

Пловдивски Университет „Паисий
Хилендарски“
Факултет по математика и информатика

Курсова Работа

По дисциплина : „Мобилни приложения“

На тема : „Мобилно приложение JOKE“

Изготвил Симе Янкулов
Специалност
Софтуерно инженерство
Факултетен номер 2001321092

Проверил
доц. Д-р С. Хаджиколева

Съдържание

1.Увод.....	3
2.Основни Функционалности	3
3.Използвани технологии и библиотеки	3
4.Потребителски инструктаж	4
5.Архитектура на приложението	6
6.Имплементация	6
7.Заключение	10
8.Библиография.....	11
9.Списък на фигурите	11

1.Увод

Целта на курсовия проект е да се изгради приложение в което може да се регистрират потребители, след което същите потребители да могат да използват основната функционалност на приложението която ще е извеждане на някаква информация от интернета по случаен избор.

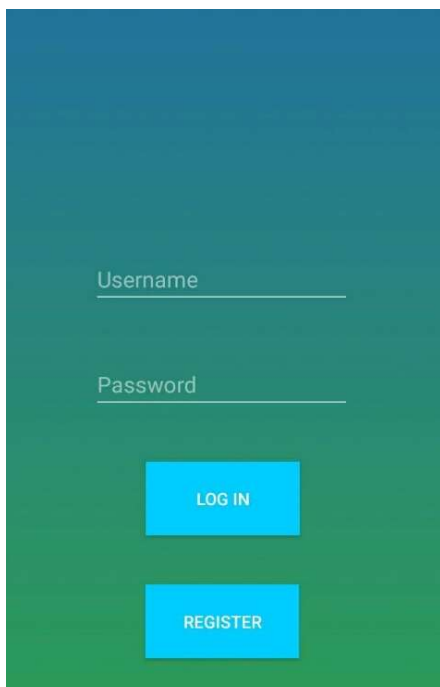
2.Основни Функционалности

- Регистриране на потребител
- Функционалност за логване на потребител
- Редактиране на потребител
- Изтриване на потребител
- Извеждане на информация чрез API

3.Използвани технологии и библиотеки

- Android Studio
- Java
- Android SDK
- GSON [1] - Java библиотека с отворен код за сериализиране и десериализиране на Java обекти в JSON. Използван при извеждането на информацията от API. [1]
- SQLite [2]– Релационна база от данни с отворен код поддържаща стандарта SQL.Използвана за запазване, редактиране и изтриване на потребители. [2]
- Regex [3]– Модел за търсене, съвпадение на шаблон или съвпадение на низ, използван за контрол на информация за потребителя. [3]
- Volley [4]– HTTP библиотека, която прави работата в мрежа за мобилните приложения по-лесна и по-бърза. Използвана е при комуникиране със сървър. [4]

4. Потребителски инструктаж

The image shows a login screen with a green-to-blue gradient background. It features two input fields: 'Username' and 'Password', both with light blue borders. Below the 'Password' field is a red 'LOG IN' button. Further down is a red 'REGISTER' button. The text is in a clean, sans-serif font.

Фигура 1 ГПИ Login Activity

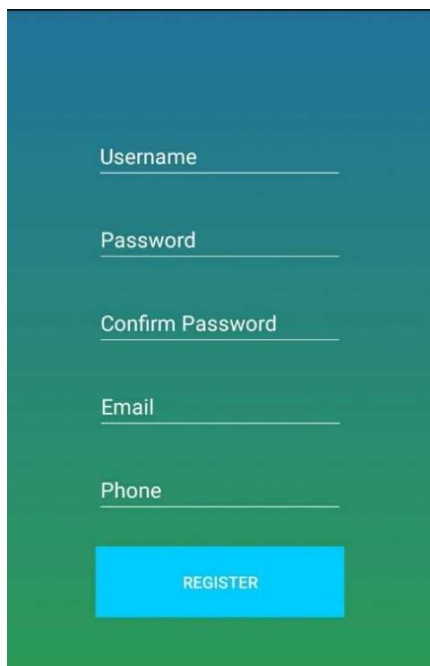
Login Activity

За потребителя да използва приложението трябва да има акаунт на приложението така че при стартиране това е първият екран който вижда потребителя. Това е доста прост входен екран който изисква от потребителя да внесе точна двойка от потребителско име и парола.

Като първо не е възможно логване с празни полета.
Това се проверява в този код.

След подпълване на полетата и натискане на бутона "LOG IN" се извършва проверка в базата [2] за дали такъв потребител съществува. Ако съществува, се стартира втория екран (Фигура 3), ако не се извежда информация че са внесени невалидни входи

При натискане на бутона „REGISTER“ потребителя се пренасочва към екрана за регистрация (Фигура 2).

The image shows a registration screen with a green-to-blue gradient background. It features five input fields: 'Username', 'Password', 'Confirm Password', 'Email', and 'Phone', all with light blue borders. At the bottom is a red 'REGISTER' button. The text is in a clean, sans-serif font.

Фигура 2 ГПИ Register Activity

Register Activity

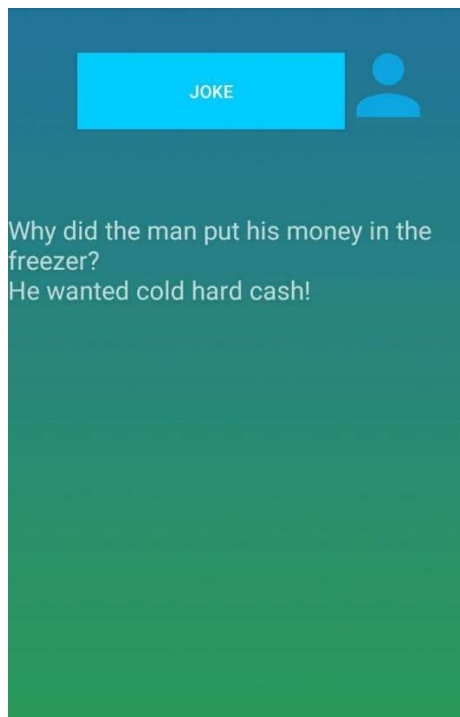
В този екран се регистрира нов потребител в базата [2]. Не е възможна регистрация с празни полета.

При самото попълване на полетата се правят regex [3] валидации, които не позволяват потребителя да внесе невалидна информация или слаба парола.

В потребителското име няма ограничения, стига да е уникално.

Паролата и потвърждението на паролата мора да са същи и те мора да са по-дълги от 8 знака, да имат поне една голяма буква, една малка буква, един знак и един номер.

Имейла и телефонния номер трябва да са уникални.



Фигура 3 ГПИ Main Activity

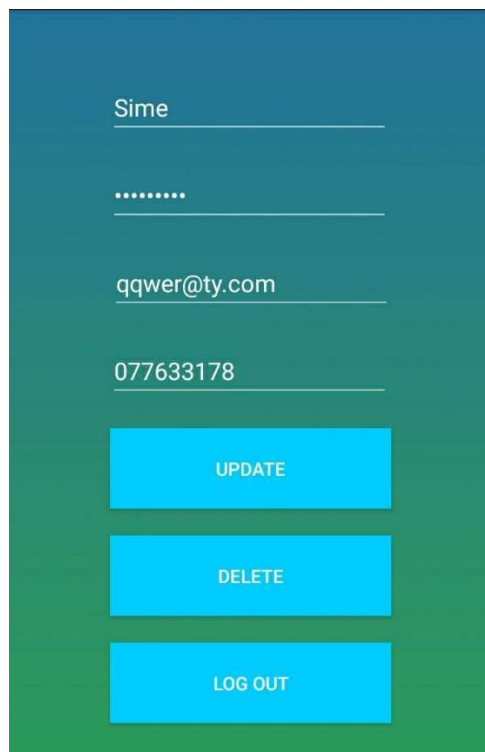
Main Activity

След като потребителя успешно се логвал или регистрирал в приложението, той ще се намира на този екран.

Съставен от няколко елемента , два бутона и текстово поле.

При натискане на бутона или иконката „Person“ или акаунт да я наречем потребителя се пренасочва на друг екран (*фигура 4*).

При натискане на бутона “JOKE” приложението използва външно API [5], който ни връща JSON обект [1] които е съставен от няколко полета. Тоя обект се десериализира и от него се взимат две полета и те се извеждат на екрана. Обектите са “setup” и “punchline” който заедно образуват някаква шега. Всяко натискане на бутона ни връща нещо различно , базирано на случаен избор.



Фигура 4 ГПИ Account Activity

Account Activity

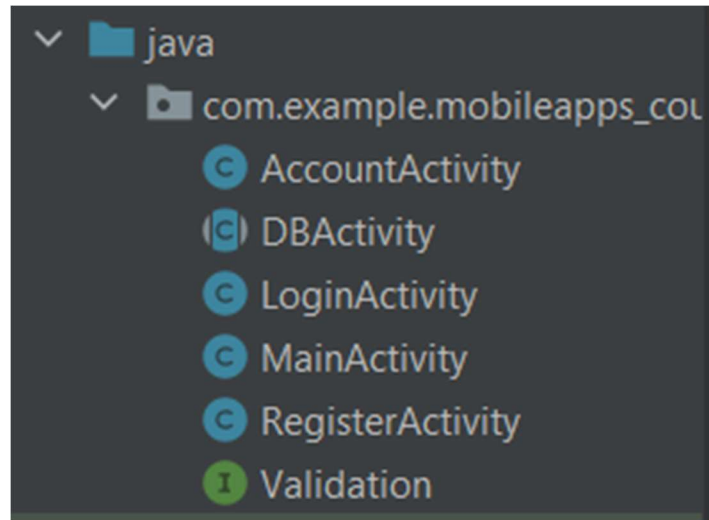
Тук потребителя може да си види информацията си.

При промяна на някакво поле и натискане на бутона “Update”, потребителя може да си промени данните. Тук също така се проверява текста дали е валиден.

С “Delete” бутона изтриваме потребителя и се връщаме на началният екран(*Фигура 1*).

И „Login“ бутона, които е изход от приложението, пренасочва потребителя към началният екран.

5. Архитектура на приложението



Фигура 5 Архитектура на приложението

- Account Activity – екрана представен на фигура 4.
- Login Activity – екрана представен на фигура 1.
- Register Activity – екрана представен на фигура 2.
- Main activity – екрана представен на фигура 3, тук също така е комуникацията с API [5].
- Validation – интерфейс в който се правят Regex [3] валидациите за вход и регистрация на потребител.
- DBActivity – абстрактен клас в който се извършва комуникацията с база от данни [2] .

6. Имплементация

1. Бутон “LOG IN” - LoginActivity

- Се проверява за празни текстови полета и се проверява в базата дали е намерен същ потребител за след това да минем към MainActivity

```
if(editUsername.getText().length() > 0 &
editPassword.getText().length() >0) {
    SelectSQL("Select * from User where Username = ? and Password =
    ?",
        new String[]{editUsername.getText().toString()
            , editPassword.getText().toString() },
        (ID, Username, Password, Phone, Email) ->
            Func(ID)
```

Фигура 6 Проверка за вход

2. Бутон „Register“ – RegisterActivity

- Вмъкване на нов потребител в базата

```
ExecSql(
    "INSERT INTO USER(Username, Password,Tel, Email) " +
    "VALUES(?, ?, ?, ?)",
    new Object[]{
        editRUsername.getText().toString(),
        editRPassword.getText().toString(),
        editPhone.getText().toString(),
        editEmail.getText().toString()
    },
    ()-> Toast.makeText(getApplicationContext(),
        "Register Successful",
        Toast.LENGTH_LONG
    ).show()
)
```

Фигура 7 Insert

3. Извличане на данни за потребител - Account Activity

```
SelectSQL(
    "Select * from User where ID="+getLoggedid(),
    null,
    (id,Username,Password,Tel,Email)-> {
        editAUsername.setText(Username);
        editAPassword.setText(Password);
        editAPhone.setText(Tel);
        editAEmail.setText(Email);
    }
)
```

Фигура 8 Select

4. Премахване на потребител – Account Activity

```
ExecSql(
    "DELETE FROM User " +
    "WHERE ID = "+getLoggedid(),
    null,
    ()->Toast.makeText(getApplicationContext(),
        "DELETE SUCCEEDED", Toast.LENGTH_LONG)
    .show()
);
```

Фигура 9 Delete

5. Актуализиране на потребител – Account Activity

```
ExecSql (
    "UPDATE User SET " +
        "Username = ?, " +
        "Password = ?, "+
        "Tel = ?, " +
        "Email = ? " +
        "WHERE ID = ?",
    new Object[]{
        editAUsername.getText().toString(),
        editAPassword.getText().toString(),
        editAPhone.getText().toString(),
        editAEmail.getText().toString(),
        getLoggedid()
    },
    ()->Toast.makeText(getApplicationContext(),
        "UPDATE SUCCEEDED", Toast.LENGTH_LONG)
        .show()
);
```

Фигура 10 Update

6. Показване на данни от API – MainActivity

```
private void jsonParse(){
    String url = "https://official-joke-api.appspot.com/random_joke";

    JSONObjectRequest request = new
    JSONObjectRequest(Request.Method.GET, url, null,
        new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {
                try {
                    textVJ.setText("");
                    textVJ.append(response.getString("setup")+"\n");
                    textVJ.append(response.getString("punchline"));

                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                error.printStackTrace();
            }
        });
    mqueue.add(request);
}
```

Фигура 11 JSON Parse

7. Валидация на текстови полета – интерфейс Validation

```
if (
    MatchString(
        "^(\\((\\+|00)?\\d+\\))|(\\+|00))? ?\\d+([\\/- -
    ]?(\\(\\d+\\)|\\d+))+(:\\d+)?$",
        editPhone.getText().toString()
    )
    &&
    MatchString("^[\\w\\.]+@([\\w\\.]+[a-zA-Z]{2,4})$",
        editEmail.getText().toString()
    )
    &&
    MatchString("(?=.*[0-9])(?=.*[a-z])(?=.*[A-
    Z])(?=.*[!@#&()-[{ }];',?/*~$^+=<>]).{8,20}$",
        editPassword.getText().toString()
    )
    &&
    editPassword.getText().toString().equals(editConfirmPassword.getText().toString())
){
    success.MatchSuccess();
}else{
    fail.MatchFail();
}
```

Фигура 12 Regex Validations

8. Създаване на таблица потребители и инициализация на базата

```
protected void InitDB() throws Exception {
    ExecSql(
        "CREATE TABLE if not exists USER(" +
        "ID integer not null PRIMARY KEY AUTOINCREMENT," +
        "Username text not null," +
        "Password text not null," +
        "Tel text not null," +
        "Email text not null," +
        "unique(Email)," +
        "unique (Username,Tel)" +
        ");",
        null,
        () -> Toast.makeText(getApplicationContext(),
            "DB INIT SUCCESSFUL", Toast.LENGTH_LONG).show());
}
```

Фигура 13 INITDB

9. Търсене в таблицата потребители ,в базата

```
protected void SelectSQL(String SelectQ, String[] args ,
OnSelectSuccess success)
    throws Exception
{
    SQLiteDatabase db;
    db = SQLiteDatabase.openOrCreateDatabase(
        getFilesDir().getPath() + "/user.db", null);

    Cursor cursor = db.rawQuery(SelectQ, args);

    while(cursor.moveToNext()) {
        @SuppressWarnings("Range") String ID =
        cursor.getString(cursor.getColumnIndex("ID"));
        setLoggedid(Integer.parseInt(ID));
        @SuppressWarnings("Range") String Username =
        cursor.getString(cursor.getColumnIndex("Username"));
        @SuppressWarnings("Range") String Password =
        cursor.getString(cursor.getColumnIndex("Password"));
        @SuppressWarnings("Range") String Tel =
        cursor.getString(cursor.getColumnIndex("Tel"));
        @SuppressWarnings("Range") String Email =
        cursor.getString(cursor.getColumnIndex("Email"));
        success.OnElementSelected(ID, Username, Password, Tel, Email);
    }
    db.close();
}
```

Фигура 14 Select

7.Заклучение

Целта на този проект ми беше да направя приложение което ще има добър потребителски интерфейс, приложението да работи с база данни и също така да комуникира с някой сървър.

Потребителският интерфейс е доста прост , и интуитивен доволен съм от това което съм направил но бих добавил още няколко неща като повече информационни съобщения.

Функционалността за регистрация на потребители и вход , ми задаваше проблеми обаче се справих добре и направих всичко това да работи както трябва, като цяло работи и заедно с валидациите и потребителското удостоверяване.

Комуникацията с сървъра е направена обаче не съм доволен от функционалността като цяло може да се подобри с това да извеждаме нещо друго , с някакви методи за търсене или нещо подобно.

Горе-долу става въпрос за едноставно приложение което си работи както трябва но с малки подобрения може да се пренасочи в било какво приложение.

8. Библиография

- [1] Google, „GSON Serialization Deserialization,“ [Онлайн]. Available: <https://github.com/google/gson>.
- [2] D. R. Hipp, „SQLite,“ [Онлайн]. Available: <https://www.sqlite.org/index.html>.
- [3] gskinner.com, „Regex101,“ [Онлайн]. Available: <https://regex101.com/>.
- [4] Google, „Volley,“ [Онлайн]. Available: <https://google.github.io/volley/>.
- [5] CodingInFlow, „Youtube,“ [Онлайн]. Available: <https://www.youtube.com/watch?v=y2xtLqP8dSQ>.
- [6] Public-Apis, „GitHub,“ [Онлайн]. Available: <https://github.com/public-apis/public-apis>.
- [7] Curious Concept, „JSON Formatter&Validator,“ [Онлайн]. Available: <https://jsonformatter.curiousconcept.com/#>.
- [8] Moosavi, „StackOverFlow,“ [Онлайн]. Available: <https://stackoverflow.com/questions/34691175/how-to-send-httprequest-and-get-json-response-in-android>.
- [9] Unknown, „Public API,“ [Онлайн]. Available: https://official-joke-api.appspot.com/random_joke.
- [10] S. Yankulov, „GitHub,“ [Онлайн]. Available: <https://github.com/SimeYankulov/MobileApplicationCourseWork>.

9. Списък на фигурите

Фигура 1 ГПИ Login Activity	4
Фигура 2 ГПИ Register Activity	4
Фигура 3 ГПИ Main Activity	5
Фигура 4 ГПИ Account Activity	5
Фигура 5 Архитектура на приложението	6
Фигура 6 Проверка за вход	6
Фигура 7 Insert	7
Фигура 8 Select	7
Фигура 9 Delete	7
Фигура 10 Update	8
Фигура 11 JSON Parse	8
Фигура 12 Regex Validations	9
Фигура 13 INITDB	9
Фигура 14 Select	10

