



SCHOOL OF SCIENCE AND ENGINEERING

Sportswear E-commerce Mobile Application

SCHOOL OF SCIENCE AND ENGINEERING

CAPSTONE FINAL REPORT

EGR 4402

Fall 2017

Student: Mehdi Bakkali Maassom

Supervised by: Dr. Nasser Assem

SPORTSWEAR E-COMMERCE MOBILE APPLICATION

Student Statement:

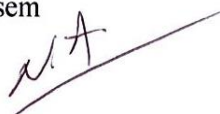
I, Mehdi Bakkali Maassom, assert that I have applied ethics to the design process and in the selection of the final proposed design. I also affirm that I have held the safety of the public to be paramount and have addressed this in the presented design wherever may be applicable.

_____ **Mehdi Bakkali Maassom** _____

Capstone student: Mehdi Bakkali Maassom

Approved by the supervisor

Dr. Nasser Assem



ACKNOWLEDGEMENTS

First, I would like to express my thankfulness to Dr. Assem Nasser, my supervisor, who assisted me throughout the development of the project. He provided me with the necessary suggestions that guided me and saved me a lot of time. I would also like to thank him for -- accepting me as a supervisee and giving me the opportunity to work on a project that highly motivates me.

I would also like to thank Al Akhawayn University and the School of Science and Engineering for giving me the opportunity to work my capstone project, which I intend to turn into a startup, in a healthy work environment. I would also like to thank SSE for providing me with the necessary resources that allowed me to develop a successful project.

I would like to thank my friends Ahmad Khalil, Hamza, Hayat, Mohamed Yassin, Taha, Zakaria, Jawad, Issam, Ahmed, and Zidane for the moral support and helpfulness that they provided me with throughout my journey at AUI.

Last but not least, I would like to express my deepest gratitude to my parents who invested in me both morally and financially, for their continuous support and without whom, I would probably not be here today.

TABLE OF CONTENTS

1	ABSTRACT	6
2	INTRODUCTION	7
3	STEEPLE Analysis	8
4	REQUIREMENTS SPECIFICATIONS	10
5	FEASIBILITY STUDY	17
6	METHODOLOGY	18
7	SOFTWARE ARCHITECTURE	19
8	DESIGN	20
	8.1 Use Case Diagram:	20
	8.2 Interaction Overview Diagram	21
	8.3 Data Model	22
9	IMPLEMENTATION	25
10	TECHNOLOGY ENABLERS	32
11	TESTING	34
12	Conclusion & Future Perspectives:	35
	Appendix A: Web Application user manual	37

Figures

Figure 1 Login Frame of Desktop App	12
Figure 2 Main Menu	12
Figure 3 Products Display	13
Figure 4 List of Orders	14
Figure 5 List of Products in an order	14
Figure 6 List of Registered Accounts	15
Figure 7 User Details and His Orders	15
Figure 8 Rapid Application Development Model	18
Figure 9 Rapid Application Development	18
Figure 10 System Architecture	19
Figure 11 Use Case Diagram	20
Figure 12 Interaction Overview Diagram	21
Figure 13 Entity Relationship Diagram Made using PHPMyAdmin	22
Figure 14 Mobile application Login	26
Figure 15 Mobile application Main Menu	27
Figure 16 Mobile application Navigation Drawer	27
Figure 17 Mobile application Product by Category	29
Figure 18 View Cart	30

1 ABSTRACT

Over the past years, online commerce has become very popular. In Morocco, the culture of e-commerce is, slowly but surely, kicking in. More than that, there is a smooth shift to using mobile devices for shopping. Therefore, “Ramo Sports”, which is a sportswear company located in Tangier, decided to expand their business open up to online market through an e-commerce system. This will get new clients from all over Morocco to use their system.

This project, initially, started in the Database Systems course with Dr. N. Assem, where a desktop application was developed for the management of the database. A web application was also developed as an interface for the e-commerce.

This capstone project started with the design of a more sophisticated backend, database with a web application synchronized with the mobile application’s database to be implemented. Afterwards, the online shopping platform was implemented as a mobile application.

The objective of this project is to have a complete, reliable, and unique e-commerce system that is expected to be an added value for the company.

2 INTRODUCTION

The Capstone Design course is required by the School of Science and Engineering as partial fulfillment of the bachelor degree in Al Akhawayn University in Ifrane. Throughout the development of the capstone project, the student about to graduate demonstrates the knowledge acquired during his four years as an undergraduate student. The development of this project required the use of most of the learned computer science skills: Database Design, Web Development, Desktop Application Development, Software Engineering, and Mobile Applications Development.

In 2015, a survey was conducted by the National Telecommunications Regulatory Agency in Morocco (ANRT Maroc) about the usage of mobile phones in Morocco. The survey's results show that 54.1 of individuals are equipped with a smart phone, which is 14.7 million units by 2015 and it is growing fast. The number of smart phone owners is growing very fast.

Ramo Sports is a store that offers various sports clothing and accessories for Men, Women and Children and meets the needs of the whole family. As a specialist in the distribution of sporting goods in Morocco, RAMO SPORTS offers its customers a wide variety of sports products.

Therefore, the development of a mobile application for online shopping is a very promising project since the use of smart phones is still growing and the market is not yet saturated. That is because the online shopping culture is not very old in Morocco.

The system that I am implementing in this capstone project is a complete e-commerce system, consisted in a database, web and mobile application (for customers), and a desktop application (for the administrator). However, the main focus of my capstone is on the Mobile Application.

3 STEEPLE Analysis

Socio-cultural:

Mobile phones have massively changed our society. They changed the way people live. In fact, almost everybody has a smart phone and use it on a daily basis for different tasks. Now, with phones full of applications, the first thing most us do in the morning is checking our smartphone. Mobile applications have changed the way we use our phones. We can say that our society is rapidly changing towards a very widespread use of mobile technologies. More than that, having a smart phone and using it for shopping is becoming a trend. Therefore, our sportswear shop needs to keep up with the societal changes in order to match our consumer needs and preferences.

Technological:

The market is rapidly changing, many of these changes are because of technological improvements. These advancements can create new markets and new opportunities. An online shop, as a mobile application, associated with our physical store is going to be the first of its kind in the sportswear market. This will allow the company to keep up with the technological trend and run the store more effectively. It will also improve the communication between the company and its consumers. This will improve the user experience as he will be informed about all the new products and have them delivered to him.

Economical:

The mobile application will be free to download for all customers. It is developed using open-source/free tools and programming languages, the profit will be mainly from the purchases made. More than that, the system developed is expected to guarantee economic growth for the company thanks to the expansion to the online market. It is also expected to gain user confidence. Also, since the e-commerce will be launched in parallel with the actual physical shop, this mobile application online shop will not affect the workflow of the overall company. This could also result in improving sales of sportswear products; which might reduce prices and increase competition.

Environmental:

Developing this online store will somehow be in favor of the environment. It will save the energy needed for the customers to physically visit the shop (fuel). This will preserve energy and prevent the pollution of the local environment. Also, creating a digitized version of the company will decrease the use of papers, hence preserve the environment.

Political:

In the Moroccan context, there are no direct consequences of launching a mobile application online store. In fact, the government of Morocco seem to have a positive position when it comes to new technologies.

Legal:

The mobile application that is going to be developed is completely legal since it doesn't infract any law in the Moroccan constitution. It won't be harmful in any way and it won't publish the users' private/personal information. Also, the company will pay the taxes for the sales made through the mobile application.

Ethical:

The mobile application will follow the code of ethics and won't harm in any way the customers, the company, the employees and the society as a whole. The consumers' rights will be respected to the maximum extent. The marketing techniques will be telling the truth about the products offered. The employees will be informed about the application's mission and goals. Also, the information of users will be encrypted and stored in secure database.

4 REQUIREMENTS SPECIFICATIONS

4.1 Functional Requirements

Introduction:

The project is composed of a Web Application (Already developed in previous work on the project), a desktop application (already developed but will be optimized and updated), these last two are linked to backend database (MySQL), and a mobile application that is going to be implemented together with a backend database (SQLite) and synchronized with the MySQL one.

This functional requirements section will be mainly about the mobile application but will also touch the desktop application that will take care of the management of orders, accounts and products. It is dedicated for the administrator of the system.

MOBILE APPLICATION:

Register:

The user of the application must be able to create an account. While creating his account, he must provide personal info: Name, address, email address, phone number... He will have to confirm his email address.

Login:

The mobile application must have login activity with a field for a username and a password. The user will be able to login if he confirmed his email address. After logging in, the user will be able to:

- view/modify his personal info.
- view latest products.
- add products to a cart.
- remove product from a cart.
- view the cart.
- confirm an order.

Home:

The mobile application must have a home screen visible for all the user of the application (whether logged in or not). The home screen should contain the latest products added to the database.

Search:

The mobile application should have a screen for product search based on some criteria:

- Price
- Name
- Brand

User Account:

The user must be able to modify his personal information and terminate his account if he has no pending orders.

Products:

The user of the application must be able to view a product list available for sale. He must be able to view their details and add them to his cart if he is logged in.

Orders:

After adding products to his cart, the user must be able to confirm his order and provide a valid address and a valid mobile phone for the products to be shipped.

Push Notifications:

The mobile application could have a push notifications functionality. It will be used to send promotions to users. It could also be used to inform users about new products.

About:

The mobile application should have a screen dedicated for general information about the company.

Desktop Application:

The desktop application (already implemented) is a database management system dedicated for the administrator to:

- Manage the accounts of users registered through the mobile application or the web application.
- Manage the products that will be displayed in the web and mobile application.
- Manage the orders made through the mobile or web application.

It must have the following functional requirements:

Login:

The login frame that will allow the administrator of the system to login using a predefined login and password.

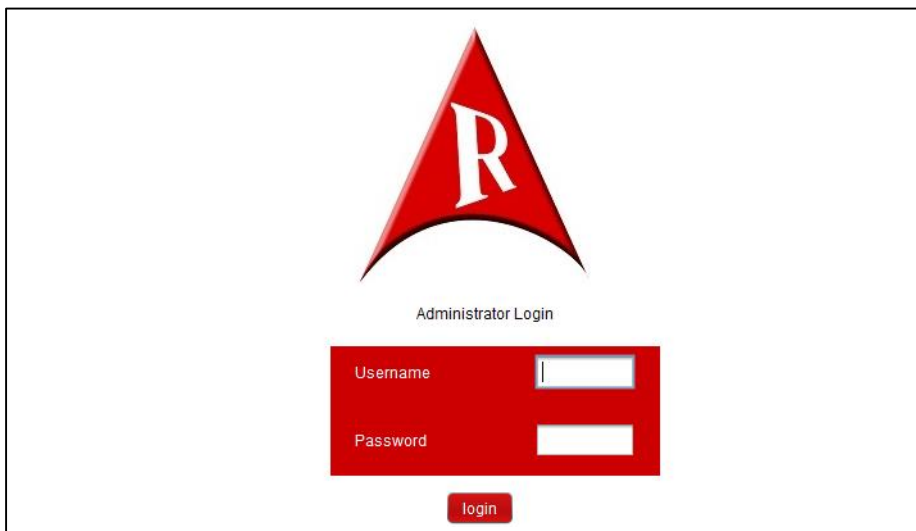


Figure 1 Login Frame of Desktop App

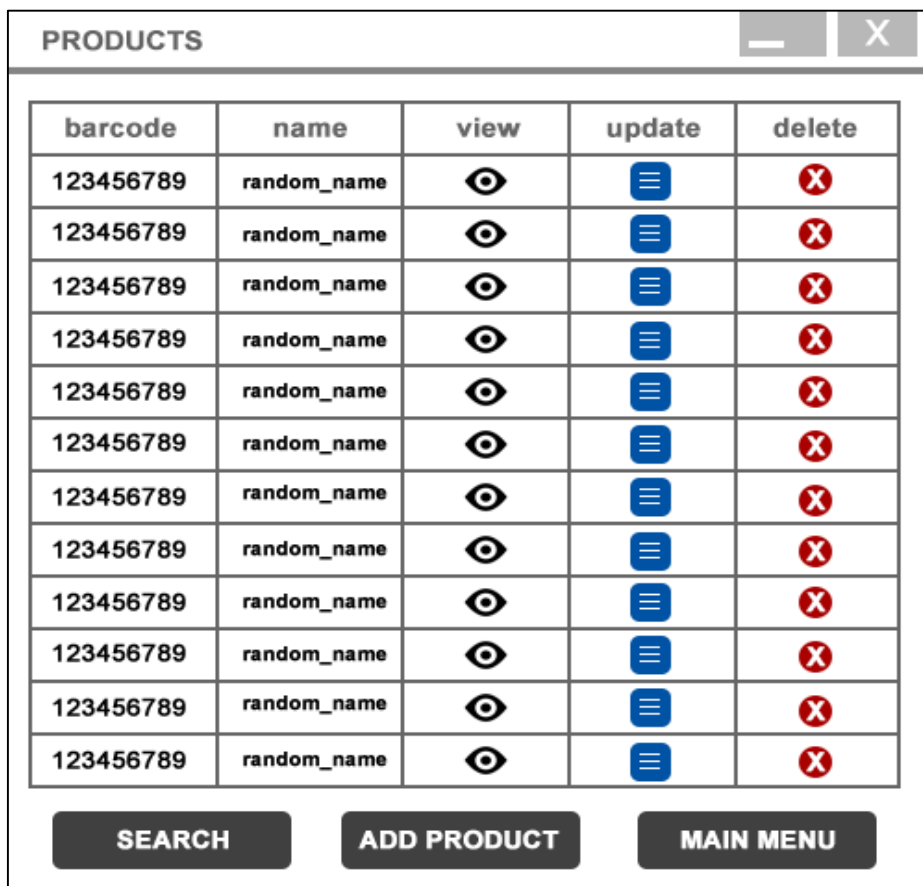
Main Menu: The application has a main menu with three options:



Figure 2 Main Menu

Product Management System:

The product management system allows the admin to view a complete list of all the products offered. It will also allow the admin to search, add, update and view products. The search is done based on many criteria.



barcode	name	view	update	delete
123456789	random_name			
123456789	random_name			
123456789	random_name			
123456789	random_name			
123456789	random_name			
123456789	random_name			
123456789	random_name			
123456789	random_name			
123456789	random_name			
123456789	random_name			
123456789	random_name			
123456789	random_name			

SEARCH ADD PRODUCT MAIN MENU

Figure 3 Products Display

E-commerce management system:

The e-commerce management system allows the admin to view a clear list of orders made by the users of the web and mobile application.

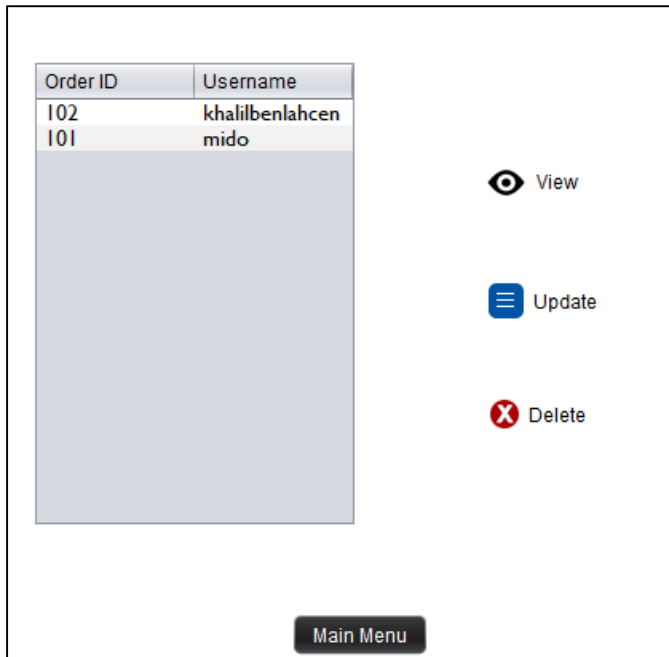


Figure 4 List of Orders

The admin will be able to view, update or delete orders.

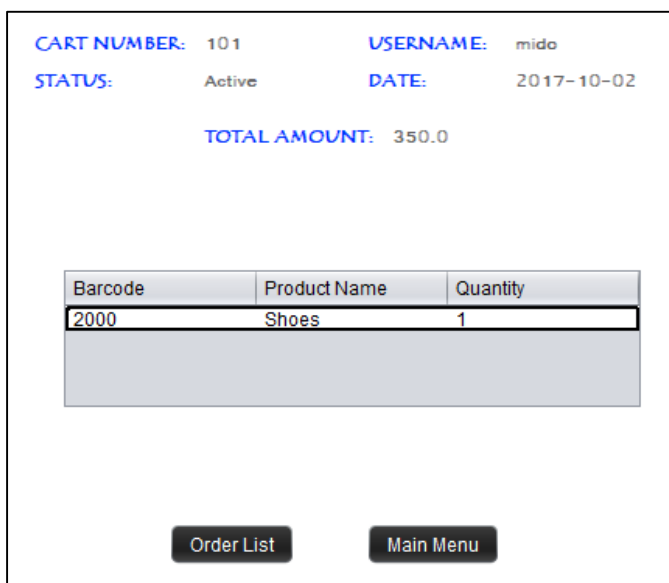


Figure 5 List of Products in an order

Accounts management system:

The accounts management system allow the admin to view or delete accounts created by the users of the web or mobile application.

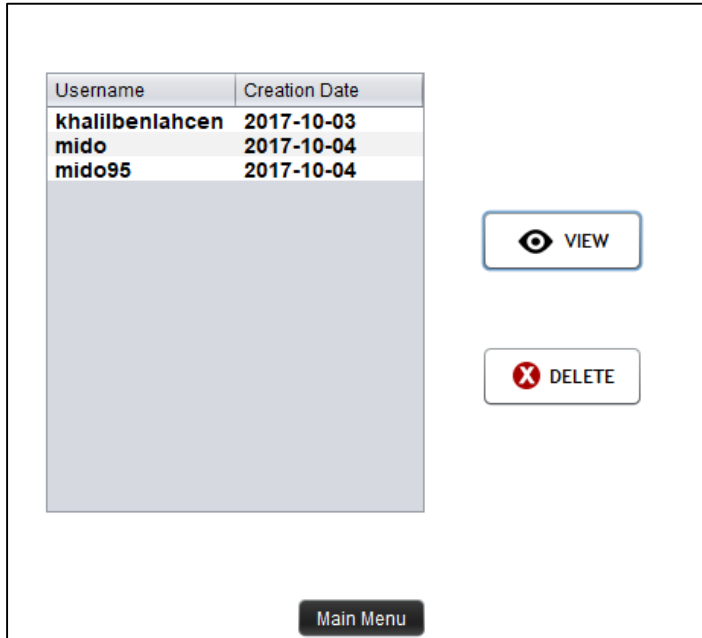


Figure 6 List of Registered Accounts

The account details shows the list of orders made by the user and their details.



Figure 7 User Details and His Orders

Some of these functionalities of the desktop applications are already implemented in the previous work on the project. However, some changes/updates are to be made in order to adapt it to mobile application that is going to be developed

4.2 Non-Functional Requirements:

Product requirements

Availability Requirements:

The system must be always available for use. Except for special cases (in case of backup), it will announced beforehand through push notifications.

Usability requirements:

The final mobile application will be entirely in French, because it is targeting the Moroccan market. It will be user-friendly and very easy to use.

Performance requirements:

The mobile application will be developed using Android Studio which will, thanks to the good algorithms that will be used, guarantee a high execution speed and a minimized response time.

Scalability requirements:

The expected number of users of the mobile application is very big. It is expected that the number of users grows on within the next few years. Therefore, the application must be highly scalable. We will achieve that thank to some MySQL solutions such as (MySQL cluster or MySQL replication)

Maintainability requirements:

The application should be maintainable in order to allow upgrades (through the Google Play Store) in the future.

Extensibility requirements:

The mobile application should be extensible. It should allow updating it and adding new features in the future.

Security requirements:

The application must be very secure because it deals with the private information of the users. This should be performed using the right encryption of data only accessed by the administrator. More than that, the system must follow these main security rules:

- Confidentiality: Only the admin has access to user personal info and orders.
- Integrity: Only the users can modify their personal info.
- Authenticity: No one can access, modify or delete other accounts' information.

5 FEASIBILITY STUDY

The feasibility study phase is a very important phase, because it is the one that helps the software engineer to see the negative sides of his project and it helps him to refine the software process. In fact, the feasibility study can make any project a successful one if it is done in a correct way. This deliverable is the feasibility study for the project I will be working on my capstone: “Sportswear Store E-commerce”

Product feasibility:

The system that will be developed will help the client achieve their objectives. Which are expanding their business and open up to the online market. This will be achieved thanks to the mobile application that will make the access to the company’s products, order them and communicate with the company extremely easy. Talking about communication, the mobile application with the “push” feature will allow the administrator to send notifications to the users’ phone about new products, promotions, or just advertising the store’s products.

Technical and Operational Feasibility:

For this project, I will be using Java & XML for the mobile application using Android Studio as it is the official IDE for android applications’ developing.

For the backend, I will be using SQLite as a database engine; as it is known for its high-reliability, full-featured, and self-contained. More than that, it is the most suitable database engine to use with an android application.

Social & Market Feasibility:

Since smart phones are being used by almost everybody, I believe that a mobile application for our client “the sportswear store” will be of great benefit. Also, since there are no similar applications for other sportswear stores, this project will be one of a kind and it is expected to become the leader of sportswear stores mobile applications’ e-commerce.

After approaching the client « Ramo Sports » for the how they want their product to be developed and what are the minimum functions the product must perform; I decided that the project is feasible to develop and to be materialized in terms of “implementation, contribution of project to organization, cost constraints, and as per values and objectives of the organization.”

6 METHODOLOGY

Rapid Application Development Model

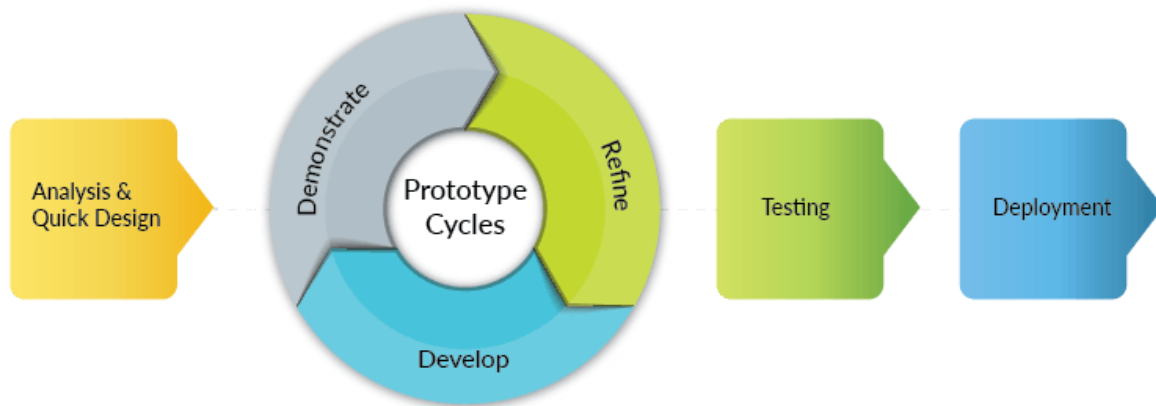


Figure 8 Rapid Application Development Model

From: <http://etutorials.xyz/sdlc-models/>

Concerning the system development methodology, I opted for the RAD (Rapid Application Development). Since it allows fast development and delivery of a high quality system, [3] it was the perfect methodology for my capstone project. In fact, I had permanent contact with the client throughout the different stages of software process. I developed each part of the project alone and had the client (Ramo Sports) test it and give me feedback about what should be changed or improved. This interaction saved me a lot of time, because the feedback was instantaneous. Accordingly, the user was involved in the design.

Also, since the project is an E-Commerce, the priority is to fulfill the business need of the company; which goes hand to hand with the main principles of the Rapid Application Development methodology. Moreover, since it is a capstone project, we have deadlines and we need to meet them. This among the basic principles of RAD. The presence of deadlines or “timeboxes” [3].

The main strengths of the RAD model:

- “The operational version of an application is available much earlier than with Waterfall, Incremental, or Spiral frameworks.” [3]
- “Because RAD produces systems more quickly and to a business focus, this approach tends to produce systems at a lower cost” [3]
- “Concentrates on essential system elements from user viewpoint.”[3]

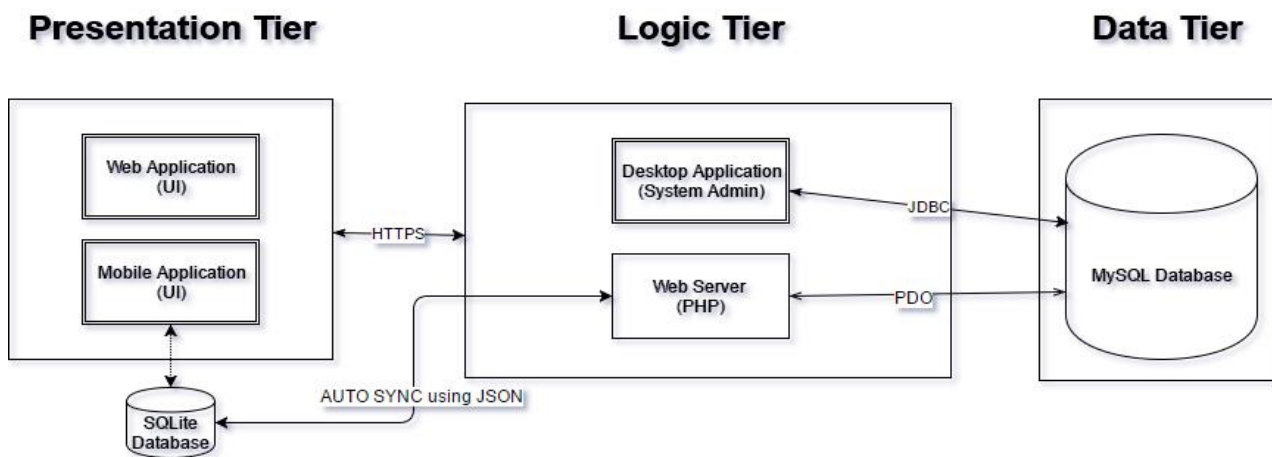
7 SOFTWARE ARCHITECTURE

The system architecture I opted for in the project is 3-tier application architecture. Because of the four main benefits it has; that most of them happen to be on the list of the non-functional requirements of the project:

- “Maintainability. Because each tier is independent of the other tiers, updates or changes can be carried out without affecting the application as a whole.”
- “Scalability. Because tiers are based on the deployment of layers, scaling out an application is reasonably straightforward.”
- “Flexibility. Because each tier can be managed or scaled independently, flexibility is increased.”
- “Availability. Applications can exploit the modular architecture of enabling systems using easily scalable components, which increases availability.” [1]

On the other hand, the system architecture I opted for is similar to the RESTful API architecture. The client connects to the web application or the mobile application in the presentation tier and makes orders. The orders are processed in the logic tier by the admin and then stored in the data tier. There is also an auto-synchronization of the SQLite database with MySQL one to add new products to the mobile application.

Figure 10 System Architecture



8 DESIGN

8.1 Use Case Diagram:

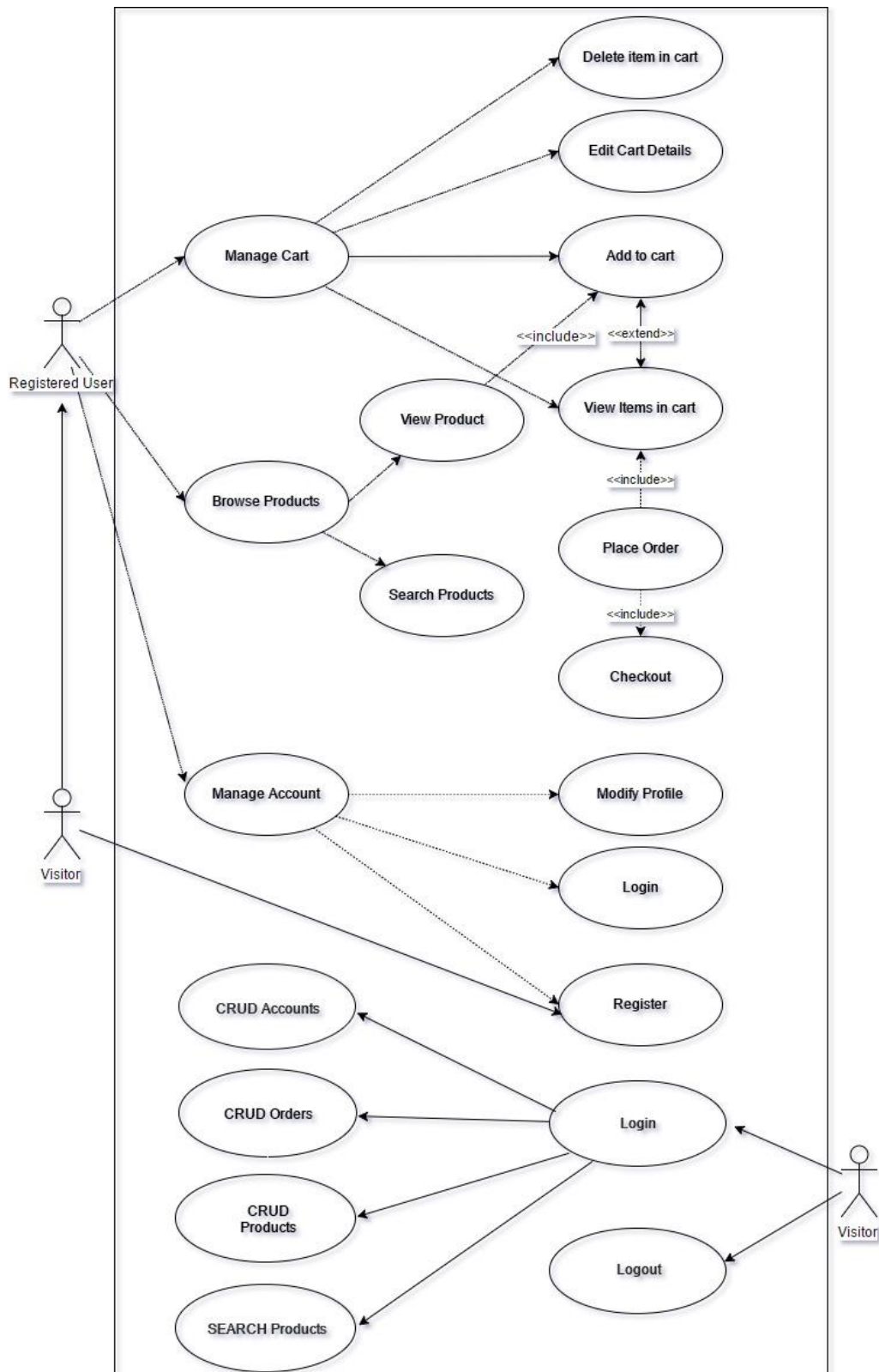


Figure 11 Use Case Diagram

The diagram above is a use case diagram which represents the mobile user interaction with the system of the mobile application as well as the administrator’s interaction with the desktop application. The diagram shows, depending on the user of the system, the different actions and relationship with each other. CRUD is the acronym for: Create, Read, Update, Delete, which are the 4 basic operations performed by the administrator of the system on its different components. The actors are:

- Administrator: Creates, Reads, Updates, and deletes accounts, orders, and products.
- Visitor: Registers to be able to search, view, and order products.
- Registered User: He needs to login to be able to create a cart and add products to it. Then he can place an order. He can also modify his profile.

8.2 Interaction Overview Diagram

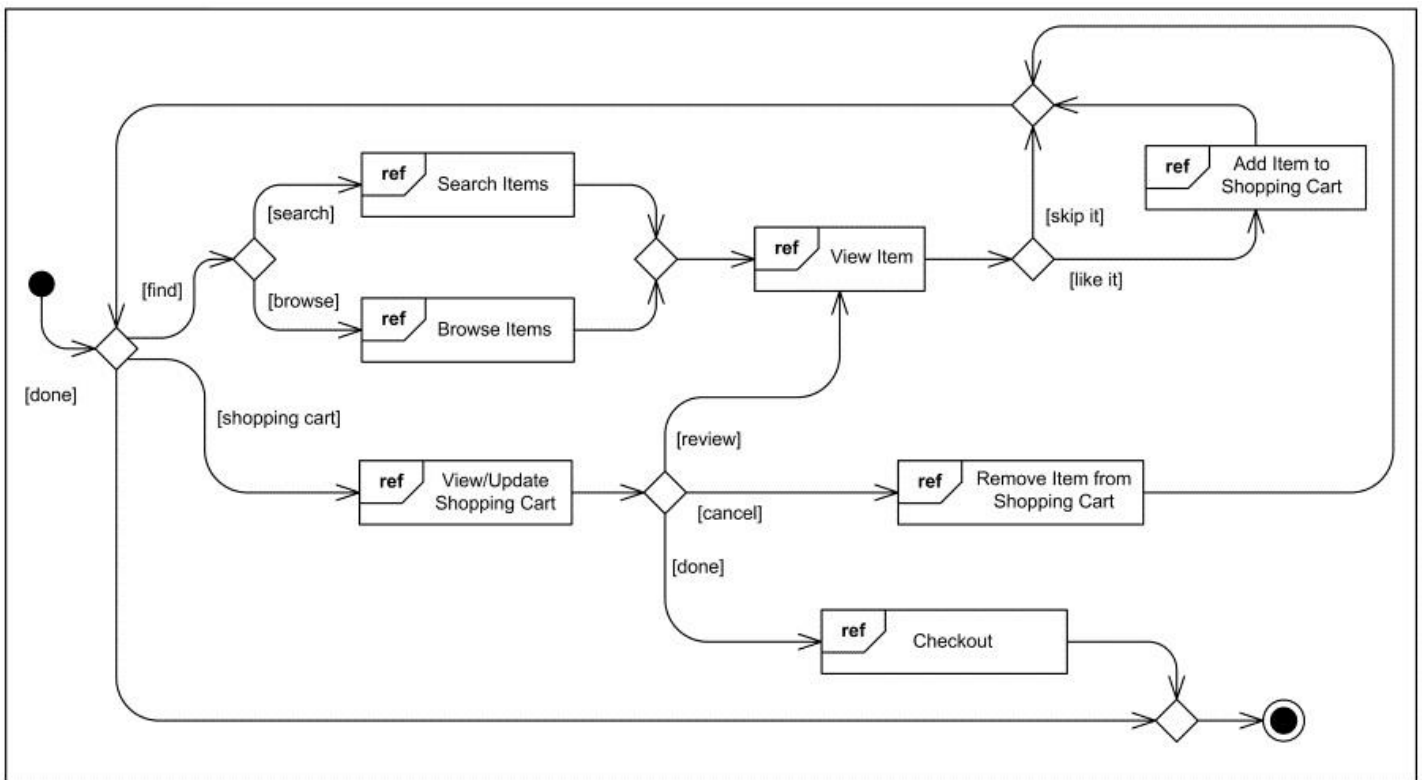


Figure 12 Interaction Overview Diagram

“Interaction overview diagrams provide overview of the flow of control where nodes of the flow are interactions or interaction uses. Interaction overview diagrams do look like activity diagrams that can only have inline interactions or interaction uses instead of invocation actions.” [2]

8.3 Data Model

Entity Relationship Diagram (ERD):

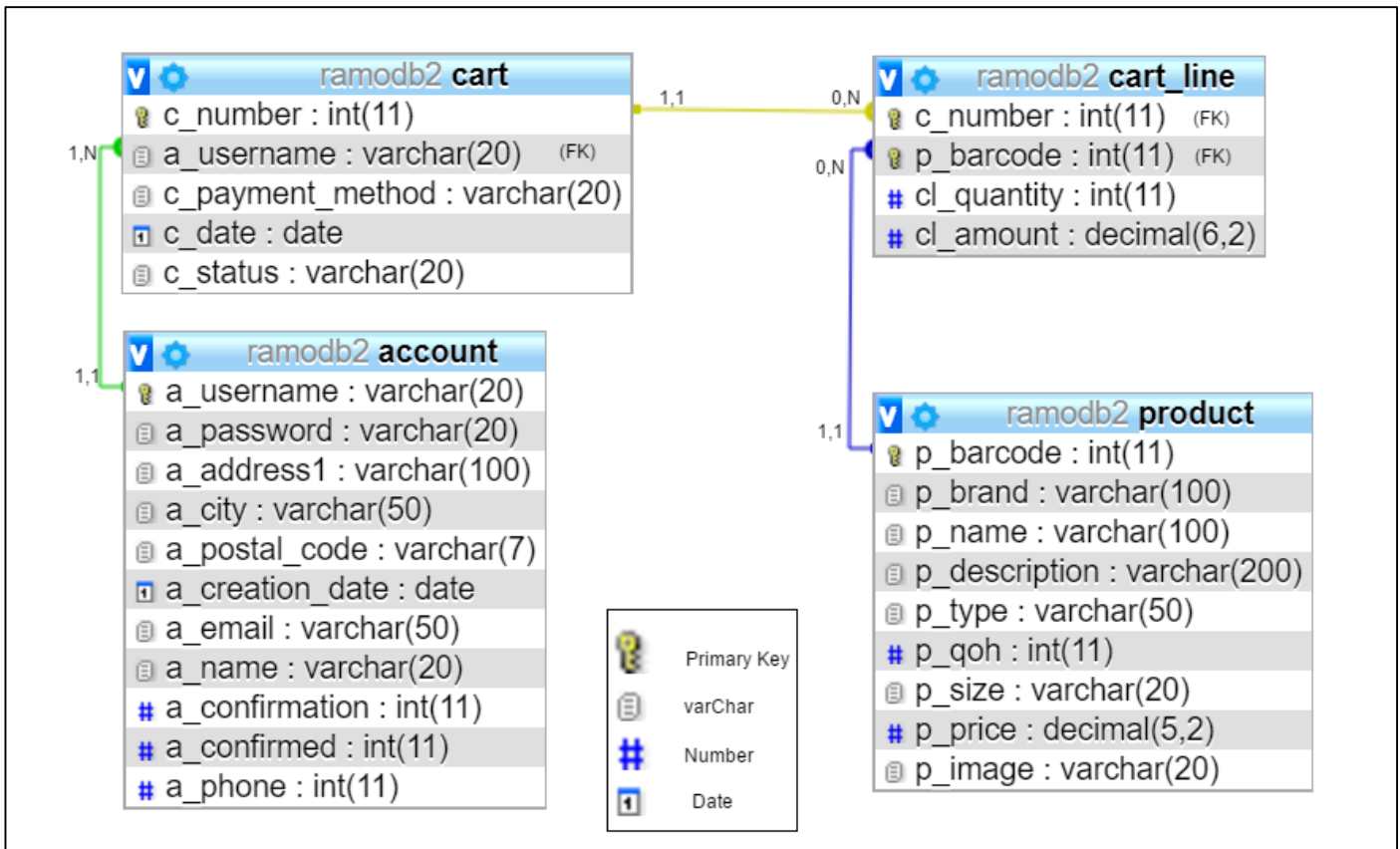


Figure 13 Entity Relationship Diagram Made using PHPMysqlAdmin

After applying the tools we learned in the Database Systems class, each table has its primary key and there is no transition relationship or a dependency relationship so in this case we can conclude that these tables are all in 3NF and that makes our implementation easier.

- **Cart** is an important entity involved in our business. This entity is linked to account entity and cart line entity. One cart can be used in one account and one account can use only one card. Moreover, Cart can have many cart lines while a cart line can have only one cart as a resource.
- **Cart line** is another important entity involved in our business. This entity is linked to entity cart which is described previously and Product entity. In fact, a cart line can have one and only one product while products can be in many cart lines.
- **Account** represents the accounts of the customers. It also holds a relationship with the table Cart, which is a unique feature of the mobile application that allows the customer to buy multiple different products online with different quantities.
- **Product**: This is one of the central and heaviest tables of our database. It contains a list of all the products (available or not) that have been, are, or will be provided by the store. It has a relationship with cart_line.

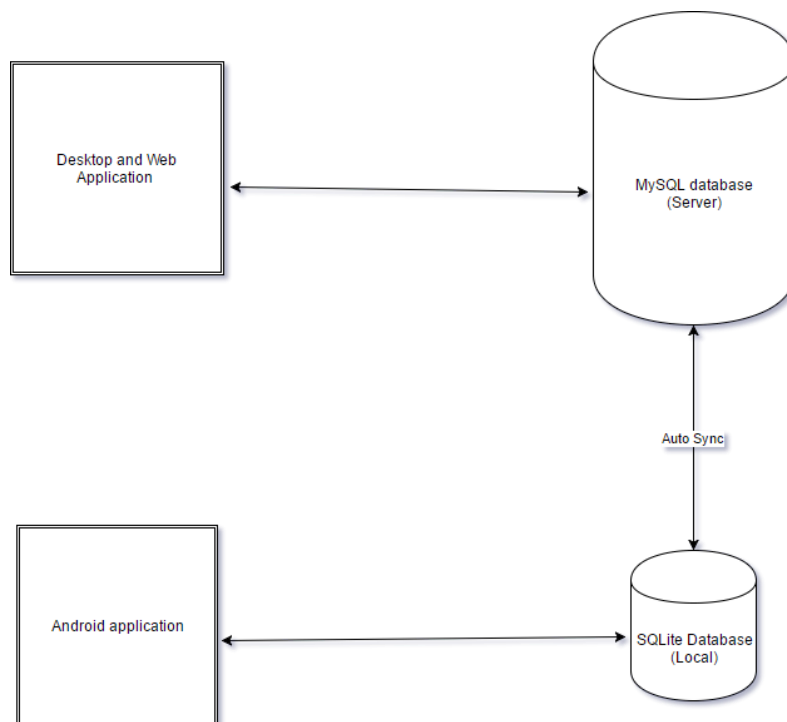
Table	Column	Data Type	Column Description	Table Description
Account	a_username	String	The username is the primary key of the table is chosen by the user.	The account table will store the information relevant to the user of the mobile application (and the web application) It stores the user's authentication info as well as his personal Information. The account/user will be able to create a cart and add cart lines to it. The a_username column will be a foreign key of the cart table.
	a_password	String	The password is chosen by the user and is used for authentication. It is encrypted using MD5 hash algorithm.	
	a_name	String	The full name of the user.	
	a_phone	String	The phone number of the user.	
	a_email	String	The email of the user, which will be confirmed using PHPMailer.	
	a_address	String	The address of the user.	
	a_city	String	The city of the user.	
	a_postal_code	String	The postal code of the user.	
	a_confirmation	Integer	This column indicates whether the confirmation email is sent.	
	a_confirmed	Integer	This column indicates if the email is confirmed.	
a_creation_date	Date	This column stores the creation date of the account and is automatically generated		
Product	p_barcode	Integer	The barcode of the product, it is the primary key of the table and it is auto-incremented.	The product table stores the product information, its barcode and a link to an image of the product.
	p_brand	String	The brand of the product.	
	p_name	String	The name of the product.	
	p_description	String	The description of the product.	
	p_type	String	The type/category of the product.	
	p_qoh	Integer	The quantity on hand (in the stock) of that product.	
	p_size	String	The size of the product.	
	p_price	Double	The price of the product.	
	p_image	String	The link of the image of the product.	

Cart	c_number	Integer	The cart number, it is the primary key of the table and it is auto-incremented.	The cart table stores general information about a cart. We can consider this table as an "order" table, because the admin of the system will receive the carts as orders.
	a_username	String	The username of the user who created the Cart, it is a foreign key.	
	c_date	Date	The date the cart was created.	
	c_payment_method	String	The payment method chosen by the user who created the cart or made the order.	
	c_status	String	The status of the order/cart (active, complete or canceled)	
Cart Line	c_number	Integer	The cart number is at the same time a foreign key and a primary key together with the product barcode. It references to the cart it belongs to.	The table Cart Line will store the different products in a cart/order. Each product in a cart will be stored in a different cart line and will reference to the same cart.
	p_barcode	Integer	The product barcode is at the same time a foreign key and a primary key together with the cart number.	
	cl_quantity	Integer	The quantity of a product added to the cart.	
	cl_amount	Double	The amount of money of a single cart line.	

9 IMPLEMENTATION

As mentioned before, the implementation of the web and desktop applications is not part of my project scope. They were implemented previously during the Database Design class. However, I had to do many changes in desktop application in order to improve it and to adapt it to the mobile application. I also had to change some things in the web application. These changes took me several days. Afterwards, I had to redesign the database in order to adapt it to the requirements of my project.

Altogether, I spent almost two weeks improving the previous parts of the project. Afterwards, I started implementing the mobile application. To be honest, it wasn't as easy as expected, I had to re-discover Android applications' development. I did a lot of research and learned a lot of new things before and during the implementation of the application. First, I took some time to design an attractive GUI. I used Photoshop for that. Then I started by implementing the Login and Register activities working with a SQLite local database. After the authentication, I started working on the display of the home screen and then the products, which are stored in the local database. This SQLite database is synchronized with the MySQL database connected with the desktop and web applications (hosted online), using JSON as follows.



The Login activity:

The following is the code and GUI of the Login activity:

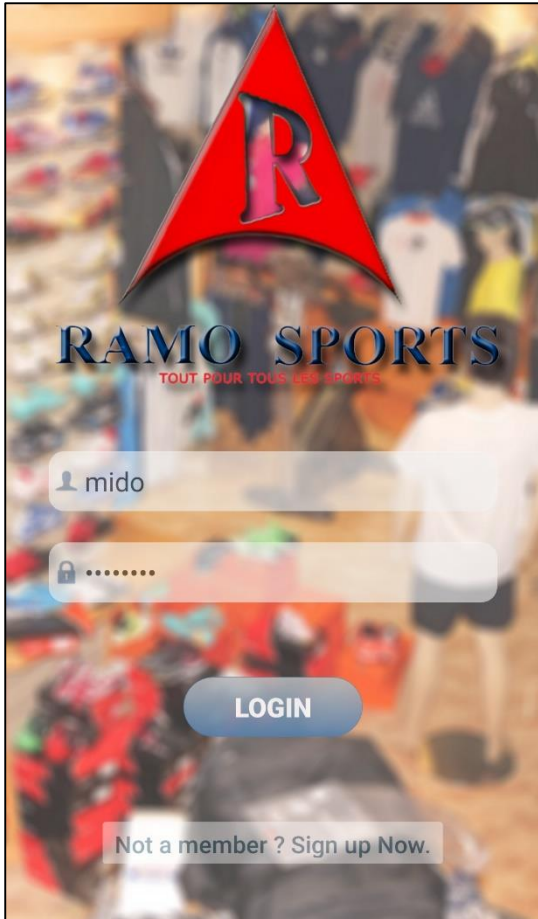


Figure 14 Mobile application Login

```
public class LoginActivity extends AppCompatActivity {
    Button register;
    Button login;
    DatabaseHelper myDb;
    EditText user, pass;
    String username, password;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        getWindow().setBackgroundDrawableResource(R.drawable.ic_background2);

        myDb = new DatabaseHelper(this);
        myDb.onCreate(myDb.getWritableDatabase());

        user = (EditText) findViewById(R.id.UsernameTextView);
        pass = (EditText) findViewById(R.id.PasswordTextView);
        login = (Button) findViewById(R.id.LoginButton112);

        login.setOnClickListener((v) -> { Login(); });
        register = (Button) findViewById(R.id.registerButton);
        register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(LoginActivity.this, Register.class);
                startActivity(intent);
            }
        });
    }

    public void Login() {
        username = user.getText().toString();
        password = pass.getText().toString();

        Boolean b = myDb.checkUser(username, password);

        if (!username.equals("") && !password.equals("")) {
            if (b == true) {
                Toast.makeText(LoginActivity.this, "Logged In", Toast.LENGTH_LONG).show();
                Intent intent = new Intent(getApplicationContext(), MenuActivity.class);
                intent.putExtra("NAME", username);
                startActivity(intent);
            } else {
                Toast.makeText(LoginActivity.this, "Please enter a valid Username/Password", Toast.LENGTH_LONG).show();
            }
        } else {
            Toast.makeText(LoginActivity.this, "Please enter your Username/Password", Toast.LENGTH_LONG).show();
        }
    }

    @Override
    public void onBackPressed() {
    }
}
```

This activity responsible for the authentication of the user. It is the first activity of the application. The user can choose to Login or Sign up if he is not registered yet.

Home and Navigation Drawer Activity:

After logging in, the Home screen is as follows:

This is the landing activity after the login.

And this is the navigation drawer menu.



Figure 15 Mobile application Main Menu

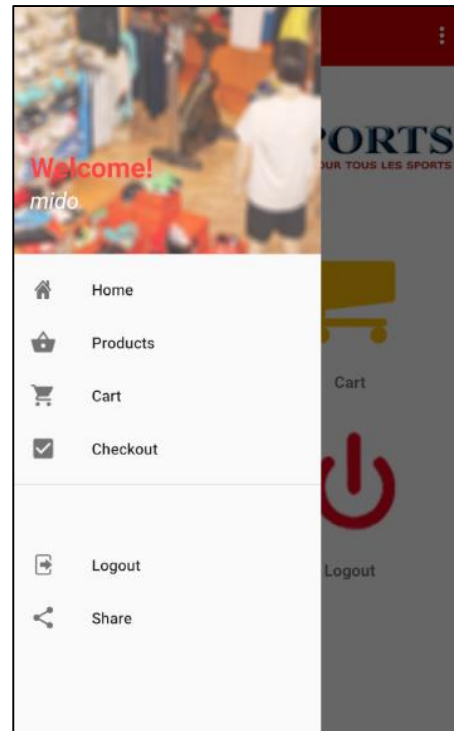


Figure 16 Mobile application Navigation Drawer

The code used to implement the navigation drawer and the Home activity is as follows:

```
public class MenuActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener {

    String name;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //Setting Default Fragment

        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        if (getFragmentManager().findFragmentById(R.id.fragment_container) == null) {
            setTitle("Home");
            ProductsFragment PF = new ProductsFragment();
            FragmentManager FM = getSupportFragmentManager();
            FM.beginTransaction().replace(R.id.fragment_container, PF).commit();
        }

        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            this, drawer, toolbar, R.string.navigation_drawer_open,
            R.string.navigation_drawer_close);
        drawer.setDrawerListener(toggle);
        toggle.syncState();

        NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);

        name = getIntent().getStringExtra("NAME");

        View header=navigationView.getHeaderView(0);

        TextView nameTV = (TextView)header.findViewById(R.id.usernameTV);
        nameTV.setText(name);
    }
}
```

```
@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();

    if (id == R.id.nav_products) {
        setTitle("Products");
        Intent intent = new Intent(getBaseContext(), Categories.class);
        intent.putExtra("NAME", name);
        startActivity(intent);
    } else if (id == R.id.nav_cart) {
        setTitle("Carts");
        CartFragment CF = new CartFragment();
        FragmentManager FM = getSupportFragmentManager();
        FM.beginTransaction().replace(R.id.fragment_container, CF).commit();
    } else if (id == R.id.nav_checkout) {
    }

    } else if (id == R.id.nav_logout) {
        Intent intent = new Intent(getBaseContext(), LoginActivity.class);
        startActivity(intent);
    } else if (id == R.id.nav_share) {
        Intent intent = new Intent(getBaseContext(), ProductActivity.class);
        startActivity(intent);
    }
    } else if (id== R.id.nav_home){
        setTitle("Home");
        ProductsFragment PF = new ProductsFragment();
        FragmentManager FM = getSupportFragmentManager();
        FM.beginTransaction().replace(R.id.fragment_container, PF).commit();
    }
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}
}
```

Then, the user will be able to view the categories of products and browse the products in each category:

This is some of the code responsible for this activity:



```
Categories | getData()
username = getIntent().getStringExtra("NAME");

gridView.setOnItemClickListener((parent, v, position, id) -> {
    String caaaaat= imageItems.get(position).getTitle();
    Toast.makeText(Categories.this, caaaaat, Toast.LENGTH_LONG).show();
    Intent intent = new Intent(Categories.this, ProductActivity.class);
    intent.putExtra("category", caaaaat);
    intent.putExtra("NAME", username);
    startActivity(intent);
});

private ArrayList<ImageItem> getData() {
    TypedArray imgs = getResources().obtainTypedArray(R.array.image_ids);
    for (int i = 0; i < imgs.length(); i++) {
        Bitmap bitmap = BitmapFactory.decodeResource(getResources(), imgs.getResourceId(i, -1));
        switch (i) {
            case 0:
                imageItems.add(new ImageItem(bitmap, "Hats"));
                break;
            case 1:
                imageItems.add(new ImageItem(bitmap, "Gloves"));
                break;
            case 2:
                imageItems.add(new ImageItem(bitmap, "Balls"));
                break;
            case 3:
                imageItems.add(new ImageItem(bitmap, "Men Sport Suit"));
                break;
        }
    }
}
```

Figure 17: Mobile application's Category List

After choosing a category, the application displays the list of products under that category in a gridView. This is done using a function call in the DatabaseHelper.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_product);

    this.category = getIntent().getStringExtra("category");
    this.username = getIntent().getStringExtra("NAME");

    gridView = (GridView) findViewById(R.id.gridView1);
    gridAdapter = new AdapterGridViewCARDS(this, R.layout.category_grid_item3, getProducts(category));
    gridView.setAdapter(gridAdapter);
    searchButton = (ImageButton) findViewById(R.id.btnSearch);
    edtKeyword = (EditText) findViewById(R.id.edtKeyword);

    final Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    final android.support.v7.app.ActionBar actionBar = getSupportActionBar();
    if (actionBar != null) {
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setTitle("Category");
    }

    username = getIntent().getStringExtra("NAME");
    gridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        public void onItemClick(AdapterView<?> parent, View v, int position, long id) {
            Integer barcode= cardItems.get(position).getBarcode();
            Toast.makeText(ProductActivity.this, String.valueOf(barcode), Toast.LENGTH_LONG).show();
            Intent intent = new Intent(ProductActivity.this, ProductDetails.class);
            intent.putExtra("barcode", barcode);
            intent.putExtra("NAME", username);
            startActivity(intent);
        }
    });
}
```

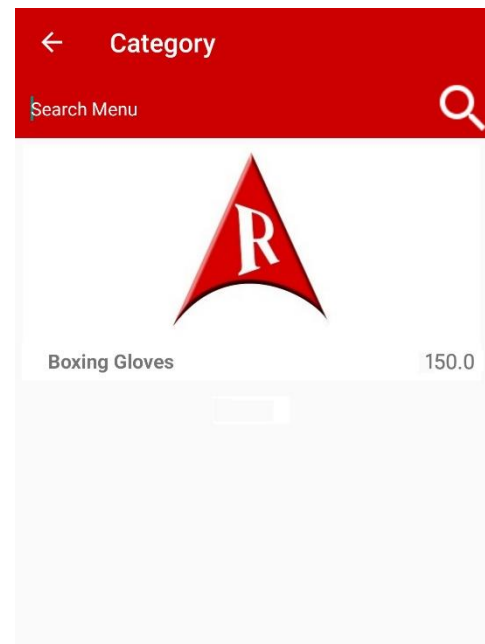


Figure 17 Mobile application Product by Category

After finding the product he is looking for, the user can view its details.

Then he can add the product to his cart.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_product_details);
    FloatingActionButton btnAdd = (FloatingActionButton) findViewById(R.id.btnAdd);

    final DatabaseHelper myDb = new DatabaseHelper(this);

    this.username = getIntent().getStringExtra("NAME");

    this.barcode = getIntent().getStringExtra("barcode", 0);
    product = myDb.getProductByBarcode(barcode);

    String mDrawableName = product.getImage();
    int resID = getResources().getIdentifier(mDrawableName, "drawable", getPackageName());
    Bitmap bitmap = BitmapFactory.decodeResource(getResources(), resID);

    ImageView productImg = (ImageView) findViewById(R.id.productImg);
    TextView nameV = (TextView) findViewById(R.id.txtName);
    TextView priceV = (TextView) findViewById(R.id.txtPrice);
    TextView descriptionV = (TextView) findViewById(R.id.txtDescription);

    final TextView testCart = (TextView) findViewById(R.id.displayCarts);
    testCart.setMovementMethod(new ScrollingMovementMethod());

    final TextView testCartLine = (TextView) findViewById(R.id.displayCartLines);
    testCartLine.setMovementMethod(new ScrollingMovementMethod());

    nameV.setText(product.getName());
    priceV.setText(String.valueOf(product.getPrice()));
    descriptionV.setText(product.getDescription());
    productImg.setImageBitmap(bitmap);

    btnAdd.setOnClickListener((v) -> {
        Integer cNumber=myDb.getUserCart(username);
        if (cNumber == 0) {
            long cartNumber = myDb.createCart(username, "Cash on delivery");
            cNumber = (int) cartNumber;
            Toast.makeText(ProductDetails.this, "Cart created number :"+ cNumber, Toast.LENGTH_SHORT).show();
        }

        Integer clQuantity = myDb.CartLineExists(barcode, cNumber);
        if (clQuantity==0) {
            myDb.createCartLine(cNumber, barcode, 1);
            Toast.makeText(ProductDetails.this, "Cart Line Created", Toast.LENGTH_SHORT).show();
        }
    });
}

```

```

public void parseJSONDataTax () {
    try {
        HttpClient client = new DefaultHttpClient();
        HttpConnectionParams.setConnectionTimeout(client.getParams(), 15000);
        HttpConnectionParams.setSoTimeout(client.getParams(), 15000);
        HttpUriRequest request = new HttpGet();
        HttpResponse response = client.execute(request);
        InputStream atomInputStream = response.getEntity().getContent();

        BufferedReader in = new BufferedReader(new InputStreamReader(atomInputStream));

        String line;
        String str = "";
        while ((line = in.readLine()) != null) {
            str += line;
        }

        // parse json data and store into tax and currency variables
        JSONObject json = new JSONObject(str);
        JSONArray data = json.getJSONArray("data"); // this is the "items: [ ] part

        JSONObject object_tax = data.getJSONObject(0);
        JSONObject tax = object_tax.getJSONObject("tax_n_currency");

        Tax = Double.parseDouble(tax.getString("Value"));

        JSONObject object_currency = data.getJSONObject(1);
        JSONObject currency = object_currency.getJSONObject("tax_n_currency");

        Currency = currency.getString("Value");
    } catch (MalformedURLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        IOException = e;
        e.printStackTrace();
    } catch (JSONException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```



Boxing Gloves

150.0

Ramo Shoes, football



(Add to cart)

Then the user can view his shopping cart, modify, and process checkout

Here is an example of the code responsible of parsing JSON:

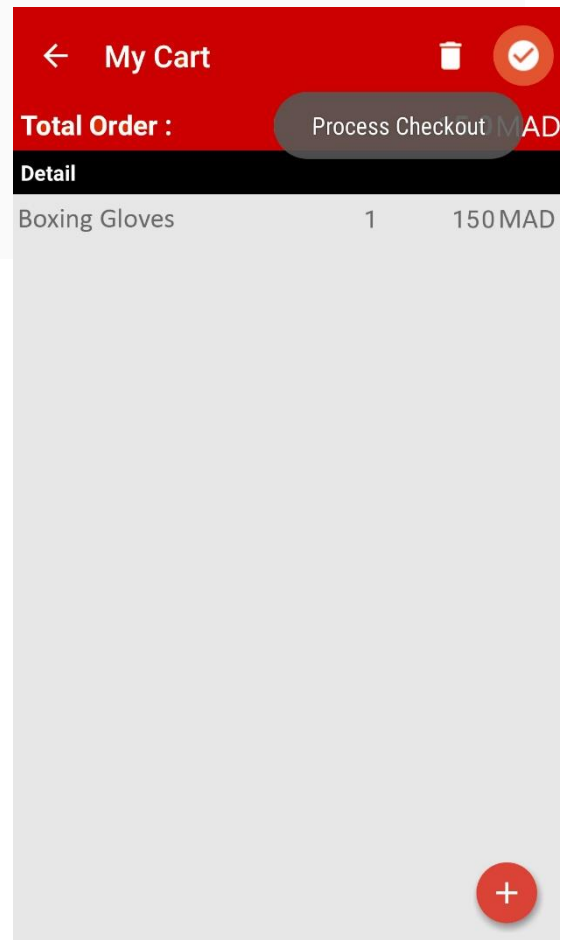


Figure 18 View Cart

Checkout:

Afterwards, the user can perform the checkout and send his order. Here is some of the code responsible for that.

```
btnTime.setOnClickListener((v) -> {
    // TODO Auto-generated method stub
    // show time picker dialog
    DialogFragment newFragment = new TimePickerFragment();
    newFragment.show(getSupportFragmentManager(), TIME_DIALOG_ID);
});

// event listener to handle send button when pressed
btnSend.setOnClickListener((arg0) -> {
    // TODO Auto-generated method stub

    // get data from all forms and send to server
    Name = edtName.getText().toString();
    Alamat = edtAlamat.getText().toString();
    Kota = edtKota.getText().toString();
    Provinsi = edtProvinsi.getText().toString();
    Email = edtEmail.getText().toString();
    Name2 = edtName2.getText().toString();
    Date = btnDate.getText().toString();
    Time = btnTime.getText().toString();
    Phone = edtPhone.getText().toString();
    Comment = edtComment.getText().toString();
    Date_n_Time = Date+" "+Time;
    if(Name.equalsIgnoreCase("") || Name2.equalsIgnoreCase("") || Email.equalsIgnoreCase("") ||
        Alamat.equalsIgnoreCase("") || Kota.equalsIgnoreCase("") || Provinsi.equalsIgnoreCase("") ||
        Date.equalsIgnoreCase("Set Date") ||
        Time.equalsIgnoreCase("Set Time") ||
        Phone.equalsIgnoreCase("")){
        Toast.makeText(ActivityCheckout.this, R.string.form_alert, Toast.LENGTH_SHORT).show();
    }else if((data.size() == 0)){
        Toast.makeText(ActivityCheckout.this, "You have not ordered yet", Toast.LENGTH_SHORT).show();
    }else{
        new sendData().execute();
    }
});

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
```

← Checkout

Your Name
Mido

Your Address

Your City

Your Province

Cash On Delivery ▾

SET DATE SET TIME

Phone Number
0635258960

Your Email
63029@ai.ma



Comment (Optional)

SEND

10 TECHNOLOGY ENABLERS





Desktop Application:

For the desktop application that was developed before, the following technology enablers were used:

JAVA		JAVA was used to develop the whole the desktop application
MySQL		MySQL was used for the backend, it is linked to both the desktop and web application.

Web Application:

For the web application that was developed before as well, the following technology enablers were used:

HTML		HTML was used as a markup language for the web application
CSS		CSS was used for the design of the Web Application
PHP		PHP combined with HTML was used for the scripting.
Bootstrap CSS Framework		Bootstrap was used to make the web application look fancier.




Mobile Application:

For the mobile application that is to be developed during the implementation phase of the project, these technology enablers are going to be used:

Android SDK		Android SDK is used to develop the mobile application
SQLite		As the android SDK comes with a package that contains SQLite specific classes, SQLite is used for the local database of the mobile application.

Other Tools:

These tools were used for different purposes:

Photoshop		Photoshop was used to design the graphical user interface of the mobile application's activities. (Backgrounds, Images, and icons)
Draw.io		Draw.io was used to draw the different diagrams of the design part of the project.
Visual Studio		Visual Studio was used to modify the code for the web, desktop and mobile application.

11 TESTING

Testing was done throughout the development of the project. In fact, I had an android phone connected with the computer. Whenever I add a feature or a functionality, I test it right away. This helped me find errors right away and have each and every functionality of the application tested. I used <http://www.freesqldatabase.com/> to host the MySQL database and test the synchronization of MySQL and SQLite database using JSON thanks to the free hosting service provided by the website.

This is a small example for testing the connection of the mobile app with the MySQL database in order to synchronize the new registered account.

```
setup.php db_sync.php add.php Untitled-2.java variables.php activity.ja

1  <?php
2  $user_name = "sql11209654";
3  $password = "WCSbDXbx5t";
4  $server = "sql11.freesqldatabase.com:3306";
5  $db_name = "sql11209654";
6
7  $con = mysqli_connect($server, $user_name, $password, $db_name);
8  if($con){
9      $Name = $_POST['a_username'];
10     $Pass = md5($_POST['a_password']);
11     $query = "insert into account(a_username) values('".$Name."', '".$Pass."')";
12     $result = mysqli_query($con,$query);
13
14     if($result){
15         $status = 'OK';
16     }
17     else {
18         $status = 'FAILED';
19     }
20 }
21 else {$status = 'FAILED';}
22
23 echo json_encode(array("response"=>$status));
24
25 mysqli_close($con);
26
27 ?>
28 |
```

12 Conclusion & Future Perspectives:

Obviously, there is still a lot to work on in the mobile application if I want to launch it to the market. There is also the possibility of developing the iOS version of the application. This mobile e-shopping store is promising and will certainly boost Ramo Sports' business. In fact, even if the application doesn't bring the company many orders (through the application), it could serve as a marketing strategy. This will be done after adding the push notifications to it and sending notifications to users whenever there is a new product or a new discount. That is among the reasons why I chose to have the user register in the application, in order to have his data. This will allow us to send him promotions to his phone as well as his email address.

I find it amazing how I learned mobile application development using Android throughout the semester while developing the application at the same time. I made use of Youtube.com tutorials and stackoverflow.com's solution to each and every problem I face while developing the app.

More than that, working on this capstone project was a great opportunity for me to put the knowledge, I've acquired during these four years, into practice. However, it showed me some of my weak points and helped me transform them into strong points.

Now, I know how to gather requirements, design, implement, and test a software. In addition, I learned how to implement the server side, the client side and the data side, each one separately from the other one.

I also got to learn how to analyze the social, technological, economical, ecological, political, legal, and ethical aspects of a project and get the best out of it.

13 References:

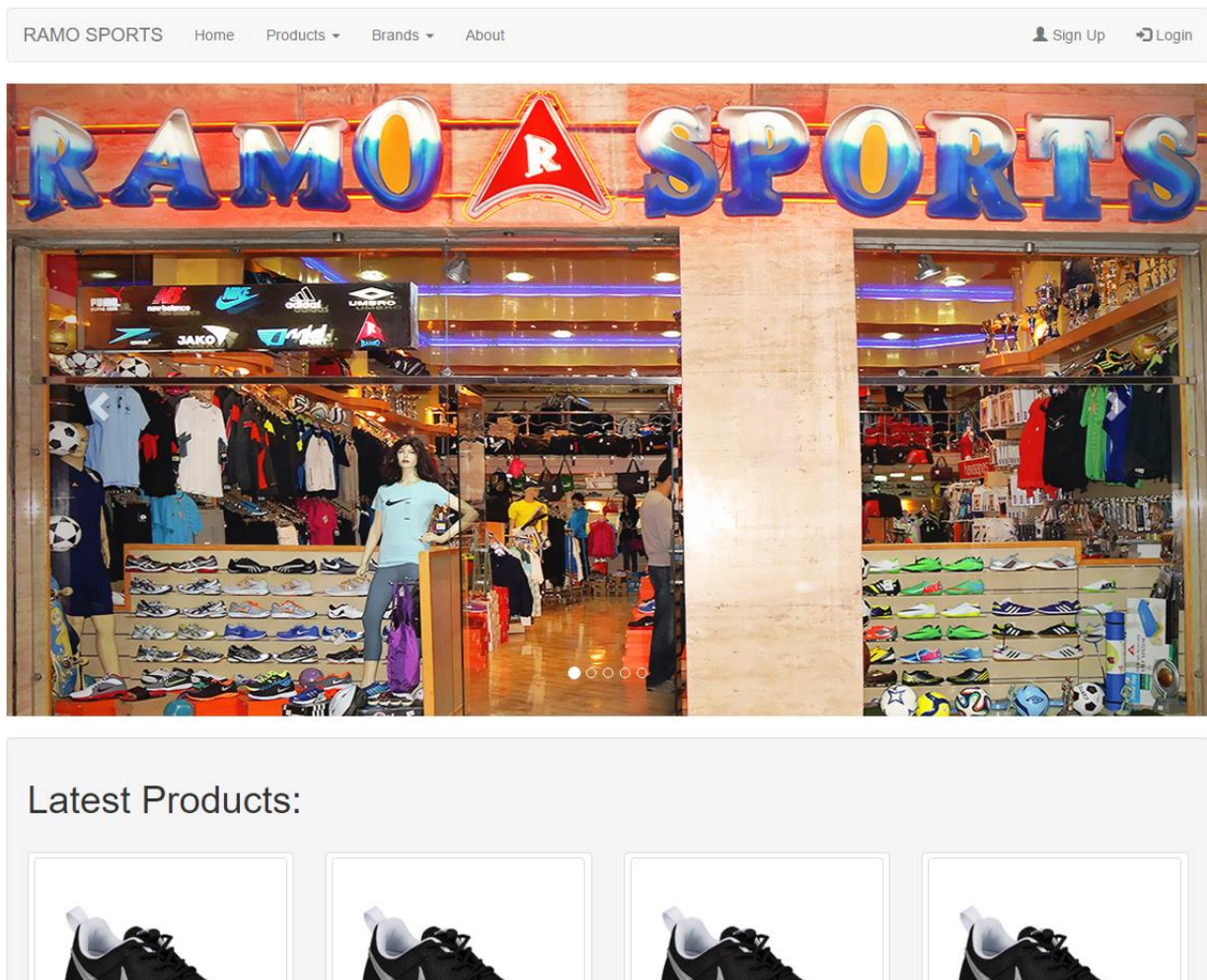
- [1]: Concepts of Three-Tier Architecture. (2011, February 28). Retrieved December 03, 2017, from <https://alitarhini.wordpress.com/2011/01/22/concepts-of-three-tier-architecture/>

- [2]: Fakhroutdinov, K. (n.d.). The Unified Modeling Language. Retrieved December 03, 2017, from <https://www.uml-diagrams.org/>



- [3]: SELECTING A DEVELOPMENT APPROACH. (n.d.). Retrieved December 3, 2017, from <https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>

Appendix A: Web Application user manual

For the Web application here is some instruction manual steps:



Full website implemented with login signup and error handling as well as input checking to avoid mysql injection.

 Sign Up  Login

Username

Password

Sign in

SPORTS Home Pro

Enter your information: ✕

Username:

Password:

Email:

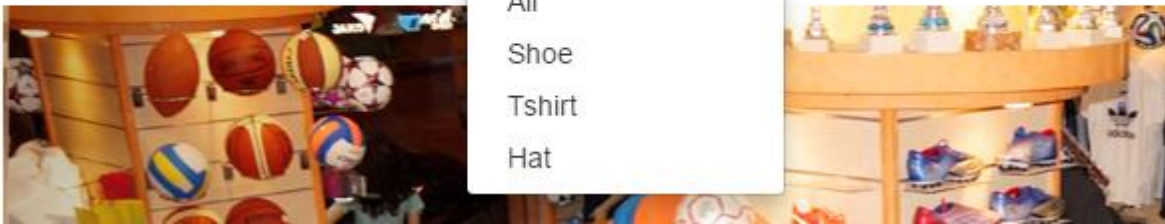
Name:

Phone:

Address:

City:

Postal Code:

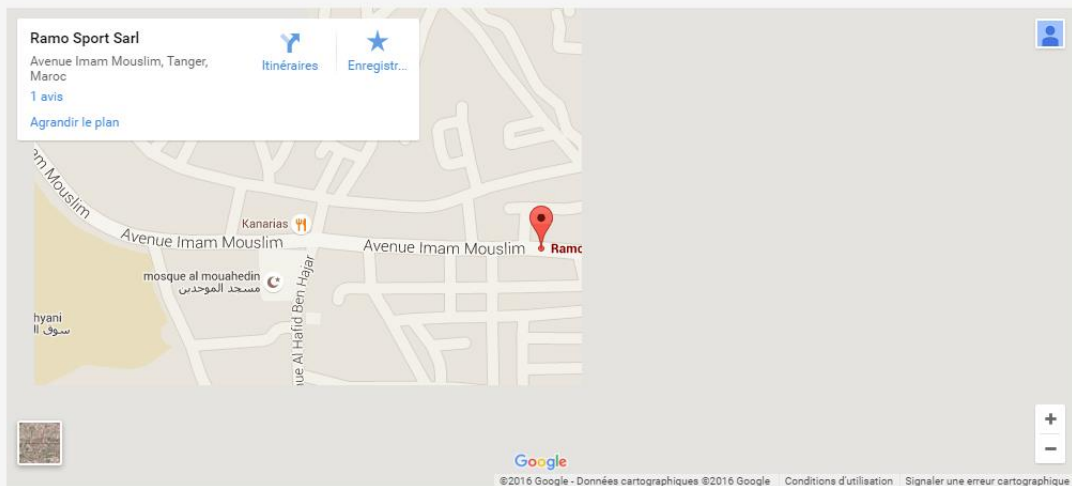


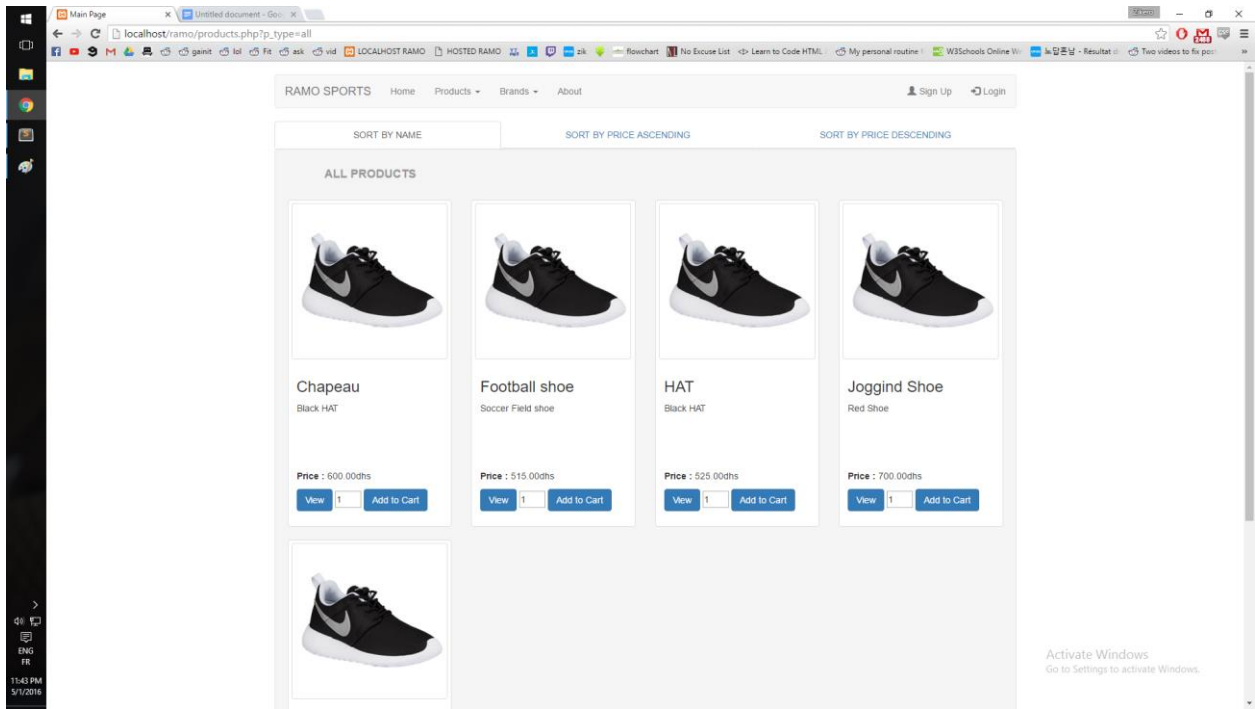
Votre magasin de sport répondant à tous vos besoins

RAMO SPORTS est un magasin qui propose différents vêtements et accessoires de sport pour Homme, Femme et Enfant qui répond aux besoins de toute la famille.

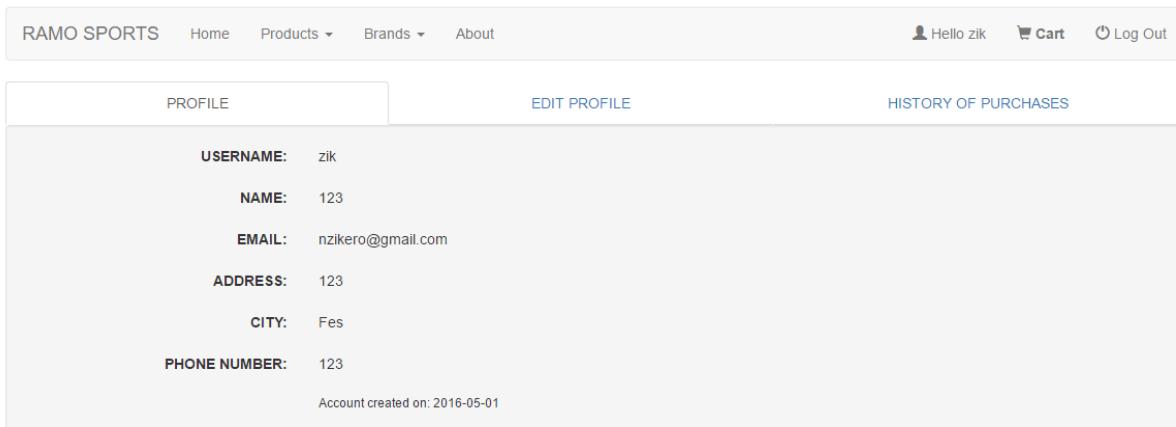
Retrouvez également sur RAMO SPORTS les grandes marques leaders sur leur segment de marché et leurs produits ainsi que d'autres accessoires sélectionnés par nos soins.

Spécialiste de la distribution d'articles de sport au Maroc. RAMO SPORTS met à disposition de ses clients une grande variété de produits sportifs. Commandes et livraisons garanties.





Products are sorted by type/brand and by alphabetical/price order.



RAMO SPORTS Home Products ▾ Brands ▾ About Hello zik Cart Log Out

PROFILE EDIT PROFILE HISTORY OF PURCHASES

Username:

Password:

Email:

Name:

Phone:

Address:

City: ▾

Postal Code:

Profile and update functions along with a history of all purchases made.

[PROFILE](#)
[EDIT PROFILE](#)
[HISTORY OF PURCHASES](#)

CART NUMBER : 3261
 Issued on : 2016-05-01
 STATUS: active

Product Barcode	Product Name	Quantity	Unit Price	Total Price
2031	Joggind Shoe	1	700.00	700.00

CART NUMBER : 3251
 Issued on : 2016-05-01
 STATUS: active



Product Barcode	Product Name	Quantity	Unit Price	Total Price
2011	HAT	1	525.00	525.00
2021	Chapeau	1	600.00	600.00

CART NUMBER : 3241
 Issued on : 2016-05-01
 STATUS: active

Product Barcode	Product Name	Quantity	Unit Price	Total Price
2011	HAT	1	525.00	525.00
2021	Chapeau	1	600.00	600.00


CART NUMBER : 3181
 Issued on : 2016-05-01
 STATUS: active

Product Barcode	Product Name	Quantity	Unit Price	Total Price
2011	HAT	1	525.00	525.00
2021	Chapeau	1	600.00	600.00

RAMO SPORTS Home Products ▾ Brands ▾ About Hello zik  Cart  Log Out

[SORT BY NAME](#)
[SORT BY PRICE](#)


ALL PRODUCTS



Chapeau
Black HAT

Price : 600.00dhs


[View](#) [Add to Cart](#)



Football shoe
Soccer Field shoe

Price : 515.00dhs


[View](#) [Add to Cart](#)



HAT
Black HAT

Price : 525.00dhs

[View](#) [Add to Cart](#)



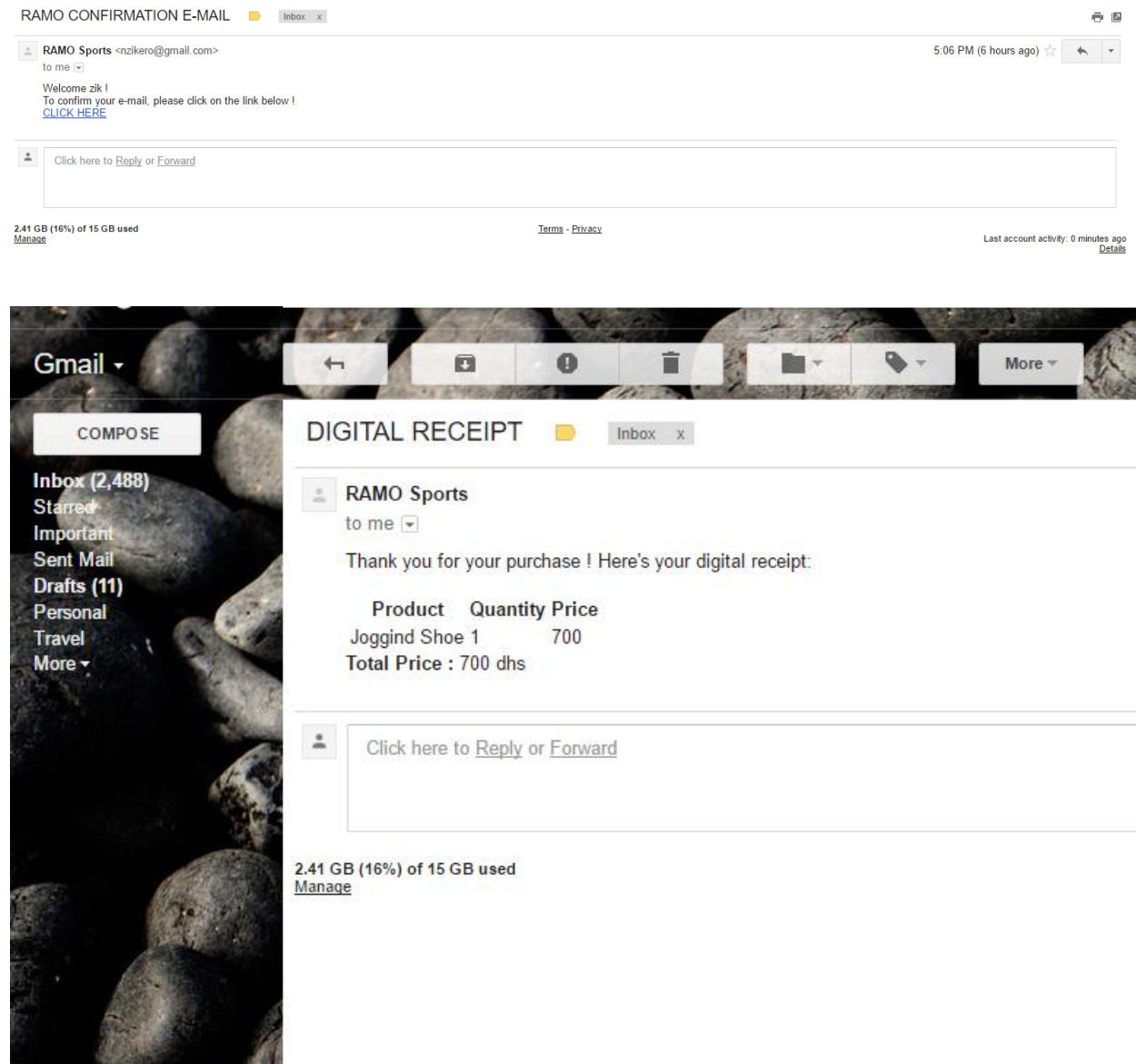
Joggind Shoe
Red Shoe

Price : 700.00dhs

[View](#) [Add to Cart](#)

Product	Quantity	Price	Delete
Chapeau	5	3000	×
Total Price : 3000 dhs			Confirm & Pay

Fully functional cart system with php sessions.



SMTP server allows us to send emails to confirm account and to send digital receipts.