

SOLE MATE

Footwear Shop Management System

CO328 SOFTWARE ENGINEERING PROJECT REPORT

GROUP 11

E/16/068 Rashmika de Silva

E/16/081 Praveen Dhananjaya

E/16/232 Shamra Marzook

INTRODUCTION

Online shopping is the process whereby the consumers directly buy goods, services, etc. from a seller interactively in real-time without an intermediary service over the internet. It is buying goods and services from merchants who sell via the internet.

Today online shopping has become the most efficient way for consumers to obtain their required goods and for the merchants to sell their goods. The online stores' goods vary from books, clothing, toys, footwear, to household appliances, hardware, software, pharmaceuticals, etc.

Many people prefer online shopping because of the convenience. For example, when a person visits a food city to buy groceries, he/she needs to walk throughout the shop, locate the required goods, and also often needs to stand in long lines to make the purchases. But shopping them online would get rid of these troubles and would be much more convenient.

It also allows a person to browse through endless possibilities, and even offers merchandise that would not be available in the local stores. Browsing through different online platforms at the same time, comparing items, their quality and their rates simultaneously makes shopping easy. The transactions occur almost immediately, and unlike a store, the online shopping portal is available 24 hours a day, 7 days a week and 365 days a year.

But towards the latter part of the year 2019, the whole world was hit with the pandemic of COVID-19, which affected the lifestyle of all the people. Humans started to fight against an unknown enemy. People were falling ill, facing critical conditions and dying. Nations were put to lock down and the people were prohibited from going out and made to stay away from each other.

Despite this, the people had to live their day to day life. They had to continue working and get access to daily needs. Merchants and store owners had to continue selling their goods to make a living. People had to buy the required goods from the stores. This paved the way for online shopping platforms to be utilized to fulfil the needs of people.

Various business opportunities for small scale businesses were brought up. Small scale startup businesses emerged. Those who already run “off-line” businesses started to go online to access new markets and overcome distances.

Sole Mate online shopping portal is implemented in order to fulfil the request of a small scale shoe shop merchant, who was in need of transitioning from the off-line business to an online shopping platform during this challenging pandemic situation.

Github link : [Sole Mate](#)

REQUIREMENT ANALYSIS

The requirements analysis and gathering processes are critical for the success of any software engineering project. Requirements analysis in software engineering is a process that determines the tasks that are required to determine the needs and conditions to design a new product or to make modifications to any existing product or application. This process considers the stakeholders' requirements and analyzes the documentation of the system. The requirements should be actionable, measurable, testable, and related to the defined needs of the system design. From the software-engineering perspective, requirements analysis is a three-step process.

1. Requirements Analysis

Requirements analysis is the activity of elaborating basic requirements established during the elicitation and negotiation tasks. Requirements analysis results in the software specification detailing the operational characteristics, interface with other system elements, and constraints that the software must meet.

Requirement analysis requires identifying the stakeholders and taking their needs into account to help them understand the implications of designing the new system, along with what modules are worth implementing and which ones are more cost-efficient, and then creating a software-requirement specification document. To clearly elicit the stakeholders' requirements, different processes, such as developing a scenario or user stories, and identifying the use case which is being used for the project, can be utilized.

Identify Key Stakeholders

- Stakeholders are people or organizations that have a valid interest or use in the system.
- Anyone who operates the system. Eg: admin, operator
- Anyone who benefits from the system Eg: customer
- Organizations responsible for the system design Eg: project team

Once the stakeholders are successfully identified, interviews are conducted through different processes; the needs and requirements of the system are identified, and a requirements specification document is prepared. The document is then discussed with the major stakeholders to identify any ambiguity with the requirements and understanding of the system.

User Story	Acceptance Criteria
As a customer I want to view a list of products so I can select some to purchase	<ul style="list-style-type: none"> • See a thumbnail image for each product • Click to view details for product • Add to cart from detail page • Search for a product • View products by category
As a customer I want to review my cart so I can make adjustments prior to checkout	<ul style="list-style-type: none"> • View quantities and items in the cart • See a total cost before checkout • Click to navigate to a product detail page • Adjust quantity of items • Remove items
As a customer I want to check out so I can get my products	<ul style="list-style-type: none"> • Trigger checkout from any page, if there are items in the cart • Enter a shipping address, a billing address, credit card number • Show total cost before finalizing • Show confirmation message after finalizing • Verify payment via our payment processor
As a customer I want to review my orders to see what is purchased in past	<ul style="list-style-type: none"> • View a list of open and completed orders • See the status of the order • Navigate to the details of the order
As an administrator I want to modify the list of products so I can adjust our offerings overtime	<ul style="list-style-type: none"> • Add or remove products • Select a category for the product • Modify category • Modify product images
As an owner I want to analyze sales	<ul style="list-style-type: none"> • View dashboard with a summary of purchases and details about the inventory.

Table 01: User Story

2. Requirements Elicitation

The requirement elicitation activities are performed to formulate the objectives for the system, what is to be accomplished, how the system or product fits into the needs of the business, and how the system is to be used. Requirements elicitation (requirements gathering) combines elements of problem-solving, elaboration, negotiation, and specification.

Requirement elicitation techniques:

Requirements elicitation is generally performed using certain techniques, such as interviews, questionnaires, observation, document analysis, studying similar systems, and prototyping. The applicability of the five techniques excerpted from are presented in Table 02. Techniques of requirement elicitation and applicability.

Technique	Applicability
Interview	Can be used when the users are known and can be contacted directly. It is a helpful technique for establishing and verifying information or procedures
Questionnaires	Used when similar types of data or information need to be obtained from a large number of respondents or from remote locations
Studying similar systems	Suitable to get information about the facts related to the system development.
Prototyping	Used where there is a great deal of uncertainty about the requirements, or where early feedback from stakeholders/users is required, or where users (public) features preferences cannot be determined precisely in advance;

Table 02: Techniques of requirement elicitation and applicability.

In this project, most of the above said techniques were used for requirement elicitation. We identified the uses of this online shoe shopping system as any user who is accessible for the internet and the administrator or the operator of the system.

- Considering the owner as the administrator, he was interviewed to obtain the requirements of the admin.
- We also asked some users who often use e-commerce sites for purchases through questionnaires, as to what features they would prefer in a shopping portal for footwear, if they were to buy using it. Further we also gave them the opportunity to express their thoughts on the features of existing footwear online shopping sites.

- We were able to gather adequate data, and with the help of these data on the existing sites, we did a study on these existing sites and gathered information and ideas for the system development.
- We created class diagrams, ER diagrams, UI/UX diagrams, etc. as initial steps in order to show how we plan to proceed with the project and provided them as prototypes for the owner. Further we showed the UI diagrams to some users and also obtained their feedback on it as well.
- Gathering the feedback, and analyzing them we improved our solution to a better user friendly one that fulfill the requirement of the stakeholders.

3. Requirements Documentation

This step involves documenting the requirements in various forms, including summary lists, natural language documents, visual documents, use cases, user stories, or process specifications. A requirement specification document is categorized in different ways according to the stakeholders' needs, helping to create a clear contract between development and business. The following sections include the different categories of requirements specification documents that are essential for designing this application: the functional requirements, constraints, system requirements, etc.

- **Requirement Classification**

Based on the functions, requirements can be classified into, functional and non-functional requirements which are the pillars of a project. They articulate the needs & wishes and specify the tasks for the development of the application.

Functional requirements describe system functionalities or services. It is all about the functions and core operations of the e-store that enable a user to take action on the website.

- I. Mobile-Friendliness - the ability to use the application in a mobile platform as well is required because at present most people use their smartphones to access these applications. The position of the essential buttons and options on web pages should change for better shopping experience in a mobile platform
- II. Third-Party Integrations - Several third party integrations will be required for the application. Specifying a number of third-party integrations will make the application structured. For example, providing the customers with more than one option for making payments such as PayPal, credit card payments, etc. is required.

- III. Product Attributes - Providing the customer with the option to choose the size and color of the shoe he needs to purchase. Adding details about the item together with several pictures and also providing the facility to zoom in so the customer is able to visualize in a better way.
- IV. Order and Checkout Flow - Specifying how the orders are processed in the store and whether and how it should be optimized. The customer should be registered in order to make a purchase or enable a guest checkout. The order status should be visible both to the customer and the admin. The user should be able to add more items into the cart before he chooses to checkout. Once he proceeds to checkout he should be able to make payments and the website shall show whether the order is confirmed and is processed.

Non-functional requirements, which define system properties and constraints. Quality attributes of a system, they form user experience and imply some global, abstract expectations from the product.

- I. Usability - the website should be intuitive and easy-to-use. Website usability is also defined by how easily a user can achieve their goal in a single page visit, how quickly they perform the tasks in the application, how memorable and intuitive the design is, number and time of errors users make. A customer should easily find the right product for them, understand what problems it solves, and make a purchase without any concerns.
- II. Security - Only the system administrator can manage and change access permissions to the system. The website must be resilient to any kind of attacks and be bound to secure the data of the customers.
- III. Performance - the web application should load fast even when the website traffic increases.
- IV. Maintainability - The system will work great now. But issues may come in future. Making the website maintainable from the initial development phase so that it is easy to identify and resolve the system faults in the future.
- V. Scalability - the website can grow and expand its functionality in future and it should happen without affecting its performance. So it should be scalable like to add more memory, servers, or disc space to complete more transactions on the website.

- **Product Perspective**

The online shopping-cart application is a web-based system. It can be accessed using Internet Explorer, Mozilla Firefox, and Google Chrome.

- **User Interface**

The two interface types found in the online shopping-cart application are as follows:

1. User Interface
2. Admin Interface

Both interfaces are described in detail later.

- **Hardware Interface**

The online shopping application needs minimum hardware requirements. The user needs a PC to make purchases while an admin of the system needs a PC to deploy the server and use the system for the operations of the system.

- **Software Interface**

Includes the requirements that are needed to run the system efficiently. The operating system needed for the system to run effectively, the interface to run the application, the framework for running python web applications, the integrated development environment to develop the application, and the third-party tool used for editing purposes are as follows:

1. Operating System in the PC
2. Web Browser: Internet Explorer, Mozilla Firefox, or Google Chrome
3. Python framework: Django
4. An IDE: A python-based IDE to develop the system

SOFTWARE DESIGN

The next phase of software development is the software design process. The design phase deals with transforming the customer requirements identified in the Requirements Specification (SRS) stage into a form implementable using a selected programming language. This whole process can be divided into sub-phases of design:

- System Modelling
- Interface Design

1. SYSTEM MODELLING

In the system modelling phase, abstract models of the system are developed, with each model presenting a different perspective of the system. This helps to understand the system and its functionality as it is and permits to specify the structure, behavior of the system by increasing the decomposition and modularization of the system. For the system modelling in this project, the visual modelling approach is used with the modelling language UML, as it allows the levels of abstraction to be raised while maintaining rigorous system syntax and semantics. Essentially, 3 different design perspectives are used in order to understand the system, namely behavioral perspective, structural perspective and interaction perspective.

- **Behavioral perspective** - To model the dynamics of the system and how it responds to the events. The **Activity diagram** is used to depict this perspective.
- **Structural perspective** - To model the organization of the system and the structure of the data that is processed by the system. The **Class diagram** is to depict this perspective.
- **Interaction perspective** - To model the interactions between the system and its environment. The **Use-Case diagram** is used to depict this perspective.

An Entity-Relationship (E-R) diagram is used to analyze the business data schema.

ER-DIAGRAM

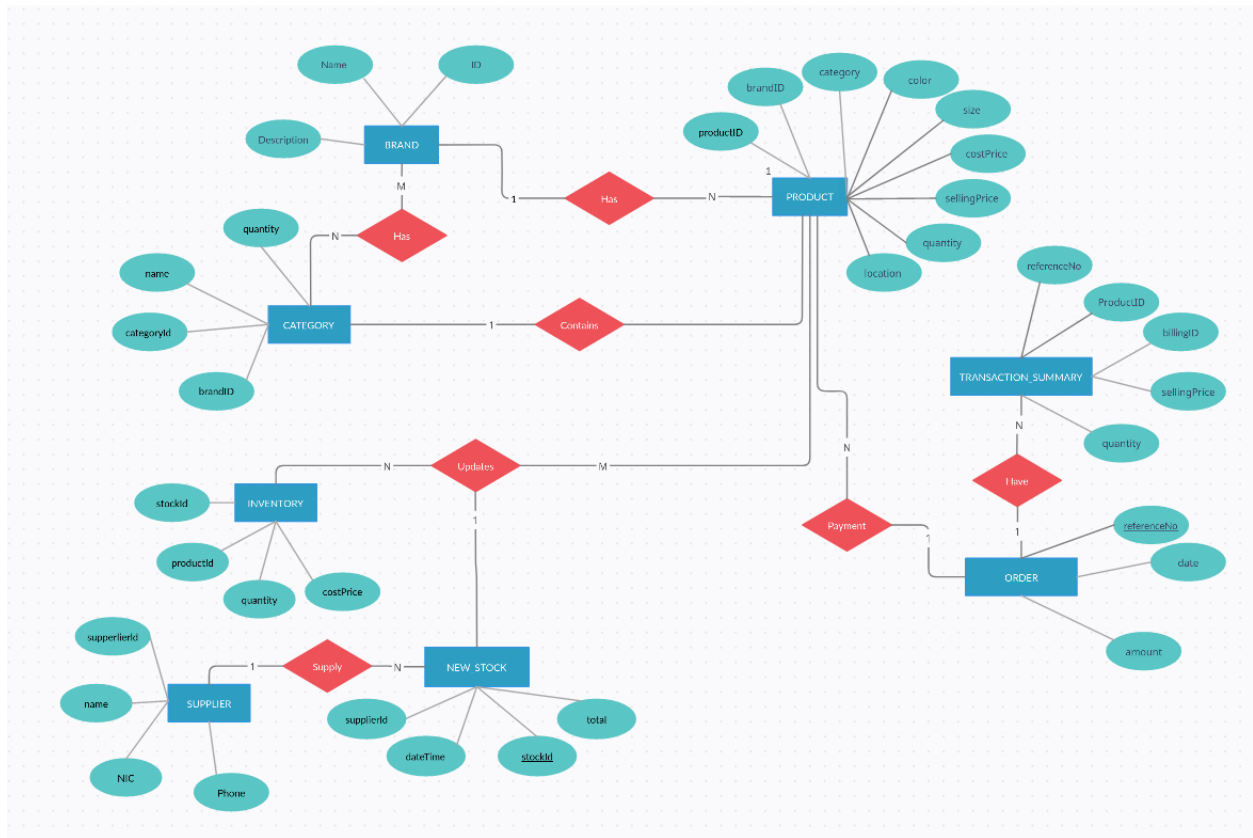


Figure 01: ER-Diagram

ER-diagram depicts the systematic analysis of interrelated entities in the system and defines the data or information structure which can be implemented in a database. In this system, 8 entities are identified which actively engage in the data flow in the system. Therefore, all those 8 entities can be put into a relational database. Those entities and their attributes are as follows;

- Product - ProductID, brandID, category, color, size, costPrize, Qty, sellinPrice
- Category - CategoryID, name, Qty, brandID
- Brand - Name, ID, description
- Inventory - ProductID, Qty, StockID, costPrice
- Order - RefNo, amount, data
- Transaction Summary - RefNo, productID, billingID, sellingPrice, Qty
- NewStock - StockID, dataTime, supplierID, total
- Supplier - NIC, Name, phoneNo, supplierID

USE CASE DIAGRAM

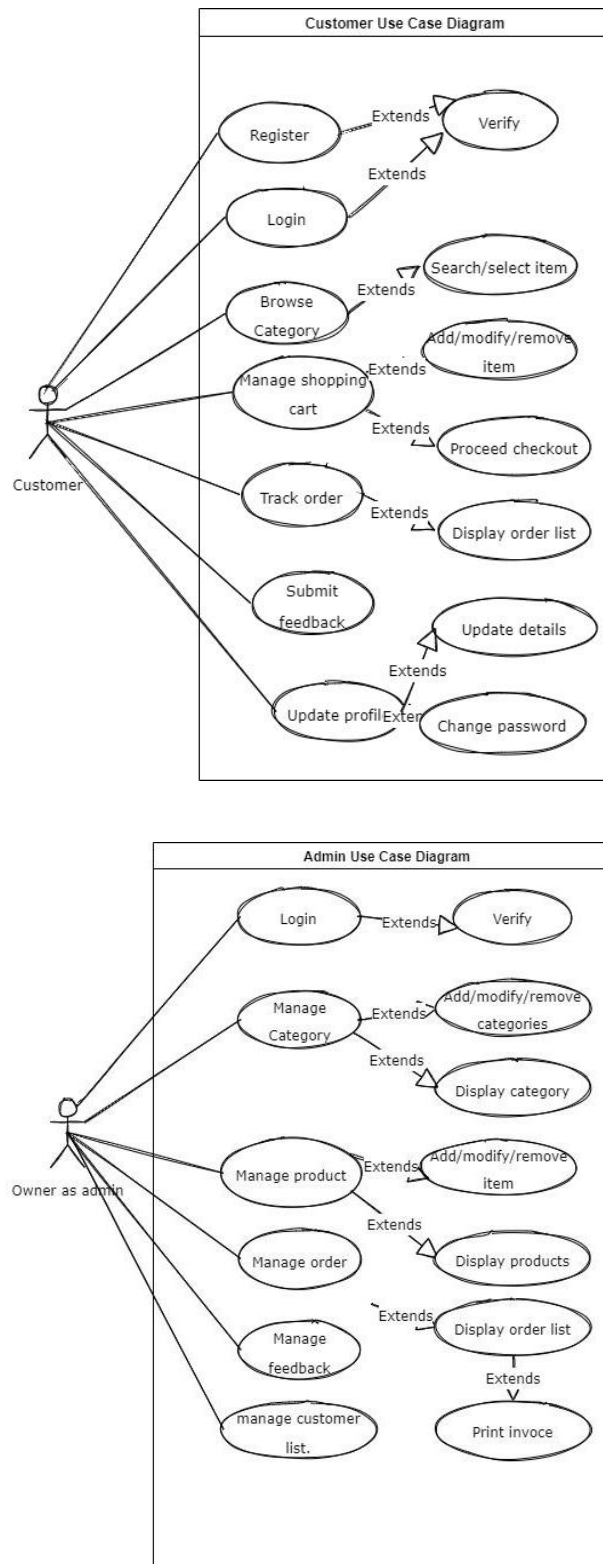


Figure 02: Use Case Diagram

A use-case diagram is the primary form of system requirements for a new software system. It depicts possible users' interactions with the system and various use cases and different type of users the system may have. In this E-commerce system, it is identified that there are two types of users that would interact with the system, namely, the consumer and the owner of the retail shop.

- **The consumer:**

The consumer is one of the main stakeholders around whose requirements the system is built. The consumer mainly interacts with E-commerce web pages and does not have administrative privileges over the system. Consumers can register themselves as a new user to the system via the register functionality and with the set credentials in the registration phase, they can login into the system and make necessary updates to their profile, if needed. Moreover, consumers can browse categories for products, select items, manage shopping cart, proceed to check out and make payments, as well as submit feedbacks. It should be highlighted that, to access the website, browse through categories, to select and add items to the shopping cart, users need not have an account. Registered account is only needed when making the payments.

- **The Owner:**

The owner of the retail store acts as the administrator of the system. The owner has all the privileges an average user has and apart from all the user privileges, the owner has some administrative privileges as well, which assists him manage the E-commerce website. The administrative privileges the owner has are as follows;

- Manage categories - add/modify/remove categories and hide existing Categories from an average user
- Manage products - add/modify/remove items and hide existing Items from an average user
- Manage orders
- Manage consumer feedback
- Manage consumer lists

ACTIVITY DIAGRAM

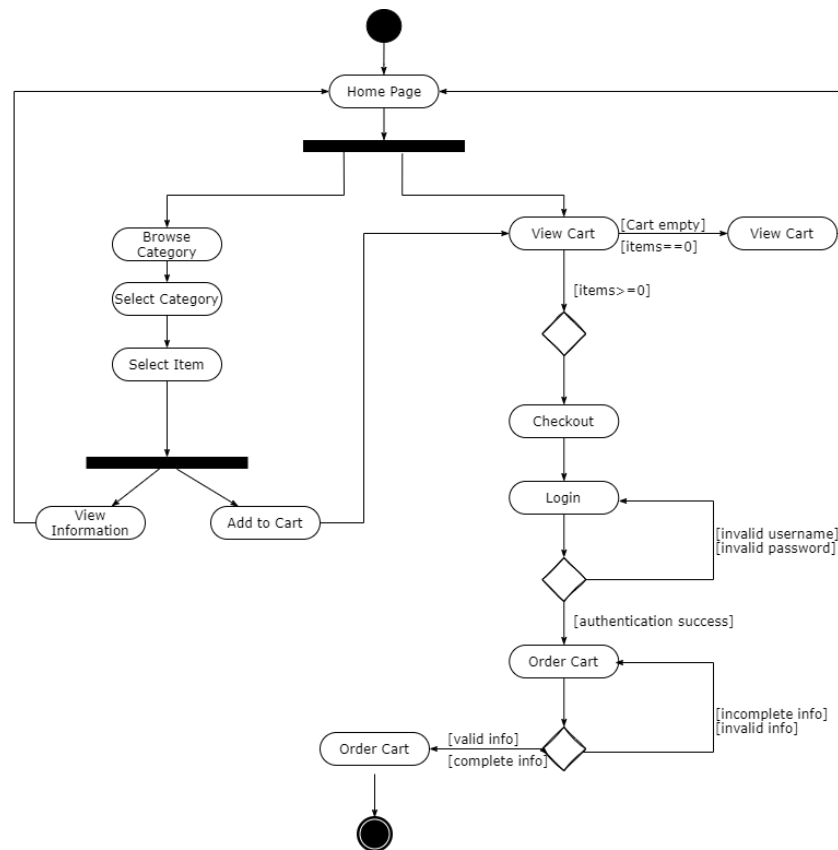


Figure 03: Activity Diagram

The Figure 03 above demonstrates the activity flow for the online footwear shopping application.

The flow begins when the user first runs the Sole Mate application that appears in the web browser. The user can browse through the available list of categories and can choose either to select a category or to directly view the cart. In the category, a user can select and view more information for details about a particular item before deciding to add it to the shopping cart by clicking on the cart icon next to the item. The user can then decide to either continue shopping by clicking the continue shopping button or can check out by clicking on the checkout option. If there are no items in the cart, then the user does not have an option to click checkout. The user can check out after doing the user authentication by logging in with the username and password.

Once the user successfully logs in/registers, the order form, where the user can put the correct information to place the order appears. After the user successfully inputs the correct information, placing an order is successful, and the user can see the success message.

The additional flow step for the administrators is that they can view the user's checkout, and the product details by using the database after the user successfully places an order. Further, he can also view details regarding the items and inventory of the system.

CLASS DIAGRAM

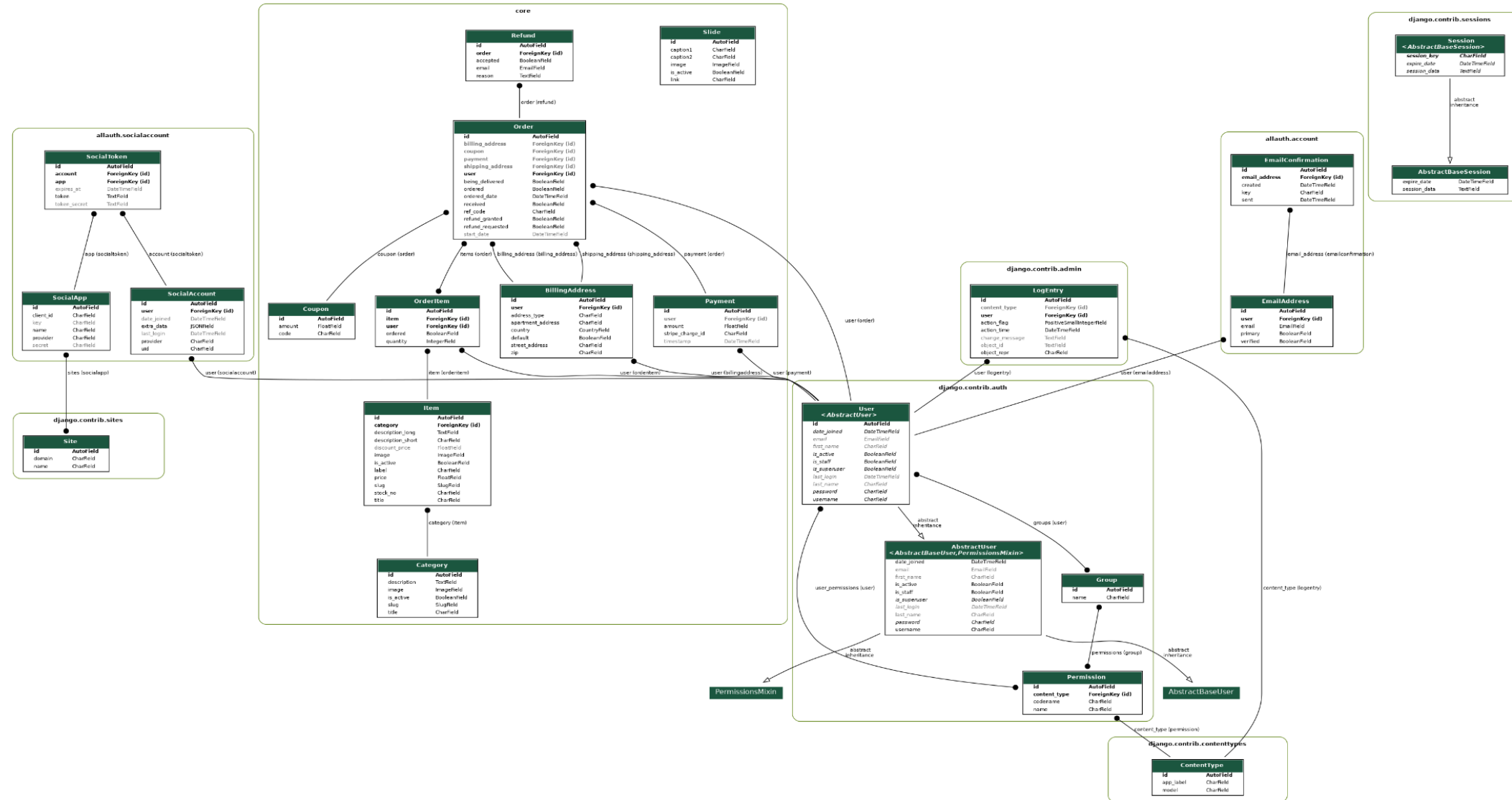


Figure 04: Class Diagram

The UML class diagram depicts the classes in the system and the associations between the classes in the objected-oriented model. It is useful in identifying the relationships between objects and categorizing the objects in an accessible and coherent manner.

In this E-commerce system, core classes are identified as follows;

- User class - Models the users of the system and contains all methods (place order, make payment etc.) and attributes (name, email, ID etc.) relevant to the user.
- Order class - Models an order that can be placed via the shopping Portal and contain all the attributes (ID, billing address, payment etc.) relevant to the order class
- Payment class - Models the payment that can be made via the portal and Contains all the attributes (ID, user, amount timestamp etc.) relevant to the payment.
- BillingAddress class - Models the billing address and contains the attributes Relevant to the billing address (ID, user, address, zip code etc.)
- Item class - Models the products that are on sale on the website and contains the attributes relevant to the item (ID, category, image, description, price etc.)
- OrderItem class - Models the items that are added to the shopping cart and And contains the attribute relevant to the ordered items (ID, item, user, quantity etc.)

User class is a standalone class that has been derived from the abstract class *AbstractUser* available in the Python Django framework. Thus it is a generalization of an abstract class. Also, it has been accessed by other core classes in the class diagram. Hence it is in an *association* relationship with other core classes in the structure.

Order class is in *association with* every other class. Due to the nature of the order class and the required attributes, it has to depend on every other class in order to get the relevant information. Among the rest of the classes, the *Item class* is a standalone class, which does not depend on any other class. *OrderItem class* in *association* relationships with *Order*, *Item* and *User classes*. The remaining classes, *BillingAddress* & *Payment*, behave almost as standalone classes, except for their access to the *User class*.

2. INTERFACE DESIGN

Interface design is the specification of the interaction between the system and its environment. This design phase proceeds at a higher level of abstraction with respect to the internal workings of the system. During the interface design phase, the internals of the system are completely ignored and the system is treated as a black box. Attention is focused on the interaction between the target system and its users, devices and how it would interact with other systems in the environment.

During the requirement analysis phase, it is identified that the stakeholder's requirement is an E-commerce website with a built-in inventory management system. The target group of consumers of the stakeholder's business mainly interacts with the E-commerce software where they can search, view and purchase products and make payments via online payments platforms. The built-in inventory management system assists the stakeholder to manage stocks of goods in his retail business. It also provides information about the availability of the products on sale when making a purchase.

Essentially, 3 key parts are considered when designing the user interfaces for the E-commerce website, namely, the storefront, products and the checkout.

- The Storefront

The virtual storefront allows the stakeholder to display their business similarly to how they would with their store outlet, highlighting key information about the brands, offers, featured products etc.

- Product pages

The product pages essentially include product information, price, images and depending on the product, the list of variations available to purchase.

- Checkout page

The checkout page features a simple and secure transactional process between consumers and the stakeholder's business. Prospective buyers are able to see a summary of their purchase including estimated delivery times and breakdown of costs. Returning users are able to login in with their details to avoid re-filling in forms about payments, billing and delivery information.

Considering all these aspects, simple, consistent user interfaces are designed to cater to stakeholders' requirements. Greater attention is given to the use of common UI elements such as input controls, navigational components, containers etc. to create effective presentation and consistency throughout the UIs. Page layout, especially the spatial relationship between items on the page and the structure of the page is purposefully managed to increase the visibility, reduce learning time and highlight the important pieces of information such as the price of items, availability of sizes and colours, preview images of the product etc. Designed UIs' are as follows;

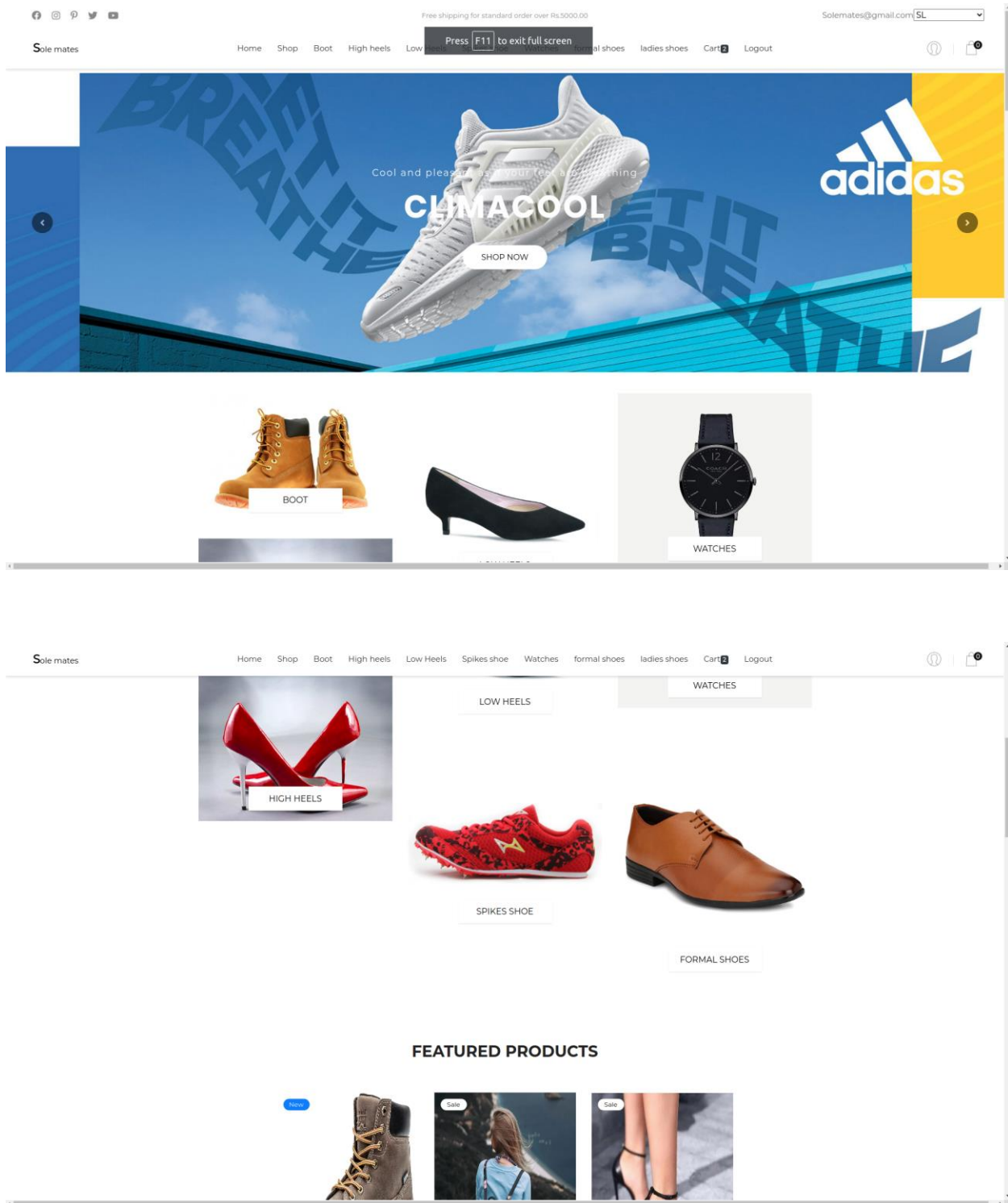


Figure 05: UI for the storefront

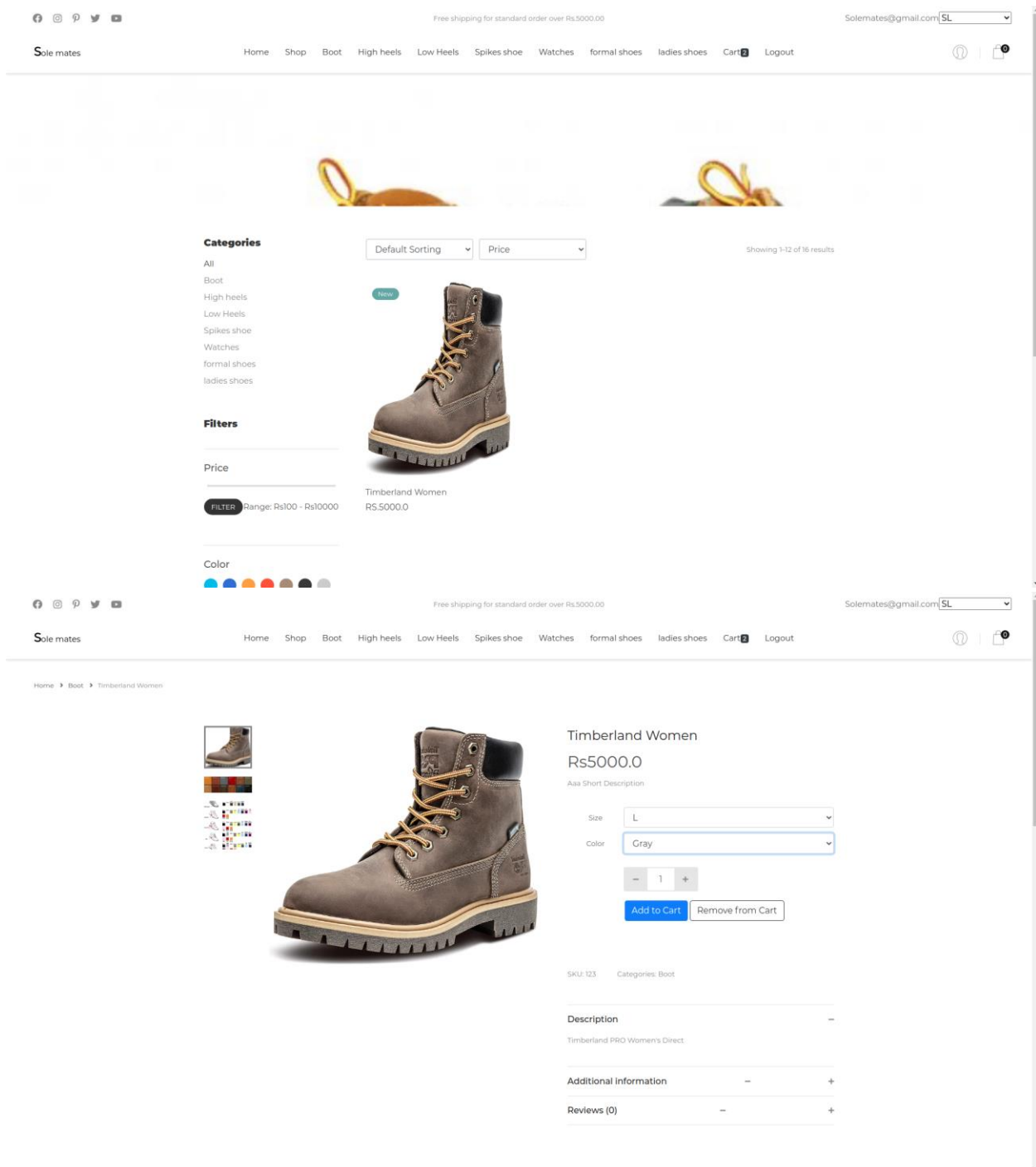
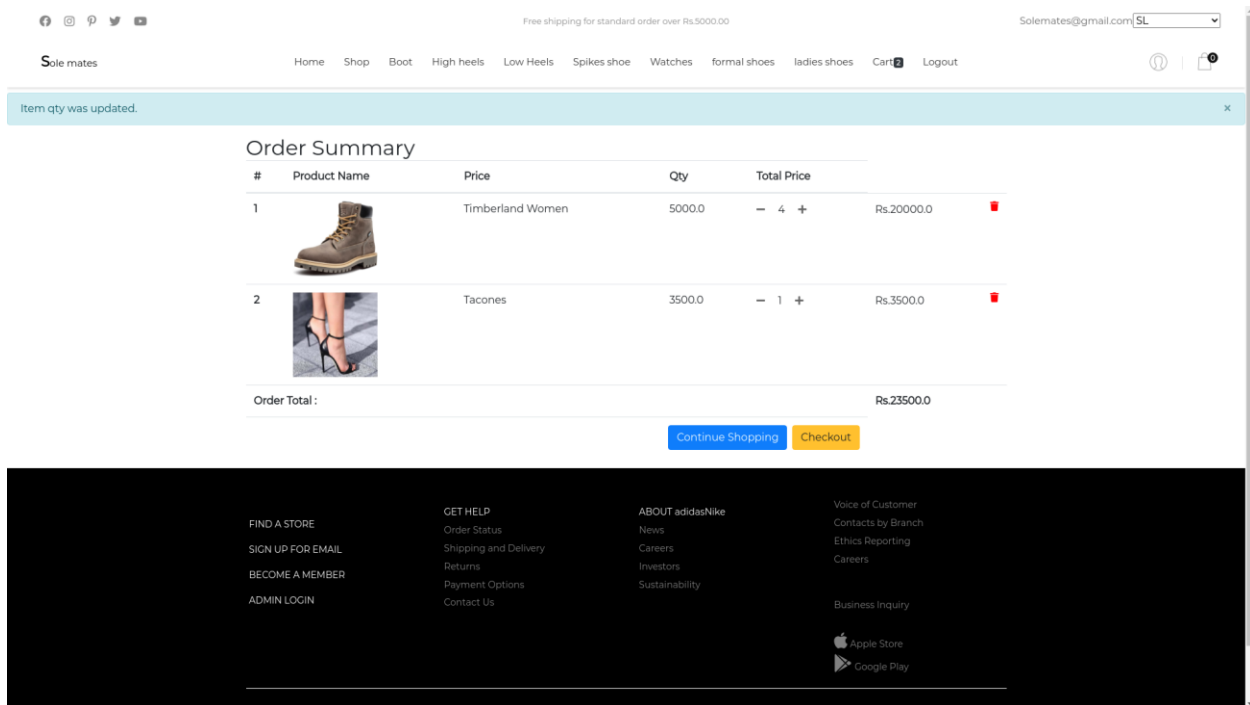


Figure 06: UI for the product pages



Checkout form

Below is an example form built entirely with Bootstrap's form controls. Each required form group has a validation state that can be triggered by attempting to submit the form without completing it.

Billing address

Address

315/1/a.Athuwaththa

Address 2 (Optional)

Naranvita

Country

Sri Lanka

City

Choose...

Zip

20500

☐ Shipping address is the same as my billing address

☐ Save this information for next time

Payment

☐ Stripe

☐ PayPal

[Continue to checkout](#)

Your cart

Timberland Women quantity: 4	Rs.20000.0
Tacones quantity: 1	Rs.3500.0
Promo code	-Rs.
Total (USD)	Rs.23500.0

Promo code [Redeem](#)

The screenshot shows the checkout page for 'Sole mates'. At the top, there's a navigation bar with social media icons, a shipping notice ('Free shipping for standard order over Rs.5000.00'), a user email dropdown ('Solemates@gmail.com'), and a shopping cart icon with a '2' badge. The main heading is 'Checkout form', followed by a subtext: 'Below is an example form built entirely with Bootstrap's form controls. Each required form group has a validation state that can be triggered by attempting to submit the form without completing it.'

The form is divided into two main sections: 'Billing address' and 'Your cart'.

Billing address section:

- Address:** A text input field containing '315/1/a.Athuaththa'.
- Address 2 (Optional):** A text input field containing 'Naranvita'.
- Country:** A dropdown menu with 'Sri Lanka' selected.
- City:** A dropdown menu with 'Choose...' selected.
- Zip:** A text input field containing '20500'.
- Shipping options:** Two checkboxes: 'Shipping address is the same as my billing address' and 'Save this information for next time'.
- Payment:** Two radio buttons: 'Stripe' and 'PayPal'.
- Continue to checkout:** A prominent blue button at the bottom of the form.

Your cart section:

- Timberland Women:** quantity: 4, Price: Rs.20000.0
- Tacones:** quantity: 1, Price: Rs.3500.0
- Promo code:** A text input field with a '-Rs.' label.
- Total (USD):** Rs.23500.0
- Promo code:** A text input field with a 'Redeem' button.

Figure 07: UI for the checkout page

The screenshot shows the login page for 'Sole mates'. The navigation bar is identical to the checkout page, but the user email dropdown shows 'SL'. The main heading is 'Sign In', followed by a subtext: 'Please sign in with one of your existing third party accounts. Or, sign up for a example.com account and sign in below:'. Below this, there are two options: 'Google' and 'or'.

Sign In form:

- Username*:** A text input field.
- Password*:** A text input field with a 'Password' placeholder.
- Remember Me:** A checkbox.
- Forgot Password?:** A link.
- Sign In:** A blue button.

Footer section:

- FIND A STORE:** A link.
- SIGN UP FOR EMAIL:** A link.
- BECOME A MEMBER:** A link.
- ADMIN LOGIN:** A link.
- GET HELP:** A link.
- Order Status:** A link.
- Shipping and Delivery:** A link.
- Returns:** A link.
- Payment Options:** A link.
- Contact Us:** A link.
- ABOUT adidasNike:** A link.
- News:** A link.
- Careers:** A link.
- Investors:** A link.
- Sustainability:** A link.
- Voice of Customer:** A link.
- Contacts by Branch:** A link.
- Ethics Reporting:** A link.
- Careers:** A link.
- Business Inquiry:** A link.
- Apple Store:** A link.
- Google Play:** A link.

Figure 08: UI for the login page

TECHNOLOGY STACK

Python	OOP Programming language
Django	Web framework with many inbuilt features that suits for a system of this nature
Django ORM	Object - relational mapping/database - Supports a pythonical way to create SQL to query and manipulate your database and get results in a pythonic fashion
Allauth Django	Authentication library - Useful in user registration and authentication components in web applications
HTML/CSS	Markup language UI and web pages designs

Table 03 : Programming Languages and Frameworks

USE OF TOOLS

Designing	<ul style="list-style-type: none">• Creately• Figma• draw.io
Testing	<ul style="list-style-type: none">• Coverage in GitHub - Ensures testing coverage• Flake8 - Code quality testing tool• Standard python unit tests - Standard code test• Django Runtime Debug - For runtime code testing
Version Controlling	<ul style="list-style-type: none">• github

Table 04 : Tools used

UNIT TESTING

Unit testing of a system evaluates and validates the system specs and ensures that the prototype system exhibits the expected behaviour stated at the requirement analysis phase, before the system is sent to the production stage. In this project, unit testing was performed in two directions; namely, interface testing and backend implementation testing. Illustrated test cases, their expected outcome and the obtained results are as follows;

Unit Test	Expected Outcome/ Functional Requirement	Outcome
Authentication	The E-commerce website shall have two types of authentication: User authentication and Admin authentication	Successful
Searching/browsing categories	The E-commerce website shall be accessible to all the users to browse all the categories and the items.	Successful
Shopping cart functionality	The users shall be able to view the items they added to the shopping cart.	Successful
Admin privileges for categories	The Admin shall be able to upload new/revised items as well as to add/modify the categories.	Successful
Admin privileges to check registered users	The Admin shall be able to view all the users registered in the system.	Successful
Admin privileges to view placed orders	The Admin shall be able to view all the information about users who placed an order from the shopping cart.	Successful
Constraints on checkout procedure	The users shall not be able to check out with an empty cart.	Successful
Constraints on checkout procedure	The users shall not be able to place an order without providing valid information for all rows in the order form.	Successful
Constraints on checkout procedure	The users shall not be able to place an order if any of the columns in the order form are left empty.	Successful

Table 05 : Unit Testing Summary

Test Results

Type	Name	Result	Pass	Test
Users				
	home page load	Pass	10	10
	login	Fail	7	10
	add to card	Pass	10	10
	check out	Pass	10	10
	payment	Fail	3	10
	auto filling	Pass	10	10
	logout	Pass	10	10

Figure 09 : Authentication and Login Testing

Type	Name	Result	Pass	Test
DataBase				
	django model	Pass		
	Add item	Pass	10	10
	Remove item	Pass	10	10
	Update item	Pass	10	10
	Add user	Fail	0	10
	Remove user	Fail	0	10
	Update user	Fail	0	10

Figure 10 : Database Access Testing

Flake8	Name	Result	
	E841	400	
	E501	107	
	E999	1203	
	E302	201	
	E402	370	
	E409	156	

Figure 11 : Code Quality Test

Type	Name	Result	Pass	Test
Users				
	home page load	Pass	10	10
	login	Fail	7	10
	add to card	Pass	10	10
	check out	Pass	10	10
	payment	Fail	3	10
	auto filling	Pass	10	10

Figure 12: Checkout Window Testing

(Payment process resulted in failure because we are unable to implement credit card pre-processing portal without obtaining permission

29 statements 26 run 3 missing 0 excluded

```
1 from django.test import TestCase
2
3
4 # Create your tests here.
5 class URLTests(TestCase):
6     def test_homepage(self):
7         response = self.client.get('/')
8         self.assertEqual(response.status_code, 200)
9
10    def test_shopPage(self):
11        response = self.client.get('/shop/')
12        self.assertEqual(response.status_code, 200)
13
14    def test_BootPage(self):
15        response = self.client.get('/category/Boot')
16        # page redirection
17        self.assertEqual(response.status_code, 301)
18
19    def test_HighHeelsPage(self):
20        response = self.client.get('/category/heels')
21        # page redirection
22        self.assertEqual(response.status_code, 301)
23
24    def test_LowHeelsPage(self):
25        response = self.client.get('/category/lowheels')
26        # page redirection
27        self.assertEqual(response.status_code, 301)
28
29    def test_SpikesPage(self):
30        response = self.client.get('/category/Spikes')
31        # page redirection
32        self.assertEqual(response.status_code, 301)
33
34    def test_FormalPage(self):
35        response = self.client.get('/category/men')
36        # page redirection
37        self.assertEqual(response.status_code, 301)
38
39    def test_LadiesPage(self):
40        response = self.client.get('/category/ladies')
41        # page redirection
42        self.assertEqual(response.status_code, 301)
43
44    def test_loginPage(self):
45        response = self.client.get('/category/accounts')
46        # page redirection
47        self.assertEqual(response.status_code, 301)
48
```

Figure 13 : Code segment from a test class


```

▶ Run pip install coverage
Collecting coverage
  Downloading coverage-6.0.2-cp38-cp38-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.whl (256 kB)
Installing collected packages: coverage
Successfully installed coverage-6.0.2
Creating test database for alias 'default'...
...../opt/hostedtoolcache/Python/3.8.12/x64/lib/python3.8/site-packages/django/views/generic/list.py:86: UnorderedObjectListWarning: Pagination may yield inconsistent results with an unordered
object_list: <class 'core.models.Item'> QuerySet.
    return self.paginator_class(
.
System check identified no issues (0 silenced).
-----
Ran 10 tests in 0.004s

OK
Destroying test database for alias 'default'...
Name                               Stmts  Miss  Cover
-----
core/__init__.py                     0      0  100%
core/admin.py                       40      4   90%
core/forms.py                       18      0  100%
core/migrations/0001_initial.py       8      0  100%
core/migrations/0002_auto_20191105_0426.py   5      0  100%
core/migrations/0003_auto_20200412_1441.py   5      0  100%
core/migrations/0004_auto_20200412_1510.py   4      0  100%
core/migrations/0005_item_stock_no.py    4      0  100%
core/migrations/0006_slide.py          4      0  100%
core/migrations/0007_auto_20200510_2016.py   4      0  100%
core/migrations/__init__.py           0      0  100%
core/models.py                      118     25   79%
core/templatetags/cart_template_tags.py   10      5   50%
core/templatetags/category_template_tags.py 37      8   78%
core/templatetags/slide_template_tags.py  11      1   91%
core/tests.py                        34      0  100%
core/urls.py                         4      0  100%
core/views.py                       228    177   22%
demo/__init__.py                     0      0  100%
demo/settings.py                     39     10   74%
demo/urls.py                         8      2   75%
manage.py                             9      2   78%
-----
TOTAL                               590    234   60%

```

Figure 14 : Coverage report from GitHub