Java for Mobile Development

Search this site

Navigation

Home
Syllabus - Fall 2015
Assignments - Fall 2015
Content - Mini 1
Content - Mini 2

Content - Mini 2 >

Designing

Step 1:
Designing Screens

First convert your wireframes into activities supporting styles for a minimum of two resolutions. (Please refer to link - http://developer.android.com/about/dashboards/index.html for popular screen sizes and densities.

If you are creating your activities using WYSIWYG editor then make sure of the following:
All Android Resources are identified by their id in Java source code.
Declarative approach on Resources

```
@+id/tableRow1
```

Means – that ID tableRow1 will create if it does not already exist.

Formal structure

```
@[package:]type/name
```

type refers to drawable, attr, id, layout, etc in R.java
Be careful not to use duplicate names in entire project as these are declared static.

Declare each referenceable User Interface and Activity with its own String Declarative
For e.g. -

```
 android:text="@string/billTotal"
android:textColor="#000"


  <TextView android:id="@+id/billTextView"
```

```
         android:layout_width="wrap_content"

 android:layout_height="wrap_content"
         android:text="@string/billTotal"
android:textColor="#000"
         android:gravity="right"
android:paddingRight="5dp"></TextView>
```

Use the following convention to name each activity:
 PackageName_ActivityName.xml

Step 2:
Designing Presentation Tier (Mainly for Input/Output)
Determine Intents for each activity based on your Navigation Flow. It is a good idea to organize code for each intent using a separate file for each defined activity. If an intent involves usage of two activities the presentation code then the code can be placed in either activity file. If it makes sense you can / should create sub packages for application sub components.

Step 3:
Designing Content Provider(For Storage)
Create DB Schema (if one is required) for storing data locally. Represent the schema using Crow Notation or UML Notation (Refer to document #1 in zip file located at following location). Please make sure you are following rules of normalization (Refer to document -(2) Introduction to RDBMS at location.) Setup a package called DBLayout in `/src` folder and add a class for each entities CRUD (Create, Read, Update, Delete Operation). You are expected to use Java Coding Standards.

Step 4:
Designing Application Tier(For business logic)
Design entities and relationship between them. These would be placed in `/entities` folder.
This step should create objects to be utilized in presentation tier. The object should be exposed to presentation tier using a

set of business method. I encourage you to not ignore this hard design requirements. I have seen several Android Apps that look like scripts with a single file populated with inner classes and methods that belong to business logic layer. You should think about using interfaces and abstract classes as appropriate to expose functionality to presentation layer.

Step 5:
Designing Integration Tier
Separate out interactions with applications on same device or remote devices. Create a package called /ws for organizing your Integration Tier.
Create subpackages called `/local` or `/remote` for local and remote services. Create interfaces for local and remote services for both client and server side. If you are accessing an existing service like Facebook then write interfaces for local service(s).

Package your design in a single document and submit.

Grading Rubric

```
#  Criteria                 Total Points   Points
Awarded
                        30        XX
1 Application supports two resolutions (Screen sizes
and densities)  2        XX
2 Confirm and check that each activity and fragment
has a correcponding  4        XX
 java file in user interface package in which each UI
object is represented
 as an instance variable with input/output
 statements setup to read/write values as needed.
3 Intents are setup to enable navigation between
```

activities or fragments.   3   XX

There should be no activity or fragment without an intent.

4 For Storage if database is used the schema is adequately designed   3   XX

(using rules of normalization).

5 All entities are adequately setup with encapsulation and   3   XX

Object Oriented Design Guidelines. Abstract classes and

Interfaces are used where necessary.

6 Web Services are implemented for local and remote use.   3   XX

7 Class Diagram and supporting Documentation for design and is clearly described. 5   XX

8 Exception Handling and some self healing is planned.
      3   XX

9 Application design has a tiered approach for ui, database,   4   XX

services, model or object layer etc.

Comments

Sign in | Recent Site Activity | Report Abuse | Print Page | Powered By Google Sites