Ministerul Educaţiei al Republicii Moldova Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatica si Microelectronica Departamentul Ingineria Software si Automatica

**RAPORT**

Lucrarea de laborator nr.5-6 *Programarea aplicatiilor mobile Tema: Telemedicine-SPRINT2*

A efectuat:

A verificat:

asist. univ. C. Rusu

Chisinau 2017

# Scopul lucrarii

## Pentru designul ofert in laboratorul nr. 5 să se implementeze partea funcțională a proiectului conform serviciului API descris mai jos.

Metodele oferite de Servicu:

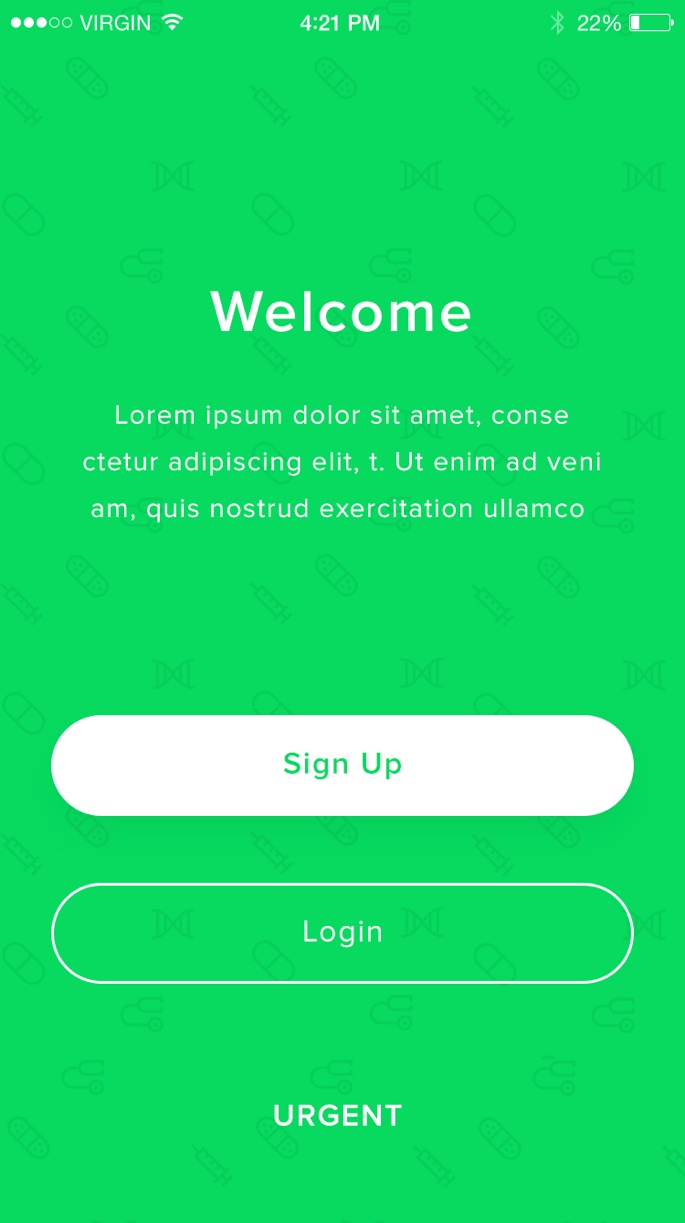
Auth (autentificare) Reg (inregistrare)

UserProfie (extragerea avansată despre utilizatorul curent autentificat) UpdateUserProfile (NotImplemented) - nu cred că o voi implementa GetDocList (vizualizarea listei de medici activi in sistem)

GetDoc (vizualizarea unui anumit medici activ in sistem) UserRequestConsultation (Adaugarea unei programari la medic)

**Desfasurarea lucrarii de laborator**

Cand începe aplicația, apare SPLASH SCREEN, adică ecran de boot, este folosit un cronometru și apoi va aparea WELCOME SCREEN



// Splash screen timer

private static int SPLASH\_TIME\_OUT = 3000; private String prevTitle;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_splash\_screen);

new Handler().postDelayed(new Runnable() { @Override

public void run() {

// This method will be executed once the timer is over

// Start your app main activity

Intent i = new Intent(SplashScreenActivity.this, WelcomeScreenActivity.class);

startActivity(i);

// close this activity finish();

}

}, SPLASH\_TIME\_OUT);

}

## În WELCOME SCREEN, putem merge la pagina de înregistrare cu butoanele Sing Up sau pe pagina de autentificare utilizând butonul Login. În acest caz, Intent este folosit, care este un obiect de schimb de mesaje, cu care pot fi folosite pentru a solicita acțiunea componentei altei cereri. Componenta Activitate este un singur ecran în aplicație. Pentru a începe o nouă instanță a componentei Activitate, trebuie să treci un obiect Intent thestartActivity () metoda. Obiectul Intent descrie operația pe care doriți să o porniți și De asemenea, conține toate celelalte date necesare.

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.welcome\_screen);

findViewById(R.id.btn\_signup).setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) { openSignupActivity();

}

});

findViewById(R.id.btn\_login).setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) { openLoginActivity();

}

});

}

public void openLoginActivity(){

Intent intent = new Intent(this, LoginActivity.class); startActivity(intent);

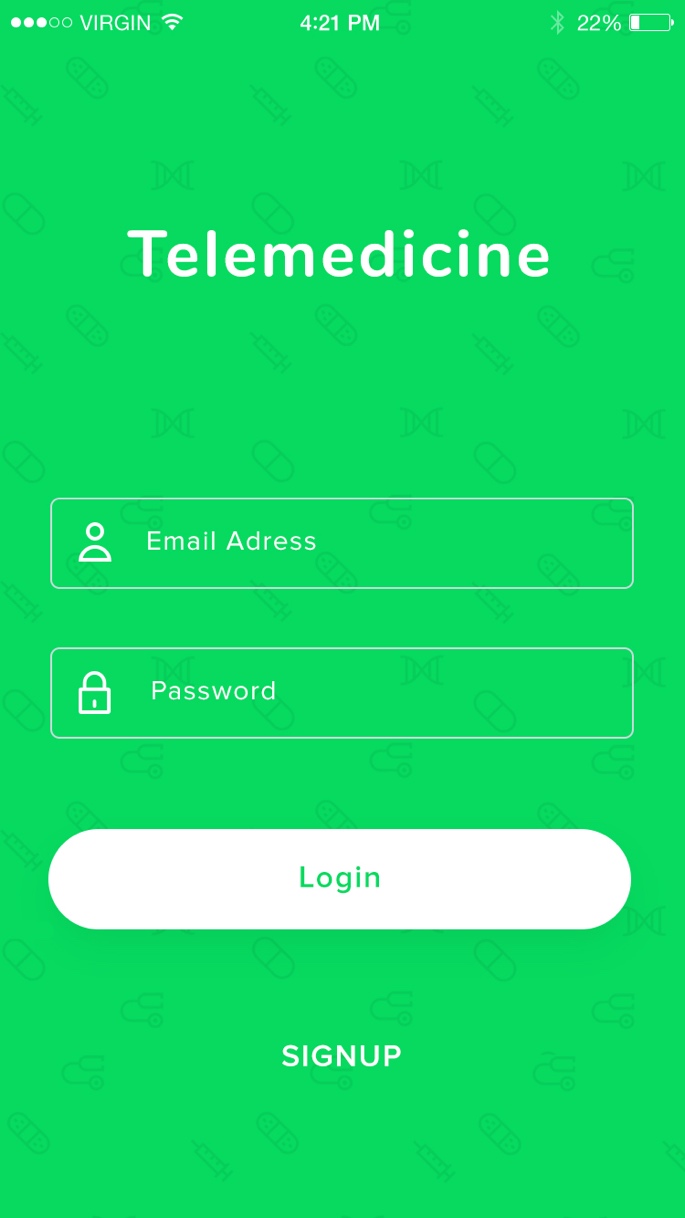
}

public void openSignupActivity(){

Intent intent = new Intent(this, SignupActivity.class); startActivity(intent);

}

## În LOGIN SCREEN, o solicitare de conectare este trimisă pe server. Dacă reușim token primit este salvat și ecranul HOME este deschis, altfel vine o notificare.



public static void sendAndHandleLoginRequest(final Context context, String validatedEmail, String validatedPassword){

RemoteApi.getApi().userAuth(new UserAuthBody(validatedEmail, validatedPassword)

).enqueue(new Callback<UserAuthResult>() { @Override

public void onResponse(Call<UserAuthResult> call, Response<UserAuthResult>

responseng(), response); if (!response.isSuccessful()){

Toast.makeText(context, "Unable to send auth request", Toast.LENGTH\_LONG).show();

}

else if (!response.body().getStatus().equalsIgnoreCase("SUCCESS")){ Toast.makeText(context, "Invalid username/password",

Toast.LENGTH\_LONG).show();

}

else if (response.body().getStatus().equalsIgnoreCase("SUCCESS")){ RemoteApi.setTOKEN(response.body().getMessage()); Log.d(context.toString(), "TOKEN - " + RemoteApi.getTOKEN()); LoginActivity.openHomeActivity(context);

}

}

@Override

public void onFailure(Call<UserAuthResult> call, Throwable t) { RemoteApiUtils.logResponse(context.getClass().toString(), t); Toast.makeText(context, "Unable to send auth request. IOException",

Toast.LENGTH\_LONG).show();

}

});

}

## În ecranul SIGNUP SCREEN, datele utilizatorului sunt introduse. În cazul în care utilizatorul nu a introdus niciun fel de date, apoi apare un avertisment că ar trebui să va asigurați-vă că ați introdus.

private boolean validateFieldsAndSetBDate(){ boolean result = true;

SimpleDateFormat formatter = new SimpleDateFormat("dd/mm/yy");

if (address.getText() == null || address.getText().toString().isEmpty()){ result = false;

address.setError("This field is required");

}

if (phone.getText() == null || phone.getText().toString().isEmpty()){ result = false;

phone.setError("This field is required");

}

if (name.getText() == null || name.getText().toString().isEmpty()){ result = false;

name.setError("This field is required");

}

if (!LoginActivity.validateEmail(email.getText().toString())){ result = false;

email.setError("Wrong email format");

}

try {

dt\_bday = formatter.parse(bday.getText().toString());

} catch (ParseException | NullPointerException e) { result = false;

bday.setError("Date should be in dd/mm/yy format");

}

return result;

}

## Ecranul COMPLET SIGNUP trimite o solicitare de înregistrare la server. Dacă este reușit, codul va fi

201. Noul utilizator va fi inregistrat și va fi deschis HOME SCREEN . În caz contrar, este emisa o notificare.

private void registerAndLogin(){

RemoteApi.getApi().userRegistration(regBody).enqueue(new Callback<Class<Void>>() { @Override

public void onResponse(Call<Class<Void>> call, Response<Class<Void>> response) { if (response.code() != 201){

RemoteApiUtils.logResponse(activity.toString(), response); Toast.makeText(activity, "Registration failed. Please try again",

Toast.LENGTH\_LONG).show();

}

else {

LoginActivity.sendAndHandleLoginRequest(activity, regBody.getEmail(), regBody.getPassword());

}

}

@Override

public void onFailure(Call<Class<Void>> call, Throwable t) { RemoteApiUtils.logResponse(activity.toString(), t); Toast.makeText(activity, "Registration failed. Please try again",

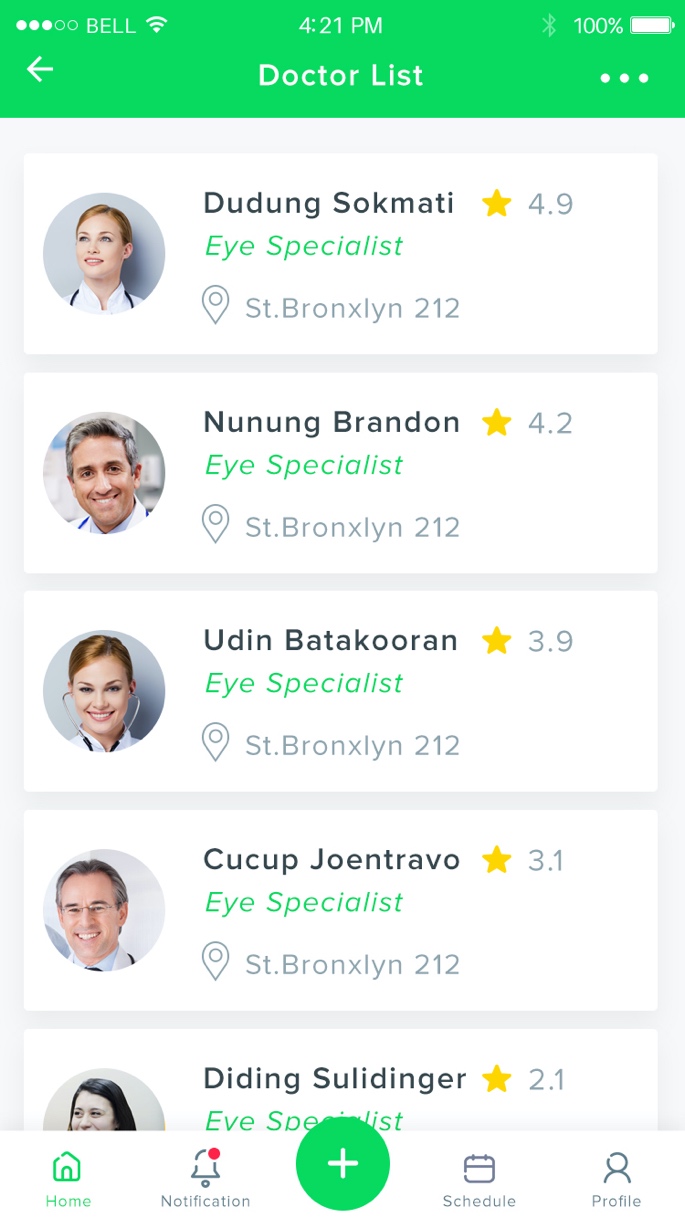
Toast.LENGTH\_LONG).show();

}

});

}

## DoctorListActivity utilizează DoctorListAdapter. Adaptorii simplifică legarea date cu controlul. Există un set de obiecte și există o componentă Vizualizare. Denumirea adaptorul este de a oferi vederi pentru container. Adaptor ia datele și metadatele unui anumit container și construiește vizualizarea fiecărui copil. De exemplu, creăm elemente de listă (un șir de șiruri de caractere) și le transmitem în lista ListView.



public void onBindViewHolder(DoctorViewHolder docviewholder, int position) { mCursor.moveToPosition(position);

DoctorProfile dp = new DoctorProfile(null,null, null,null,null,null,null);

// icon

String doctorPhoto = dp.getPhoto();

//docviewholder.iconView.setText(doctorPhoto);

//forecastAdapterViewHolder.iconView.setImageResource(weatherImageId);

// first & last name

String doctorName = dp.getFullName(); docviewholder.nameView.setText(doctorName);

// speciality

String speciality = dp.getSpecs(); docviewholder.descriptionView.setText(speciality);

//stars

Double stars = dp.getStars(); String st = Double.toString(stars);

docviewholder.adressView.setText(st);

//adress

String adress = dp.getAddress(); docviewholder.adressView.setText(adress);

}

@Override

public int getItemCount() {

if (null == mCursor) return 0; return mCursor.getCount();

}

# Concluzii

## În cadrul acestei lucrări de laborator, partea logică a aplicației Telemedicine utilizând următorul punct de vedere din lucrările de laborator nr. 5: SPLASH SCREEN, WELCOME SCREEN, SCREEN LOGIN / SIGNUP, SCREEN HOME, APPROVED REQUEST SCREEN, DOCTOR LIST, DOCTOR CONTACTS.

Au fost implementate metode precum înregistrarea, autentificarea, utilizarea, obținerea unei liste de doctori și detalii despre acestea, adăugând consultatiile medicului. În rezultat a fost o interfață de utilizator cu drepturi depline și o parte logică a aplicației Android numită Telemedicine.

**Bibliografie**

1. Tutorial [Resursă electronică]. Regim de acces: <https://www.youtube.com/>
2. Material ajutator [Resursă electronică]. Regim de acces: <https://stackoverflow.com/>

**SplashScreenActivity**

package com.example.vicup.telemedicine;

# Anexa Codul sursa

import android.content.Intent;

import android.support.v7.app.AppCompatActivity; import android.os.Bundle;

import android.os.Handler;

public class SplashScreenActivity extends AppCompatActivity {

// Splash screen timer

private static int SPLASH\_TIME\_OUT = 3000; private String prevTitle;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_splash\_screen);

new Handler().postDelayed(new Runnable() {

/\*

* Showing splash screen with a timer. This will be useful when you
* want to show case your app logo / company

\*/

@Override

public void run() {

// This method will be executed once the timer is over

// Start your app main activity

Intent i = new Intent(SplashScreenActivity.this, WelcomeScreenActivity.class); startActivity(i);

// close this activity finish();

}

}, SPLASH\_TIME\_OUT);

}

@Override

protected void onStart() { super.onStart();

prevTitle = getTitle().toString(); setTitle("");

}

@Override

protected void onStop() { super.onStop(); setTitle(prevTitle);

}

}

# WelcomeScreenActivity

package com.example.vicup.telemedicine;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity; import android.os.Bundle;

import android.view.View; import android.widget.Button;

public class WelcomeScreenActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.welcome\_screen);

findViewById(R.id.btn\_signup).setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) { openSignupActivity();

}

});

findViewById(R.id.btn\_login).setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) { openLoginActivity();

}

});

}

public void openLoginActivity(){

Intent intent = new Intent(this, LoginActivity.class); startActivity(intent);

}

public void openSignupActivity(){

Intent intent = new Intent(this, SignupActivity.class); startActivity(intent);

}

}

# LoginActivity

package com.example.vicup.telemedicine;

import android.content.Context; import android.content.Intent;

import android.graphics.drawable.Drawable; import android.support.v7.app.AppCompatActivity; import android.os.Bundle;

import android.util.Log; import android.view.View;

import android.widget.EditText; import android.widget.Toast;

import com.example.vicup.telemedicine.api.RemoteApi; import com.example.vicup.telemedicine.api.RemoteApiUtils;

import com.example.vicup.telemedicine.api.entities.UserAuthBody; import com.example.vicup.telemedicine.api.entities.UserAuthResult;

import java.util.regex.Matcher; import java.util.regex.Pattern;

import retrofit2.Call; import retrofit2.Callback; import retrofit2.Response;

public class LoginActivity extends AppCompatActivity implements View.OnClickListener{

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.login\_activity); leftIconsInit();

findViewById(R.id.btn\_login).setOnClickListener(this); findViewById(R.id.btn\_signup).setOnClickListener(this);

}

@Override

public void onClick(View v) {

final EditText email = (EditText) findViewById(R.id.email); final EditText password = (EditText) findViewById(R.id.pass);

switch (v.getId()){

case R.id.btn\_login:

if (!validateEmail(email.getText().toString())) { email.setError("Invalid Email"); email.requestFocus();

} else if (!validatePassword(password.getText().toString())) { password.setError("Invalid Password");

password.requestFocus();

} else {

sendAndHandleLoginRequest(v.getContext(), email.getText().toString(), password.getText().toString());

}

break;

case R.id.btn\_signup: openSignupActivity(v.getContext()); break;

}

}

private void leftIconsInit(){

final EditText email = (EditText) findViewById(R.id.email); final EditText password = (EditText) findViewById(R.id.pass);

final float density = getResources().getDisplayMetrics().density;

final Drawable drawable\_email = getResources().getDrawable(R.drawable.login); final Drawable drawable\_password = getResources().getDrawable(R.drawable.pass);

final int width = Math.round(27 \* density); final int height = Math.round(30 \* density);

drawable\_email.setBounds(6, 3, width, height); email.setCompoundDrawables(drawable\_email, null, null, null); drawable\_password.setBounds(6, 3, width, height); password.setCompoundDrawables(drawable\_password, null, null, null);

}

/\*\*

* Sends login request to the server.
* <ul>
* <li>On success, save received token and open HomeScreen activity;</li>
* <li>On failure, make a toast with appropriate message.</li>

\* </ul>

* @param context Context to make a Toast with error message
* @param validatedEmail Email in valid format (according to {@link #validateEmail(String)})
* @param validatedPassword Password in valid format (according to {@link #validatePassword(String)})

\*/

public static void sendAndHandleLoginRequest(final Context context, String validatedEmail, String validatedPassword){

RemoteApi.getApi().userAuth(new UserAuthBody(validatedEmail, validatedPassword)

).enqueue(new Callback<UserAuthResult>() { @Override

public void onResponse(Call<UserAuthResult> call, Response<UserAuthResult>

response) {

RemoteApiUtils.logResponse(this.getClass().toString(), response);

if (!response.isSuccessful()){

Toast.makeText(context, "Unable to send auth request", Toast.LENGTH\_LONG).show();

}

else if (!response.body().getStatus().equalsIgnoreCase("SUCCESS")){ Toast.makeText(context, "Invalid username/password",

Toast.LENGTH\_LONG).show();

}

else if (response.body().getStatus().equalsIgnoreCase("SUCCESS")){ RemoteApi.setTOKEN(response.body().getMessage()); Log.d(context.toString(), "TOKEN - " + RemoteApi.getTOKEN()); LoginActivity.openHomeActivity(context);

}

}

@Override

public void onFailure(Call<UserAuthResult> call, Throwable t) { RemoteApiUtils.logResponse(context.getClass().toString(), t); Toast.makeText(context, "Unable to send auth request. IOException",

Toast.LENGTH\_LONG).show();

}

});

}

public static void openSignupActivity(Context context){

Intent intent = new Intent(context, SignupActivity.class); context.startActivity(intent);

}

/\*\*

* Call on successful login only

\*/

public static void openHomeActivity(Context context){ Intent intent = new Intent(context, HomeActivity.class); context.startActivity(intent);

}

//Return true if password is valid and false if password is invalid public static boolean validatePassword(String password) {

if (password == null){ return false;

}

return password.length() > 6;

}

//Return true if email is valid and false if email is invalid public static boolean validateEmail(String email) {

if (email == null || email.isEmpty()){ return false;

}

String emailPattern = "^[\_A-Za-z0-9-\\+]+(\\.[\_A-Za-z0-9-]+)\*@"

+ "[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)\*(\\.[A-Za-z]{2,})$";

Pattern pattern = Pattern.compile(emailPattern); Matcher matcher = pattern.matcher(email);

return matcher.matches();

}

}

# SignupActivity

package com.example.vicup.telemedicine;

import android.content.Intent; import android.os.Bundle;

import android.support.annotation.Nullable; import android.support.v7.app.AppCompatActivity; import android.view.View;

import android.widget.EditText;

import com.example.vicup.telemedicine.api.entities.UserRegBody; import java.util.Date;

import java.text.ParseException; import java.text.SimpleDateFormat;

public class SignupActivity extends AppCompatActivity implements View.OnClickListener{

public static String REG\_BODY\_EXTRAS\_KEY = "REG\_BODY"; private UserRegBody regBody;

private EditText name; private EditText email; private EditText bday; private Date dt\_bday; private EditText phone; private EditText address; private String prevTitle;

@Override

protected void onStart() { super.onStart();

prevTitle = getTitle().toString(); setTitle("Registration");

}

@Override

protected void onStop() {

super.onStop(); setTitle(prevTitle);

}

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_signup);

initVariables(); setViewClickListeners(); fillInFields();

}

private void setViewClickListeners(){ findViewById(R.id.userPhoto).setOnClickListener(this); findViewById(R.id.btn\_reg\_next).setOnClickListener(this);

}

private void initVariables(){

name = (EditText) findViewById(R.id.txt\_name); email = (EditText) findViewById(R.id.txt\_email); bday = (EditText) findViewById(R.id.txt\_bday); phone = (EditText) findViewById(R.id.txt\_phone); address = (EditText) findViewById(R.id.txt\_address);

}

// FOR TEST ONLY

private void fillInFields(){ name.setText("");

email.setText("");

bday.setText("");

phone.setText(""); address.setText("");

}

@Override

public void onClick(View v) { switch (v.getId()){

case R.id.userPhoto: break;

case R.id.btn\_reg\_next:

if (!validateFieldsAndSetBDate()){ break;

}

regBody = new UserRegBody(); regBody.setAddress(address.getText().toString()); regBody.setEmail(email.getText().toString()); regBody.setFullName(name.getText().toString()); regBody.setBirthday(dt\_bday);

regBody.setPhone(phone.getText().toString());

/\*\*

\* TODO

\*/ regBody.setBase64photo(null);

// start new activity and send completed fields to it Intent i = new Intent(this, CompleteSignupActivity.class); i.putExtra(REG\_BODY\_EXTRAS\_KEY, regBody); startActivity(i);

break;

}

}

// setting dt\_bdate here to avoid handling ParseException twice private boolean validateFieldsAndSetBDate(){

boolean result = true;

SimpleDateFormat formatter = new SimpleDateFormat("dd/mm/yy");

if (address.getText() == null || address.getText().toString().isEmpty()){ result = false;

address.setError("This field is required");

}

if (phone.getText() == null || phone.getText().toString().isEmpty()){ result = false;

phone.setError("This field is required");

}

if (name.getText() == null || name.getText().toString().isEmpty()){ result = false;

name.setError("This field is required");

}

if (!LoginActivity.validateEmail(email.getText().toString())){ result = false;

email.setError("Wrong email format");

}

try {

dt\_bday = formatter.parse(bday.getText().toString());

} catch (ParseException | NullPointerException e) { result = false;

bday.setError("Date should be in dd/mm/yy format");

}

return result;

}

}

# CompleteSignupActivity

package com.example.vicup.telemedicine;

import android.content.Context;

import android.support.v7.app.AppCompatActivity; import android.os.Bundle;

import android.view.View; import android.widget.EditText; import android.widget.Toast;

import com.example.vicup.telemedicine.api.RemoteApi; import com.example.vicup.telemedicine.api.RemoteApiUtils;

import com.example.vicup.telemedicine.api.entities.UserRegBody;

import retrofit2.Call; import retrofit2.Callback; import retrofit2.Response;

public class CompleteSignupActivity extends AppCompatActivity implements View.OnClickListener{

private UserRegBody regBody; private EditText username; private EditText password; private String prevTitle;

private final Context activity = this;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_complete\_signup);

setViewClickListeners(); initVariables();

}

@Override

protected void onStart() { super.onStart();

prevTitle = getTitle().toString(); setTitle("Complete your registration");

}

@Override

protected void onStop() { super.onStop(); setTitle(prevTitle);

}

private void setViewClickListeners(){ findViewById(R.id.btn\_register).setOnClickListener(this);

}

private void initVariables(){

username = (EditText) findViewById(R.id.txt\_user\_name); password = (EditText) findViewById(R.id.txt\_password); regBody = (UserRegBody) this.getIntent()

.getSerializableExtra(SignupActivity.REG\_BODY\_EXTRAS\_KEY);

}

@Override

public void onClick(View v) { switch (v.getId()){

case R.id.btn\_register:

if (!validateFields()){ break;

}

regBody.setUsername(username.getText().toString()); regBody.setPassword(password.getText().toString());

registerAndLogin(); break;

}

}

private boolean validateFields(){ boolean result = true;

if (username.getText() == null || username.getText().toString().isEmpty()){ result = false;

username.setError("This field is required");

}

if (password.getText() == null ||

!LoginActivity.validatePassword(password.getText().toString())){ result = false;

password.setError("This field is required");

}

return result;

}

/\*\*

* Sends registration request to the server. Uses <b>regBody</b> local variable to get necessary data<br/>
* On registration success, sends login request too using {@link LoginActivity#sendAndHandleLoginRequest(Context, String, String)}.
* <ul>
* <li>On success - registrates new user, saves received token and opens HomeScreen activity;</li>
* <li>On failure - makes a toast with appropriate message.</li>

\* </ul>

\*/

private void registerAndLogin(){

RemoteApi.getApi().userRegistration(regBody).enqueue(new Callback<Class<Void>>() { @Override

public void onResponse(Call<Class<Void>> call, Response<Class<Void>> response) { if (response.code() != 201){

RemoteApiUtils.logResponse(activity.toString(), response); Toast.makeText(activity, "Registration failed. Please try again",

Toast.LENGTH\_LONG).show();

}

else {

LoginActivity.sendAndHandleLoginRequest(activity, regBody.getEmail(), regBody.getPassword());

}

}

@Override

public void onFailure(Call<Class<Void>> call, Throwable t) { RemoteApiUtils.logResponse(activity.toString(), t); Toast.makeText(activity, "Registration failed. Please try again",

Toast.LENGTH\_LONG).show();

}

});

}

}

# HomeActivity

package com.example.zveri.telemedicine;

import android.support.v7.app.AppCompatActivity; import android.os.Bundle;

public class HomeActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_home);

}

}

# ApprovedRequestActivity

package com.example.vicup.telemedicine;

import android.content.Intent; import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

public class ApprovedRequestActivity extends AppCompatActivity { @Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);

Intent intent = getIntent(); setContentView(R.layout.approved\_request\_screen);

}

}

# DoctorListActivit

package com.example.vicup.telemedicine;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity; import android.support.v7.widget.LinearLayoutManager; import android.support.v7.widget.RecyclerView;

import android.widget.Toast;

public class DoctorListActivity extends AppCompatActivity implements DoctorListAdapter.ListItemClickListener{

private Toast mToast;

private DoctorListAdapter mDoctorListAdapter; private RecyclerView mRecyclerView;

private int mPosition = RecyclerView.NO\_POSITION;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.doctors\_list);

mRecyclerView = (RecyclerView) findViewById(R.id.recyclerview\_forecast);

LinearLayoutManager layoutManager =

new LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false); mRecyclerView.setLayoutManager(layoutManager); mRecyclerView.setHasFixedSize(true);

mDoctorListAdapter = new DoctorListAdapter(this, this);

/\* Setting the adapter attaches it to the RecyclerView in our layout. \*/ mRecyclerView.setAdapter(mDoctorListAdapter);

}

@Override

public void onListItemClick(int clickedItemIndex) {

if (mToast != null) {

mToast.cancel();

}

String toastMessage = "Item #" + clickedItemIndex + " clicked."; mToast = Toast.makeText(this, toastMessage, Toast.LENGTH\_LONG);

mToast.show();

}

}

# DoctorListAdapter

package com.example.vicup.telemedicine;

import android.content.Context; import android.database.Cursor;

import android.support.annotation.NonNull; import android.support.v7.widget.RecyclerView; import android.view.LayoutInflater;

import android.view.View; import android.view.ViewGroup;

import android.widget.ImageView; import android.widget.TextView;

import com.example.vicup.telemedicine.api.entities.DoctorProfile;

class DoctorListAdapter extends RecyclerView.Adapter<DoctorListAdapter.DoctorViewHolder> {

/\* The context we use to utility methods, app resources and layout inflaters \*/ private final Context mContext;

/\*

* Below, we've defined an interface to handle clicks on items within this Adapter. In the
* constructor of our ForecastAdapter, we receive an instance of a class that has implemented
* said interface. We store that instance in this variable to call the onClick method whenever
* an item is clicked in the list.

\*/

final private ListItemClickListener mOnClickListener;

public interface ListItemClickListener {

void onListItemClick(int clickedItemIndex);

}

private Cursor mCursor;

/\*\*

* Creates a ForecastAdapter.

\*

* @param context Used to talk to the UI and app resources
* @param clickHandler The on-click handler for this adapter. This single handler is called
* when an item is clicked.

\*/

public DoctorListAdapter(@NonNull Context context, ListItemClickListener clickHandler){ mContext = context;

mOnClickListener = clickHandler;

}

@Override

public DoctorViewHolder onCreateViewHolder(ViewGroup viewGroup, int viewType){

View view = LayoutInflater.from(mContext).inflate(R.layout.doctors\_list, viewGroup,

false);

view.setFocusable(true);

return new DoctorViewHolder(view);

}

/\*\*

* OnBindViewHolder is called by the RecyclerView to display the data at the specified
* position. In this method, we update the contents of the ViewHolder to display the weather
* details for this particular position, using the "position" argument that is conveniently
* passed into us.

\*

* @param docviewholder The ViewHolder which should be updated to represent the
* contents of the item at the given position in the data

set. set.

\* @param position The position of the item within the adapter's data

\*/

@Override

public void onBindViewHolder(DoctorViewHolder docviewholder, int position) { mCursor.moveToPosition(position);

DoctorProfile dp = new DoctorProfile(null,null, null,null,null,null,null);

// icon

String doctorPhoto = dp.getPhoto();

//docviewholder.iconView.setText(doctorPhoto);

//forecastAdapterViewHolder.iconView.setImageResource(weatherImageId);

// first & last name

String doctorName = dp.getFullName(); docviewholder.nameView.setText(doctorName);

// speciality

String speciality = dp.getSpecs(); docviewholder.descriptionView.setText(speciality);

//stars

Double stars = dp.getStars(); String st = Double.toString(stars);

docviewholder.adressView.setText(st);

//adress

String adress = dp.getAddress(); docviewholder.adressView.setText(adress);

}

@Override

public int getItemCount() {

if (null == mCursor) return 0; return mCursor.getCount();

}

/\*\*

* A ViewHolder is a required part of the pattern for RecyclerViews. It mostly behaves as
* a cache of the child views for a forecast item. It's also a convenient place to set an
* OnClickListener, since it has access to the adapter and the views.

\*/

class DoctorViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {

final ImageView iconView;

final TextView nameView;

final TextView descriptionView; final TextView starView;

final TextView adressView;

public DoctorViewHolder(View view){ super(view);

iconView = (ImageView) view.findViewById(R.id.doctor\_icon); nameView = (TextView) view.findViewById(R.id.name);

descriptionView = (TextView) view.findViewById(R.id.doctor\_description); starView = (TextView) view.findViewById(R.id.stars);

adressView = (TextView) view.findViewById(R.id.adress);

itemView.setOnClickListener(this);

}

that

/\*\*

* This gets called by the child views during a click. We fetch the date that has been
* selected, and then call the onClick handler registered with this adapter, passing
* date.

\*

* @param v the View that was clicked

\*/ @Override

public void onClick(View v) {

int clickedPosition = getAdapterPosition(); mOnClickListener.onListItemClick(clickedPosition);

}

}

}

# DoctorDetailsActivity

package com.example.vicup.telemedicine;

import android.support.v7.app.AppCompatActivity; import android.os.Bundle;

import com.example.vicup.telemedicine.R;

public class DoctorDetailsActivity extends AppCompatActivity { @Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_doctor\_details);

}

}