

Software Design Description Document for "Stars Accessories" E-Commerce Web-Application

Ahmed Hany, Adham Samir, Amer Mohamed, Mostafa Ahmed
Supervised by: Dr. Essam Eliwa, Eng. Nada Ayman

June 19, 2021

Table 1: Document version history

Version	Date	Reason for Change
1.0	7-May-2021	SDD first version's description are defined.
1.1	13-May-2021	Added Sequence Diagram.
1.2	7-June-2021	Database Schema updated.
1.3	13-June-2021	Requirement Matrix updated.
1.4	18-June-2021	Appendices updated.

GitHub: [Stars Accessories Web-Application](#)

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
1.3	Intended audience	3
1.4	Reference Material	3
1.5	Definitions and Acronyms	4
2	System Overview	4
2.1	System Scope	5
2.2	System objectives	5
2.3	System Timeline	5
3	Design viewpoints	7
3.1	Context viewpoint	7
3.2	Composition viewpoint	9
3.2.1	Design Rationale	10
3.3	Logical viewpoint	11
3.4	Algorithm viewpoint	14
3.5	Interaction viewpoint	16
3.6	Interface viewpoint	16
4	Data Design	17
4.1	Data Description	17
4.2	Database design description	17
5	Human Interface Design	18
5.1	Screen Images	18
6	Requirements Matrix	24
7	APPENDICES	25
7.1	Github	25
7.2	Other appendices as appropriate	27

Abstract

In today's fast-paced market world, it's critical to react to client demands in the most efficient and timely way possible, while still maintaining a well-organized structure. Dealing with customers through social media is a true hustle. With no database management, a straightforward way to communicate with customers, and a navigated system that helps them reach their objective, customers don't feel comfortable nor satisfied. We aim to assist "Stars accessories" customers through their shopping journey by having an easy-to-use and practical web application. Furthermore, the simplicity of accessing the inventory to Administrators and Managers and assist them in tracking placed orders and following the process. We aim to follow the Agile methodology of development as planning is incremental. It is easier to adapt the strategy to represent changing consumer needs. The proposed solution is a web application that will ease both customers and administrators buying/selling processes.

1 Introduction

1.1 Purpose

The purpose of the software design document for "Stars Accessories E-Commerce Web-Application" is to aid in the software development process by providing an outline of the software design and specification by which it describes the system architecture, system components, and software requirements as agreed upon by the customer and the project team. [1]

1.2 Scope

This software design description for "Stars Accessories E-Commerce Web-Application" describes system design and architecture thus provides the main design viewpoints and perspective of the system to communicate to key design stakeholders. This SDD document is used to record design resolution and their reasoning in order to avoid disputes over what was previously agreed upon. This SDD shows how the software system will be designed to complement and fulfill the requirements. The software design description for the project would define the overall system architecture, System overview, Design viewpoints, Data design and human interface design [2]

1.3 Intended audience

Project team, Stars Accessories stakeholders, and the supervising doctors. The document use is intended for the use and editing in collaboration by those mentioned solely.

1.4 Reference Material

Software Requirement Specification Document for "Stars Accessorises" E-Commerce Web-Application and Testing plan.

1.5 Definitions and Acronyms

Provide definitions of all terms, acronyms, and abbreviations that might exist to properly interpret the SDD. These definitions should be items used in the SDD that are most likely not known to the audience.

Term	Definition
Software Design Document (SDD)	Used as the primary medium for communicating software design information.
Design Entity	An element of a design that is structurally and functionally distinct from other elements.
SA	Stars Accessories.
Ecommerce	Electronic Commerce.
CRUD	Create, Retrieve, Update, Delete.
GUI	Graphical User Interface.

2 System Overview

A web-based application that aims to facilitate the sales and purchases of our independent growing business “Stars Accessories” that is in the market of Women’s wearings. We aim to a system that encompasses and applies the vision that the “Stars Accessories” brand hold for the future which is to expand, promote and provide a next-level user experience by having a well-integrated system as per figure 1:

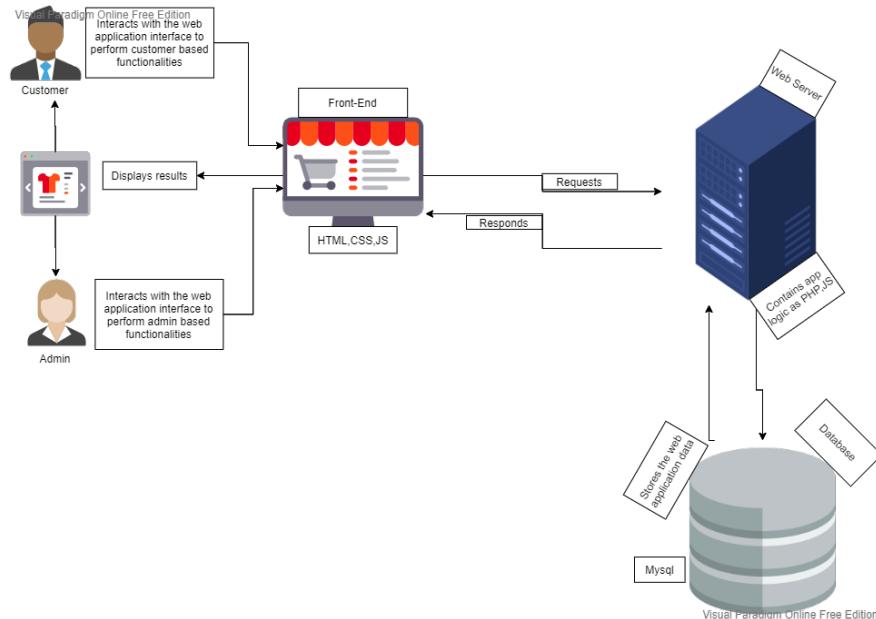


Figure 1: SA System Overview

2.1 System Scope

The intended software system to be produced is an online E-commerce web application. The availability of the system is intended to cover all the interested online shoppers in Egypt, as the system is a cross-platform. The system shall be fully-controlled by the Admins, allowing them to add, delete, and edit products according to their current inventory. The customers shall browse the web-application, add their desired products in the shopping cart, check out and start checking the order status up till the delivery time. The overall scope is implementing an accessible, convenient means of communication that will be able to fulfill Stars Accessories growing reach, as Instagram is the main selling platform for the brand, the communication between the client and the customers occur solely through direct messages, our goal is to elevate that level of service that requests and purchases would be stored in the store's own database.

2.2 System objectives

- Implementing an accessible, convenient means of communication that will be able to fulfill Stars Accessories growing reach, as Instagram being the main selling platform for the brand, the communication between the client and the customers occurs solely through direct messages, our goal is to elevate that level of service that requests and purchases would be stored in the store's own database.
- Achieving a manageable database will consequently lead us to be able to create a more convenient user experience represented in creating user profiles where the user can have more access and ability to browse the products and add it to his cart.
- Achieving a manageable database will consequently lead us to be able to create a more convenient user experience represented in creating user profiles where the user can have more access and the ability to browse the products and add them to his cart.
- As for now, Stars accessories brand has roughly a total of 28 thousand followers, our goal is to expand this number by 20%.
- Reducing administrative costs by 50%.
- Increasing the number of loyal customers.
- Improving customer service.
- Increasing clients' satisfaction.
- To build a usable system that achieve a score equal to or higher than 70 in System Usability Scale (SUS) **brooke1996sus**.

2.3 System Timeline

Tasks will be assigned to team members by their initials.

Table 2: Project name time plan

ID	Task	Start Date	Number of Hours	Team Member
1	Requirements Gathering	1/4/2021	18	Everyone
2	Client Interview	4/4/2021	2	AH
3	Proposal Document	5/4/2021	72	Everyone
4	Requirement analysis	14/4/2021	12	Everyone
5	Initial SRS	16/4/2021	56	Everyone
6	Proposal feedback and review	28/4/2021	6	AS,AH
7	Web Application Wireframes	30/4/2021	24	AH
8	Front-End Implementation	4/5/2021	56	Everyone
9	Updating SRS	6/5/2021	6	Everyone
10	Front-End Validation and CRUD operations	7/5/2021	28	Everyone
11	Phase 1 Discussion	10/5/2021	1	Everyone
12	Simple Implementation	12/5/2021	124	AH,AM
13	Scenario Analysis	22/5/2021	48	AS,MA
14	SDD and Testing Documents	26/5/2021	130	Everyone
15	Full Implementation	6/6/2021	248	Everyone
16	Final SRS and SDD	19/6/2021	72	Everyone
17	Final Discussion	21/6/2021	1	Everyone

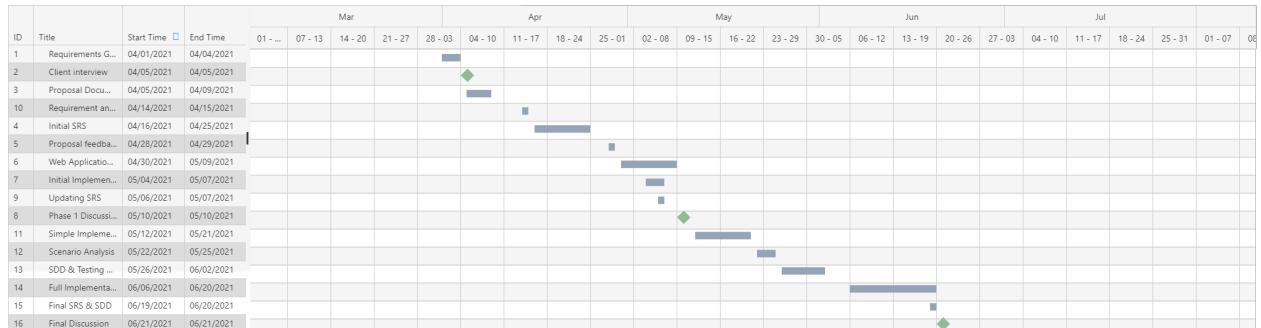


Figure 2: SA GANTT Chart

3 Design viewpoints

3.1 Context viewpoint

SA System shall evolve around 3 parties: Customers, Admins, Shipping company, and a database interacting with the system to connect those 3 parties together, as per Figure 3.

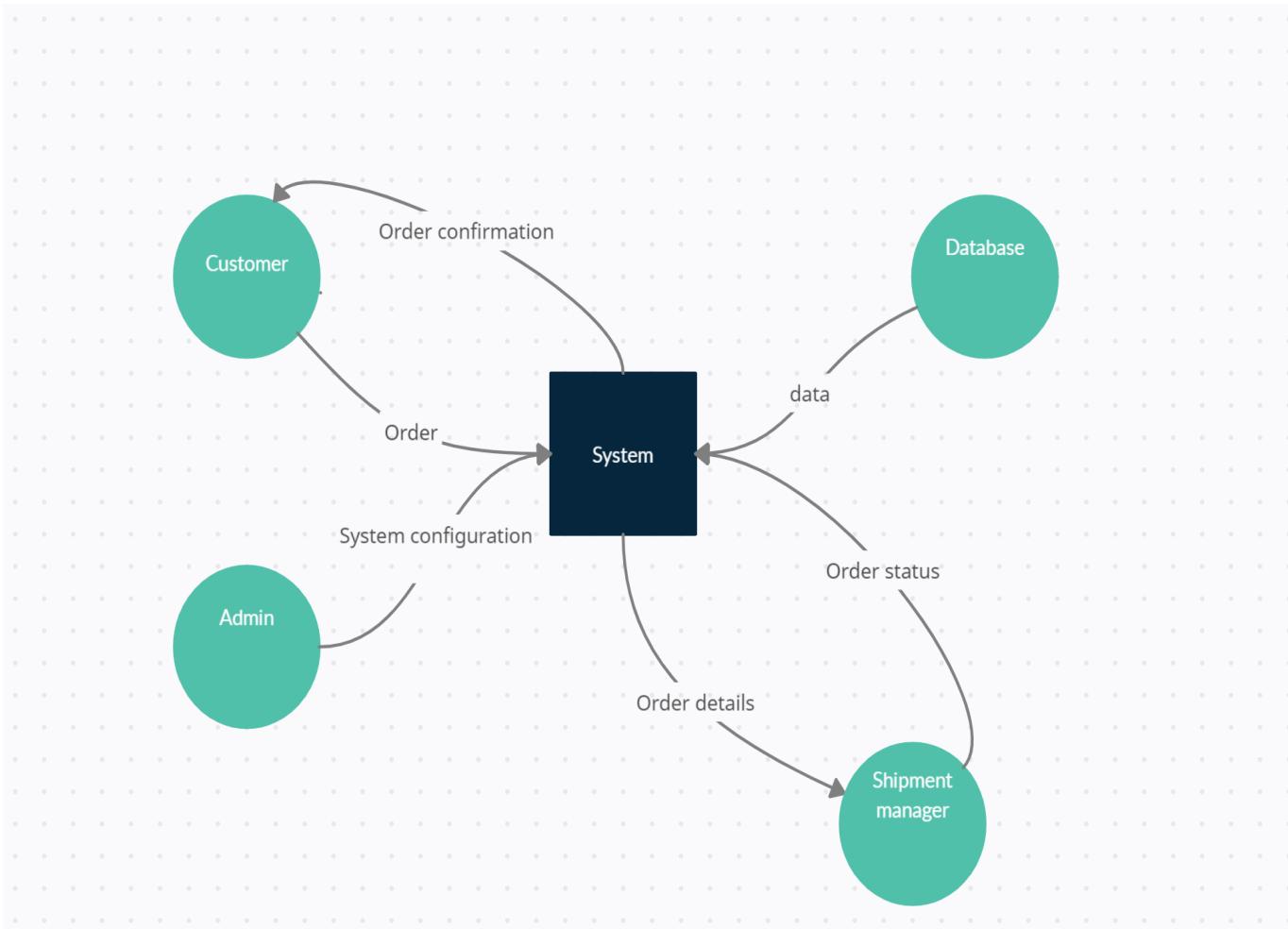


Figure 3: SA System Context

The following is a Use Case diagram that describes the functionalities of the involved parties in SA, as per Figure 4.

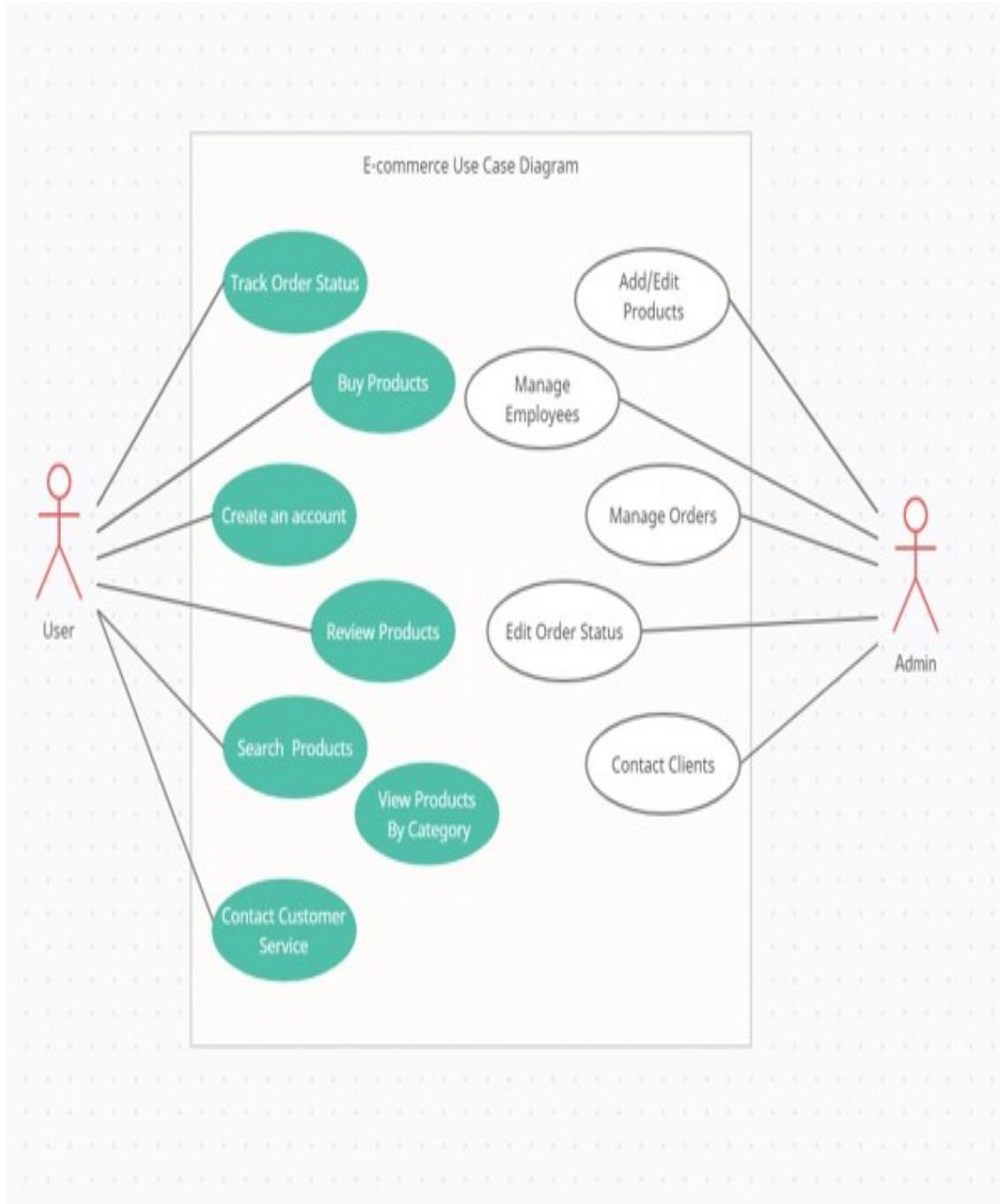


Figure 4: SA Use Case

3.2 Composition viewpoint

MVC pattern functions in way that it takes advantage of the way it is structured “A three interconnected parts called the model, the view, and the controller.” Where these constituents are managed to build and develop aspects of any web-application that makes it’s development process more rapid, changeable and reusable. [3]

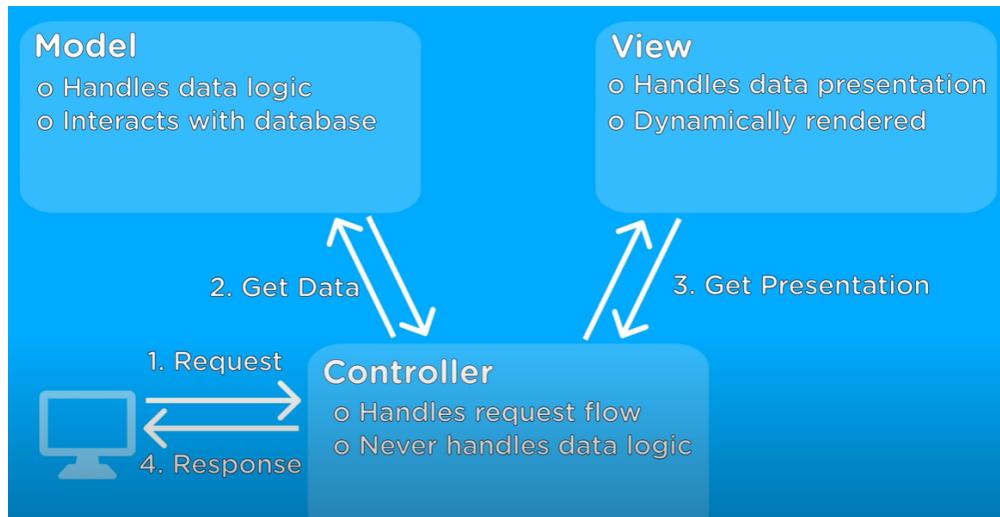


Figure 5: MVC design pattern

Model:

- Interprets the data logic based on the request sent from the controller.
- Interacts with the database.
- Responsible for all the validation, saving, adding, deleting, updating of the data.
- After interpreting the data, sends it back to the controller.
- Never interacts with the view.

View:

- Handles the presentation of the data sent by the controller.
- Dynamically rendered.
- Sends its final presentation of the data back to the controller.
- Never interacts with the model.

Controller:

- Handles the entire request flow from the client.
- Acts as a middle man between the model and the view.
- Never interacts with the data logic.
- Sends the final presentation of the data to the user.

MVC Architecture Pattern

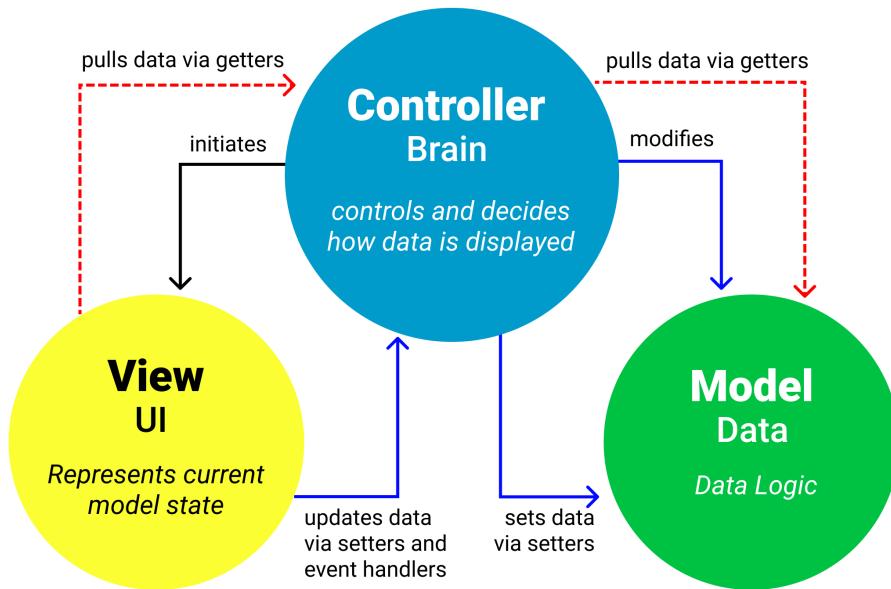


Figure 6: MVC design architecture

3.2.1 Design Rationale

The MVC architecture pattern is one of the most commonly used patterns in designing web-applications thus choosing it to be the pattern to base on it the project structure was a choice unaccompanied by much quarrels or debates for a variety of reasons; Our e-commerce web application is a complex one with lots of classes, methods, operations and the would be tedious task of handling them concurrently, so a pattern that relies on dissecting the system into sub-systems as MVC does by separating the presentation of the data and the logic of the data would make the development process much more less complex, accelerated and efficient.

3.3 Logical viewpoint

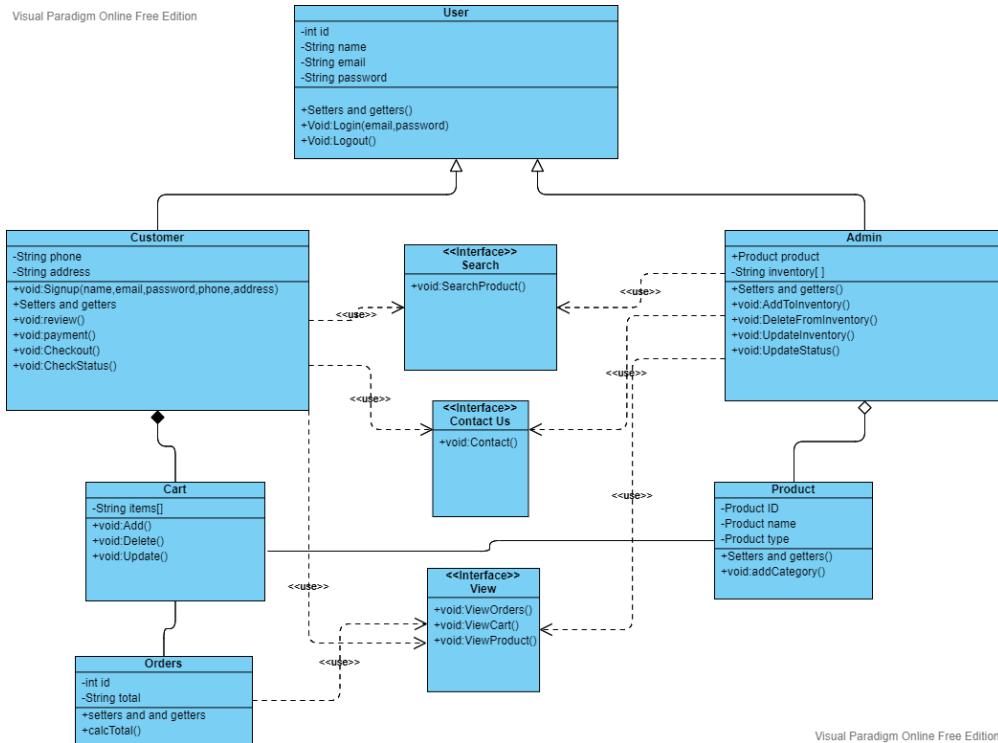


Figure 7: Class Diagram

Table 3: Class Name - User

Abstract or Concrete:	Concrete.
List of Superclasses	None.
List of Subclasses	Customer,Admin
Purpose	Represent the account
Collaborations	User class is parent class to both Customer and admin classes, in which both of these classes inherit the attributes and functions of it.
Attributes	id:int, name:string , phone number:string, email:string, password:string
Operations	Login>Email,Password):Void, Logout:Void
Constraints	must be logged in

Table 4: Class Name -Admin

Abstract or Concrete:	Concrete.
List of Superclasses	User.
List of Subclasses	None
Purpose	Supervise the inventory, communicate with the customer and handle complaints.
Collaborations	Admin class is a subclass from User class in which it has an aggregation relation between Product class. The aim of the class is to represent and achieve the administrator's functionality such as adding, deleting , updating products in the inventory and replying to customer's complaints.
Attributes	product:product , inventory:string[]
Operations	AddToInventory:void(), DeleteFromInventory:void() ,UpdateInventory:void(), UpdateStatus:Void() SearchProdut():void, ContactUs():void, ViewOrder():void, ViewCart():void, ViewProduct():void.
Constraints	must be logged in

Table 5: Class Name - Customer

Abstract or Concrete:	Concrete.
List of Superclasses	User.
List of Subclasses	None
Purpose	Make purchases and browse products.
Collaborations	Customer class is a subclass from User class in which it has a composition relation between Cart class. The aim of the class is to represent and achieve the customer's functionality such as sign up, login, search product and add product to cart, review and give ratings to products and check status of the product shipment.
Attributes	phone:string , address:string
Operations	Sign up(Name,Email,Password,Phone,Address):Void review():void, payment():void, Checkout():void, CheckStatus():void , SearchProdut():void, ContactUs():void, ViewOrder():void, ViewCart():void, ViewProduct():void.
Constraints	must be logged in

Table 6: Class Name - Cart

Abstract or Concrete:	Concrete.
List of Superclasses	None.
List of Sublasses	None.
Purpose	Store Customer's purchases.
Collaborations	Cart class acts as a composition class in customer's class, where each customer must have a cart where he can add,delete,update products from it.
Attributes	items:string[]
Operations	add:void(), delete:void(), update:void()
Constraints	must be logged in

Table 7: Class Name - Product

Abstract or Concrete:	Concrete.
List of Superclasses	None.
List of Sublasses	None
Purpose	Fulfills the product's details such as its ID, name, type.
Collaborations	Product class is an aggregation class from admin, where admin fulfills the product's details and labels each product under its appropriate category.
Attributes	Productid:String, ProductName:string , ProductType:string,
Operations	addCategory():string
Constraints	must be logged in

Table 8: Class Name - Order

Abstract or Concrete:	Concrete.
List of Superclasses	None.
List of Sublasses	None
Purpose	Displays the orders and calculates the total of the order.
Collaborations	Order class has an association relationship with the Cart class.
Attributes	id:int, total:string
Operations	CalcTotal()
Constraints	must be logged in

3.4 Algorithm viewpoint

Overview of code used in implementation

```
1  <?php
2
3  class user
4  {
5      public $username ;
6      public $password;
7      public $email;
8      public $phonenumber;
9      public $role;
10     public $conn;
11
12     function create_connection()
13     {
14         $this->conn = new mysqli("localhost", "root", "", "stars");
15         return $this->conn;
16     }
17 }
```

- User class that acts as a super class to both client and admin classes.
- Included within the class the (**create-connection**) function responsible for the connection to the database.

```
class client extends user
{
    function signin($phonenumber,$password)
    {
        $this->create_connection();
        $sql="SELECT * from accounts where phonenumber='$phonenumber' and password='$password'";
        $result=mysqli_query($this->conn,$sql);
        return $result;
    }

    function signup($username,$email,$password,$phonenumber)
    {
        $conn=$this->create_connection();
        $sql="SELECT * FROM accounts where email='$email'";
        $result=mysqli_query($conn,$sql);
        $num_rows = mysqli_num_rows($result);
        if($num_rows >= 1)
        {
            echo "<h2> class='incorrect'>Email already exists</h2>";
        }
        else
        {
            $role="";
            $sql="INSERT INTO accounts (username,email,password,phonenumber,role) values('$username','$email','$password',',$phonenumber' ,'$role')";
            $result = mysqli_query($conn,$sql);
            header("location:home.php");
            return $result;
        }
    }
}
```

- Client class is a sub-class from user class.
- Included within the class the (**Sign up**) function that validates whether a user exists or not and if not classifies him in the database with role=1 that means he is not an admin but a user.
- Included within the class the (**Sign in**) function where if the user has already signed up is presented with the parameters that are (*phone number, password*) that redirects him to the home page.

```

class admin extends user
{
    function signin($username,$password)
    {
        $this->create_connection();
        $sql="SELECT * from accounts where username='".$username and password='".$password."'";
        $result=mysqli_query($this->conn,$sql);
        return $result;
    }

    function insertproduct($name,$price,$image)
    {
        $this->create_connection();
        $sql="INSERT into products (name,price,image) values('".$this->name."','".$this->price."','".$this->image')";
        $result=mysqli_query($this->conn,$sql);
        $this->close_connection();
        return mysqli_insert_id($this->conn);
    }

    function deleteproduct($name)
    {
        $this->create_connection();
        $sql = "DELETE FROM products WHERE name = $name ";
        $result = mysqli_query($this->conn,$sql);
    }

    function editproducts($name)
    {
        $this->create_connection();
        $sql = "UPDATE FROM products WHERE name = $name values('".$this->name."','".$this->price."','".$this->image')";
        $result = mysqli_query($this->conn,$sql);
    }
}

```

- Admin class is a sub-class from user class.
- Included within the class the (**Sign in**) function where the admin enters his pre-given data (*username, password*) that redirects him to the admin page.
- Included within the class (**insertproduct**) function where the admin can access and add products to the inventory that would be stored in the **products** table in the database.
- Included within the class (**deleteproduct**) function where the admin can access and delete products from the inventory that are stored in the **products** table in the database.
- Included within the class (**editproduct**) function where the admin can access and edit products from the inventory that are stored in the **products** table in the database.

3.5 Interaction viewpoint

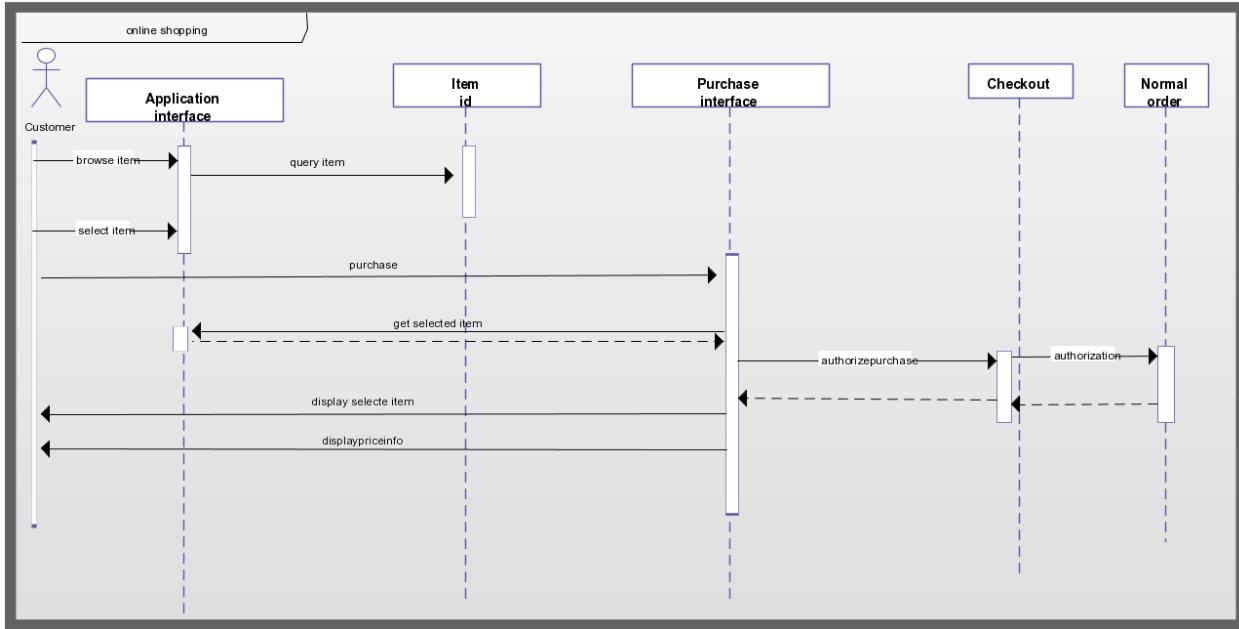


Figure 8: Sequence diagram of the hierarchy followed to purchase an item

3.6 Interface viewpoint

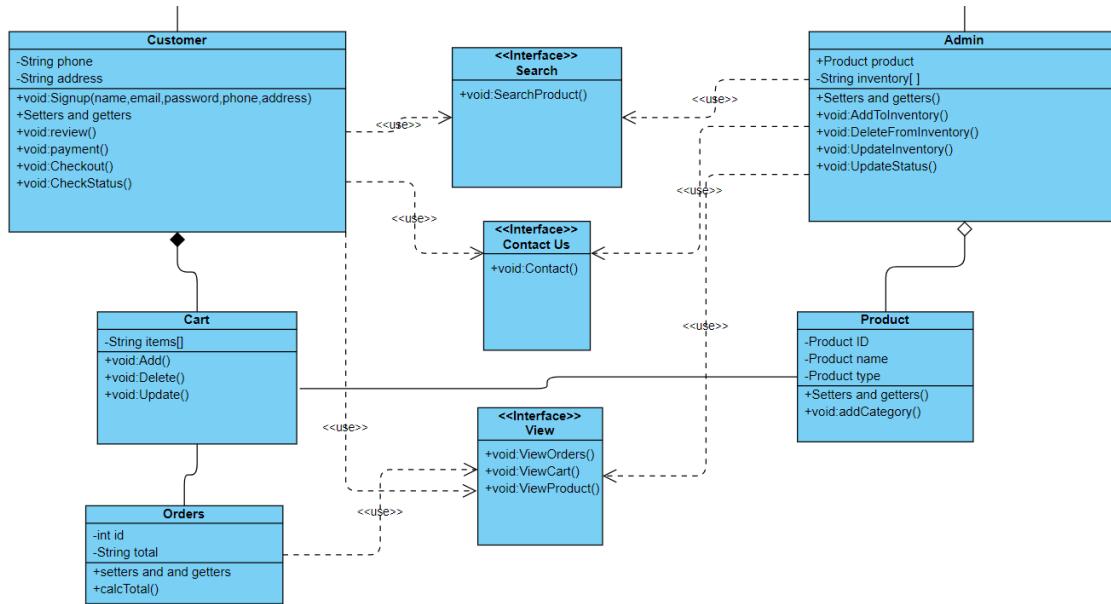


Figure 9: Class interfaces

4 Data Design

4.1 Data Description

- Data is originally stored as files.
- The admin uses web page forms to capture the data into the website.
- Database should be large enough to accommodate all products with their description, price, and images.
- Star Accessories has an established following of 28 thousand people, we're expecting at least 80% of that number to access the website.
- Each product has a unique ID number that's auto-incremented, Users also have an ID number that is randomly generated and stored.
- All data will be stored in SQL database.
- The user will be asked to fill in his information including delivery address, and personal information. The admin upon adding a product will be asked to provide all information about the product such as name, description, price, and available quantities. The user upon checking out will be asked to provide his product's preference. That includes color, size, and the quantity of the product he's ordering.

4.2 Database design description

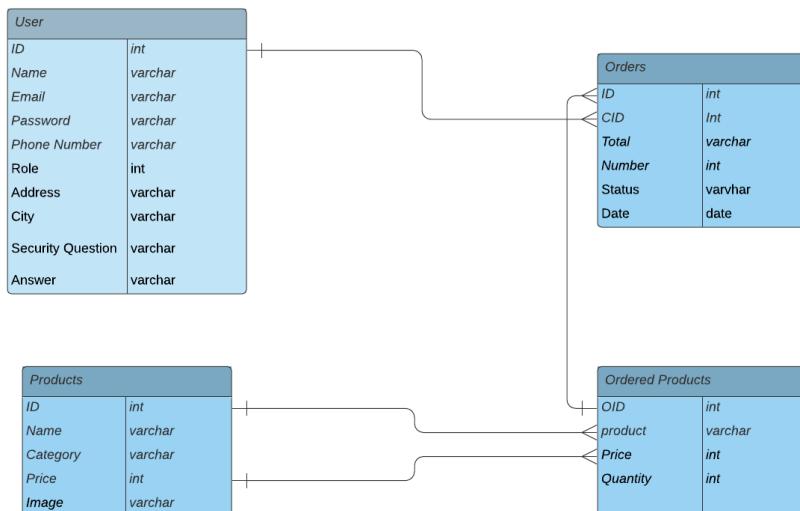


Figure 10: Database schema

5 Human Interface Design

5.1 Screen Images

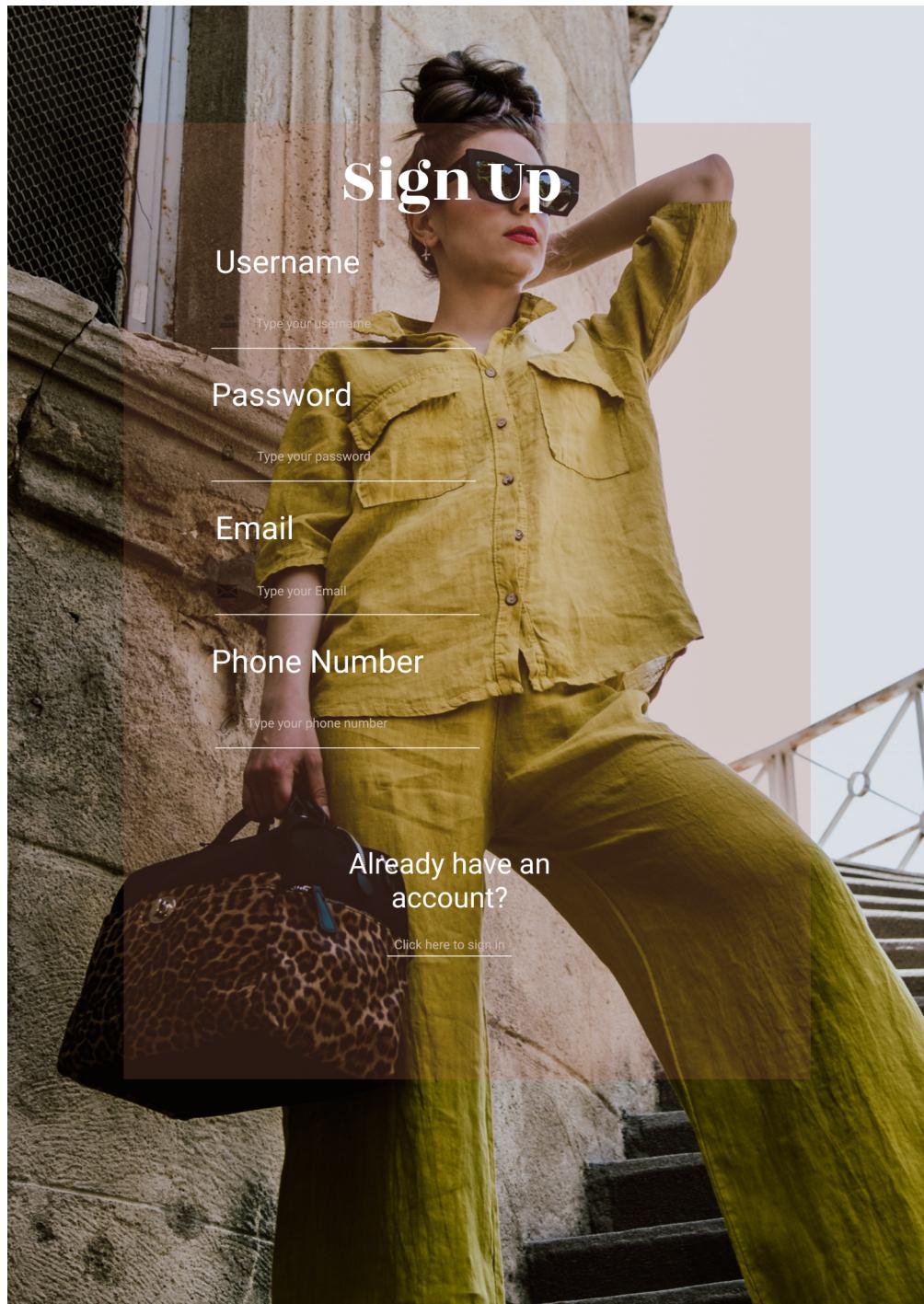


Figure 11: Sign Up Page

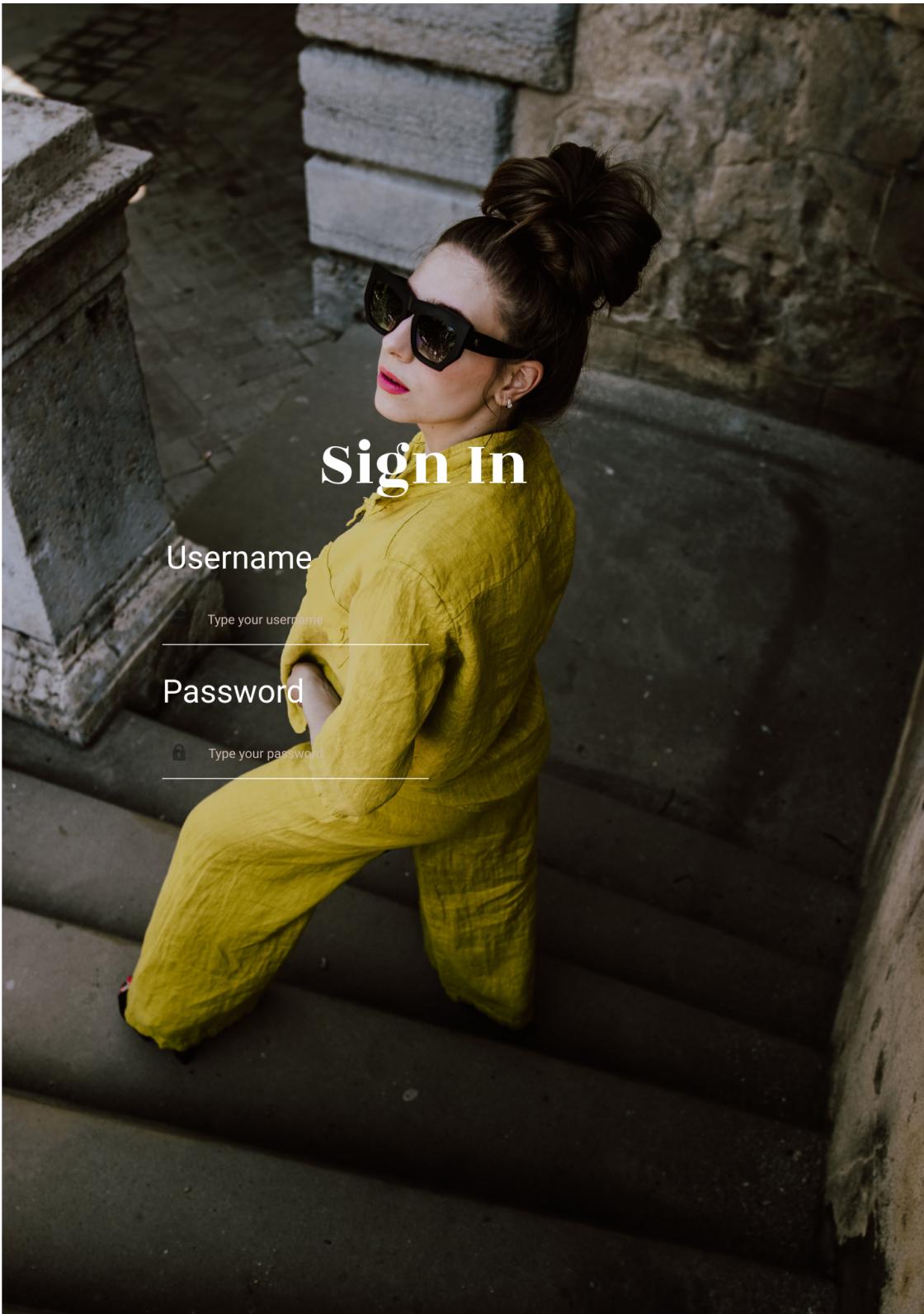
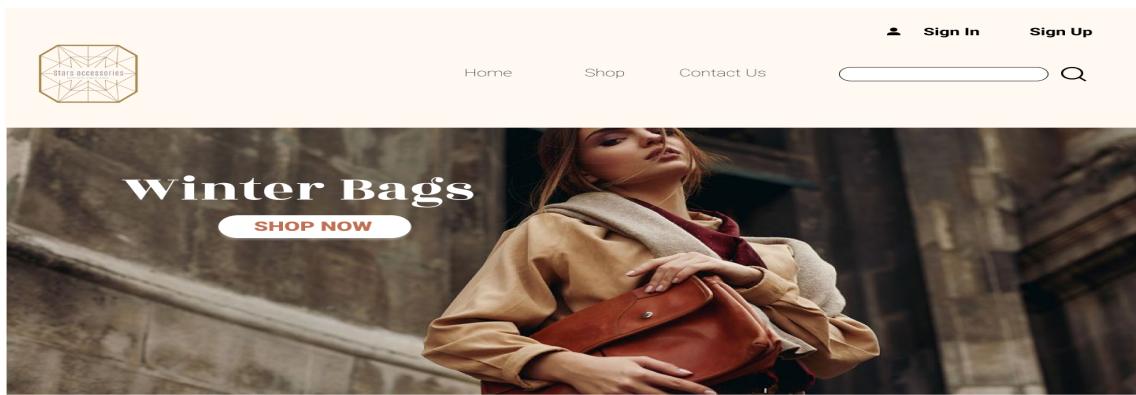


Figure 12: Sign In Page



Our latest Products



Gucci chouette
800 EGP



Valentino DCA
1200 EGP



Prada eliore
450 EGP

Browse



Bags



Sun Glasses



Wallets

Stars Accessories Online Shop

We connect people and products opening up a world of possibility. From bracelets to backpacks to tshirts and shorts – we give you access to everything you need and want. Our range is unparalleled, and our prices unbeatable.

Our products

- >Sun Glasses
- >Women Bags
- >Wallets
- >Accessoires

Useful links

- >Sign up
- >Login

Contact Us



Figure 13: Homepage



All Products

SALE UP TO
20%

Product Categories

- Sun Glasses
- Women Bags
- Wallets
- Accessoires

SPECIAL EASTER COLLECTION



Stars Accessories Online Shop

We connect people and products opening up a world of possibility. From bracelets to backpacks to t-shirts and shirts we give you access to everything you need and want. Our range is unparalleled, and our prices unbeatable.

Our products

- >Sun Glasses
- >Women Bags
- >Wallets
- >Accessoires

Useful links

- >Sign up
- >Log in

Contact Us



Figure 14: Products Page

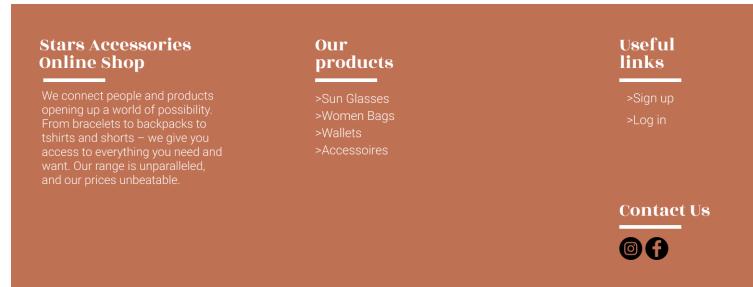
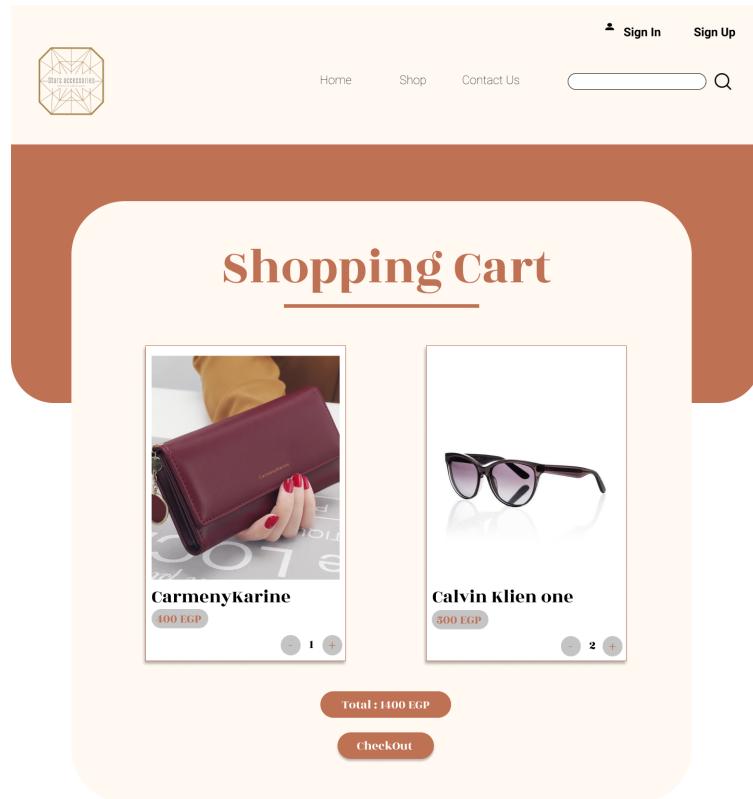


Figure 15: Cart Page

The screenshot shows the contact form on the website. At the top right are links for 'Sign In' and 'Sign Up'. Below them is a navigation bar with 'Home', 'Shop', and 'Contact Us' links, along with a search bar. The main content area has a brown header with the text 'Contact Us'. It contains two input fields: one for 'Name' and one for 'Message', both represented by light gray rectangles. A large orange button at the bottom is labeled 'Submit'.

Contact Us

Name

Message

Submit



Figure 16: Contact Us Page

6 Requirements Matrix

Table 9: Requirements Ratrix

Req. ID	Req Desc	Class	Status
FR01	Takes the user's info. and stores it into the database	client	Developed
FR02	Takes input string from user and searches products in database by name	client	Developed
FR03	Takes input string from user and adds it to the review table in the database	Product	Developed
FR04	Takes selected product from user and adds it to the cart	Cart	Developed
FR05	Tells the users the order shipping status	Order	Developed
FR06	Allows Admin to change order's current status	Admin	Developed
FR07	allows the admin user to add products to the inventory	Admin	Developed
FR08	allows the admin user to edit products in the inventory	Admin	Developed
FR09	allows the admin user to delete products from the inventory	Admin	Developed
FR10	Tells the Admin all user orders on site and info	Admin	Developed
FR11	function for clients to send their complaints to the admin	User	Developed

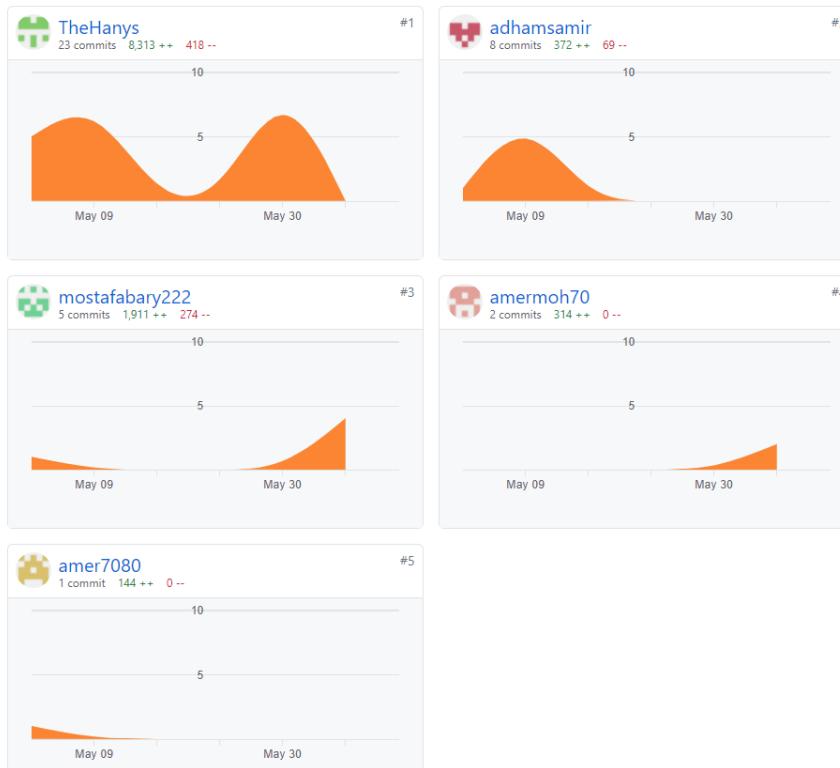
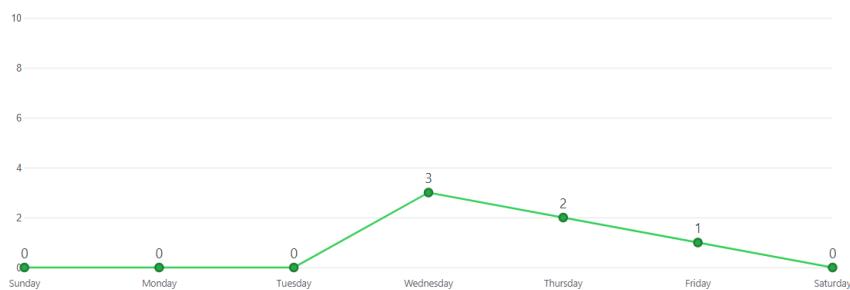
7 APPENDICES

7.1 Github

The screenshot shows a GitHub repository page for a project named "mostafabary222". The repository is described as an "E-commerce website for Stars Accessories". It has 41 commits, 1 branch, and 0 tags. The commit history lists numerous files updated, including CSS, PHP scripts like Cart.php and Contact Us.php, and various configuration and functional requirement files. The repository has 413 additions and 416 deletions across 23 files. The repository is public and has no releases or packages published.

Excluding merges, **3 authors** have pushed **17 commits** to master and **17 commits** to all branches. On master, **23 files** have changed and there have been **1,433 additions** and **416 deletions**.





7.2 Other appendices as appropriate

A Survey was conducted on the clients and followers of Stars Accessories, to gather more information on what problems they faced while dealing with the current Marketplace [4]. Thus enhance their experience when they get to deal with our web-application. [4]

2,173 Clients participated in the survey, which was conducted via paid-advertisements and discount vouchers for the participants.

The Survey is shown in the following figures with some statistical guide for the participants answers:

What do you think of our posts general layout?

2,173 responses

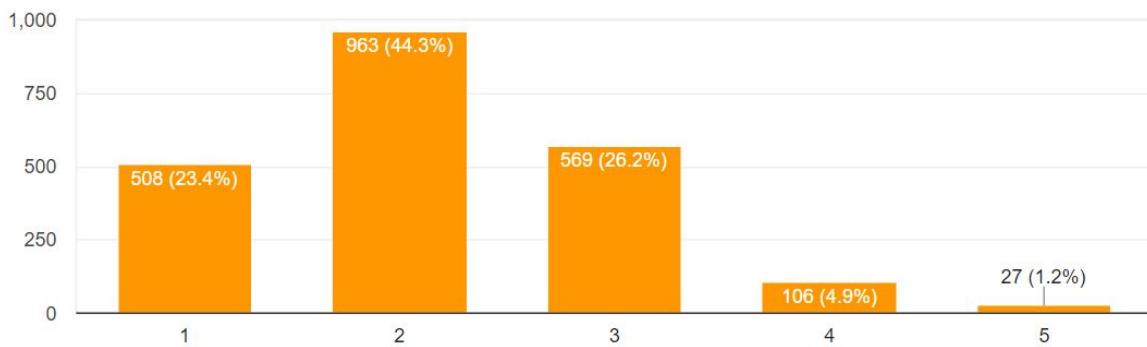


Figure 17: SA Clients Survey, Question 1

What do you think of the products distribution in the page?

2,173 responses

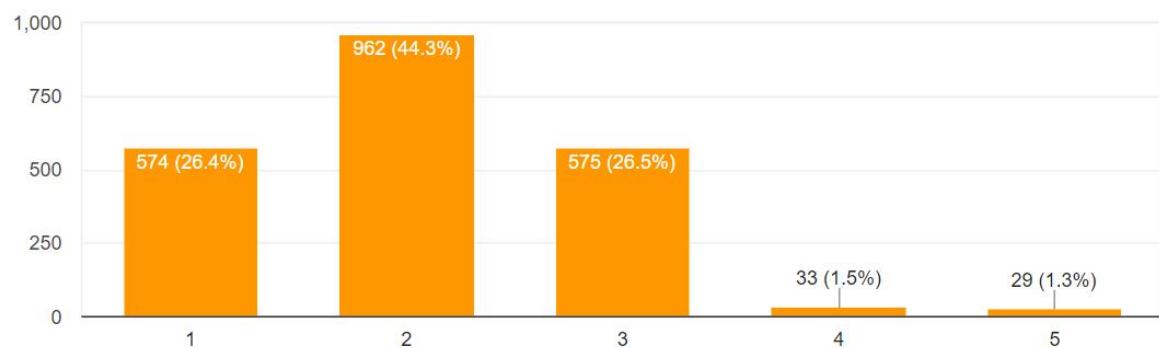


Figure 18: SA Clients Survey, Question 2

How easy is it to search for a certain product?

2,173 responses

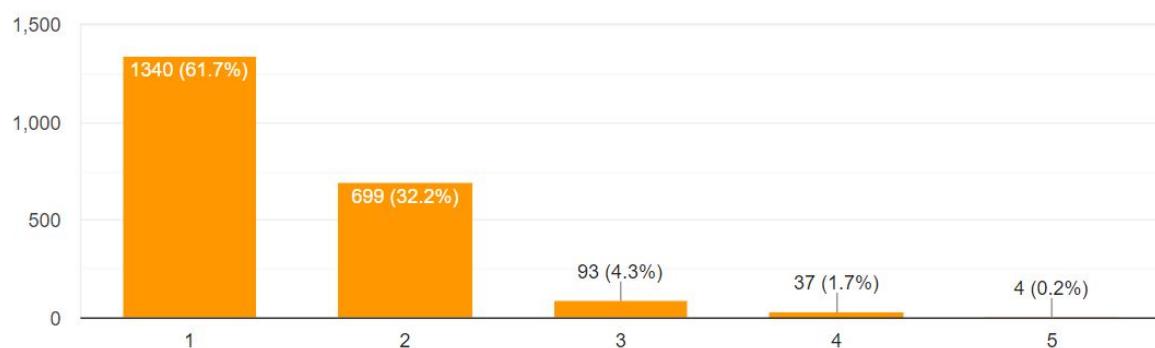


Figure 19: SA Clients Survey, Question 3

Did you buy any of our products?

2,173 responses

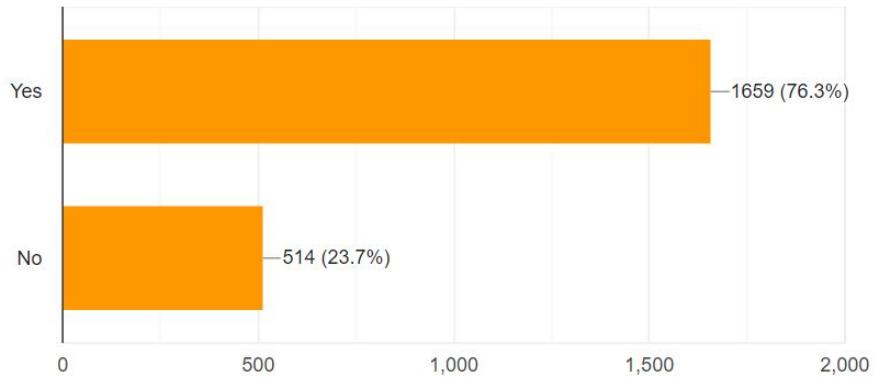


Figure 20: SA Clients Survey, Question 4

If yes, what do you think about our message response time generally and our after-sale customer service specifically?

1,668 responses

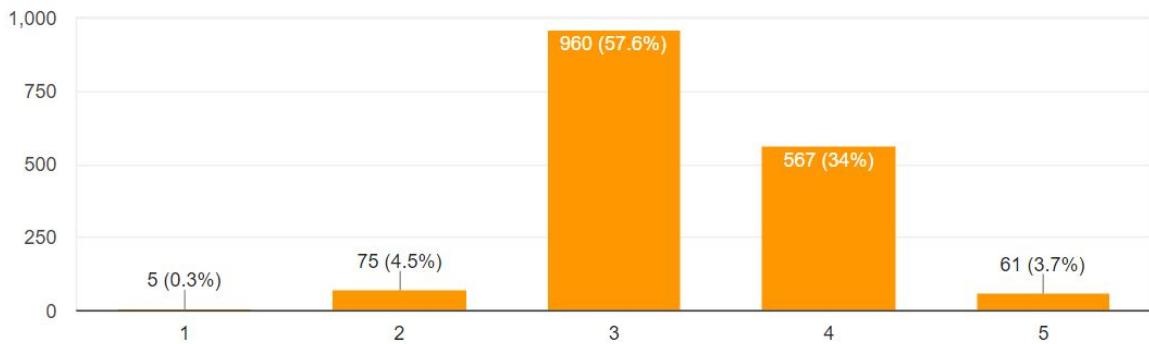


Figure 21: SA Clients Survey, Question 5

Survey Conclusion: The Clients of SA suffered from chaotic display for the products, along with a non-categorized layout. They also had very hard time finding a certain product, as there was no way to search for it rather than manually. Nevertheless, most users agreed that the customer support and response time was good.

Now we're able to put all the attention on the aspects that bothered the clients and users, along with keeping the features they liked.

References

- [1] Lars Ericson and Stephen Shine. “Minutia Deviation Tool: Software Design Description (SDD),(Version 2.0)”. In: (2015).
- [2] Koichiro Ochimizu. “Software Design Methodologies Goal and Scope”. In: () .
- [3] John Deacon. “Model-view-controller (mvc) architecture”. In: *Online][Citado em: 10 de março de 2006.] http://www.jdl.co.uk/briefings/MVC.pdf* (2009).
- [4] Betty Purwandari, Briant Otmen, and Larastri Kumaralalita. “Adoption factors of e-marketplace and instagram for micro, small, and medium enterprises (MSMEs)”. In: *Proceedings of the 2019 2nd International Conference on Data Science and Information Technology*. 2019, pp. 111–116.