

Создание многофункционального поля ввода для Android

Радионов Сергей Владимирович

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация:

В данной работе описан процесс разработки многофункционального поля ввода для Android. Представлен процесс разработки и верстки поля в среде разработки Android Studio.

Ключевые слова: Android, поле ввода, многофункциональное, Android Studio.

Creating your own multifunctional input field for Android

Radionov Sergei Vladimirovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This paper describes the process of developing a multifunctional input field for Android. The process of development and layout of a field in the Android Studio development environment is presented.

Keyword: Android, input field, feature rich, Android Studio.

В настоящее время очень много людей пользуются различными приложениями на мобильных устройствах, которые конкурируют между собой не только в количестве и качестве функционала, но и удобстве и красоте интерфейса. Ведь пользователь выберет более красивый интерфейс, даже если функционал немного уступает. Поэтому нельзя при разработке веб-сервиса отводить качество интерфейса на второй план.

Именно поэтому в настоящее время в цивилизованных странах широкой популярностью пользуются устройства для удаленного управления системами отопления через сеть Интернет. Прикладные программы позволяют организовать простой доступ к системе через смартфон, ноутбук или планшет.

Актуальность задачи разработки приложения Android для информационной системы «Удаленное управление температурой в помещении» обусловлена необходимостью удобных внесений изменений в настройки устройства регулирования по Wi-Fi необходимо приложение. Работа в приложении гораздо удобнее и комфортнее работы в браузере. Поэтому и было решено разработать приложение для системы «Удаленное управление температурой в помещении».

Для реализации системы в среде разработки Android Studio на языке Java были рассмотрены различные литературные материалы. В статье В.И.Макаров анализируются основные способы создания пользовательского интерфейса GUI. Рассматривается специфика разработки интерфейса пользователя при создании Android-приложений применимые к среде разработки Android Studio. А также дается основная характеристика элементов пользовательского интерфейса [1]. В работе Ю.А.Шитиков, А.В.Фесенко описан процесс создания мобильного приложения для android os в среде разработки android studio [2]. Исследование А.В.Макарцев, В.Ю.Пирогов содержит сравнение сред разработки android приложений eclipse и android studio[3]. В статье Н.В.Бужинской, Д.С.Стародумовой рассматриваются вопросы проектирования мобильных приложений для телефонов на базе операционной системы Android. Важным в этом процессе являются выбор платформы разработки и построение диаграммы, описывающей функции и поведение мобильного приложения[4]. Темой статьи Нагибина Е.Н. является выявление оптимального пути для хранения и загрузки изображений в бизнес-приложениях на базе ОС Android. Большинство Android-приложений в качестве источника изображений используют Интернет. Это продиктовано как тем, что большинство продуктов на Android используют Интернет-соединение, тем самым делая отображаемую в них информацию уникальной и не поддающейся предварительной загрузке, так и тем, что изображения, содержащиеся в исполняемом файле apk (формат архивных файлов-приложений в Android) увеличивают размер этого файла, что усложняет его скачивание из Google Play (магазина приложений в Android). В ходе данной статьи в качестве бизнес-приложения будет рассматриваться экран со списком новостей, в котором каждая новость будет уникальной и состоять из изображения, заголовка и описания. Для выявления оптимального пути хранения и загрузки изображений необходимо: 1) Провести обзор существующих способов работы с изображениями в Android 2) Провести испытания всех найденных способов в равных условиях 3) Выбрать наиболее подходящий путь для хранения и загрузки приложений[5]. В работе А.К.Ахметова приводится история создания и развития операционной системы Android и Google Play - магазина приложений для устройств, созданных на платформе Android. Приводятся статистические и прогнозируемые данные по продажам Android-смартфонов, статистические данные по количеству приложений в магазине Google Play, а также результаты опросов, проводимых среди разработчиков мобильных приложений. Обосновывается функциональное назначение Google Play и приводятся основные механизмы обеспечения совместимости устройств. Приводится краткое описание процедуры создания приложения с указанием средств и инструментов, необходимых при разработке. Указываются некоторые недостатки системы. Приводятся советы начинающим разработчикам мобильных приложений, выкладываемые опытными разработчиками в профильном интернет-ресурсе AndroidAuthority. В заключении выносятся преимущества платформы Android и дается

обоснование оптимальности платформы Android для начинающих разработчиков приложений[6]. В статье А.С.Винокурова и Р.И.Баженова рассматривается проект разработки мобильного приложения для коммерческой организации, в свободно распространяемой интегрированной среде разработки Android Studio. Подробно рассмотрены все этапы проектирования мобильного приложения. Представлен обзор проектного предложения, рассмотрены мобильные приложения конкурирующих фирм, реализован план график проекта в программной системе GanttProject, проведен анализ выбора платформы разработки мобильного приложения, описан интерактивный прототип сценария взаимодействия пользователя с мобильным приложением, разработанный в программной системе Balsamiq Mockups, показан графический дизайн интерфейса мобильного приложения, описан процесс разработки облачной SQL-ориентированной базы данных и процесс тестирования, публикации мобильного приложения в Google Play. Практическим результатом исследования является рабочее мобильное приложение для мобильной платформы Android, соответствующее всем предъявленным требованиям заказчика[7]. Не менее важны иностранные исследования [8-9].

Цель исследования: разработать многофункциональное поле ввода для Android.

Для создания своего поля ввода сначала необходимо сверстать его структуру. Для этого создадим файл `edit_text_button_icon.xml` в папке `layouts`. Опишем структуру будущего поля ввода (Рис.1). Далее необходимо описать внешний вид фона поля ввода, для этого создадим файл `edit_text.xml` в папке `drawable` (Рис.2).

После описания внешнего вида нового элемента интерфейса необходимо написать свой класс, который будет наследоваться от `FrameLayout`. В этом классе будет вся логика создаваемого элемента интерфейса. Напишем конструктор класса (Рис.3). Опишем дополнительные функции поля ввода (Рис.4).

```
<FrameLayout
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="wrap_content">

  <EditText
    android:id="@+id/edittext"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textEmailAddress"
    android:ems="10"
    android:textColor="?attr/textColor"
    android:textColorHint="?hintColor"
    android:paddingStart="35dp"
    android:paddingEnd="5dp"
    android:drawablePadding="10dp"
    android:background="@drawable/edit_text"
    android:paddingTop="7dp"
    android:paddingBottom="7dp"
    android:autoFillHints=""
    android:textSize="20sp"
    android:textCursorDrawable="@drawable/cursor_color"/>

  <ImageView
    android:id="@+id/icon"
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_gravity="start|center_vertical"
    android:layout_marginStart="3dp"
    app:tint="?iconColor"/>

  <ImageView
    android:id="@+id/visibility_btn"
    android:visibility="gone"
    android:layout_width="36dp"
    android:layout_height="36dp"
    android:padding="3dp"
    android:layout_gravity="end|center_vertical"
    app:tint="?iconColor"
    android:background="@drawable/ripple"/>

</FrameLayout>
```

Рис.1. Верстка поля ввода

```

<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:state_enabled="true" android:state_focused="true">
    <shape android:shape="rectangle">
      <solid android:color="?inputBgColor"/>
      <stroke android:color="?colorAccent" android:width="1dp"/>
    </shape>
  </item>
  <item android:state_enabled="true" android:state_focused="false">
    <shape android:shape="rectangle">
      <solid android:color="?inputBgColor"/>
      <stroke android:color="?dividerColor" android:width="1dp"/>
    </shape>
  </item>
</selector>

```

Рис.2. Описание фона поля

```

public class EditTextButtonIcon extends FrameLayout {

    EditText editText;
    boolean curVisibility = false;
    Context context;
    int inputTypes;
    Drawable icon;
    ImageView imageView, visibilityBtn;

    public EditTextButtonIcon(@NonNull Context context, @Nullable AttributeSet attrs) {
        super(context, attrs);
        this.context = context;
        TypedArray attributes = context.obtainStyledAttributes(attrs, R.styleable.EditTextButtonIcon);
        String hint = attributes.getString(R.styleable.EditTextButtonIcon_hint);
        inputTypes = attributes.getInt(R.styleable.EditTextButtonIcon_android_inputType, EditorInfo.TYPE_NULL);
        icon = attributes.getDrawable(R.styleable.EditTextButtonIcon_iconSrc);
        @ColorInt int color = attributes.getInt(R.styleable.EditTextButtonIcon_icolor, defValue: 0);
        boolean isPassword = attributes.getBoolean(R.styleable.EditTextButtonIcon_isPassword, defValue: false);
        LayoutInflater inflater = (LayoutInflater) context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        inflater.inflate(R.layout.edit_text_button_icon, root: this, attachToRoot: true);

        visibilityBtn = findViewById(R.id.visibility_btn);
        visibilityBtn.setOnClickListener(v -> ChangeVisibility());

        editText = findViewById(R.id.edittext);
        imageView = findViewById(R.id.icon);

        editText.setHint(hint);
        editText.setInputType(inputTypes);

        imageView.setImageDrawable(icon);
        if (color != 0)
            imageView.setColorFilter(color, PorterDuff.Mode.SRC_IN);

        setIsPassword(isPassword);
    }
}

```

Рис.3. Класс и конструктор поля ввода

```
public EditText getEditText() { return editText; }  
public static EditText getEditTextById(AppCompatActivity activity, int id) {  
    return ((EditTextButtonIcon) activity.findViewById(id)).getEditText();  
}  
public static EditText getEditTextById(View v, int id) {  
    return ((EditTextButtonIcon) v.findViewById(id)).getEditText();  
}  
public void ChangeVisibility(boolean visibility) {  
    if (visibility) {  
        visibilityBtn.setImageResource(R.drawable.ic_visibility_on);  
        editText.setTransformationMethod(HideReturnsTransformationMethod.getInstance());  
    } else {  
        visibilityBtn.setImageResource(R.drawable.ic_visibility_off);  
        editText.setTransformationMethod>PasswordTransformationMethod.getInstance();  
        // editText.setTransformationMethod(new RombPasswordTransformMethod());  
    }  
}  
public void ChangeVisibility() {  
    curVisibility = !curVisibility;  
    ChangeVisibility(curVisibility);  
}  
public void setIsPassword(boolean isPassword){  
    Typeface cache = editText.getTypeface();  
    if (isPassword){  
        visibilityBtn.setVisibility(VISIBLE);  
        editText.setPadding(editText.getPaddingLeft(), editText.getPaddingTop(), Utils.convertDpToPx(10), editText.getPaddingRight());  
        ChangeVisibility(curVisibility);  
        editText.setInputType(InputType.TYPE_CLASS_TEXT | InputType.TYPE_TEXT_VARIATION_PASSWORD);  
        imageView.setImageResource(R.drawable.ic_baseline_lock);  
    } else {  
        visibilityBtn.setVisibility(GONE);  
        editText.setTransformationMethod(HideReturnsTransformationMethod.getInstance());  
        editText.setPadding(editText.getPaddingLeft(), editText.getPaddingTop(), Utils.convertDpToPx(10), editText.getPaddingRight());  
        editText.setInputType(inputTypes);  
        imageView.setImageDrawable(icon);  
    }  
    editText.setTypeface(cache);  
}
```

Рис.4. Функции класса поля ввода

После описания верстки и логики разработанного поля ввода можно использовать его в своих активити (Рис.5). Если запустить приложение, поля будут выглядеть как на рисунке 6. Внешний вид заполненных полей представлен на рисунке 7.

```
<com.fellowcode.easypassword.Views.EditTextButtonIcon
    android:id="@+id/nameEdit"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textEmailAddress"
    app:hint="Name"
    app:iconSrc="@drawable/ic_baseline_assignment" />

<com.fellowcode.easypassword.Views.EditTextButtonIcon
    android:id="@+id/loginEdit"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:inputType="textEmailAddress"
    app:hint="Email"
    app:iconSrc="@drawable/ic_baseline_person" />

<com.fellowcode.easypassword.Views.EditTextButtonIcon
    android:id="@+id/passwordEdit"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    app:hint="Password"
    app:iconSrc="@drawable/ic_baseline_lock"
    app:isPassword="true" />
```

Рис.5. Пример использования в верстке

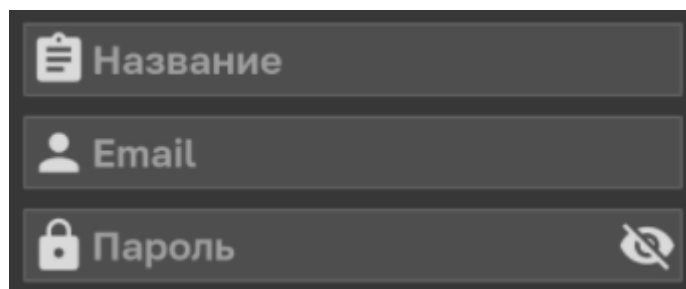


Рис.6. Внешний вид полей ввода

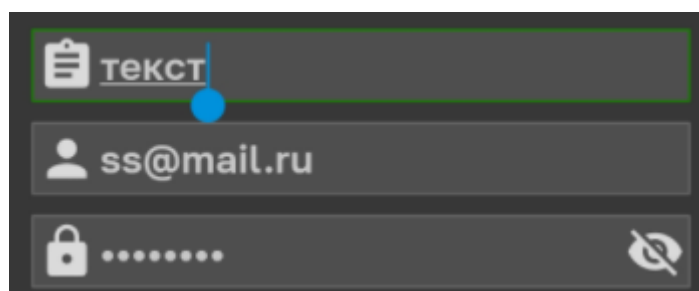


Рис.7. Внешний вид полей ввода после заполнения

Таким образом, было разработано многофункциональное поле ввода.

Библиографический список

1. Макаров В.И. Особенности разработки пользовательского интерфейса для android-приложений в среде разработки android studio // Теория и практика современной науки. 2018. № 7 (37). С. 222-224.
2. Шитиков Ю.А., Фесенко А.В. Создание мобильного приложения для android os в среде разработки android studio // В сборнике: Восемнадцатая всероссийская студенческая научно-практическая конференция Нижневартковского государственного университета Статьи докладов. ответственный редактор А.В. Коричко. 2016. С. 1459-1463.
3. Макарецев А.В., Пирогов В.Ю. Сравнение сред разработки android приложений eclipse и android studio // В сборнике: Актуальные проблемы теории и методики обучения информатике, математике и экономике Материалы молодежной всероссийской научно-практической конференции. Шадринский государственный педагогический университет; ответственный редактор: И. Н. Слинкина. 2016. С. 57-58.
4. Бужинская Н.В., Стародумова Д.С. Проектирование мобильных приложений для смартфонов на базе операционной системы android // Современная техника и технологии. 2017. № 4 (68). С. 16-18.
5. Нагибин Е.Н. Работа с изображениями в бизнес-приложениях на базе os android // В сборнике: International Conference on Science, Agriculture, Engineering and Management Conference Proceedings. 2017. С. 63-70.
6. Ахметов А.К. Операционная система android: история создания и развития. Разработка приложений для платформы android // Скиф. Вопросы студенческой науки. 2017. № 9 (9). С. 108-115.
7. Винокуров А.С., Баженов Р.И. Разработка мобильного приложения для теннисного клуба в среде android studio // Постулат. 2016. № 11 (13). С. 34.
8. Rogers R. et al. Android application development: Programming with the Google SDK. – O'Reilly Media, Inc., 2009.
9. Meier R. Professional Android 4 application development. John Wiley & Sons, 2012.
10. Радионов С.В. Разработка информационной системы «Удаленное управление температурой в помещении» // Постулат. 2019. № 6.