

# E-BIKE SHOP

**ALESSANDRO  
GIONANGELI**

**MATRICOLA: 305082**



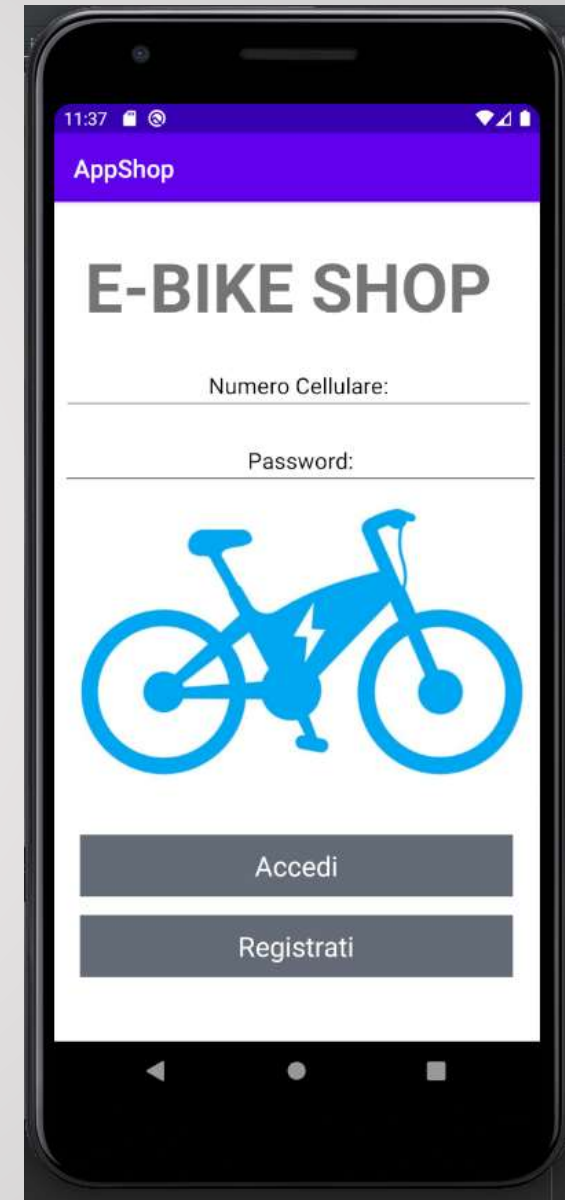
# Presentazione progetto:

## OBIETTIVO:

Realizzare un negozio elettronico che permetta la vendita di prodotti per la mobilità sostenibile.

## FUNZIONALITA':

- **Erogatori:** Ai proprietari dell' e-commerce è concesso di inserire i prodotti all'interno del catalogo e una volta fatto, di visionarli ed eliminarli.
- **Fruitori:** Gli utenti usufruendo del servizio possono selezionare i prodotti interessati ed una volta scelti procedere con il pagamento.



# Struttura applicazione

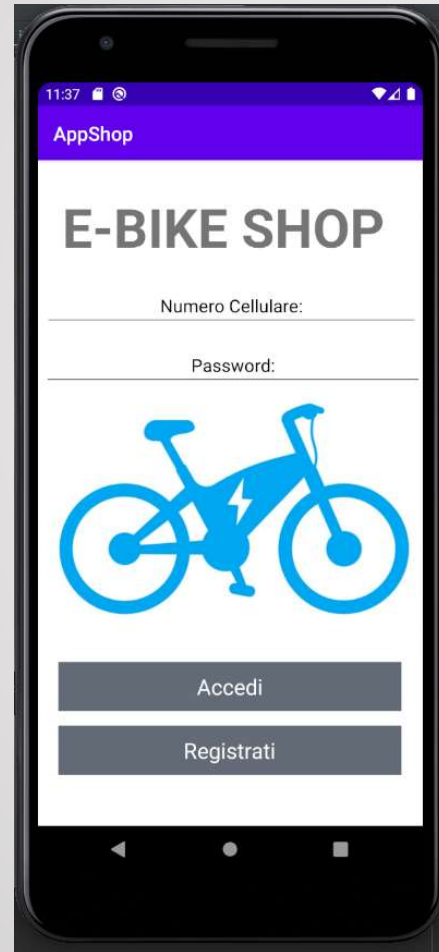
## ANALISI PROGETTO:

L'applicazione è realizzata in linguaggio Kotlin, linguaggio di programmazione di Google nato per la realizzazione di applicazioni Android.

- **FRONT-END:** Le pagine dell'app (Activity) sono disegnate, per quanto riguarda l'aspetto grafico, utilizzando il linguaggio XML. Il tutto è strutturato in maniera minimale per rendere più user friendly possibile l'utilizzo, senza particolari possibilità di distrazioni ed errori da parte degli utilizzatori.
- **BACK-END:** Sotto questo aspetto l'applicazione si serve del software XAMPP che consente di attivare in locale un servizio di server web Apache e un database PhpMyAdmin. All'interno di un progetto realizzato con il programma NetBeans è racchiuso il codice PHP e le query in SQL che andranno a comunicare con il database. Per quanto riguardano le richieste http da Android Studio al server si integra una libreria chiamata "Volley" che consente con poche righe di codice di comunicare lato client/server.

# Front-end

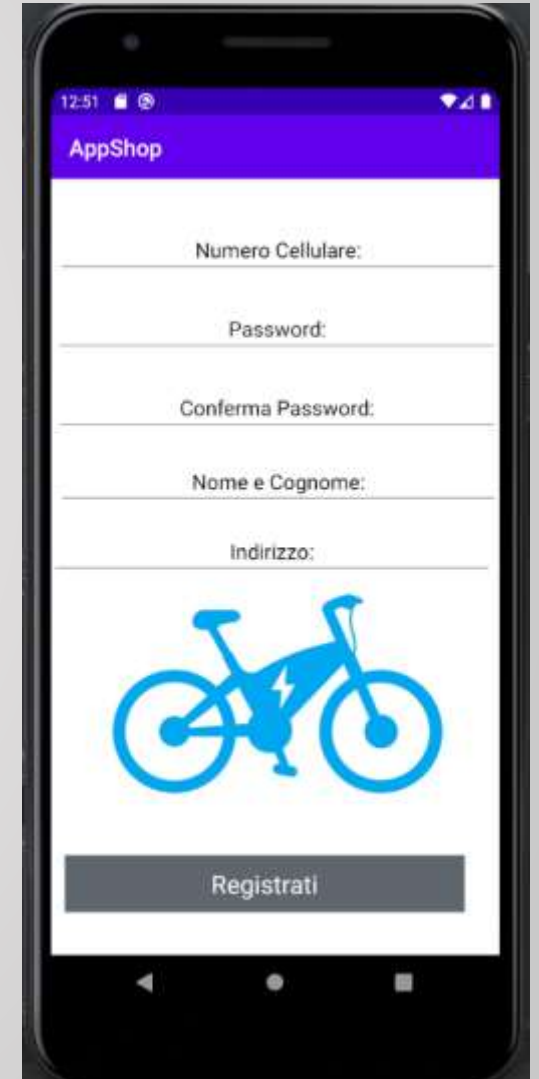
- Nella activity di apertura si presenta all'utilizzatore la possibilità di accedere al negozio elettronico inserendo le credenziali oppure di procedere con la registrazione se non si dispone di un account.



## AREA CLIENTI E AMMINISTRATIVA:

← La pagina del login prevede l'inserimento del numero di telefono e di una password

La pagina di registrazione richiede l'inserimento di alcuni campi obbligatori →



# Front-end

- Se l'account registrato è di tipo amministratore allora l'applicazione non rimanderà alla pagina di accesso dei clienti dove è possibile selezionare la categoria di prodotto da acquistare, ma si accederà all'area amministrativa.



## AREA CLIENTI E AMMINISTRATIVA:

← L'area amministrativa consente all'erogatore del servizio la possibilità di inserire prodotti nel catalogo o di eliminarli se sono già all'interno

Se si è un fruitore allora si verrà rimandati nella pagina di selezione della categoria →



# Front-end

## DESIGN APPLICAZIONE:

- Le pagine sono disegnate secondo il linguaggio XML e ogni parte del design è collegata ad una funzione gestita dal linguaggio Kotlin

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="horizontal"
    android:padding="5dp"
    android:background="@drawable/gradient"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <ImageView
        android:id="@+id/item_photo"
        android:layout_width="150dp"
        android:layout_height="150dp"
        app:srcCompat="@mipmap/ic_launcher_round" />

    <LinearLayout
        android:layout_width="300dp"
        android:layout_height="140dp"
        android:orientation="vertical">

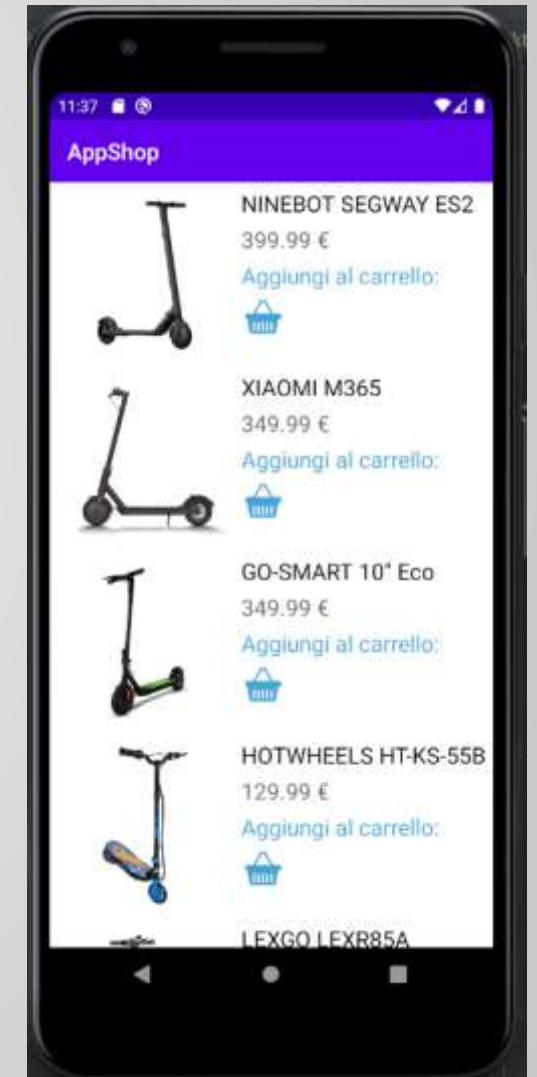
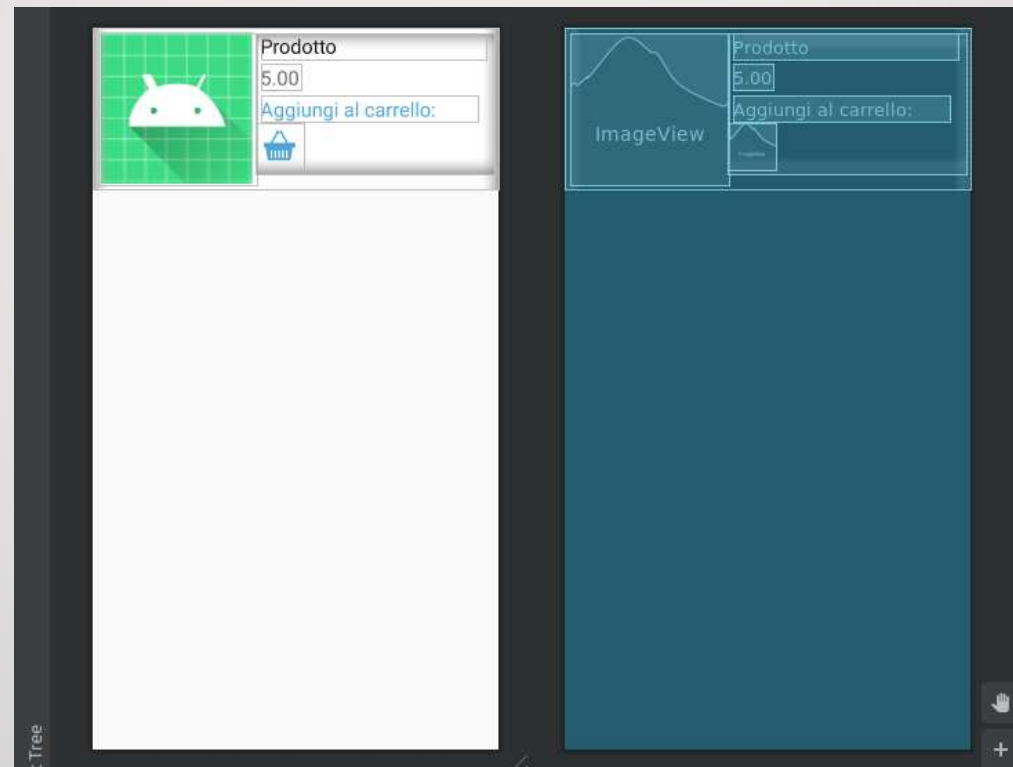
        <TextView
            android:id="@+id/item_name"
            android:layout_width="220dp"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:text="Prodotto"
            android:textColor="@android:color/black"
            android:textSize="18sp" />

        <TextView
            android:id="@+id/item_price"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:layout_marginTop="5dp"
            android:text="5.00"
            android:textColor="@android:color/black"
            android:textSize="18sp" />

        <TextView
            android:layout_width="220dp"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:text="Aggiungi al carrello:"
            android:textColor="@android:color/black"
            android:textSize="18sp" />

    </LinearLayout>

</LinearLayout>
```





L'app ha uno stile architettoriale basato sull'acronimo REST(Representational State Transfer) e attraverso l'uso del protocollo HTTP per interfacciare il client e il server vengono fatte richieste utilizzando il metodo GET, POST, PUT, DELETE

- **GET:** è consigliato in quelle richieste in cui nell'URL è necessario passare dei parametri.
- **POST:** performa nella stessa maniera del metodo GET, ma i parametri della richiesta non vengono passati tramite una query string.
- **PUT:** il metodo PUT viene usato nel momento in cui si vuole modificare una risorsa
- **DELETE:** invece il metodo DELETE quando la risorsa deve essere eliminata.

# Back-end

- Per l'app E-Bike Shop sono state utilizzate richieste di tipo GET utilizzando una sintassi specifica del linguaggio Volley che consente di richiamare l'URL e passare i parametri necessari. Questi sono interpretati con il formato json.

```
<?php
$con= new mysqli("localhost","root","","AppShop");
$stmt=$con->prepare("select * from items where category=?");
$stmt->bind_param("s", $_GET["category"]);
$stmt->execute();
$rs=$stmt->get_result();
$arr=array();
while ($row=$rs->fetch_assoc())
{
    array_push($arr,$row);
}

echo json_encode($arr);
```

# ARCHITETTURA APPLICAZIONE:

- Inoltre in questo progetto è stato utilizzato un server web in locale e che pertanto non è dotato di un dominio in internet, ma necessita di un indirizzo IP per raggiungerlo.

```
var cat:String=intent.getStringExtra( name: "cat")
var url="http://192.168.43.227/AppShop/get_items.php?category="+cat
var list=ArrayList<Item>()

var rq:RequestQueue=Volley.newRequestQueue( context: this)
var jar=JSONArrayRequest(Request.Method.GET,url, jsonRequest: null,Response.Listener { response ->

    for(x in 0..response.length()-1)
        list.add(Item(response.getJSONObject(x).getInt( name: "id"),response.getJSONObject(x).getString( name: "name"),
            response.getJSONObject(x).getDouble( name: "price"),response.getJSONObject(x).getString( name: "photo"))))

    var adp=ItemAdapter( context: this,list)
    item_rv.layoutManager=LinearLayoutManager( context: this)
    item_rv.adapter=adp

},Response.ErrorListener { error ->
    Toast.makeText( context: this,error,message,Toast.LENGTH_LONG).show()
})

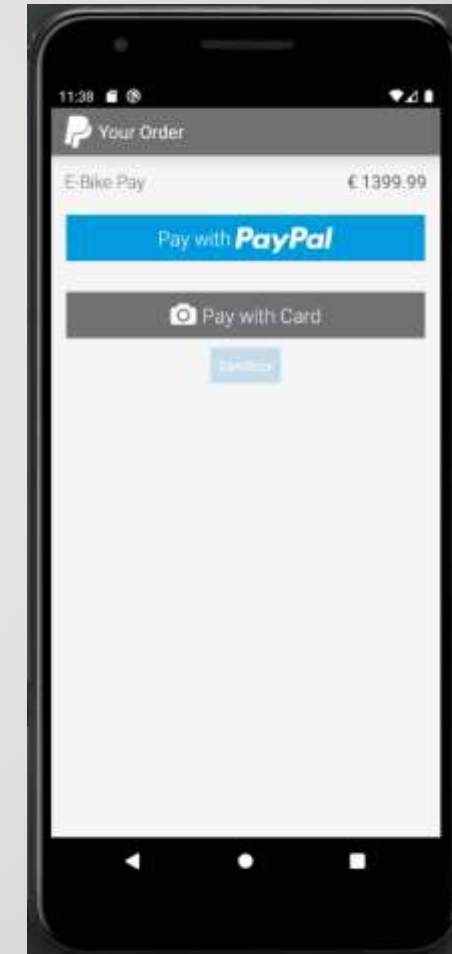
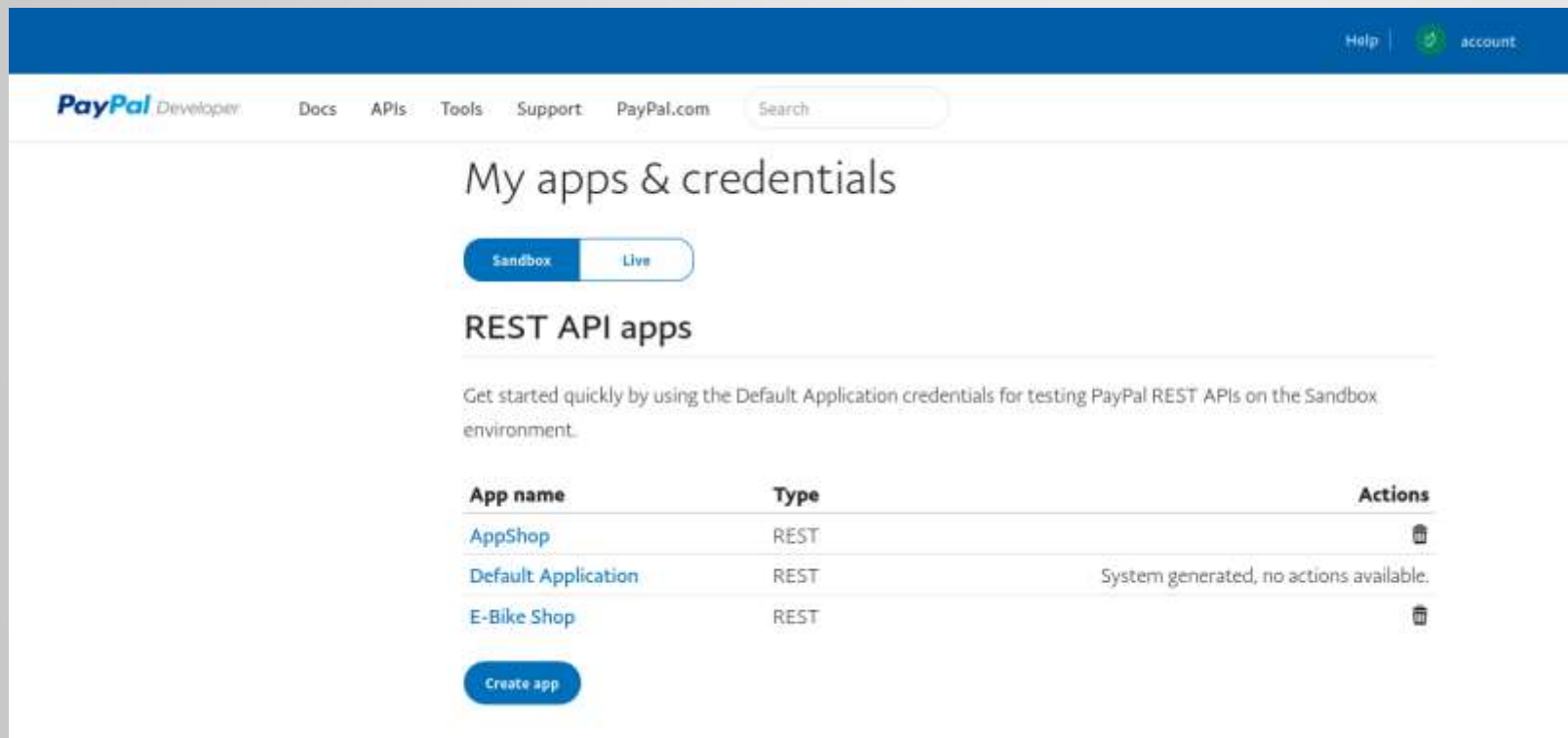
rq.add(jar)
```



# Pagamento

## STRUMENTI FINANZIARI

- Una volta selezionati i prodotti desiderati si può procedere con il pagamento del conto e il servizio finanziario Paypal offre delle API di tipo REST che sono state integrate nell'app



- Accedendo con email e password è possibile selezionare il fondo con cui pagare, saldo Paypal o carta di credito.

# Database

## BASE DI DATI COLLEGATO ALL'APP:

- Con il software Xampp è stato creato un database in locale ed è stato collegato all'Ebike-shop App.
- Il database è formato da 6 tabelle ↓

Tabella	Azione	Righe	Tipo	Codifica caratteri	Dimensione	Overhead
<input type="checkbox"/> <b>amministrators</b>	★ Mostra Struttura Cerca Inserisci Svuota Elimina	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> <b>bill</b>	★ Mostra Struttura Cerca Inserisci Svuota Elimina	92	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> <b>bill_det</b>	★ Mostra Struttura Cerca Inserisci Svuota Elimina	111	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> <b>items</b>	★ Mostra Struttura Cerca Inserisci Svuota Elimina	24	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> <b>temp_order</b>	★ Mostra Struttura Cerca Inserisci Svuota Elimina	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> <b>users</b>	★ Mostra Struttura Cerca Inserisci Svuota Elimina	12	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<b>6 tabelle</b>	<b>Totali</b>	<b>240</b>	<b>InnoDB</b>	<b>utf8mb4_general_ci</b>	<b>96.0 KiB</b>	<b>0 B</b>