

Brokerage: A web-based application connecting stores and customers
Graduation Project 2

Project report submitted in partial fulfillment of the requirement for the award of the Degree
of Bachelor of Science in Information Technology

By

Bandar Khalid Alyahya – ID: 391101290

Abdulmalik Ibrahim Alobaid – ID: 391101269

Abdulaziz Hamad Al-Abduljabbar - ID: 391101265

**Under the Supervision of
Mr. Ahsan Ahmed**



**Department of Information Technology
College of Computer and Information Sciences
Majmaah University
Al Majmaah
Kingdom of Saudi Arabia**

Third Semester 2022-2023

CERTIFICATE

This is to certify that the project report entitled “**Brokerage: A web-based application connecting stores and customers**” being submitted by

Bandar Khalid Alyahya – ID: 391101290

Abdulmalik Ibrahim Alobaid – ID: 391101269

Abdulaziz Hamad Al-Abduljabbar – ID: 391101265

in partial fulfillment for the award of the Degree of Bachelor of Science in Information Technology to the Majmaah University, Al Majmaah, Kingdom of Saudi Arabia during the academic year 2022-2023 is a record of authentic work carried out under my supervision and guidance and it has not formed the basis for the award of any Degree/Diploma/Associateship/Fellowship or other similar title to any candidate of any University.

Signature of Supervisor

Name: Mr. Ahsan Ahmed

Countersigned by

Head of the Department

Seal of the College

DECLARATION

We, **Bandar, Abdulmalik, and Abdulaziz** hereby declare that the project report entitled **“Brokerage: A web-based application connecting stores and customers”** submitted to Majmaah University, Al Majmaah, Kingdom of Saudi Arabia, in partial fulfillment for the award of the Degree of Bachelor of Science in Information Technology during the academic year 2022-2023 is a record of authentic work carried out under the supervision and guidance of Mr. Ahsan Ahmed and it has not formed the basis for the award of any Degree/Diploma/Associateship/Fellowship or other similar title to any candidate of any University.

Signature of Student(s)

Bandar Khalid Alyahya – ID: 391101290

Abdulmalik Ibrahim Alobaid – ID: 391101269

Abdulaziz Hamad Al-Abduljabbar – ID: 391101265

ACKNOWLEDGEMENT

The team would like to express their gratitude and appreciation to the College of Computer and Information Sciences, Majmaah University for providing the permission and various hardware/software resources for completing the project successfully.

Special thanks to our honorable rector **Dr. Saleh Abdullah Almuzel**, Majmaah University for his constant help and for providing us with all facilities needed for this project work.

Our sincere gratitude to **Dr. Mohammed AlShehri**, Dean, College of Computer and Information Sciences, Majmaah University for supporting and encouraging the project work in every phase.

Sincere thanks to **Dr. Ahmed Alhuseen, Vice Dean (Academic Affairs)** College of Computer and Information Sciences, Majmaah University for his continued support in completing this project successfully.

Our special thanks to our supervisor **Mr. Ahsan Ahmed**, College of Computer and Information Sciences, Majmaah University. His inspiring guidance and valuable advice and continuous encouragement at every stage in the progress of this project work enabled us to complete the project on time.

Bandar Khalid Alyahya – ID: 391101290

Abdulmalik Ibrahim Alobaid – ID: 391101269

Abdulaziz Hamad Al-Abduljabbar – ID: 391101265

Table of Contents

1. Abstract	8
2. Introduction	9
3. Literature review	13
4. Motivation of the project	17
5. Scope of the project	17
6. Problem statement	17
7. Project baseline requirement	18
8. Expected Outcomes	18
9. Functional requirement	19
10. Non-functional requirement	19
11. Hardware and Software requirement	20
12. Class diagram	21
13. E-R diagram	22
14. Activity diagram	23
15. Use case diagram	24
16. Gantt chart	25
17. Design	26
18. Development.....	35
19. Testing	40
20. Conclusion and References	45

Table of Figures

Figure 1: Various criteria for a good web application	10
Figure 2: Architecture of MVC	11
Figure 3: Wastetco.....	14
Figure 4: Speero.....	15
Figure 5: Haraj.....	16
Figure 6: Class diagram	21
Figure 7: Entity Relationship diagram	22
Figure 8: Activity diagram	23
Figure 9: Use Case diagram	24
Figure 10: Gantt Chart	25
Figure 11: Login	26
Figure 12: Register.....	26
Figure 13: Store Registration	27
Figure 14: Home Page (Admin)	27
Figure 15: Home Page (User)	28
Figure 16: Subcategory Page.....	28
Figure 17: Manage Users	29
Figure 18: Create New User	29
Figure 19: Manage Roles	30
Figure 20: Create New Role	30
Figure 21: Manage Categories	31
Figure 22: Create New Category	31
Figure 23: Create New Subcategory	32
Figure 24: Subcategory Management	32

Figure 25: Manage Stores.....	33
Figure 26: Create New Store	33
Figure 27: Show Store	34
Figure 28: Manage Orders	34
Figure 29: Show Order	35
Figure 30: Order Summary	35
Figure 31: Types of Testing	40
Figure 32: Password do not match	41
Figure 33: Password must be at least 8 characters	41
Figure 34: Email Validation	41
Figure 35: Role updated successfully by Admin	42
Figure 36: Store Created Successfully	42
Figure 37: The Papers field must be of type : jpg, jpeg, png, pdf only	43
Figure 38: Deleted User Successfully	43
Figure 39: Papers Validation	44
Figure 40: Received Store Registration request	44

Abstract:

The usage of the internet around the globe has seen a great demand in the world of Information and Communication Technology (ICT). The number of internet and electronic device users has been increasing rapidly during the last two decades. During the last few years, the demand for web applications has increased, and people make use of online services to access their requirements. The main purpose of this project is to build a website that will link all telecommunications stores with customers. Instead of going physically to the stores and searching for the right price for the required thing, requires a lot of time and effort. This web-based application minimizes the time and removes the gap between users and stores and thus finally reducing the user's efforts. While sitting at a location, users can reach all stores and can order as per their requirements. This web application is developed based on Model View Controller (MVC) architecture using the programming language Hypertext Preprocessor (PHP). To make the work easier, the Laravel framework is used. Developing the project in modules will provide scalability to this project. The tracking module of this web application will allow the users to monitor their order requests. Interactive and easy simple interfaces will display all the main categories available in the stores and start to go deeper little by little until the customer completes his order.

Keywords: Information and Communication Technology (ICT); Hypertext Preprocessor (PHP); Model View Controller (MVC)

Introduction:

All internet users use online services to perform their daily activities, whether it is normal tasks or is social arrangements or personal recreation. Most online users hold a high opinion of the Internet as a place to conduct everyday tasks and pursue the everyday pleasures of life, such as checking the weather, doing their banking, communicating with friends and family, and playing games [1]. Some of the uses of the internet in our life are as follows: use of internet in Education; online shopping; Research and Development; business promotion and Innovation; Communication; Digital Transactions; financial transactions; tour and travel, etc. [2]. During the last two decades, rapid development has been observed in the development and usage of web-based application. The web application is an application program that is usually deployed on a local/remote server, and users can access it using client software known as a web browser by its domain name. The common benefits of Web applications are as: allows multiple users access to the same version of an application; it does not need to be installed on individual machines; web apps can be accessed on multiple platforms; and can be accessed through multiple browsers i.e., Mozilla, Safari, Chrome, etc. [3]. The various criteria for a good web application are shown in figure 1 and shown below as follows:

- The availability of necessary resources
- Meet User Demands Quickly [5]
- Reply to Customer Feedback Wherever It Is Provided
- Include the Latest and the Greatest Technologies [4]
- Easy to Use
- Optimized for Search and the Social Web [6]



Figure 1: Various criteria for a good web application [7]

Model-View-Controller (MVC) framework is an architectural pattern that separates an application into three main logical parts: first Model, second View, and Controller. MVC helps to create projects that can be expanded and developed in the future [8-9]. The Model is the backend that contains all the data logic, so it stored the data and its related logic. The view is quite simply a user interface that displays data in front of the user. The Controller is responsible for interactions between the system and the user in general, and we can say that he is the mastermind of the application [10]. The simple architecture of MVC is depicted in figure 2. There are many MVC web frameworks. Some of the common frameworks are Laravel, Yii, SpringMVC, and Catalyst. These frameworks have the features of URL Routing for SEO Friendly URLs. Additionally, they are powerful with URL-mapping, and have good control of URL and HTML, testing gets easier using such type of framework.

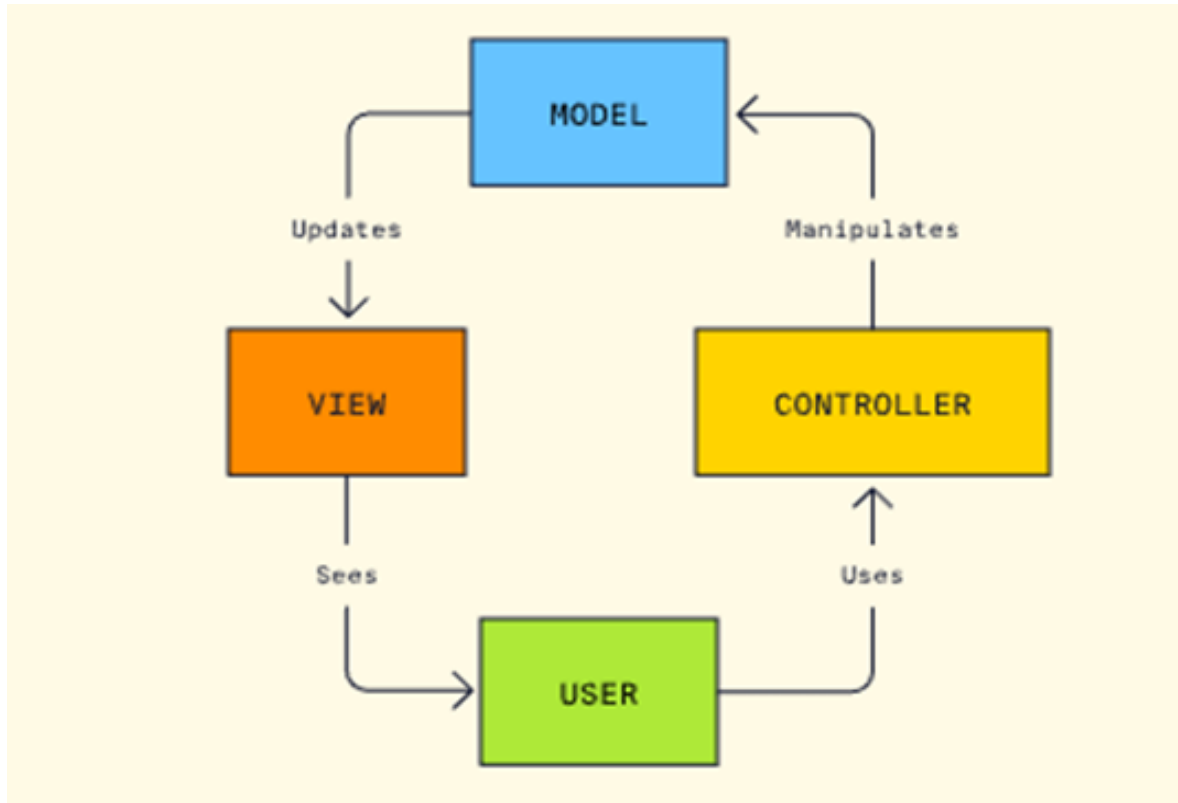


Figure 2: Architecture of MVC [11]

MVC architecture provides a clean separation of concerns (SoC), All classes and objects are independent of each other so that you can test them separately, only uses a Front Controller pattern which processes web application requests through a single controller. Easier support for new types of clients, the design allows logical grouping of related actions, etc. [9]. There are different versions of MVC: Model-view-presenter (MVP), Model-view-adapter (MVA), Model-view-viewmodel (MVVM), and Hierarchical model-view-controller (HMVC) [8].

Laravel is a web application framework with expressive, elegant syntax. A web framework provides a structure and starting point for creating your application, allowing you to focus on creating something amazing while Laravel sweats the details. Laravel strives to provide an amazing developer experience while providing powerful features such as thorough dependency injection, an expressive database abstraction layer, queues and scheduled jobs, unit, and integration testing, and more. Whether you are new to PHP web frameworks or have years of experience, Laravel is a framework that can grow with you. It will help you take your first steps as a web developer or give you a boost as you take your expertise to the next level. Laravel is a “progressive” framework. it means that Laravel grows with you. If you are just taking your first steps into web

development, Laravel's vast library of documentation, guides, and video tutorials will help you. If you are a senior developer, Laravel gives you robust tools for dependency injection, unit testing, queues, real-time events, and more. Laravel is fine-tuned for building professional web applications and is ready to handle enterprise workloads. Laravel is incredibly scalable. Thanks to the scaling-friendly nature of PHP and Laravel's built-in support for fast, distributed cache systems like Redis, horizontal scaling with Laravel is a breeze. In fact, Laravel applications have been easily scaled to handle hundreds of millions of requests per month. Laravel combines the best packages in the PHP ecosystem to offer the most robust and developer-friendly framework available [12].

Laravel is a free and open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller (MVC) architectural pattern and based on Symfony. Some of the features of Laravel are a modular packaging system with a dedicated dependency manager, diverse ways for accessing relational databases, utilities that aid in application deployment and maintenance, and its orientation toward syntactic sugar. The MIT License is a permissive free software license originating at the Massachusetts Institute of Technology (MIT) in the late 1980s. As a permissive license, it puts only extremely limited restrictions on reuse and has, therefore, high license compatibility [13].

This web-based application will link all telecommunications stores with customers by providing an adjustable and easy-to-use interface. When the Customer enters our website, he/she will need to sign in by providing his/her name, phone number, and email. The main page will display the main categories that are provided by the stores such as: Devices, cables, accessories, Power banks, etc. when the user chooses one of the main categories, the website will display the subcategories that are related to the chosen category and the user will continue choosing one category after another until his/her order is completed. Let us take an example, If the user chooses the main category devices, subcategories will be displayed like mobile phones, laptops, tablets, printers, etc. let us say that he/she chooses mobile phones, the most popular mobile phones brands will be displayed like apple, Samsung, Huawei, Nokia, etc. After choosing one of these brands, it will display the common models for this brand. So that if apple is chosen, these models will be displayed: iPhone 14, iPhone 13 pro, iPhone 12 pro max, etc. After the user chooses the model, he will be requested to choose the storage capacity (128GB, 256GB, 512GB, etc.). Then the user will choose the color that the user wants for his/her mobile phone. After that, the User will choose the

status of the mobile phone either New or used. Then a summary of the user's order will be displayed, and the user will confirm the order or modify it. If the user confirmed the order, then automatically the order details will be sent to the stores that have been registered with us along with the customer contact details. The stores will receive the order by email and check the stock, If the requested item is available or not, and then will reply to the customer either by email or phone number (SMS, WhatsApp) and offer the item price along with its details. That was the customer's view of our website. On the other side, the stores can register on our website by applying a request through a page that contains a form of necessary details that are requested from the store owners. After the stores apply their request, the administrators will review their requests and check their credibility, and the approved stores will be added to the store's table in our database. Finally, there will be a page for the administrators that called "Dashboard". In this page, the admin can manage the website and the categories. The main modules of this website will be the registration and login modules, categories module, Stores Subscription module, and dashboard module.

Literature Review

A lot of work has been done in the field of development, and a lot of projects have been developed for maintaining inventories related to stores. This section highlights some of the related works which have been done in recent years. Wastetco website is an Egyptian company that was developed in September 2020. Wastetco is an intermediary service website between the seller and the buyer that works to guarantee the rights of both parties in all large and small electronic financial transactions. After the delivery of the transaction and the approval of the satisfaction of both parties, a 5% commission is deducted from the buyer upon the completion of the entire transaction, and it is deducted from the value of the amount as expenses for the website, and the remaining amount is delivered to the seller, after the successful delivery and completion of the transaction [14].

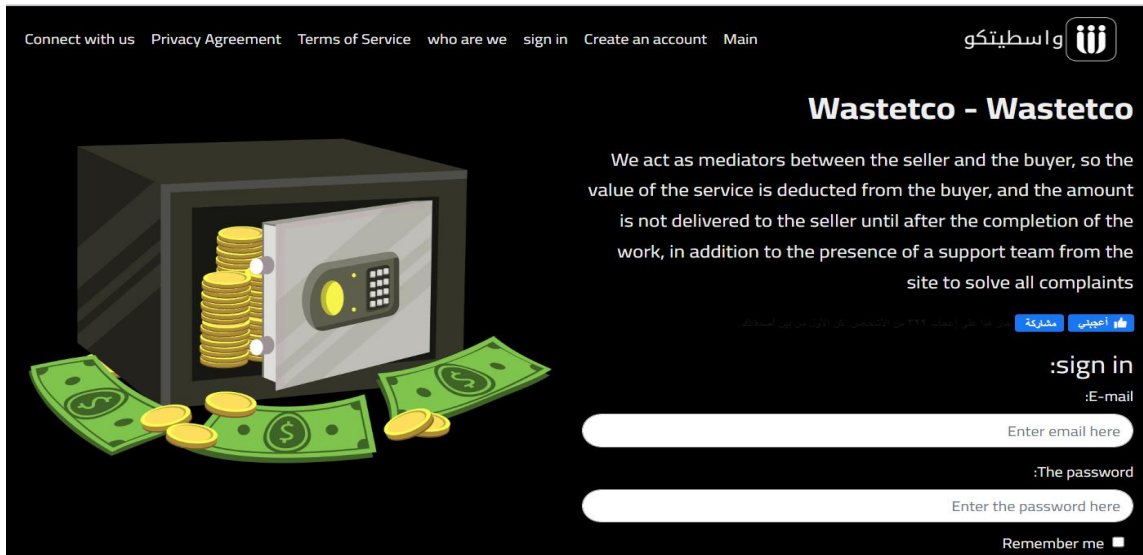


Figure 3 Wastetco

One of the disadvantages of this website is that it interferes in the buying and selling process not done directly between the seller and the buyer, where the buyer pays the amount of the service to the website, then the site takes its commission of the amount and then delivers the remaining amount to the seller. Another disadvantage of this website is that once the service is requested, the amount of the service is deducted directly from the buyer's account, and it is better that the amount be deducted after completing the required service and requires the buyer's approval for the same. Also, for the seller, the seller is requested to provide his service and specify the amount and all the details of the service. Which is considered negative if the seller has more than one service, so it is requested to provide each service alone with all its details and update the details and amount of the service continuously. Another example is Speero [15], which is a Saudi company that was developed in 2017 by Abdullah bin Shamlan and Amin Mahfouz. It links the customer and the merchant and is specialized in providing spare parts solutions to the customer in an easier way. It provides three types of services. The first service is that the merchant who owns the spare parts can register on the site and enters all the required data to be included in the database of Speero. The second service is that Speero owns warehouses for new auto spare parts. The customer can find a catalog of a few car brands, such as Toyota, Nissan, and others. This helps you, as a customer, to view all the details of your car parts (in the catalog), which make it easy for the customers that have no experience in auto mechanics. The third service help the customer to find the perfect price for the product that he/she wants by providing the parts name and details, and

Speero sends your request to all shop owners participating on the website, as is the case with our project, but there are many problems with this site. First, you as a customer, must know the exact name of the piece, and you do not have a catalog to know the exact shape or name of the piece, and therefore there is a significant percentage of error or that the piece is not suitable. Secondly, Speero requires you to pay an amount of ten riyals at the beginning. The reason, as they put it, is to ensure the seriousness of the customer, but this means that the customer must have a bank card to make this request. Thirdly, and finally, one of the biggest problems is that the customer cannot order more than one product at time, which means that the customer need to order each one separately. Which also means that each pricing process to request a piece, an amount of 10 riyals must be paid, so if we say that you have 5 spare parts for the car that you want to search for, you will pay an amount of 50 riyals just to inquire, and this is a high amount, especially if the customer has a large number of spare parts.

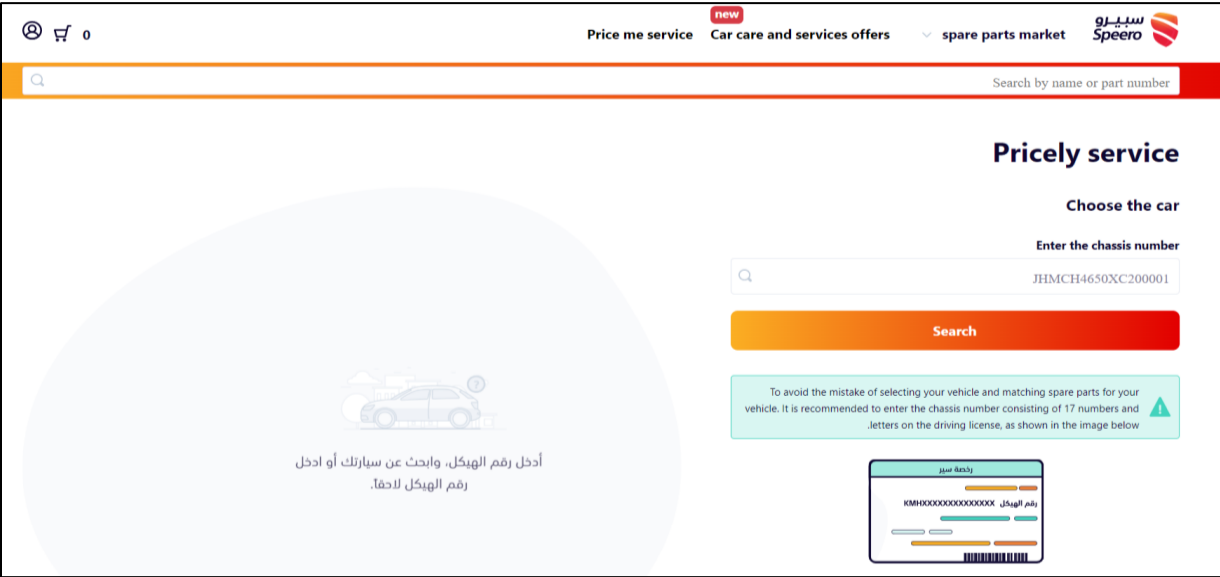


Figure 4 Speero

Haraj [16] is an e-commerce platform founded in 2007 by Yousef Al-Rashidi. The website serves two parties, the seller, and the buyer. This platform enables the sellers to make announce for their commodity, product, or anything they wants to sell, and they should give full details of this product and pictures of it, and they may set a fixed price or leave it for bidding. As for the user or the buyer, they can search for the desired item, or choose from the provided categories to display comparable results. The platform is free for the seller and the buyer, but it takes a commission

from the seller for every sale made through this platform. The idea of our website is like this platform, the only difference is that instead of the seller presenting his item or product, the buyer or the user is the one who submits his request for the product he needs, and his request reaches all the stores registered with us. The problem with Haraj platform is that the seller is obligated to provide all the details of the product he wants to offer, determine its price, and update his announcements in case of any changes in the product or the price. However, many sellers do not abide to that, and they publish their announcement incomplete and do not contain the full product details. Another disadvantage is that the sellers are not only shop owners, but they are also individuals, and some products may not be authorized or not guaranteed. Also, for the customer or buyer, he is obligated to search for the required thing, review all the announcements that appears to him, communicate with the sellers, negotiate with them, and many other details that waste the customer time and effort.

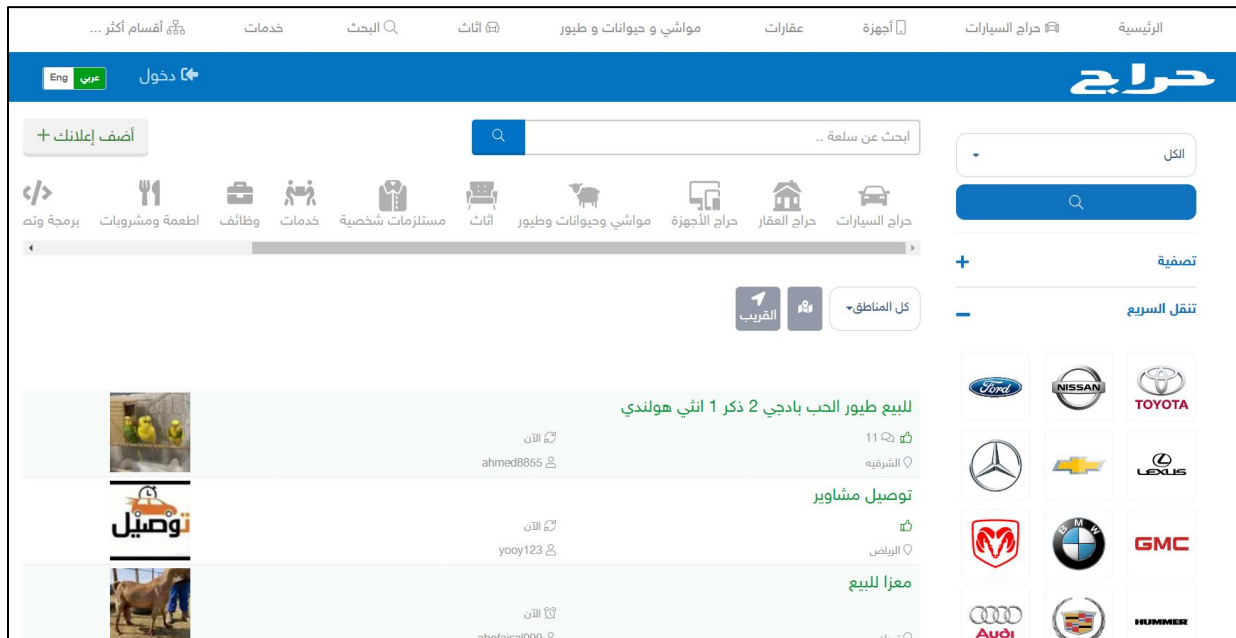


Figure 5 Haraj

Our website is characterized by its ease of use and facilitating the communication between the stores and the customers. As for the stores, it is not obligated to submit any announcements or publish the product and provide its details. All it must do is to be registered with us so that it will be added to the database of stores, and it will receive customers' requests and contact information. As for customers, they are not obligated to search for every product they want and review different

announcements. All they must do is to choose from the categories provided to them until the order is completed, and then the order will be sent to all registered stores. Also, there will be no user fees, only a commission will be taken from the stores for every sale made through the website, after agreeing with the stores on that commission.

The motivation of the project:

To facilitate the users to search for the products, and reduce the effort exerted by stores to publish and market their products. Upon completion of this project, the users will be able to reach the stores and find the products easier and faster, and the stores will also benefit from faster contact from the customers without requiring any effort or cost from them, by receiving an email containing the customer order and his contact info.

Scope of the project:

The main purpose of this project is to facilitate the process of ordering products for customers and to easily link them with the stores without the need to do any advertisements or communicate with the stores separately. It helps in bringing more customers to the stores without the need for marketing or building online stores, which increases the sales, and there are no fees for stores or customers, only a percentage for each sale done through this website. PHP's latest framework named Laravel is used to build the website and it is expected to be completed within the next semester.

Problem statement:

Nowadays, one of the problems that the user(s) faces is that if they want to order a specific product, they must search for the contact numbers of the stores or their website address, and then communicate with all the stores and compare prices and brands to find the perfect product. Recently with digital transformation, all the stores are forced to create online stores and social media accounts to publish their products and market their products through social media platforms such as Twitter, Snapchat, and Google. This requires a lot of effort and cost. Additionally, this process is time-consuming. There is a requirement for a system that can be enough efficient and act as a man in middle for handling the communication between various stores and customers.

Project Baseline requirement

After deep thinking and planning, we found that it is appropriate at the beginning of the project to be limited to a small scale so that it would be easy to provide the service, and problems and errors would be dealt with quickly and remedied. After that, the scope of the project will be expanded to cover more cities and stores. The beginning of the plan will be as follows: The project starts in the city of Majmaah as we are the native of this city and we know the stores and what they need, so it will be a good start as per our knowledge. After the success of this project, the project will expand to cover more stores in various cities of Kingdom of Saudi Arabia. As this a Graduation Project, therefore our main goal will be to complete this project within the next semester of this academic year. The minimum required services like developing the website, database, testing and find the weaknesses in it to avoid attacks and intrusions can be done by the end of the next semester.

Expected Outcomes:

Every work/task have some outcomes, and the expected main outcome of this project is to provide a platform that connect various stores for the users. Correcting the errors, improving the weakness by providing high security to the website, making better performance, correcting the defects that are present in the competitors and distinguishing from them with many features that they lack are the goal of this project. The turnout from users should be extremely high and they are confident in it. The distinction is that the services are limited to enable the service to be provided in the maximum level and the best possible performance. Malfunctions and delays must be avoided. The website will be easy to use due to the uncomplicated way to order, which does not request any inputs from the user, just to choose from the categories.

Functional requirements:

The website will contain several modules, and every module is dedicated to a specific job:

- **Registration and Login Module:**

This module is intended to allow the user to register on the website with his username, email, phone number, and password, Then login through the username and password.

- **Categories Module:**

This module will contain the main categories and all their underling subcategories until the user completes the order.

- **Order Summary Module:**

This module contains the details of the user's order to display before confirming it.

- **Stores Subscription Module:**

This module will enable the stores to join us by applying some necessary information and papers to be accepted and add them to stores database.

- **Dashboard Module:**

This module is for the administrators to manage the website and users accounts and to keep track of the performance of the website.

- **Dashboard Login Module:**

This module is for the administrators to login to the dashboard and it will be separate from the user login.

Non-functional requirements:

Accuracy: The system must provide accurate information to the users.

Security: The system should be secure, and SSL/TLS can be enabled that support secure transactions between client and server. SSH can also be implemented to transfer the files code to the server in a secure way.

Performance: The system should be reliable from crashing and the speed should be fast enough.

Usability: The system should be easy to use and easy to understand for new users.

Scalability: The system should be able to grow to hold the growing users and stores and database records.

Portability: The system should be able to run on all platforms and must be defaced with various browsers and OS.

Hardware requirement:

- Any smart device with at least 4 GB RAM.
- Processor Intel i3 or higher.

Software requirement

Web browser:

Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge, and Microsoft Internet Explorer.

Operating System:

Windows 7 or higher, and Mac OS.

Web Server and database:

XAMPP server is a package software that comes with Apache web server, PHP environment, and MySQL database.

Programming Environment and Platform:

Visual Studio Code Environment and Laravel Platform will be used.

Class Diagram:

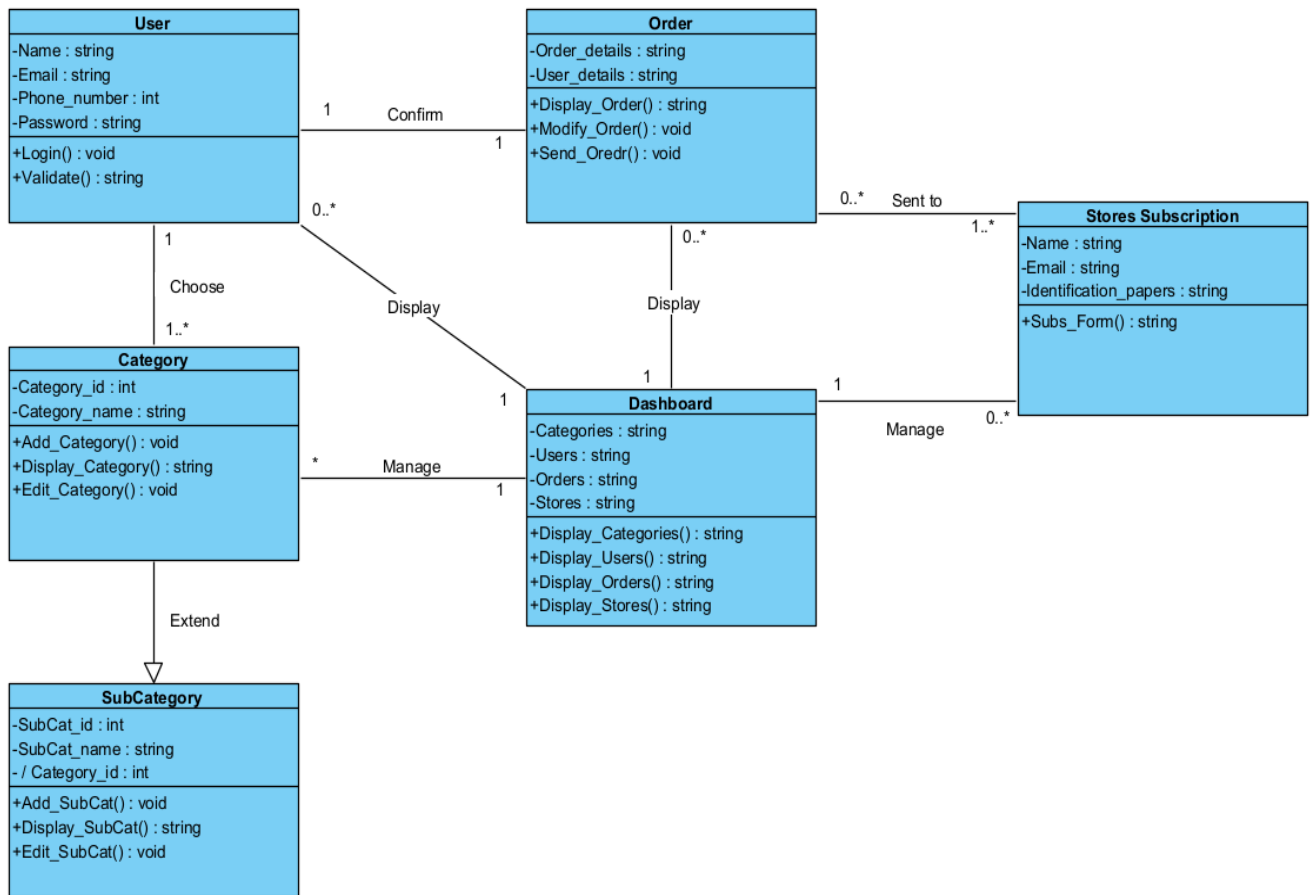


Figure 6 Class diagram

E-R Diagram:

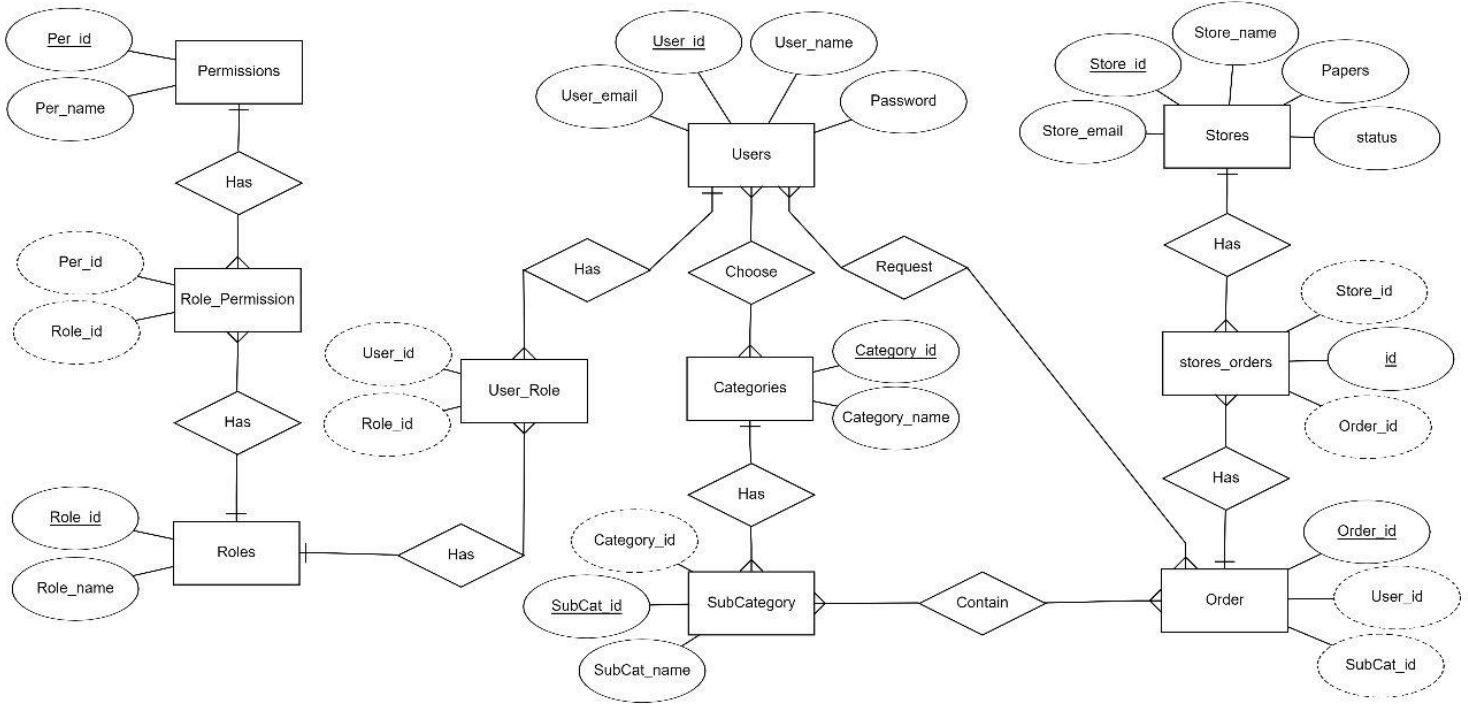


Figure 7 Entity Relationship diagram

Activity Diagram:

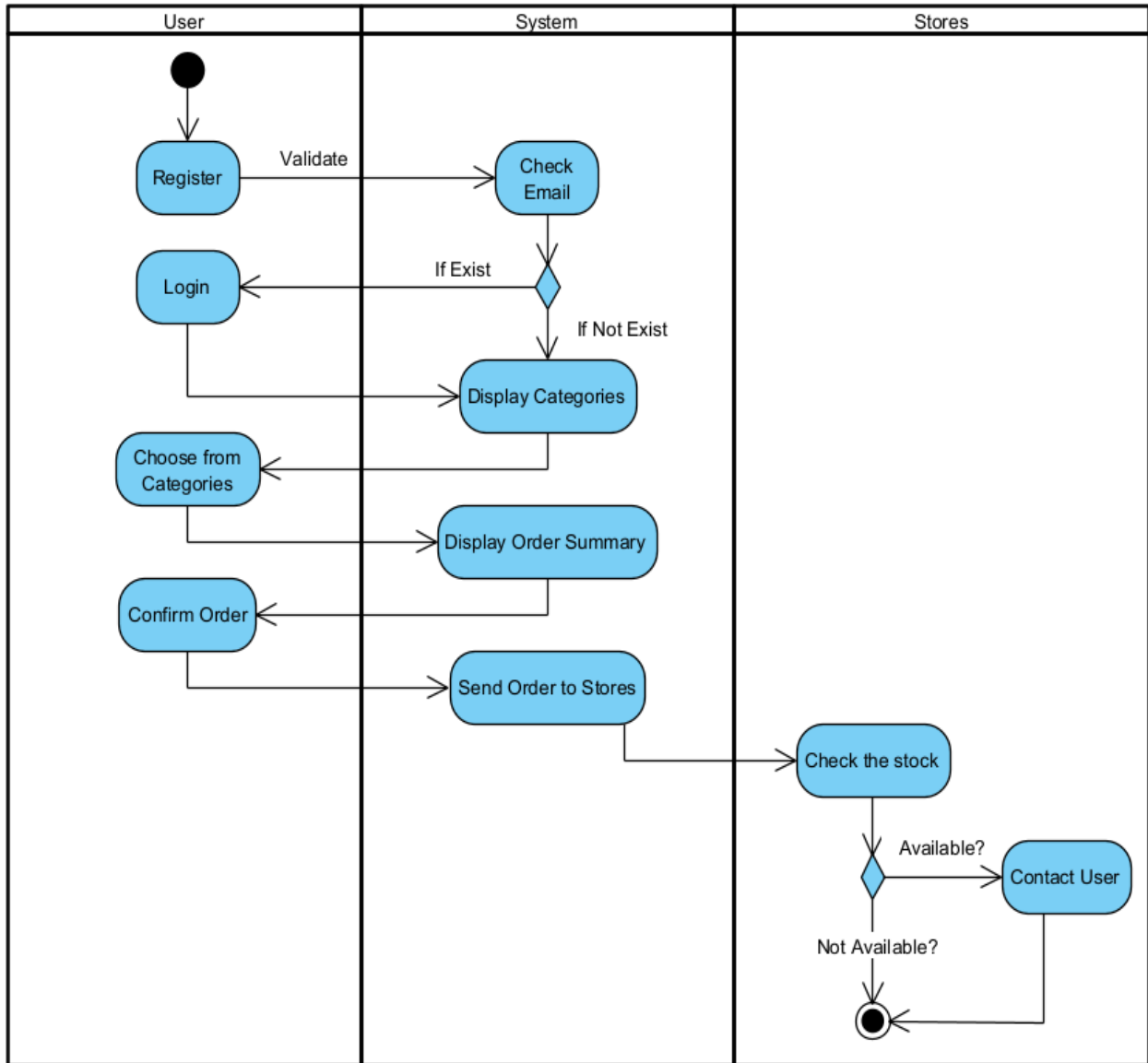


Figure 8 Activity diagram

Use Case Diagram:

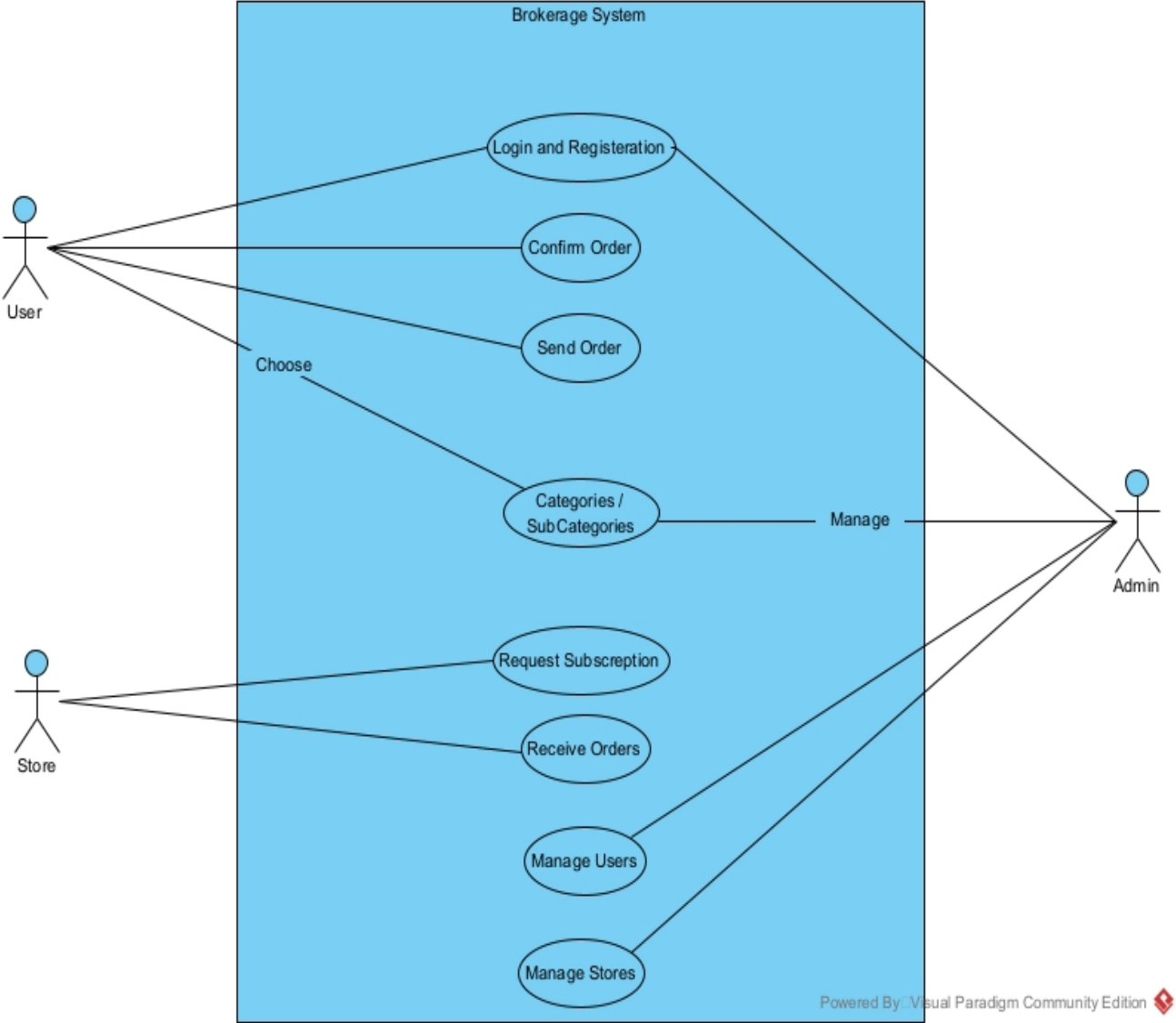


Figure 9 Use Case diagram

Gantt Chart:

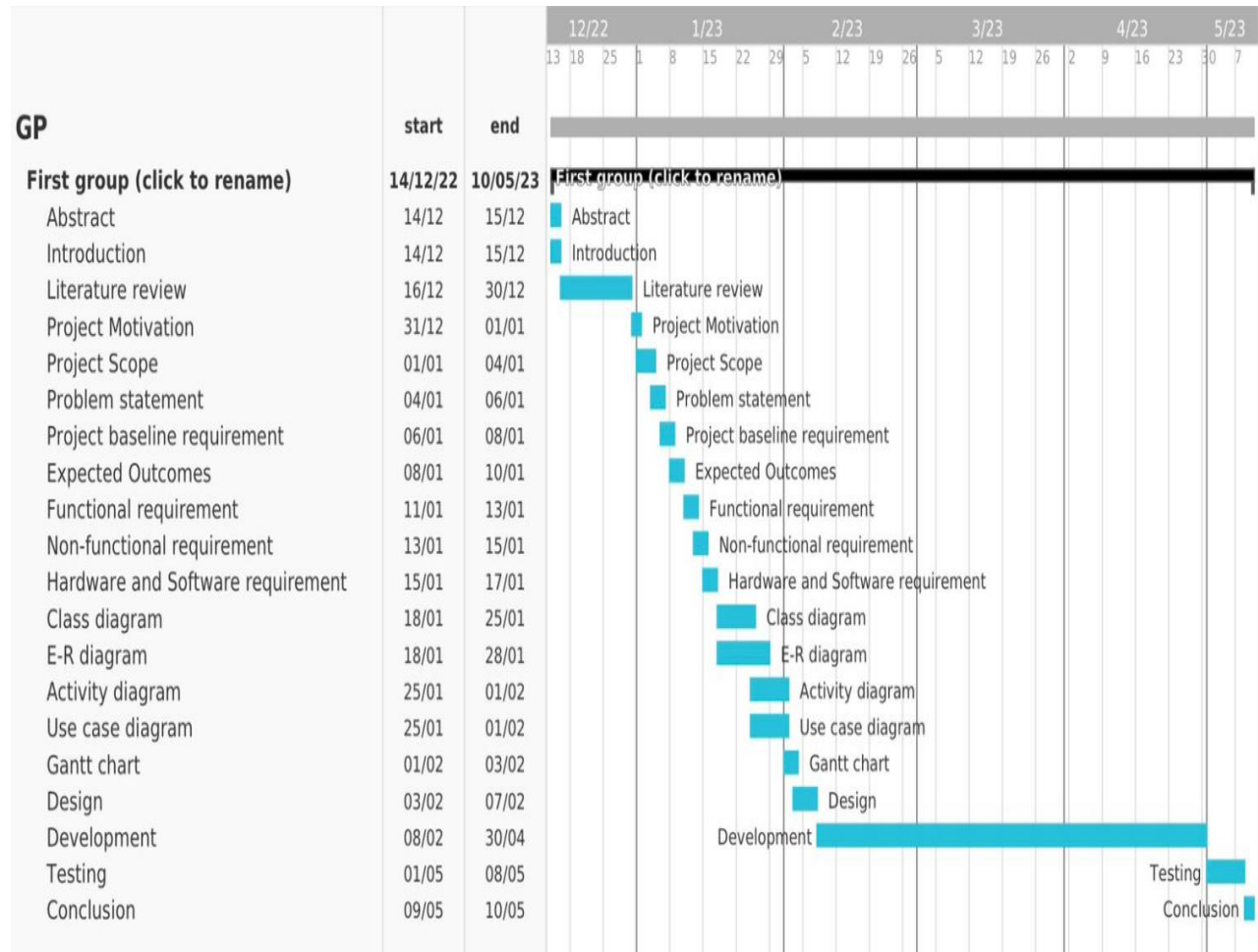


Figure 10 Gantt Chart

Design:

GP2 Home Page Login Register Stores Registration

Login

Email Address

Password

Remember Me

[Forgot Your Password?](#)

Figure 11 Login

GP2 Home Page Login Register Stores Registration

Register

Name

Email Address

Password

Confirm Password

Figure 12 Register

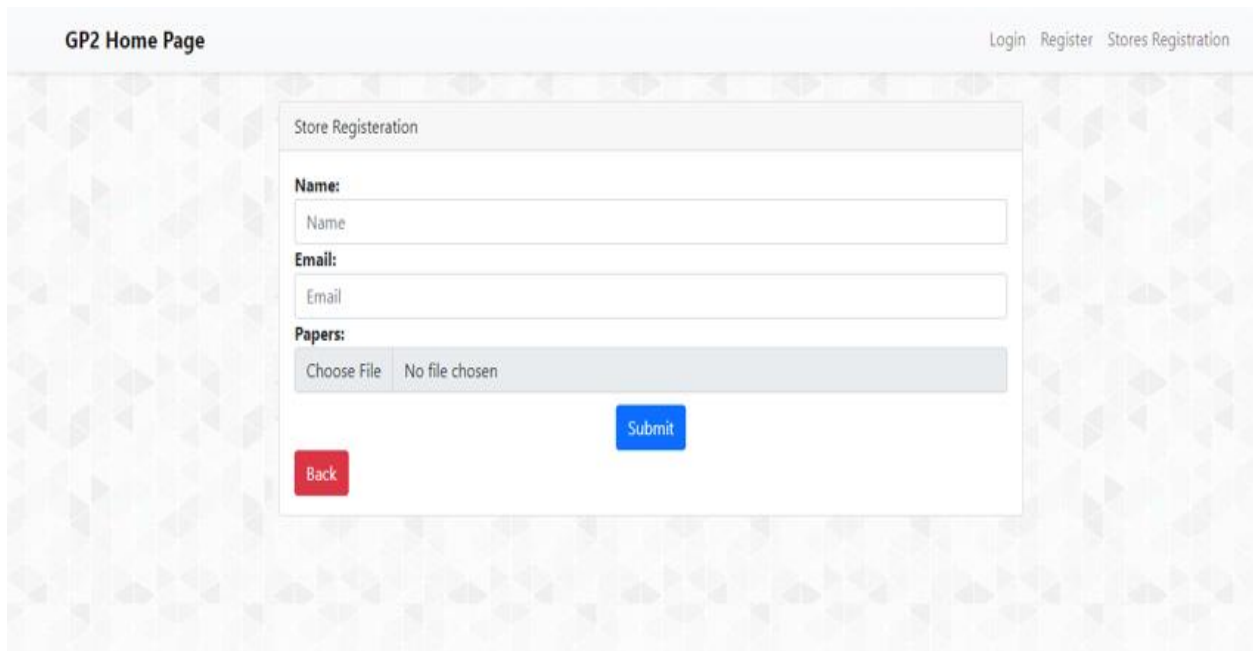


Figure 13 Store Registration

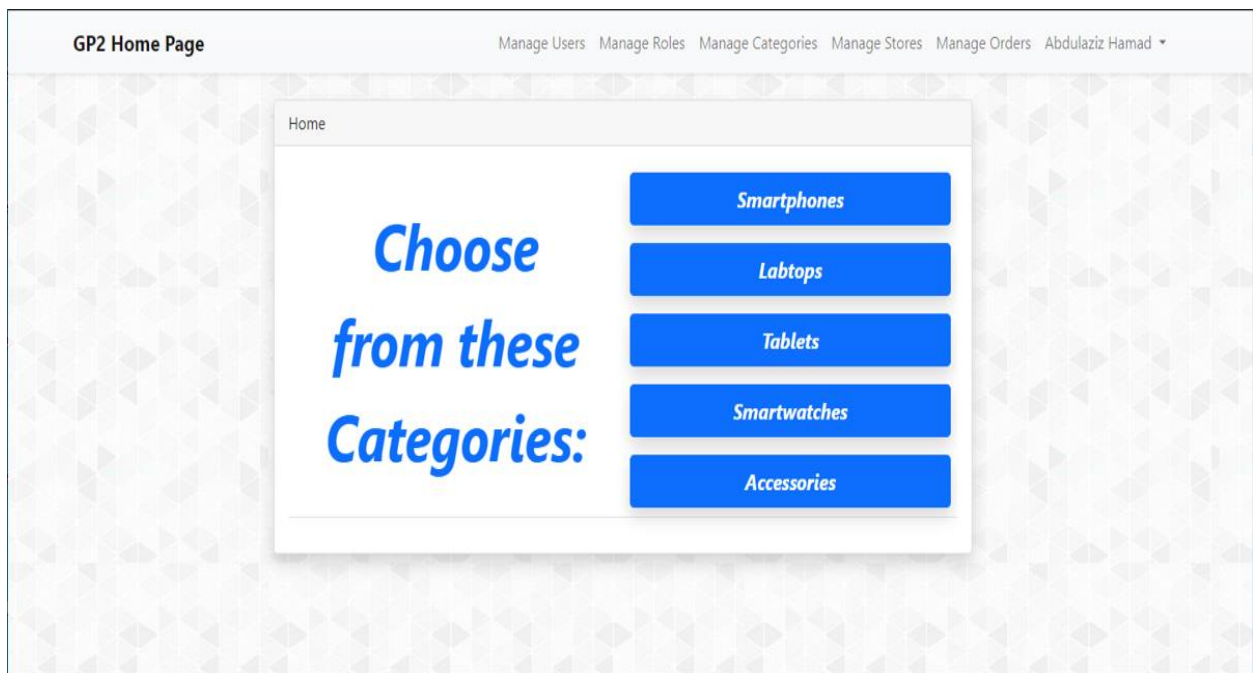


Figure 14 Home Page (Admin)

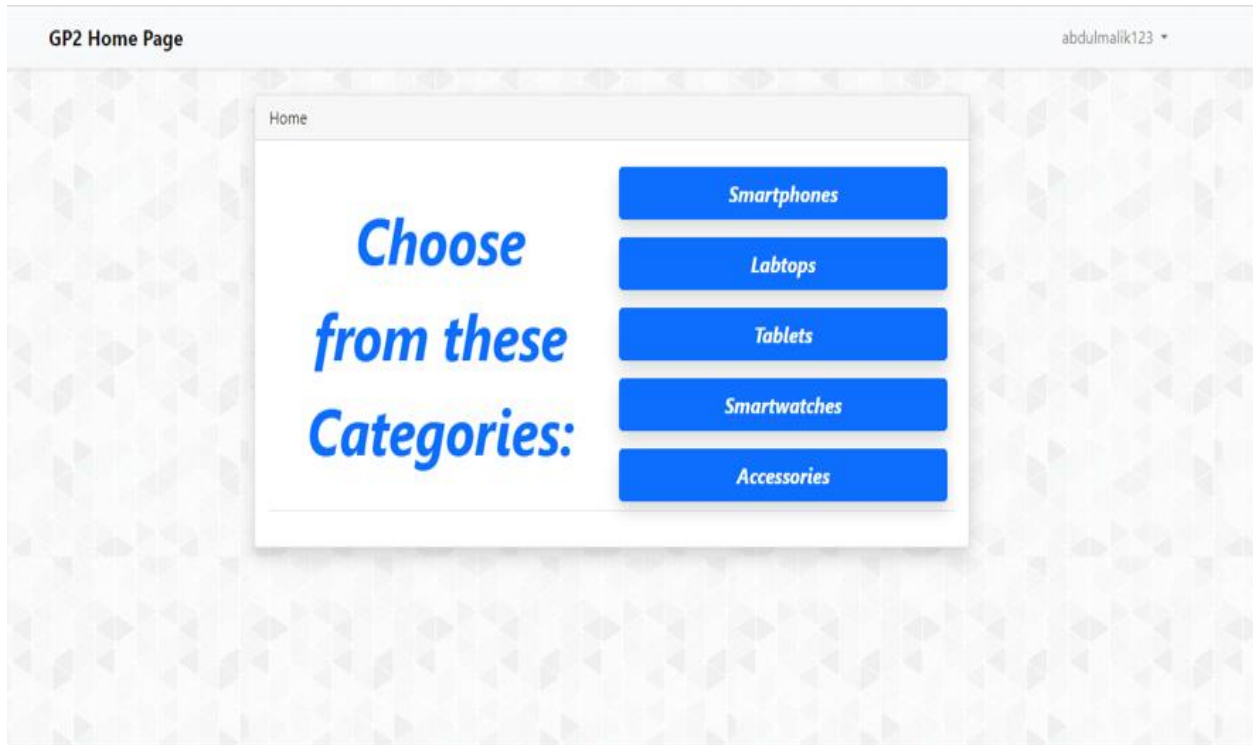


Figure 15 Home Page (User)

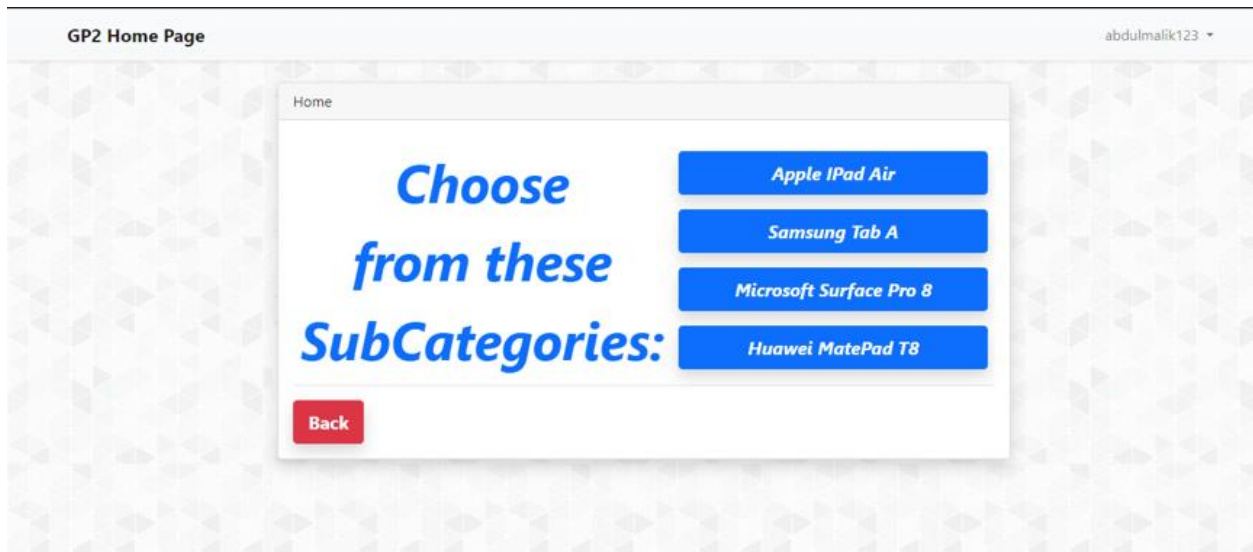


Figure 16 Subcategory Page

GP2 Home Page Manage Users Manage Roles Manage Categories Manage Stores Manage Orders Abdulaziz Hamad ▾

Users Management

Create New User

No	Name	Email	Roles	Action
1	LaravelTuts	admin@laraveltuts.com	Admin	Show Edit Delete
2	Abdulaziz Hamad	azeez142111@gmail.com	Admin	Show Edit Delete
3	Abdulmalik	abdulmalik@gmail.com	Member	Show Edit Delete
4	ahmed	a.ahmed@mu.edu.sa	Member	Show Edit Delete
5	test	test@gmail.com	Member	Show Edit Delete
6	abdulmalik123	test123@gmail.com	Admin	Show Edit Delete

Figure 17 Manage Users

GP2 Home Page Manage Users Manage Roles Manage Categories Manage Stores Manage Orders Abdulaziz Hamad ▾

Create New User

Name:

Email:

Password:

Confirm Password:

Role:

Admin
 Member

[Submit](#)

[Back](#)

Figure 18 Create New User

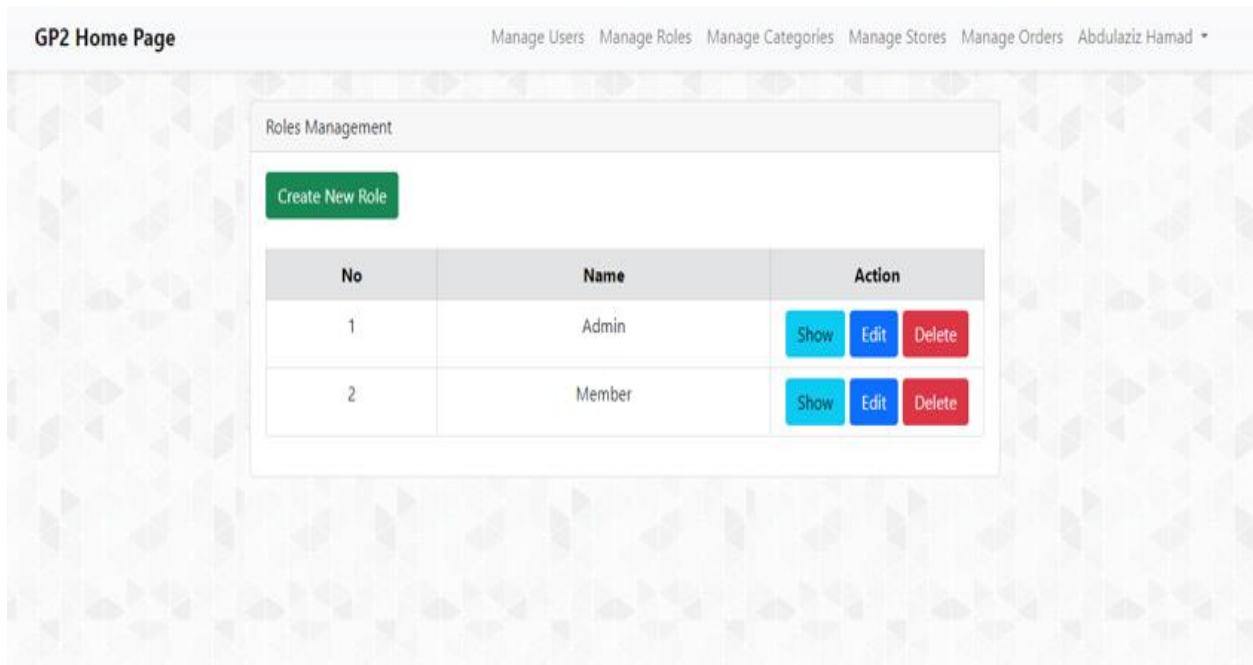


Figure 19 Manage Roles

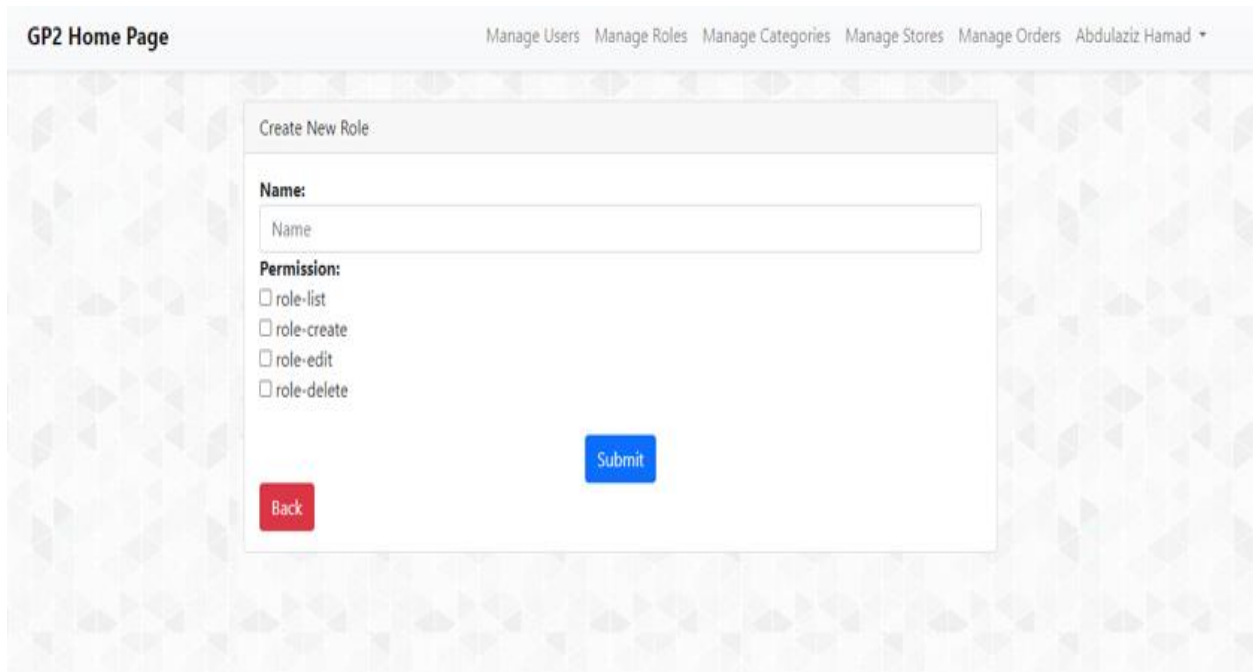


Figure 20 Create New Role

Categories Management

Create New Category Create New SubCategory

No	Name	Action
1	Smartphones	SubCategories Show Edit Delete
2	Laptops	SubCategories Show Edit Delete
3	Tablets	SubCategories Show Edit Delete
4	Smartwatches	SubCategories Show Edit Delete
5	Accessories	SubCategories Show Edit Delete

Figure 21 Manage Categories

Create New Category

Name:

Submit

Back

Figure 22 Create New Category

Create New SubCategory

Name:

Main Category:

- Smartphones
- Laptops
- Tablets
- Smartwatches

[Submit](#)

[Back](#)

Figure 23 Create New Subcategory

GP2 Home Page Manage Users Manage Roles Manage Categories Manage Stores Manage Orders Abdulaziz Hamad ▾

SubCategories Management

No	Name	Action
1	IPhone 14 pro max	Show Edit Delete
2	IPhone 14	Show Edit Delete
3	Samsung Galaxy S20	Show Edit Delete
4	Samsung Galaxy S20 Ultra	Show Edit Delete

[Back](#)

Figure 24 Subcategory Management

Stores Management

Create New Store

No	Name	Email	Status	Action
1	32iz	azeez142111@gmail.com	Approved	Show Edit Delete
2	test	test@gmail.com	Not Approved	Show Edit Delete
3	test2	test2@gmail.com	Not Approved	Show Edit Delete
4	test3	test3@gmail.com	Not Approved	Show Edit Delete
5	test5	test5@gmail.com	Not Approved	Show Edit Delete
6	test6	test6@gmail.com	Not Approved	Show Edit Delete
7	ahsan	a.ahmed@mu.edu.sa	Not Approved	Show Edit Delete
8	store1	store123@gmail.com	Approved	Show Edit Delete

Figure 25 Manage Stores

Create New Store

Name:

Email:

Papers:

Status: Not Approved Approved

Figure 26 Create New Store

Show Store

Name: 32iz
Email: azeez142111@gmail.com
Papers:

Popular Brand label	Product categories	Brand	Family labels
Security vendor/ analysis			Do you want to automate checks?
Almalayzi	Tripal/Win Security 8427321	Almalayzi	Tripal/Cluster Win32/Trojan 33884348
AlMalayzi	Gen/Variant/Gauche 848808	Amey AVL	Tripal/Win32/EGeneric
Almalayzi	Tripal/Gauche 828282	Amey	Win32/Malware/gem
AVIS	Win32/Malware/gem	Amey (no cloud)	78-Virus/Amey
Baidu	Win32/Tripal/Cluster/Agent 2	EGeneric	Gen/Variant/Gauche 848808
BitDefender/Trust	Gen/Mal/Dropt 34308 amey/GEN/1	Blue Fin	Win32/Adware/Amey 81
ClamAV	Win/Malware/Emoney 897789 0	CloudScan/Paros	Win32/malware_gen/Amey_1084 (3)
ClamAV	Amey	Clam	Malware (no cloud 18)
Cyren	Win32/Suspicious 84 6m/EGeneric	Drift	Tripal/Cluster 14313
DrWeb	Malware (high confidence)	EGeneric	Gen/Variant/Gauche 848808 (2)
eforum	Gen/Variant/Gauche 848808	EMF/AD32	Win32/Tripal/Cluster/Agent 8434
F-secure	Win32/Malware/Amey	EGeneric	Gen/Variant/Gauche 848808
Geminet	EGeneric	Amey	Tripal/Win32/Generic
Jiangmin	Tripal/Amey 8282	K/Function	Amey (no cloud 18)
K7Anti	Amey (no cloud 18)	Kaspersky	Tripal/Win32/Agent 8434
Level	Tripal/Malware/Amey	SASL	Malware (no cloud 18)
Malware	Tripal/Malware 178101 amey	Malware	Gen/Variant/Gauche 848808
McAfee-integration	Malware/Amey 8282/Amey 8282	Malware	Tripal/Win32/Trojan/Amey
NANOAntivirus	Tripal/Win32/malware	Win32/Amey	Amey (no cloud 18)
Nano	Tripal/Amey 8282/Amey 8282	Amey (no cloud 18)	Tripal/Win32/Cluster 8434
Avira	Amey (no cloud 18)	Amey (no cloud 18)	Amey (no cloud 18)

Status: Approved

[Back](#)

Figure 27 Show Store

Orders Management

Order ID	Subcategory ID	User ID	Action
102	14	2	Show Delete
103	14	2	Show Delete
104	14	2	Show Delete
105	14	2	Show Delete
106	14	2	Show Delete
107	14	2	Show Delete
108	15	2	Show Delete
109	22	2	Show Delete

Figure 28 Manage Orders

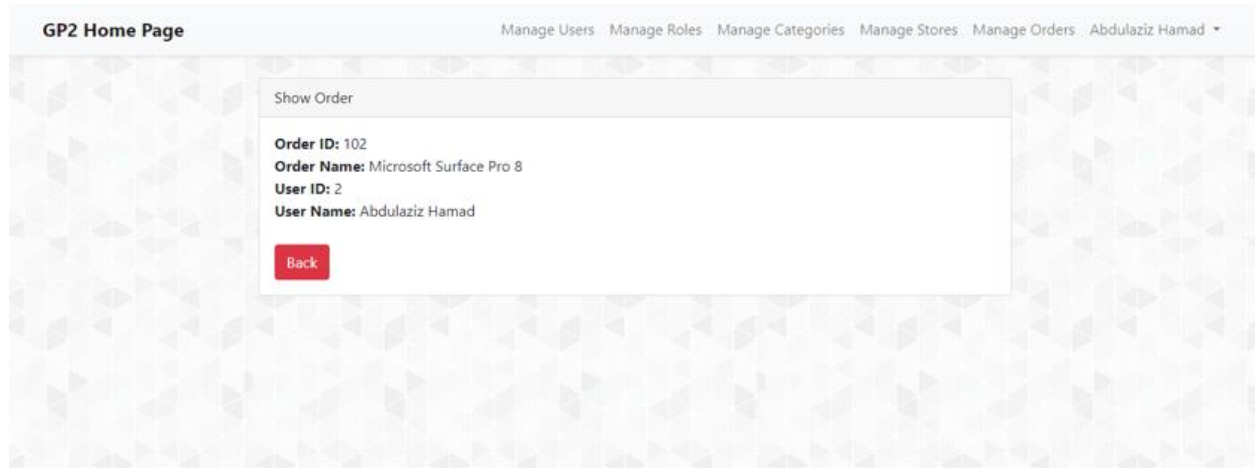


Figure 29 Show Order

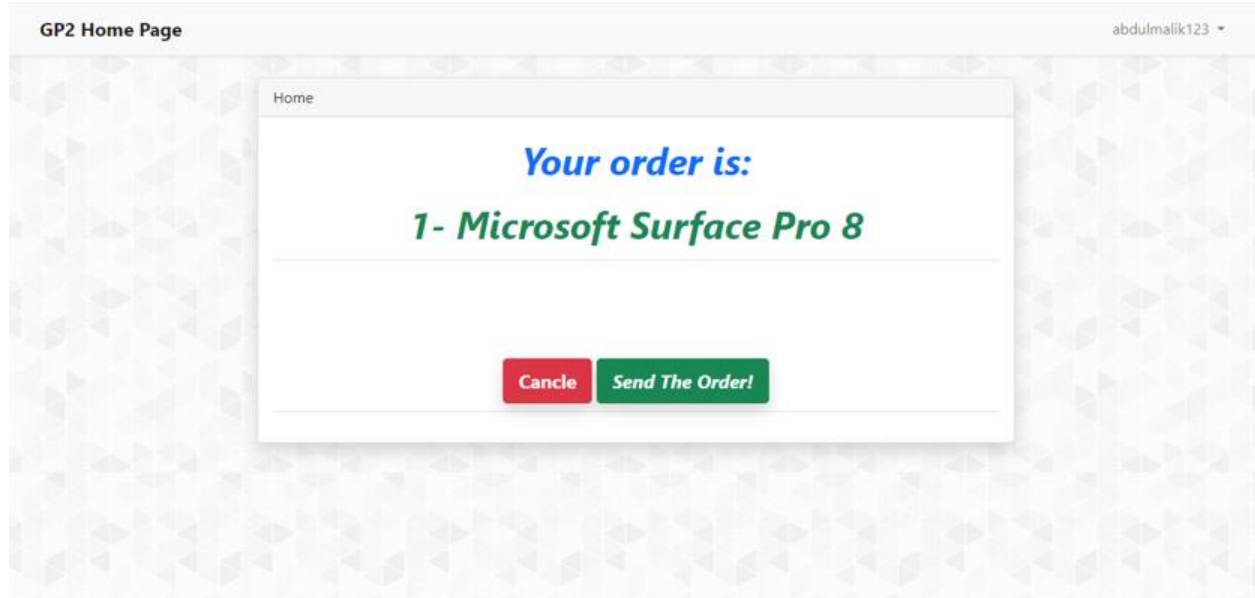


Figure 30 Order Summary

Development:

Various Controllers, Views, and models are developed using the PHP Laravel platform. As an example, the code for a few controllers, models, and views is shown below.

Controllers

RegisterController:

```
<?php
```

```
namespace App\Http\Controllers\Auth;
use App\Http\Controllers\Controller;
use App\Providers\RouteServiceProvider;
use App\Models\User;
use Spatie\Permission\Models\Role;
use Illuminate\Foundation\Auth\RegistersUsers;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;

class RegisterController extends Controller
{

    use RegistersUsers;

    /**
     * Where to redirect users after registration.
     *
     * @var string
     */
    protected $redirectTo = RouteServiceProvider::HOME;

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest');
    }

    /**
     * Get a validator for an incoming registration request.
     *
     * @param array $data
     * @return \Illuminate\Contracts\Validation\Validator
     */
    protected function validator(array $data)
    {
        return Validator::make($data, [
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
            'password' => ['required', 'string', 'min:8', 'confirmed'],
        ]);
    }

    /**
     * Create a new user instance after a valid registration.

```

```

*
* @param array $data
* @return \App\Models\User
*/
protected function create(array $data)
{
    $user = User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => Hash::make($data['password']),
    ]);

    $user->assignRole('Member');

    return $user;
}
}

```

Model

User.php:

```
<?php
```

```

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;
use Spatie\Permission\Traits\HasRoles;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable, HasRoles;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.

```

```

*
* @var array
*/
protected $casts = [
    'email_verified_at' => 'datetime',
];
}

```

Views -> layouts

app.blade.php:

```

<!doctype html>
<html lang="{{ app()->getLocale() }}">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- CSRF Token -->
    <meta name="csrf-token" content="{{ csrf_token() }}">

    <title>{{ config('app.name', 'Home Page') }}</title>

    @vite(['resources/sass/app.scss', 'resources/js/app.js'])
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/css/bootstrap.min.css" rel="stylesheet" >
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" ></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.min.js" integrity="sha384-
    Atwg2Pkwv9vp0ygtn1JAojH0nYbwNjLPhwyoVbhoPwBhjQPR5VtM2+xf0Uwh9KtT"
    crossorigin="anonymous"></script>

    <!-- Fonts -->
    <link rel="dns-prefetch" href="//fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet">
    <style type="text/css">
    body{
        background: url('/storage/bg1.jpg') !important;
        /*background: navy !important;*/
    } /* Adding !important forces the browser to overwrite the default style applied by Bootstrap */
</style>
</head>
<body>
    <div id="app">
        <nav class="navbar navbar-expand-md navbar-light bg-light bg-gradient shadow-sm">
            <div class="container">
                <a class="navbar-brand fw-bold " href="{{ url('/GP2') }}">
                    GP2 Home Page
                </a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
                target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-
                label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>

                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <!-- Left Side Of Navbar -->
                    <ul class="navbar-nav me-auto"></ul>

                    <!-- Right Side Of Navbar -->

```

```

<ul class="navbar-nav ms-auto">
  <!-- Authentication Links -->
  @guest
    @if (Route::has('login'))
      <li class="nav-item">
        <a class="nav-link" href="{{ route('login') }}">{{ __('Login') }}</a>
      </li>
    @endif
    @if (Route::has('register'))
      <li class="nav-item">
        <a class="nav-link" href="{{ route('register') }}">{{ __('Register') }}</a>
      </li>
    @endif
    <li class="nav-item">
      <a class="nav-link" href="{{ route('store.form') }}">{{ __('Stores Registration') }}</a>
    </li>
  @else
    @can('role-list')
      <li><a class="nav-link" href="{{ route('users.index') }}">Manage Users</a></li>
      <li><a class="nav-link" href="{{ route('roles.index') }}">Manage Roles</a></li>
      <li><a class="nav-link" href="{{ route('categories.index') }}">Manage Categories</a></li>
      <li><a class="nav-link" href="{{ route('stores.index') }}">Manage Stores</a></li>
      <li><a class="nav-link" href="{{ route('orders.index') }}">Manage Orders</a></li>
    @endcan
    <li class="nav-item dropdown">
      <a id="navbarDropdown" class="nav-link dropdown-toggle" href="" role="button" data-bs-
toggle="dropdown" aria-haspopup="true" aria-expanded="false" v-pre>
        {{ Auth::user()->name }}
      </a>

      <div class="dropdown-menu dropdown-menu-end" aria-labelledby="navbarDropdown">
        <a class="dropdown-item" href="{{ route('logout') }}"
onclick="event.preventDefault();
document.getElementById('logout-form').submit();">
          {{ __('Logout') }}
        </a>

        <form id="logout-form" action="{{ route('logout') }}" method="POST" class="d-none">
          @csrf
        </form>
      </div>
    </li>
  @endguest
</ul>
</div>
</div>
</nav>
<main class="py-4">
  <div class="container">
    @yield('content')
  </div>
</main>
</div>
</body>
</html>

```

Testing:

Testing the system is very important, whether before or after the launch of the product, and the main reason for that is to ensure that the system is safe from technical problems and defects.

There are fields and specializations that are only interested in this field because of its necessity and because it is one of the most important reasons for the success or failure of the system. Also, make sure that the system reaches the desire of users. There are a number of different types of testing, the most important of which are: Unit Testing: method of testing individual units or components of a software application. It is typically done by developers and is used to ensure that the individual units of the software are working as intended. Integration Testing: method of testing how different units or components of a software application interact with each other. It is used to identify and resolve any issues that may arise when different units of the software are combined. [17] White Box Testing: testing the product's underlying structure, architecture, and code to validate input-output flow and enhance design, usability, and security. Black Box Testing: involves testing against a system where the code and paths are invisible. There are Grey Box also, which combined both white and black box testing. [18]

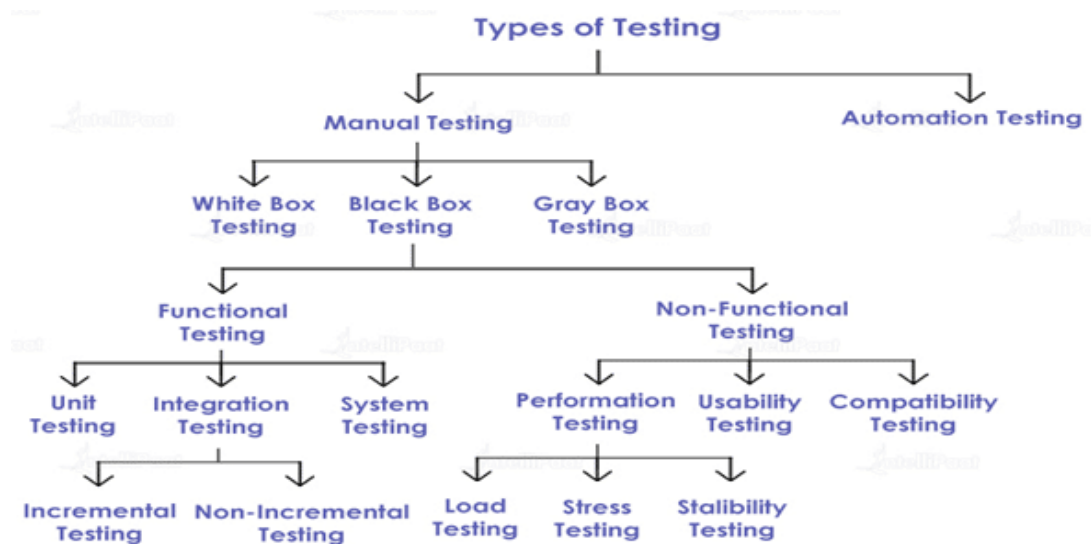


Figure 31 Types of Testing [19]

A few example of various testing performed on system are shown below:

GP2 Home Page Login Register Stores Registration

Login

Email Address !

These credentials do not match our records.

Password

Remember Me

[Forgot Your Password?](#)

Figure 32 Password do not match

Register

Name

Email Address

Password !

The password field must be at least 8 characters.

Confirm Password

Figure 33 Password must be at least 8 characters

Register

Name

Email Address

Password ! Please enter a part following '@'. 'abc@' is incomplete.

Confirm Password

Figure 34 Email Validation

Roles Management

Create New Role

Role updated successfully

No	Name	Action
1	Admin	Show Edit Delete
2	Member	Show Edit Delete

Figure 35 Role updated successfully by Admin

Stores Management

Create New Store

Store created successfully

No	Name	Email	Status	Action
1	32iz	azeez142111@gmail.com	Approved	Show Edit Delete
2	test	test@gmail.com	Not Approved	Show Edit Delete
3	test2	test2@gmail.com	Not Approved	Show Edit Delete
4	test3	test3@gmail.com	Not Approved	Show Edit Delete
5	test5	test5@gmail.com	Not Approved	Show Edit Delete
6	test6	test6@gmail.com	Not Approved	Show Edit Delete
7	ahsan	a.ahmed@mu.edu.sa	Not Approved	Show Edit Delete
8	store1	store123@gmail.com	Approved	Show Edit Delete
9	store2	store2@gmail.com	Not Approved	Show Edit Delete

Figure 36 Store Created Successfully

Edit Store

Whoops! There were some problems with your input.

- The papers field must be a file of type: jpg, jpeg, png, pdf.

Name:

Email:

Papers:

Choose File No file chosen

Status: Not Approved Approved

Submit

Back

Figure 37 The Papers field must be of type : jpg, jpeg, png, pdf only.

No	Name	Email	Roles	Action
1	LaravelTuts	admin@laraveltuts.com	Admin	Show Edit Delete
2	Abdulaziz Hamad	azeez142111@gmail.com	Admin	Show Edit Delete
3	Abdulmalik	abdulmalik@gmail.com	Member	Show Edit Delete
4	ahmed	a.ahmed@mu.edu.sa	Member	Show Edit Delete
5	abdulmalik123	test123@gmail.com	Member	Show Edit Delete

Figure 38 Deleted User Successfully

Store Registration

Whoops! There were some problems with your input.

- The papers field is required.

Name:

Email:

Papers:
 No file chosen

Figure 39 Papers Validation

Store Registration

We have received your request and it is under review. You will start receiving orders through your email when your request is approved!

Name:

Email:

Papers:
 No file chosen

Figure 40 Received Store Registration request

Conclusion:

The website will provide an effective platform to connect stores and customers using the internet, the user can find and order any device they want. The website will make the communication process between customers and stores easier, it will save time for users, and replace the need for stores to build an online store. The website can be extended in the future to cover more stores in different cities, and include different types of stores besides the telecommunication stores. Of course, we have a future vision to develop the level of services, the user interface and the application of the highest levels of protection for the information of stores and users to raise confidence between us. Also, provide better services to customers through user feedback.

References:

1. **Pew Research Center:** <https://www.pewresearch.org/internet/2004/08/11/the-internet-and-daily-life/> [accessed online on: 19 December, 2022]
2. **Asianet Fiber:** <https://asianetbroadband.in/importance-of-internet-technology-for-easy-life/> [accessed online on: 19 December, 2022]
3. **TechTarget:** <https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app> [accessed online on: 19 December, 2022]
4. **Wingravity:** <https://www.wingravity.com/blog/top-10-things-to-look-for-in-great-web-apps-2/> [accessed online on: 19 December, 2022]
5. **LinkedIn:** <https://www.linkedin.com/pulse/5-considerations-successful-web-application-design-jesse> [accessed online on: 19 December, 2022]
6. **Ideasonpurpose:** <https://www.ideasonpurpose.com/on/7-qualities-great-website/> [accessed online on 19 December 2022]
7. **LinkedIn:** https://www.linkedin.com/pulse/characteristics-web-application-nripesh-sonavane?trk=public_profile_article_view [accessed online on: 19 December, 2022]
8. **w3schools:** <https://www.w3schools.in/mvc-architecture> [accessed online on 19 December 2022]
9. **guru99:** <https://www.guru99.com/mvc-tutorial.html> [accessed online on 19 December 2022]
10. **freecodecamp:** <https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/> [accessed online on: 19 December, 2022]
11. **codecademy:** <https://www.codecademy.com/article/mvc> [accessed online on 19 December 2022]
12. **Laravel:** <https://laravel.com/docs/9.x/installation> [accessed online on 19 December 2022]
13. **Wikipedia:** <https://en.wikipedia.org/wiki/Laravel> [accessed online on 19 December 2022]
14. **Wastetco:** <https://www.wastetco.com/> [accessed online on 29 December 2022]
15. **Speero:** <https://speero.net/> [accessed online on 29 December 2022]
16. **Haraj:** <https://haraj.com.sa/> [accessed online on 31 December 2022]
17. **Geeksforgeeks:** <https://www.geeksforgeeks.org/types-software-testing/> [accessed online on 11 may 2023]
18. **perfecto.io:** <https://www.perfecto.io/resources/types-of-testing> [accessed online on 11 may 2023]
19. **intellipaat:** <https://intellipaat.com/blog/types-of-software-testing/?US> [accessed online on 11 may 2023]