

warwick C<>ding



by Wesley Haigh



Lecture 6

ViewUserActivity



Recap Exercise

- To get the profile pictures working:
 - Need to modify **MainActivity** so that it creates a folder to store the pictures
 - Need to modify the **AndroidManifest.xml** to allow permission to create folders on the phone
 - Need to save pictures to this folder when record is created
 - Need to get these pictures when user **hasPicture**



Recap Exercise

- Add **setUpFolders()** function in **MainActivity**

```
private void setUpFolders(){
    File dir = new File(Environment.getExternalStorageDirectory().toString() + "/WarwickCoding/");
    File dir2 = new File(Environment.getExternalStorageDirectory().toString() + "/WarwickCoding/.ProfilePictures/");
    try{
        if(dir.mkdir()) {
            Log.e("directory 1", "Directory created");
        } else {
            Log.e("directory 1", "Directory is not created");
        }
    } catch(Exception e){
        e.printStackTrace();
    }
    try{
        if(dir2.mkdir()) {
            Log.e("directory 2", "Directory created");
        } else {
            Log.e("directory 2", "Directory is not created");
        }
    } catch(Exception e){
        e.printStackTrace();
    }
}
```



Recap Exercise

- Modify the **AndroidManifest.xml** (found in the **manifests** folder)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.warwickcodingapp" >

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.INTERACT_ACROSS_USERS_FULL" />
```



Recap Exercise

- Edit **createRecord()** function in **UsersActivity**

```
private void createRecord() {
    User newUser = new User();

    //add the users name after checking it is not empty
    String name = _name.getText().toString();
    if(!name.equals("")) {
        newUser.setName(name);
    }
    else {
        showErrorDialog("Error Creating Record", "Name must not be nothing");
    }
    newUser.setEmail(_email.getText().toString());
    newUser.setAge(Integer.parseInt(_age.getText().toString()));
    newUser.setLocation(_location.getText().toString());
    newUser.setGender(_gender.getSelectedItemPosition() == 0);
    newUser.setHasPicture(_profileBitmap != null);
    newUser.setId(_userTable.createRow(newUser));
    PictureServices.saveProfilePicFromBitmap(_profileBitmap, newUser.getId());
    Toast.makeText(this, "Created User", Toast.LENGTH_SHORT).show();
    finish();
}
```



Recap Exercise

- Edit **setHasPicture()** function in **User**

```
public void setHasPicture(boolean hasPicture) {  
    this._hasPicture = hasPicture;  
    if (hasPicture) {  
        String mPath = Environment.getExternalStorageDirectory().toString() +  
            "/WarwickCoding/.ProfilePictures/" + "profile_" + this._id + ".jpg";  
        Bitmap profilePic = PictureServices.decodeSampledBitmapFromFile(mPath, 90, 90);  
        this.setProfilePicture(profilePic);  
    }  
}
```





ViewUserActivity




Add a New Activity

- Create an activity called **ViewUserActivity**

 Customize the Activity 

Creates a new blank activity with an app bar.


Blank Activity

Activity Name:

Layout Name:

Title:

☐ Launcher Activity ☒ Use a Fragment

Hierarchical Parent: ▼ ...

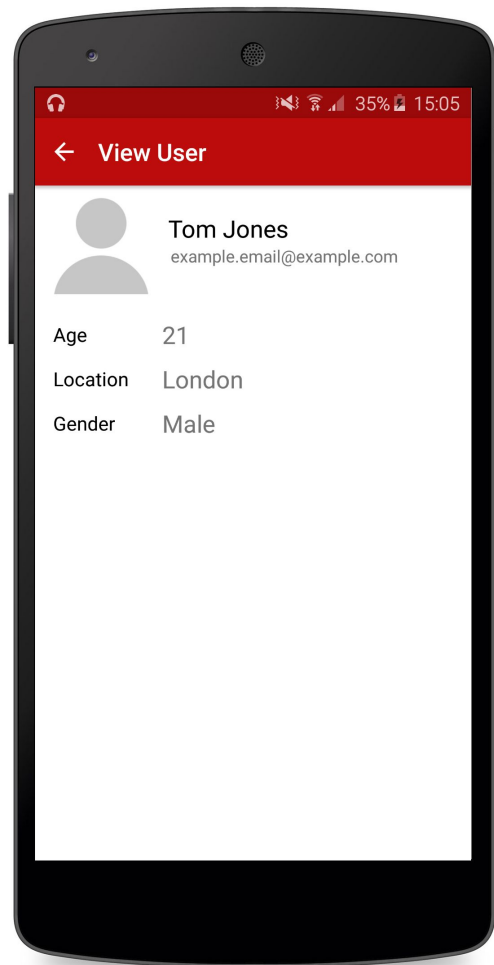
Package name: ▼

Target Source Set: ▼



Add a New Activity

- Remove the Floating Action Button (FAB) in `activity_view_users.xml`
- Edit **`fragment_view_users.xml`** so that it looks like the right
- (Can copy the code from `content_add_record.xml` and change `EditTexts` to `TextViews`)





First Step - XML

Remember:

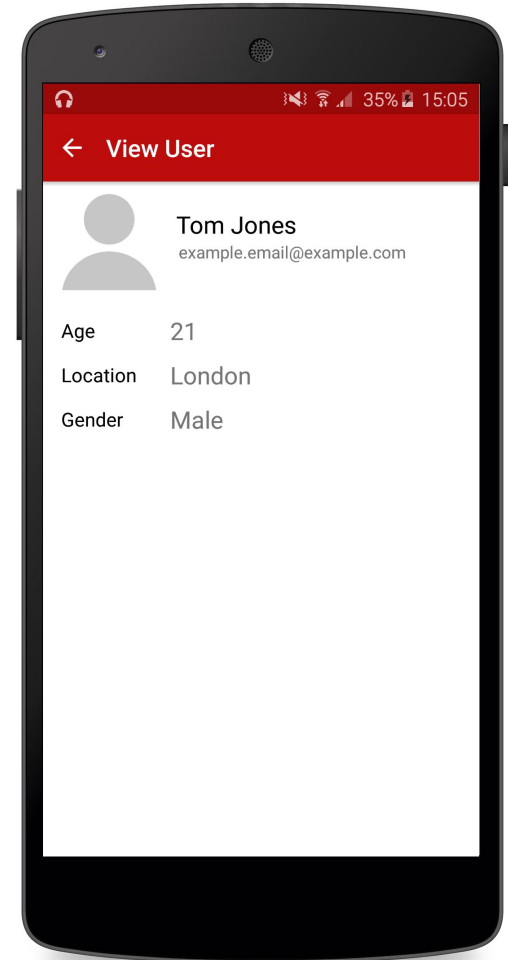
- Add relevant String resources to **strings.xml**
- Give the TextViews and Imageview relevant ids





Second Step - Java

- Functionality of the activity:
 - All of the functionality comes in the setup of the activity.
 - Want to pass an id that will be used to lookup the User record
 - Then once the record has been found, populate the page with the correct details





Java

- Modify ViewUsersActivity.java
 - **getXMLControls()**

```
private void getXMLControls() {  
    _name = (TextView) findViewById(R.id.name);  
    _email = (TextView) findViewById(R.id.email);  
    _age = (TextView) findViewById(R.id.age);  
    _location = (TextView) findViewById(R.id.location);  
    _gender = (TextView) findViewById(R.id.gender);  
    _profilePicture = (ImageView) findViewById(R.id.profilePicture);  
}
```



Java - ViewUserActivity

```
public class ViewUserActivity extends AppCompatActivity {
    public final static String idKey = "UserId";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_user);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        if (getSupportActionBar() != null)
            getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        if (getIntent().getExtras() != null) {
            int id = getIntent().getExtras().getInt(idKey);
            UserTable userTable = new UserTable(this);
            User user = userTable.getUserFromID(id);
        }
        else {
            finish();
        }
    }
}
```



Java - ViewUserActivity

- Create **setUserDetails(User user)** function

```
private void setUserDetails(User user) {  
    _name.setText(user.getName());  
    _email.setText(user.getEmail());  
    _age.setText(user.getAge()+"");  
    _location.setText(user.getLocation());  
    String gender = "";  
    if (user.getGender()) {  
        gender = "Male";  
    }  
    else {  
        gender = "Female";  
    }  
    _gender.setText(gender);  
    if (user.getHasPicture()) {  
        _profilePicture.setImageBitmap(user.getProfilePicture());  
    }  
}
```



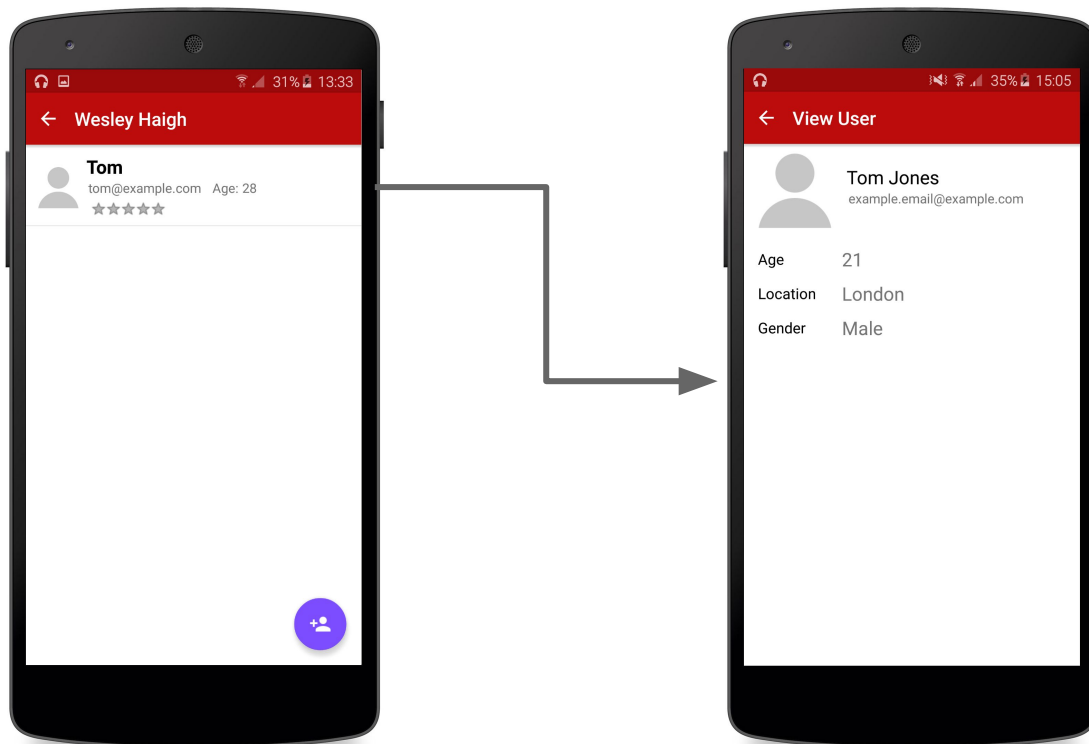
Java

- Now we need to connect ViewUserActivity to UsersActivity

```
private void initialiseList() {  
    final ArrayList<User> users = _usersTable.getAllUsers();  
  
    _userListAdapter = new UserListAdapter(this, users);  
    _userList.setAdapter(_userListAdapter);  
    _userList.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
        @Override  
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
            User user = users.get(position);  
            if (user != null) {  
                Intent intent = new Intent(UsersActivity.this, ViewUserActivity.class);  
                intent.putExtra(ViewUserActivity.idKey, user.getId());  
                startActivity(intent);  
            }  
        }  
    });  
}
```




Final Product





End of Lecture Exercise

- Fix the problem that when you create a record and go back to the list it isn't updated
 - Need to use **startActivityForResult(Intent intent)** in UsersActivity.java
 - Need to add class variable **ADDED_RECORD** (int) = 1
 - Add **onActivityResult** function to UsersActivity



End of Lecture Exercise

- All code is in UsersActivity.java
- Add **ADDED_RECORD** variable

```
public static final int ADDED_RECORD = 1;
```

- Modify to use **startActivityForResult(Intent intent)**

```
fab.setOnClickListener((view) → {  
    Intent intent = new Intent(UsersActivity.this, AddRecordActivity.class);  
    startActivityForResult(intent, ADDED_RECORD);  
});
```



End of Lecture Exercise

- All code is in UsersActivity.java
- Add **onActivityResult** function to UsersActivity

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == ADDED_RECORD) {
        initialiseList();
    }
}
```

warwick
C<>ding