



UNIVERSITÀ
DEGLI STUDI
DELL' AQUILA

KNOW WHERE IT IS !

SERVICE - ORIENTED SOFTWARE ENGINEERING

DT0203
a.y.2019/2020

MARCO AUTILI
University of L'Aquila, Italy

By,

Jayasurya Arasur Subramanian (267412)

Moises Romero Romo (266567)

Gagana Kaveriappa (267359)

Hilal Thaha (267365)

Abstract

‘Know where it is!’ is a website which allows users to search nearby shops where the product that they need is available, when they are visiting a new place. When we are visiting or relocating to any new place, it is very hard and time consuming to figure out where to get some necessary products, so our website will provide information and locations of all the shops which have the product that we are looking for. It also help the buyer to get a best buy.

Introduction

Many people consider the search for their needed product as an extremely time-consuming part of their responsibilities when they are visiting a new place. The enormity of time taken is partly caused by the increasing number of many small shops in a certain place, which grows exponentially every year according to the popularity of the locations. The currently used method for searching a product is searching any shops in the surrounding through any search engines or search physically and go there and investigate about the product which we need. Due to non-availability of product details in the webpage it is more tough to search through the internet. To overcome this we present a hybrid recommender system, which will provide both the information of the product and location of all shops which have the product that we are searching for as a result.

Objective of the project:

1. Help people to find the location of the product.
2. Giving the user a best buy (comparing the price and location of the product).
3. Giving some promotion to the shop.

Existing System:

There is no such system in existence. People will search for the product manually or with the help of google maps people will find the location of nearby shops and go there manually to find the product is there are not. The manual searching is more time consuming and error prone. It is very difficult for a person to search for a product in the unknown location. There are chances of getting lost and wasting too much of time and energy for searching the product.

Proposed System

The proposed system is designed to eliminate the drawbacks of manual search. It is designed by keeping in mind all the drawbacks of the present system in order to provide a permanent solution to the problems. The primary aim of this system is to help the user to find the products location and help the shop owners to promote their shops. This system is developed using following constraints

1. Developing both REST services and SOAP services interacting each other.
2. Developing an web application.
3. Adopting Apache CXF.
4. Mavenizing the project.
5. Use of WSDL 1.1 for Message Exchanging Patterns
6. Developing at least an asynchronous Web Service (polling and/or callback approaches)
7. Use of Thread Pool Executor for serving requests. (for a subset of services)
8. Use Open API and Swagger for all REST services.

The advantages of this system are

1. User friendliness is the keyword for all the new software in the market. The proposed system incorporates this concept into itself to guide the user.
2. The proposed system is expected to reduce the time for searching the products to the user.
3. Data redundancy is avoided
4. Data integrity is maintained
5. Security is ensured

Packages Selected

Front End	: Java (Maven)
Web Tools	: HTML
Web Server	: Apache Tomcat 7.0
Validation	: Java Script
Backend	: MySQL

Database Description

Table : Product

#	Field	Shema	Type	Character Set
1	product_id	t1	INT UNSIGNED	binary
2	product_name	t1	VARCHAR	utf8mb4
3	product_price	t1	DOUBLE	binary
4	product_count	t1	INT	binary
5	category_id	t1	INT	binary
6	store_id	t1	INT UNSIGNED	binary
7	product_image	t1	VARCHAR	utf8mb4

Table : User Groups

#	Field	Shema	Type	Character set
1	Id	t1	INT UNSIGNED	binary
2	user_id	t1	INT UNSIGNED	binary
3	group_id	t1	MEDIUMINT UNSIGNED	binary

Table : Groups

#	Field	Schema	Type	Character set
1	id	t1	MEDIUMINT UNSIGNED	binary
2	name	t1	VARCHAR	utf8mb4
3	description	t1	VARCHAR	utf8mb4

Table : Category

#	Field	Schema	Type	Character set
1	id	t1	MEDIUMINT UNSIGNED	binary
2	name	t1	VARCHAR	utf8mb4
3	description	t1	VARCHAR	utf8mb4

Table : Users

#	Field	Shema	Type	Character set
1	id	t1	INT UNSIGNED	binary
2	ip_address	t1	VARCHAR	utf8mb4
3	username	t1	VARCHAR	utf8mb4
4	password	t1	VARCHAR	utf8mb4
5	salt	t1	VARCHAR	utf8mb4
6	email	t1	VARCHAR	utf8mb4
7	activation_code	t1	VARCHAR	utf8mb4
8	forgotten_password_code	t1	VARCHAR	utf8mb4
9	forgotten_password_time	t1	INT UNSIGNED	binary
10	remember_code	t1	VARCHAR	utf8mb4
11	created_on	t1	INT UNSIGNED	binary
12	last_login	t1	INT UNSIGNED	binary
13	active	t1	TINYINT UNSIGNED	binary
14	approved	t1	TINYINT	binary
15	first_name	t1	VARCHAR	utf8mb4
16	last_name	t1	VARCHAR	utf8mb4
17	company	t1	VARCHAR	utf8mb4
18	phone	t1	VARCHAR	utf8mb4
19	country	t1	VARCHAR	utf8mb4
20	state	t1	VARCHAR	utf8mb4
21	city	t1	VARCHAR	utf8mb4
22	address_line_1	t1	VARCHAR	utf8mb4
23	address_line_2	t1	VARCHAR	utf8mb4
24	latitude	t1	VARCHAR	utf8mb4
25	longitude	t1	VARCHAR	utf8mb4
26	zip	t1	VARCHAR	utf8mb4
27	seller_username	t1	VARCHAR	utf8mb4
28	profile_picture_file	t1	VARCHAR	utf8mb4
29	total_sales	t1	FLOAT	binary

Architectural Design

Dynamic web applications are presented as a three-tier architecture, in which the Java Maven play a key role in the business logic layer of the architecture. Designing the application in layers, or tiers, is useful for many different reasons. In a multiple tier design,

each tier can be run on a separate machine, or machines, allowing for improved processing performance. Depending on the design, multiprocessor machines, or many different independent computers can be used to improve performance. Efficient layering can give structure to our application, promote scalability, and ease long-term maintenance requirements for our code.

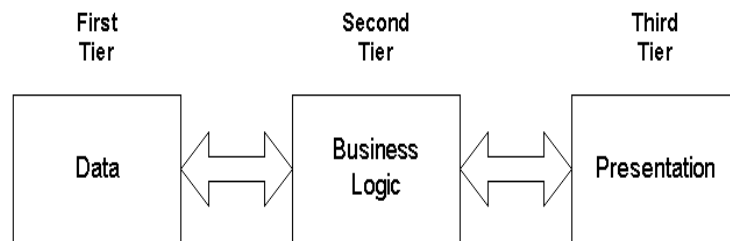


Fig 1: Three Tier Architecture

First Tier: *Persistent Data Storage* – The underlying technology to implement this layer could be a variety of things including cookies, server-side files or databases.

Second Tier: *Business Logic Layer* – This is where one codes any rules regarding the data stored and generally is the bulk of the application.

Third Tier: *Presentation* – Enables the user to see the results of the business logic applied to the data stored.

Working Principle

We have so many e-commerce websites in use in which each and every website sells the product. If we order the product it will take a day or two to deliver the product. It will be more time consuming. Consider the situation where you go for a trip to a place which you are not familiar with, during that time you need some product urgently, but you don't know where to buy. During that time our website will come in handy. Just you type the product which you need in our website it will take care of the product availability and the location of the product near you. To get those we are implementing the following services.

Product_fetch () service:

If a user enters the product name and his current location in the website, our website will search for the product availability in the nearby location. It is implemented using SOAP

web service. It takes product name and user_id as the input and fetches all the product details from the product table with the same name. We have used the Google Distance Matrix API with the locations of user and shop to get 'distance' of the shop from current location and 'duration' it would take to reach there as response. We intend to display products which are cheaper and are available in shops which are closer, on the top of our search results. So, we have ordered the products according to 'ProductOrder' which is calculated by, $\text{ProductOrder} = \text{product_price} * \text{distance}$. The response of this service is an ordered list of product details in each shop along with the distance and duration to help user reach the shop. Since this would be a very busy service with a lot of requests, we have also used the concept of threading to obtain smooth and fast execution

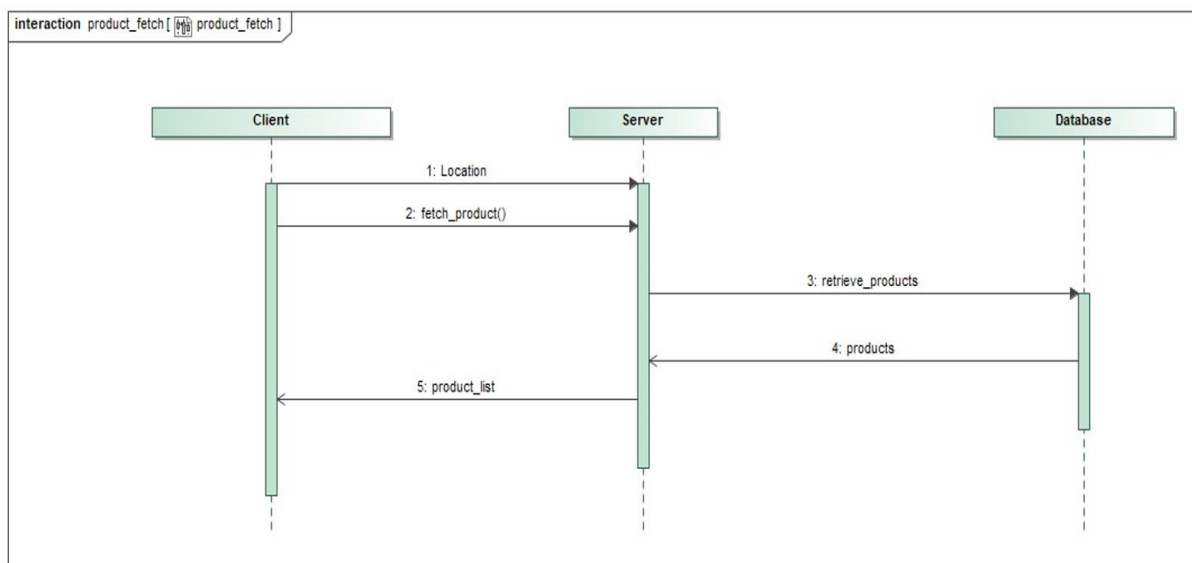


Fig:2 Product fetch service

Add_product () service:

If a user (shop owner) want to add the product in their store add_product () service is called. This is a SOAP service developed in Java that provides the necessary functionality to add all the stuff about the products in our application into the database and is called by the interface.

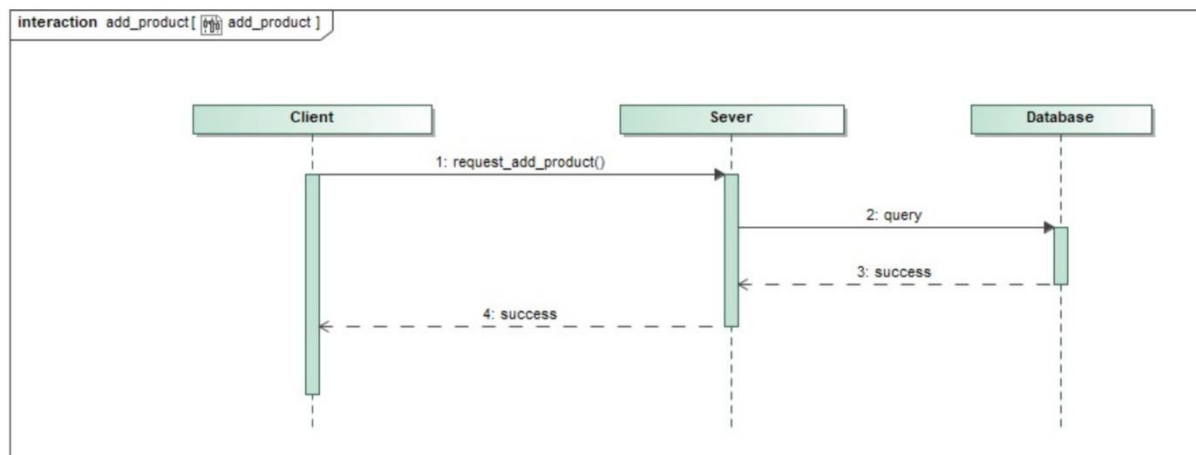


Fig:3 add_product service

Edit_product () service:

If the user (shop owner) needs to make some changes in the product details then this service is called. It is also a SOAP service which provides the necessary functionality to edit all the stuff about the products in in the database and is called by the interface.

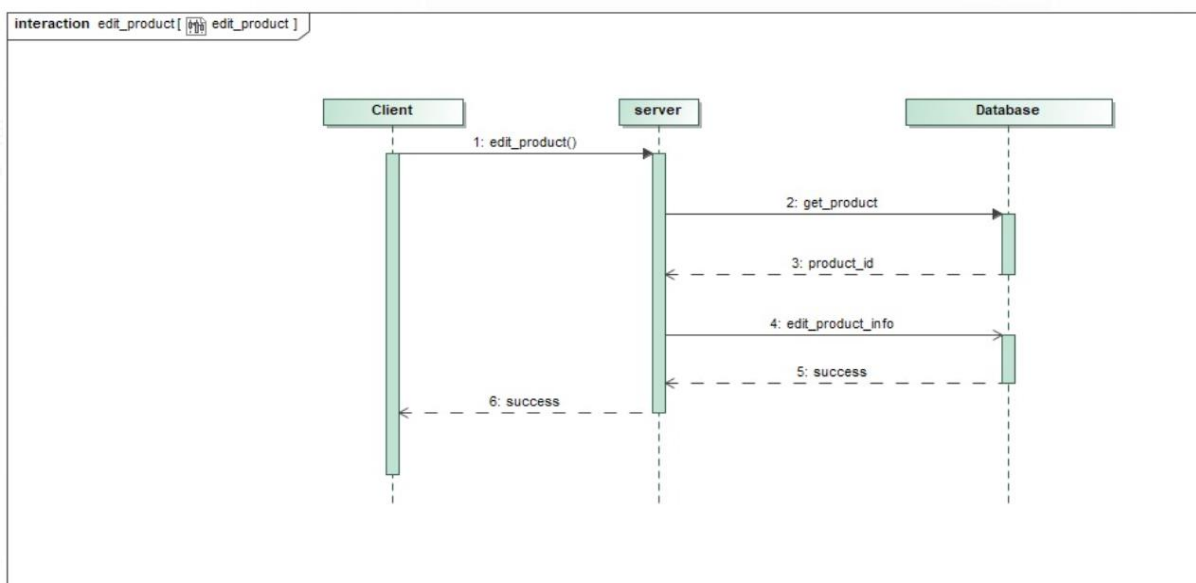


Fig:4 Edit_Product service

USER_INFO () service:

This is an REST service used to log on into the system. It will collect all the needed information and store those information at the database. While storing the data every data is encoded which ensures the privacy of the users. It use SWAGER UI which is an open-source software framework backed by a large ecosystem of tools that helps developers design, build,

document, and consume RESTful web services. The Swagger toolset includes support for automated documentation, code generation, and test-case generation.

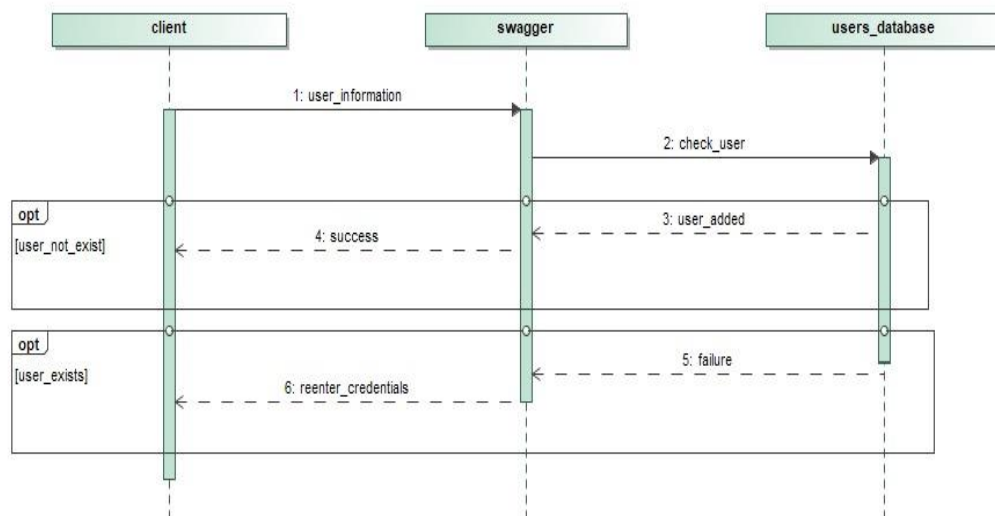


Fig 5: User_info() service

The data are collected from the client through swagger and updated in the database. Then our system will check for the existence of the user, if the user is not existing our system will add the new user. And if the user is already existing it will not create a new user and it will leave an error message.

LOGIN() Service:

This SOAP service takes care of the user login into the system. Here the user give their credentials to the server, it checks the user entered credentials are correct. If the credentials match to the user's data it will login to the system else it will give an error message and ask the user to reenter the credentials correctly.

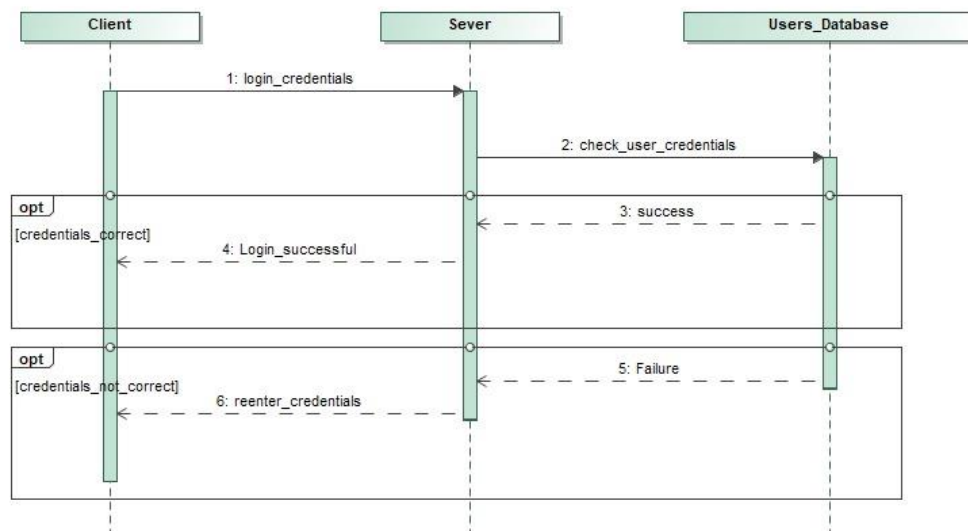


Fig 6: Login() service

User Interface

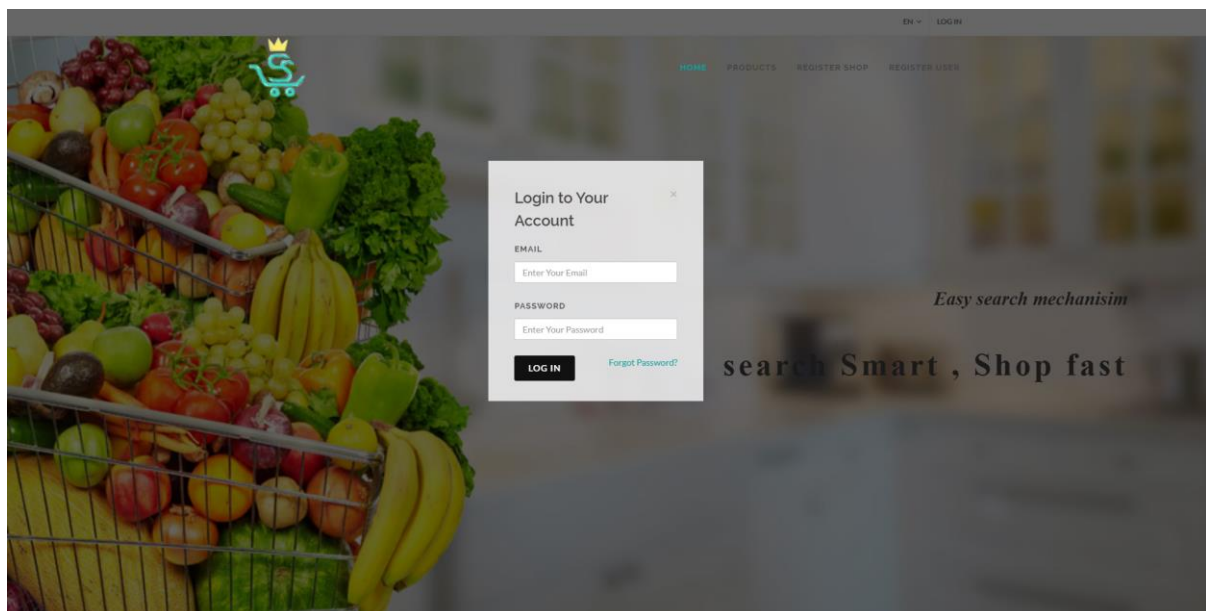


Fig 7: UI Login Page

Figure 7 represent the login page of our system. In which Home bar provides the advertisement of the shops, Product bar provides list of products available in the nearby location. And the Register shop is used for registering the shop with our system in which shop owners will upload their details and their shop details with our system. Users can register them

self with our system by using register system. In which they will provide their information. The user can reset their password using their email.

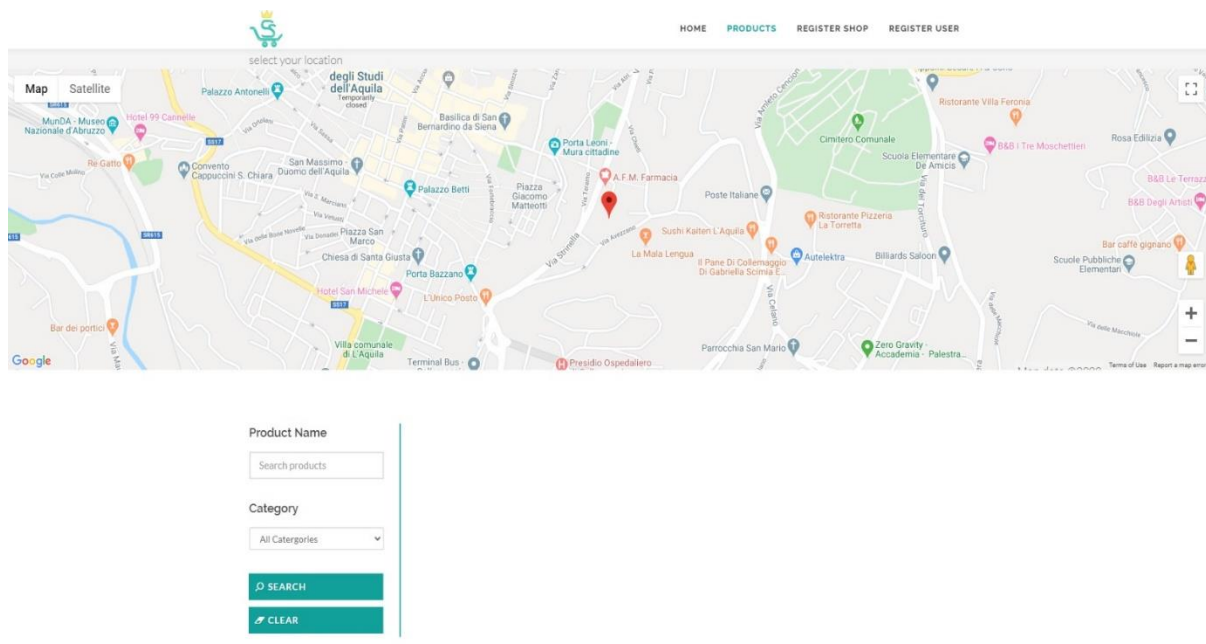


Fig 8: UI Product search

Figure 8 represents the page where the user will type the product which he wants with that product category. Our system will automatically get his location and search for the product in nearby to his location. If the system finds the product it will display the list of shops in the order of best buy. Best buy in the means our system will automatically rank the shop with the help of price and distance of the shop from the user's location. It will then guide the user to the selected shop with the help of google maps.

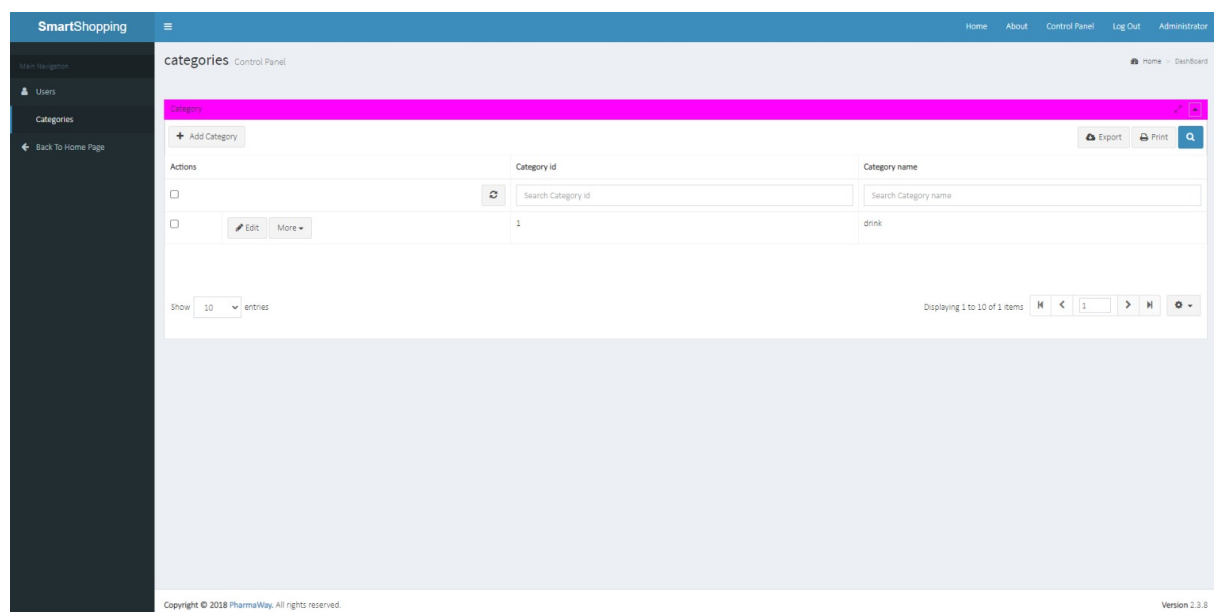
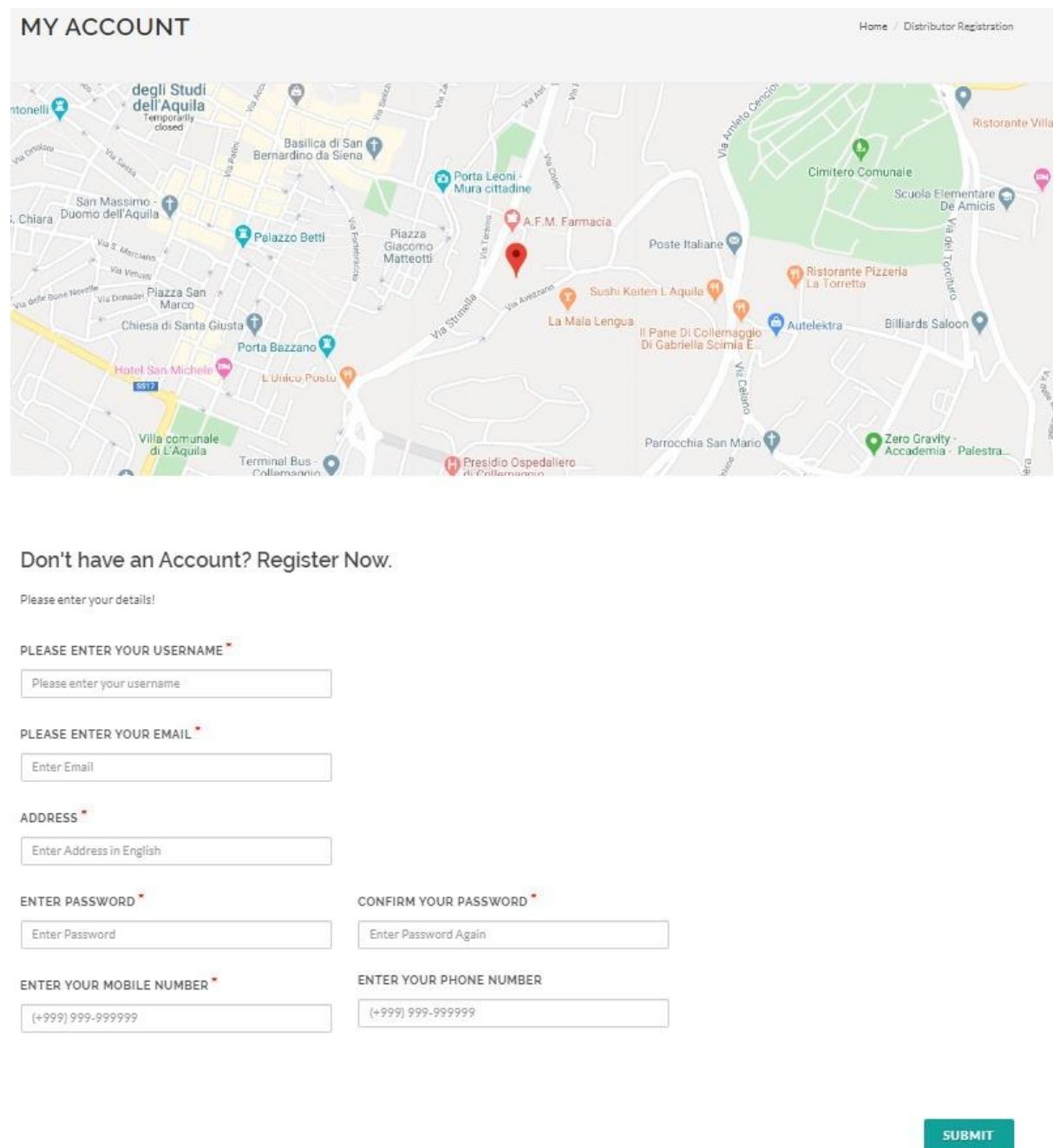


Fig 9: UI Categories registration

Figure 9 represents the admin page where the admin can Add, Edit and Remove the Users, Shop owners and Categories. With the help of this administrators can keep an eye on the system about what's going on and help to remove any fake details present.



MY ACCOUNT Home / Distributor Registration

degli Studi dell'Aquila Temporarily closed

Basilica di San Bernardino da Siena

Porta Leoni - Mura cittadine

A.F.M. Farmacia

Poste Italiane

Cimitero Comunale

Scuola Elementare De Amicis

Ristorante Villa

San Massimo - Duomo dell'Aquila

Piazza Giacomo Matteotti

Piazza San Marco

Chiesa di Santa Giusta

Hotel San Michele

Villa comunale di L'Aquila

Terminal Bus - Collarmannio

Presidio Ospedaliero di Collarmannio

Sushi Kanten L'Aquila

La Mala Lengua

Il Pane Di Collarmaggio Di Gabriella Scimia E...

Ristorante Pizzeria La Torretta

Autelektra

Billiards Saloon

Perrocchia San Mario

Zero Gravity - Accademia - Palestra

Don't have an Account? Register Now.

Please enter your details!

PLEASE ENTER YOUR USERNAME *

Please enter your username

PLEASE ENTER YOUR EMAIL *

Enter Email

ADDRESS *

Enter Address in English

ENTER PASSWORD *

Enter Password

CONFIRM YOUR PASSWORD *

Enter Password Again

ENTER YOUR MOBILE NUMBER *

(+999) 999-999999

ENTER YOUR PHONE NUMBER

(+999) 999-999999

SUBMIT

Fig 10: UI Registration Page

Figure 10 represents the registration page for our shop owners, where the shop owner can register their details and choose where their shop is located using the map, which will detect the longitude and latitude and register them into the database when registering. Once the shop owner registers their details here then they will become a shop in our system. Then they can log in into our system whenever they want with the login credentials they provide.

Conclusion

The system is developed for identifying the needed product with an ease in an unknown location. This system is of vital use of sales and give the location of the product. “Know Where It Is!” helps to identify the needed product with an ease in an unknown location. It also helps the user to get a best buy. So, the user can have a dual benefit. Not only the buyer the seller also gets a huge benefit by promoting his shop.