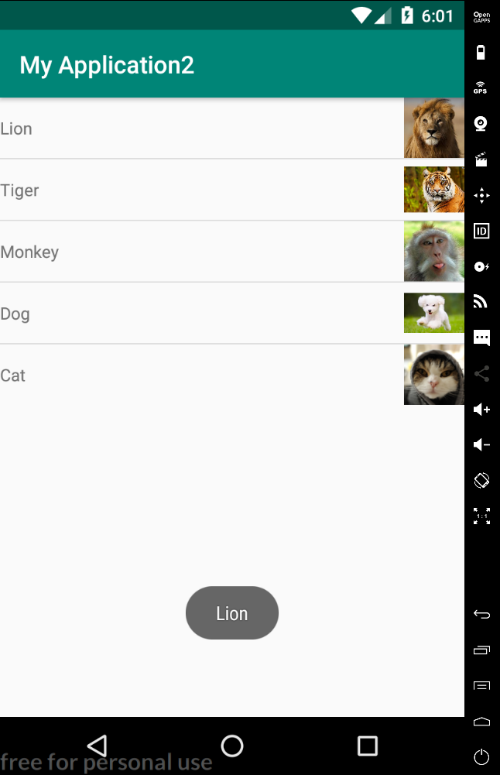
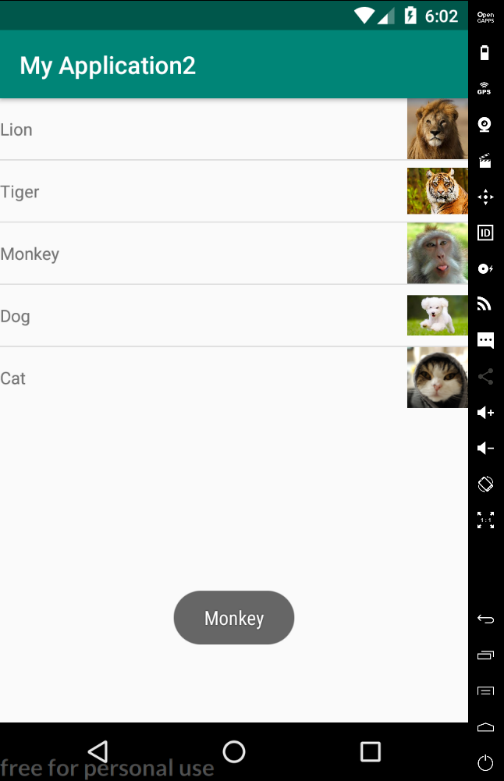
一、利用SimpleAdapter实现如下界面效果

（1）注意列表项的布局

（2）图片使用QQ群附件资源

（3）使用Toast显示选中的列表项信息

效果图如下：点击对应的item会显示不同的toast

部分代码：

/\*\*

\*实例化simpleAdapter对象，传入四个参数，this：当前应用程序环境

\* listitem：数据源,包括文本，图片id等

\* keys：数据源的属性名

\* view\_id：布局文件中的组件id

\*/

SimpleAdapter SA = new SimpleAdapter(this,listitem,R.layout.item,keys,view\_id);

final ListView listView = findViewById(R.id.listitem);

listView.setAdapter(SA);

//listView的toast响应函数

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

String val ="";

if(position==0){

val="Lion";

}else if(position==1){

val="Tiger";

}else if(position==2){

val="Monkey";

}else if(position==3){

val="Dog";

}else if(position==4){

val="Cat";

}else{

val="ERROR";

}

Toast toast = Toast.makeText(MainActivity.this,val,Toast.LENGTH\_LONG);

toast.show();

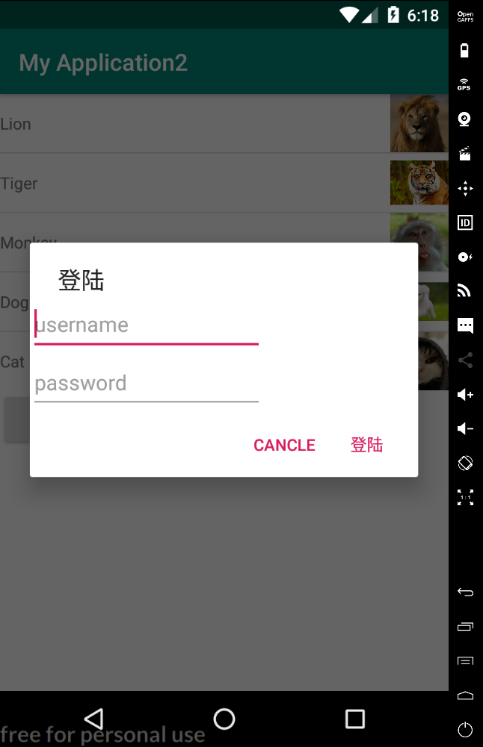
}

});

}

二、创建如图所示的自定义对话框 ◼ 请创建一个如图所示的布局， ◼ 调用 AlertDialog.Builder 对象上的 setView() 将布局添加到 AlertDialog

效果图



部分代码：在界面上生成一个登陆按钮，在布局文件中的button属性中设置android：onclick=“Login”，在mainactivity中定义按钮的点击函数，在login.xml文件中定义登陆的表单样式

//登陆按钮响应函数

public void Login(View view){

LinearLayout loginForm = (LinearLayout) getLayoutInflater().inflate(R.layout.login,null);

new AlertDialog.Builder(this)

.setTitle("登陆")

.setView(loginForm)

.setPositiveButton("登陆",new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

Toast toast = Toast.makeText(MainActivity.this,"点击登陆按钮",Toast.LENGTH\_SHORT);

toast.show();

}

})

.setNegativeButton("cancle",new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

Toast toast = Toast.makeText(MainActivity.this,"点击q取消按钮",Toast.LENGTH\_SHORT);

toast.show();

}

})

.create()

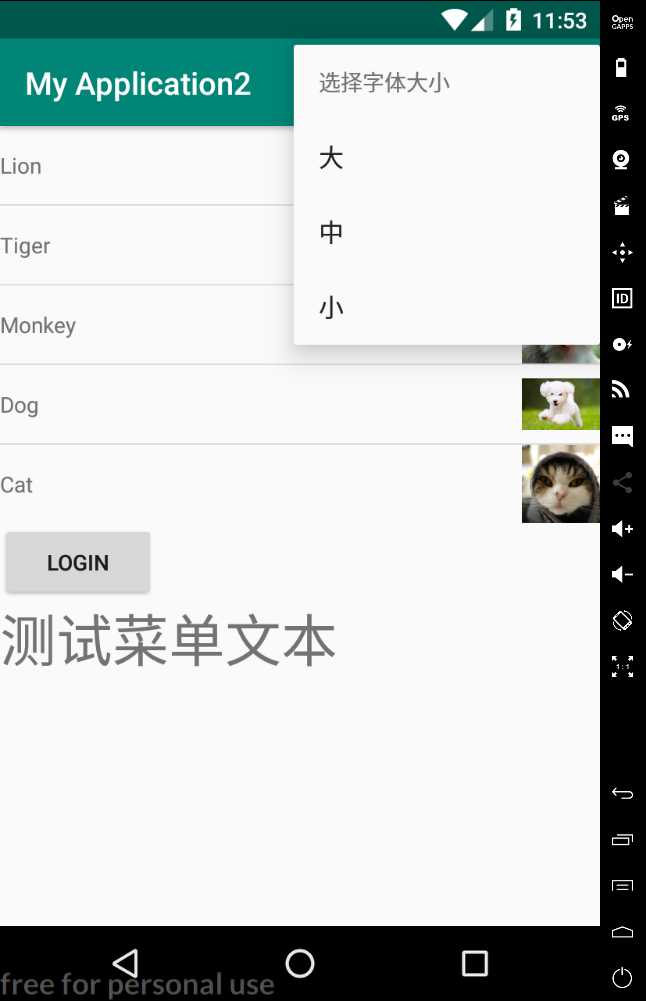
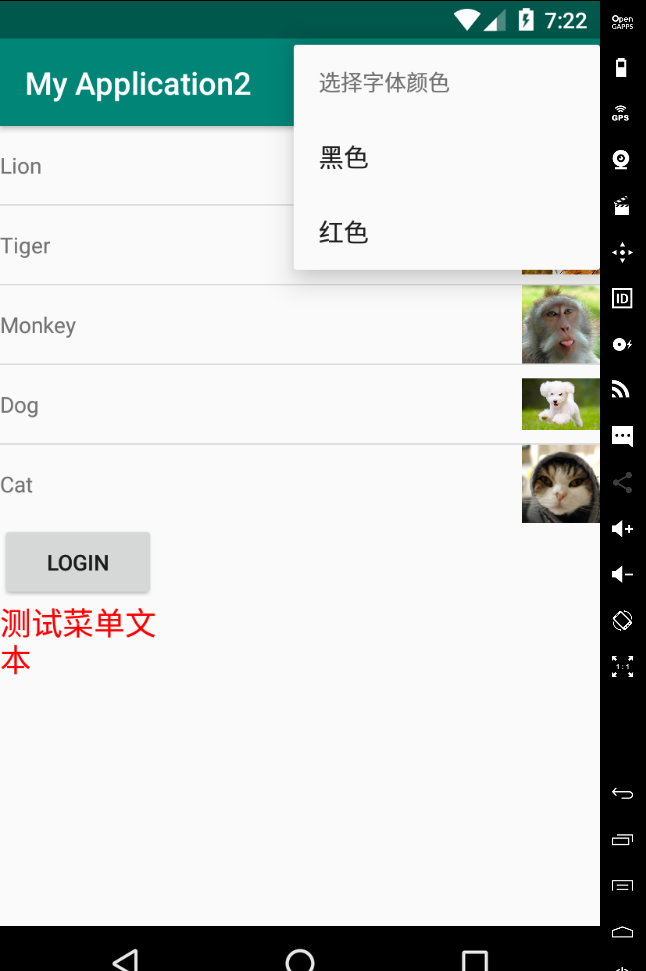
.show();

}

}

三、使用XML定义菜单 ◼ 字体大小（有小，中，大这3个选项；分 别对应10号字，16号字和20号字）； 点击之后设置测试文本的字体 ◼ 普通菜单项，点击之后弹出Toast提示 ◼ 字体颜色（有红色和黑色这2个选项）， 点击之后设置测试文本的字体

效果图：



部分代码：

//程序开始运行即创建菜单项

public boolean onCreateOptionsMenu(Menu menu){

SubMenu fontMenu = menu.addSubMenu("字体大小");

fontMenu.setHeaderTitle("选择字体大小");

fontMenu.add(0,0,0,"大");

fontMenu.add(0,1,0,"中");

fontMenu.add(0,2,0,"小");

menu.add(0,3,0,"普通菜单项");

SubMenu colorMenu = menu.addSubMenu("字体颜色");

colorMenu.setHeaderTitle("选择字体颜色");

colorMenu.add(0,4,0,"黑色");

colorMenu.add(0,5,0,"红色");

return super.onCreateOptionsMenu(menu);

}

//点击菜单项响应函数

public boolean onOptionsItemSelected(MenuItem menuItem){

switch (menuItem.getItemId()){

case 0:

textView.setTextSize(18\*2);

break;

case 1:

textView.setTextSize(14\*2);

break;

case 2:

textView.setTextSize(10\*2);

break;

case 3:

Toast.makeText(MainActivity.this,"点击了普通菜单项",Toast.LENGTH\_LONG).show();

break;

case 4:

textView.setTextColor(Color.BLACK);

break;

case 5:

textView.setTextColor(Color.RED);

break;

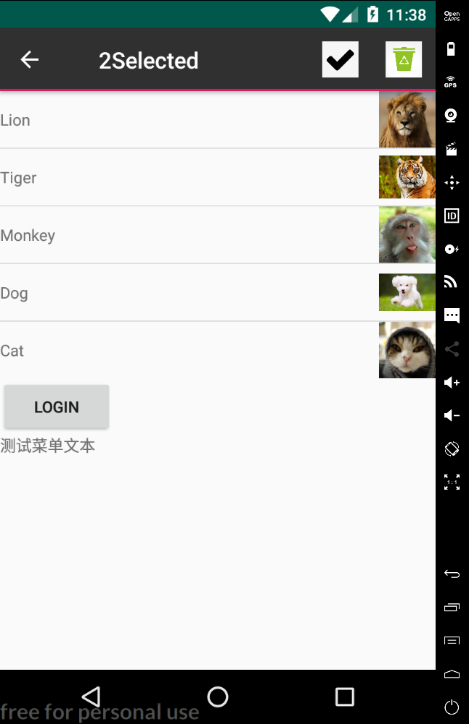
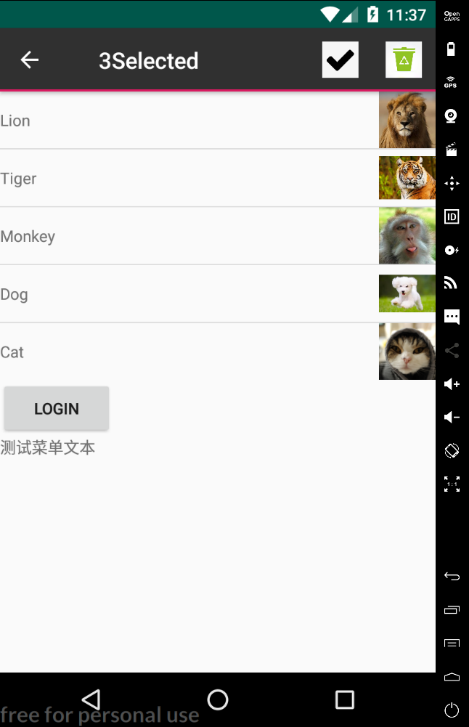
}

return true;

}

四、创建上下文操作模式(ActionMode)的上下文菜单

效果图：直接使用本实验的listview作为菜单项，



代码片段：

该函数监听item被选中状态的改变，使用变量num来记录被选定的item数量，并显示。

长按菜单项时执行该函数，绑定显示的菜单文件，actionmode\_menu是事先被定义的menu文件

