**Комп’ютерний практикум №2**

**«Activity and Fragment Lifecycle.**

**Робота з базовими елементами UI в Android»**

Виконали:

студенти групи ІС-62с

Кравенський

Ізбіцький

Соловьов

Олійник

Яськов

public class MainActivity extends AppCompatActivity implements MainView {

private static final String TAG = "MainActivity";

public static final String EXTRA\_LAST\_FRAGMENT = "EXTRA\_LAST\_FRAGMENT";

private MainPresenter mainPresenter;

private ChartsFragment chartsFragment;

private DataFragment dataFragment;

private MaterialDialog confirmDialog;

private boolean isServiceRunning;

private String lastFragment;

private View container;

@Override

protected void onCreate(Bundle savedInstanceState) {

setTheme(R.style.AppTheme);

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

if (savedInstanceState == null) {

lastFragment = DataFragment.TAG;

}

container = findViewById(R.id.container);

mainPresenter = new MainPresenter();

chartsFragment = new ChartsFragment();

dataFragment = new DataFragment();

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

toolbar.setTitle(R.string.app\_name);

setSupportActionBar(toolbar);

setupSpinner(toolbar);

setupDialog();

}

private void setupDialog() {

confirmDialog = new MaterialDialog.Builder(this)

.title(R.string.dialog\_delete\_title)

.content(R.string.dialog\_delete\_content)

.positiveText(R.string.dialog\_agree)

.negativeText(R.string.dialog\_disagree)

.onPositive(new MaterialDialog.SingleButtonCallback() {

@Override

public void onClick(@NonNull MaterialDialog dialog, @NonNull DialogAction which) {

ApplicationDAO dao = new ApplicationDAO(getApplicationContext());

try {

mainPresenter.onClickAgreeConfirmDialog(dao);

} finally {

dao.close();

}

Updatable interfaceUpdate = (Updatable) getSupportFragmentManager()

.findFragmentByTag(lastFragment);

if (interfaceUpdate != null) {

interfaceUpdate.onUpdate();

}

}

})

.build();

}

private void setupSpinner(Toolbar toolbar) {

View toolbarSpinner = LayoutInflater.from(this).inflate(R.layout.toolbar\_spinner, toolbar, false);

ActionBar.LayoutParams layoutParams = new ActionBar.LayoutParams(

ViewGroup.LayoutParams.MATCH\_PARENT, ViewGroup.LayoutParams.MATCH\_PARENT);

toolbar.addView(toolbarSpinner, layoutParams);

StringSpinnerAdapter adapter = new StringSpinnerAdapter(this);

adapter.addItems(getResources().getStringArray(R.array.spinner\_items\_data));

Spinner spinner = (Spinner) toolbarSpinner.findViewById(R.id.toolbar\_spinner);

spinner.setAdapter(adapter);

}

@Override

protected void onPause() {

super.onPause();

mainPresenter.unbindView();

}

@Override

protected void onResume() {

super.onResume();

mainPresenter.bindView(this);

if (Build.VERSION.SDK\_INT == Build.VERSION\_CODES.LOLLIPOP\_MR1 && !checkAcces()) {

Log.d(TAG, "onResume: access = " + checkAcces());

Snackbar.make(container, R.string.message\_need\_access, Snackbar.LENGTH\_INDEFINITE)

.setAction(R.string.action\_button\_settings, new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(Settings.ACTION\_USAGE\_ACCESS\_SETTINGS));

}

}).setActionTextColor(Color.YELLOW)

.show();

} else {

if (lastFragment.equals(DataFragment.TAG)) {

mainPresenter.onShowListOfData();

} else if (lastFragment.equals(ChartsFragment.TAG)) {

mainPresenter.onClickCharts();

}

}

}

@TargetApi(Build.VERSION\_CODES.LOLLIPOP\_MR1)

private boolean checkAcces() {

AppOpsManager appOps = (AppOpsManager) getSystemService(Context.APP\_OPS\_SERVICE);

int mode = appOps.checkOpNoThrow("android:get\_usage\_stats",

android.os.Process.myUid(), getPackageName());

return mode == AppOpsManager.MODE\_ALLOWED;

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.action\_bar\_menu, menu);

RelativeLayout layoutItem = (RelativeLayout) menu.findItem(R.id.item\_menu\_switch).getActionView();

SwitchCompat serviceSwitch = (SwitchCompat) layoutItem.findViewById(R.id.switch\_for\_action\_bar);

serviceSwitch.setChecked(isServiceRunning);

serviceSwitch.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {

@Override

public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {

if (isChecked) {

mainPresenter.onEnableService();

} else {

mainPresenter.onDisableService();

}

}

});

return super.onCreateOptionsMenu(menu);

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

/\*

case R.id.item\_settings: {

mainPresenter.onClickSettings();

break;

}

\*/

case R.id.item\_charts: {

mainPresenter.onClickCharts();

break;

}

case android.R.id.home: {

mainPresenter.onBackToListOfData();

break;

}

case R.id.item\_clear\_all\_apps: {

mainPresenter.onClickClearData();

}

}

return super.onOptionsItemSelected(item);

}

@Override

public void showConfirmationDialog() {

confirmDialog.show();

}

@Override

public void showListOfData(boolean animationExit) {

Log.d(TAG, "showListOfData: animationExit == " + animationExit);

FragmentManager manager = getSupportFragmentManager();

FragmentTransaction transaction = manager.beginTransaction();

if (!animationExit) {

transaction.setCustomAnimations(R.anim.enter\_from\_right, R.anim.exit\_to\_left);

} else {

transaction.setCustomAnimations(R.anim.enter\_from\_left, R.anim.exit\_to\_right);

}

transaction.replace(R.id.container, dataFragment, DataFragment.TAG).commit();

lastFragment = DataFragment.TAG;

}

@Override

public void goToSettings() {

startActivity(new Intent(this, SettingsActivity.class));

}

@Override

public void goToCharts() {

mainPresenter.onShowHomeButton();

getSupportFragmentManager()

.beginTransaction()

.setCustomAnimations(R.anim.enter\_from\_right, R.anim.exit\_to\_left)

.replace(R.id.container, chartsFragment, ChartsFragment.TAG)

.commit();

lastFragment = ChartsFragment.TAG;

}

@Override

public boolean checkStateService() {

return isServiceRunning(TrackerService.class);

}

private boolean isServiceRunning(Class<?> serviceClass) {

ActivityManager manager = (ActivityManager) getSystemService(Context.ACTIVITY\_SERVICE);

for (ActivityManager.RunningServiceInfo service : manager.getRunningServices(Integer.MAX\_VALUE)) {

if (serviceClass.getName().equals(service.service.getClassName())) {

return true;

}

}

return false;

}

@Override

public void syncStateSwitch(boolean state) {

isServiceRunning = state;

}

@Override

public void visibleHomeButton(boolean isVisible) {

ActionBar actionBar = getSupportActionBar();

if (actionBar != null) {

actionBar.setDisplayHomeAsUpEnabled(isVisible);

}

}

@Override

public void showMassageAfterDelete() {

Snackbar.make(container, R.string.message\_data\_deleted, Snackbar.LENGTH\_SHORT).show();

}

@Override

public void showStateService(boolean isEnableService) {

Snackbar.make(container,

isEnableService ? getString(R.string.message\_tracker\_start) : getString(R.string.message\_tracker\_stop),

Snackbar.LENGTH\_SHORT).show();

}

@Override

public void enableService() {

startService(createIntent());

}

@Override

public void disableService() {

stopService(createIntent());

}

private Intent createIntent() {

return new Intent(this, TrackerService.class);

}

@Override

protected void onSaveInstanceState(Bundle outState) {

outState.putString(EXTRA\_LAST\_FRAGMENT, lastFragment);

super.onSaveInstanceState(outState);

}

@Override

protected void onRestoreInstanceState(Bundle savedInstanceState) {

if (savedInstanceState != null) {

lastFragment = savedInstanceState.getString(EXTRA\_LAST\_FRAGMENT);

}

super.onRestoreInstanceState(savedInstanceState);

}

