ROTOINVENTORY

Author: Safeer Khan

Objective: To create an android application to give information of inventory identified using barcode/qr code and to use and store data on an online database.

Tech used: Android studio-java & xml, zxing library(for barcode), AsyncData and io operations on java, MySQL, php, Wamp server, phpMyAdmin, free hosting sites-(freesqldatabase.com),(000webhostapp.com)

Inorder to make this app, the approach was to first get individual parts of the application working and then get the app together. So the procedure was slip up into QRCODE helper, Database helper and main project.

1. Setting up QR code reader.

This application is going to use camera and flash of the device, hence we need to add permissions in the manifest.

```
<uses-permission android:name="android.permission.CAMERA" />
```

This app requires ZXING library which handles the QR scanner and consist of scanner functions, this library doesn't cone preinstalled on Android Studio so we need to download. We do so by implementing It in the build.gradle file of android Studio.

```
implementation 'me.dm7.barcodescanner:zxing:1.9'
```

Once downloaded we can use this library in our app. A new activity which is needed which implements ZXingScannerView.ResultHandler, thereby overriding all parent class functions. A zxing object is created so as to use its functions. Several functions are made but what interests us is the handle Result function. This function takes raw result as input which is an object of Result class of zxing library. This object returns the text obtained from the barcode thereby being useful for our project.

The layout xml will consist of title bar and standard layout with a toggle to switch flashlight and relative layout consisting of zxing scanner layout which comes predefined with zxing library

In order to invoke scanner activity we use Intent Integrator which thereby starts the scanner and the respective application of scan result is thereby carried out.

2. Setting up database handler.

Android natively has support for SQLite database, but it is a localised database, i.e. it cannot be used over a server. Android cannot directly access any database like MySQL, derby, etc hence we need to use an interface which can be accessed by the app and can itself fire sql query. We use Php script to achieve this. This php script will be hosted over a free hosting site thereby making it remotely accessible through internet in the app. The MYSQL database will also be hosted over a free hosting site, making a complete online database interface, which will be accessible by the app through internet.

A separate class of background worker is used to make database operations. This class is called Background worker in this app and it extends AsyncTask; which allows us to use these functions in the background operations. This class consist of a function which connects to the php script(API) and uses HttpURLConnection variable to stream input and output. When this method is invoked the data is accessed from the API and according query is fired. The result of query is then returned for the API and then to the app thus giving the status of the operation.

3. Setting up the .php script (API):

The first script will consist sign in data of the database. Php conist of special function for MySQL which connects to the server using defined credentials and then returns a Boolean. If returned true then the connections to database is established and further operations can be carried out.

This php is hosted on a free web hosting site.

```
<?php
$db_name = "rotonity";
$mysql_username=$_POST["user_name"];
$mysql_password=$_POST["password"];
$server_name="localhost";
$conn =mysqli_connect($server_name, $mysql_username, $mysql_password, $db_name);
if($conn){
echo "connection success";
}
else{
echo "un";
}
?>
```

The above script will be called before every operation with the Database.

Another script is requiorses which will act as the main Interface for the application. This script uses Conn.php to connect to the database and then fires queries intor the database. This script takes data via POST method which we will employ in our application .the output of SQL will displayed by this php which will be interprete by th app to get reqired data.

For Data flow I have made use of single string return data which is formatted with a ": " to separate entries which will be converted into meaningful data in our java code.

This script takes upto7 inputs. First one being Type of Operation to be carried out which defines which table and which query. Since our application also features a login page, it will then use an if else ladder to complete subsequent tasks. Login will take two more entries- username and password and will fire query into loginData table in the DataBase. The result will give login status and admin status. Admin status is weather the given user is an admin or not. It will return an boolen value.

The other operations for this script are further divided into if-else and Switch consisting of appropriate SQL queries. (This part seems pretty self-explanatory so no going in its details)

4. Application Structure

The application has class named Global which consists of the static variable RESULT. This will be widely used between all the classes and needs to have a one value at a time, hence static. Every result from the API will be collected by Background_worker and stored in RESULT.

The application is structures as follows:-