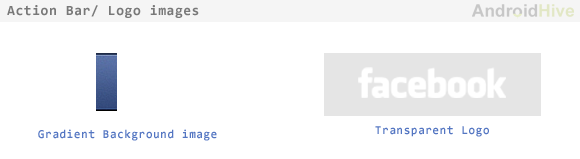
The main goal is to achieve dashboard screen layout and provide navigation to related screens on selecting appropriate icon on the dashboard.  
So let’s start by creating simple project.

**1**. Create a new project by going to **File ⇒ New Android Project**. Fill all the details and name your activity as **AndroidDashboardDesignActivity**.  
**2**. In this project i am separating dashboard screen into Action Bar(Header), Dashboard and Footer. Finally i will merge all layouts together.  
**3**. Under **res/values** create a new xml file and name it as **styles.xml**  
( Right Click on res/layout ⇒ New ⇒ Android XML File) and fill it with following code.

|  |
| --- |
| styles.xml |
| <resources>      <style name="ActionBarCompat">          <item name="android:layout\_width">fill\_parent</item>          <item name="android:layout\_height">50dp</item>          <item name="android:orientation">horizontal</item>          <item name="android:background">@drawable/actionbar\_background</item>      </style>        <style name="DashboardButton">          <item name="android:layout\_gravity">center\_vertical</item>          <item name="android:layout\_width">wrap\_content</item>          <item name="android:layout\_height">wrap\_content</item>          <item name="android:gravity">center\_horizontal</item>          <item name="android:drawablePadding">2dp</item>          <item name="android:textSize">16dp</item>          <item name="android:textStyle">bold</item>          <item name="android:textColor">#ff29549f</item>          <item name="android:background">@null</item>      </style>       <style name="FooterBar">          <item name="android:layout\_width">fill\_parent</item>          <item name="android:layout\_height">40dp</item>          <item name="android:orientation">horizontal</item>          <item name="android:background">#dedede</item>      </style>  </resources> |

⇒ Designing Actionbar (Header)

Design two images for 1. Gradient background for action-bar background, 2. Logo of the application.

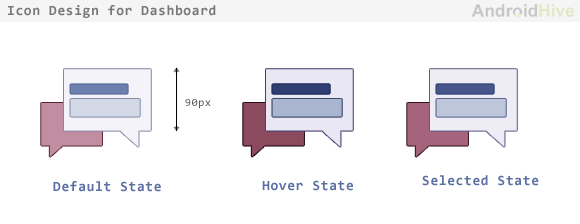


**4**. Create new xml file under **res/layouts** and name it as **actionbar\_layout.xml**  
( Right Click on res/layout ⇒ New ⇒ Android XML File) and type the following code.

|  |
| --- |
| actionbar\_layout.xml |
| <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"      style="@style/ActionBarCompat" >        <ImageView          android:layout\_width="wrap\_content"          android:layout\_height="fill\_parent"          android:clickable="false"          android:paddingLeft="15dip"          android:scaleType="center"          android:src="@drawable/facebook\_logo" />    </LinearLayout> |

⇒ Designing Dashboard

For dashboard design your icons using some graphic editor software (i used Photoshop to create icons). Design each icon for three stages **Default State**, **Hover state** and **Selected state**. Create all icons with 90px height.



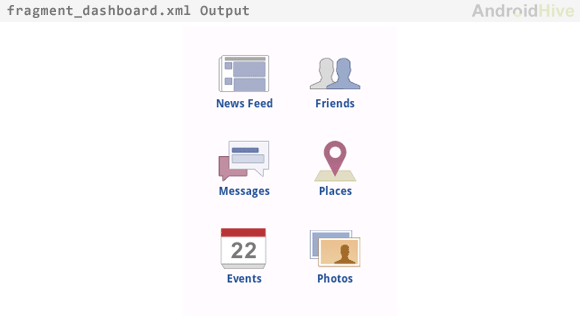
**5**. Create a new class under res/package. **Right Click on src/package folder ⇒ New ⇒ Class** and name it as **DashboardLayout.java** and fill it with following code.

|  |
| --- |
| DashboardLayout.java |
| package com.androidhive.dashboard;  /\*   \* Copyright 2011 Google Inc.   \*   \* Licensed under the Apache License, Version 2.0 (the "License");   \* you may not use this file except in compliance with the License.   \* You may obtain a copy of the License at   \*   \*      <http://www.apache.org/licenses/LICENSE-2.0>   \*   \* Unless required by applicable law or agreed to in writing, software   \* distributed under the License is distributed on an "AS IS" BASIS,   \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.   \* See the License for the specific language governing permissions and   \* limitations under the License.   \*/  import android.content.Context;  import android.util.AttributeSet;  import android.view.View;  import android.view.ViewGroup;    /\*\*   \* Custom layout that arranges children in a grid-like manner, optimizing for even horizontal and   \* vertical whitespace.   \*/  public class DashboardLayout extends ViewGroup {        private static final int UNEVEN\_GRID\_PENALTY\_MULTIPLIER = 10;        private int mMaxChildWidth = 0;      private int mMaxChildHeight = 0;        public DashboardLayout(Context context) {          super(context, null);      }        public DashboardLayout(Context context, AttributeSet attrs) {          super(context, attrs, 0);      }        public DashboardLayout(Context context, AttributeSet attrs, int defStyle) {          super(context, attrs, defStyle);      }        @Override      protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {          mMaxChildWidth = 0;          mMaxChildHeight = 0;            // Measure once to find the maximum child size.            int childWidthMeasureSpec = MeasureSpec.makeMeasureSpec(                  MeasureSpec.getSize(widthMeasureSpec), MeasureSpec.AT\_MOST);          int childHeightMeasureSpec = MeasureSpec.makeMeasureSpec(                  MeasureSpec.getSize(widthMeasureSpec), MeasureSpec.AT\_MOST);            final int count = getChildCount();          for (int i = 0; i < count; i++) {              final View child = getChildAt(i);              if (child.getVisibility() == GONE) {                  continue;              }                child.measure(childWidthMeasureSpec, childHeightMeasureSpec);                mMaxChildWidth = Math.max(mMaxChildWidth, child.getMeasuredWidth());              mMaxChildHeight = Math.max(mMaxChildHeight, child.getMeasuredHeight());          }            // Measure again for each child to be exactly the same size.            childWidthMeasureSpec = MeasureSpec.makeMeasureSpec(                  mMaxChildWidth, MeasureSpec.EXACTLY);          childHeightMeasureSpec = MeasureSpec.makeMeasureSpec(                  mMaxChildHeight, MeasureSpec.EXACTLY);            for (int i = 0; i < count; i++) {              final View child = getChildAt(i);              if (child.getVisibility() == GONE) {                  continue;              }                child.measure(childWidthMeasureSpec, childHeightMeasureSpec);          }            setMeasuredDimension(                  resolveSize(mMaxChildWidth, widthMeasureSpec),                  resolveSize(mMaxChildHeight, heightMeasureSpec));      }        @Override      protected void onLayout(boolean changed, int l, int t, int r, int b) {          int width = r - l;          int height = b - t;            final int count = getChildCount();            // Calculate the number of visible children.          int visibleCount = 0;          for (int i = 0; i < count; i++) {              final View child = getChildAt(i);              if (child.getVisibility() == GONE) {                  continue;              }              ++visibleCount;          }            if (visibleCount == 0) {              return;          }            // Calculate what number of rows and columns will optimize for even horizontal and          // vertical whitespace between items. Start with a 1 x N grid, then try 2 x N, and so on.          int bestSpaceDifference = Integer.MAX\_VALUE;          int spaceDifference;            // Horizontal and vertical space between items          int hSpace = 0;          int vSpace = 0;            int cols = 1;          int rows;            while (true) {              rows = (visibleCount - 1) / cols + 1;                hSpace = ((width - mMaxChildWidth \* cols) / (cols + 1));              vSpace = ((height - mMaxChildHeight \* rows) / (rows + 1));                spaceDifference = Math.abs(vSpace - hSpace);              if (rows \* cols != visibleCount) {                  spaceDifference \*= UNEVEN\_GRID\_PENALTY\_MULTIPLIER;              }                if (spaceDifference < bestSpaceDifference) {                  // Found a better whitespace squareness/ratio                  bestSpaceDifference = spaceDifference;                    // If we found a better whitespace squareness and there's only 1 row, this is                  // the best we can do.                  if (rows == 1) {                      break;                  }              } else {                  // This is a worse whitespace ratio, use the previous value of cols and exit.                  --cols;                  rows = (visibleCount - 1) / cols + 1;                  hSpace = ((width - mMaxChildWidth \* cols) / (cols + 1));                  vSpace = ((height - mMaxChildHeight \* rows) / (rows + 1));                  break;              }                ++cols;          }            // Lay out children based on calculated best-fit number of rows and cols.            // If we chose a layout that has negative horizontal or vertical space, force it to zero.          hSpace = Math.max(0, hSpace);          vSpace = Math.max(0, vSpace);            // Re-use width/height variables to be child width/height.          width = (width - hSpace \* (cols + 1)) / cols;          height = (height - vSpace \* (rows + 1)) / rows;            int left, top;          int col, row;          int visibleIndex = 0;          for (int i = 0; i < count; i++) {              final View child = getChildAt(i);              if (child.getVisibility() == GONE) {                  continue;              }                row = visibleIndex / cols;              col = visibleIndex % cols;                left = hSpace \* (col + 1) + width \* col;              top = vSpace \* (row + 1) + height \* row;                child.layout(left, top,                      (hSpace == 0 && col == cols - 1) ? r : (left + width),                      (vSpace == 0 && row == rows - 1) ? b : (top + height));              ++visibleIndex;          }      }  } |

**6**. Now we need to create a layout file dashboard screen. Create a new xml file under **src/layouts** and name it as **fragment\_layout.xml**  
( Right Click on res/layout ⇒ New ⇒ Android XML File)

|  |
| --- |
| fragment\_layout.xml |
| <!-- Your package folder -->  <com.androidhive.dashboard.DashboardLayout      xmlns:android="<http://schemas.android.com/apk/res/android>"      android:layout\_width="fill\_parent"      android:layout\_height="fill\_parent"      android:layout\_weight="1"      android:background="#f8f9fe" >      <!--  News Feed Button -->      <Button          android:id="@+id/btn\_news\_feed"          style="@style/DashboardButton"          android:drawableTop="@drawable/btn\_newsfeed"          android:text="News Feed" />        <!--  Friends Button -->      <Button          android:id="@+id/btn\_friends"          style="@style/DashboardButton"          android:drawableTop="@drawable/btn\_friends"          android:text="Friends" />        <!--  Messages Button -->      <Button          android:id="@+id/btn\_messages"          style="@style/DashboardButton"          android:drawableTop="@drawable/btn\_messages"          android:text="Messages" />        <!--  Places Button -->      <Button          android:id="@+id/btn\_places"          style="@style/DashboardButton"          android:drawableTop="@drawable/btn\_places"          android:text="Places" />        <!--  Events Button -->      <Button          android:id="@+id/btn\_events"          style="@style/DashboardButton"          android:drawableTop="@drawable/btn\_events"          android:text="Events" />        <!--  Photos Button -->      <Button          android:id="@+id/btn\_photos"          style="@style/DashboardButton"          android:drawableTop="@drawable/btn\_photos"          android:text="Photos" />    </com.androidhive.dashboard.DashboardLayout> |

The output of above code will be



⇒ Designing Footer

**7**. Create a new xml file under **res/layout** and name it as **footer\_layout.xml** and type the following code.

|  |
| --- |
| footer\_layout.xml |
| <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"      style="@style/FooterBar" >      <TextView android:text="www.facebook.com"              android:layout\_width="fill\_parent"              android:layout\_height="wrap\_content"              android:textColor="#606060"              android:gravity="center"              android:paddingTop="10dip"/>  </LinearLayout> |

⇒ Merging all layout together

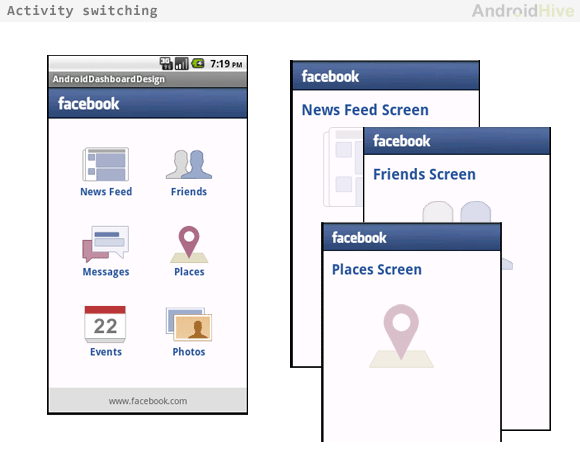
So far we designed Action bar, Dashboard and Footer. Finally we need to merge all layouts in one xml file.

**8**. Create a new xml file under **src/layouts** and name it as **dashboard\_layout.xml and type following code.**  
( Right Click on res/layout ⇒ New ⇒ Android XML File)

|  |
| --- |
| dashboard\_layout.xml |
| <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"      android:id="@+id/home\_root"      android:orientation="vertical"      android:layout\_width="fill\_parent"      android:layout\_height="fill\_parent">          <!-- Include Action Bar -->          <include layout="@layout/actionbar\_layout"/>            <!--  Include Fragmented dashboard -->          <include layout="@layout/fragment\_layout"/>            <!--  Include Footer -->          <include layout="@layout/footer\_layout"/>    </LinearLayout> |

⇒ Switching between dashboard activities.

So far we created a static dashboard which has no functioning at all. So we need to add some functionality like launching separate activity for each icon on the dashboard. In this case i had 6 icons on my dashboard, so i need 6 Activities one for each icon. In my previous article [How to switch between Activities in Android](https://www.androidhive.info/2011/08/how-to-switch-between-activities-in-android/) i explained switching between activites.

\*

**9**. Open your main activity class file and paste following code.(**AndroidDashboardDesignActivity.java**)

|  |
| --- |
| AndroidDashboardDesignActivity.java |
| package com.androidhive.dashboard;    import android.app.Activity;  import android.content.Intent;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  import androidhive.dashboard.R;    public class AndroidDashboardDesignActivity extends Activity {        @Override      public void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.dashboard\_layout);            /\*\*           \* Creating all buttons instances           \* \*/          // Dashboard News feed button          Button btn\_newsfeed = (Button) findViewById(R.id.btn\_news\_feed);            // Dashboard Friends button          Button btn\_friends = (Button) findViewById(R.id.btn\_friends);            // Dashboard Messages button          Button btn\_messages = (Button) findViewById(R.id.btn\_messages);            // Dashboard Places button          Button btn\_places = (Button) findViewById(R.id.btn\_places);            // Dashboard Events button          Button btn\_events = (Button) findViewById(R.id.btn\_events);            // Dashboard Photos button          Button btn\_photos = (Button) findViewById(R.id.btn\_photos);            /\*\*           \* Handling all button click events           \* \*/            // Listening to News Feed button click          btn\_newsfeed.setOnClickListener(new View.OnClickListener() {                @Override              public void onClick(View view) {                  // Launching News Feed Screen                  Intent i = new Intent(getApplicationContext(), NewsFeedActivity.class);                  startActivity(i);              }          });           // Listening Friends button click          btn\_friends.setOnClickListener(new View.OnClickListener() {                @Override              public void onClick(View view) {                  // Launching News Feed Screen                  Intent i = new Intent(getApplicationContext(), FriendsActivity.class);                  startActivity(i);              }          });            // Listening Messages button click          btn\_messages.setOnClickListener(new View.OnClickListener() {                @Override              public void onClick(View view) {                  // Launching News Feed Screen                  Intent i = new Intent(getApplicationContext(), MessagesActivity.class);                  startActivity(i);              }          });            // Listening to Places button click          btn\_places.setOnClickListener(new View.OnClickListener() {                @Override              public void onClick(View view) {                  // Launching News Feed Screen                  Intent i = new Intent(getApplicationContext(), PlacesActivity.class);                  startActivity(i);              }          });            // Listening to Events button click          btn\_events.setOnClickListener(new View.OnClickListener() {                @Override              public void onClick(View view) {                  // Launching News Feed Screen                  Intent i = new Intent(getApplicationContext(), EventsActivity.class);                  startActivity(i);              }          });            // Listening to Photos button click          btn\_photos.setOnClickListener(new View.OnClickListener() {                @Override              public void onClick(View view) {                  // Launching News Feed Screen                  Intent i = new Intent(getApplicationContext(), PhotosActivity.class);                  startActivity(i);              }          });      }  } |

**10**. Create a new class under res/package. **Right Click on src/package folder ⇒ New ⇒ Class** and name it as **NewsFeedActivity.java** and fill it with following code.  
You also need to create  
**NewsFeedActivity.java  
FriendsActivity.java  
MessagesActivity.java  
PlacesActivity.java  
EventsActivity.java  
PhotosActivity.java**

|  |
| --- |
| NewsFeedActivity.java |
| package com.androidhive.dashboard;    import android.app.Activity;  import android.os.Bundle;  import androidhive.dashboard.R;    public class NewsFeedActivity extends Activity {       /\*\* Called when the activity is first created. \*/      @Override      public void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.news\_feed\_layout);      }  } |

**11**. Also create a layout xml files for all your activities.  
**news\_feed\_layout.xml  
friends\_layout.xml  
messages\_layout.xml  
places\_layout.xml  
events\_layout.xml  
photos\_layout.xml**

|  |
| --- |
| news\_feed\_layout.xml |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:background="#f8f9fe"      android:orientation="vertical" >        <include layout="@layout/actionbar\_layout" />        <LinearLayout          android:layout\_width="fill\_parent"          android:layout\_height="wrap\_content" >            <TextView              android:layout\_width="fill\_parent"              android:layout\_height="wrap\_content"              android:padding="15dip"              android:text="News Feed Screen"              android:textColor="#ff29549f"              android:textSize="25dip"              android:textStyle="bold" />      </LinearLayout>    </LinearLayout> |

Adding Activity entries in AndroidManifest.xml

**12**. Dont’t forget to add an entry of all activities in your AndroidManifest.xml file. Open you AndroidManifest.xml file and modify the code as below

|  |
| --- |
| AndroidManifest.xml |
| <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="<http://schemas.android.com/apk/res/android>"      package="androidhive.dashboard"      android:versionCode="1"      android:versionName="1.0" >        <uses-sdk android:minSdkVersion="8" />        <application          android:icon="@drawable/ic\_launcher"          android:label="@string/app\_name" >          <activity              android:label="@string/app\_name"              android:name="com.androidhive.dashboard.AndroidDashboardDesignActivity" >              <intent-filter >                  <action android:name="android.intent.action.MAIN" />                    <category android:name="android.intent.category.LAUNCHER" />              </intent-filter>          </activity>            <!-- News Feed Activity -->          <activity android:name="com.androidhive.dashboard.NewsFeedActivity" ></activity>            <!-- Friends Activity -->          <activity android:name="com.androidhive.dashboard.FriendsActivity" ></activity>            <!-- Messages Activity -->          <activity android:name="com.androidhive.dashboard.MessagesActivity" ></activity>            <!-- Places Activity -->          <activity android:name="com.androidhive.dashboard.PlacesActivity" ></activity>            <!-- Events Activity -->          <activity android:name="com.androidhive.dashboard.EventsActivity" ></activity>            <!-- Photos Activity -->          <activity android:name="com.androidhive.dashboard.PhotosActivity" ></activity>      </application>    </manifest> |

Run your project by **right clicking on your project folder ⇒ Run As ⇒ 1 Android Application**.