СОДЕРЖАНИЕ

[Введение 4](#_Toc3233859)

[1 Экономическая характеристика объекта 5](#_Toc3233860)

[2 Постановка задачи 6](#_Toc3233861)

[2.1 Экономическое содержание задачи 6](#_Toc3233862)

[2.2 Входная информация 6](#_Toc3233863)

[3 Технические и инструментальные средства для решения задачи 7](#_Toc3233864)

[3.1 Обоснование выбора языка программирования 7](#_Toc3233865)

[3.2 Технические требования к компьютеру для решения задачи 9](#_Toc3233866)

[4 Информационное обеспечение задачи 10](#_Toc3233867)

[4.1 Классификаторы, используемые для решения задачи 10](#_Toc3233868)

[5 Программное обеспечение задачи 13](#_Toc3233869)

[5.1 Описание методов 13](#_Toc3233870)

[5.2 Алгоритм решения задачи 15](#_Toc3233871)

[5.3 Описание блок – схемы 15](#_Toc3233872)

[Заключение 19](#_Toc3233873)

[Список использованных источников 20](#_Toc3233874)

[Приложение А 21](#_Toc3233875)

[Приложение Б 22](#_Toc3233876)

# Введение

Смартфоны давно превзошли по количеству традиционные мобильные телефоны. Рынок современных смартфонов делится на 3 основных операционных системы: Apple iOS, Android и Microsoft Windows Phone. Каждый из производителей определяет собой платформу для разработки мобильных приложений. Рынок мобильных приложений является одним из самых быстрорастущих.

В настоящее же время смартфоны прочно укоренились в нашей повседневной жизни благодаря широкому спектру функций. С помощью смартфонов мы можем не только позвонить, но и выйти в интернет, зайти в социальную сеть, просмотреть или отредактировать какие-либо документы.

Большое предприятие предполагает больше количество различной документации. Большая часть из них хранится в цифровом формате в базах данных. Эти документы можно так же просматривать и редактировать, проследить на каком этапе они находятся, кто сейчас ими занимается при помощи смартфона. Это особенно полезно для инжиниринговой компании «Атоммашэкспорт».

Актуальность темы обусловлена тем, что она позваляет управлять документами дистанционно с помощью смартфона.

Цель курсового проекта заключается в создании мобильного приложения «Документооборот» на базе ОС Android.

В курсовом проекте решаются задачи:

* формирование базы данных;
* фильтрация документов по типу;
* подписание, согласование документов с помощью смартфона.

# 1 Экономическая характеристика объекта

АО «Атоммашэкспорт» было создано на базе производственного объединения атомного энергетического машиностроения «Атоммаш». С 1 июля 1989 года внешнеэкономический отдел ПО «Атоммаш» преобразован в самостоятельную хозрасчетную организацию – внешнеторговую фирму (ВТФ) «Атоммашэкспорт». В 1995 г. предприятие получило статус открытого акционерного общества – ОАО «Атоммашэкспорт». В 2015 году Решением общего собрания акционеров переименовано в акционерное общество - АО «Атоммашэкспорт».

Коллектив компании представлен менеджерами, конструкторами и технологами, имеющими опыт работы в институтах и на производственных предприятиях энергетической отрасли и владеющими современными компьютерными технологиями разработки технической документации. Объединение инженерного и коммерческого опыта, современных компьютерных технологий позволяет нашему предприятию предлагать инжиниринговые работы и услуги в целом ряде отраслей и, в первую очередь, в атомной энергетике.

Инжиниринговая компания «Атоммашэкспорт» выполняет комплексные конструкторские разработки и изготовление оборудования для атомных, тепловых станций и нефтегазохимии. По требованию заказчика возможна поставка оборудования с обеспечением работ по его монтажу, наладке и вводу в эксплуатацию.

Конструкторское подразделение компании состоит из специалистов, имеющих многолетний опыт в проектировании оборудования для АЭС, а также для нефтегазового комплекса [1].

# 2 Постановка задачи

## 2.1 Экономическое содержание задачи

Цель курсового проекта – создание мобильного приложения «Документооборот» на базе ОС Android для АО «Атоммашэкспорт».

Разработанный модуль позволяет производить различные действия над документами с помощью смартфона.

Программа создана на языке Java, графический интерфейс приложения дружественный и понятный, сама программа проста в использовании и не требует специального образования или прохождение курсов.

## 2.2 Входная информация

Входная информация представляет собой документы различного вида, хранящиеся в базе данных.

2.3 Выходная информация

Выходной информацией является уже обновленный документ над которым были произведены какие-либо из доступных действий.

# 3 Технические и инструментальные средства для решения задачи

## 3.1 Обоснование выбора языка программирования

Данное приложение сделано с помощью языка Java – сильно типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре с помощью виртуальной Java-машины [2].

В качестве среды для разработки приложения была выбрана программа Android Studio. Она основана на программном обеспечении IntelliJIDEA от компании JetBrains. Существует множество сред разработки, но данная была выбрана в ввиду ее удобного графического интерфейса и средств отладки. Также основными особенностями Android Studio является возможность вёрстки в реальном времени, доступно множество вариантов размеров и разрешений экранов. Присутствует раздел справки, что намного облегчает работу в среде разработки. Встроены инструменты улучшения качества приложений и монетизации. Имеются инструменты для отслеживания эффективности рекламных объявлений. Добавлено средство взаимодействия с бета-тестерами и много другое. Вдобавок ко всему, новая среда разработки обладает функцией GoogleCloudMessaging, которая позволяет посылать данные с сервера на Android-устройства через облако. Это отличный способ посылать push-уведомления в приложения. Еще одна полезная вещь в Android Studio - инструмент анализа производительности, который называется MemoryMonitor. Он выдает всю информацию об использовании памяти, что дает возможность оптимизировать работу приложения. Android Studio - новая и полностью интегрированная среда разработки приложений, выпущенная компанией Google. Данный продукт призван снабдить разработчиков новыми инструментами для создания приложений, а также предоставить альтернативу устаревшей среде разработки Eclipse. Среда разработки поддерживает систему «Codeinjection», данная система позволяет редактировать фрагменты кода с привязкой к различным языкам. На сайте разработчиков Google отмечает, что планирует регулярно обновлять версию среды разработки [2].

Главные возможности среды разработки Android Studio:

* включены все «интеллектуальные» возможности по редактированию кода;
* есть возможность визуального просмотра будущего приложения;
* сборка проекта работает быстрее, чем в альтернативных средах разработки;
* удобный конструктор интерфейсов;
* встроенный SDK Manager;
* удобное и интуитивно понятное логирование проекта;
* указанные цвета и рисунки, использованные в layout'e отображаются на границе в виде небольших превью, которые легко помогают понять какой конкретно ресурс используется;
* среда разработки является технологиями компании Google;
* при выборе ресурса, его содержимое отображается во всплывающих JavaDoc;
* возможность создания одним кликом layout, сразу привязанным к activity;
* управление локализацией и переводами;
* предпросмотр приложений на разных разрешениях, языках, и версиях API во встроенном эмуляторе;
* мониторинг используемой памяти [3],[4].

## 3.2 Технические требования к компьютеру для решения задачи

Рекомендуемые системные требования к компьютеру на которой будет установлена IDE Android Studio следующие:

* Операционная система – Microsoft® Windows® 7/8/10 (32- или 64-bit);
* оперативная память – 3 ГБ минимум, рекомендуется 8 ГБ;
* пространство на жестком диске – рекомендуется 4 ГБ;
* монитор – 1280 х 800 минимальное разрешение экрана.

# 4 Информационное обеспечение задачи

## 4.1 Классификаторы, используемые для решения задачи

Классификатор представляет собой набор таблиц (справочников).

Объект – некоторая сущность в цифровом пространстве, обладающая определённым состоянием и поведением в данной предметной области.

Атрибут – элемент, описывающий любую характеристику объекта, важную для данной предметной области. Объектом в данной базе данных будет: информация о работнике приведенная в таблице 4.1, документе показанных на таблице 4.2 и местонахождении документа на таблице 4.3.

Таблица 4.1 – Атрибуты работника

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Работник | Идентификационный код  Фамилия  Имя  Отчество  Должность  Телефон  Электронная почта  Пароль |

Таблица 4.2 – Атрибуты документа

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Документ | Идентификационный код  Тип документа  Путь к документу  Дата создания |

Таблица 4.3 – Атрибуты местонахождения документа

|  |  |
| --- | --- |
| Объект | Атрибуты |
| Местонахождение | Идентификационный код документа  Идентификационный код сотрудника  Статус документа  Дата и время завершения |

Для выполнения проекта была выбрана СУБД SQL Server. Схема БД представлена на рисунке 4.1.

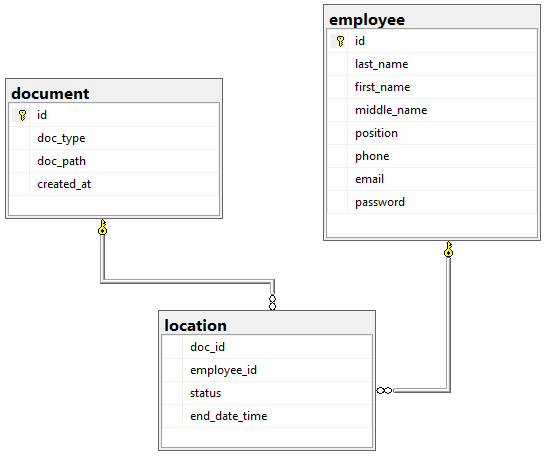


Рисунок 4.1 – Связи таблиц в БД

Структура таблиц документов, местонахождения и работников базы данных представлена на рисунках 4.2 – 4.4 соответственно.

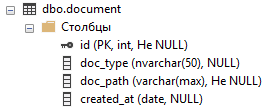


Рисунок 4.2 – Таблица «Документ»

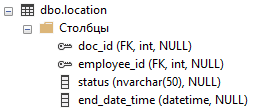


Рисунок 4.3 – Таблица «Местонахождение»

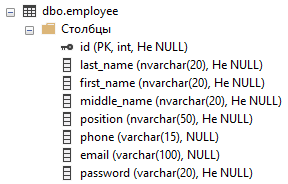


Рисунок 4.4 – Таблица «Работник»

# 5 Программное обеспечение задачи

## 5.1 Описание методов

Для примера взята часть кода для входа в приложение под своей учетной записью. При нажатии на кнопку вызывается активити, для дальнейшей работы с приложением.

package ru.atomexp.android;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.Toast;  
import com.google.gson.Gson;  
import com.google.gson.GsonBuilder;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
  
public class LoginActivity extends AppCompatActivity {  
  
 private Button btnSignIn;  
 private ImageButton btnQrCode;  
 private EditText etLogin;  
 private EditText etPass;  
 private User user;  
 private Gson GSON;  
 public static final String *APP\_PREFERENCES* = "authorizedUser";  
 public static final String *SAVED\_USER* = "savedUser";  
 public static SharedPreferences *spUser*;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
  
 *spUser* = getSharedPreferences(*APP\_PREFERENCES*, *MODE\_PRIVATE*);  
 GSON = new GsonBuilder().create();  
  
 btnSignIn = findViewById(R.id.*signInButton*);  
 btnQrCode = findViewById(R.id.*qrCodeButton*);  
 etLogin = findViewById(R.id.*loginText*);  
 etPass = findViewById(R.id.*passwordText*);  
 }  
  
 public void signIn(View view) {  
 Toast.*makeText*(this, "Нажатие", Toast.*LENGTH\_SHORT*).show();  
 try {  
 try {  
 Class.*forName*("net.sourceforge.jtds.jdbc.Driver");  
 Toast.*makeText*(this, "Зарегистрирован", Toast.*LENGTH\_LONG*).show();  
 } catch(Exception ex){  
 Toast.*makeText*(this, "Не зарегистрирован" + ex.getMessage(), Toast.*LENGTH\_LONG*).show(); }  
 final String DB\_URL = "jdbc:jtds:sqlserver://192.168.31.80:1433/documents";  
 final String DB\_USER = "p1onerw0w";  
 final String DB\_PASSWORD = "4321193";  
 final Gson GSON = new GsonBuilder().create();  
 String login = etLogin.getText().toString();  
 String password = etPass.getText().toString();  
 String query =  
 "select \* from users " +  
 "where (phone = '" + login + "' or email = '" + login + "') " +  
 "and [password] = '" + password + "'";  
 try {  
 Connection connection = DriverManager.*getConnection*(DB\_URL, DB\_USER, DB\_PASSWORD);  
 Toast.*makeText*(this, "Подключение создано", Toast.*LENGTH\_LONG*).show();  
 } catch (Exception ex) {  
 Toast.*makeText*(this, "Не подключен" + ex.getLocalizedMessage(), Toast.*LENGTH\_LONG*).show();  
 }  
 if(!*spUser*.contains(*SAVED\_USER*)) {  
 try {  
 user = User.*userBuild*(SqlServerService.*getResultSet*(query));  
  
 String gsonUser = GSON.toJson(user);  
  
 SharedPreferences.Editor editor = *spUser*.edit();  
 editor.putString(*SAVED\_USER*, gsonUser);  
 editor.apply();  
  
 Intent intent = new Intent(this, MainActivity.class);  
 intent.putExtra("user", gsonUser);  
 Toast.*makeText*(this, "Авторизация", Toast.*LENGTH\_SHORT*).show();  
 startActivity(intent);  
  
 } catch(SQLException sqlex) {  
 Toast.*makeText*(this, "Ошибка SQL", Toast.*LENGTH\_LONG*).show(); } catch(Exception ex) {  
 Toast.*makeText*(this, "NON-SQL ошибка", Toast.*LENGTH\_LONG*).show();  
 }  
  
 } else {  
  
 }  
 } catch(Exception ex) {   
 Log.*e*(this.getClass().getSimpleName(), "Получено исключение", ex);  
 }  
 }  
}

## 5.2 Алгоритм решения задачи

Для проведения каких-либо действий с документами сначала необходимо выбрать раздел: подписать, согласовать, испольнить, ознакомиться или проверить документ.

## 5.3 Описание блок – схемы

Последовательное описание блок – схемы приложения приведено ниже:

Шаг 1. Запуск программы.

Шаг 2. Вопрос: Верно ли указан логин и пароль? Если да, то переходим к следующему шагу, иначе остаемся на 2 шаге.

Шаг 3. Вход в приложение под своей учетной записью.

Шаг 4. Вопрос: Выбрано ли действие? Если да, то переходим к шагу 5, если нет, то переходим к 7 шагу.

Шаг 5. Выполнение

Шаг 6. Возвращаемся к 4 шагу.

Шаг 7. Закрытие приложения.

Блок-схема приведена в приложении А.

Листинг программы приведен в приложении Б.

5.4 Описание интерфейса программы

В активити входа в приложение, представленном на рисунке 5.1, находятся два текстовых поля логин и пароль, кнопка авторизации. Для дальнейшей работы нужно верно указать логин и пароль. Если логин и пароль верны, то откроется основное активити, представленное на рисунке 5.2 с возможностью выбора действия над документами. При нажатии на кнопку происходит открытие нужного нам активити, которая показана на рисунке 5.3.



Рисунок 5.1 – Активити входа

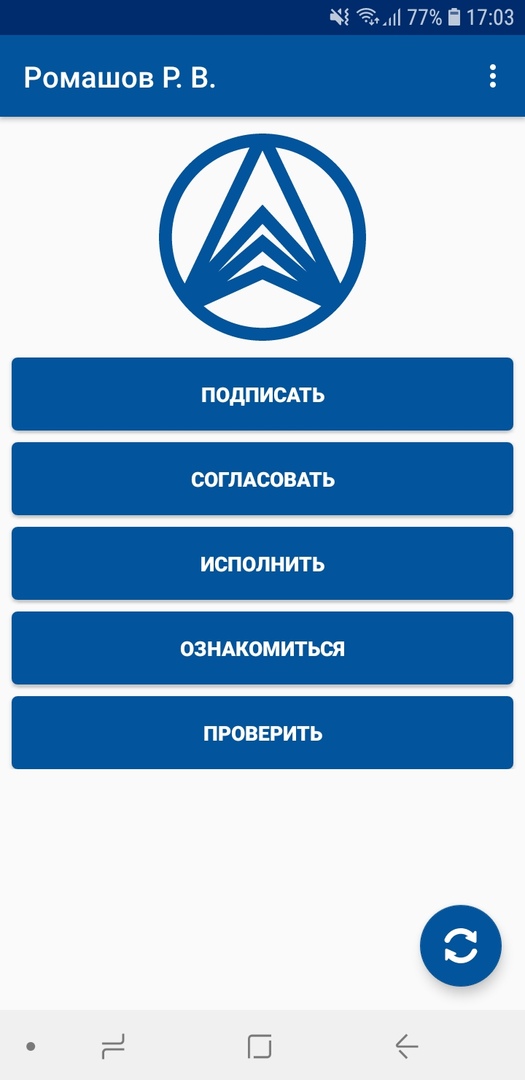


Рисунок 5.2 – Основное активити

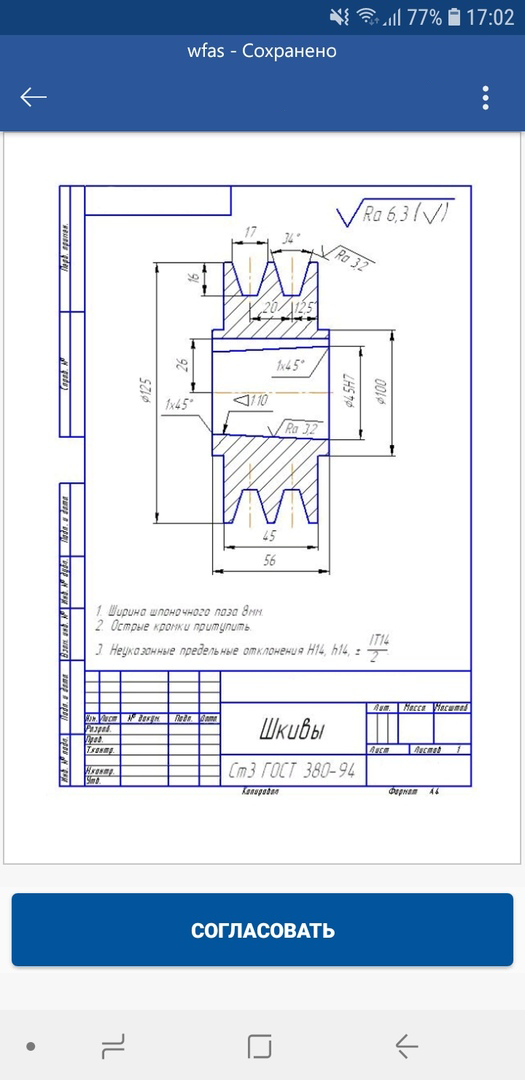


Рисунок 5.3 – Активити согласования

Для успешной работы с программой необходимо:

1. Версия Android 4.0 и выше.
2. Подключение к интернету.
3. Учетная запись.

# Заключение

Разработанное мобильное приложение документооборота для АО “Атоммашэкспорт” представляет собой простое приложение с лаконичным дизайном в стиле Material Design.

Программа разработана на языке Java в интегрированной среде разработки Android Studio с использованием СУБД SQL Server. Приложение интуитивно понятно даже неопытному пользователю и не затрудняет его работу с документами с помощью смартфона

В дальнейшем в приложение будет обновляться и будут добавляться новые функции необходимые для работы с документами.

В процессе написания кода для данной программы были приобретены практические навыки в работе в СУБД SQL Server и языке программирования Java.

Цель курсового проекта достигнута. Задачи выполнены.

# Список использованных источников

Интернет ресурсы:

* 1. АО “Атоммашэкспорт”. О компании [Электронный ресурс] – Режим доступа: http://www.atomexp.ru/page/about
  2. Википедия. Язык программирования Java [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Java
  3. Android Developers. Android Studio [Электронный ресурс] – Режим доступа: http://developer.android.com
  4. Habr. Российский блог информационных технологий [Электронный ресурс] – Режим доступа: https://habr.com/ru/

# Приложение А

(обязательное)

Блок – схема программы

Начало

Запуск программы

Введены верные логин и пароль?

Нет

Да

Вход в приложение

Нет

Выбрано действие?

Закрытие

Выполнение

Да

Сохранение документа

Рисунок А.1–Блок схема программы

# Приложение Б

(обязательное)

Листинг программы

create database documents

go

use documents

go

create table employee

(

id int identity primary key,

last\_name nvarchar(20) not null default '',

first\_name nvarchar(20) not null default '',

middle\_name nvarchar(20) not null default '',

position nvarchar(50) not null default '',

phone varchar(15) constraint ck\_users\_phone check (phone like '[8][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),

email varchar(100) constraint ck\_users\_email check (email like '%[@]%'),

[password] varchar(20) not null,

constraint uq\_employee\_phone unique (phone),

constraint uq\_employee\_email unique (email)

)

go

create table document

(

id int identity primary key,

doc\_type nvarchar(50) constraint ck\_document\_doc\_type check (doc\_type IN('Чертеж','Договор','Соглашение','Отчет')),

doc\_path varchar(max) not null,

created\_at date default getdate()

)

go

create table [location]

(

doc\_id int constraint fk\_location\_document foreign key references document(id),

employee\_id int constraint fk\_location\_employee foreign key references employee(id),

[status] nvarchar(50) constraint ck\_location\_status check ([status] IN('Ожидание','В работе','Завершен')),

end\_date\_time datetime null

)

go

package ru.atomexp.android;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.Toast;  
  
import com.google.gson.Gson;  
import com.google.gson.GsonBuilder;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
  
public class LoginActivity extends AppCompatActivity {  
  
 private Button btnSignIn;  
 private ImageButton btnQrCode;  
 private EditText etLogin;  
 private EditText etPass;  
 private User user;  
 private Gson GSON;  
 public static final String *APP\_PREFERENCES* = "authorizedUser";  
 public static final String *SAVED\_USER* = "savedUser";  
 public static SharedPreferences *spUser*;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
 *spUser* = getSharedPreferences(*APP\_PREFERENCES*, *MODE\_PRIVATE*);  
 GSON = new GsonBuilder().create();  
 btnSignIn = findViewById(R.id.*signInButton*);  
 btnQrCode = findViewById(R.id.*qrCodeButton*);  
 etLogin = findViewById(R.id.*loginText*);  
 etPass = findViewById(R.id.*passwordText*);  
 }  
  
 public void signIn(View view) {  
 Toast.*makeText*(this, "Нажатие", Toast.*LENGTH\_SHORT*).show();  
 try {  
 try {  
 Class.*forName*("net.sourceforge.jtds.jdbc.Driver");  
 Toast.*makeText*(this, "Зарегистрирован", Toast.*LENGTH\_LONG*).show();  
 } catch(Exception ex){  
 Toast.*makeText*(this, "Не зарегистрирован" + ex.getMessage(), Toast.*LENGTH\_LONG*).show(); }  
 final String DB\_URL = "jdbc:jtds:sqlserver://192.168.31.80:1433/documents";  
 final String DB\_USER = "p1onerw0w";  
 final String DB\_PASSWORD = "4321193";  
 final Gson GSON = new GsonBuilder().create();  
 String login = etLogin.getText().toString();  
 String password = etPass.getText().toString();  
 String query =  
 "select \* from users " +  
 "where (phone = '" + login + "' or email = '" + login + "') " +  
 "and [password] = '" + password + "'";  
 try {  
 Connection connection = DriverManager.*getConnection*(DB\_URL, DB\_USER, DB\_PASSWORD);  
 Toast.*makeText*(this, "Подключение создано", Toast.*LENGTH\_LONG*).show();  
 } catch (Exception ex) {  
 Toast.*makeText*(this, "Не подключен" + ex.getLocalizedMessage(), Toast.*LENGTH\_LONG*).show();  
 }  
 if(!*spUser*.contains(*SAVED\_USER*)) {  
 try {  
 user = User.*userBuild*(SqlServerService.*getResultSet*(query));  
 String gsonUser = GSON.toJson(user);  
  
 SharedPreferences.Editor editor = *spUser*.edit();  
 editor.putString(*SAVED\_USER*, gsonUser);  
 editor.apply();  
 Intent intent = new Intent(this, MainActivity.class);  
 intent.putExtra("user", gsonUser);  
 Toast.*makeText*(this, "Авторизация", Toast.*LENGTH\_SHORT*).show();  
 startActivity(intent);  
 } catch(SQLException sqlex) {  
 Toast.*makeText*(this, "Ошибка SQL", Toast.*LENGTH\_LONG*).show(); } catch(Exception ex) {  
 Toast.*makeText*(this, "Иная ошибка", Toast.*LENGTH\_LONG*).show();  
 }  
 }   
 } catch(Exception ex) {  
 Toast.*makeText*(this, "Снаружи " + ex.getMessage(), Toast.*LENGTH\_LONG*).show();  
 Log.*e*(this.getClass().getSimpleName(), "Получено исключение", ex);  
 }  
 }  
}

package ru.atomexp.android;  
  
import android.os.Bundle;  
import android.support.design.widget.NavigationView;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.widget.TextView;  
import android.widget.Toast;  
import com.google.gson.Gson;  
import com.google.gson.GsonBuilder;  
import com.google.gson.JsonObject;  
import org.json.JSONObject;  
  
public class MainActivity extends AppCompatActivity  
 implements NavigationView.OnNavigationItemSelectedListener {  
 private TextView navHeaderTitle;  
 private TextView navHeaderSubTitle;  
 private User thisUser;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 Toolbar toolbar = findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
 DrawerLayout drawer = findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
 this, drawer, toolbar, R.string.*navigation\_drawer\_open*, R.string.*navigation\_drawer\_close*);  
 drawer.addDrawerListener(toggle);  
 toggle.syncState();  
 NavigationView navigationView = findViewById(R.id.*nav\_view*);  
 navigationView.setNavigationItemSelectedListener(this);  
 if(savedInstanceState == null) { navigationView.setCheckedItem(R.id.*m\_messages*);  
 getSupportActionBar().setTitle(R.string.*messages*);  
 }  
 navHeaderTitle = findViewById(R.id.*headerTitle*);  
 navHeaderSubTitle = findViewById(R.id.*headerSubTitle*);  
 Gson GSON = new GsonBuilder().create();  
 thisUser = GSON.fromJson((String)getIntent().getSerializableExtra("user"), User.class);  
 String headerTitle = thisUser.getLastName() + thisUser.getFirstName();  
 navHeaderTitle.setText(headerTitle);  
 String headerSubTitle = thisUser.getPosition();  
 navHeaderSubTitle.setText(headerSubTitle);  
 }  
  
 @Override  
 public void onBackPressed() {  
 DrawerLayout drawer = findViewById(R.id.*drawer\_layout*);  
 if (drawer.isDrawerOpen(GravityCompat.*START*)) {  
 drawer.closeDrawer(GravityCompat.*START*);  
 } else {  
 *super.onBackPressed();* }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) { getMenuInflater().inflate(R.menu.*main*, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) { int id = item.getItemId(); if (id == R.id.*action\_settings*) {  
 Toast.*makeText*(MainActivity.this, "Settings", Toast.*LENGTH\_LONG*).show();  
 }  
 return super.onOptionsItemSelected(item);  
 }