



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«МИРЭА – Российский технологический университет»  
**РТУ МИРЭА**

---

**Институт информационных технологий (ИИТ)**  
**Кафедра Математического обеспечения и стандартизации информационных технологий**

Отчет по выполнению практического задания №7

**Тема:**  
**Взаимодействие с другими приложениями**

**Дисциплина:**  
**Разработка мобильных приложений**

Выполнил:  
Студент группы ИКБО-03-20

Цемкало А.Р.

Принял:  
Доцент кафедры МОСИТ ИИТ

Чернов Е. А.

Москва 2022 г.

## СОДЕРЖАНИЕ

Ход работы.....	3
Хранение наборов ключ-значение.....	3
Хранение файлов.....	<b>Ошибка! Закладка не определена.</b>
Хранение данных в базе данных SQLite.....	<b>Ошибка! Закладка не определена.</b>
Вывод.....	12

## Ход работы

### Создание неявного намерения

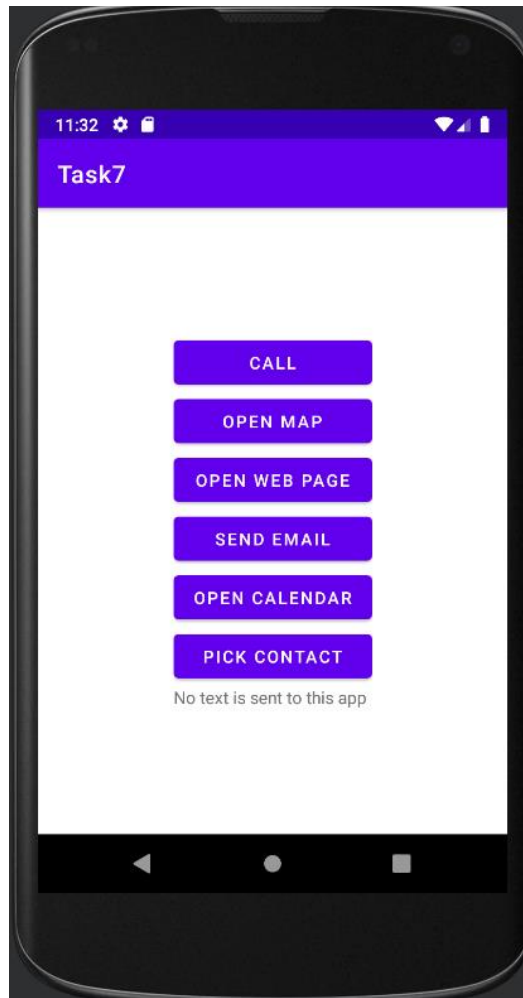


Рисунок 1 – вид приложения до нажатия на кнопки

```
public void call() {  
    Uri number = Uri.parse("tel:5551234");  
    Intent callIntent = new Intent(Intent.ACTION_DIAL, number);  
    if (checkIntent(callIntent)) {  
        startActivity(callIntent);  
    }  
}
```

Рисунок 2 – метод вызова по номеру телефона

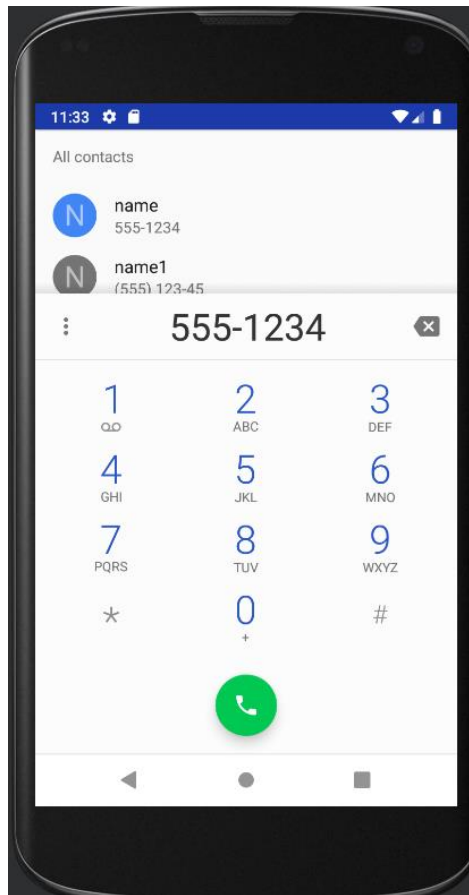


Рисунок 3 – вид смартфона после нажатия на кнопку CALL

```
public void openMap() {  
    Uri location = Uri.parse("geo:0,0?q=1600+Amphitheatre+Parkway,+Mountain+View,+California");  
    Intent mapIntent = new Intent(Intent.ACTION_VIEW, location);  
  
    if (checkIntent(mapIntent)) {  
        startActivity(mapIntent);  
    }  
}
```

Рисунок 4 – метод просмотра карты

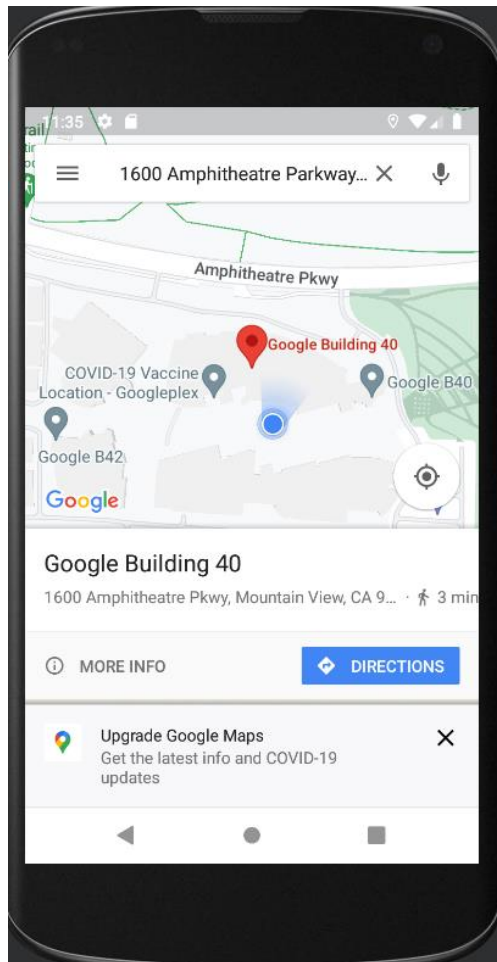


Рисунок 5 – вид смартфона после нажатия на кнопку OPEN MAP

```
public void openWebPage() {  
    Uri webpage = Uri.parse("https://ru.wikipedia.org/wiki/Заглавная_страница");  
    Intent webIntent = new Intent(Intent.ACTION_VIEW, webpage);  
    if (checkIntent(webIntent)) {  
        startActivity(webIntent);  
    }  
}
```

Рисунок 6 – метод просмотра web-страницы

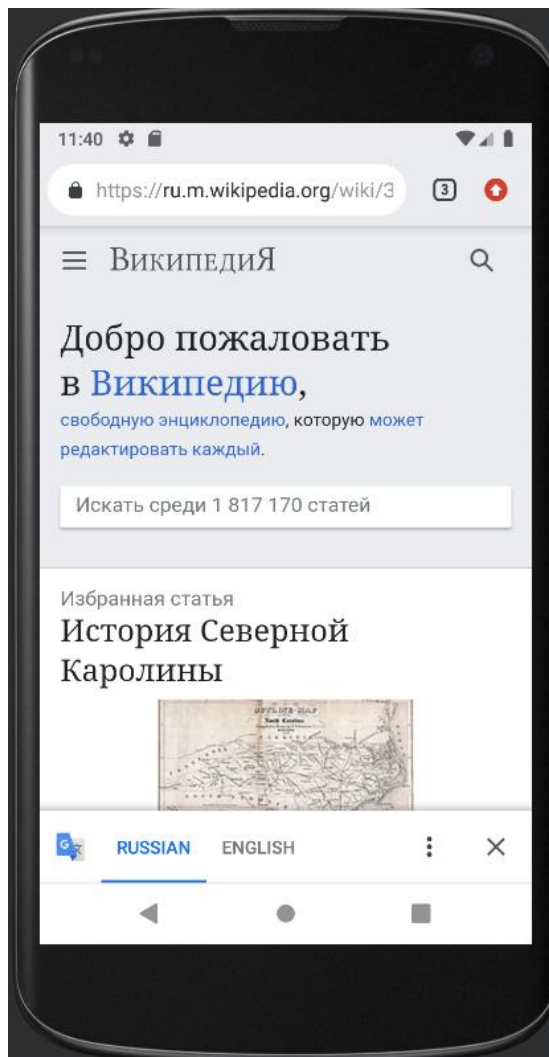


Рисунок 7 – вид смартфона после нажатия на кнопку OPEN WEB PAGE

```
public void sendMail() {  
    Intent emailIntent = new Intent(Intent.ACTION_SEND);  
    emailIntent.setType("text/plain");  
    emailIntent.putExtra(Intent.EXTRA_EMAIL, new String[] {"tsemkaloalena@gmail.com"});  
    emailIntent.putExtra(Intent.EXTRA_SUBJECT, value: "Email subject");  
    emailIntent.putExtra(Intent.EXTRA_TEXT, value: "Email message text");  
    emailIntent.putExtra(Intent.EXTRA_STREAM, Uri.parse("content://path/to/email/attachment"));  
  
    String title = "Send email";  
    Intent chooser = Intent.createChooser(emailIntent, title);  
    if (emailIntent.resolveActivity(getPackageManager()) != null) {  
        startActivity(chooser);  
    }  
}
```

Рисунок 8 – метод отправки письма с вложением

```

public void createCalendar() {
    Intent calendarIntent = new Intent(Intent.ACTION_INSERT, CalendarContract.Events.CONTENT_URI);
    Calendar beginTime = Calendar.getInstance();
    Calendar endTime = Calendar.getInstance();
    beginTime.set( year: 2012, month: 0, date: 19, hourOfDay: 7, minute: 30);
    endTime.set( year: 2012, month: 0, date: 19, hourOfDay: 10, minute: 30);
    calendarIntent.putExtra(CalendarContract.EXTRA_EVENT_BEGIN_TIME, beginTime.getTimeInMillis());
    calendarIntent.putExtra(CalendarContract.EXTRA_EVENT_END_TIME, endTime.getTimeInMillis());
    calendarIntent.putExtra(CalendarContract.Events.TITLE, value: "Ninja class");
    calendarIntent.putExtra(CalendarContract.Events.EVENT_LOCATION, value: "Secret dojo");
    if (checkIntent(calendarIntent)) {
        startActivity(calendarIntent);
    }
}
}

```

Рисунок 10 – метод создания события в календаре

```

public Boolean checkIntent(Intent intent) {
    PackageManager packageManager = getPackageManager();
    List<ResolveInfo> activities = packageManager.queryIntentActivities(intent, 0);
    return activities.size() > 0;
}

```

Рисунок 11 – метод проверки наличия приложения для получения намерения

### Показ диалога выбора приложений

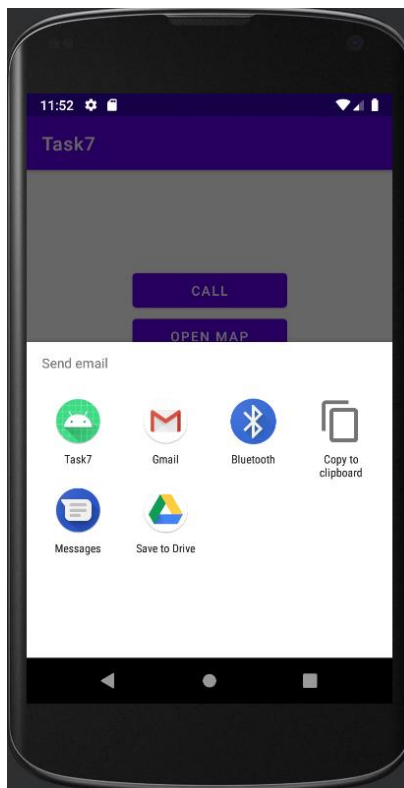


Рисунок 12 – вид смартфона после нажатия на кнопку SEND EMAIL (показ диалога выбора)

## Получение результата выполнения явления

```
public void pickContact() {  
    Intent pickContactIntent = new Intent(Intent.ACTION_PICK, Uri.parse("content://contacts"));  
    pickContactIntent.setType(ContactsContract.CommonDataKinds.Phone.CONTENT_TYPE);  
    startActivityForResult(pickContactIntent, PICK_CONTACT_REQUEST);  
}
```

Рисунок 13 – метод запуска явления для выбора контакта

```
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
    if (requestCode == PICK_CONTACT_REQUEST) {  
        if (resultCode == RESULT_OK) {  
            Uri contactUri = data.getData();  
            String[] projection = {ContactsContract.CommonDataKinds.Phone.NUMBER};  
            Cursor cursor = getContentResolver().query(contactUri, projection, selection: null, se  
            cursor.moveToFirst();  
  
            int column = cursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER);  
            String number = cursor.getString(column);  
            System.out.println(number);  
        }  
    }  
}
```

Рисунок 14 – метод получения результата



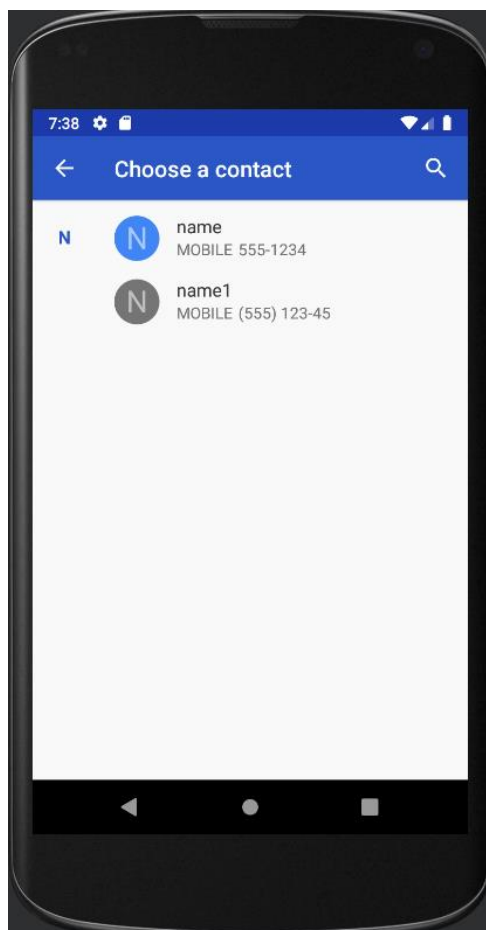


Рисунок 15 – открытие контактной книжки при нажатии на кнопку PICK CONTACT

```
I/System.out: 555-1234  
D/EGL_emulation: eglMakeCurr
```

Рисунок 16 – получение выбранного номера

## Разрешение другим приложениям запускать явления

```
<intent-filter>
    <action android:name="android.intent.action.SENDTO" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="sms"/>
    <data android:scheme="smsto"/>
</intent-filter>
<intent-filter>
    <action android:name="android.intent.action.SEND" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:mimeType="image/*"/>
    <data android:mimeType="text/plain"/>
</intent-filter>
```

Рисунок 17 – добавление фильтра намерений в манифесте

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    initView();

    Intent intent = getIntent();
    try {
        if (intent.getType().contains("image/")) {
            Uri data = intent.getClipData().getItemAt(0).getUri();
            Bitmap bitmap = MediaStore.Images.Media.getBitmap(this.getContentResolver(), data);
            imageFromOtherApp.setImageBitmap(bitmap);
            Intent result = new Intent("com.example.RESULT_ACTION", Uri.parse("content://result_uri"));
            setResult(Activity.RESULT_OK, result);
            //finish();
        } else if ("text/plain".equals(intent.getType())) {
            textFromOtherApp.setText(intent.getClipData().getItemAt(0).getText().toString());
            Intent result = new Intent("com.example.RESULT_ACTION", Uri.parse("content://result_uri"));
            setResult(Activity.RESULT_OK, result);
            finish();
        }
    } catch (NullPointerException | IOException exception) {
        exception.printStackTrace();
    }
}
```

Рисунок 18 – управление намерениями в явлении

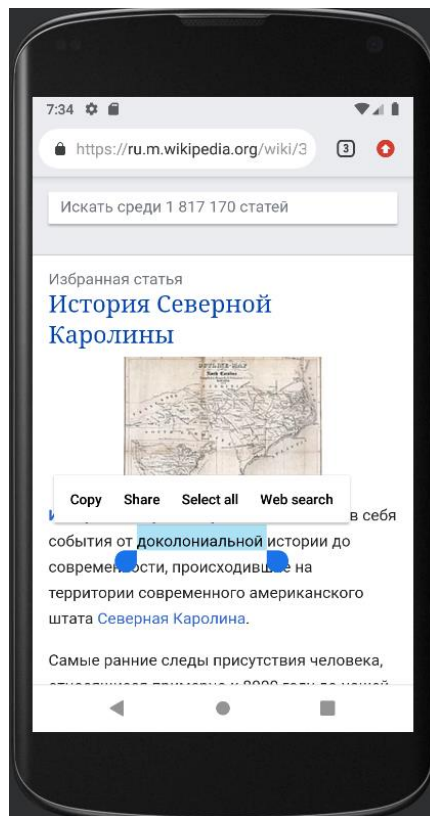


Рисунок 19 – выбор текста для отправки в другое приложение

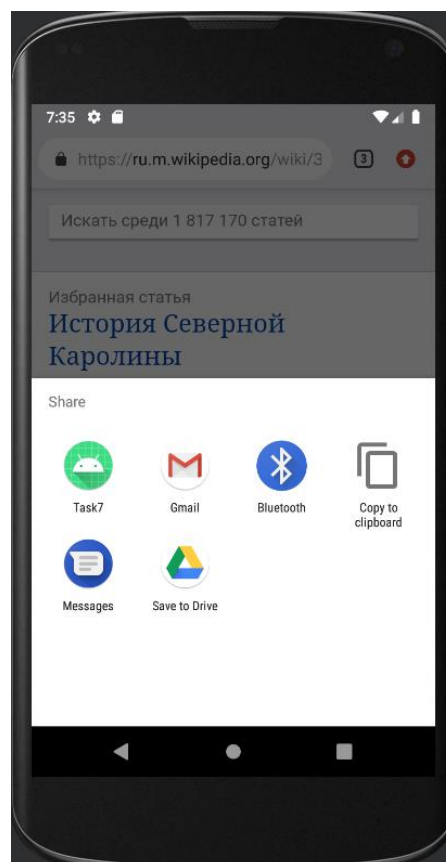


Рисунок 20 – диалоговое окно с выбором приложений для отправки текста

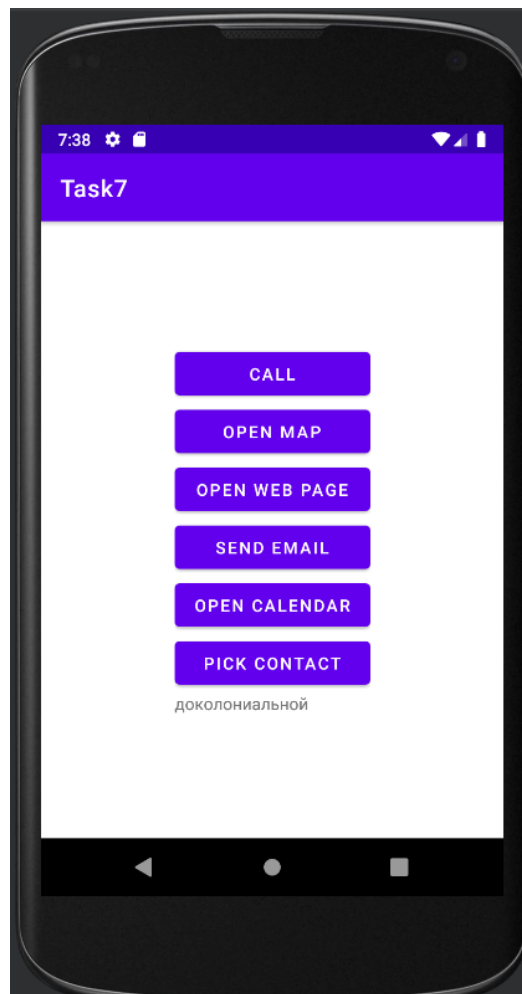


Рисунок 21 – получение текста из другого приложения

## **Вывод**

Создано приложение с использованием намерений для некоторых базовых взаимодействий с другими приложениями, таких как запуск другого приложения, получение результата его выполнения, приложение способно реагировать на намерения других приложений.