on it. MySQL works with an OS to install a relational database into a computer's storage system, control users, enable network access, etc. (Wikipedia, 2024)

React or ReactJS is a free open-source frontend JavaScript-based library used to develop the user interface of the software. It is controlled by Facebook and a community of companies and individual developers. React is used for the advancing of single-page, mobile, or server-rendered systems. Its advantage is that it rerenders only those parts that have changed avoiding in this way unneeded rerendering of unchanged elements. (Wikipedia, 2024) It is a very popular library in web development even though it is not a programming language.

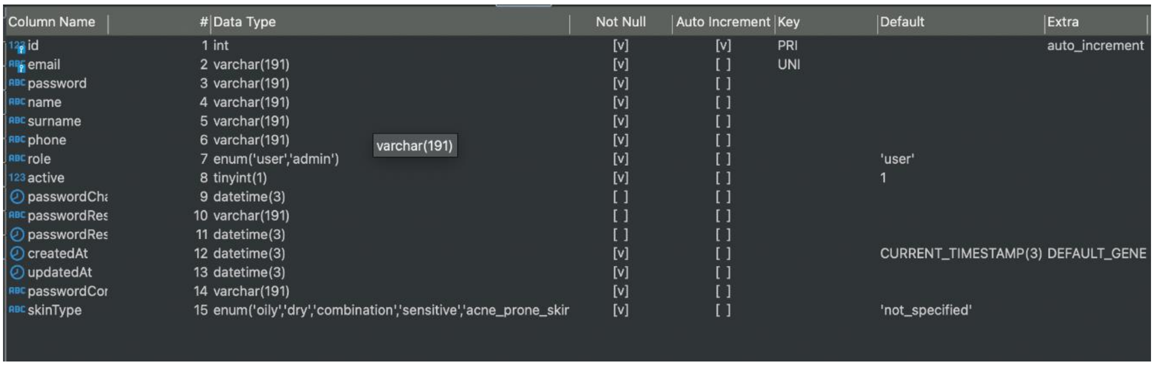
Bootstrap is a free open-source CSS framework concerned with the styling of the front end. It contains HTML, CSS, and JavaScript design templates for font type, buttons, navigation, and other interface components. The main purpose of utilizing BootStrap in a project is to apply its choices of color, size, style, and layout. After added it provides basic styles for all HTML elements present. The developers can further use the CSS classes defined In Bootstrap to adjust the display of their systems' contents. (Wikipedia, 2024) With the utilization of Bootstrap, the process of styling the SkinBliss interface has become effortless.

Next.js is a new free open-source framework created by the company Vercel, providing web applications developed with React with server-side rendering and static website generation. It is one of the most popular React frameworks and is a recommended “toolchain” when starting a new web application development. (Wikipedia, 2024)

Redux is an open-source JS library concerned with state management. It is used to store the state of the variables on the web application. Redux can be considered as “a storage” that can be accessed by the web application components in a structured way. It has to be accessed through a “Reducer” and “Actions”. (ElHousieny, 2021)

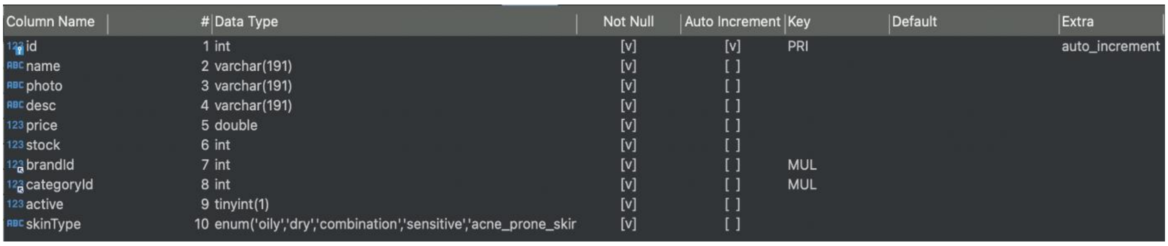
4.2 Database Schema Overview

As previously mentioned, we have used MySQL for the organization of the data from the database. A well-organized database allows for efficient retrieval of information in RDBMS applications, reducing both resources and time. In this sector of the paper, we are going to explain all the database tables that are crucial for the establishment of the software structure and data organization and maintenance. Below are appended the database table schemas of this app.



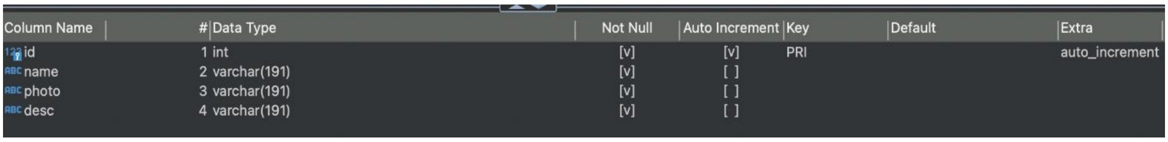
Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra
id	1	int	[v]	[v]	PRI		auto_increment
email	2	varchar(191)	[v]	[ ]	UNI		
password	3	varchar(191)	[v]	[ ]			
name	4	varchar(191)	[v]	[ ]			
surname	5	varchar(191)	[v]	[ ]			
phone	6	varchar(191)	[v]	[ ]			
role	7	enum('user','admin')	[v]	[ ]		'user'	
active	8	tinyint(1)	[v]	[ ]		1	
passwordCha	9	datetime(3)	[ ]	[ ]			
passwordRes	10	varchar(191)	[ ]	[ ]			
passwordRes	11	datetime(3)	[ ]	[ ]			
createdAt	12	datetime(3)	[v]	[ ]		CURRENT_TIMESTAMP(3)	DEFAULT_GENE
updatedAt	13	datetime(3)	[v]	[ ]			
passwordCor	14	varchar(191)	[v]	[ ]			
skinType	15	enum('oily','dry','combination','sensitive','acne_prone_skir	[v]	[ ]		'not_specified'	

Figure 18 User Schema



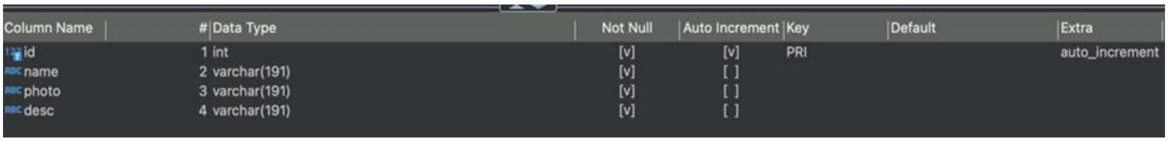
Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra
id	1	int	[v]	[v]	PRI		auto_increment
name	2	varchar(191)	[v]	[ ]			
photo	3	varchar(191)	[v]	[ ]			
desc	4	varchar(191)	[v]	[ ]			
price	5	double	[v]	[ ]			
stock	6	int	[v]	[ ]			
brandId	7	int	[v]	[ ]	MUL		
categoryId	8	int	[v]	[ ]	MUL		
active	9	tinyint(1)	[v]	[ ]			
skinType	10	enum('oily','dry','combination','sensitive','acne_prone_skir	[v]	[ ]			

Figure 19 Product Schema



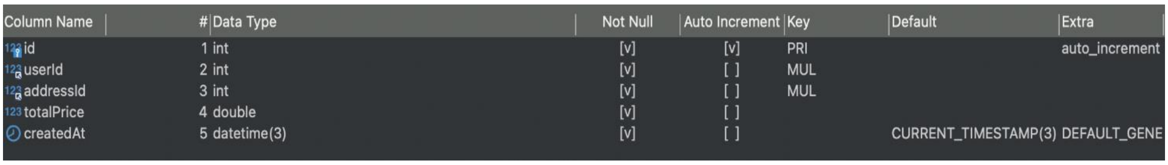
Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra
id	1	int	[v]	[v]	PRI		auto_increment
name	2	varchar(191)	[v]	[ ]			
photo	3	varchar(191)	[v]	[ ]			
desc	4	varchar(191)	[v]	[ ]			

Figure 20 Category Schema



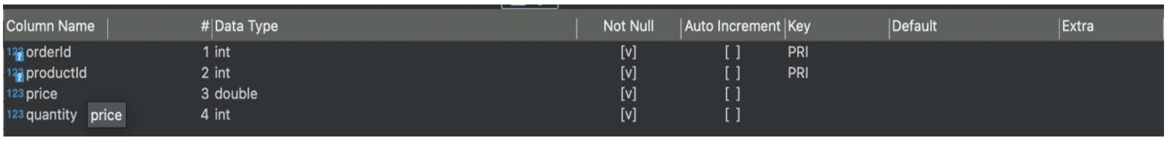
Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra
id	1	int	[v]	[v]	PRI		auto_increment
name	2	varchar(191)	[v]	[ ]			
photo	3	varchar(191)	[v]	[ ]			
desc	4	varchar(191)	[v]	[ ]			

Figure 21 Brand Schema



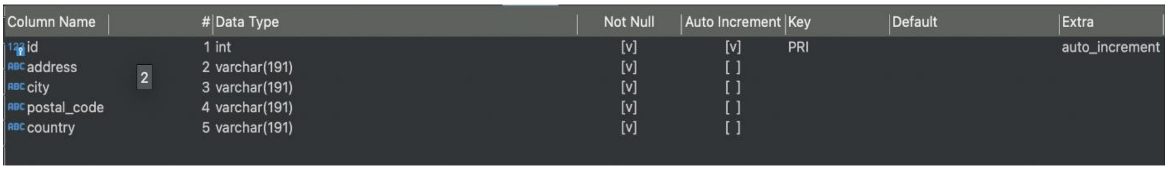
Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra
id	1	int	[v]	[v]	PRI		auto_increment
userId	2	int	[v]	[ ]	MUL		
addressId	3	int	[v]	[ ]	MUL		
totalPrice	4	double	[v]	[ ]			
createdAt	5	datetime(3)	[v]	[ ]		CURRENT_TIMESTAMP(3)	DEFAULT_GENE

Figure 22 Order Schema



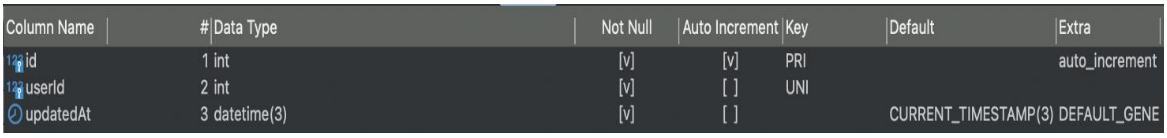
Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra
orderId	1	int	[v]	[ ]	PRI		
productId	2	int	[v]	[ ]	PRI		
price	3	double	[v]	[ ]			
quantity price	4	int	[v]	[ ]			

Figure 23 OrderProduct Schema



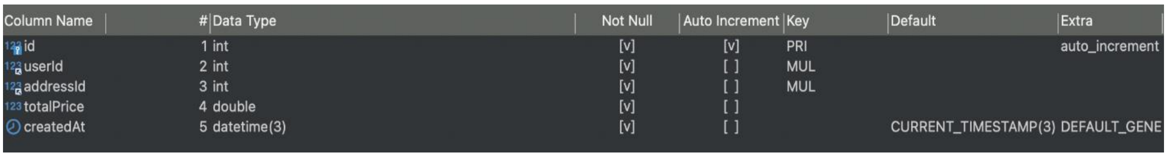
Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra
id	1	int	[v]	[v]	PRI		auto_increment
address 2	2	varchar(191)	[v]	[ ]			
city	3	varchar(191)	[v]	[ ]			
postal_code	4	varchar(191)	[v]	[ ]			
country	5	varchar(191)	[v]	[ ]			

Figure 24 Address Schema



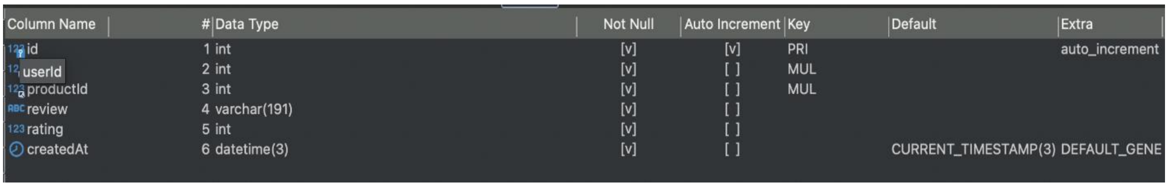
Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra
id	1	int	[v]	[v]	PRI		auto_increment
userId	2	int	[v]	[ ]	UNI		
updatedAt	3	datetime(3)	[v]	[ ]		CURRENT_TIMESTAMP(3) DEFAULT_GENE	

Figure 25 ShoppingCart Schema



Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra
id	1	int	[v]	[v]	PRI		auto_increment
userId	2	int	[v]	[ ]	MUL		
addressId	3	int	[v]	[ ]	MUL		
totalPrice	4	double	[v]	[ ]			
createdAt	5	datetime(3)	[v]	[ ]		CURRENT_TIMESTAMP(3) DEFAULT_GENE	

Figure 26 CartProduct Schema



Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra
id	1	int	[v]	[v]	PRI		auto_increment
userId	2	int	[v]	[ ]	MUL		
productId	3	int	[v]	[ ]	MUL		
review	4	varchar(191)	[v]	[ ]			
rating	5	int	[v]	[ ]			
createdAt	6	datetime(3)	[v]	[ ]		CURRENT_TIMESTAMP(3) DEFAULT_GENE	

Figure 27 Review Schema

4.3 Software's Features

A well-designed website makes an immediate positive impression on customers, which leads to higher user experience and improves customer engagement. (M., 2024) As a result, before designing SkinBliss, we undertook thorough research on how to make it as good as it could be. One of the primary reasons e-commerce websites succeed is their user-friendly features, which we did our best to design. SkinBliss' features would include:

User Interface: The user Interface will serve as the gateway for all the people interested in seeing the SkinBliss web application. It presents the visual part of this web application and allows the users to see the different sections of the application.

Authentication: This is used to secure access to the specific features and information of each role account (user/admin). During the registration process, the users fill in their information such as name, surname, email, phone number, and password which are securely stored and used later for the login process.

Customer Panel: Customers can create an account by entering their name, last name, email, phone number, and password. After someone has registered on our software he/she can log in or sign up for an account easily from any page of the website. Registered users can browse the products available, set orders, view order history, leave reviews, and update personal data via their account dashboard.

Admin Panel: The software administrator logs into the admin panel by inputting his credentials. The web application administrator can manage the products and categories by updating, deleting, or adding new ones, manage client orders (approve or cancel them), manage user accounts (deactivate non-active accounts), and view the products available, categories, and reviews, just as users can.

Product Catalogue: The product catalog or categorization is the most important part of all our web applications. This part makes it very easy for the client to search for what he needs. Products are organized based on their function, brand, and reviews, allowing the user to browse products based on their preferences and needs.

Personalized Recommendations: As mentioned above we are going to categorize our products based on their function and purpose. Users will select their skin characteristics and algorithms will analyze this data to generate later the recommended products based on their skin imperfections.

Shopping Cart: Products from product listings or details pages can be added to users' shopping carts with just one click. The shopping cart page will present a summary of all selected items, including quantities, prices, and subtotal. Users can modify quantities, remove items, or proceed to checkout directly from the shopping cart.

Checkout Process: Users will be guided through the necessary steps to finish their purchase during the checkout process. The user will securely submit payment information, choose a shipping method, and enter their shipping address. Before completing the purchase, you can review the order summary and total cost. Users will receive an email with order details and a confirmation message after the process is successful.

Order Management: This system will be used to enable customers to manage their orders. Once the order is set, they can check it through the section.

Review Section: SkinBliss includes a review section where users can share their experiences and opinions on purchased products. A transparent review section will contribute to the platform's credibility and help new customers make more informed decisions about product suitability.

Comprehensive Product Information: Each of our products will have detailed information about its function and ingredients appended with the respective pictures. Additionally, SkinBliss will be equipped with a transparent review section, where new customers can see the honest opinions of people who have already used the products.

Performance and security: The web application is designed for fast loading times and efficient data recovery, increasing overall responsiveness. As for the security part, it includes security measures such as encryption protocols and secure payment processing, safeguarding user data and privacy.

CHAPTER 5:
TESTING

Software testing is a critical process in the software life cycle that aims to examine software's quality, functionality, and performance before its deployment. It is the last step before the software launch where the developers have the opportunity to check the product manually or run test cases and fix the bugs or errors that may appear, to improve the reliability and correctness of the software. (Katalon, 2024)

The primary purpose of testing is to defect the errors early so it reduces the developers the time and effort that will be needed to fix them later. Furthermore, testing is conducted to ensure that the software that is going to be delivered meets the specified requirements and functions, making sure that everything is working properly. The testing process is made through test cases. A test case consists of a comprehensive set of inputs that check if the requirements of the app are met.

5.1 Test Cases for Registration Process

Test Case ID: Test_Register_1

Test Objective: Verify if a new user can register successfully to the app.

Test Steps:

Go to the Registration Page.

Enter the valid information on all the required fields (name, surname, phone number, email, password, password confirm).

Click Sign Up.

Test Input: Valid Data

Expected Result: You should agree to the Terms of Conditions.

Test Result: You should agree to the Terms of Conditions.

Test Case ID: Test_Register_2

Test Objective: Verify if a new user can register successfully to the app.

Test Steps:

Go to the Registration Page.

Enter the valid information on all the required fields (name, surname, phone number, email, password, password confirm).

Agree to the Terms of Conditions.

Click Sign Up.

Test Input: Valid Data

Expected Result: The user registered successfully.

Test Result: The user registered successfully.

Test Case ID: Test_Register_3

Test Objective: Verify if a new user can register successfully to the app.

Test Steps:

Go to the Registration Page.

Enter the information on all the required fields (name, surname, phone number, email, password, password confirm).

Agree to the Terms of Conditions.

Click Sign Up.

Test Input: Invalid Email, Other Data Valid

Expected Result: Registration not successful, email entered is invalid.

Test Result: Registration not successful, email entered is invalid.

5.2 Test Cases for Login Process

Test Case ID: Test_Login_1

Test Objective: Verify if a user can log in successfully to the app.

Test Steps:

Go to the Log in Page.

Enter the information on all the required fields (email, password).

Click Log In.

Test Input: Valid Data

Expected Result: Log in Successful.

Test Result: Log in Successful.

Test Case ID: Test_Login_2

Test Objective: Verify if a user can log in successfully to the app.

Test Steps:

Go to the Log in Page.

Enter the information on all the required fields (email, password).

Click Log In.

Test Input: Invalid Data

Expected Result: Invalid Data, login failed.

Test Result: Invalid Data, login failed.

5.3 Test Case for Product Search

Test Case ID: Test_Search_1

Test Objective: Verify if a product will appear after searching its name.

Test Steps:

Enter the Product Name in the Search bar.

Click the “Search” button.

Test Input: Valid Data

Expected Result: All the products that fulfill the search condition will appear.

Test Result: All the products that fulfill the search condition will appear.

Test Case ID: Test_Search_2

Test Objective: Verify if a product will appear after searching its name.

Test Steps:

Enter the Product Name in the Search bar.

Click the “Search” button.

Test Input: Invalid Data

Expected Result: The product does not exist.

Test Result: The product does not exist.

5.4 Test Case for Product Browse by Category

Test Case ID: Test_Search_by_Category

Test Objective: Verify if users can browse products by category.

Test Steps:

Go to the Home Page.

Select Product Category.

Expected Result: Products in the selected category are displayed.

Test Result: Products in the selected category are displayed.

5.5 Test Case for Adding Products to the Cart.

Test Case ID: Test_Add_Product_to_Cart

Test Objective: Verify if users can add products to the cart.

Test Steps:

Go to a Product Detail Page.

Select “Add to Cart”.

Expected Result: Products are added to the cart, message “Added to Cart” is displayed.

Test Result: Products are added to the cart, message “Added to Cart” is displayed.

5.6 Test Case for CheckOut Process

Test Case ID: Test_Checkout_Process_1

Test Objective: Verify if the checkout process works correctly.

Test Steps:

Navigate to the Cart page.

Enter the valid information on all the required fields (email, password).

Click the “Checkout” button.

Enter the information on all the required fields and the shipping information.

Test Input: Valid Data

Expected Result: Redirects you to the page where you can see your complete order receipt.

Test Result: Redirects you to the page where you can see your complete order receipt.

Test Case ID: Test_Checkout_Process_2

Test Objective: Verify if the checkout process works correctly.

Test Steps:

Navigate to the Cart page.

Enter the valid information on all the required fields (email, password).

Click the “Checkout” button.

Enter the information on all the required fields and the shipping information.

Test Input: Invalid Data

Expected Result: The data entered is incorrect.

Test Result: The data entered is incorrect.

5.7 Test Case for Leaving A Product Review.

Test Case ID: Test_Leave_Review

Test Objective: Ensuring that a logged-in user can leave reviews on the products.

Test Steps:

Go to a product detail page.

Go to the Reviews Section.

Enter a review and a rating (1-5 stars).

Select the “Submit Review” button.

Expected Result: The review is displayed in the reviews section.

Test Result: The review is displayed in the reviews section.

5.8 Test Case for Calculating Skin Type

Test Case ID: Calculate_Skin_Type

Test Objective: Ensuring that the software can generate the Skintype after the user answers the provided questions.

Test steps:

User Answers to the following questions:

How often does your skin feel shiny?

How often do you have enlarged pores?

How often do you have blackheads or breakouts?

How often does your skin feel tight or dry?

How often do you use a moisturizer?

How often does your skin look flaky or ashy?

Do you experience oily skin in certain areas and dryness in others?

How often do you experience redness or irritation?

How often does your skin react to new products or environmental changes?

How often do you have breakouts?

How often do you have inflamed or cystic acne?

Do you have dark spots or patches on your skin?

How often do you have post-inflammatory hyperpigmentation (dark marks after acne

or injury)?

How often does your skin appear uneven in tone or texture?

Expected Results: The app generates the user's type of skin (normal, dry, oily, combination, acne-prone, dry acne-prone, sensitive, hyperpigmentation.)

Actual Results: The app generates the user's type of skin (normal, dry, oily, combination, acne-prone, dry acne-prone, sensitive, hyperpigmentation.)

CHAPTER 6:
CONCLUSION AND FUTURE WORK

At the end of this thesis, we can describe all the stages we went through to deliver successful software that fulfills all the established requirements and reflects simplicity and functionality. It is developed using technologies such as JavaScript, HTML, CSS, Reajct.js, Node.js, Express.js, and MySQL to provide user-friendly and efficient software with a secure database.

SkinBliss is an e-commerce website created to make the process of purchasing skincare products online easier and more reliable. As mentioned above, online shopping offers convenience, but the skincare industry is very sensitive and has made people distrustful of e-commerce skincare. Our software with its features aims to increase the reliability and comfort of people in their purchases. The system is divided into two main interfaces, the client and the admin. Clients have limited access to the software but sufficient to fulfill their requirements. A client has the opportunity to browse a variety of products from which he can choose, through easy navigation on the website pages.

Categorizing products in an efficient and easily accessible way is one of the features that sets our system apart from those already on the market. Filtering products based on price range, function, and brand aims to make the process of finding the product easier. An innovative and crucial feature of SkinBliss is the personalization of the routine, which will be automatically generated by the system after the user has performed a virtual skin test where all the characteristics of the skin will be given as input, and will receive the type of skin, together with the recommended products. Skin issues and information products will be stored in a warehouse, to ensure the efficiency of our website and customer satisfaction.

The review section further reflects the transparency of the software since it cannot be modifiable by the administrator. Moreover, it is a great assistance for customers to make well-thought-out choices regarding their purchases.

After browsing the products, the clients can easily add them to the cart with a simple click, and the cart's information is visible to the client at every time. The client can manage the cart products whenever before placing an order. SkinBliss includes a feature that saves the shipping details of a client from the previous order, making the checkout process easier and more convenient for the user, and improving the shopping experience.

On the other hand, the admin will have an important role in managing and maintaining the system, controlling and managing users, orders, and products listed in the system. Admin can update the information, add or remove products from the system as well as change their categories ensuring accurate information. Order management includes the actions of order approval and cancellation that intend to simplify consumers' responsibilities and save the time that they require to understand the canceling process.

During this thesis, I have provided clear information on all the processes followed and in addition, I have attached UML diagrams to show a better picture of how the system and its components interact with each other.

Future updates would include adding a wishlist to make it easier for clients to save the products they want to receive in the future, creating a management system that will notify clients with emails when the products are back in stock, expanding payment options by including other forms of payment such as PayPal or card payments and also develop coupon codes for discounts. Finally, future work would include continuous updates to improve the software's activity and performance.

REFERENCES

Addo, R. M. (2023, October 3). *Ethical Coding and Privacy*. Retrieved from Linkedin: https://www.linkedin.com/pulse/ethical-coding-privacy-comprehensive-guide-developers-marvin-addo/

Advent Biztech Solution Pvt Ltd. (2023, March 27). *Exploring the Role of Technology in the Rapidly Evolving E-commerce Landscape*. Retrieved from Linkedin: https://www.linkedin.com/pulse/e-commerce-industry-has-seen-unprecedented/

Albanian Constitution. (2016, November 17). Retrieved from https://www.venice.coe.int/webforms/documents/default.aspx?pdffile=CDL-REF(2016)064-e

AppCurators. (2023, March 2). *E-commerce and the shift towards online shopping*. Retrieved from Linkedin: https://www.linkedin.com/pulse/e-commerce-shift-towards-online-shopping-appabodes/

ATLASSIAN. (n.d.). *Agile Methodology*. Retrieved from https://www.atlassian.com/agile#:~:text=The%20Agile%20methodology%20is%20a,planning%2C%20executing%2C%20and%20evaluating.

Awati, R. (n.d.). *Object-Relational Mapping (ORM)*. Retrieved from TechTarget: https://www.theserverside.com/definition/object-relational-mapping-ORM#:~:text=Object%2Drelational%20mapping%20(ORM)%20is%20a%20way%20to%20align,language%20and%20a%20relational%20database.

AWS. (n.d.). *SSL/TLS*. Retrieved from https://aws.amazon.com/compare/the-difference-between-ssl-and-tls/#:~:text=Both%20SSL%20and%20TLS%20are,SSL%202.0%20publicly%20in%201995.

Barney, N. (2023, November). *Authentication*. Retrieved from TechTarget: https://www.techtarget.com/searchsecurity/definition/authentication#:~:text=E%2Dcommerce%20transactions.,fraud%20and%20enhances%20customer%20trust.

ElHousieny, R. (2021, January 30). *What is Redux?* Retrieved from https://medium.com/swlh/what-is-redux-b16b42b33820

GeeksforGeeks. (2023, April 24). *Levels of Data Flow Diagrams*. Retrieved from https://www.geeksforgeeks.org/levels-in-data-flow-diagrams-dfd/

GeeksforGeeks. (2024, April 15). *Express.js Tutorial*. Retrieved from https://www.geeksforgeeks.org/express-js/

Heli, V. (2019). *TOO GOOD TO BE TRUE, Misleading Advertising of Cosmetic Products.* Retrieved from https://www.theseus.fi/bitstream/handle/10024/266420/Vaisanen_Heli.pdf?sequence=2&isAllowed=y

Herich, D. (2020, April 2). *Beauty Gets Personal*. Retrieved from https://www.gcimagazine.com/brands-products/skin-care/article/21849004/beauty-gets-personalpart-1-of-3-skin-care

Katalon. (2024). *What is Software Testing? Definition, Types, and Tools*. Retrieved from Katalon: https://katalon.com/resources-center/blog/software-testing

Legislative Requirements. (2023). Retrieved from GenieAI: https://www.genieai.co/define/legislative-requirements#:~:text=Legislative%20Requirement%20means%20a%20legal%20requirement%20imposed%20under%20any%20existing,requirement%20of%20a%20government%20entity.

LucidChart. (2024). *Data Flow Diagrams*. Retrieved from https://www.lucidchart.com/pages/data-flow-diagram

M., M. (2024, February 07). *What Makes a Good Website? 15 Characteristics to Stand Out.* Retrieved from https://www.hostinger.com/tutorials/what-makes-a-good-website?ppc_campaign=google_search_generic_hosting_all&bidkw=defaultkeyword&lo=9069992&gad_source=1&gclid=CjwKCAjw9IayBhBJEiwAVuc3fnojVW5BDD4b5pftWdM4hlAokEzAAJyEqgQ6MZa0aUBOkSSVaV3lERoC5scQAvD_BwE

Mehta, J. (2023, November 18). *The benefits of a user-friendly website*. Retrieved from https://abmatic.ai/blog/benefits-of-user-friendly-website

MIRO. (n.d.). *UML Diagrams*. Retrieved from https://miro.com/diagramming/what-is-a-uml-diagram/

NestJs. (n.d.). *Prisma*. Retrieved from https://docs.nestjs.com/recipes/prisma

Panfili, D. (2020, October 7). *The Importance of Having a User-Friendly Website*. Retrieved from https://zerogravitymarketing.com/blog/importance-of-user-friendly-web-design/

S., D. R. (2023, September 9). *BeautyeCommerce Growth Trends in 2023*. Retrieved from Linkedin: https://www.linkedin.com/pulse/beauty-ecommerce-growth-trends-2023-dr-rupal-shah-agarwal/

Semah, B. (2022, December 5). *What Exactly is Node.js? Explained for Beginners*. Retrieved from https://www.freecodecamp.org/news/what-is-node-js/

Wikipedia. (2024, April 23). *Bootstrap (Front-End Framework)*. Retrieved from https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)

Wikipedia. (2024, May 15). *MySQL*. Retrieved from https://en.wikipedia.org/wiki/MySQL

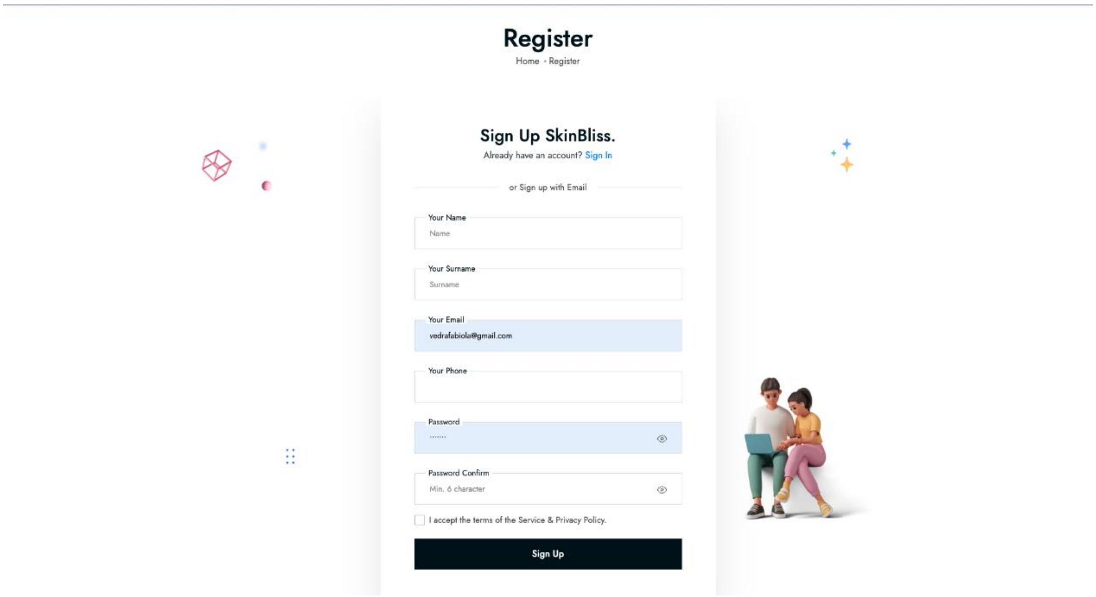
Wikipedia. (2024, April 16). *Next.Js*. Retrieved from https://en.wikipedia.org/wiki/Next.js

Wikipedia. (2024, May 12). *React (JavaScript Library)*. Retrieved from https://en.wikipedia.org/wiki/React_(JavaScript_library)

Zulfiqar, P. (2024). *What is the event-driven, nonblocking I/O model in Node JS?* Retrieved from Educative: https://www.educative.io/answers/what-is-the-event-driven-non-blocking-i-o-model-in-node-js

APPENDICES

APPENDIX A: System Screenshots



Register

Home • Register

Sign Up SkinBliss.

Already have an account? Sign In

or Sign up with Email

Your Name: Name

Your Surname: Surname

Your Email: vedrafabiola@gmail.com

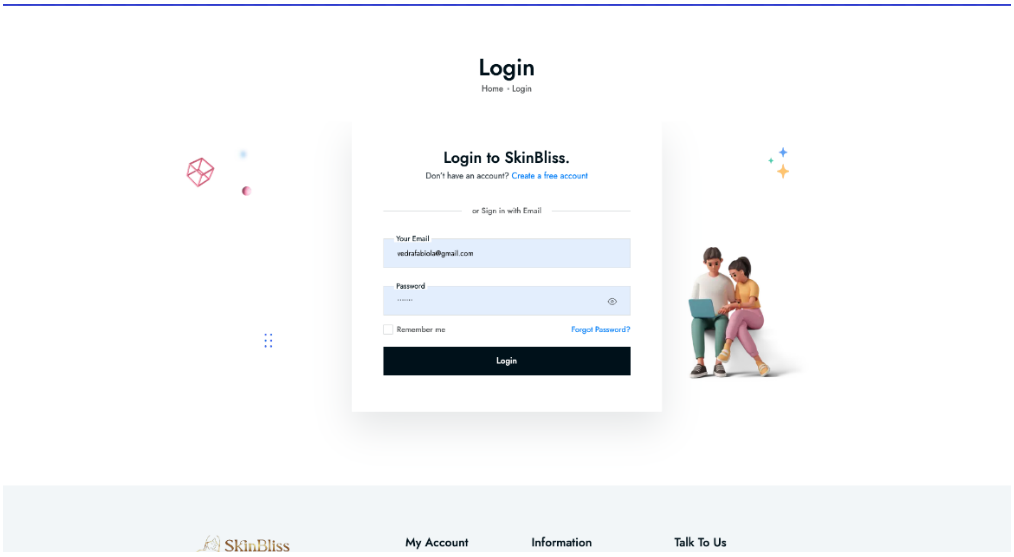
Your Phone:

Password: ·······

Password Confirm: Min. 6 character

□ I accept the terms of the Service & Privacy Policy.

Figure 28 Sign Up Process



Login

Home · Login

Login to SkinBliss.

Don't have an account? Create a free account

or Sign in with Email

Your Email: vedrafabiola@gmail.com

Password: ·······

□ Remember me

Figure 29 LogIn Process

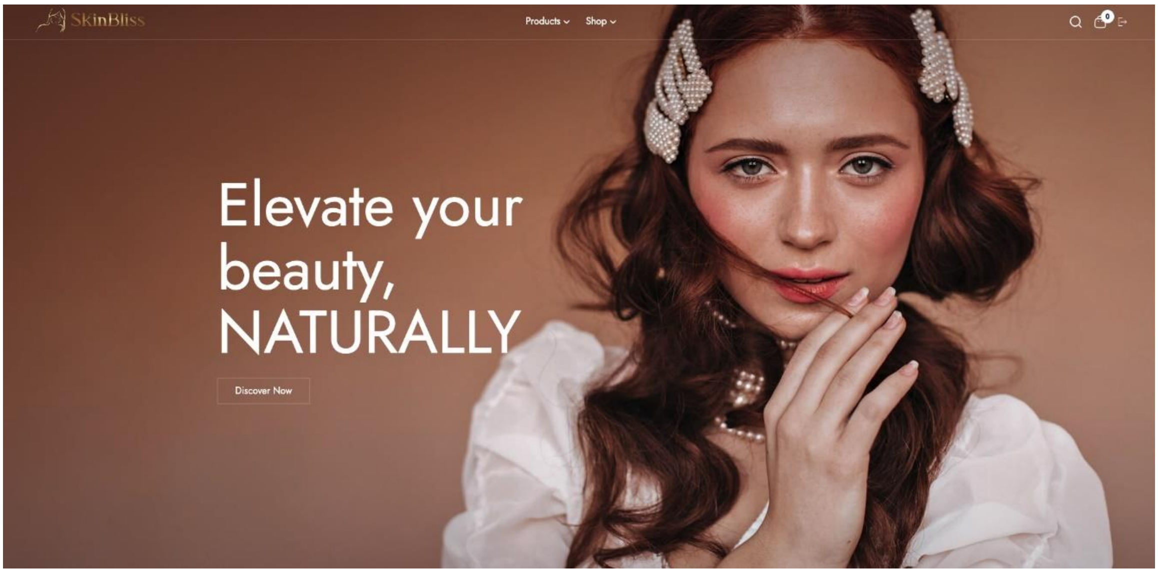
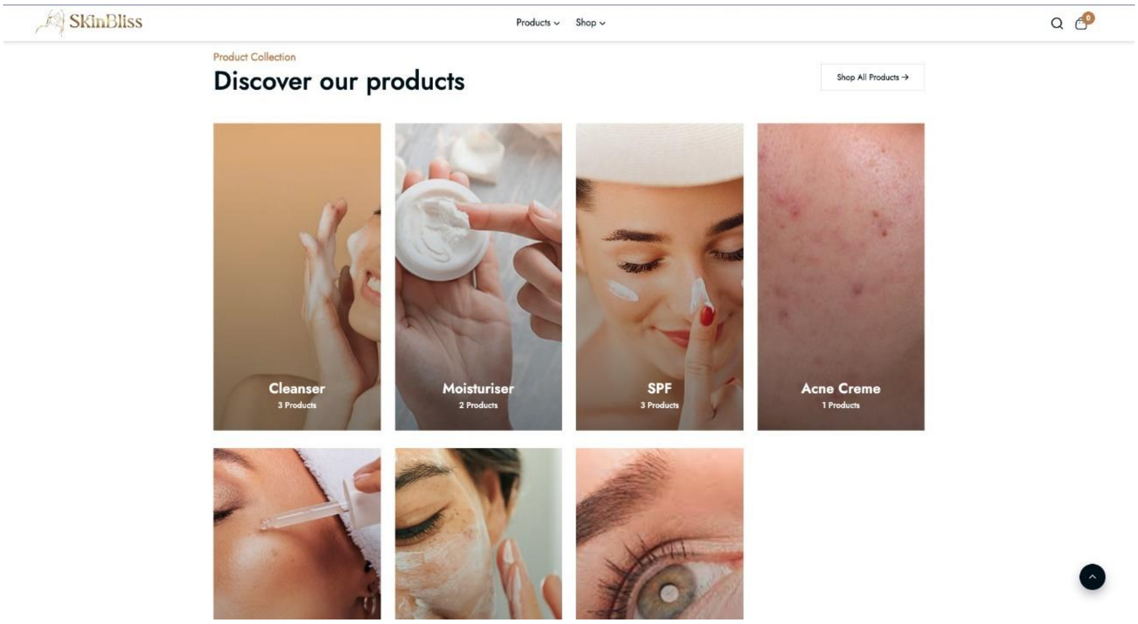


Figure 30 Applications Home Page



Product Collection

Discover our products

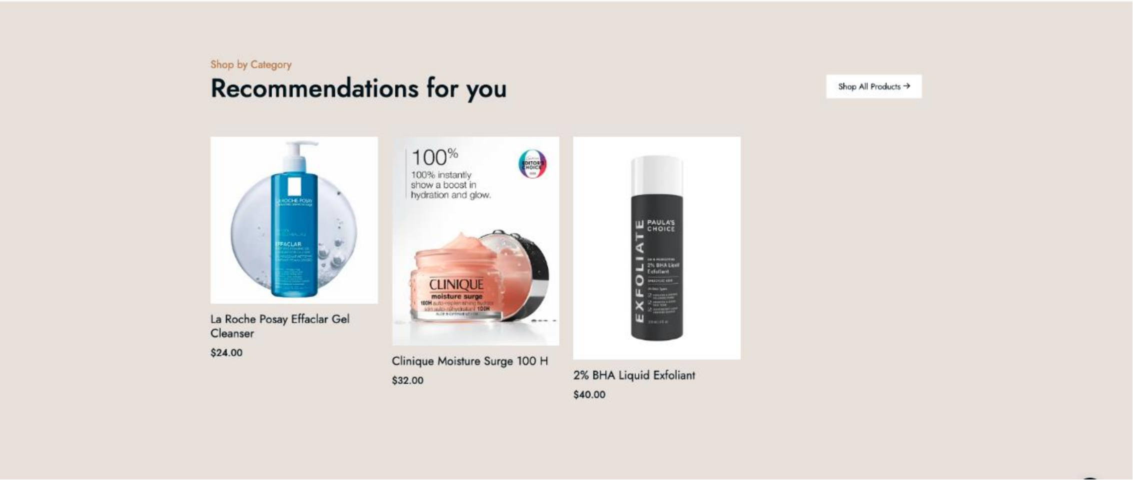
Cleanser
3 Products

Moisturiser
2 Products

SPF
3 Products

Acne Creme
1 Products

Figure 31 Applications Home Page2



Shop by Category

Recommendations for you

La Roche Posay Effaclar Gel Cleanser
$24.00

Clinique Moisture Surge 100 H
$32.00

2% BHA Liquid Exfoliant
$40.00

Figure 32 Product Recommendation after SkinTest

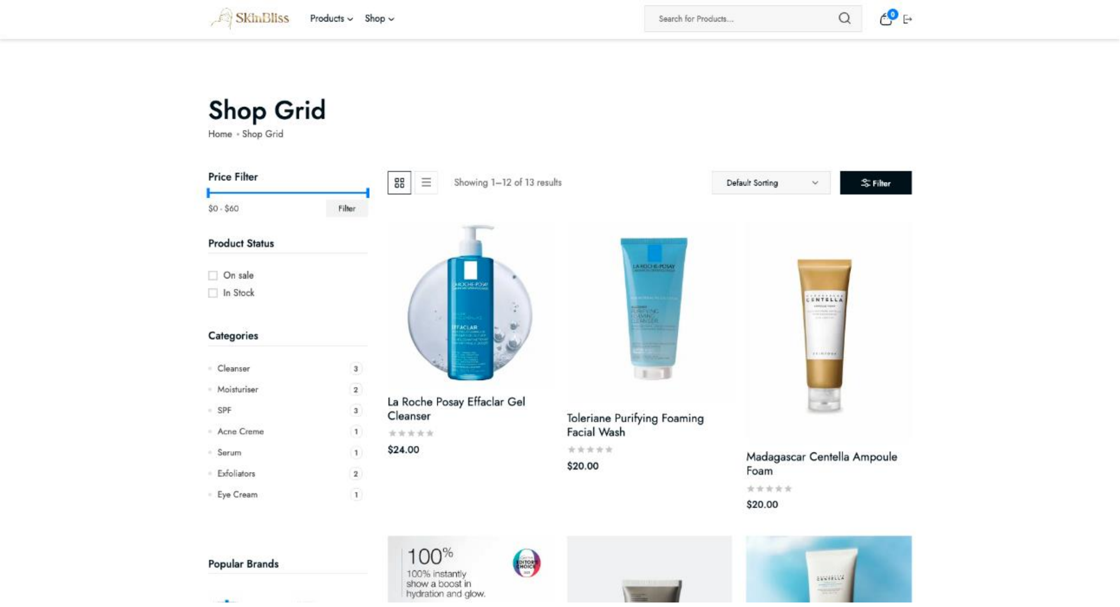


Figure 33 Product Browsing

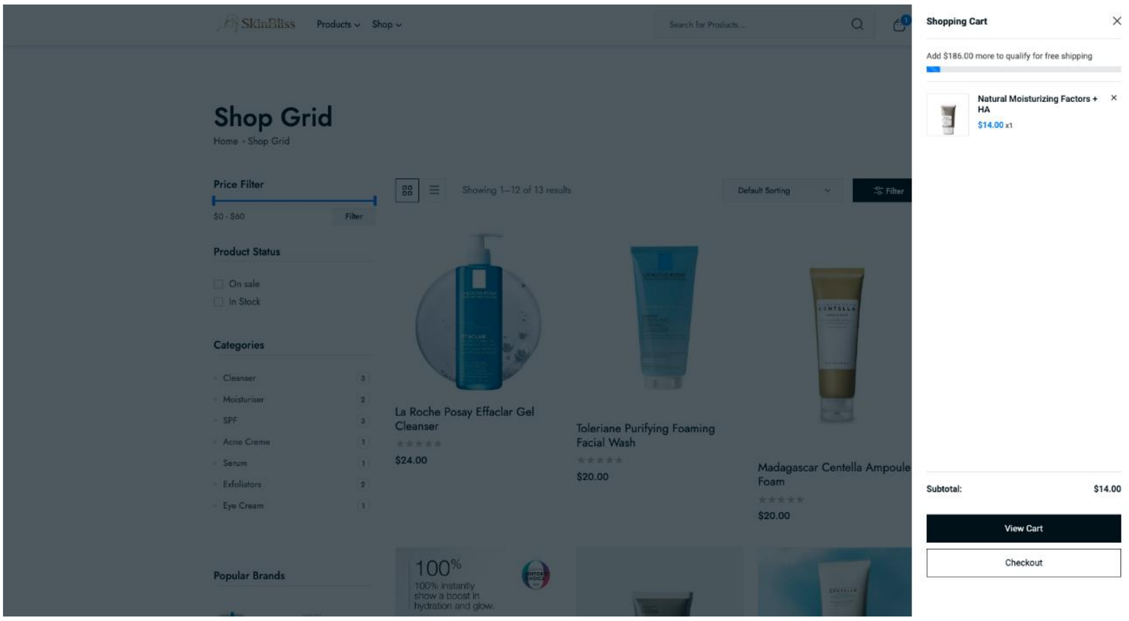
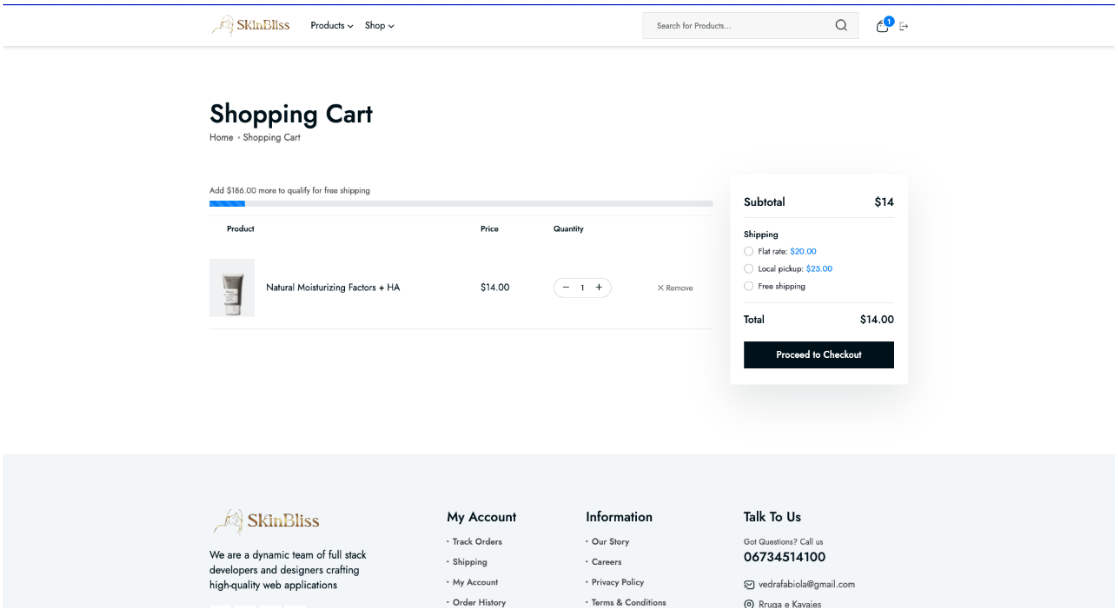


Figure 34 Product Add to Shopping Cart



Shopping Cart

Home · Shopping Cart

Add $186.00 more to qualify for free shipping

Product	Price	Quantity	
Natural Moisturizing Factors + HA	$14.00	− 1 +	✕ Remove

Subtotal	$14
Shipping ○ Flat rate: $20.00 ○ Local pickup: $25.00 ○ Free shipping	
Total	$14.00

SkinBliss

We are a dynamic team of full stack developers and designers crafting high-quality web applications

My Account
- Track Orders
- Shipping
- My Account
- Order History

Information
- Our Story
- Careers
- Privacy Policy
- Terms & Conditions

Talk To Us

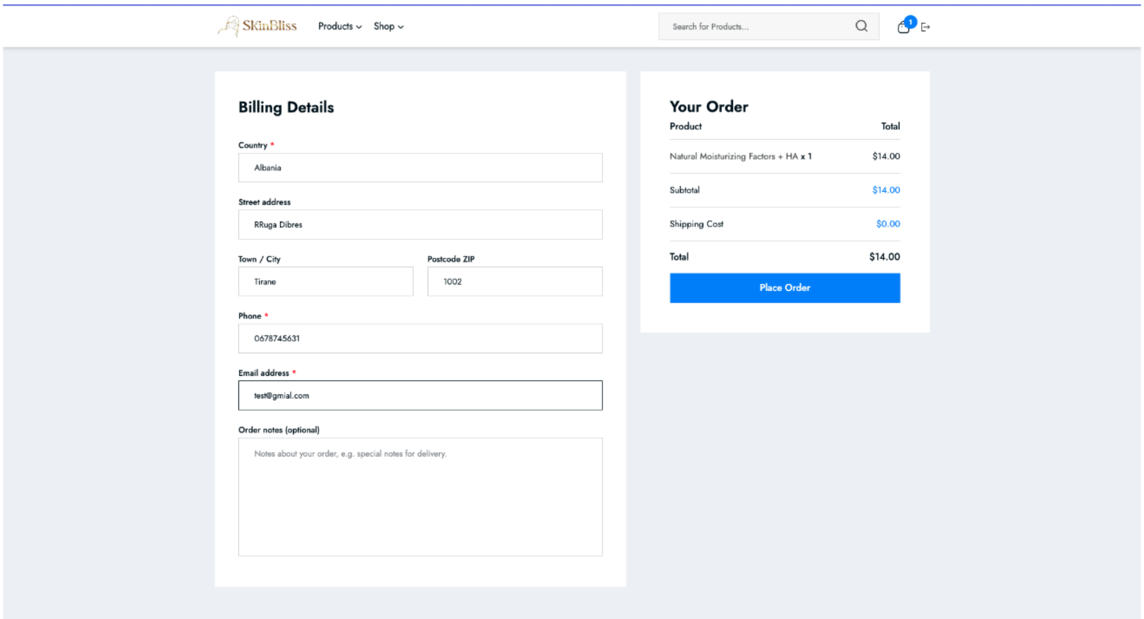
Got Questions? Call us

06734514100

✉ vedrafabiola@gmail.com

◎ Rruga e Kavajes

Figure 35 Shopping Cart Management



Billing Details

Country *
Albania

Street address
RRuga Dibres

Town / City
Tirane

Postcode ZIP
1002

Phone *
0678745631

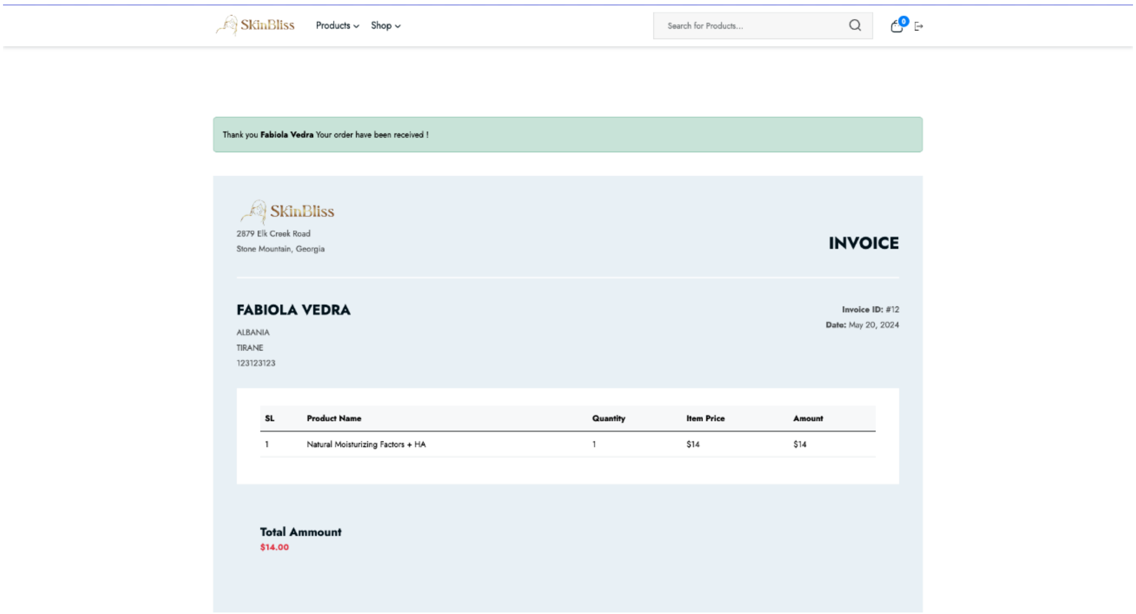
Email address *
test@gmial.com

Order notes (optional)
Notes about your order, e.g. special notes for delivery.

Your Order

Product	Total
Natural Moisturizing Factors + HA **x 1**	$14.00
Subtotal	$14.00
Shipping Cost	$0.00
Total	$14.00

Figure 36 Order Creation



Thank you **Fabiola Vedra** Your order have been received !

SkinBliss
2879 Elk Creek Road
Stone Mountain, Georgia

INVOICE

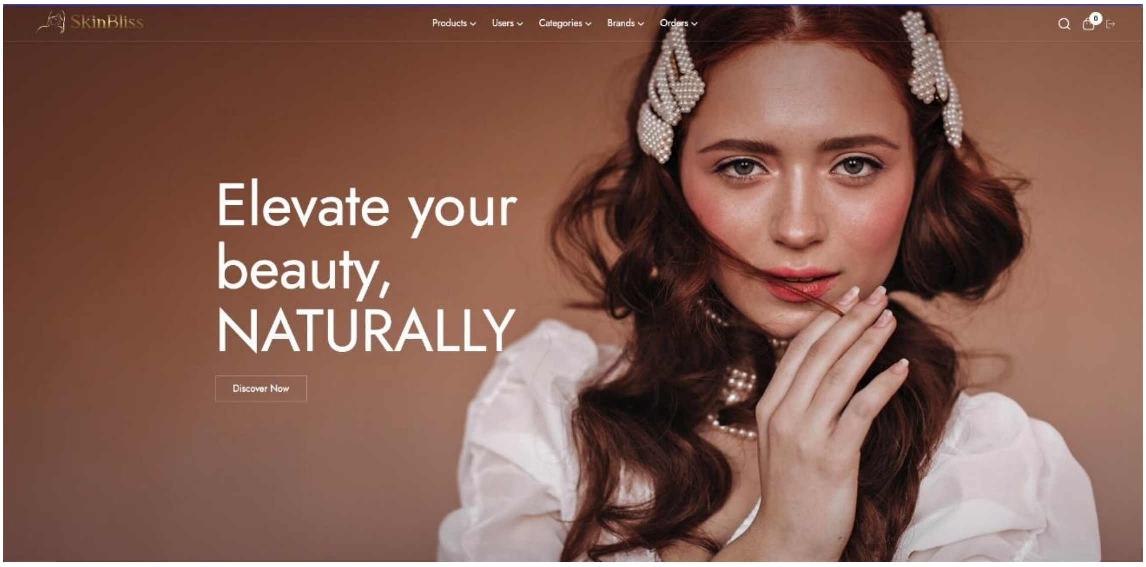
FABIOLA VEDRA
ALBANIA
TIRANE
123123123

Invoice ID: #12
Date: May 20, 2024

SL	Product Name	Quantity	Item Price	Amount
1	Natural Moisturizing Factors + HA	1	$14	$14

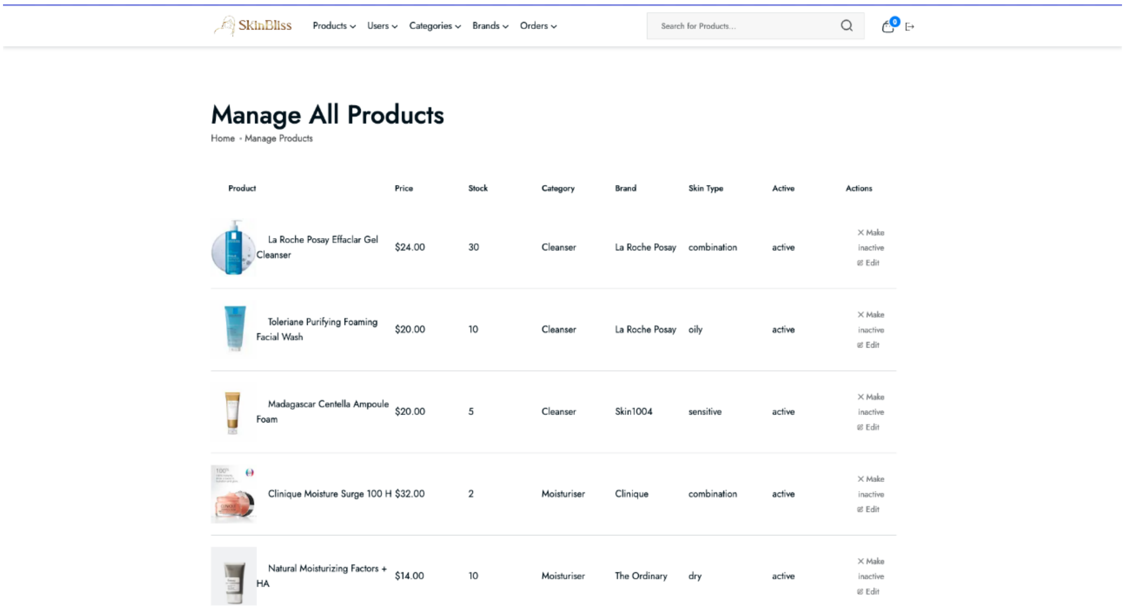
Total Ammount
$14.00

Figure 37 Order Receipt



Elevate your beauty, NATURALLY

Figure 38 Homepage for Admin Login

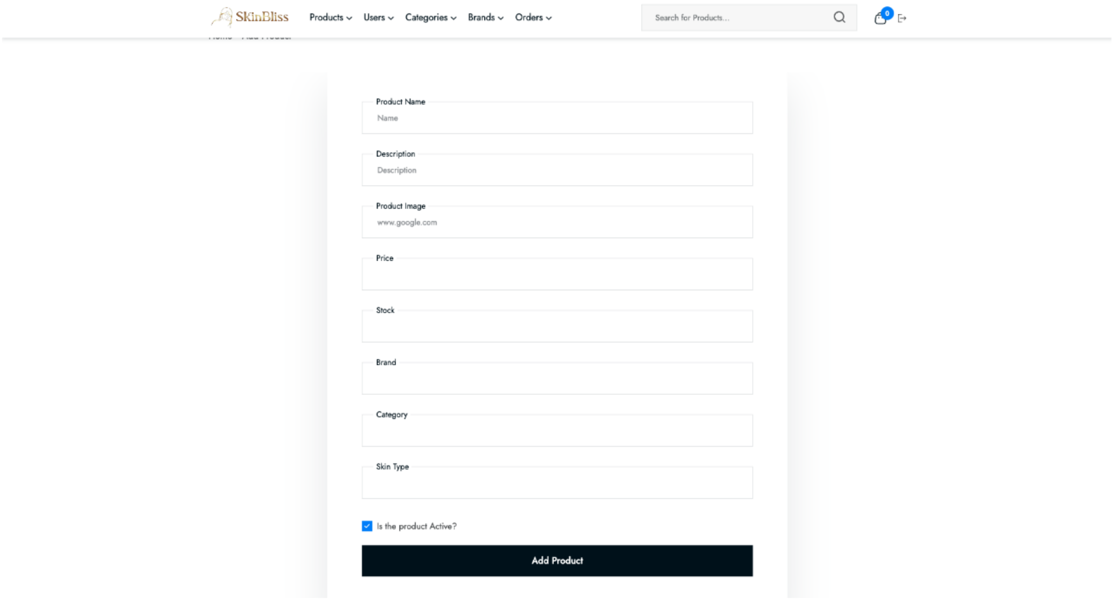


Manage All Products

Home · Manage Products

Product	Price	Stock	Category	Brand	Skin Type	Active	Actions
La Roche Posay Effaclar Gel Cleanser	$24.00	30	Cleanser	La Roche Posay	combination	active	✕ Make inactive Edit
Toleriane Purifying Foaming Facial Wash	$20.00	10	Cleanser	La Roche Posay	oily	active	✕ Make inactive Edit
Madagascar Centella Ampoule Foam	$20.00	5	Cleanser	Skin1004	sensitive	active	✕ Make inactive Edit
Clinique Moisture Surge 100 H	$32.00	2	Moisturiser	Clinique	combination	active	✕ Make inactive Edit
Natural Moisturizing Factors + HA	$14.00	10	Moisturiser	The Ordinary	dry	active	✕ Make inactive Edit

Figure 39 Product Management



Product Name: Name

Description: Description

Product Image: www.google.com

Price:

Stock:

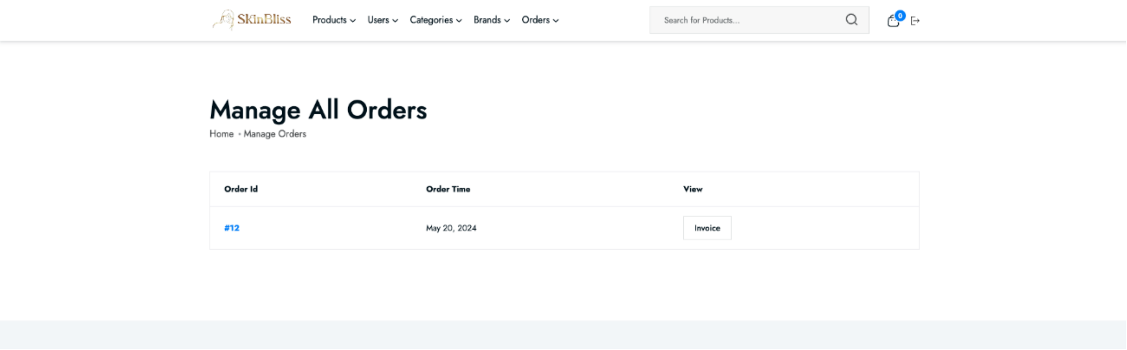
Brand:

Category:

Skin Type:

☑ Is the product Active?

Figure 40 New Product Addition from Admin

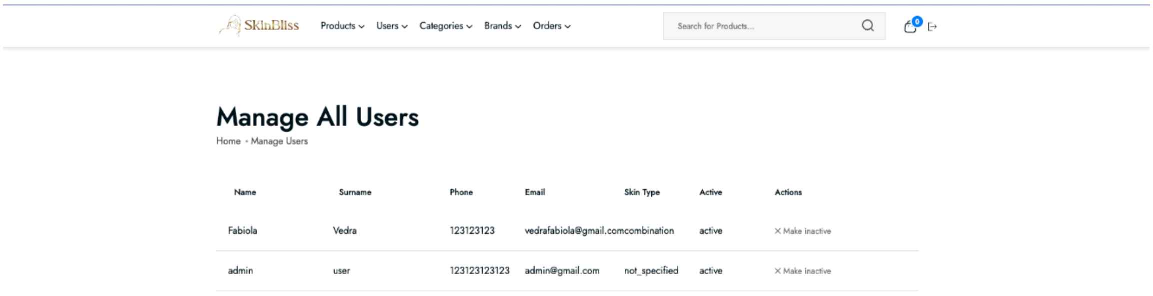


Manage All Orders

Home • Manage Orders

Order Id	Order Time	View
#12	May 20, 2024	Invoice

Figure 41 Order Management from Admin



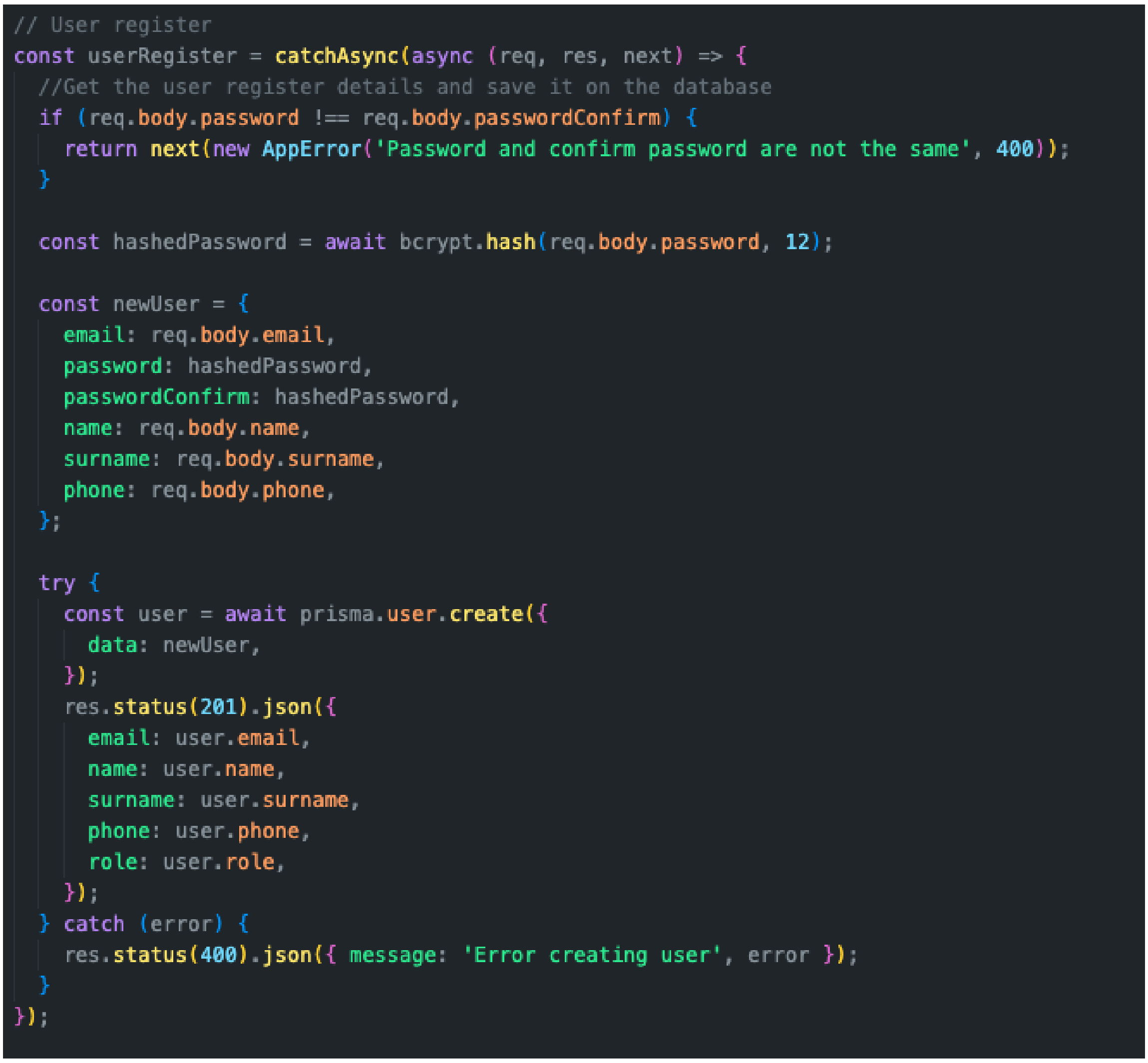
Manage All Users

Home • Manage Users

Name	Surname	Phone	Email	Skin Type	Active	Actions
Fabiola	Vedra	123123123	vedrafabiola@gmail.com	combination	active	✕ Make inactive
admin	user	123123123123	admin@gmail.com	not_specified	active	✕ Make inactive

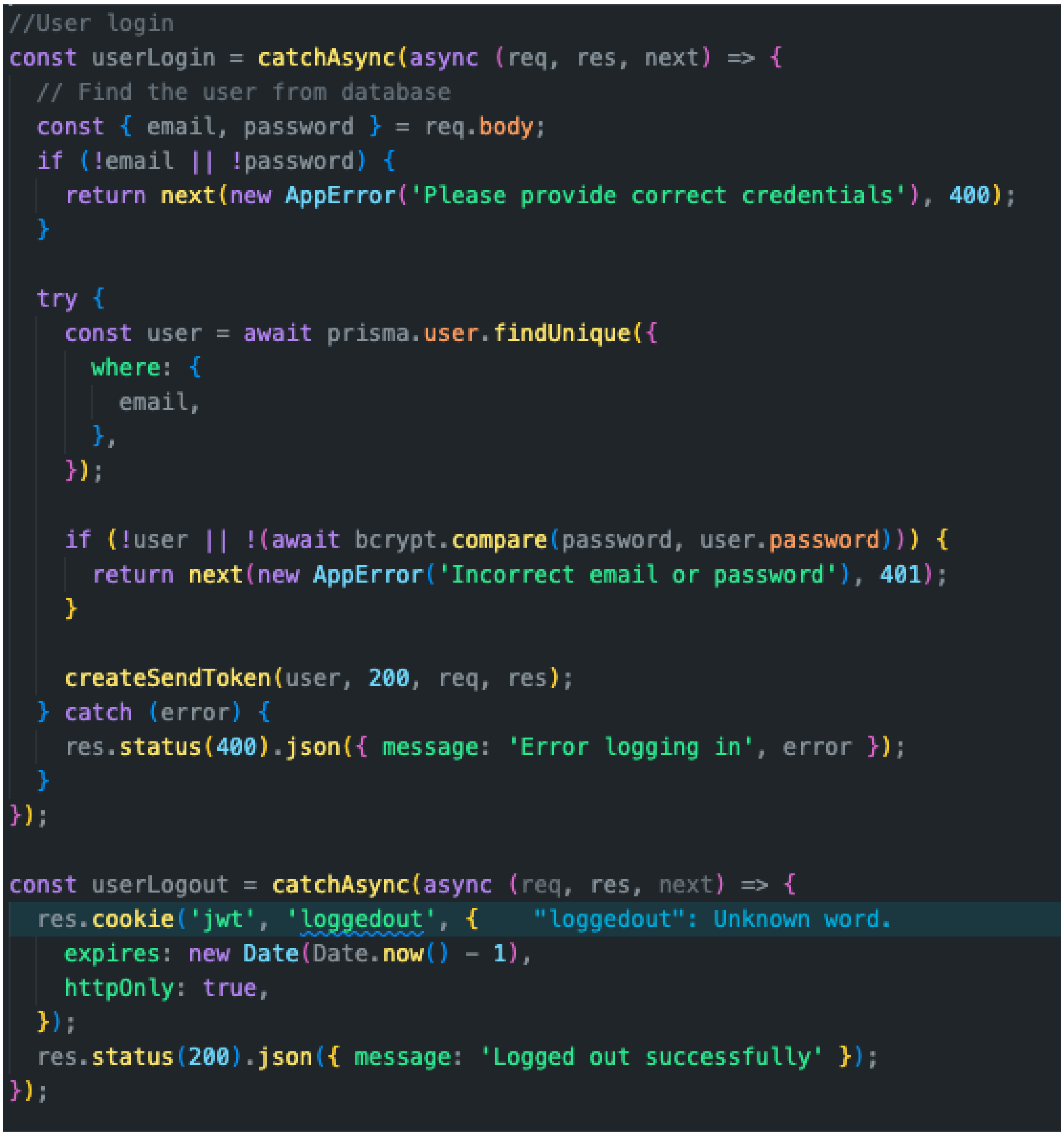
Figure 42 User Management from Admin

APPENDIX B: Important Code Pieces



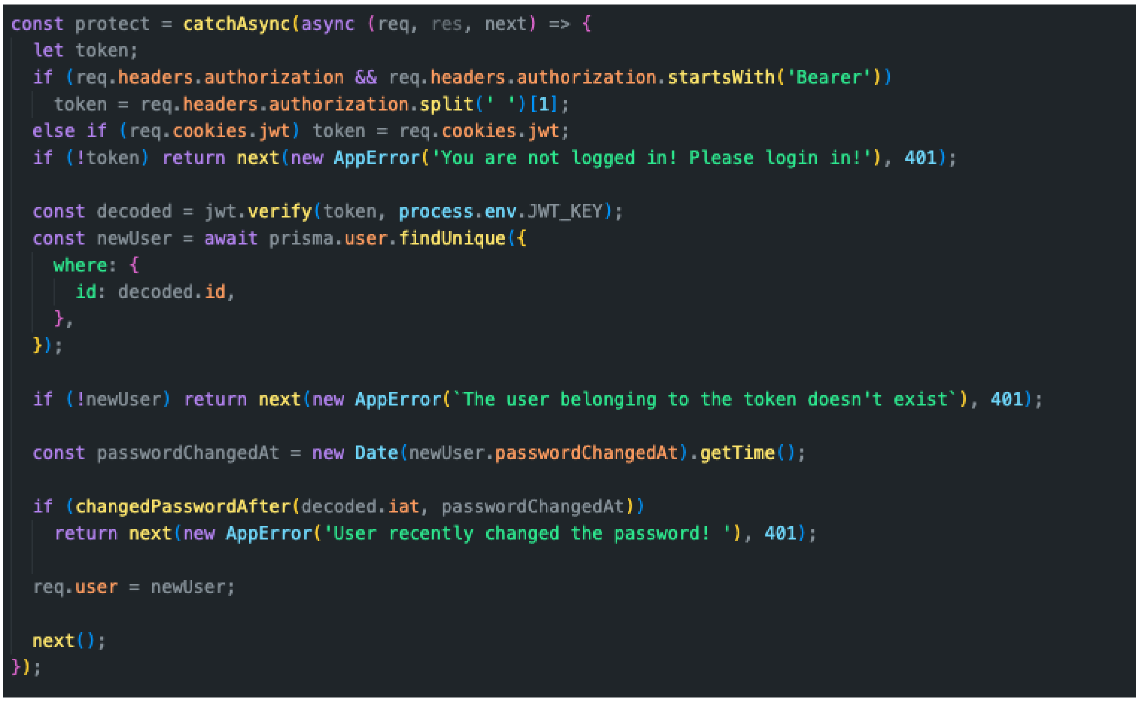
```
// User register
const userRegister = catchAsync(async (req, res, next) => {
  //Get the user register details and save it on the database
  if (req.body.password !== req.body.passwordConfirm) {
    return next(new AppError('Password and confirm password are not the same', 400));
  }

  const hashedPassword = await bcrypt.hash(req.body.password, 12);

  const newUser = {
    email: req.body.email,
    password: hashedPassword,
    passwordConfirm: hashedPassword,
    name: req.body.name,
    surname: req.body.surname,
    phone: req.body.phone,
  };

  try {
    const user = await prisma.user.create({
      data: newUser,
    });
    res.status(201).json({
      email: user.email,
      name: user.name,
      surname: user.surname,
      phone: user.phone,
      role: user.role,
    });
  } catch (error) {
    res.status(400).json({ message: 'Error creating user', error });
  }
});
```

Figure 43 User Register



```
//User login
const userLogin = catchAsync(async (req, res, next) => {
  // Find the user from database
  const { email, password } = req.body;
  if (!email || !password) {
    return next(new AppError('Please provide correct credentials'), 400);
  }

  try {
    const user = await prisma.user.findUnique({
      where: {
        email,
      },
    });

    if (!user || !(await bcrypt.compare(password, user.password))) {
      return next(new AppError('Incorrect email or password'), 401);
    }

    createSendToken(user, 200, req, res);
  } catch (error) {
    res.status(400).json({ message: 'Error logging in', error });
  }
});

const userLogout = catchAsync(async (req, res, next) => {
  res.cookie('jwt', 'loggedout', {    "loggedout": Unknown word.
    expires: new Date(Date.now() - 1),
    httpOnly: true,
  });
  res.status(200).json({ message: 'Logged out successfully' });
});
```

Figure 44 User Login



```
const protect = catchAsync(async (req, res, next) => {
  let token;
  if (req.headers.authorization && req.headers.authorization.startsWith('Bearer'))
    token = req.headers.authorization.split(' ')[1];
  else if (req.cookies.jwt) token = req.cookies.jwt;
  if (!token) return next(new AppError('You are not logged in! Please login in!'), 401);

  const decoded = jwt.verify(token, process.env.JWT_KEY);
  const newUser = await prisma.user.findUnique({
    where: {
      id: decoded.id,
    },
  });

  if (!newUser) return next(new AppError(`The user belonging to the token doesn't exist`), 401);

  const passwordChangedAt = new Date(newUser.passwordChangedAt).getTime();

  if (changedPasswordAfter(decoded.iat, passwordChangedAt))
    return next(new AppError('User recently changed the password! '), 401);

  req.user = newUser;

  next();
});
```

Figure 45 Aurthorization Middleware