

Lab 4: Web API



MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

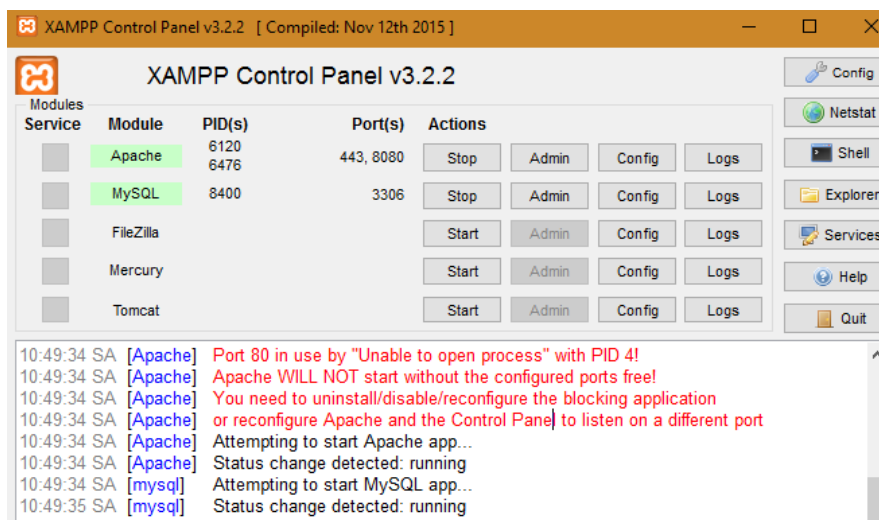
- ✓ Hiểu được Web API
- ✓ Xây dựng DATABASE trên Web Server sử dụng MySQL
- ✓ Sử dụng PHP tạo API login và register

BÀI 1 (2 ĐIỂM)

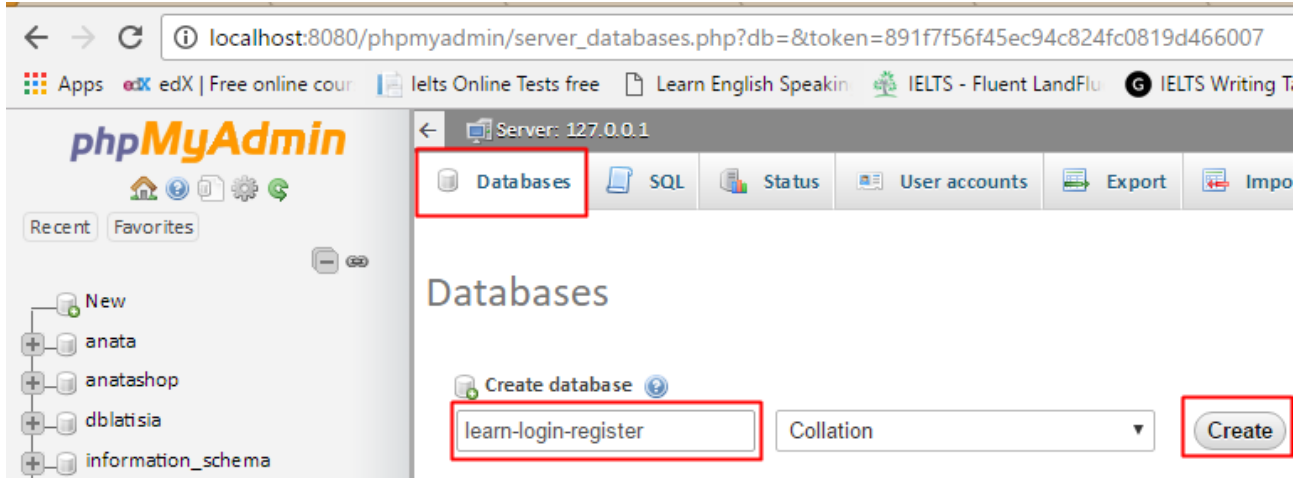
Tạo cơ sở dữ liệu và Web server cho chương trình đăng nhập, đăng ký của ứng dụng android. Bài 1 sẽ được sử dụng ở bài tiếp theo. Sử dụng mysql của xampp, và php để tương tác với database.

HƯỚNG DẪN:

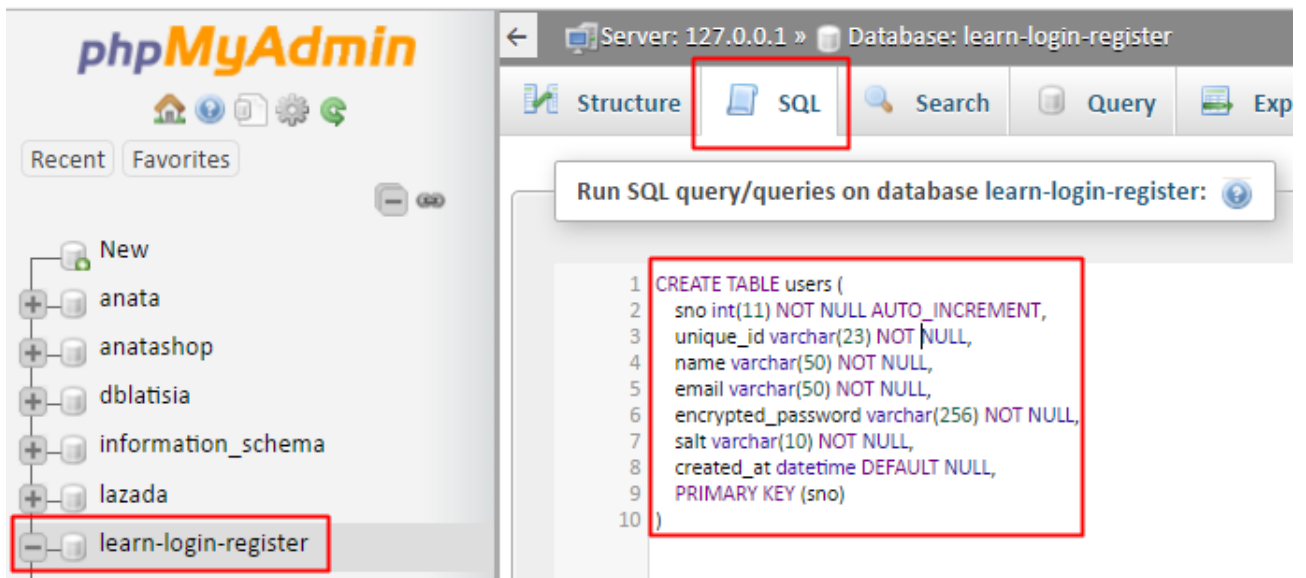
- ✓ Sinh viên vào địa chỉ <https://www.apachefriends.org/download.html> để cài đặt xampp
- ✓ Sau khi cài xong, start server như sau



- ✓ Mở trình duyệt gõ địa chỉ <http://localhost/phpmyadmin/>
- ✓ Chọn tab Databases, đặt tên và nhấn nút create để tạo database mới



- ✓ Chọn database vừa tạo, chọn tab SQL, và tiến hành tạo bảng users với các trường sau. Sno là serial number tự động tăng và là khóa chính, encrypted_password và salt là trường được sử dụng lưu mật khẩu mã hóa và chuỗi random salt



```
CREATE TABLE users (
  sno int(11) NOT NULL AUTO_INCREMENT,
  unique_id varchar(23) NOT NULL,
  name varchar(50) NOT NULL,
  email varchar(50) NOT NULL,
  encrypted_password varchar(256) NOT NULL,
  salt varchar(10) NOT NULL,
  created_at datetime DEFAULT NULL,
  PRIMARY KEY (sno)
)
```

- ✓ Mở thư mục htdocs ở thư mục xampp, tạo folder mới đặt tên learn-login-

register và tạo 3 file php như sau.

This PC > Local Disk (C:) > xampp > htdocs > learn-login-register

Name	Date modified	Type	Size
DBOperations.php	28/12/2016 11:03 SA	PHP File	0 KB
Functions.php	28/12/2016 11:03 SA	PHP File	0 KB
Index.php	28/12/2016 11:03 SA	PHP File	0 KB

- ✓ File DBOperations.php chứa các hàm kết nối với MySQL database như insertData(), checkLogin(), checkUserExist() và các câu lệnh thực thi được định nghĩa ở file này.

```
<?php
class DBOperations{

    private $host = '127.0.0.1';
    private $user = 'root';
    private $db = 'learn-login-register';
    private $pass = '';
    private $conn;

    public function __construct() {

        $this->conn = new PDO("mysql:host=".$this->host.";dbname=".$this->db,
        $this->user, $this->pass);

    }

    public function insertData($name,$email,$password){

        $unique_id = uniqid("", true);
        $hash = $this->getHash($password);
        $encrypted_password = $hash["encrypted"];
        $salt = $hash["salt"];

        $sql = 'INSERT INTO users SET unique_id =:unique_id,name =:name,
        email =:email,encrypted_password =:encrypted_password,salt
        =:salt,created_at = NOW()';
```

```

$query = $this->conn->prepare($sql);
$query->execute(array('unique_id' => $unique_id, ':name' => $name, ':email'
=> $email,
    ':encrypted_password' => $encrypted_password, ':salt' => $salt));

if ($query) {

    return true;

} else {

    return false;

}
}

public function checkLogin($email, $password) {

    $sql = 'SELECT * FROM users WHERE email = :email';
    $query = $this->conn->prepare($sql);
    $query->execute(array(':email' => $email));
    $data = $query->fetchObject();
    $salt = $data->salt;
    $db_encrypted_password = $data->encrypted_password;

    if ($this->verifyHash($password.$salt,$db_encrypted_password) ) {

        $user["name"] = $data->name;
        $user["email"] = $data->email;
        $user["unique_id"] = $data->unique_id;
        return $user;
    }
}

```

```
    } else {  
  
        return false;  
    }  
}  
  
public function checkUserExist($email){  
  
    $sql = 'SELECT COUNT(*) from users WHERE email =:email';  
    $query = $this -> conn -> prepare($sql);  
    $query -> execute(array('email' => $email));  
  
    if($query){  
  
        $row_count = $query -> fetchColumn();  
  
        if ($row_count == 0){  
  
            return false;  
  
        } else {  
  
            return true;  
  
        }  
    } else {  
  
        return false;  
    }  
}  
  
public function getHash($password) {
```

```

    $salt = sha1(rand());
    $salt = substr($salt, 0, 10);
    $encrypted = password_hash($password.$salt, PASSWORD_DEFAULT);
    $hash = array("salt" => $salt, "encrypted" => $encrypted);

    return $hash;
}
public function verifyHash($password, $hash) {

    return password_verify ($password, $hash);
}
}
?>

```

- ✓ File Functions.php chứa các hàm gọi các phương thức được định nghĩa ở DBOperations.php xử lý việc đăng nhập, đăng ký, thay đổi mật khẩu và cuối cùng trả kết quả JSON success hoặc failure.

```

<?php

require_once 'DBOperations.php';

class Functions{

private $db;

public function __construct() {

    $this -> db = new DBOperations();

}
}

```

```
public function registerUser($name, $email, $password) {

    $db = $this -> db;

    if (!empty($name) && !empty($email) && !empty($password)) {

        if ($db -> checkUserExist($email)) {

            $response["result"] = "failure";
            $response["message"] = "User Already Registered !";
            return json_encode($response);

        } else {

            $result = $db -> insertData($name, $email, $password);

            if ($result) {

                $response["result"] = "success";
                $response["message"] = "User Registered Successfully !";
                return json_encode($response);

            } else {

                $response["result"] = "failure";
                $response["message"] = "Registration Failure";
                return json_encode($response);

            }
        }
    } else {

        return $this -> getMsgParamNotEmpty();
    }
}
```

```
    }  
}  
  
public function loginUser($email, $password) {  
  
    $db = $this -> db;  
  
    if (!empty($email) && !empty($password)) {  
  
        if ($db -> checkUserExist($email)) {  
  
            $result = $db -> checkLogin($email, $password);  
  
            if(!$result) {  
  
                $response["result"] = "failure";  
                $response["message"] = "Invaild Login Credentials";  
                return json_encode($response);  
  
            } else {  
  
                $response["result"] = "success";  
                $response["message"] = "Login Sucessful";  
                $response["user"] = $result;  
                return json_encode($response);  
  
            }  
        } else {  
  
            $response["result"] = "failure";  
            $response["message"] = "Invaild Login Credentials";  
            return json_encode($response);  
        }  
    }  
}
```



```
    }  
    } else {  
  
        return $this -> getMsgParamNotEmpty();  
    }  
}  
  
public function changePassword($email, $old_password, $new_password) {  
  
    $db = $this -> db;  
  
    if (!empty($email) && !empty($old_password) && !empty($new_password)) {  
  
        if(!$db -> checkLogin($email, $old_password)){  
  
            $response["result"] = "failure";  
            $response["message"] = 'Invalid Old Password';  
            return json_encode($response);  
  
        } else {  
  
            $result = $db -> changePassword($email, $new_password);  
  
            if($result) {  
  
                $response["result"] = "success";  
                $response["message"] = "Password Changed Successfully";  
                return json_encode($response);  
  
            } else {  
  
                $response["result"] = "failure";
```

```

        $response["message"] = 'Error Updating Password';
        return json_encode($response);

    }
}
} else {

    return $this -> getMsgParamNotEmpty();
}
}

public function isEmailValid($email){

    return filter_var($email, FILTER_VALIDATE_EMAIL);
}

public function getMsgParamNotEmpty(){

    $response["result"] = "failure";
    $response["message"] = "Parameters should not be empty !";
    return json_encode($response);

}

public function getMsgInvalidParam(){

    $response["result"] = "failure";
    $response["message"] = "Invalid Parameters";
    return json_encode($response);

}

public function getMsgInvalidEmail(){

```

```

$response["result"] = "failure";
$response["message"] = "Invalid Email";
return json_encode($response);

}
}

```

- ✓ File Index.php là file chính nhận JSON data từ phía android, parse JSON data và gửi data đến file Functions.php xử lý

```

<?php

require_once 'Functions.php';

$fun = new Functions();

if ($_SERVER['REQUEST_METHOD'] == 'POST')
{
    $data = json_decode(file_get_contents("php://input"));

    if(isset($data -> operation)){

        $operation = $data -> operation;

        if(!empty($operation)){

            if($operation == 'register'){

                if(isset($data -> user ) && !empty($data -> user) && isset($data -> user ->
name)
                    && isset($data -> user -> email) && isset($data -> user -> password)){

                    $user = $data -> user;

```

```
$name = $user -> name;
$email = $user -> email;
$password = $user -> password;

if ($fun -> isValidEmail($email)) {

    echo $fun -> registerUser($name, $email, $password);

} else {

    echo $fun -> getMsgInvalidEmail();
}

} else {

    echo $fun -> getMsgInvalidParam();

}

} else if ($operation == 'login') {

    if(isset($data -> user ) && !empty($data -> user) && isset($data -> user ->
email) && isset($data -> user -> password)){

        $user = $data -> user;
        $email = $user -> email;
        $password = $user -> password;

        echo $fun -> loginUser($email, $password);

    } else {

        echo $fun -> getMsgInvalidParam();
    }
}
```

```
    }  
    } else if ($operation == 'chgPass') {  
  
        if(isset($data -> user ) && !empty($data -> user) && isset($data -> user ->  
email) && isset($data -> user -> old_password)  
        && isset($data -> user -> new_password)){  
  
            $user = $data -> user;  
            $email = $user -> email;  
            $old_password = $user -> old_password;  
            $new_password = $user -> new_password;  
  
            echo $fun -> changePassword($email, $old_password, $new_password);  
  
        } else {  
  
            echo $fun -> getMsgInvalidParam();  
  
        }  
    }  
} else {  
  
    echo $fun -> getMsgParamNotEmpty();  
  
}  
} else {  
  
    echo $fun -> getMsgInvalidParam();  
  
}  
} else if ($_SERVER['REQUEST_METHOD'] == 'GET'){
```

FPT POLYTECHNIC

```
echo "Learn Login API";

}
```

- ✓ Ví dụ về các đoạn code trên, đăng ký ở android sẽ gửi lên server JSON dạng giống như sau

```
{
  "operation": "register",
  "user": {
    "name": "Nguyen Van A",
    "email": "anv@fpt.edu.vn",
    "password": "anguyen"
  }
}
```

- ✓ Nếu đăng ký thành công, thì kết quả trả về sẽ là

```
{
  "result": "success",
  "message": "User Registered Successfully !"
}
```

- ✓ Còn đăng ký thất bại, thì kết quả trả về sẽ là

```
{
  "result": "failure",
  "message": "User Already Registered !"
}
```

- ✓ Login có dạng JSON request như sau khi gửi từ android lên server

```
{
  "operation": "login",
  "user": {
    "email": "anv@fpt.edu.vn",
    "password": "anguyen"
  }
}
```

- ✓ Nếu đăng nhập thành công thì kết quả trả về sẽ là

```
{
  "result": "success",
  "message": "Login Successful",
  "user": {
    "name": "Nguyen Van A",
    "email": "ang@fpt.edu.vn",
    "unique_id": "56fab5891ae0e6.72636328"
  }
}
```

✓ Nếu đăng nhập thất bại thì kết quả trả về sẽ là

```
{
  "result": "failure",
  "message": "Invalid Login Credentials"
}
```

✓ Changepassword có dạng JSON request như sau

```
{
  "operation": "chgPass",
  "user": {
    "email": "ang@fpt.eud.vn",
    "old_password": "anguyen",
    "new_password": "nguyea"
  }
}
```

✓ Nếu thay đổi mật khẩu thành công thì kết quả trả về sẽ là

```
{
  "result": "success",
  "message": "Password Changed Successfully"
}
```

✓ Nếu thay đổi mật khẩu thất bại thì kết quả trả về sẽ là

```
{
  "result": "failure",
  "message": "Invalid Old Password"
}
```

✓ Kết thúc bài 1

✓ Bây giờ chúng ta đã tạo thành công PHP API cho Login và Registration.

BÀI 2 (2 ĐIỂM)

Kết hợp với bài 1, viết chương trình đăng nhập, đăng ký với PHP và MySQL trên android. Sử dụng thư viện Retrofit networking của Third party để tương tác với server. Thiết kế giao diện theo chuẩn Material Design và dữ liệu tạm thời sẽ lưu ở SharedPreferences.

HƯỚNG DẪN:

- ✓ Tạo project android trong android studio, mở file build.gradle và import thư viện Design, Retrofit, GSON như sau:

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:24.2.1'
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:design:24.2.1'
    compile 'com.squareup.retrofit2:retrofit:2.0.0'
    compile 'com.squareup.retrofit2:converter-gson:2.0.0'
}
```

- ✓ Thêm INTERNET permission ở file AndroidManifest.xml

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

- ✓ Mở file layout activity_main.xml và thay đổi như sau

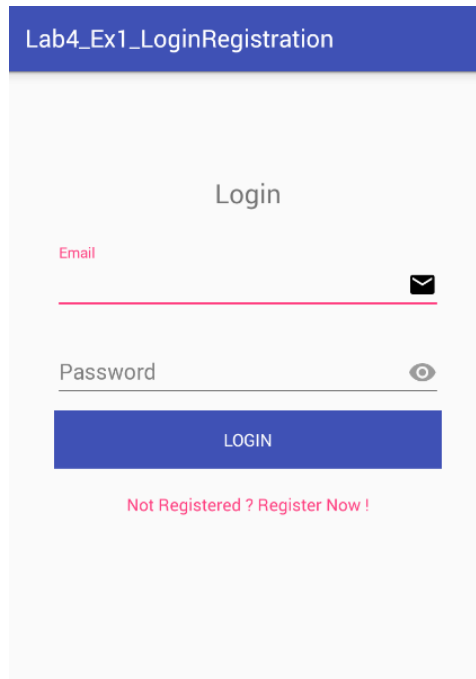
```
<?xml version="1.0" encoding="utf-8" ?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="example.com.android.lab4_ex1_loginregistration.MainActivity">

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <FrameLayout
            android:id="@+id/fragment_frame"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />

    </LinearLayout>
</android.support.design.widget.CoordinatorLayout>
```


- ✓ Tạo file layout mới đặt tên fragment_login.xml , thiết kế giao diện trang login như bên dưới



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_marginLeft="40dp"
    android:layout_marginRight="40dp"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:text="Login"
        android:textSize="22sp"
        android:textAlignment="center"
        android:layout_marginBottom="20dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <android.support.design.widget.TextInputLayout
        android:layout_marginTop="8dp"
        android:layout_marginBottom="8dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:hint="Email"
            android:id="@+id/et_email"
            android:drawableRight="@drawable/ic_email_black_24dp"
            android:inputType="textEmailAddress"
            />
    </android.support.design.widget.TextInputLayout>

    <Button
        android:text="LOGIN"
        android:layout_marginTop="10dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:text="Not Registered ? Register Now !"
        android:layout_marginTop="10dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:hint="Password"
        android:id="@+id/et_password"
        android:drawableRight="@drawable/ic_key_black"
        android:inputType="textPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</android.support.design.widget.TextInputLayout>

<android.support.v7.widget.AppCompatButton
    android:id="@+id/btn_login"
    android:text="Login"
    android:background="@color/colorPrimary"
    android:textColor="@android:color/white"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

<TextView
    android:id="@+id/tv_register"
    android:layout_marginTop="20dp"
    android:textColor="@color/colorAccent"
    android:text="Not Registered ? Register Now !"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<ProgressBar
    style="@style/Base.Widget.AppCompat.ProgressBar"
    android:id="@+id/progress"
    android:visibility="invisible"
    android:layout_marginTop="10dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:indeterminate="true" />

</LinearLayout>

```

- ✓ Tạo tiếp layout mới đặt tên fragment_register.xml, thiết kế giao diện trang Register như bên dưới

Lab4_Ex1_LoginRegistration

Register

😊

✉

👁

REGISTER

Already Registered ? Login Now !

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_marginLeft="40dp"
    android:layout_marginRight="40dp"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:text="Register"
        android:textSize="22sp"
        android:textAlignment="center"
        android:layout_marginBottom="20dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <android.support.design.widget.TextInputLayout
        android:layout_marginTop="8dp"
        android:layout_marginBottom="8dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:hint="Name"
            android:id="@+id/et_name"
            android:drawableRight="@drawable/ic_mood"
            android:inputType="textPersonName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
    </android.support.design.widget.TextInputLayout>
</LinearLayout>
```

```

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:hint="Email"
        android:id="@+id/et_email"
        android:drawableRight="@drawable/ic_email_black_24dp"
        android:inputType="textEmailAddress"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:hint="Password"
        android:id="@+id/et_password"
        android:drawableRight="@drawable/ic_key_black"
        android:inputType="textPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</android.support.design.widget.TextInputLayout>

<android.support.v7.widget.AppCompatButton
    android:id="@+id/btn_register"
    android:text="Register"
    android:background="@color/colorPrimary"
    android:textColor="@android:color/white"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

<TextView
    android:id="@+id/tv_login"
    android:layout_marginTop="20dp"
    android:textColor="@color/colorAccent"
    android:text="Already Registered ? Login Now !"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<ProgressBar
    style="@style/Base.Widget.AppCompat.ProgressBar"
    android:id="@+id/progress"
    android:visibility="invisible"
    android:layout_marginTop="10dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:indeterminate="true" />

```

```
</LinearLayout>
```

- ✓ Bây giờ ta cần tạo 3 model class để giữ Request và Response data. Sau đó data ở những class này sẽ serialized bởi Gson Converter khi ta gửi POST request tới server và ngược lại
- ✓ Tạo class model mới đặt tên User.java, lớp này sẽ chứa những trường quan trọng chúng ta sử dụng trong android app như là name, email, password, old_password, new_password, unique_id. Tạo getters và setters cho các fields.

```
package example.com.android.lab4_ex1_loginregistration;

public class User {

    private String name;
    private String email;
    private String unique_id;
    private String password;
    private String old_password;
    private String new_password;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getUnique_id() {
        return unique_id;
    }

    public void setUnique_id(String unique_id) {
        this.unique_id = unique_id;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```

FPT POLYTECHNIC

```

public String getOld_password() {
    return old_password;
}

public void setOld_password(String old_password) {
    this.old_password = old_password;
}

public String getNew_password() {
    return new_password;
}

public void setNew_password(String new_password) {
    this.new_password = new_password;
}
}

```

- ✓ Model class tiếp sẽ đặt tên là ServerRequest.java với 2 trường là operation và object của User class.

```

package example.com.android.lab4_ex1_loginregistration;

public class ServerRequest {
    private String operation;
    private User user;

    public void setOperation(String operation) {
        this.operation = operation;
    }

    public void setUser(User user) {
        this.user = user;
    }
}

```

- ✓ Và model class cuối cùng cho việc Handling Response đặt tên là ServerResponse.java với 3 trường là result, message và user. Trường thứ 3 chỉ sử dụng cho việc đăng nhập.

```

package example.com.android.lab4_ex1_loginregistration;

public class ServerResponse {
    private String result;
    private String message;
    private User user;

    public String getResult() {
        return result;
    }

    public String getMessage() {
        return message;
    }
}

```

```
public User getUser() {
    return user;
}
```

- ✓ Tạo Retrofit Interface. Ta cần tạo 1 interface và định nghĩa endpoints. Chúng ta sử dụng POST để gửi JSON data. Endpoint được định nghĩa @POST annotation và request URL của ta là **http://192.168.80.1:8080/learn-login-register/**, **http://192.168.80.1:8080/** là base url và **learn-login-regist/r.** là endpoint.
- ✓ Trong request method operation() của ta, đối tượng ServerResponse được wrapped trong Call object. Call request sẽ được chạy Asynchronos vì vậy chúng ta không cần lo lắng việc UI blocking hay AsyncTask.
- ✓ Chúng ta đã thiết lập đối tượng ServerRequest như là body cho Request. Data được lưu trữ trong ServerRequest object được serialized bởi Gson to JSON và gửi như là request lên server.
- ✓ Sau khi nhận được JSON response, data sẽ được lưu trữ ở ServerResponse object. Như đã định nghĩa.
- ✓ Tạo interface mới đặt tên RequestInterface với nội dung như sau

```
package example.com.android.lab4_ex1_loginregistration;

import retrofit2.Call;
import retrofit2.http.Body;
import retrofit2.http.POST;

public interface RequestInterface {
    @POST("learn-login-register/")
    Call<ServerResponse> operation(@Body ServerRequest request);
}
```

- ✓ Tiếp theo tạo class mới đặt tên Constants.java. lớp này được sử dụng tạo các static variables cho việc sử dụng sau này

```
package example.com.android.lab4_ex1_loginregistration;

public class Constants {
    public static final String BASE_URL = "http://192.168.80.1:8080/";
    public static final String REGISTER_OPERATION = "register";
    public static final String LOGIN_OPERATION = "login";
    public static final String CHANGE_PASSWORD_OPERATION = "chgPass";

    public static final String SUCCESS = "success";
    public static final String FAILURE = "failure";
    public static final String IS_LOGGED_IN = "isLoggedIn";

    public static final String NAME = "name";
    public static final String EMAIL = "email";
}
```

```

    public static final String UNIQUE_ID = "unique_id";

    public static final String TAG = "LearnAPI";
}

```

- ✓ Tạo mới Fragment đặt tên là RegisterFragment.java kế thừa Fragment. Trong file này sẽ chứa phương thức registerProcess(), phương thức này được gọi khi butto register nhấn. registerProcess() sẽ xử lý việc kết nối mạng và trả kết quả về.

```

package example.com.android.lab4_ex1_loginregistration;

import android.app.Fragment;
import android.app.FragmentTransaction;
import android.os.Bundle;
import android.support.design.widget.Snackbar;
import android.support.v7.widget.AppCompatButton;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class RegisterFragment extends Fragment implements
View.OnClickListener{

    private AppCompatButton btn_register;
    private EditText et_email, et_password, et_name;
    private TextView tv_login;
    private ProgressBar progress;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {

        View view =
inflater.inflate(R.layout.fragment_register, container, false);
        initView(view);
        return view;
    }

    private void initView(View view){

        btn_register =
(AppCompatButton) view.findViewById(R.id.btn_register);
        tv_login = (TextView) view.findViewById(R.id.tv_login);
        et_name = (EditText) view.findViewById(R.id.et_name);
    }
}

```


FPT POLYTECHNIC

```

        et_email = (EditText)view.findViewById(R.id.et_email);
        et_password = (EditText)view.findViewById(R.id.et_password);

        progress = (ProgressBar)view.findViewById(R.id.progress);

        btn_register.setOnClickListener(this);
        tv_login.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) {
        switch (view.getId()) {
            case R.id.tv_login:
                goToLogin();
                break;

            case R.id.btn_register:

                String name = et_name.getText().toString();
                String email = et_email.getText().toString();
                String password = et_password.getText().toString();

                if(!name.isEmpty() && !email.isEmpty()
&& !password.isEmpty()) {

                    progress.setVisibility(View.VISIBLE);
                    registerProcess(name, email, password);

                } else {

                    Snackbar.make(getView(), "Fields are empty !",
Snackbar.LENGTH_LONG).show();
                }
                break;

        }
    }

    private void registerProcess(String name, String email, String password)
    {

        Retrofit retrofit = new Retrofit.Builder()
            .baseUrl(Constants.BASE_URL)
            .addConverterFactory(GsonConverterFactory.create())
            .build();

        RequestInterface requestInterface =
retrofit.create(RequestInterface.class);

        User user = new User();
        user.setName(name);
        user.setEmail(email);
        user.setPassword(password);
        ServerRequest request = new ServerRequest();
        request.setOperation(Constants.REGISTER_OPERATION);
        request.setUser(user);
        Call<ServerResponse> response = requestInterface.operation(request);
        response.enqueue(new Callback<ServerResponse>() {

```

```

        @Override
        public void onResponse(Call<ServerResponse> call,
retrofit2.Response<ServerResponse> response) {

            ServerResponse resp = response.body();
            Snackbar.make(getView(), resp.getMessage(),
Snackbar.LENGTH_LONG).show();
            progress.setVisibility(View.INVISIBLE);
        }

        @Override
        public void onFailure(Call<ServerResponse> call, Throwable t) {

            progress.setVisibility(View.INVISIBLE);
            Log.d(Constants.TAG, "failed");
            Snackbar.make(getView(), t.getLocalizedMessage(),
Snackbar.LENGTH_LONG).show();

        }
    });

}

private void goToLogin() {

    Fragment login = new LoginFragment();
    FragmentTransaction ft = getFragmentManager().beginTransaction();
    ft.replace(R.id.fragment_frame, login);
    ft.commit();
}
}

```

- ✓ Tương tự tạo Fragment mới đặt tên LoginFragment.java. ở file này nêu đăng nhập thành công thì các giá trị sẽ được lưu ở SharedPreferences và người dùng sẽ được chuyển hướng đến Profile Fragment. Chúng ta cũng sẽ lưu 1 biến boolean trong SharedPreferences để kiểm tra người dùng đăng nhập hay không.

```

package example.com.android.lab4_ex1_loginregistration;

import android.app.Fragment;
import android.app.FragmentTransaction;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.design.widget.Snackbar;
import android.support.v7.widget.AppCompatButton;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;

```

```

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class LoginFragment extends Fragment implements View.OnClickListener
{

    private AppCompatActivity btn_login;
    private EditText edt_email, edt_password;
    private TextView tv_register;
    private ProgressBar progressBar;
    private SharedPreferences pref;

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_login, container,
false);
        initView(view);
        return view;
    }

    private void initView(View view) {
        pref = getActivity().getPreferences(0);
        btn_login = (AppCompatActivity) view.findViewById(R.id.btn_login);
        edt_email = (EditText) view.findViewById(R.id.et_email);
        edt_password = (EditText) view.findViewById(R.id.et_password);
        tv_register = (TextView) view.findViewById(R.id.tvRegister);
        progressBar = (ProgressBar) view.findViewById(R.id.progress);
        tv_register.setOnClickListener(this);
        btn_login.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) {
        switch (view.getId()) {
            case R.id.btn_login:
                String email = edt_email.getText().toString();
                String password = edt_password.getText().toString();

                if(!email.isEmpty() && !password.isEmpty()){
                    progressBar.setVisibility(View.VISIBLE);
                    loginProcess(email,password);
                }else {
                    Snackbar.make(getView(), "Fields are
empty !", Snackbar.LENGTH_LONG).show();
                }
                break;
            case R.id.tvRegister:
                Snackbar.make(getView(), "Hello", Snackbar.LENGTH_LONG).show();
                goToRegister();
                break;
        }
    }
}

```

```

    }

    private void loginProcess(String email, String password){
        Retrofit retrofit = new Retrofit.Builder()
            .baseUrl(Constants.BASE_URL)
            .addConverterFactory(GsonConverterFactory.create())
            .build();

        RequestInterface requestInterface =
retrofit.create(RequestInterface.class);

        User user = new User();
        user.setEmail(email);
        user.setPassword(password);
        ServerRequest request = new ServerRequest();
        request.setOperation(Constants.LOGIN_OPERATION);
        request.setUser(user);

        Call<ServerResponse> response = requestInterface.operation(request);

        response.enqueue(new Callback<ServerResponse>() {
            @Override
            public void onResponse(Call<ServerResponse> call,
Response<ServerResponse> response) {
                ServerResponse resp = response.body();

Snackbar.make(getView(), resp.getMessage(), Snackbar.LENGTH_LONG).show();

                if(resp.getResult().equals(Constants.SUCCESS)){
                    SharedPreferences.Editor editor = pref.edit();
                    editor.putBoolean(Constants.IS_LOGGED_IN, true);

editor.putString(Constants.EMAIL, resp.getUser().getEmail());
                    editor.putString(Constants.NAME,
resp.getUser().getName());

editor.putString(Constants.UNIQUE_ID, resp.getUser().getUnique_id());
                    editor.apply();
                    goToProfile();
                }
                progressBar.setVisibility(View.INVISIBLE);
            }

            @Override
            public void onFailure(Call<ServerResponse> call, Throwable t) {
                progressBar.setVisibility(View.INVISIBLE);
                Log.d(Constants.TAG, "failed");

Snackbar.make(getView(), t.getMessage(), Snackbar.LENGTH_LONG).show();
            }
        });
    }

    private void goToRegister() {
        Fragment register = new RegisterFragment();
        FragmentTransaction ft = getFragmentManager().beginTransaction();
        ft.replace(R.id.fragment_frame, register);
    }

```

```

        ft.commit();
    }


    private void goToProfile() {
        Fragment profile = new ProfileFragment();
        FragmentTransaction ft = getFragmentManager().beginTransaction();
        ft.replace(R.id.fragment_frame, profile);
        ft.commit();
    }
}


```


- ✓ Chúng ta đã hoàn chỉnh chức năng đăng ký, đăng nhập cho ứng dụng, ở bài tiếp theo ta sẽ làm tiếp phần hiển thị thông tin user và tạo chức năng thay đổi mật khẩu cho người dùng.

Lab4_Ex1_LoginRegistration

Register

Name
 

Email
 

Password
 


REGISTER


Already Registered ? Login Now !

User Registered Successfully !

Lab4_Ex1_LoginRegistration

Login

Email
 

Password
 

LOGIN

Not Registered ? Register Now !

BÀI 3 (2 ĐIỂM)

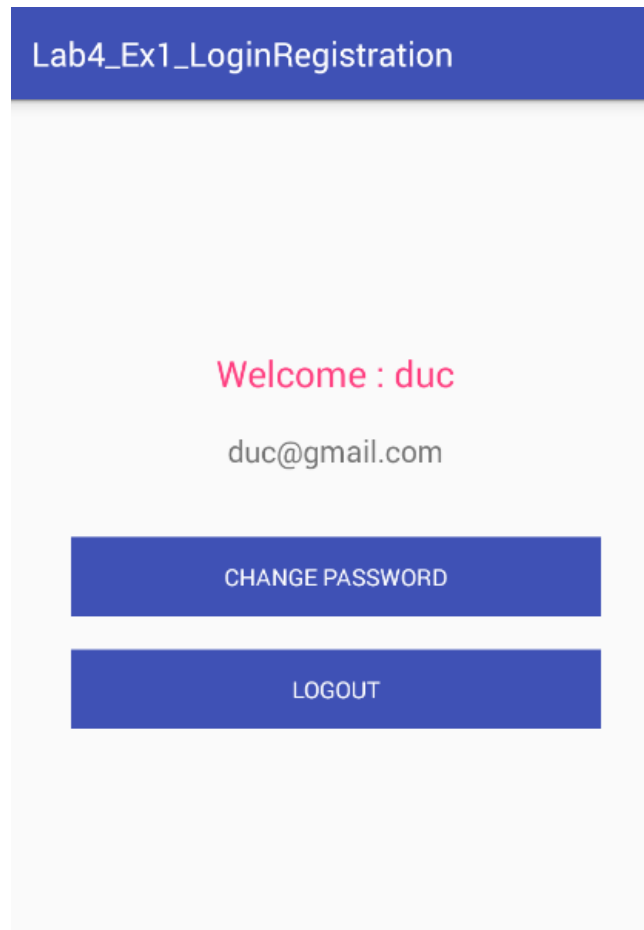
Kết hợp với bài 1 và 2, xây dựng trang Profile, tạo thêm chức changePassword với

FPT POLYTECHNIC

PHP và MySQL trên android. Sử dụng thư viện Retrofit networking của Third party để tương tác với server. Modern Material Design.

HƯỚNG DẪN:

- ✓ Tạo file layout mới đặt tên là fragment_profile.xml và thiết kế giao diện như bên dưới :



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_marginLeft="40dp"
    android:layout_marginRight="40dp"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/tv_name"
        android:layout_marginTop="20dp"
        android:textSize="22sp"
        android:textColor="@color/colorAccent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
```

```

<TextView
    android:id="@+id/tv_email"
    android:textSize="18sp"
    android:layout_marginTop="20dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

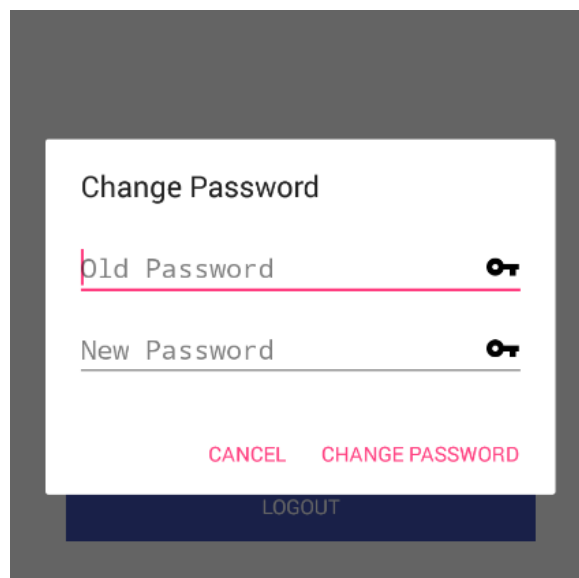
<android.support.v7.widget.AppCompatButton
    android:id="@+id/btn_chg_password"
    android:layout_marginTop="40dp"
    android:text="Change Password"
    android:background="@color/colorPrimary"
    android:textColor="@android:color/white"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<android.support.v7.widget.AppCompatButton
    android:id="@+id/btn_logout"
    android:layout_marginTop="20dp"
    android:text="Logout"
    android:background="@color/colorPrimary"
    android:textColor="@android:color/white"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

</LinearLayout>

```

- ✓ Ta cần tạo cho Dialog 1 layout, đặt tên là dialog_change_password.xml, và thiết kế giao diện như bên dưới:



```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

```

```

        android:orientation="vertical"
        android:paddingLeft="20dp"
        android:paddingRight="20dp"
        android:paddingTop="20dp"
        android:paddingBottom="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:id="@+id/et_old_password"
            android:hint="Old Password"
            android:drawableRight="@drawable/ic_key_black"
            android:inputType="textPassword"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <EditText
            android:layout_marginTop="10dp"
            android:id="@+id/et_new_password"
            android:drawableRight="@drawable/ic_key_black"
            android:hint="New Password"
            android:inputType="textPassword"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <TextView
            android:visibility="gone"
            android:id="@+id/tv_message"
            android:layout_marginTop="10dp"
            android:drawableLeft="@drawable/ic_error_black"
            android:gravity="center"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <ProgressBar
            style="@style/Base.Widget.AppCompat.ProgressBar"
            android:id="@+id/progress"
            android:visibility="gone"
            android:layout_marginTop="10dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:indeterminate="true" />

    </LinearLayout>

```

- ✓ Tiếp đến tạo class mới đặt tên ProfileFragment.java, lớp này sẽ xử lý việc hiển thị các giá trị đã được lưu ở SharedPreferences lên textview. Thêm vào đó là xử lý chức năng cho 2 button changePassword và Logout.

```

package example.com.android.lab4_ex1_loginregistration;

import android.app.Fragment;
import android.app.FragmentTransaction;
import android.content.DialogInterface;

```


FPT POLYTECHNIC

```

import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AlertDialog;
import android.support.v7.widget.AppCompatButton;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

/**
 * Created by Han on 28/12/2016.
 */

public class ProfileFragment extends Fragment implements
View.OnClickListener {

    private TextView tv_name, tv_email, tv_message;
    private SharedPreferences pref;
    private AppCompatButton btn_change_password, btn_logout;
    private EditText edt_old_password, edt_new_password;
    private AlertDialog dialog;
    private ProgressBar progressBar;

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_profile, container,
false);
        initView(view);
        return view;
    }

    @Override
    public void onViewCreated(View view, Bundle savedInstanceState) {
        pref = getActivity().getPreferences(0);
        tv_name.setText("Welcome : " + pref.getString(Constants.NAME, ""));
        tv_email.setText(pref.getString(Constants.EMAIL, ""));
    }

    private void initView(View view) {
        tv_name = (TextView) view.findViewById(R.id.tv_name);
        tv_email = (TextView) view.findViewById(R.id.tv_email);
    }

```

```

        btn_change_password = (AppCompatButton)
view.findViewById(R.id.btn_chg_password);
        btn_logout = (AppCompatButton) view.findViewById(R.id.btn_logout);
        btn_change_password.setOnClickListener(this);
        btn_logout.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) {
        switch (view.getId()) {
            case R.id.btn_chg_password:
                showDialog();
                break;
            case R.id.btn_logout:
                logout();
                break;
        }
    }

    private void showDialog() {
        AlertDialog.Builder builder = new
AlertDialog.Builder(getActivity());
        LayoutInflator inflater = getActivity().getLayoutInflater();
        View view = inflater.inflate(R.layout.dialog_change_password, null);
        edt_old_password = (EditText)
view.findViewById(R.id.et_old_password);
        edt_new_password = (EditText)
view.findViewById(R.id.et_new_password);
        tv_message = (TextView) view.findViewById(R.id.tv_message);
        progressBar = (ProgressBar) view.findViewById(R.id.progress);
        builder.setView(view);
        builder.setTitle("Change Password");
        builder.setPositiveButton("Change Password", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {

            }
        });
        builder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                dialogInterface.dismiss();
            }
        });
        dialog = builder.create();
        dialog.show();
        dialog.getButton(AlertDialog.BUTTON_POSITIVE).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String old_password = edt_old_password.getText().toString();
                String new_password = edt_new_password.getText().toString();
                if (!old_password.isEmpty() && !new_password.isEmpty()) {
                    progressBar.setVisibility(View.VISIBLE);
                    changePasswordProcess(tv_email.getText().toString(),
old password, new password);

```

```

        } else {
            tv_message.setVisibility(View.VISIBLE);
            tv_message.setText("Fields are empty");
        }
    }
});
}

private void logout() {
    SharedPreferences.Editor editor = pref.edit();
    editor.putBoolean(Constants.IS_LOGGED_IN, false);
    editor.putString(Constants.NAME, "");
    editor.putString(Constants.EMAIL, "");
    editor.putString(Constants.UNIQUE_ID, "");
    editor.apply();
    goToLogin();
}

private void goToLogin() {
    Fragment login = new LoginFragment();
    FragmentTransaction ft = getFragmentManager().beginTransaction();
    ft.replace(R.id.fragment_frame, login);
    ft.commit();
}

private void changePasswordProcess(String email, String old_password,
String new_password) {
    Gson gson = new GsonBuilder()
        .setLenient()
        .create();

    Retrofit retrofit = new Retrofit.Builder()
        .baseUrl(Constants.BASE_URL)
        .addConverterFactory(GsonConverterFactory.create())
        .build();

    RequestInterface requestInterface =
retrofit.create(RequestInterface.class);
    User user = new User();
    user.setEmail(email);
    user.setOld_password(old_password);
    user.setNew_password(new_password);
    ServerRequest request = new ServerRequest();
    request.setOperation(Constants.CHANGE_PASSWORD_OPERATION);
    request.setUser(user);

    Call<ServerResponse> response = requestInterface.operation(request);

    response.enqueue(new Callback<ServerResponse>() {
        @Override
        public void onResponse(Call<ServerResponse> call,
retrofit2.Response<ServerResponse> response) {
            ServerResponse resp = response.body();

            if (resp.getResult().equals(Constants.SUCCESS)) {

```

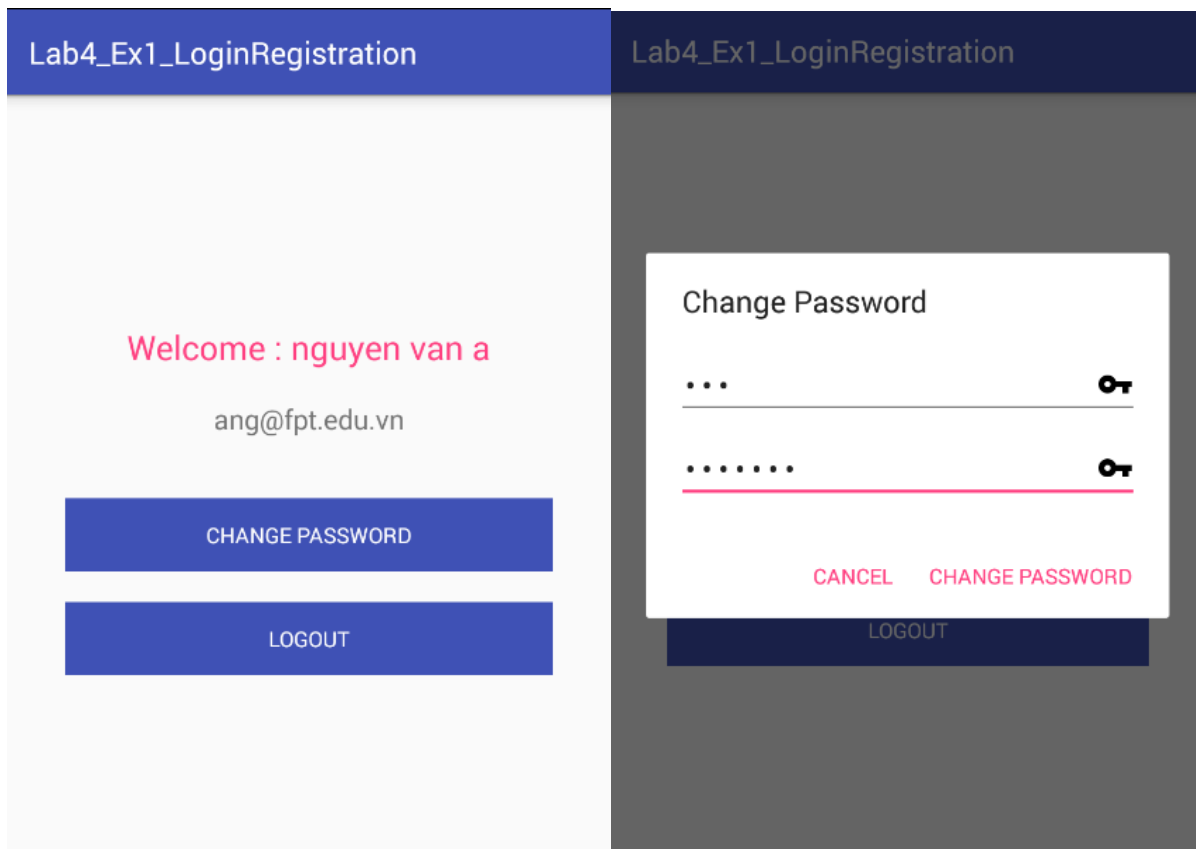
```

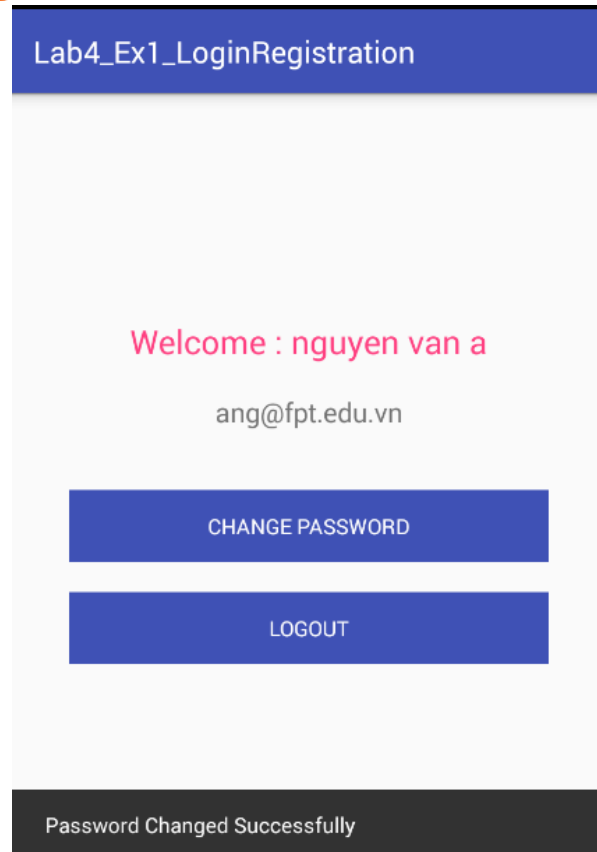
        progressBar.setVisibility(View.GONE);
        tv_message.setVisibility(View.GONE);
        dialog.dismiss();
        Snackbar.make(getView(), resp.getMessage(),
Snackbar.LENGTH_LONG).show();
    } else {
        progressBar.setVisibility(View.GONE);
        tv_message.setVisibility(View.VISIBLE);
        tv_message.setText(resp.getMessage());
    }
}

@Override
public void onFailure(Call<ServerResponse> call, Throwable t) {
    Log.d(Constants.TAG, "failed "+t.getMessage());
    progressBar.setVisibility(View.GONE);
    tv_message.setVisibility(View.VISIBLE);
    tv_message.setText(t.getMessage());
}
});
}
}

```

✓ Chạy ứng dụng và xem kết quả như bên dưới





BÀI 4 (2 ĐIỂM)

Kết hợp với bài 1 và 2, tạo chức năng resetPassword với PHP và MySQL trên android. Sử dụng thư viện Retrofit networking của Third party để tương tác với server và thư viện PHPMailer cho việc gửi reset password tới mail.

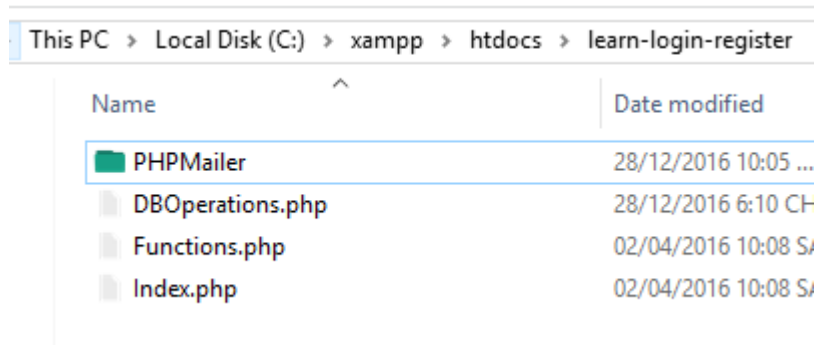
HƯỚNG DẪN:

- ✓ Ta và phpMyadmin và tạo thêm 1 bảng trong database ở bài 1 như sau :

```
CREATE TABLE password_reset_request (
sno int(11) NOT NULL AUTO_INCREMENT,
email varchar(50) NOT NULL,
encrypted_temp_password varchar(256) NOT NULL,
salt varchar(10) NOT NULL,
created_at datetime DEFAULT NULL,
PRIMARY KEY (sno)
)
```

FPT POLYTECHNIC

- ✓ Ta sẽ sử dụng thư viện PHPMailer cho việc gửi mail từ localhost. Download php từ [Github](#) repository này. Giải nén và Copy PHPMailer vừa download về vào thư mục root project: learn-login-register -> PHPMailer



- ✓ ở file DBOperations.php ta sẽ thêm 2 phương thức passwordResetRequest() và resetPassword(). passwordResetRequest() method cho bước đầu tiên của quá trình reset password. 1 random string sẽ được tạo và lưu vào database với thời gian và email. Code này sẽ được gửi đến user bằng email. Nếu dòng đã tồn tại thì sẽ được update new data.
- ✓ Phương thức resetPassword() cho bước cuối cùng. Unique code được xác nhận bởi thời gian đếm ngược. nếu thời gian nhỏ hơn 120 giây mật khẩu cũ sẽ được thay thế với mật khẩu mới.
- ✓ Mở file DBOperations.php và thêm đoạn code như bên dưới

```
public function passwordResetRequest($email){

    $random_string =
    substr(str_shuffle(str_repeat("0123456789abcdefghijklmnopqrstuvwxyz", 6)), 0,
    6);

    $hash = $this->getHash($random_string);

    $encrypted_temp_password = $hash["encrypted"];

    $salt = $hash["salt"];

    $sql = 'SELECT COUNT(*) from password_reset_request WHERE email
```

```
=:email';

$query = $this->conn->prepare($sql);

$query->execute(array('email' => $email));

if($query){

    $row_count = $query->fetchColumn();

    if ($row_count == 0){

        $insert_sql = 'INSERT INTO password_reset_request SET email
=:email,encrypted_temp_password =:encrypted_temp_password,

        salt =:salt,created_at = :created_at';

        $insert_query = $this->conn->prepare($insert_sql);

        $insert_query->execute(array(':email' => $email,
':encrypted_temp_password' => $encrypted_temp_password,

        ':salt' => $salt, ':created_at' => date("Y-m-d H:i:s")));

        if ($insert_query) {

            $user["email"] = $email;

            $user["temp_password"] = $random_string;
```

```
        return $user;

    } else {

        return false;

    }

} else {

    $update_sql = 'UPDATE password_reset_request SET email
=:email,encrypted_temp_password =:encrypted_temp_password,
    salt =:salt,created_at = :created_at';

    $update_query = $this -> conn -> prepare($update_sql);

    $update_query -> execute(array(':email' => $email,
':encrypted_temp_password' => $encrypted_temp_password,
    ':salt' => $salt, ':created_at' => date("Y-m-d H:i:s")));

    if ($update_query) {

        $user["email"] = $email;

        $user["temp_password"] = $random_string;

        return $user;
    }
}
```



```
        } else {  
  
            return false;  
  
        }  
    }  
} else {  
  
    return false;  
  
}  
}  
  
public function resetPassword($email,$code,$password){  
  
    $sql = 'SELECT * FROM password_reset_request WHERE email = :email';  
    $query = $this -> conn -> prepare($sql);  
    $query -> execute(array(':email' => $email));  
    $data = $query -> fetchObject();  
    $salt = $data -> salt;  
    $db_encrypted_temp_password = $data -> encrypted_temp_password;  
  
    if ($this -> verifyHash($code.$salt,$db_encrypted_temp_password) ) {
```

```
$old = new DateTime($data -> created_at);  
$now = new DateTime(date("Y-m-d H:i:s"));  
$diff = $now->getTimestamp() - $old->getTimestamp();  
  
if($diff < 120) {  
  
    return $this -> changePassword($email, $password);  
  
} else {  
  
    false;  
}  
} else {  
  
    return false;  
}  
}
```

- ✓ Tiếp đến mở file Functions.php và thêm đoạn code import thư viện sau và 2 phương thức resetPasswordRequest() và resetPassword().

```
<?php

require_once 'DBOperations.php';
require 'PHPMailer/PHPMailerAutoload.php';
class Functions{

    private $db;
    private $mail;

    public function __construct() {

        $this -> db = new DBOperations();
        $this -> mail = new PHPMailer;

    }
}
```

```
public function resetPasswordRequest($email){
```

```
    $db = $this -> db;
```

```
    if ($db -> checkUserExist($email)) {
```

```
        $result = $db -> passwordResetRequest($email);
```

```
        if(!$result){
```

```
            $response["result"] = "failure";
```

```
            $response["message"] = "Reset Password Failure";
```

```
            return json_encode($response);
```

```
    } else {

        $mail_result = $this ->
        sendEmail($result["email"],$result["temp_password"]);

        if($mail_result){

            $response["result"] = "success";

            $response["message"] = "Check your mail for reset password code.";

            return json_encode($response);

        } else {

            $response["result"] = "failure";

            $response["message"] = "Reset Password Failure";

            return json_encode($response);

        }

    }

} else {

    $response["result"] = "failure";

    $response["message"] = "Email does not exist";
```

```
        return json_encode($response);

    }

}

public function resetPassword($email,$code,$password){

    $db = $this -> db;

    if ($db -> checkUserExist($email)) {

        $result = $db -> resetPassword($email,$code,$password);

        if(!$result){

            $response["result"] = "failure";
            $response["message"] = "Reset Password Failure";
            return json_encode($response);

        } else {

            $response["result"] = "success";
```

```
$response["message"] = "Password Changed Successfully";  
  
return json_encode($response);  
  
}  
} else {  
  
    $response["result"] = "failure";  
    $response["message"] = "Email does not exist";  
    return json_encode($response);  
  
}  
}
```

- ✓ Thêm tiếp 2 phương thức là `sendEmail()` và `sendPHPMail()`. 2 PT này được sử dụng cho việc gửi email.

```
public function sendEmail($email,$temp_password){  
  
    $mail = $this->mail;  
    $mail->isSMTP();  
    $mail->Host = 'smtp.gmail.com';  
    $mail->SMTPAuth = true;  
    $mail->Username = 'your.email@gmail.com';
```

```
$mail->Password = 'password';

$mail->SMTPSecure = 'ssl';

$mail->Port = 465;


$mail->From = 'your.email@gmail.com';

$mail->FromName = 'Your Name';

$mail->addAddress($email, 'Your Name');


$mail->addReplyTo('your.email@gmail.com', 'Your Name');


$mail->WordWrap = 50;

$mail->isHTML(true);


$mail->Subject = 'Password Reset Request';

$mail->Body = 'Hi,<br><br> Your password reset code is
<b>'.$temp_password.'</b> . This code expires in 120 seconds. Enter this code
within 120 seconds to reset your password.<br><br>Thanks,<br>Learn2Crack.';


if(!$mail->send()) {

    return $mail->ErrorInfo;

} else {
```

```

        return true;

    }

}

public function sendPHPMail($email,$temp_password){

    $subject = 'Password Reset Request';

    $message = 'Hi,nn Your password reset code is '.$temp_password.' . This
code expires in 120 seconds. Enter this code within 120 seconds to reset your
password.nnThanks,nLearn2Crack.';

    $from = "your.email@example.com";

    $headers = "From:" . $from;

    return mail($email,$subject,$message,$headers);

}

```

- ✓ Mở file index.php thêm 2 điều kiện resPassReq và resPass để handle quá trình reset password.

```

Else if ($operation == 'resPassReq') {

if(isset($data -> user) && !empty($data -> user) &&isset($data -> user ->
email)){

```



```
$user = $data -> user;
$email = $user -> email;

echo $fun -> resetPasswordRequest($email);

} else {

echo $fun -> getMsgInvalidParam();

}

}else if ($operation == 'resPass') {

if(isset($data -> user) && !empty($data -> user) && isset($data -> user -> email)
&& isset($data -> user -> password)
&& isset($data -> user -> code)){

$user = $data -> user;
$email = $user -> email;
$code = $user -> code;
$password = $user -> password;
```

```
echo $fun -> resetPassword($email,$code,$password);

} else {

echo $fun -> getMsgInvalidParam();

}

}
```

✓ Request từ android lên server tương tự như sau

```
{
  "operation": "resPassReq",
  "user": {
    "email": "raj.amalw@learn2crack.com"
  }
}
```

✓ Nếu request success thì response như sau

```
{
  "result": "success",
  "message": "Check your mail for reset password code."
}
```

✓ Và request kết thúc quá trình reset password sẽ giống như sau

```
{
  "operation": "resPass",
  "user": {
    "email": "raj.amalw@learn2crack.com",
    "code": "bcfqa3",
    "password": "rajamalw"
  }
}
```

```
}
}
```

✓ Nếu request success thì kết quả như sau

```
{
  "result": "success",
  "message": "Password Changed Successfully"
}
```

✓ Tạo file layout mới đặt tên fragment_password_reset.xml. Thêm 3 EditText để nhập email, code and password. Và textview hiển thị countdown timer.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_marginLeft="40dp"
    android:layout_marginRight="40dp"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:text="Learn2Crack"
        android:textSize="22sp"
        android:textAlignment="center"
        android:layout_marginBottom="20dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/timer"
        android:textSize="22sp"
        android:textStyle="bold"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <android.support.design.widget.TextInputLayout
        android:layout_marginTop="8dp"
        android:layout_marginBottom="8dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:hint="Email"
            android:id="@+id/et_email"
            android:drawableRight="@drawable/ic_email_black_24dp"
            android:inputType="textEmailAddress"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
    </android.support.design.widget.TextInputLayout>
</LinearLayout>
```

```

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:hint="Code"
        android:id="@+id/et_code"
        android:drawableRight="@drawable/ic_key_black"
        android:inputType="textPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:hint="Password"
        android:id="@+id/et_password"
        android:drawableRight="@drawable/ic_key_black"
        android:inputType="textPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</android.support.design.widget.TextInputLayout>

<android.support.v7.widget.AppCompatButton
    android:id="@+id/btn_reset"
    android:text="Forgot Password"
    android:background="@color/colorPrimary"
    android:textColor="@android:color/white"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

<ProgressBar
    style="@style/Base.Widget.AppCompat.ProgressBar"
    android:id="@+id/progress"
    android:visibility="invisible"
    android:layout_marginTop="10dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:indeterminate="true" />

</LinearLayout>

```

FPT POLYTECHNIC

- ✓ Mở file Constants.java thêm 2 constant sau :

```
public static final String RESET_PASSWORD_INITIATE = "resPassReq";
public static final String RESET_PASSWORD_FINISH = "resPass";
```

- ✓ Mở tiếp file User.java và thêm trường code .

```
private String code;
```

- ✓ Tạo Fragment mới đặt tên ResetpasswordFragment.java, khởi tạo, ánh xạ các view và bắt sự kiện click cho button

```
package example.com.android.lab4_ex1_loginregistration;

import android.app.Fragment;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.support.design.widget.Snackbar;
import android.support.v7.widget.AppCompatButton;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;

public class ResetPasswordFragment extends Fragment implements
View.OnClickListener{

    private AppCompatButton btn_reset;
    private EditText et_email,et_code,et_password;
    private TextView tv_timer;
    private ProgressBar progress;
    private boolean isResetInitiated = false;
    private String email;
    private CountDownTimer countDownTimer;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {

        View view =
inflater.inflate(R.layout.fragment_password_reset,container,false);
        initViews(view);
        return view;
    }

    private void initViews(View view){

        btn_reset = (AppCompatButton)view.findViewById(R.id.btn_reset);
```

```

        tv_timer = (TextView) view.findViewById(R.id.timer);
        et_code = (EditText) view.findViewById(R.id.et_code);
        et_email = (EditText) view.findViewById(R.id.et_email);
        et_password = (EditText) view.findViewById(R.id.et_password);
        et_password.setVisibility(View.GONE);
        et_code.setVisibility(View.GONE);
        tv_timer.setVisibility(View.GONE);
        btn_reset.setOnClickListener(this);
        progress = (ProgressBar) view.findViewById(R.id.progress);

    }

    @Override
    public void onClick(View view) {
        switch (view.getId()) {

        }
    }
}

```

- ✓ Trong FragmentResetPassword Thêm 2 phương
 initiateResetPasswordProcess() và finishResetPasswordProcess(). Khi success
 response được nhận thì trong initiateResetPasswordProcess() count down
 timer bắt đầu, hidden code và new password field được hiển thị. Sau khi
 nhận code và new password thì finishResetPasswordProcess() được gọi. nếu
 thời gian hết thì tự động quay trở lại trang login.

```

private void initiateResetPasswordProcess(String email) {
    Retrofit retrofit = new Retrofit.Builder()
        .baseUrl(Constants.BASE_URL)
        .addConverterFactory(GsonConverterFactory.create())
        .build();

    RequestInterface requestInterface =
        retrofit.create(RequestInterface.class);
    User user = new User();
    user.setEmail(email);
    ServerRequest request = new ServerRequest();
    request.setOperation(Constants.RESET_PASSWORD_INITIATE);
    request.setUser(user);
    Call<ServerResponse> response = requestInterface.operation(request);

    response.enqueue(new Callback<ServerResponse>() {
        @Override
        public void onResponse(Call<ServerResponse> call,
            Response<ServerResponse> response) {
            ServerResponse resp = response.body();
            Snackbar.make(getView(), resp.getMessage(),
                Snackbar.LENGTH_LONG).show();

            if (resp.getResult().equals(Constants.SUCCESS)) {
                Snackbar.make(getView(), resp.getMessage(),

```

```

Snackbar.LENGTH_LONG).show();
        et_email.setVisibility(View.GONE);
        et_code.setVisibility(View.VISIBLE);
        et_password.setVisibility(View.VISIBLE);
        tv_timer.setVisibility(View.VISIBLE);
        btn_reset.setText("Change Password");
        isResetInitiated = true;
        startCountdownTimer();

    } else {

        Snackbar.make(getView(), resp.getMessage(),
Snackbar.LENGTH_LONG).show();

    }

    progress.setVisibility(View.INVISIBLE);
}

@Override
public void onFailure(Call<ServerResponse> call, Throwable t) {
    progress.setVisibility(View.INVISIBLE);
    Log.d(Constants.TAG, "failed");
    Snackbar.make(getView(), t.getLocalizedMessage(),
Snackbar.LENGTH_LONG).show();
}

});
}

private void finishResetPasswordProcess(String email, String code, String
password) {
    Retrofit retrofit = new Retrofit.Builder()
        .baseUrl(Constants.BASE_URL)
        .addConverterFactory(GsonConverterFactory.create())
        .build();

    RequestInterface requestInterface =
retrofit.create(RequestInterface.class);

    User user = new User();
    user.setEmail(email);
    user.setCode(code);
    user.setPassword(password);
    ServerRequest request = new ServerRequest();
    request.setOperation(Constants.RESET_PASSWORD_FINISH);
    request.setUser(user);
    Call<ServerResponse> response = requestInterface.operation(request);

    response.enqueue(new Callback<ServerResponse>() {
        @Override
        public void onResponse(Call<ServerResponse> call,
Response<ServerResponse> response) {
            ServerResponse resp = response.body();
            Snackbar.make(getView(), resp.getMessage(),
Snackbar.LENGTH_LONG).show();

            if (resp.getResult().equals(Constants.SUCCESS)) {

                Snackbar.make(getView(), resp.getMessage(),

```

```

Snackbar.LENGTH_LONG).show();
        countdownTimer.cancel();
        isResetInitiated = false;
        goToLogin();

    } else {

        Snackbar.make(getView(), resp.getMessage(),
Snackbar.LENGTH_LONG).show();

    }

    progress.setVisibility(View.INVISIBLE);
}

@Override
public void onFailure(Call<ServerResponse> call, Throwable t) {
    progress.setVisibility(View.INVISIBLE);
    Log.d(Constants.TAG, "failed");
    Snackbar.make(getView(), t.getLocalizedMessage(),
Snackbar.LENGTH_LONG).show();
}

});
}

```

✓ Thêm tiếp 2 phương thức startCountdownTimer() và goToLogin()

```

private void startCountdownTimer(){
    countdownTimer = new CountDownTimer(120000, 1000) {

        public void onTick(long millisUntilFinished) {
            tv_timer.setText("Time remaining : " + millisUntilFinished /
1000);
        }

        public void onFinish() {
            Snackbar.make(getView(), "Time Out ! Request again to reset
password.", Snackbar.LENGTH_LONG).show();
            goToLogin();
        }
    }.start();
}

private void goToLogin(){

    Fragment login = new LoginFragment();
    FragmentTransaction ft = getFragmentManager().beginTransaction();
    ft.replace(R.id.fragment_frame, login);
    ft.commit();
}

```


FPT POLYTECHNIC

✓ Xử lý sự kiện click cho button

```
@Override
public void onClick(View view) {
    switch (view.getId()) {

        case R.id.btn_reset:

            if (!isResetInitiated) {

                email = et_email.getText().toString();
                if (!email.isEmpty()) {
                    progress.setVisibility(View.VISIBLE);
                    initiateResetPasswordProcess(email);
                } else {

                    Snackbar.make(getView(), "Fields are empty !",
Snackbar.LENGTH_LONG).show();
                }
            } else {

                String code = et_code.getText().toString();
                String password = et_password.getText().toString();

                if (!code.isEmpty() && !password.isEmpty()) {

                    finishResetPasswordProcess(email, code, password);
                } else {

                    Snackbar.make(getView(), "Fields are empty !",
Snackbar.LENGTH_LONG).show();
                }

            }

            break;
    }
}
```

✓ Cuối cùng qua LoginFragment thêm phương thức để bắt sự kiện chuyển sang trang FragmentResetPassword

```
private void goToResetPassword() {

    Fragment reset = new ResetPasswordFragment();
    FragmentTransaction ft = getFragmentManager().beginTransaction();
    ft.replace(R.id.fragment_frame, reset);
    ft.commit();
}
```

✓ Chạy ứng dụng và xem kết quả như bên dưới

Lab4_Ex1_LoginRegistration

Reset Password

Email

hanthienduc.96@gmail.com

✉

FORGOT PASSWORD

Lab4_Ex1_LoginRegistration

Reset Password

Time remaining : 116

Code

✉

Password

✉

CHANGE PASSWORD

Lab4_Ex1_LoginRegistration

Login

Email

✉

Password

✉

LOGIN

Forgot Password

Not Registered ? Register Now !

FPT POLYTECHNIC

BÀI 5 (2 ĐIỂM):

Giảng viên cho thêm