

# MOBILE PACKET MONITOR FRONT END



By Josue Martins

Supervisor Dr. W.B Tucker

& Co-Supervisor Mr. M. Norman



# Introduction

## What is an Mobile Packet Monitor software?

- Will monitor the data usage on a mobile phone.
- It will store the information about the data used and present to the user in organized manner.
- This will help mobile phone users make a well informed decision about the data usage on the Mobile Phone.

## Why is the Mobile Packet Monitor important?

- Mobile users aren't aware of how much they spend on their mobile.
  - It will provide awareness of data Usage and cost.
  - It will help users make better financial decisions on internet Data usage thus reducing cost of using Internet on a mobile phone.

## Mobile users



## Expensive Mobile phones Bills

MESSAGING/DATA CHARGES	
Text Messaging Madness plan	\$2.99
Text messaging surcharges	\$0.90
Data Madness plan	\$15.00
<b>Total:</b>	<b>\$18.89</b>
TAXES, FEES, AND SURCHARGES	
Federal tax	\$1.39
FCC universal service charge	\$0.82
State and local charges	\$4.85
Regulatory fees	\$0.30
Administrative charges	\$0.40
<b>Total:</b>	<b>\$8.36</b>
INSURANCE	
Handset insurance	\$3.00
<b>Total:</b>	<b>\$3.00</b>
<b>TOTAL AMOUNT DUE:</b>	<b>\$116.48</b>

## Solution Android packet monitor



# Design

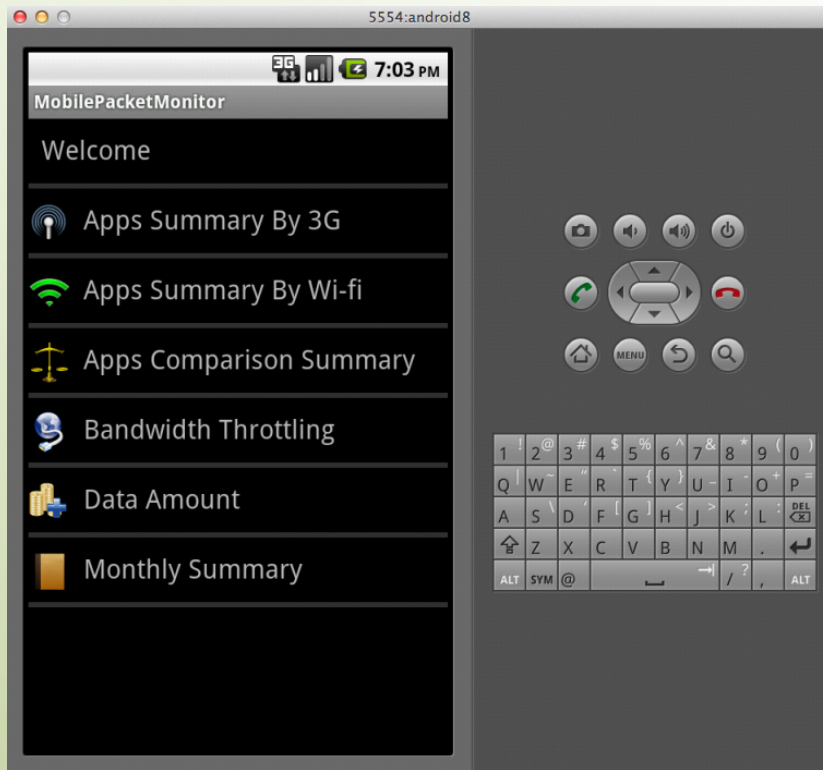


## User interface Specification Goals

- Don't make our users look stupid.
  - Not making users guess at what the program is doing.
  - Provide good documentation on the menu like proper titles.
- Let users have fun (or at least not get too bored).
  - providing summaries in an educational manner.
  - Icons.
- Don't allow users to make big mistakes.
  - Providing detailed feedback and visual results so users can understand the software better.
- To facilitate the user to get work done effortless and in the timely manner.
  - providing good performance.
  - Providing shortcut keys to access commonly-used features.



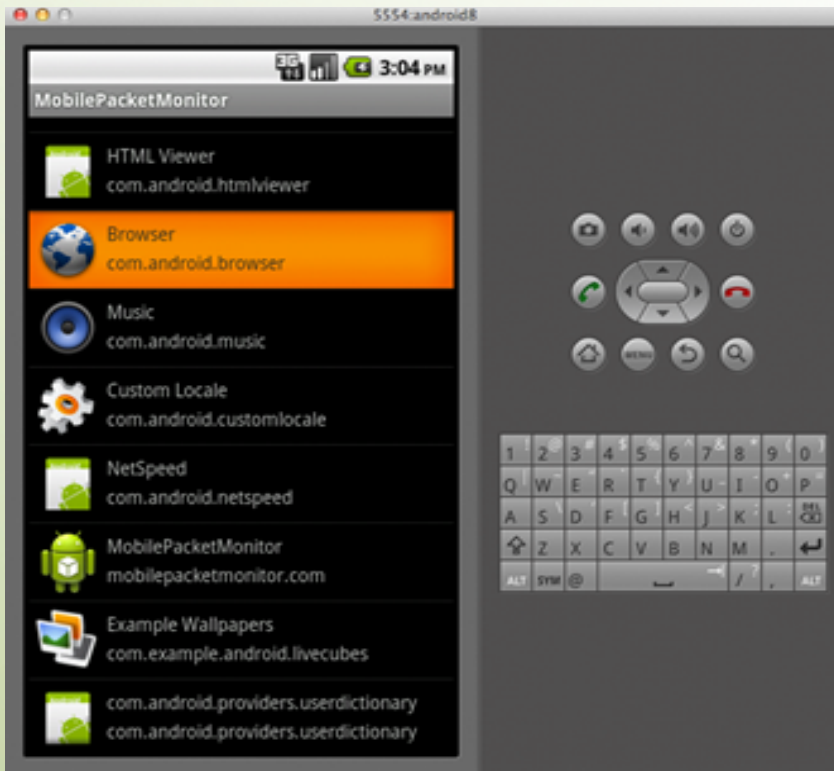
# We recap in Requirement Analysis and User Interface Specification



- Simple User Interface.
- full Screen run time and Touch screen interaction.
- Let users have fun or at least not get too bored.
  - • providing summaries in an educational manner.
  - • Icons.

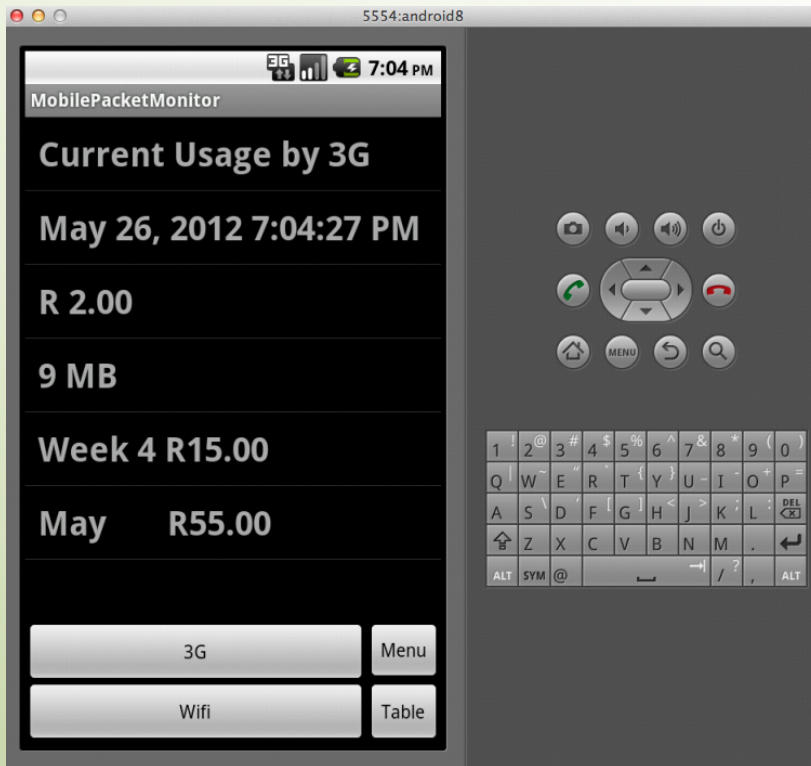


# Recap in Requirement analysis



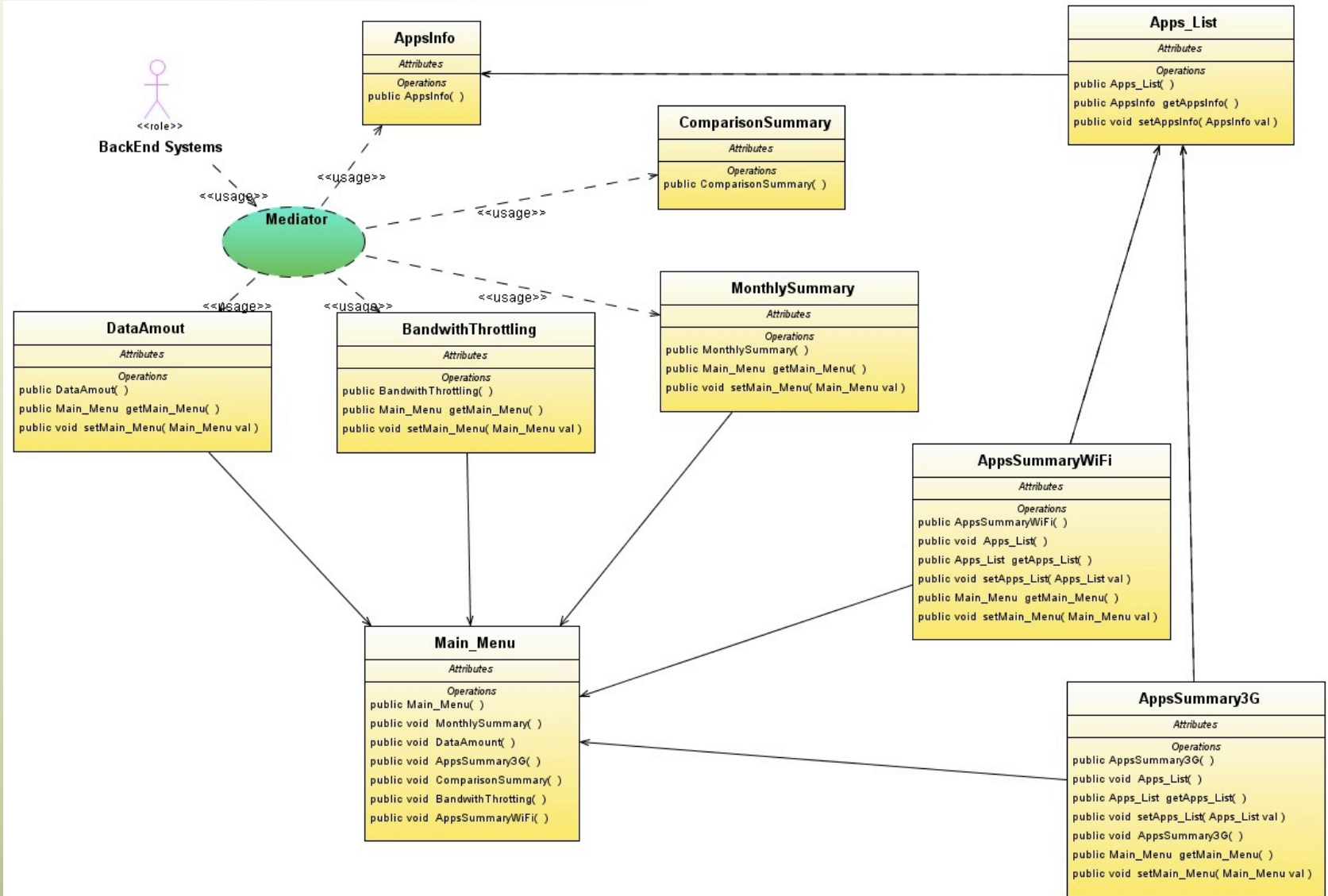
- Bigger Buttons so that users cant have difficulties clicking the application to be viewed.

# Recap in User Requirement and User Interface Specification



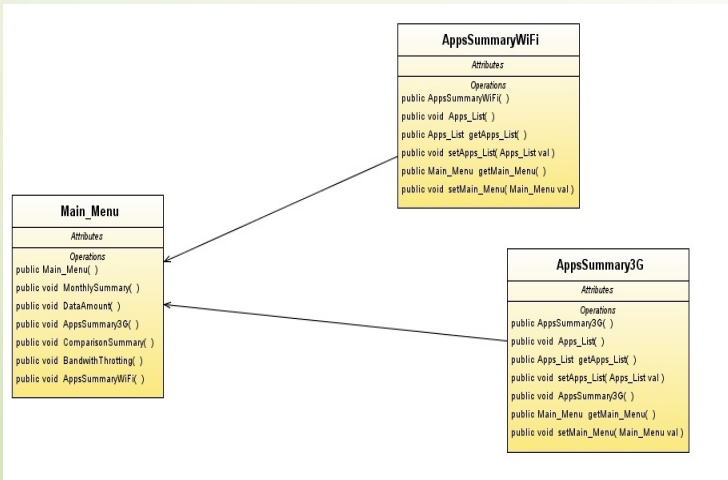
- Don't allow users to make big mistakes.
  - Providing detailed feedback and visual results so users can understand the software better.
- Shortcuts.
  - To facilitate the user to get work done effortless and in the timely manner.
  - Pressing back all the time can be really frustrating.

# High Level Design





# Low level Design

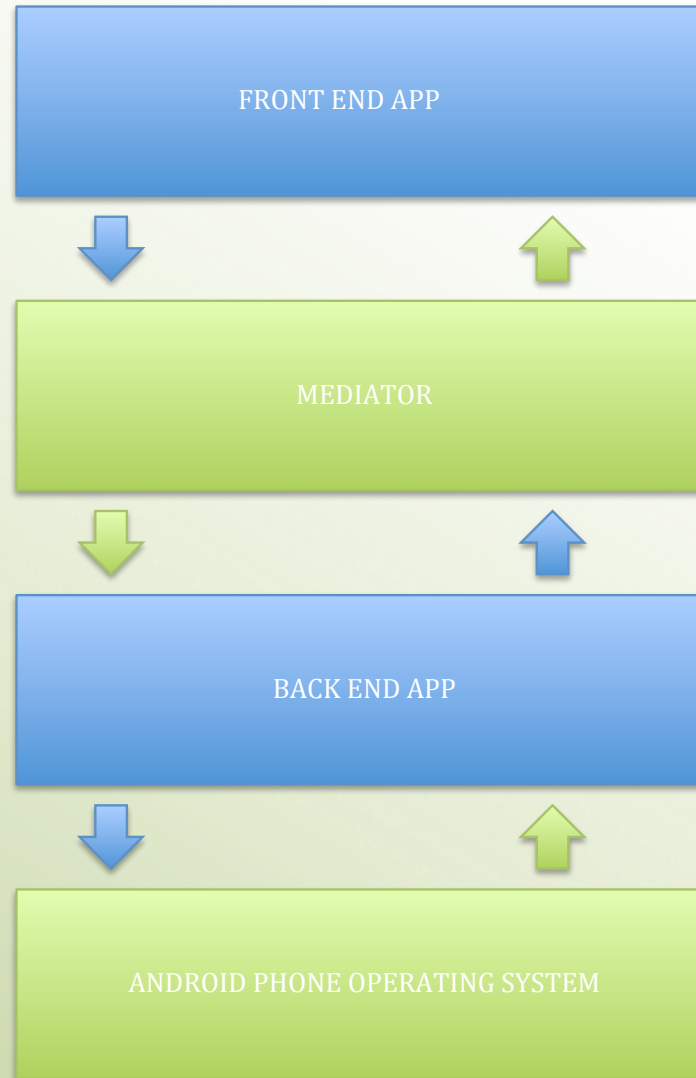


@Override

```

public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);    setContentView(R.layout.main);
    setListAdapter((ListAdapter) new ArrayAdapter<String>
        (this,android.R.layout.simple_list_item_1,getResources().getStringArray(R.array.Menu_a
        rray)));//caling array list of the menu    ListView lv= getListView();
    lv.setOnItemClickListener(new OnItemClickListener(){
        public void onItemClick(AdapterView<?>parent,View view,int position,long id)
        {
            //each position lets the list view call
            one activity according ot the indeox the of array
            if(position==0)
            {
                Intent lp=new
                Intent(getApplicationContext(),Summary3g.class);//currently working with 3g class
                lp.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(lp);
            }
            if(position==1)
            {
                Intent lp=new
                Intent(getApplicationContext(),SummarybyWifi.class);
                lp.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(lp);
            }
        }
    });
}
  
```

# Front End integration with Back End



# Tools used to bring the Prototype to life

- Eclipse
- Netbeans
- Android SDK and Emulator
- Android Mobile Phone
- Java Programming language
- Java SDK
- XML



Demo

Project Plan 2012

Term 1	Feb	April 04	May 30	Sep 12	Nov 7
<b>Project Analysis</b>					
Researching for a Possible Project					
Learning Android development					
Interviewing users					
Adding the findings on the research to the final Document.					
Finalizing the Documentation.					
First Term Presentation					
<b>Term 2</b>					
<b>Project Design and Development</b>					
Designing & analyzing the UI					
Developing prototype					
Coding the solution					
Term Presentation					
<b>Term 3</b>					
<b>Project Implementation</b>					
Coding the Solution					
Testing incrementally as we code					
Term presentation					
<b>Term 4</b>					
<b>Project Testing, Evaluation and Presentation</b>					
Coding the Testing the Final Product					
Deployment					
Final Presentation					

# References

**Android Developers. Retrieved April 03, 2012, from <http://developer.android.com/guide/basics/what-is-android.html>**

**Huang, Y. (2009, September 14). BEGIN ANDROID JOURNEY IN HOURS. Retrieved April 03, 2011, from <http://www.cs.uiuc.edu/class/fa09/cs425/mps/tutorial.pdf>**

**Marsden, M. J. & Gary Marsden (2005). Mobile Interaction Design. In M. J. Marsden, Mobile Interaction Design (p. 5). Wiley.**

**Matias Duarte, R. F. , Rich Fulcher, Roman Nurik, Adam Powell & Christian Robertson(2011). Google 11 IO. Retrieved April 03, 2011, from [http://static.googleusercontent.com/external\\_content/untrusted\\_dlcp/www.google.com/en//events/io/2011/static/presofiles/designing\\_and\\_implementing\\_android\\_uis\\_for\\_phones\\_and\\_tablets.pdf](http://static.googleusercontent.com/external_content/untrusted_dlcp/www.google.com/en//events/io/2011/static/presofiles/designing_and_implementing_android_uis_for_phones_and_tablets.pdf)**





# The End Questions?

