ANDROID APPLICATION OF QUICK ORGANIZER

By

POONAM SATISH BAXI

B.E, Maharashtra Institute of Technology College Of Engineering, India, 2008

A REPORT

Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2012

Approved by:

Major Professor
Dr. Daniel Andresen

Abstract

The aim of this project is to develop an Android application for managing and organizing daily activities. Mobile application development is a growing trend in computer industry. Lot of desktop applications is now becoming available as mobile applications with increasing demand in market. Android is one of the most popular platforms in mobile technology and gives lot of space for creative development as it is open source. There are various discussion forums and official Android development support websites that encourages mobile and tablet application development.

The Quick Organizer application provides three main features for managing and organizing everyday tasks.

1. Calendar with smooth navigation buttons and flexible layout to view all calendar events. This calendar synchronizes with Google calendar and calendar application in Android device and allows user to create events in all the user calendar accounts.
2. Notes management for creating new note, viewing all notes for current month and search notes for the user account that synchronizes with the Ever Notes application.
3. Tasks management to create daily to-do lists with deadline for every task. The user can see all the tasks created with clear demarcation between complete and incomplete tasks with help of strike-out tasks when completed.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to take this opportunity to thank my major professor Dr. Daniel Andresen for his valuable guidance and encouragement throughout the project.

I would also like to thank Dr. Mitchell Neilsen and Dr. Doina Caragea for graciously accepting to serve on my committee. My special thanks to Dr. Mitchell Neilsen for providing an Android device for testing of my project.

I would also like to thank all my professors of CIS department for giving me an opportunity to learn new and innovative things and expand my knowledge base. I would also like to thank my boss Dr. Brian McCornack from Entomology department for giving me opportunity to gain hands on experience in Android development.

Chapter 1 - Introduction

The project is an Android application of personal organizer targeting common people for scheduling their events and everyday tasks. The main aim of this project is to provide one mobile application for organizing meetings, scheduling events and tasks in calendar, synchronizing calendar events with Google calendar, creating and managing notes in integration with Ever Notes cloud service and tracking everyday tasks.

Chapter 2 – Motivation

The main motivation of this project is to learn mobile application development and build an application that manages routine tasks with ease. There are several free applications for Android phones and iPhone in market for personal organizer and scheduler. After reviewing the various applications available, I found out that there is no application that does various organizing tasks together in one interface. This motivated me to build an application that provides tools to organizer events in calendar, integrates and syncs with Google calendar as well as local phone calendar, manage sticky notes in synchronization with Ever Notes mobile application and track everyday tasks.

Chapter 3 – Requirement Analysis

3.1 Requirements Gathering

The project needed several requirements to be gathered in order to provide a user friendly Graphical User Interface (GUI) for organizing stuff without asking users to remember their actions. I reviewed several mobile applications in market and a closer look towards the screen design helped me decide my design of the screens in the application. One of the important requirement analysis involved choosing a suitable android platform version which is compatible with most of the Android phones and Android tablets in market today. Market survey shows that around 63% Android devices use version of 2.0 and higher, hence this project is developed on version 2.3.3 Gingerbread platform.

3.2 Requirement Specification

3.2.1 Software Requirements

For Development:

Operating System: Windows 7

Language: Android SDK, Java

Database: SQLite

Tools: Eclipse SDK 3.5

Technologies used: Java, SQLite, Android.

Debugger: Android Dalvik Debug Monitor service

For running the application:

Operating System: Android 2.3 or higher versions

3.2.2 Hardware Requirements

For Development:

Processor: P IV or higher

RAM: 256 MB

Space on disk: minimum 250MB

For running the application:

Device: Android version 2.3 and higher

Minimum space to execute: 5.0MB

Chapter 4 - System Architecture and Design

4.1 System architecture

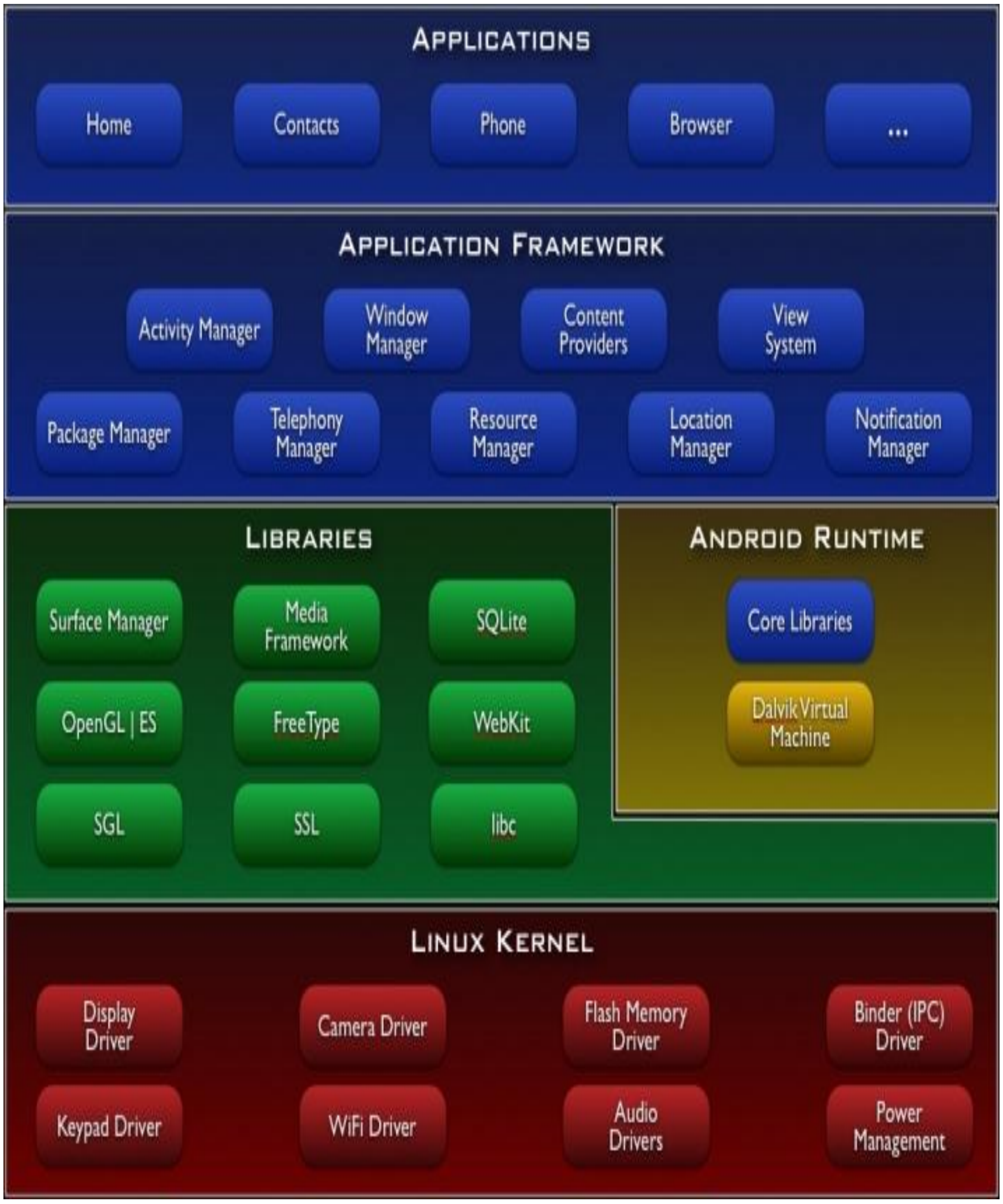


Figure 1: Android architecture

(Media Wiki, 2011)

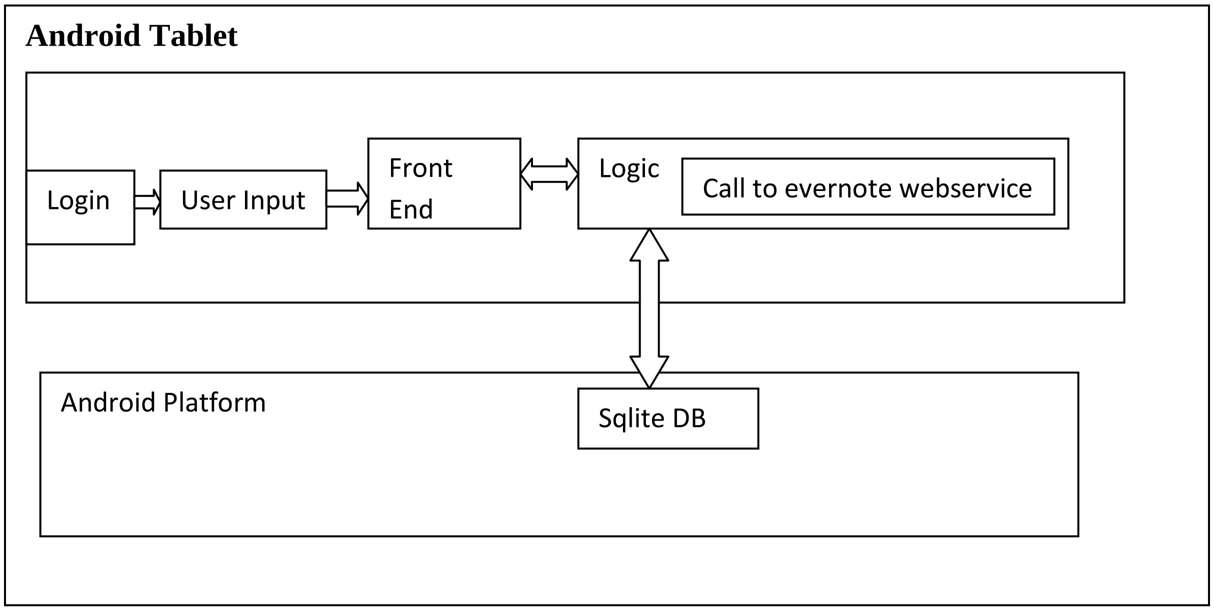


Figure 2: System Architecture

The above diagram explains the system architecture of this application. The input to this application is via touch event of user. There are 3 main screens to capture user interaction with the application. Based on the touch event of specific widget, an appropriate action takes place in background and expected response is displayed back to the user. The application runs on Android platform which in present in the Android device. The application also interacts with external cloud service provided by Ever Notes application which is one of the top 5 applications for managing notes on Android devices.

The application stores and retrieves tasks created by user from local Android SQLite database. There is a database interaction module that performs different database operations on local device database and also maintains information about the tasks completed by user. The login module provides a secure login to the application and also requires the user to register for the first time use. The application provides remember login facility so that the user on the device does not have to login into the application every time.

4.2 System Design

The requirement gathereing is followed by system design which is captured using Unified Modelling Language(UML) diagrams. The two main views of a system model are captured using below diagrams.

Static(Structurel view) – Class diagram

Behavioral view – Use Case diagram

4.2.1 Use Case Diagram

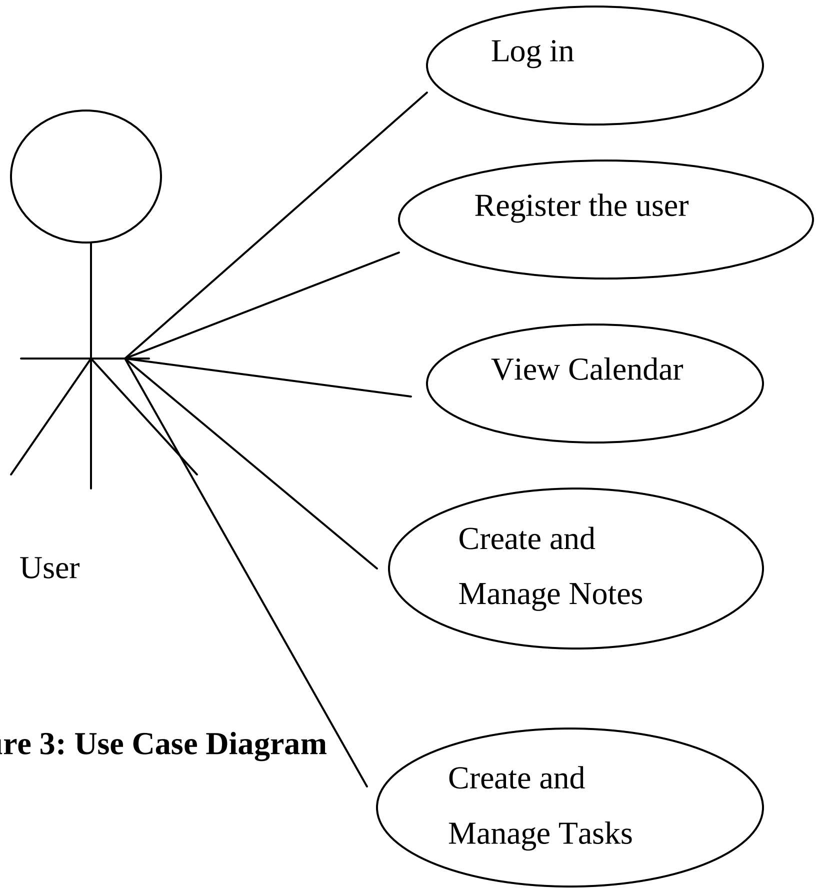
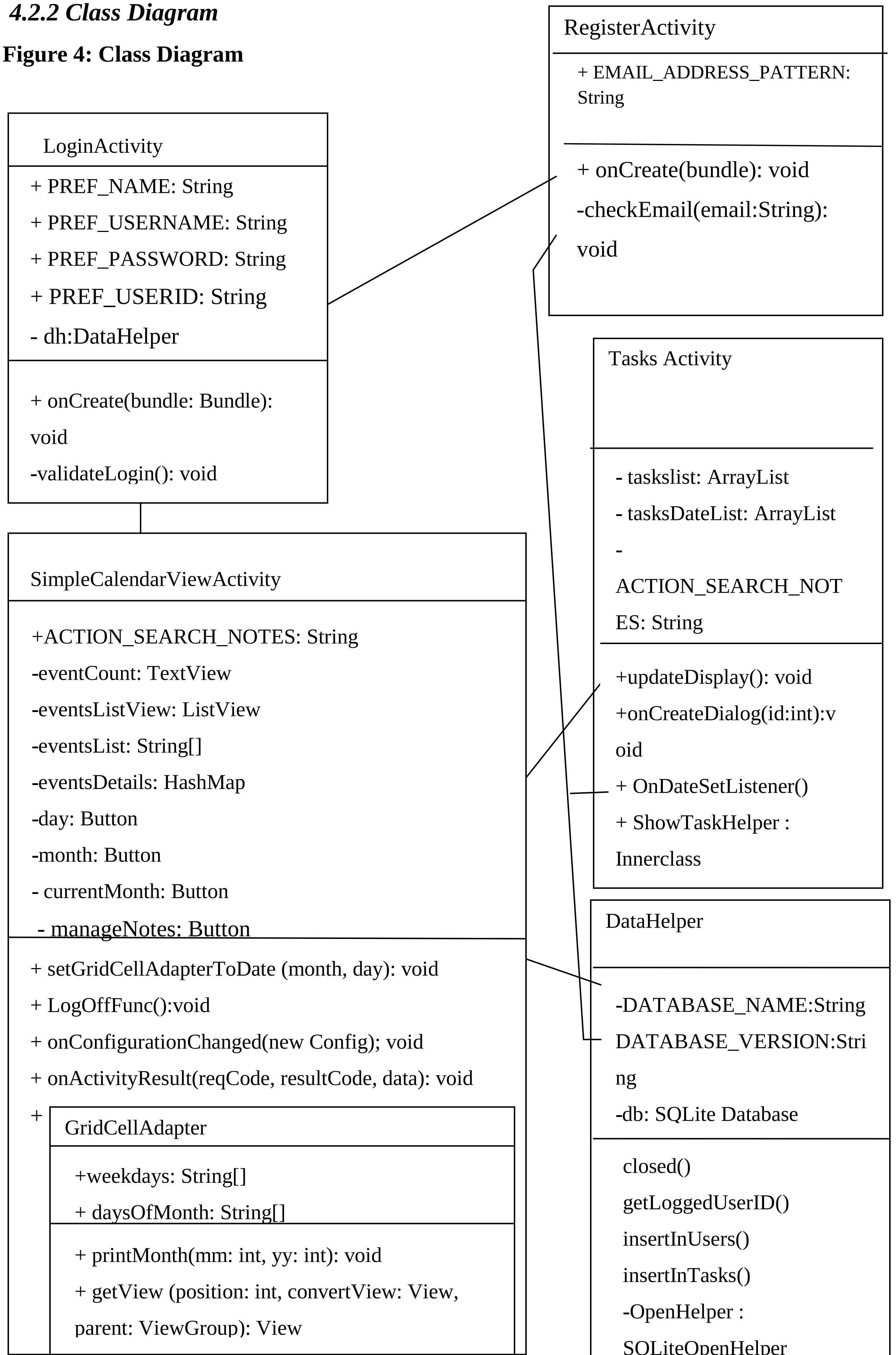


Figure 3: Use Case Diagram

4.2.2 Class Diagram

Figure 4: Class Diagram



The Class Diagram depicts static structure of the system using objects, attributes, operations and relationships. This diagram gives an inside viewof the system with different attributes and methods inside various classes andtheir interaction with each other. Every use case maps to one screen which is represented by an activity. In Android, each activity is a class with Graphical User Interface and code behind the screen. To maintain modularity of the application, each such activity is converted into a class. This keeps each feature implementation in separate class and also allows loose coupling between the different modules of the application. The above functionality is captured with help of below main activities:

Login Activity

This activity provides a first level login to the application with the help of username and password. The activity also implements functionality of *remembering users* with help of *Shared Preferences* concept in Android. This avoids re-entering of login credentials every time a user logins into the application.

Register Activity

This activity handles the registration of account for the first time users. The registration involves creation of new username with user choice password and a valid email id. This user information is stored in the local device database. The main reason behind not keeping the data on any shared server is as this application aims at providing a personal organizer for maintaining the user's personal activities which they might wish to have on their personal device. Even though the account of user is private to his/her mobile/tablet, the events created by the user are added to his/her Google calendar based on user's choice. This helps keeping the events data secured and is not affected by device crash and can be accessed even via Google account. Also the notes created by user are stored in the Ever Notes account of the user and can be accessed via Ever Notes application from anywhere.

SimpleCalendarViewActivity

This activity is the main activity in the application. The activity displays a sophisticated calendar with day and month view. The user interface of calendar is similar to the Google calendar's interface taking into account user's ease of operation. This activity displays a scrolling list of events of that particular date chosen by the user and provides a view of entire event details. A smooth navigation facility is provided to navigate between notes, tasks management and calendar. Notes activity makes use of intents concept in Android and connects to the Ever Notes web service. This view displays all the notes of user for the current month. There is facility to search notes based on keyword, date and commonly used grammar. The view of notes can be changed based on title, date created, date modified and location of note creation. The user can create new notes in the same interface which helps user save from tough job of remembering any sequence of events to perform.

Tasks Activity

This activity allows user to create new tasks, view all created tasks with clear demarcation between complete and incomplete tasks. Task creation makes use of rich GUI for choosing date in calendar and adding tasks to the list of tasks.

Data Helper

This class performs all the database operations for interaction of application with the SQLite database in an Android device.

Chapter 5 - Android Framework Components

5.1 AndroidManifest.xml file

Every application must have an AndroidManifest.xml file in its root directory. The manifest presents essential information about the application to the Android system, information the system must have before it can run any of the application's code. The manifest gives information about all the activities in the application along with intent and intent-filters. It also states all the permissions required by the application like access to internet along with read and write from and to the calendar. The xml also states the minsdk version of this application for stating its compatibility with various Android platform versions.

```
<? xmlversion="1.0"encoding="utf-8"?>
<manifestxmlns:android="http://schemas..com/apk/res/android"
package="com.oranizer.android.activity"
android: versionCode="1"
android:versionName="1.0">
<uses-sdkandroid:minSdkVersion="10"/>
<uses-permissionandroid:name="android.permission.WRITE_CALENDAR"></uses-permission>
<uses-permissionandroid:name="android.permission.INTERNET"></uses-permission>
<uses-permissionandroid:name="android.permission.READ_CALENDAR"></uses-permission>
<applicationandroid:icon="@drawable/icon"android:label="@string/app_name"android:debugg
able="true">
<activityandroid:name=".LoginActivity"
android:label="@string/app_name"android:windowSoftInputMode="stateHidden">
<intent-filter>
<actionandroid:name="android.intent.action.MAIN"/>
<categoryandroid:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>
        <activityandroid:name=".RegisterActivity"
android:label="@string/app_name"android:windowSoftInputMode="stateHidden"/>



        <activityandroid:name=".SimpleCalendarViewActivity"android:configChanges="orientat
ion"android:windowSoftInputMode="stateHidden"
android:label="@string/app_name"/>
```

```
                    <activityandroid:name=".Tasks Activity"
android:label="@string/app_name"android:windowSoftInputMode="stateHidden"/>

</application>
</manifest>
```

5.2 Activities

An Activity is an application component that provides a screen with which users can interact in order to do something. The Quick organizer application contains several activities like LoginActivity, RegisterActivity, SimpleCalendarViewActivity and TasksActivity.

5.3 Intent

Intent is an abstract description about some operation that the current activity needs to perform or launch of a new activity. Intent is used for intercommunication between activities within an application or between different applications. Intent contains information required for the new component to perform the requested task as well as information about launch of a new activity to the Android system. The information contains component name which is the full package name of activity to be launched, data to operate on and action to be performed. The actions used in this application are ACTION_EDIT and ACTION_SEARCH_NOTES. The quick organizer application makes use of several intents for making call to calendar add event as well as connecting to the search notes functionality of Ever Notes application cloud service.

Create a new calendar event:

```
Calendar cal = Calendar.getInstance(); Intent intent = new
                                Intent(Intent.ACTION_EDIT);
                                intent.setType("vnd.android.cursor.item/event");
                                intent.putExtra("beginTime", cal.getTimeInMillis());

                                       intent.putExtra("endTime",
cal.getTimeInMillis()+60*60*1000);
startActivityForResult(intent, 10);
```

Manage notes:

```
Intent intent = new Intent();
 intent.setAction(ACTION_SEARCH_NOTES);
 intent.putExtra(SearchManager.QUERY, query);
```

There are two main types of intents namely explicit intents and implicit intents. Explicit intents make use of component name to call activity within the same application or a known service. Implicit intents do not use component name as they involve call to activities outside this application. To provide information about implicit intents which the application wish to handle, intent filters are used and stated in Android-manifest.xml file.

```
<activityandroid:name=".LoginActivity"
android:label="@string/app_name"android:windowSoftInputMode="stateHidden">
<intent-filter>
<actionandroid:name="android.intent.action.MAIN"/>
<categoryandroid:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>
```

In this application, intent filter indicates that LoginActivity is the main activity to be launched upon start of application.

5.4 SQLite Database

Android platform has a built-in database engine SQLite for applications to create and manipulate data using local device memory. Any databases you create will be accessible by name to any class in the application, but not outside the application. A subclass of SQLiteOpenHelper class can be created and new database is created by overriding onCreate() method of this class. A SQLite Database object is used to handle read and write operations with the newly created database tables.

The Quick Organizer application creates a database *OrganizerDB* with two tables' *users* and *tasksTable*. Users table stores information about accounts created by users of this application. The tasksTable maintains tasks created by users and status of the task as well.

Chapter 6 – Implementation

The Quick Organizer application is designed to provide three main features to the user. A user can see calendar view with month and day filter option very similar to Google calendar layout. The calendar view provides options to create new events in the calendar based on calendar chosen by the user, view all events of date picked by user along with a popup window to show event details and also a quick traversal to next and previous date/month view. The notes screen synchronizes with notes created by user in his Ever Notes account and displays all notes of current month. User can also create new notes; choose suitable filter based on title, location of notes created, date of creation and date of modification. A quick search facility is also provided to search notes based on keyword matching pattern. The third screen allows a user to view all his tasks created for keeping track of his daily activities, create new tasks with deadline of completion and marking facility to indicate completed tasks.

The GUI is implemented in xml files and business logic is written using java language. The total lines of code in this application is 2752 which includes java and xml files. There are about 1400 lines of code in calendar screen as it supports the main functionality of displaying a calendar with its events and navigation to other two main features of the application. The custom calendar design requires several validations some of which are check for leap year and displaying dates in correct format for a given month/year. Breakdown of LOC is as given below:-

Language	LOC
Java	2193
Xml	559

6.1 Graphical User Interface

The user interface is kept simple, understandable and consistent with most commonly used calendar applications and in specific Google calendar. This application requires smooth navigation between several tasks and consistent layout with most widely used calendar interfaces. The GUI has been designed to be compatible with both landscape and portrait mode. This application is designed targeting Android tablet users rather than mobile users as this type of applications are most widely used on wider screen than small screen of mobile device. The inter navigation between all three main functionalities of this application makes it easy to use and requires no memorization of navigational instructions. A consolidated view with create, search facility and easy viewing reduces the number of actions by the user and enhances the feel of managing online routine works. The use of popup window for showing event details reduces user effort from navigating to a new screen.

6.1.1 First Screen

Login screen is displayed to user when the application is launched. The Android-manifest file indicates this activity as the starting point of Quick Organizer application.

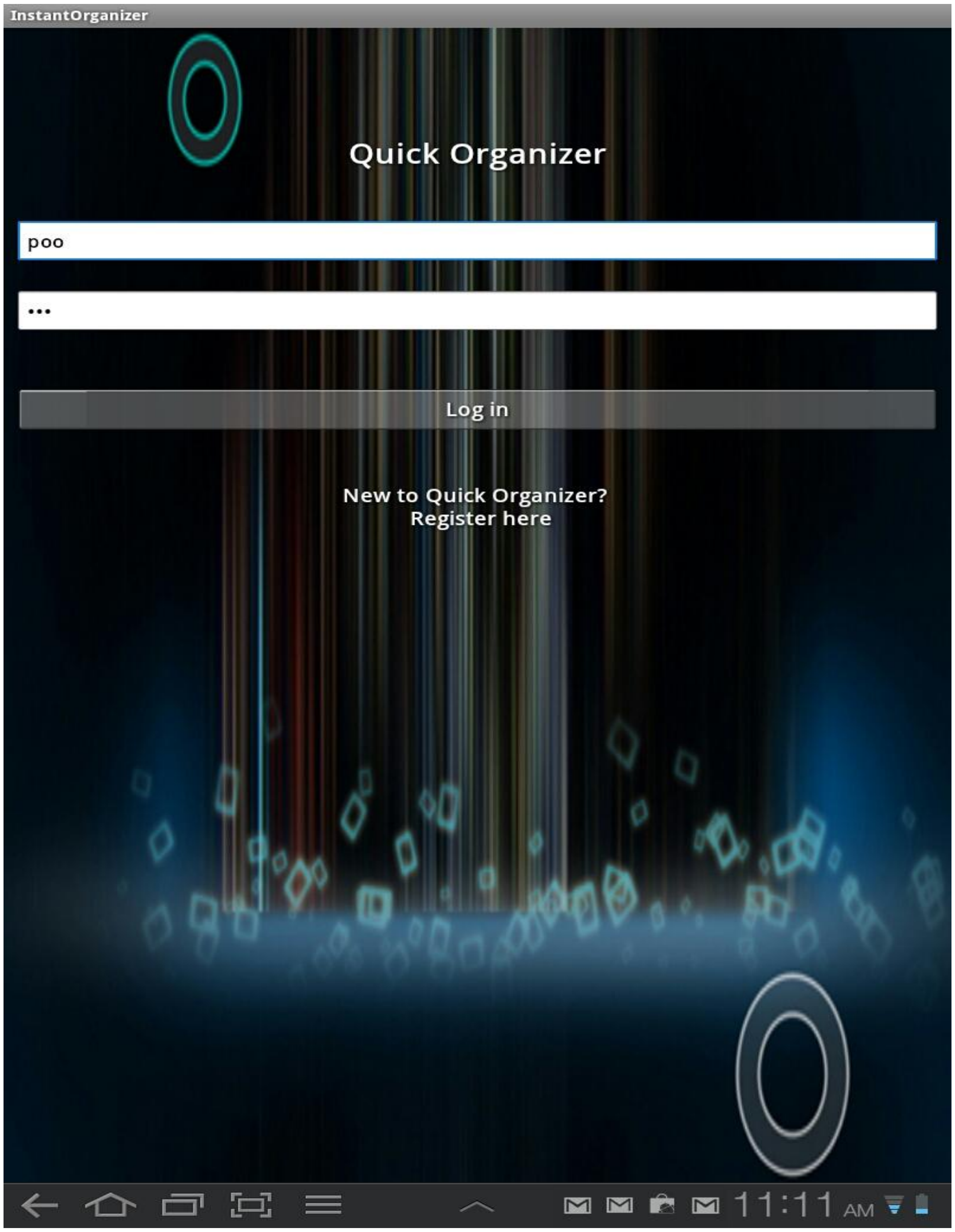


Figure 5: Login Screen

6.1.2 Registration Screen

First time users need to create an account by providing basic information like username (of user’s choice), password and a valid email address. Basic validation rules are applied while accepting user details in order to avoid spam information coming into the application. User is navigated to the calendar screen after successful account creation.

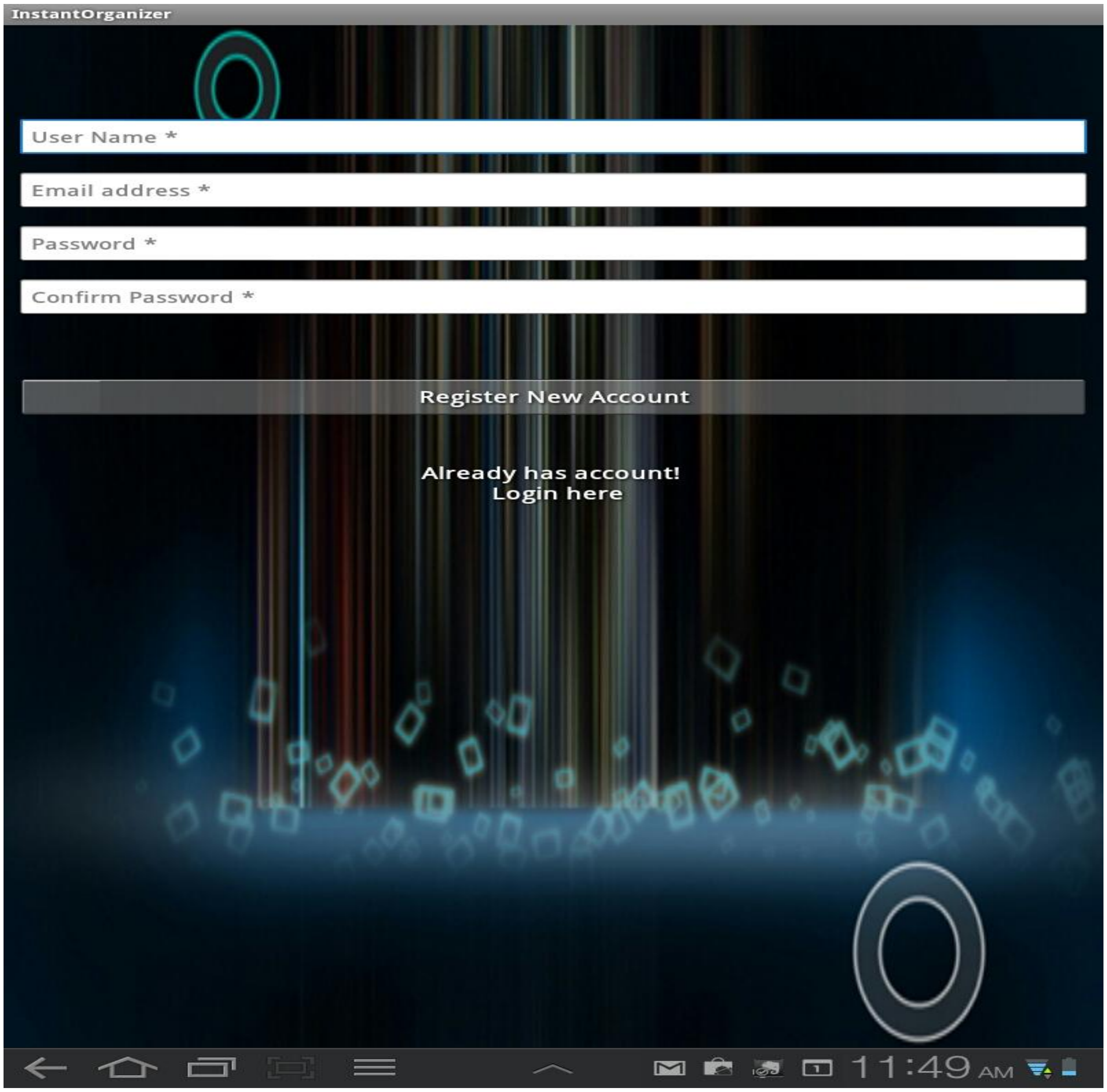
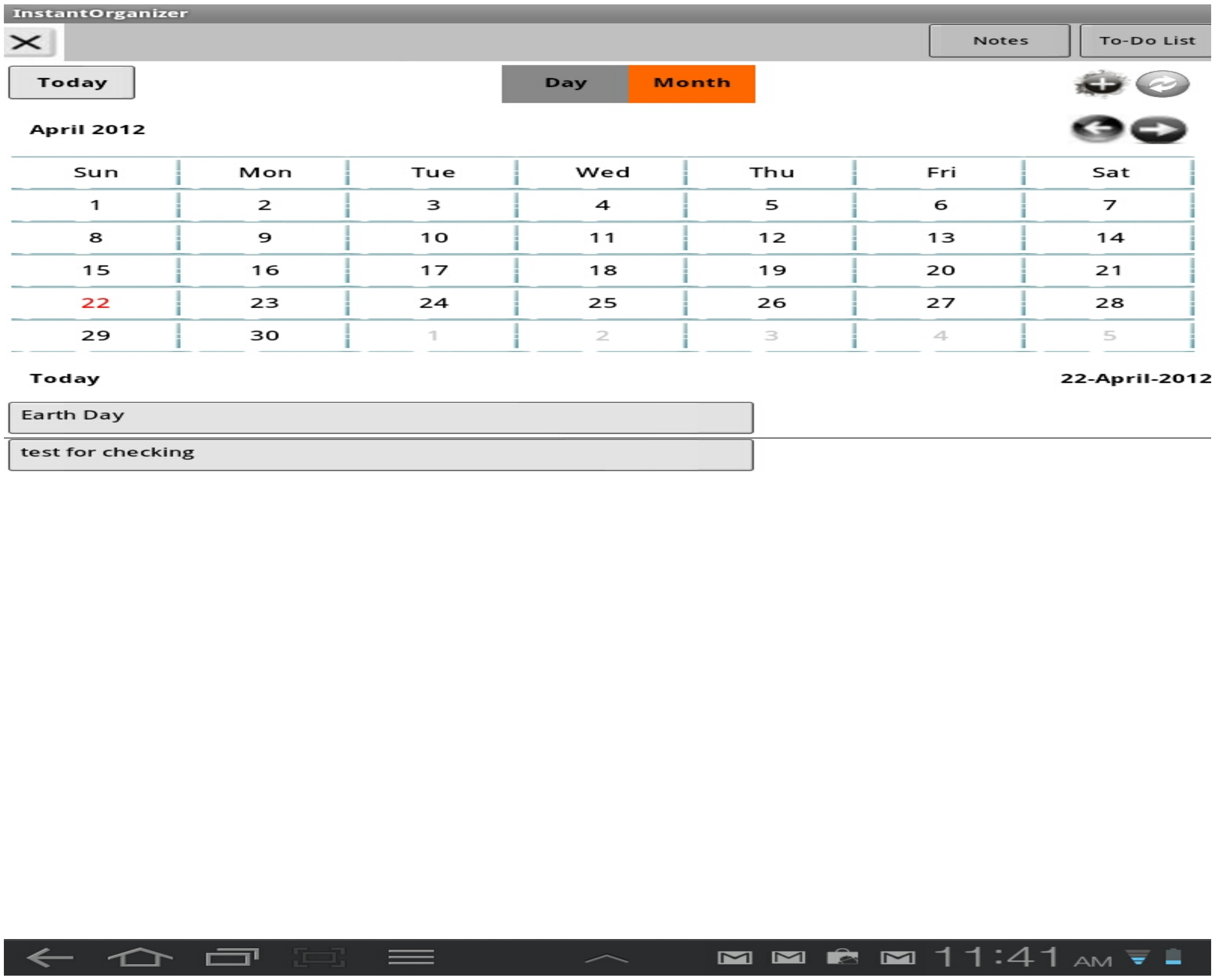


Figure 6: Register Me Screen

6.1.3 Calendar Screen

This screen displays a custom calendar with month and day view and default view set to day displaying current date events. User can navigate between day and month view by simply clicking the corresponding buttons (Android developers, 2012). The default date selected is current date with navigational buttons to go to next/previous month/date. A new event creation is activated by touch on the plus icon which navigates user to the event creation screen provided by calendar Application programming interface (API) (Mortensen). The scrolling feature of events list makes it easy to go through entire list of events for a selected date (WordPress, 2011). A refresh button is provided to synchronize events with external calendars in case of new event creation. There are buttons to navigate to Notes screen and to-do list screen with option to exit from the application by clicking on the cross icon in corner of the screen.



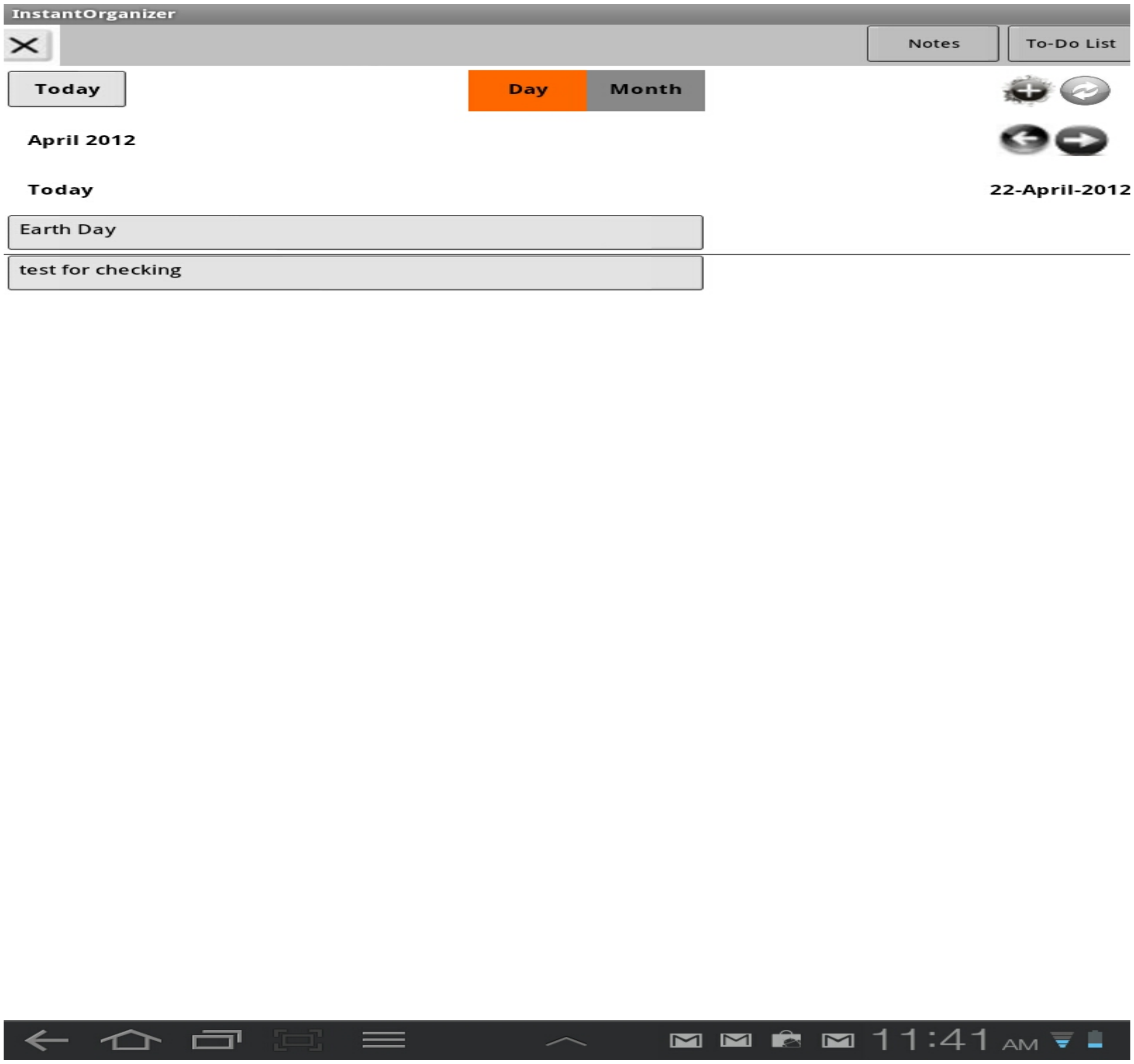
Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Today 22-April-2012

Earth Day

test for checking

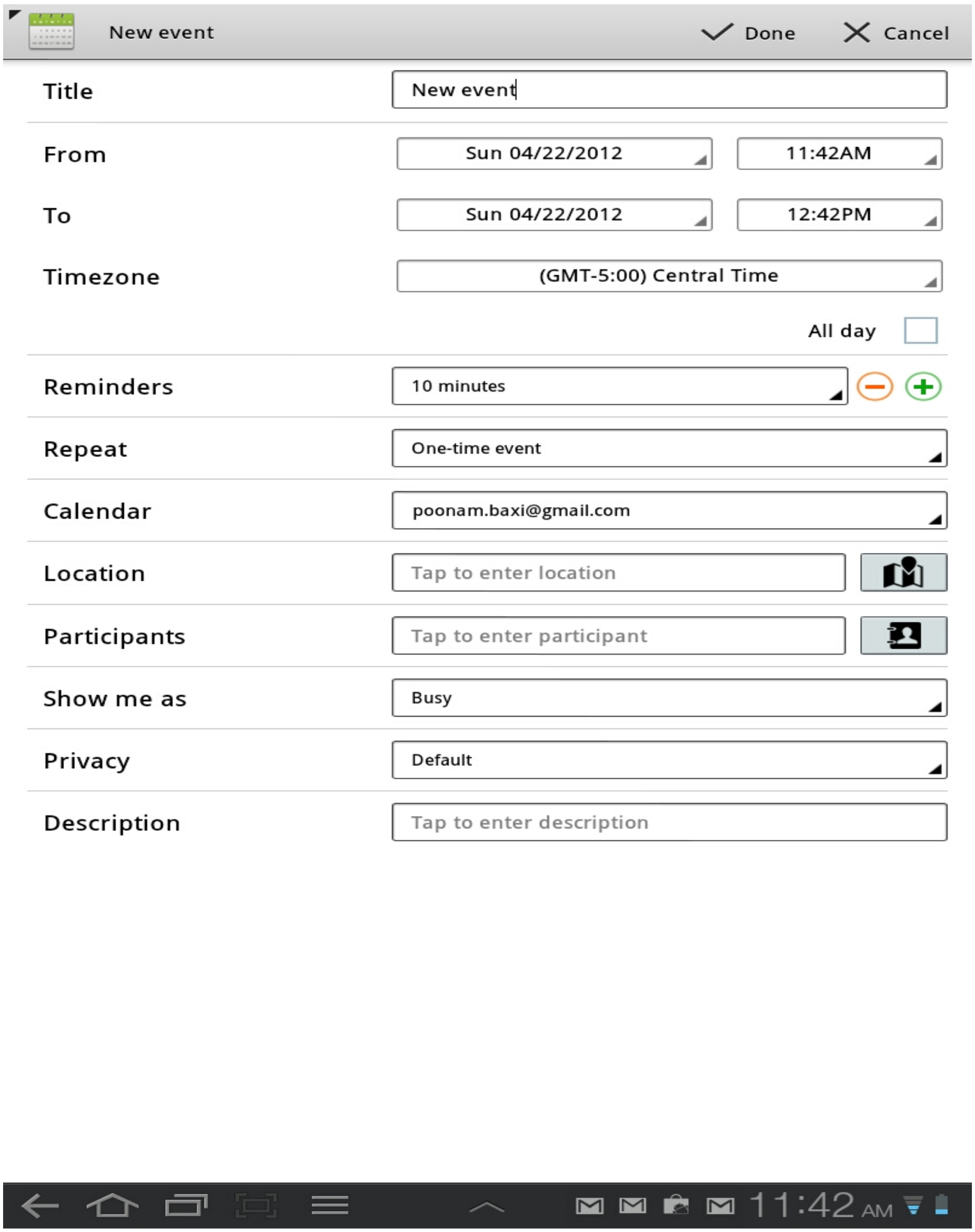
Figure 7: Calendar Screen



Earth Day

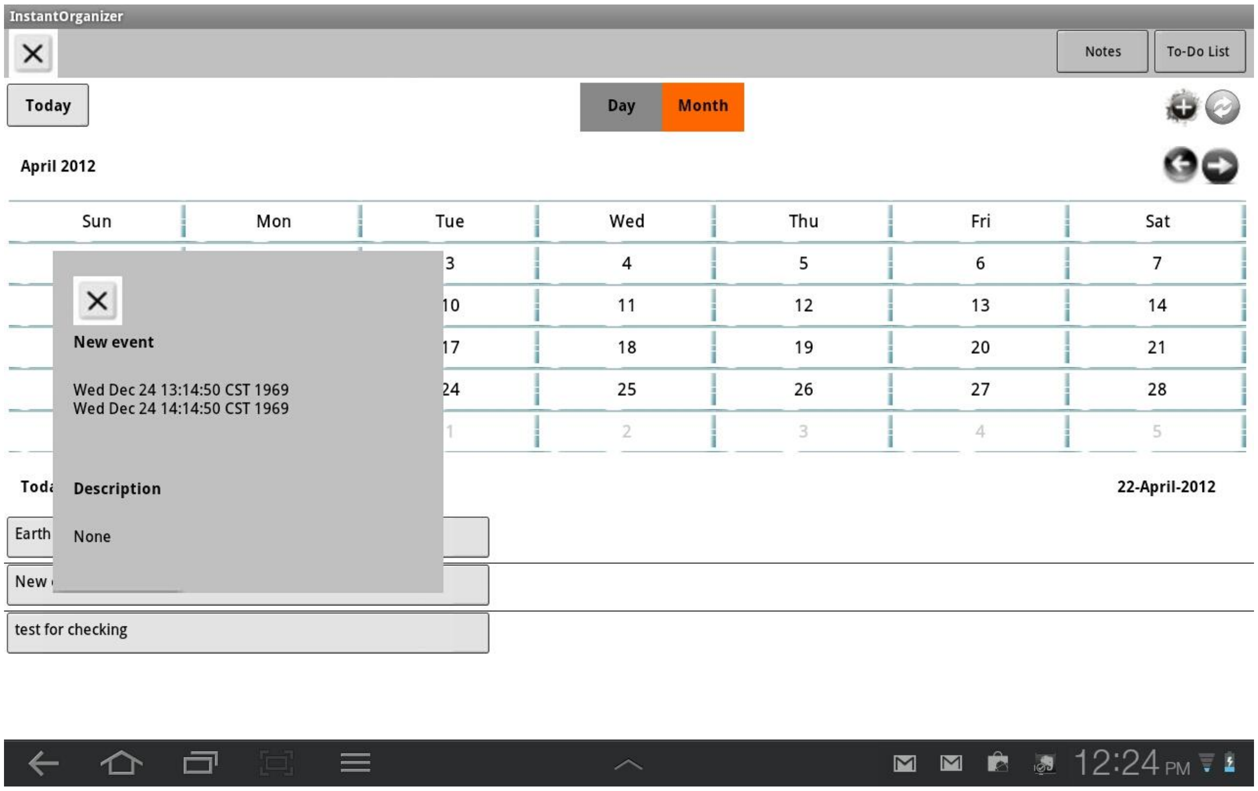
test for checking

Figure 8: Day Mode of Calendar Screen



Title	New event
From	Sun 04/22/2012 11:42AM
To	Sun 04/22/2012 12:42PM
Timezone	(GMT-5:00) Central Time
	All day ☐
Reminders	10 minutes
Repeat	One-time event
Calendar	poonam.baxi@gmail.com
Location	Tap to enter location
Participants	Tap to enter participant
Show me as	Busy
Privacy	Default
Description	Tap to enter description

Figure 9: Create a new event in calendar



New event

Wed Dec 24 13:14:50 CST 1969
Wed Dec 24 14:14:50 CST 1969

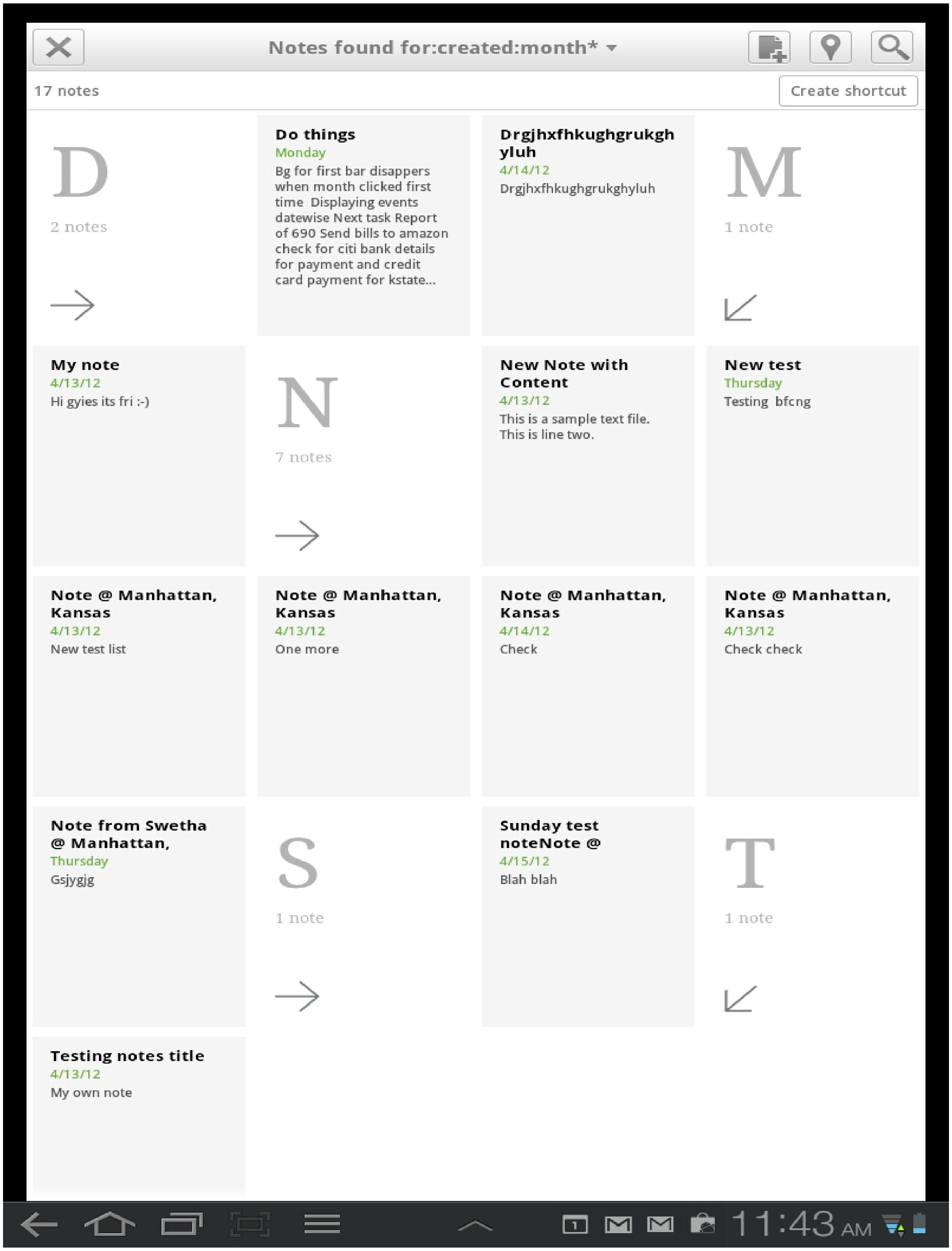
Description

None

Figure 10: Pop-up window of Event Details

6.1.4 Notes Screen

This screen displays sticky notes created by user for the current month. The Ever Notes application for managing notes runs a cloud service and allows applications to synchronize with the user’s account on this application for externally managing the notes created by user (EverNote Developer, 2010). The Notes activity pulls login information of currently logged in user of Ever Notes application on the Android device and fetches all the notes belonging to his account. A user can create new notes with notepad look and feel, search notes based on common search filters like month, day, keyword and other commonly used grammar expressions. This screen also provided a Google map view of places where the notes were created.



Do things
Monday
Bg for first bar disappers when month clicked first time Displaying events datewise Next task Report of 690 Send bills to amazon check for citi bank details for payment and credit card payment for kstate...

Drgjhxfhkughgrukghyluh
4/14/12
Drgjhxfhkughgrukghyluh

My note
4/13/12
Hi gyies its fri :-)

New Note with Content
4/13/12
This is a sample text file. This is line two.

New test
Thursday
Testing bfcng

Note @ Manhattan, Kansas
4/13/12
New test list

Note @ Manhattan, Kansas
4/13/12
One more

Note @ Manhattan, Kansas
4/14/12
Check

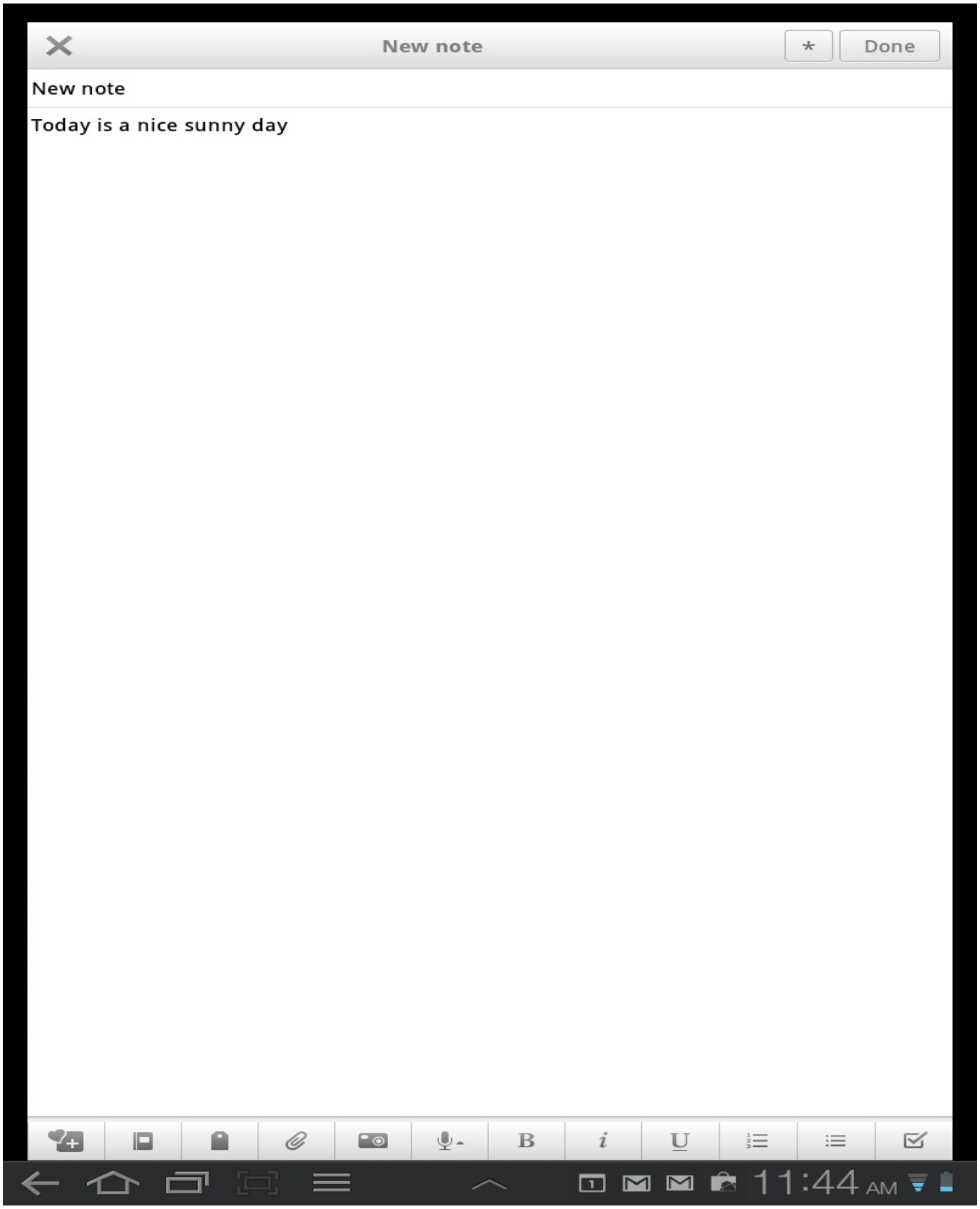
Note @ Manhattan, Kansas
4/13/12
Check check

Note from Swetha @ Manhattan,
Thursday
Gsjygjg

Sunday test noteNote @
4/15/12
Blah blah

Testing notes title
4/13/12
My own note

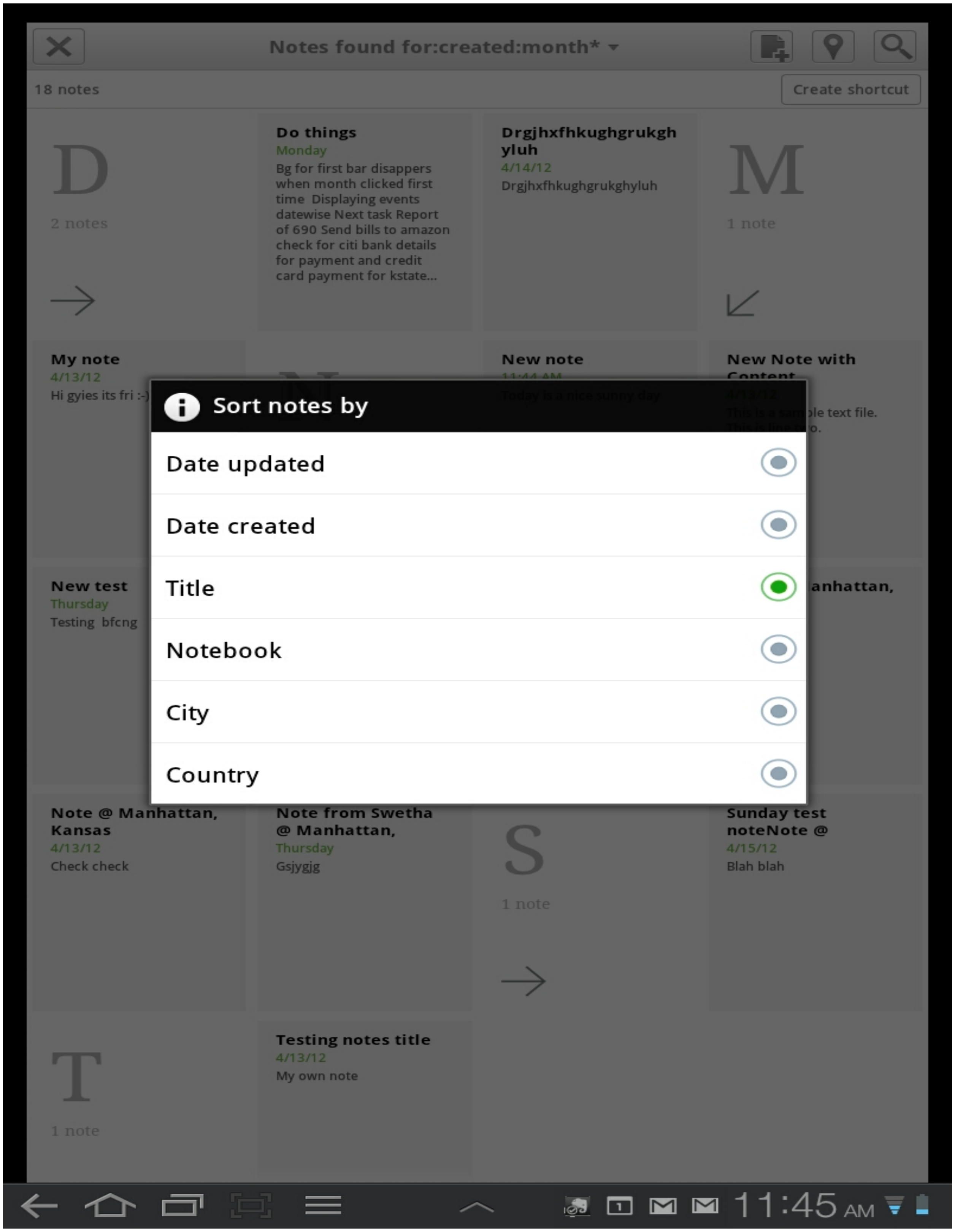
Figure 11: Notes Screen



New note

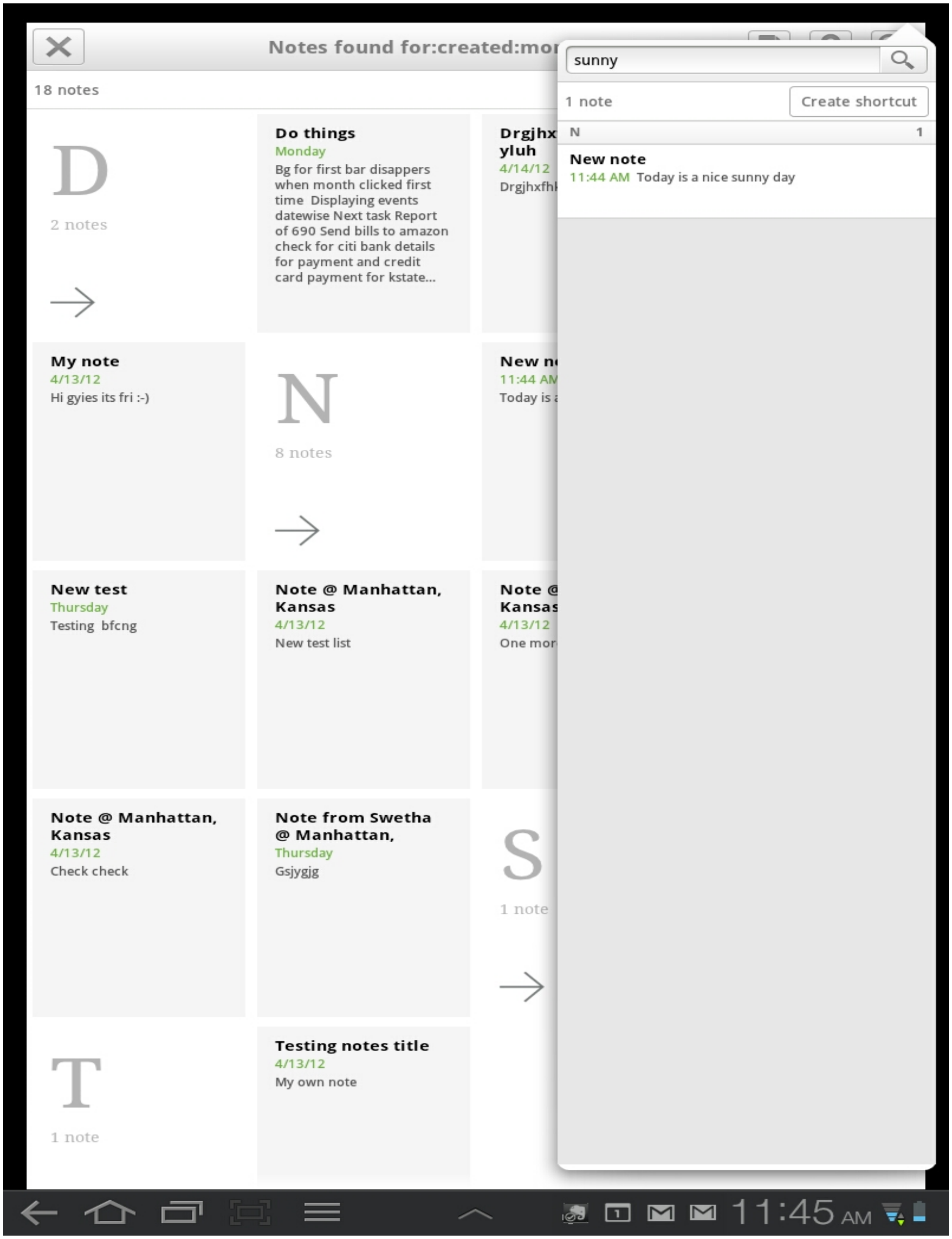
Today is a nice sunny day

Figure 12: Create new note



Date updated
Date created
[✓ Title]
Notebook
City
Country

Figure 13: View notes using filter


Notes found for:created:mor

18 notes

sunny

1 note

Create shortcut

N 1

New note
11:44 AM Today is a nice sunny day

D
2 notes

Do things
Monday
Bg for first bar disappers when month clicked first time Displaying events datewise Next task Report of 690 Send bills to amazon check for citi bank details for payment and credit card payment for kstate...

Drgjhx yluh
4/14/12
Drgjhxfhk

My note
4/13/12
Hi gyies its fri :-)

N
8 notes

New n
11:44 AM
Today is a

New test
Thursday
Testing bfcng

Note @ Manhattan, Kansas
4/13/12
New test list

Note @ Kansas
4/13/12
One mor

Note @ Manhattan, Kansas
4/13/12
Check check

Note from Swetha @ Manhattan,
Thursday
Gsjygjg

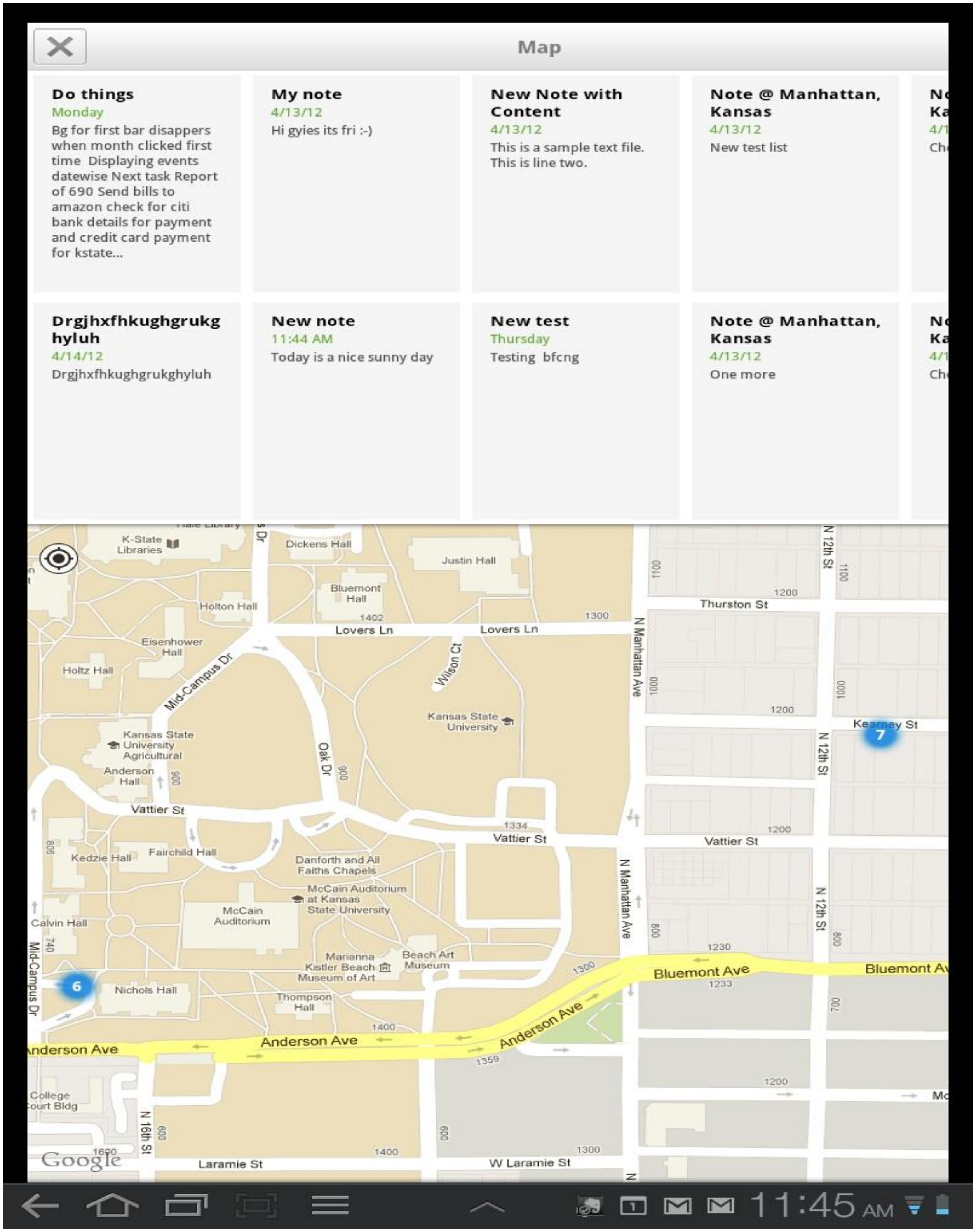
S
1 note

T
1 note

Testing notes title
4/13/12
My own note

11:45 AM

Figure 14: Search a note based on keyword



Map

Do things
Monday
Bg for first bar disappers when month clicked first time Displaying events datewise Next task Report of 690 Send bills to amazon check for citi bank details for payment and credit card payment for kstate...

My note
4/13/12
Hi gyies its fri :-)

New Note with Content
4/13/12
This is a sample text file. This is line two.

Note @ Manhattan, Kansas
4/13/12
New test list

Drgjhxfhkughgrukghyluh
4/14/12
Drgjhxfhkughgrukghyluh

New note
11:44 AM
Today is a nice sunny day

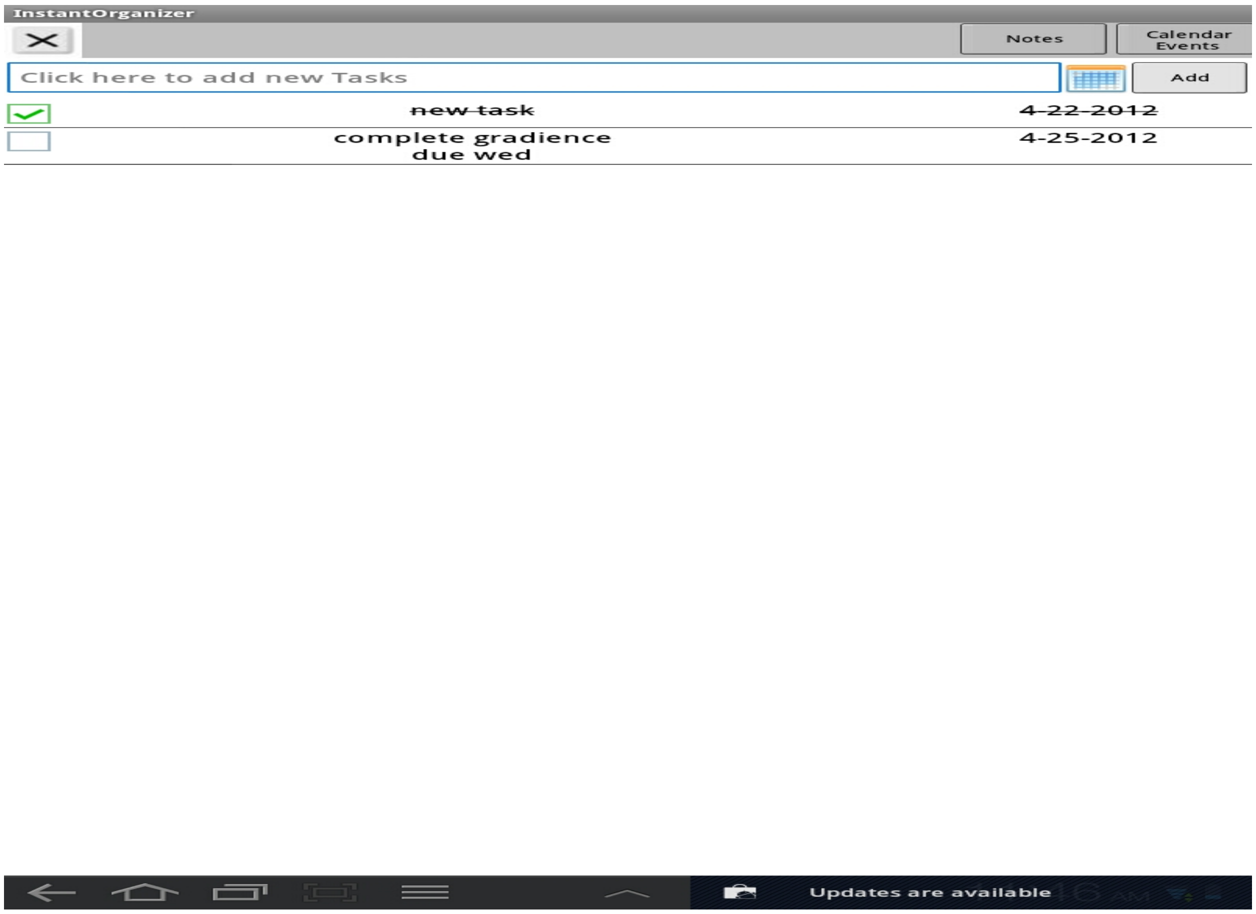
New test
Thursday
Testing bfcng

Note @ Manhattan, Kansas
4/13/12
One more

Figure 15: View notes location wise

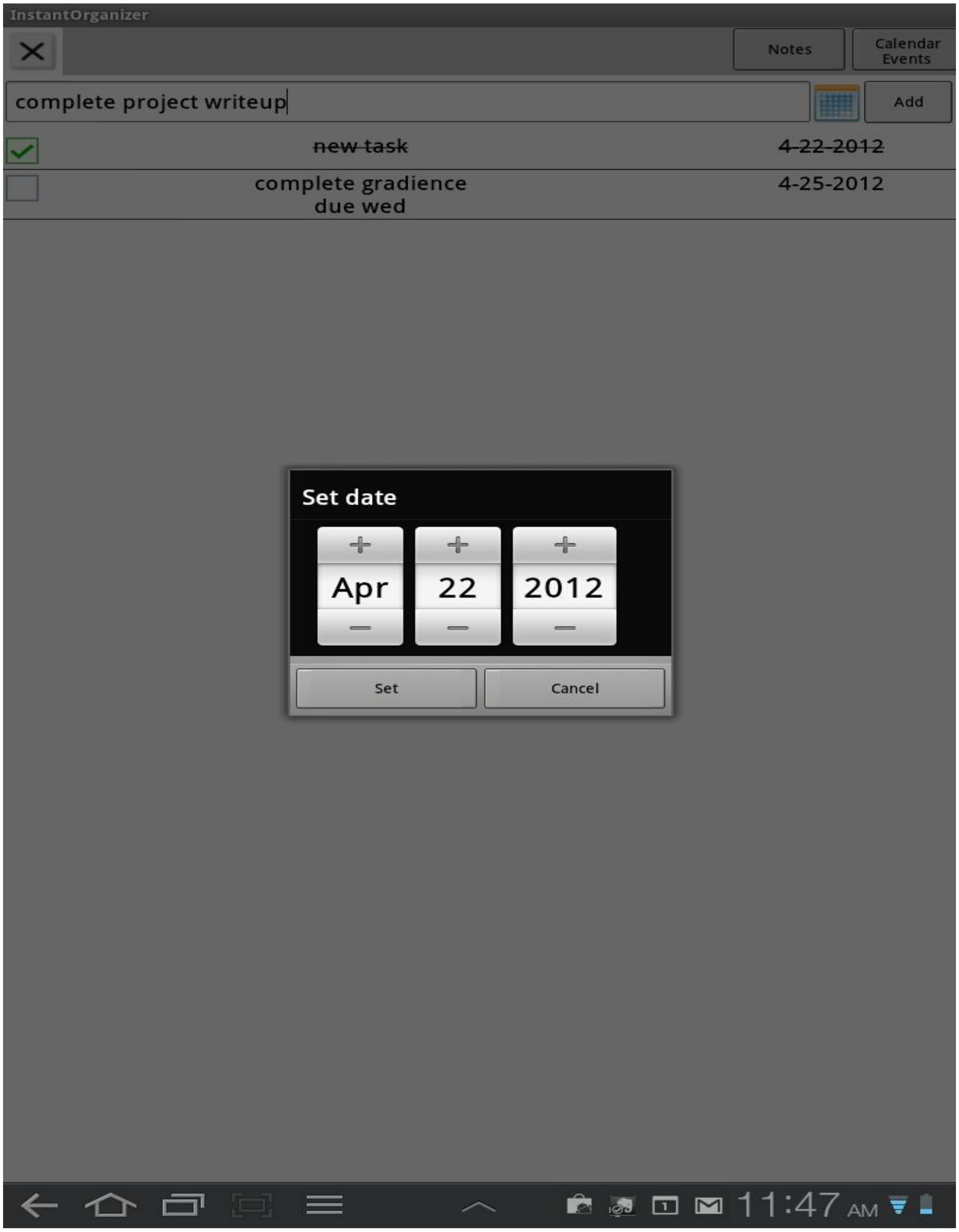
6.1.5 Tasks Manager Screen

The task manager screen displays a to-do list created by the logged in user. There is an option to create a new to-do task along with the deadline for completion of task (Android developers, 2012). The to-do list provides a consistent view with most-popular scratch-pad view for creating to-do lists used in everyday life. Tasks are stored in local device SQLite database (Vogel, 2011).



✓	~~new task~~	~~4-22-2012~~
	complete gradience due wed	4-25-2012

Figure 16: Task management screen

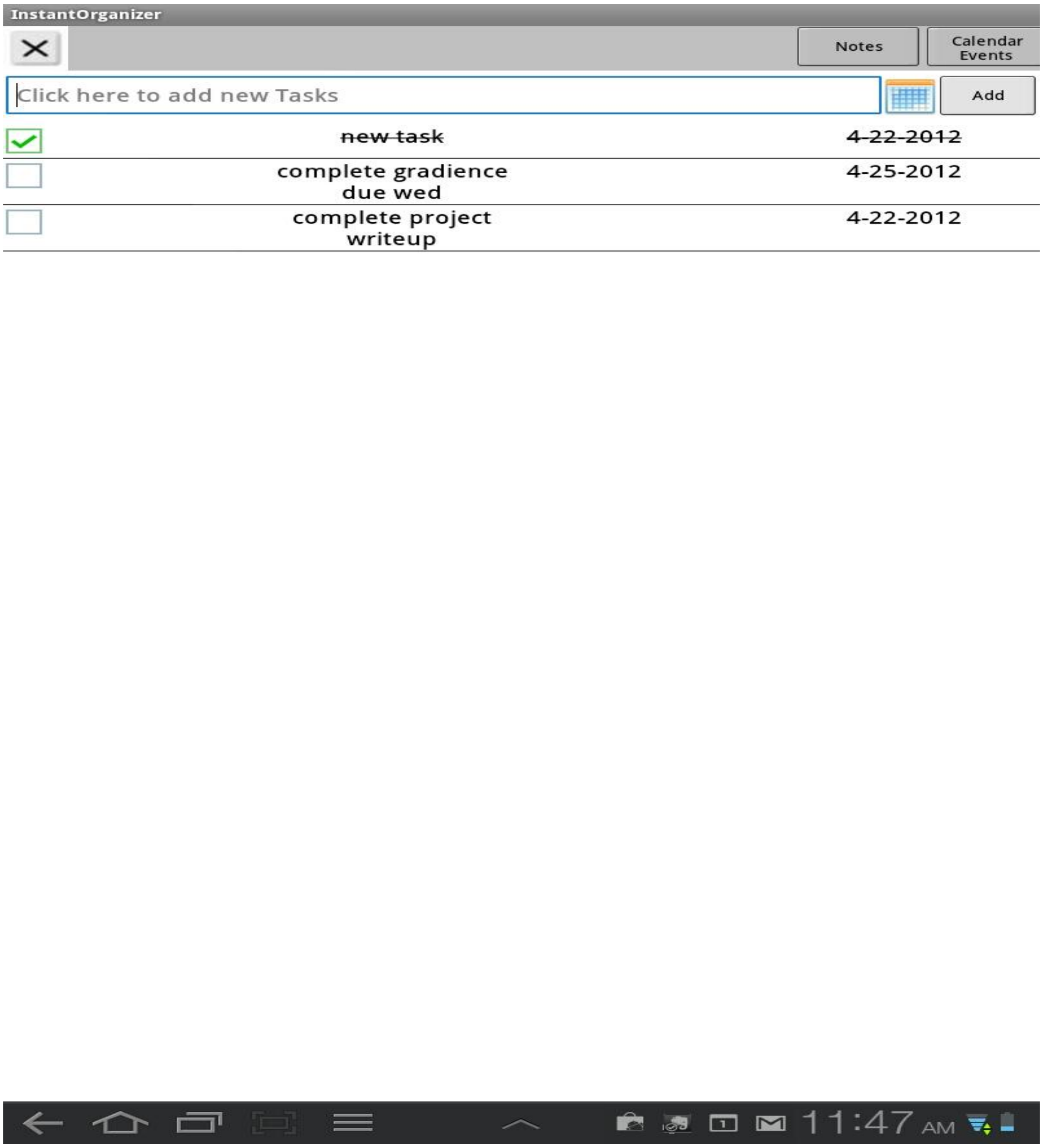


complete project writeup

✓	~~new task~~	~~4-22-2012~~
	complete gradience due wed	4-25-2012

Apr 22 2012

Figure 17: Create a new task



✓	~~new task~~	~~4-22-2012~~
☐	complete gradience due wed	4-25-2012
☐	complete project writeup	4-22-2012

Figure 18: View tasks list with task status

Chapter 7 - Testing and Logging

7.1 Logger and Debugger

Android provides a default logger tool to add debugging messages at various levels like debug, error message in source code. The Dalvik Debug monitor server (DDMS) gives a complete view of all logging messages for the currently connected device. Android Device Bridge (ADB) shell provides access to the SQLite database and table entries for data on the applications of connected device.

7.2 Performance

The application is tested manually for performance based on transit time from one activity to another. On an average, it takes approximately 0.4ms to load the screen for notes management as this screen makes a remote web service call to synchronize with Ever Notes account of the user. All other screens load a bit faster and take around 0.02ms response time.

7.3 Unit Testing

Unit testing involves testing of each individual piece of code and functionality testing of every independent screen without interaction with outside world. The Quick Organizer application is tested manually with the help of below test cases for ensuring correct functionality.

Sr. No	Test Case Description	Expected Result	Actual Result
1	On load of calendar screen	Set default mode as day view and display events of current date	Pass
2	On click of Today’s button	Display events of current date	Pass

	on calendar screen		
3	On click of month button of calendar screen	Display a custom view of calendar with selected date highlighted and list of events populated in scrolling list	Pass
4	On click of forward arrow in month mode of calendar screen	Display calendar set with next month calendar dates and events associated with the selected date	Pass
5	On click of backward arrow in month mode of calendar screen	Display calendar set with previous month calendar dates and events associated with the selected date	Pass
6	On click of forward arrow in day mode of calendar screen	Display the next date and events associated with the next date	Pass
7	On click of forward arrow in day mode of calendar screen	Display the previous date and events associated with the previous date	Pass

8	On click of date in calendar on calendar screen	Show the date selected highlighted and display list of scrolling events if any of the selected date	Pass
9	On click of event from events list on calendar screen	Show a popup window with event name, start and end timestamp and location of event with a close icon.	Pass
10	On click of any note in note management screen	Opens a window with note in the notepad along with edit note option	Pass
11	On click of search icon on notes management screen	Shows a search box with notes that satisfy the search query typed in search box	Pass
12	On click of add note of notes management screen	Launch a create new note screen with notepad view	Pass
13	On touch of	Allows user to	Pass

	"click here to add new task" of tasks management screen	make note of new task	
14	On click of add tasks button of tasks management screen	If task description is not empty , then create a new tasks with chosen calendar date as deadline and update the tasks list displayed below on screen	Yes

Table 1: Unit Test Cases

7.4 Integration Testing

After each piece of code is tested individually, the application is tested as a whole for proper navigation, inter-compatibility with different modules and out- of-application calls. The below test cases verify the correctness of interface testing of this application.

Sr. No	Test Case Description	Expected Result	Actual Result
1	On click of login button of LoginActivity screen	If valid username, password are entered, then	Pass

		navigate to calendar screen	
2	On click of register here link of LoginActivity screen	Navigate to register activity	Pass
3	On click of Notes button on calendar screen	Navigate to notes management screen	Pass
4	On click of to-list of calendar screen	Navigate to tasks management screen	Pass
5	On click of Done button on add event screen	Navigate back to calendar screen	Pass
6	On click of close icon of calendar screen	Display alert box for exit from application	Pass
7	On click of close icon of notes screen	Navigate back to the parent screen	Pass
8	On click of close icon of tasks management screen	Navigate to calendar screen	Pass
9	On click of notes button of tasks management screen	Navigate to notes management screen	Pass
10	On click of	Navigate to	Pass

	calendar events button of tasks management screen	calendar screen	
11	Change of orientation from landscape to portrait mode	Avoids reloading of activity, stores the previous state and changes layout to match the changed orientation	Pass
12	Change of orientation from portrait to landscape mode	Avoids reloading of activity, stores the previous state and changes layout to match the changed orientation	Pass

Table 2: Integration test cases

7.5 Compatibility Testing

Different Android devices have different screen and display resolution based on number of pixels and size of screen. I could not test this application on emulator as the emulator does not support calendar API and hence need to be tested on the real device itself. This application is more specifically designed for Android tablets which have wider screen than the Android phones. The small screen of Android mobiles is not feasible for the overall GUI of such applications; hence they are more popular on tablets. The application is tested to support both portrait and landscape mode and maintain state between orientation changes. The application is designed in such a way that orientation change does not reload the activity but displays the new layout with previous saved state.

Portrait mode

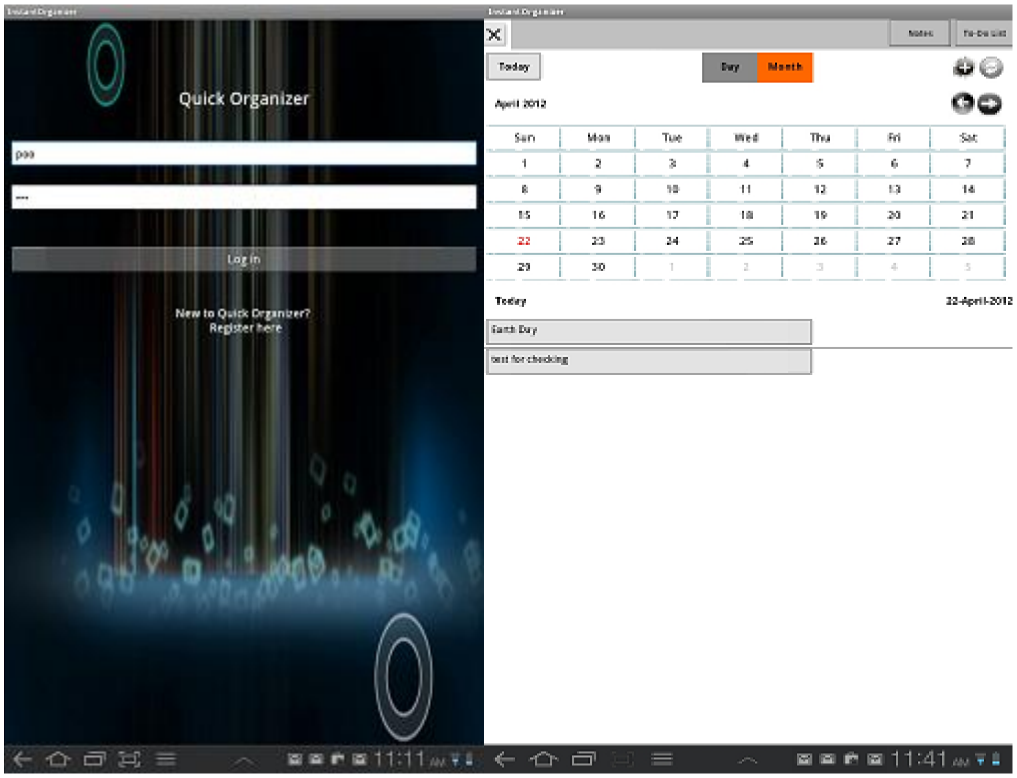
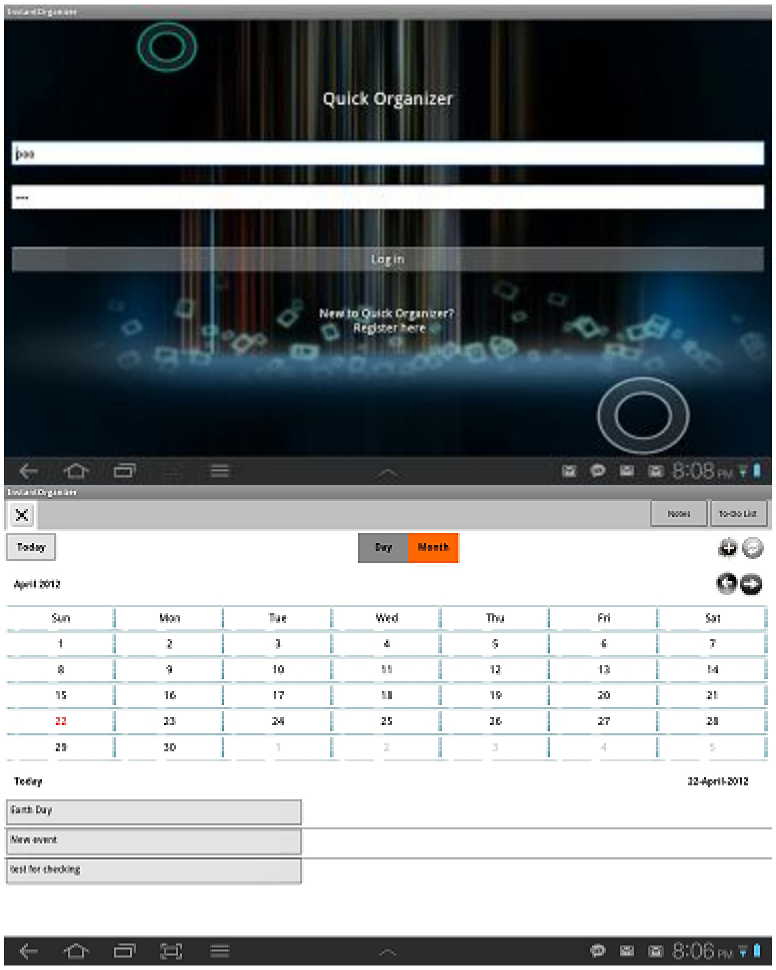


Figure 19: Compatibility Testing - Portrait mode

The above figure shows the GUI of this application in portrait orientation of Android device. Portrait orientation is where the height of the screen is greater than the width. The application handles switching from landscape to portrait mode by overriding onConfigurationChanged method. This method saves the previous state of the screen which is very important as user should be able to see the same content in spite of orientation change. By default, Android reloads an activity behind the current focused screen after orientation change. Quick Organizer application informs the system that any configuration change will be handled by the application itself and prevents reloading of activity thus reducing considerable amount of I/O.

Landscape mode



Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Today 22-April-2012

Earth Day

New event

test for checking

Figure 20: Compatibility testing - Landscape mode

Landscape orientation is where the width of the page is greater than the height. Quick Organizer application handles orientation change from portrait to landscape orientation by loading the new layout with previous state saved and reloaded. This avoids reloading of event data from calendars and reestablishing connection with remote Ever Notes web service.

Chapter 8 – Conclusion

The application is successfully implemented and tested on the real devices. This project increased my knowledge of mobile application development and understanding of various challenges in this field. Also, it gave me good exposure to learn development in various software development life cycle phases. The overall experience of developing GUI for a small screen with clean and elegant interface in contrast with my previous desktop application development was very nice. This application development improved my mobile development skills and analytical abilities.

Chapter 9 – Future Work

The Quick Organizer application can be extended further in below ways:

1. The application can be extended to provide a shared access among family members to the same account. This will allow family members to inform each other about things to buy or things to be do even remotely.
2. An additional feature of sending the calendar events data or tasks created via email or text message will be of help to the user.

Chapter 10 – References

(2010, December 28). Retrieved April 2012, from EverNote Developer: http://dev.evernote.com/documentation/local/chapters/Android.php

(2012). Retrieved February 2012, from Android developers: http://developer.android.com/index.html

Media Wiki. (2011, June 13). *Android Architecture*. Retrieved from http://elinux.org/Android_Architecture

Mortensen, P. (n.d.). Retrieved March 2012, from Stackoverflow: http://stackoverflow.com/questions/3721963/how-to-add-calendar-events-in-android

Murphy, & Mark. (2011). *Beginning Android 3.* New York: Apress.

Owens, Allen, G., & Mike. (2010). *The Definitive Guide to SQLite, Second Edition.* New York: Apress.

Vogel, L. (2011, July 9). *Android SQLite*. Retrieved from Android SQLite: http://www.vogella.de/articles/AndroidSQLite/article.html

WordPress. (2011, April 14). *Android Tutorials*. Retrieved from Android Tutorials and Android Development Course for Beginners: http://www.androidtutorials.org/