

T-110.5130 Mobile Systems Programming Project Specification

A.M. Ahsan Feroz
abdullah.feroz@aalto.fi

Md. Mohsin Ali Khan
md.m.khan@aalto.fi

Elena Oat
elena.oat@aalto.fi

Hasan M. A. Islam
hasan.islam@aalto.fi

Mesbahul Islam Nahin
mdislam@cs.helsinki.fi

Tutor: Antero Juntunen

May 6, 2013

Chapter 1

Introduction

Smartphones and tablets are ubiquitous everywhere. Typically, a smartphone is a mobile phone built on a mobile operating system, with more advanced computing capability and connectivity than traditional phone. The rapid adoption of mobile platforms is being driven by the numerous applications that allow users to perform everyday transactions. The examples of mobile operating systems (OS) used by modern smartphones are Google's Android, Apple's iOS, Nokia's Symbian, etc. Android is an open-source platform founded in October 2003 by Andy Rubin and backed by Google, along with major hardware and software developers (e.g., Intel, HTC, ARM, Motorola, Samsung). We are very enthusiastic to build application in Android for the course 'Mobile Systems Programming'. The main motivation behind choosing Android platform are the followings:

- Android is the most widely used platform for mobile respecting the number of users and number of applications.
- Android app developers use the classic open source Linux OS. Any open source provides help for users to access its source code transparently and is available to any developer who wants to modify it or see how it works.
- Google provides all Android developers with an open source software development kit for the Android OS. They can create applications for Android and then test them on an Android simulator before loading them onto an actual Droid phone.
- The future of the Android platform looks bright for Android application developers who love to have access to everything. We can expect to see some interesting innovations both from Google and from the public as Android gains popularity.

Nevertheless, our rudimentary plan is to develop a 'Task Manager' which is initially named as 'AndroTasker'. Now-a-days, people are so busy with their daily activities and hence, it is more likely for one to forget a schedule that might be very important. In the present hi tech world, mobile phones could be a great medium to keep track of the all the things and notify the user in the right time to perform a task. The very first version of our Android Task Manager will be a typical task manager which would be able to maintain and

keep track of all of our schedules of every moment. The user can give temporal input to our application and the app will be responsible to remind the user by means of alarm about the temporal events. The reminder can be set for each minutes of a day and in a year. The application will always be running in the background and the reminder will pop up in a small screen at the desired time. The application will also have a view screen which can be used as an widget in the desktop with the list of tasks.

1.1 Goals and scope

The primary goal of the project is to learn how to develop a mobile application for Android platform. We want to make sure that all of the group members understand the basic building blocks of an Android Mobile application along with their significance and underlying concepts. We want to make sure that we understand how an Android application interacts with user. And back to back how it interacts with the Android Operating System to get the job done that is requested by the user. We want to learn how to publish an Android application in the market. The secondary goal of the application is to make sure that we understand how to interact with other Android applications. The scope of the project is to develop an application that provides the user functionality to do the following activities:

1. View Existing Events
 - Day view
 - Week view
 - Month view
2. Create New Events
3. Edit Existing Events
4. Alarm and Bar notifications of the Events at the due time

1.2 Why Andro Task Manager

The motivation behind choosing the task manager to develop is to learn many aspects of the android development environment, such as

- How to manage Activity Life Cycle in Android.
- Building a flexible UI using **Fragments**
- design app's screen hierarchy and forms of navigation so that users can effectively and intuitively traverse our app content using various navigation patterns.
- How to store user and app data (e.g., SQLite)
- How to interact with other apps (using **Intent**) i.e., how to exploit Android's one of the most important feature to send the user to another app based on "action" (secondary goal).

Respecting the type of our application, it can be said that this is useful for all range of people who use smart phones smartly and hopefully the number is countless. Task manager can help to manage every aspect of managing task. It reduces the administrative tasks. It will reduce paperwork. It will help to manage resources with greater control.

Chapter 2

Work practices and tools

In this chapter, all planned work practices for our project will be outlined in brief such as how our group works, how the documentation of our work will be assembled so that each and every member could be able to know the events though one may be absent for any particular reason.

2.1 Time Tracking

For tracking our progress and communication, we mostly use Facebook. A private group was created, which includes all the members of our group. FB is our main platform to agree on when/where the next meeting is, as well as share our thoughts and useful links.

2.2 Meetings and Documentation

We meet once a week, as a rule. Meeting duration is for three hours. We have also met once for the whole day, when we were working on the UI of our application. The time we spent is written down and is tracked by the project leader.

We haven't had to deal with documentation very much, as we make decisions during the meeting and since every member is present, we are all up to date with the situation. For other cases we use Google Docs. And again, some decisions are made on Facebook group wall and all people who have access could follow up with the process.

In general, we don't have a clear division on who will do what, we just have project leader who coordinates the meetings times and does other organizational task.

2.3 Version Control

Git is a free and open source distributed version control system designed to efficiently handle everything from small to very large projects with a good speed. Git is easy to learn and has a tiny footprint with lightning fast performance. It subordinates SCM tools like Subversion, CVS, Perforce, and ClearCase with

features like cheap local branching, convenient staging areas, and multiple workflows.

We have decided to use **Github** for sharing code and other resources and controlling versions. Every group member has an account on **Github**. A repository was created, which serves as a place for common development. All commits are accompanied by a message which states what changes have been made to the previous version.

2.4 Design

In order to better understanding, we are going to develop custom calendar day view, week view and month view. Android development kit provides **Layout** such as **Relative Layout**, **List View**, **Grid View**, **Table Layout**. In some UI design, **Grid View** seems advantageous. We can also use the combination of these making our UI attractive.

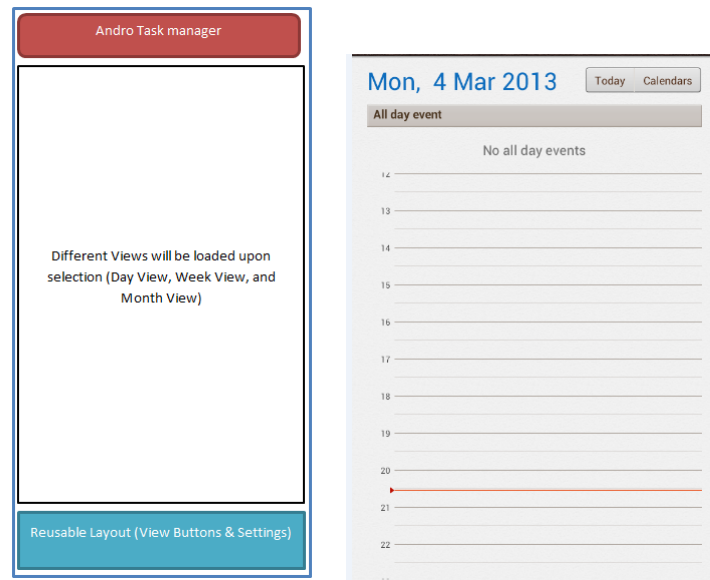


Figure 2.1: High level view and Day view

Figure 2.1 provides the core fundamental UI for our project. Different view will be loaded upon user "action". At the lower portion, it indicates **Reusable layout** of Android feature. Android offers a variety of widgets to provide small and re-usable interactive elements to efficiently reuse complete layouts and embed another layout inside the current layout. Reusing layouts is particularly powerful as it allows you create reusable complex layouts that is expected for our desired project. It also will provide help for our application to extract commonalities across multiple layouts, managed separately.

We ideate the week view and month view as like Figure 2.2. User can store their task in our app. Android provides several options to save persistent application data. As app data should be **private**, we can utilize **Shared**

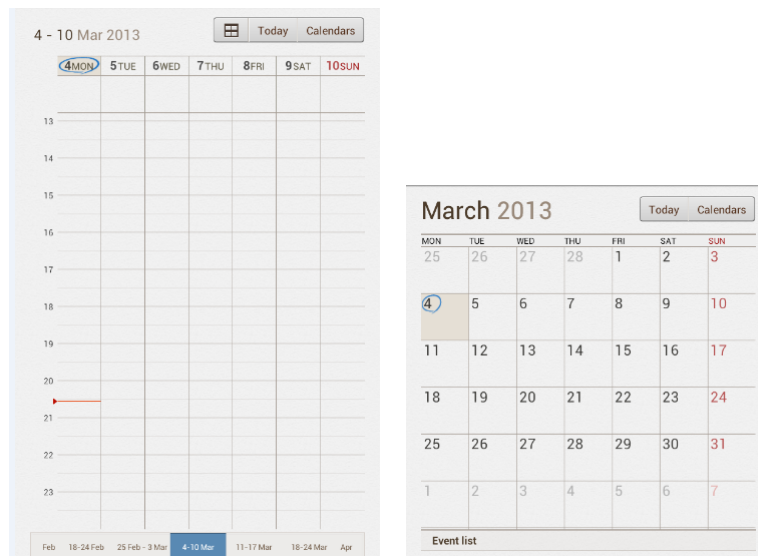


Figure 2.2: Week and Month view UI

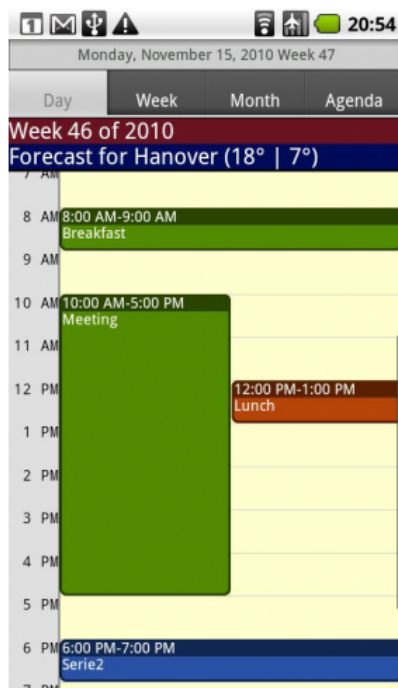


Figure 2.3: Scheduler Sample (Imaginary)

Preferences and SQLite Storage. Our app should not require that much storage. Figure 2.1 represents an imaginary figure with user's stored tasks. UI will

use `GridView` or `Listview` displaying items in a two-dimensional, scrollable grid. The grid items are automatically inserted to the layout using a `ListAdapter`.

For testing purposes we use our own devices (Android smartphones), as this provides an easier and faster way to check the latest changes.

2.5 Quality Assurance

We try to only follow the official tutorials and documentation that is provided by developer.android.com and avoid amateur blogs. However, this is sometimes challenging, because not all answers are available on official Android's website. YouTube channels help also in learning basics, e.g. Marakana's channel. We try to also read the best practices that were set by Android community, but sometimes it is hard to do it in the beginning, because you don't know yet all the concepts.

2.6 Tools

Tools that will be used for this project are the followings:

- i. Android devices (Android 2.3.5, Android 4.0, Android 4.1.2)
- ii. Eclipse for Mobile Developers (Version: Juno Service Release 1) + ADT
- iii. GIT, Github application on Windows
- iv. Google docs
- v. Facebook group.

2.7 Schedule and resources

Task	Deadline	Required Hours	Load Distribution per person
Finalizing The Application to be developed	25-Jan-13	10	2
Finalizing the Scope of the Application	31-Jan-13	10	2
Finalizing the User Interface, Data Flow and Browsing Sequence	10-Feb-13	30	6
Environment Setup for Coding	15-Feb-13	25	5
Understanding the Basic Building Blocks of an Android Application and Developing a Test Application	25-Feb-13	25	5
Developing the Designed User, Interface	15-Mar-13	50	10
Developing the Underlying Events, Implement the Data Flow and Browsing Sequence	15-Apr-13	150	30
Preparing Test Case of the Application	20-Apr-13	25	5
Testing1	25-Apr-13	25	5
Bug Fixing 1	27-Apr-13	40	8
Testing2	30-Apr-13	25	5
Bug Fixing 2	5-May-13	40	5

Table 2.1: Scheduling and Load Distribution.

Chapter 3

System overview

To create, edit (Update or Delete) or view an event an user first select a suitable view: day, week or month view. If user choose day view then a listview with hours of the selected day will be shown. User can select a time slot to create new event or edit existing event.

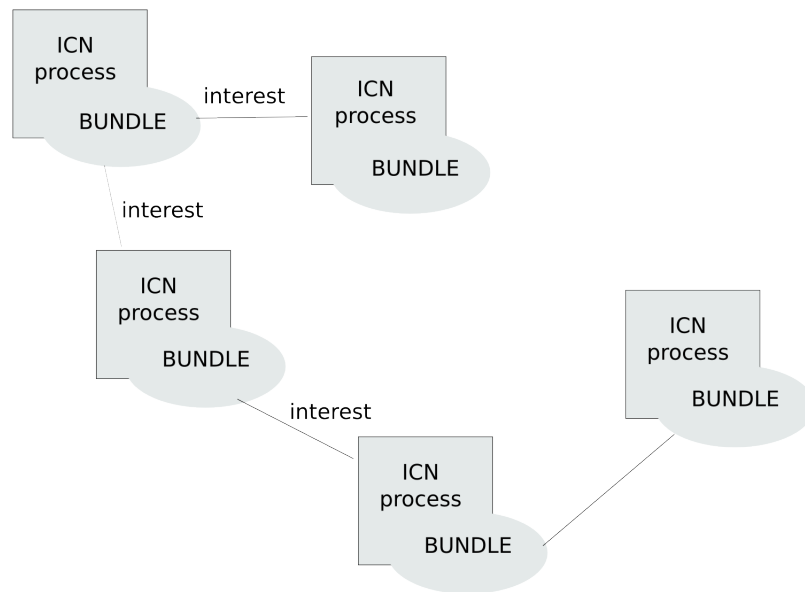


Figure 3.1: Basic System Overview

User will be notified based on the stored tasks into our app's database.

3.1 Requirements

Functional requirements states what the system can do. Some functional requirements are as follows:

Functional Requirements :

1. **Display:** AndroTask Manager will pose the following views for users.
 - Day view
 - Week view
 - Month view
2. **Users Action:** An user can do the following actions:
 - Create Event
 - Edit Event
 - View Event
 - Delete Event
3. **Notification** There will be notification prior to an event.
 - alarm notification
 - visual notification on mobile screen
4. **Settings:** Users can set options according to their preference:
 - User can set alarm type and sound
 - User can set weekdays and weekends
5. **onStart** Application will starts with Day View of current date.
6. **Default** If there are existing event(s) of current date then the event(s) will load and show by default

Non-Functional Requirements: qualities of the system.

- A user can't create an event for a day that has passed away already
- The design is user friendly