**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №8

по дисциплине: **«Разработка приложений для мобильных устройств»**

на тему: **Обращение с данными и их долговременное хранение**

Выполнил: студент гр. ИТП-31

Дашкевич Д.А.

Принял: ассистент

Белявский Е. В.

Гомель 2019

**Цель**: изучить программирование долговременного хранения данных мобильного приложения.

**Ход работы**

**Вариант 2**

# **Задание:**

Добавить в приложение, разработанное в лабораторной работе №7 функционал, выполняющий сохранение всех данных приложения в БД *Android* и выборку их при необходимости. Выполнить запуск приложения на эмуляторе.

2. *Customer*: *id*, Фамилия, Имя, Отчество, Адрес, Номер кредитной карточки, Номер банковского счёта.

Создать массив объектов. Вывести:

а) список покупателей в алфавитном порядке;

б) список покупателей, у которых номер кредитной карточки находится в заданном интервале.

**Результат работы приложения:**

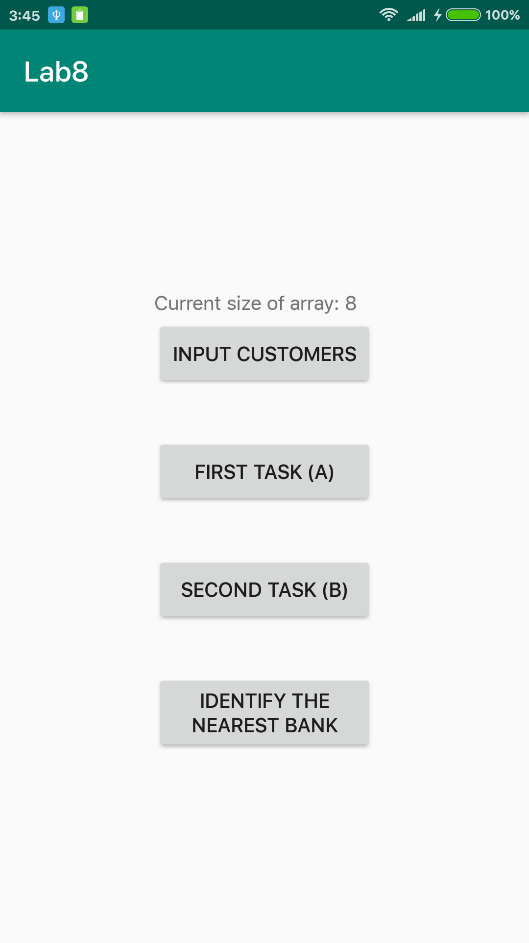


Рисунок 1 – Главное окно приложения с кнопками меню, у которого были сохранены восемь записей

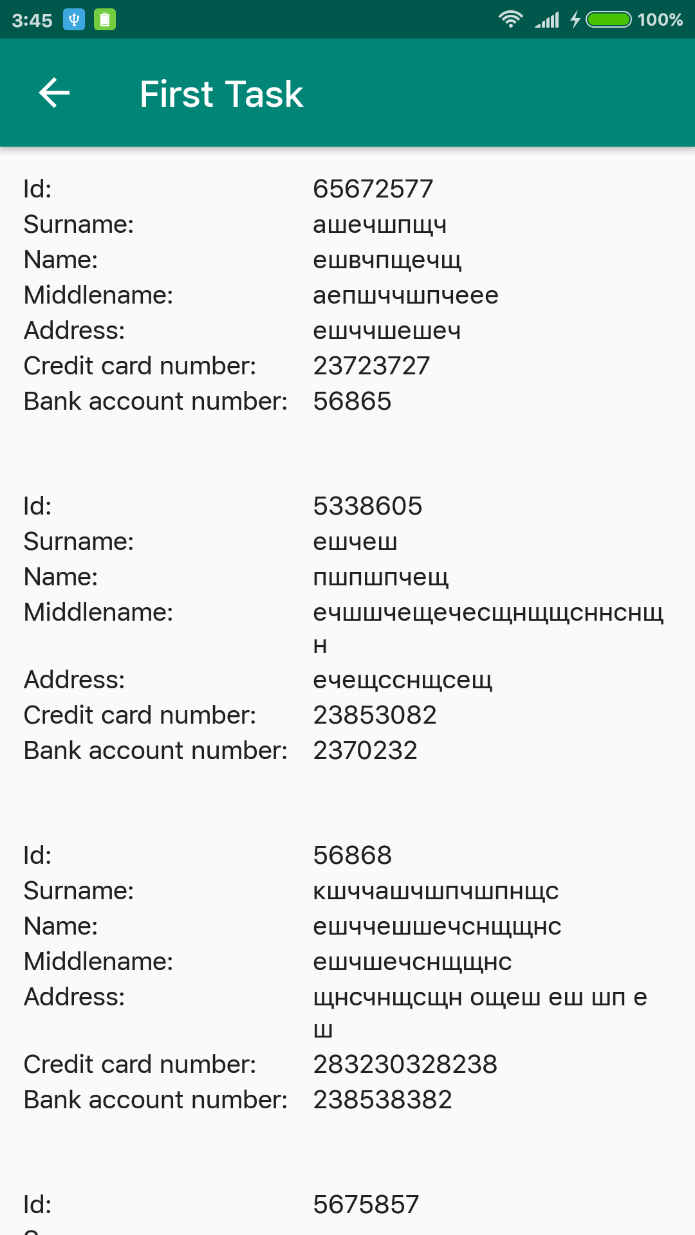


Рисунок 2 – Список клиентов в алфавитном порядке

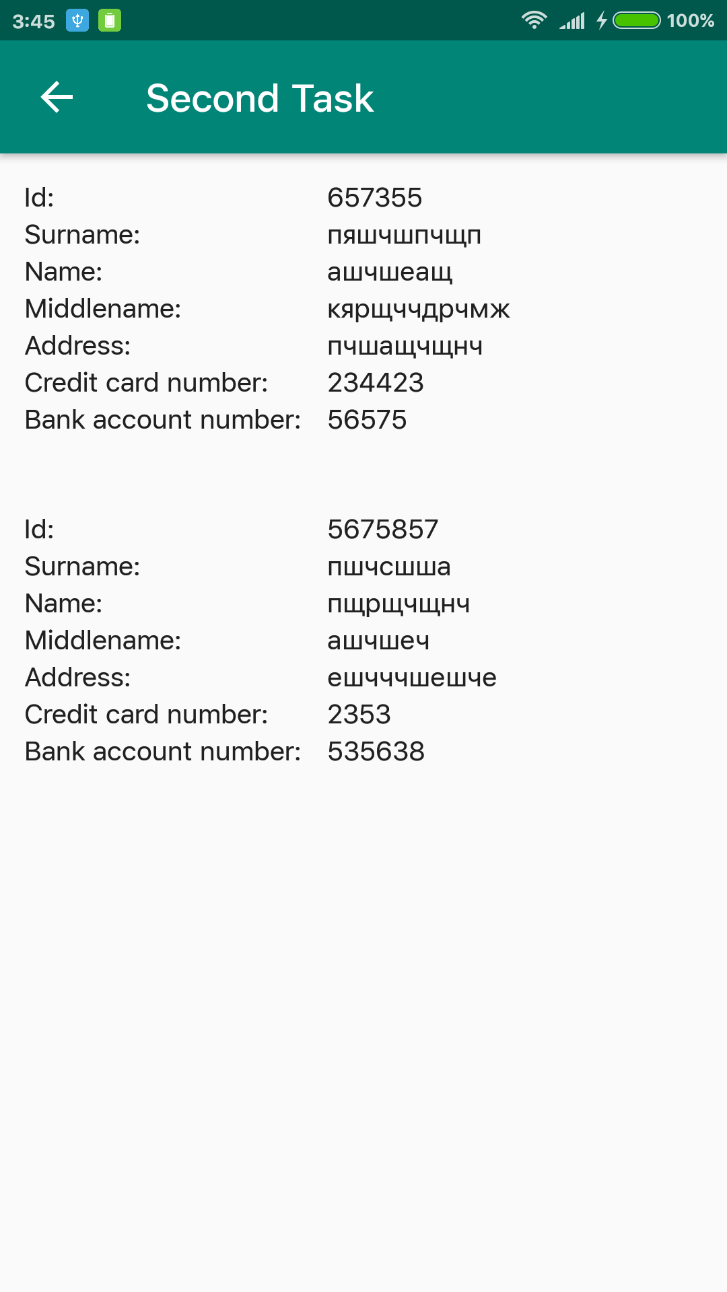


Рисунок 3 – Список клиентов с номерами кредитных карт в диапазон от 100 до 830000

**Вывод:** в результате выполнения лабораторной работы было изучено программирование долговременного хранения данных в *Android*, также написано и запущено приложение, которое использовало *SQLite*.

**Приложение А**

**Исходный код приложения**

**MainActivity.java:**

**public class** MainActivity **extends** AppCompatActivity {  
  
 **private static final int *PERMISSION\_REQUEST\_CODE*** = 94;  
 **private static final** String ***CURRENT\_SIZE\_STRING*** = **"Current size of array: "**;  
  
 **private final** Bank[] **mBanks** = {  
 **new** Bank(52.41481045106871, 30.950411769683832, **"Банк Дабрабыт"**),  
 **new** Bank(52.39978226246882, 30.913480858474717, **"Беларусбанк"**),  
 **new** Bank(52.4072590718227, 30.915726999999986, **"Приорбанк, ЦБУ 400/1"**),  
 **new** Bank(52.44407957177429, 31.001480499999996, **"БелВЭБ"**),  
 **new** Bank(52.42550557178379, 30.994814499999972, **"РРБ-Банк"**)  
 };  
  
  
 **private** TextView **mCurrentSizeTextView**;  
 **private** TextView **mNearestBankTextView**;  
  
 **private** LocationManager **mLocationManager**;  
 **private** LocationListener **mLocationListener**;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 Log.*d*(**"MAIN"**, **"onCreate"**);  
 setContentView(R.layout.***activity\_main***);  
 bindButtons();  
  
 **mCurrentSizeTextView** = findViewById(R.id.***currentSizeOfArrayTextView***);  
 **mNearestBankTextView** = findViewById(R.id.***nearestBankTextView***);  
  
 **mLocationManager** = (LocationManager) getSystemService(***LOCATION\_SERVICE***);  
  
 Database.*initDatabase*(**this**);  
 CustomerList.*createInstance*();  
 **int** size = CustomerList.*getInstance*().getSize();  
 String t = ***CURRENT\_SIZE\_STRING*** + size;  
 **mCurrentSizeTextView**.setText(t);  
  
 **mLocationListener** = createLocationListener();  
 }  
  
 @Override  
 **protected void** onStart() {  
 **super**.onStart();  
 Log.*d*(**"MAIN"**, **"onStart"**);  
 }  
  
 @Override  
 **protected void** onResume() {  
 **super**.onResume();  
 Log.*d*(**"MAIN"**, **"onResume"**);  
 }  
  
 @Override  
 **protected void** onPause() {  
 **super**.onPause();  
 Log.*d*(**"MAIN"**, **"onPause"**);  
 }  
  
 @Override  
 **protected void** onStop() {  
 **super**.onStop();  
 Log.*d*(**"MAIN"**, **"onStop"**);  
 }  
  
 @Override  
 **protected void** onDestroy() {  
 Log.*d*(**"MAIN"**, **"onDestroy"**);

Database.closeDatabase();  
 **mLocationManager**.removeUpdates(**mLocationListener**);  
 **super**.onDestroy();  
 }  
  
  
 **private void** bindButtons() {  
 Button inputButton = findViewById(R.id.***inputCustomersButton***);  
 Button openFirstActivityButton = findViewById(R.id.***firstTaskButton***);  
 Button openSecondActivityButton = findViewById(R.id.***secondTaskButton***);  
 Button identifyNearestBankButton = findViewById(R.id.***identifyNearestBankButton***);  
  
 inputButton.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 MainActivity.**this**.openInputArraySizeDialog();  
 }  
 });  
  
 openFirstActivityButton.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 openActivity(**new** Intent(MainActivity.**this**, FirstActivity.**class**));  
 }  
 });  
  
 openSecondActivityButton.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 openActivity(**new** Intent(MainActivity.**this**, SecondActivity.**class**));  
 }  
 });  
  
 identifyNearestBankButton.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (ContextCompat.*checkSelfPermission*(getApplicationContext(),  
 Manifest.permission.***ACCESS\_FINE\_LOCATION***)  
 != PackageManager.***PERMISSION\_GRANTED*** || ContextCompat.*checkSelfPermission*(getApplicationContext(),  
 Manifest.permission.***ACCESS\_COARSE\_LOCATION***)  
 != PackageManager.***PERMISSION\_GRANTED***) {  
 ActivityCompat.*requestPermissions*(MainActivity.**this**,  
 **new** String[]{Manifest.permission.***ACCESS\_FINE\_LOCATION***,  
 Manifest.permission.***ACCESS\_COARSE\_LOCATION***},  
 ***PERMISSION\_REQUEST\_CODE***);  
 } **else** {  
 **mLocationManager**.requestLocationUpdates(LocationManager.***GPS\_PROVIDER***,  
 10000, 0, **mLocationListener**);  
 }  
 }  
 });  
 }  
  
 **private void** openActivity(Intent intent) {  
 **if** (CustomerList.*getInstance*() != **null**) {  
 startActivity(intent);  
 } **else** {  
 Toast.*makeText*(**this**, **"No data available!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
  
 **private void** openInputArraySizeDialog() {  
 createInputArraySizeDialog().show();  
 }  
  
 @SuppressLint(**"InflateParams"**)  
 **private** AlertDialog createInputArraySizeDialog() {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
 LayoutInflater inflater = **this**.getLayoutInflater();  
 **final** View view = inflater.inflate(R.layout.***input\_array\_size\_dialog***, **null**);  
 builder.setView(view)  
 .setPositiveButton(**"Accept"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** id) {  
 acceptArraySizeDialog(view);  
 }  
 }).setNegativeButton(**"Cancel"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 dialog.dismiss();  
 }  
 }).setCancelable(**false**);  
 **return** builder.create();  
 }  
  
 **private void** acceptArraySizeDialog(View view) {  
 EditText sizeOfArrayEditText = view.findViewById(R.id.***sizeOfArrayEditText***);  
 String s = sizeOfArrayEditText.getText().toString();  
 **if** (!s.isEmpty()) {  
 **int** size = Integer.*valueOf*(s);  
 **if** (size > 0) {  
 CustomerList.*createInstance*(size);  
 String t = ***CURRENT\_SIZE\_STRING*** + size;  
 **mCurrentSizeTextView**.setText(t);  
 CustomerList.*getInstance*().deleteCustomers();  
 openInputCustomerDialog();  
 } **else** {  
 Toast.*makeText*(getApplicationContext(), **"Size must be positive!"**,  
 Toast.***LENGTH\_SHORT***).show();  
 }  
 } **else** {  
 Toast.*makeText*(getApplicationContext(), **"The input field must be filled in!"**,  
 Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
  
  
 **private void** openInputCustomerDialog() {  
 AlertDialog dialog = createInputCustomerDialog();  
 dialog.show();  
 }  
  
 @SuppressLint(**"InflateParams"**)  
 **private** AlertDialog createInputCustomerDialog() {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
 LayoutInflater inflater = **this**.getLayoutInflater();  
 **final** View view = inflater.inflate(R.layout.***input\_customer\_dialog***, **null**);  
 builder.setView(view)  
 .setPositiveButton(**"Add"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** id) {  
 addCustomer(view);  
 **if** (CustomerList.*getInstance*().getSize() != CustomerList.*getInstance*().getmMaxSize()) {  
 openInputCustomerDialog();  
 }  
 }  
 }).setCancelable(**false**);  
 **return** builder.create();  
 }  
  
 **private void** addCustomer(View view) {  
 String customerSurname = ((EditText)  
 view.findViewById(R.id.***customerSurnameEditText***)).getText().toString();  
 String customerName = ((EditText)  
 view.findViewById(R.id.***customerNameEditText***)).getText().toString();  
 String customerMiddlename = ((EditText)  
 view.findViewById(R.id.***customerMiddlenameEditText***)).getText().toString();  
 String customerAddress = ((EditText)  
 view.findViewById(R.id.***customerAddressEditText***)).getText().toString();  
 **try** {  
 **int** customerId = Integer.*valueOf*(((EditText)  
 view.findViewById(R.id.***customerIdEditText***)).getText().toString());  
 **long** customerCreditCardNumber = Long.*valueOf*(((EditText)  
 view.findViewById(R.id.***customerCreditCardNumberEditText***)).getText().toString());  
 **long** customerBankAccountNumber = Long.*valueOf*(((EditText)  
 view.findViewById(R.id.***customerBankAccountNumberEditText***)).getText().toString());  
 CustomerList.*getInstance*().addCustomer(**new** Customer(customerId, customerSurname,  
 customerName, customerMiddlename, customerAddress, customerCreditCardNumber,  
 customerBankAccountNumber));  
 } **catch** (NumberFormatException ex) {  
 Toast.*makeText*(**this**, **"Input fields must be filled in!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
  
 **private** LocationListener createLocationListener() {  
 **return new** LocationListener() {  
 @Override  
 **public void** onLocationChanged(Location location) {  
 showLocation(location);  
 }  
  
 @Override  
 **public void** onProviderDisabled(String provider) {}  
  
 @Override  
 **public void** onProviderEnabled(String provider) {  
 **if** (ContextCompat.*checkSelfPermission*(getApplicationContext(),  
 Manifest.permission.***ACCESS\_FINE\_LOCATION***)  
 == PackageManager.***PERMISSION\_GRANTED*** && ContextCompat.*checkSelfPermission*(getApplicationContext(),  
 Manifest.permission.***ACCESS\_COARSE\_LOCATION***)  
 == PackageManager.***PERMISSION\_GRANTED***) {  
 showLocation(**mLocationManager**.getLastKnownLocation(provider));  
 }  
 }  
  
 @Override **public void** onStatusChanged(String provider, **int** status, Bundle extras) {}  
 };  
 }  
  
 **private void** showLocation(Location location) {  
 **if** (location == **null**)  
 **return**;  
 **if** (location.getProvider().equals(LocationManager.***GPS\_PROVIDER***)  
 || location.getProvider().equals(LocationManager.***NETWORK\_PROVIDER***)) {  
 **int** idx = -1;  
 **float** min = Float.***MAX\_VALUE***;  
 **for** (**int** i = 0; i < **mBanks**.**length**; i++) {  
 **float** t = **mBanks**[i].getLocation().distanceTo(location);  
 **if** (t < min) {  
 min = t;  
 idx = i;  
 }  
 }  
 String t = **"Nearest bank: "** + **mBanks**[idx].getName();  
 **mNearestBankTextView**.setText(t);  
 }  
 }  
  
  
 **private class** Bank {  
 **private** Location **location**;  
 **private** String **name**;  
  
 Bank(**double** latitude, **double** longitude, String name) {  
 **this**.**location** = **new** Location(**""**);  
 **this**.**location**.setLatitude(latitude);  
 **this**.**location**.setLongitude(longitude);  
 **this**.**name** = name;  
 }  
  
 Location getLocation() {  
 **return location**;  
 }  
  
 String getName() {  
 **return name**;  
 }  
 }  
}

**FirstActivity.java:**

**public class** FirstActivity **extends** AppCompatActivity {  
 @Override  
 **protected void** onCreate(@Nullable Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_first***);  
 ActionBar actionBar = getSupportActionBar();  
 **if** (actionBar != **null**) {  
 actionBar.setHomeButtonEnabled(**true**);  
 actionBar.setDisplayHomeAsUpEnabled(**true**);  
 }  
 setDataForRecyclerView();  
 }  
  
 **private void** setDataForRecyclerView() {  
 RecyclerView recyclerView = findViewById(R.id.***customersRecyclerView***);  
 recyclerView.setHasFixedSize(**true**);  
 recyclerView.setLayoutManager(**new** LinearLayoutManager(**this**));  
 recyclerView.setAdapter(**new** CustomersAdapter(CustomerList.*getInstance*()  
 .getCustomerArrayInAlphabeticalOrder()));  
 }  
  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **if** (item.getItemId() == android.R.id.***home***) {  
 **this**.finish();  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
}

**SecondActivity.java:**

**public class** SecondActivity **extends** AppCompatActivity {  
 @Override  
 **protected void** onCreate(@Nullable Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_second***);  
 ActionBar actionBar = getSupportActionBar();  
 **if** (actionBar != **null**) {  
 actionBar.setHomeButtonEnabled(**true**);  
 actionBar.setDisplayHomeAsUpEnabled(**true**);  
 }  
 openInputRangeDialog();  
 }  
  
 **private void** setDataForRecyclerView(View itemView) {  
 **try** {  
 **long** begin = Long.*valueOf*(((EditText)  
 itemView.findViewById(R.id.***beginEditText***)).getText().toString());  
 **long** end = Long.*valueOf*(((EditText)  
 itemView.findViewById(R.id.***endEditText***)).getText().toString());  
 RecyclerView recyclerView = findViewById(R.id.***customersRecyclerView***);  
 recyclerView.setHasFixedSize(**true**);  
 recyclerView.setLayoutManager(**new** LinearLayoutManager(**this**));  
 recyclerView.setAdapter(**new** CustomersAdapter(CustomerList.*getInstance*()  
 .getCustomerArrayWithCardNumbersInRange(begin, end)));  
 } **catch** (NumberFormatException ex) {  
 Toast.*makeText*(**this**, **"Input fields must be filled in!"**, Toast.***LENGTH\_SHORT***).show();  
 openInputRangeDialog();  
 }  
  
 }  
  
 **private void** openInputRangeDialog() {  
 createInputRangeDialog().show();  
 }  
  
 @SuppressLint(**"InflateParams"**)  
 **private** AlertDialog createInputRangeDialog() {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
 LayoutInflater inflater = **this**.getLayoutInflater();  
 **final** View view = inflater.inflate(R.layout.***input\_range\_dialog***, **null**);  
 builder.setView(view)  
 .setPositiveButton(**"Accept"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** id) {  
 setDataForRecyclerView(view);  
 }  
 }).setCancelable(**false**);  
 **return** builder.create();  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **if** (item.getItemId() == android.R.id.***home***) {  
 **this**.finish();  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
}

**CustomersAdapter.java:**

**public class** CustomersAdapter **extends** RecyclerView.Adapter<CustomersAdapter.CustomerViewHolder> {  
  
 **private** List<Customer> mCustomers;  
  
 **public** CustomersAdapter(List<Customer> customers) {  
 **this**.mCustomers = customers;  
 }  
  
 @NonNull  
 @Override  
 **public** CustomerViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, **int** i) {  
 View view = LayoutInflater.from(viewGroup.getContext())  
 .inflate(R.layout.customer\_item, viewGroup, **false**);  
 **return new** CustomerViewHolder(view);  
 }  
  
  
 @Override  
 **public void** onBindViewHolder(@NonNull CustomersAdapter.CustomerViewHolder holder, **int** position) {  
 Customer customer = mCustomers.get(position);  
 holder.idView.setText(String.valueOf(customer.getId()));  
 holder.surnameView.setText(customer.getSurname());  
 holder.nameView.setText(customer.getName());  
 holder.middlenameView.setText(customer.getMiddlename());  
 holder.addressView.setText(customer.getAddress());  
 holder.creditCardNumberView.setText(String.valueOf(customer.getCreditCardNumber()));  
 holder.bankAccountNumberView.setText(String.valueOf(customer.getBankAccountNumber()));  
 }  
  
 @Override  
 **public int** getItemCount() {  
 **return** mCustomers.size();  
 }  
  
 **class** CustomerViewHolder **extends** RecyclerView.ViewHolder {  
 TextView idView, surnameView, nameView, middlenameView,  
 addressView, creditCardNumberView, bankAccountNumberView;  
  
 CustomerViewHolder(@NonNull View view) {  
 **super**(view);  
 idView = view.findViewById(R.id.customerIdTextView);  
 surnameView = view.findViewById(R.id.customerSurnameTextView);  
 nameView = view.findViewById(R.id.customerNameTextView);  
 middlenameView = view.findViewById(R.id.customerMiddlenameTextView);  
 addressView = view.findViewById(R.id.customerAddressTextView);  
 creditCardNumberView = view.findViewById(R.id.customerCreditCardNumberTextView);  
 bankAccountNumberView = view.findViewById(R.id.customerBankAccountNumberTextView);  
 }  
 }  
}

**Customer.java:**

**public class** Customer {  
  
 **private int mId**;  
 **private** String **mSurname**;  
 **private** String **mName**;  
 **private** String **middlename**;  
 **private** String **mAddress**;  
 **private long mCreditCardNumber**;  
 **private long mBankAccountNumber**;  
  
 **public** Customer(**int** id, String surname, String name, String middlename, String address,  
 **long** creditCardNumber, **long** bankAccountNumber) {  
 **this**.**mId** = id;  
 **this**.**mSurname** = surname;  
 **this**.**mName** = name;  
 **this**.**middlename** = middlename;  
 **this**.**mAddress** = address;  
 **this**.**mCreditCardNumber** = creditCardNumber;  
 **this**.**mBankAccountNumber** = bankAccountNumber;  
 }  
  
 **public int** getId() {  
 **return mId**;  
 }  
  
 **public** String getSurname() {  
 **return mSurname**;  
 }  
  
 **public** String getName() {  
 **return mName**;  
 }  
  
 **public** String getMiddlename() {  
 **return middlename**;  
 }  
  
 **public** String getAddress() {  
 **return mAddress**;  
 }  
  
 **public long** getCreditCardNumber() {  
 **return mCreditCardNumber**;  
 }  
  
 **public long** getBankAccountNumber() {  
 **return mBankAccountNumber**;  
 }  
}

**CustomerList.java:**

**public class** CustomerList {  
 **private static** CustomerList *sCustomerList* = **null**;  
  
 **private int mMaxSize**;  
 **private** CustomersCache **mCustomersCache**;  
  
 **private** CustomerList(**int** maxSize) {  
 **this**.**mMaxSize** = maxSize;  
 **this**.**mCustomersCache** = **new** CustomersCache();  
 }  
  
 **public static void** createInstance(**int** size) {  
 *sCustomerList* = **new** CustomerList(size);  
 }  
  
 **public static void** createInstance() {  
 *createInstance*(0);  
 }  
  
 **public static** CustomerList getInstance() {  
 **return** *sCustomerList*;  
 }  
  
 **public void** addCustomer(Customer customer) {  
 **mCustomersCache**.addCustomer(customer);  
 }  
  
 **public int** getmMaxSize() {  
 **return mMaxSize**;  
 }  
  
 **public int** getSize() {  
 **return mCustomersCache**.getNumberOfCustomers();  
 }  
  
 **public** List<Customer> getCustomerArrayInAlphabeticalOrder() {  
 **return mCustomersCache**.getCustomers();  
 }  
  
 **public** List<Customer> getCustomerArrayWithCardNumbersInRange(**long** begin, **long** end) {  
 **return mCustomersCache**.getCustomersInCardNumberRange(begin, end);  
 }  
  
 **public void** deleteCustomers() {  
 **mCustomersCache**.deleteCustomers();  
 }  
}

**DatabaseHelper.java:**

**public final class** DatabaseHelper **extends** SQLiteOpenHelper {  
  
 **private static final** String ***DATABASE\_NAME*** = **"lab8\_db"**;  
 **private static final int *DATABASE\_VERSION*** = 1;  
  
 **static final** String ***TABLE\_CUSTOMERS*** = **"customers"**;  
 **static final** String ***KEY\_ID*** = **"id"**;  
 **static final** String ***KEY\_SURNAME*** = **"surname"**;  
 **static final** String ***KEY\_NAME*** = **"name"**;  
 **static final** String ***KEY\_MIDDLENAME*** = **"middlename"**;  
 **static final** String ***KEY\_ADDRESS*** = **"address"**;  
 **static final** String ***KEY\_CREDIT\_CARD\_NUMBER*** = **"creditCardNumber"**;  
 **static final** String ***KEY\_BANK\_ACCOUNT\_NUMBER*** = **"bankAccountNumber"**;  
  
 **private static final** String ***CREATE\_CUSTOMERS\_SQL*** = **"CREATE TABLE "** + ***TABLE\_CUSTOMERS*** + **" ("** + ***KEY\_ID*** + **" INTEGER, "** + ***KEY\_SURNAME*** + **" TEXT, "** + ***KEY\_NAME*** + **" TEXT, "** + ***KEY\_MIDDLENAME*** + **" TEXT, "** + ***KEY\_ADDRESS*** + **" TEXT, "** + ***KEY\_CREDIT\_CARD\_NUMBER*** + **" INTEGER, "** + ***KEY\_BANK\_ACCOUNT\_NUMBER*** + **" INTEGER)"**;  
  
 **private static final** String ***DROP\_CUSTOMERS\_SQL*** = **"DROP TABLE IF EXISTS "** + ***TABLE\_CUSTOMERS***;  
  
 DatabaseHelper(@Nullable Context context) {  
 **super**(context, ***DATABASE\_NAME***, **null**, ***DATABASE\_VERSION***);  
 }  
  
 @Override  
 **public void** onCreate(SQLiteDatabase db) {  
 db.execSQL(***CREATE\_CUSTOMERS\_SQL***);  
 }  
  
 @Override  
 **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
 db.execSQL(***DROP\_CUSTOMERS\_SQL***);  
 onCreate(db);  
 }  
  
 @Override  
 **public void** onDowngrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
 onUpgrade(db, oldVersion, newVersion);  
 }  
}

**Database.java:**

**public final class** Database {  
  
 **private static** DatabaseHelper *sDbHelper*;  
 **private static** SQLiteDatabase *sSqLiteDatabase*;  
  
 **private** Database() {}  
  
 **public static void** initDatabase(Context context) {  
 **if** (*sDbHelper* == **null**) {  
 *sDbHelper* = **new** DatabaseHelper(context);  
 }  
 *getDatabase*();  
 }  
  
 **static** SQLiteDatabase getDatabase() {  
 **if** (*sSqLiteDatabase* == **null** || *sSqLiteDatabase*.isReadOnly()) {  
 *sSqLiteDatabase* = *sDbHelper*.getWritableDatabase();  
 }  
 **return** *sSqLiteDatabase*;  
 }  
  
 **public static** DatabaseHelper getHelper() {  
 **return** *sDbHelper*;  
 }  
  
 **public static void** closeDatabase() {  
 *sDbHelper* = **null**;  
 *sSqLiteDatabase*.close();  
 *sSqLiteDatabase* = **null**;  
 }  
}

**CustomersCache.java:**

**public class** CustomersCache {  
  
 **private static final** String ***GET\_NUMBER\_OF\_CUSTOMERS*** = **"SELECT COUNT(\*) FROM "** +  
 DatabaseHelper.***TABLE\_CUSTOMERS***;  
  
 **private static final** String ***GET\_CUSTOMERS\_SQL*** = **"SELECT \* FROM "** +  
 DatabaseHelper.***TABLE\_CUSTOMERS*** + **" ORDER BY "** + DatabaseHelper.***KEY\_SURNAME***;  
  
 **private static final** String ***GET\_CUSTOMERS\_IN\_CARD\_NUMBER\_RANGE\_SQL*** = **"SELECT \* FROM "** +  
 DatabaseHelper.***TABLE\_CUSTOMERS*** + **" WHERE "** + DatabaseHelper.***KEY\_CREDIT\_CARD\_NUMBER*** +  
 **" >= ? AND "** + DatabaseHelper.***KEY\_CREDIT\_CARD\_NUMBER*** + **" <= ?"**;  
  
 **private static final** String ***ADD\_CUSTOMER\_SQL*** = **"INSERT OR REPLACE INTO "** +  
 DatabaseHelper.***TABLE\_CUSTOMERS*** +**" ("** +  
 DatabaseHelper.***KEY\_ID*** + **", "** + DatabaseHelper.***KEY\_SURNAME*** + **", "** +  
 DatabaseHelper.***KEY\_NAME*** + **", "** + DatabaseHelper.***KEY\_MIDDLENAME*** + **", "** +  
 DatabaseHelper.***KEY\_ADDRESS*** + **", "** + DatabaseHelper.***KEY\_CREDIT\_CARD\_NUMBER*** + **", "** +  
 DatabaseHelper.***KEY\_BANK\_ACCOUNT\_NUMBER*** + **") VALUES (?, ?, ?, ?, ?, ?, ?)"**;  
  
 **private static final** String ***DELETE\_CUSTOMERS*** = **"DELETE FROM "** + DatabaseHelper.***TABLE\_CUSTOMERS***;  
  
  
 **public int** getNumberOfCustomers() {  
 Cursor cursor = Database.*getDatabase*().rawQuery(***GET\_NUMBER\_OF\_CUSTOMERS***, **null**);  
 cursor.moveToFirst();  
 **int** number = cursor.getInt(0);  
 cursor.close();  
 **return** number;  
 }  
  
 **public** List<Customer> getCustomers() {  
 Cursor cursor = Database.*getDatabase*().rawQuery(***GET\_CUSTOMERS\_SQL***, **null**);  
 **return** getCustomersByCursor(cursor);  
 }  
  
 **public** List<Customer> getCustomersInCardNumberRange(**long** begin, **long** end) {  
 Cursor cursor = Database.*getDatabase*().rawQuery(***GET\_CUSTOMERS\_IN\_CARD\_NUMBER\_RANGE\_SQL***,  
 **new** String[] {String.*valueOf*(begin), String.*valueOf*(end)});  
 **return** getCustomersByCursor(cursor);  
 }  
  
 **private** List<Customer> getCustomersByCursor(Cursor cursor) {  
 **int** indexId = cursor.getColumnIndex(DatabaseHelper.***KEY\_ID***);  
 **int** indexSurname = cursor.getColumnIndex(DatabaseHelper.***KEY\_SURNAME***);  
 **int** indexName = cursor.getColumnIndex(DatabaseHelper.***KEY\_NAME***);  
 **int** indexMiddlename = cursor.getColumnIndex(DatabaseHelper.***KEY\_MIDDLENAME***);  
 **int** indexAddress = cursor.getColumnIndex(DatabaseHelper.***KEY\_ADDRESS***);  
 **int** indexCreditCardNumber = cursor.getColumnIndex(DatabaseHelper.***KEY\_CREDIT\_CARD\_NUMBER***);  
 **int** indexBankAccountNumber = cursor.getColumnIndex(DatabaseHelper.***KEY\_BANK\_ACCOUNT\_NUMBER***);  
  
 List<Customer> list = **new** ArrayList<>();  
 **if** (cursor.moveToFirst()) {  
 **do** {  
 list.add(**new** Customer(cursor.getInt(indexId), cursor.getString(indexSurname),  
 cursor.getString(indexName), cursor.getString(indexMiddlename),  
 cursor.getString(indexAddress), cursor.getLong(indexCreditCardNumber),  
 cursor.getLong(indexBankAccountNumber)));  
 } **while** (cursor.moveToNext());  
 }  
 cursor.close();  
 **return** list;  
 }  
  
  
 **public void** addCustomer(Customer customer) {  
 SQLiteStatement sqLiteStatement = Database.*getDatabase*().compileStatement(***ADD\_CUSTOMER\_SQL***);  
 sqLiteStatement.bindLong(1, customer.getId());  
 sqLiteStatement.bindString(2, customer.getSurname());  
 sqLiteStatement.bindString(3, customer.getName());  
 sqLiteStatement.bindString(4, customer.getMiddlename());  
 sqLiteStatement.bindString(5, customer.getAddress());  
 sqLiteStatement.bindLong(6, customer.getCreditCardNumber());  
 sqLiteStatement.bindLong(7, customer.getBankAccountNumber());  
 sqLiteStatement.execute();  
 }  
  
  
 **public void** deleteCustomers() {  
 Database.*getDatabase*().execSQL(***DELETE\_CUSTOMERS***);  
 }  
}

**activity\_main.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".controller.MainActivity"**>  
  
 <**TextView  
 android:id="@+id/currentSizeOfArrayTextView"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="120dp"  
 android:layout\_marginEnd="8dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
 <**Button  
 android:id="@+id/inputCustomersButton"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="140dp"  
 android:layout\_marginEnd="8dp"  
 android:text="@string/input\_customers"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
 <**Button  
 android:id="@+id/firstTaskButton"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="220dp"  
 android:layout\_marginEnd="8dp"  
 android:text="@string/first\_task\_a"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
 <**Button  
 android:id="@+id/secondTaskButton"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="300dp"  
 android:layout\_marginEnd="8dp"  
 android:text="@string/second\_task\_b"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
  
 <**Button  
 android:id="@+id/identifyNearestBankButton"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="380dp"  
 android:layout\_marginEnd="8dp"  
 android:text="@string/identify\_the\_nearest\_bank"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
 <**TextView  
 android:id="@+id/nearestBankTextView"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="460dp"  
 android:layout\_marginEnd="8dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

**input\_array\_size\_dialog.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**EditText  
 android:id="@+id/sizeOfArrayEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_size\_of\_array"  
 android:importantForAutofill="no"  
 android:inputType="number"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
</**LinearLayout**>

**input\_customer\_dialog.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**EditText  
 android:id="@+id/customerIdEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_id"  
 android:importantForAutofill="no"  
 android:inputType="number"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
 <**EditText  
 android:id="@+id/customerSurnameEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_surname"  
 android:importantForAutofill="no"  
 android:inputType="text"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
 <**EditText  
 android:id="@+id/customerNameEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_name"  
 android:importantForAutofill="no"  
 android:inputType="text"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
 <**EditText  
 android:id="@+id/customerMiddlenameEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_middlename"  
 android:importantForAutofill="no"  
 android:inputType="text"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
 <**EditText  
 android:id="@+id/customerAddressEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_address"  
 android:importantForAutofill="no"  
 android:inputType="text"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
 <**EditText  
 android:id="@+id/customerCreditCardNumberEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_credit\_card\_number"  
 android:importantForAutofill="no"  
 android:inputType="number"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
 <**EditText  
 android:id="@+id/customerBankAccountNumberEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_bank\_account\_number"  
 android:importantForAutofill="no"  
 android:inputType="number"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
</**LinearLayout**>

**input\_range\_dialog.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**EditText  
 android:id="@+id/beginEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_begin\_of\_range"  
 android:importantForAutofill="no"  
 android:inputType="number"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
 <**EditText  
 android:id="@+id/endEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/enter\_end\_of\_range"  
 android:importantForAutofill="no"  
 android:inputType="number"  
 tools:ignore="LabelFor"  
 tools:targetApi="o"** />  
  
</**LinearLayout**>

**activity\_first.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/customersRecyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**/>  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

**activity\_second.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/customersRecyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**/>  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

**customer\_item.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="12dp"  
 android:padding="12dp"**>  
  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:text="@string/id"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"** />  
  
 <**TextView  
 android:id="@+id/customerIdTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"**/>  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:text="@string/surname"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"** />  
  
 <**TextView  
 android:id="@+id/customerSurnameTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"**/>  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
 <**TextView  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:text="@string/name"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"** />  
 <**TextView  
 android:id="@+id/customerNameTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"**/>  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
 <**TextView  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:text="@string/middlename"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"** />  
 <**TextView  
 android:id="@+id/customerMiddlenameTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"**/>  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
 <**TextView  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:text="@string/address"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"** />  
 <**TextView  
 android:id="@+id/customerAddressTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"**/>  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
 <**TextView  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:text="@string/credit\_card\_number"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"** />  
 <**TextView  
 android:id="@+id/customerCreditCardNumberTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"**/>  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
 <**TextView  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:text="@string/bank\_account\_number"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"** />  
 <**TextView  
 android:id="@+id/customerBankAccountNumberTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"**/>  
 </**LinearLayout**>  
</**LinearLayout**>