GRAPHICAL USER INTERFACE DEVELOPMENT FOR ANDROID OPERATING SYSTEM

I.G. Rusinov, V.V. Sokolova
Scientific supervisor: Assoc. Prof. V.V. Sokolova
Tomsk Polytechnic University, Russia, Tomsk, Lenin str., 30, 634050
E-mail: RusinovIG@gmail.com

В статье описываются возможности создания пользовательского интерфейса для мобильных приложений на базе операционной системы Google Android. Рассмотрена архитектура интерфейса мобильного приложения, а так же основные элементы, такие как разметки, текстовые поля, кнопки, элементы списков и т.д.

Introduction

At present time, there is a rapid growth in popularity of various mobile devices such as smart phones, communicators, tablets, etc. One of the most popular operating systems for such devices is the Google Android [1].

Android – is a software stack for mobile devices that includes an operating system (OS), software, middleware, as well as basic user applications (e-mail-client, calendar, maps, browser, contacts, etc.). The popularity of the platform due to the large number of applications such as games and entertaining applications, and various business applications such as office applications, schedulers, calendars, etc.

The actual scope for mobile devices on Android platform could be the development of mobile applications for travel planning. Using a mobile device to store all necessary information on any trip provides its availability, and event notification service will remember user about important events.

Android provides a variety of elements to create a graphical user interface, from simple buttons to different widgets that simplify design of user interfaces.

Graphical user interface in Android OS

Graphical user interface in Android is generated by using the View and ViewGroup objects. View class is the base class for ViewGroup.

View Objects – basic modules for displaying a user interface on Android platform. Class View is the core for the subclasses, called widgets, which offer fully implemented user interface objects, such as text fields, buttons, etc.

Class ViewGroup is the core for the subclasses, called layouts, which form the location of the user interface on the application form.

Android supports the following ViewGroup types:

- LinearLayout – arranges views in a single column or a single row.
- AbsoluteLayout – enables to specify the exact location of its children. Location is set by the following
attributes: android:layout_x and android:layout_y.

- TableLayout – groups views into rows and columns.
- TableRow – element designates a row in the table. Each row can contain one or more views, that forms table cells.
- RelativeLayout – enables to specify how child views are positioned relative to each other.
- FrameLayout – a placeholder on screen that can be used to display a single view. Views that added to a FrameLayout are always anchored to the top left of the layout.
- ScrollView – a special type of FrameLayout in that it enables users to scroll through a list of views that occupy more space than the physical display. The ScrollView can contain only one child view or ViewGroup, which normally is a LinearLayout.

Different layouts can be combined in application in order to create required graphical interface.

**User interface elements in Android OS**

Basic interface elements in Android applications are the following [2].

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TextView</td>
<td>Basic element used to display text.</td>
</tr>
<tr>
<td>EditText</td>
<td>Subclass of class TextView, allows the user to edit text.</td>
</tr>
<tr>
<td>Button</td>
<td>Button element.</td>
</tr>
<tr>
<td>ImageButton</td>
<td>The same as Button, but in allows to display an image.</td>
</tr>
<tr>
<td>CheckBox</td>
<td>Special type of button has two states: checked and unchecked.</td>
</tr>
<tr>
<td>ToggleButton</td>
<td>Displays a checked / unchecked state with light indicator.</td>
</tr>
<tr>
<td>RadioButton</td>
<td>Has two states: checked and unchecked. Once it is checked, it can’t be unchecked by user.</td>
</tr>
<tr>
<td>RadioGroup</td>
<td>Used to group multiple RadioButton items. Allows to check out only one RadioButton in group.</td>
</tr>
</tbody>
</table>

Setting the date and time – one of the most frequently used operations in mobile applications. Android supports this functionality by using elements TimePicker and DatePicker. TimePicker allows the user to specify time of the day in 24-hour or 12 hour formats. DatePicker is similar to the element TimePicker and allows the user to specify a particular date.

List items allow to display large sets of elements. Android has two types of lists: ListView and SpinnerView. ListView displays a large array of items as a vertical scrolling list. If it’s needed to display a large list of items, but there is no enough space for the location of the element such as a ListView, SpinnerView can be used. It displays one item from the list, and allows the user to choose between them.

The example below shows capabilities of Android to create complex user interfaces. Figure 1 shows list of events, where each element of list is a combination of 3 elements: ImageView and two TextViews.
Figure 2 shows the example of forms in Android application. It contains the following elements: EditText, DatePicker, TimePicker, and Button.

**Conclusion and Future Work**

This article discusses the capabilities of Android mobile platform for creating graphical user interfaces. The popularity and rapid growth in the number of users on the Android platform due to its openness, simplicity of application development, including the rich possibilities for the interface development. Google Android supports a wide range of ready to use elements such as text fields, buttons, lists, etc.

Because of it, Android has been chosen as a platform for developing of mobile application for travel planning. This application will allow saving into the mobile device all the necessary information about travel, set up alerts to remind user about upcoming events of travel, as well as share information about traveling with help of e-mail or social networking.

**REFERENCES**