Temporal Lock for Android Applications

GRADUATE PROJECT

Submitted to the Faculty of
The School of Engineering & Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

By
Prem Chander Racharla
Summer 2015

Committee Members

Dr. Ajay Katangur
Committee Chairperson

Dr. Dulal Kar
Committee Member
ABSTRACT

Nowadays, smart devices are becoming ubiquitous. These devices have so many applications that are inbuilt, and user installed. The user is unaware of the time he or she spends on the particular application. Dependence on the applications constructs the user restriction capability weaker that results in the human uncontrollability towards these apps. The time restriction application helps the user to pick the apps from the list of applications present in the device and allows them to set a time limit on the usage of these apps over a twenty four hour period. After the time limit expires, the access over the application is denied, and the user needs approval to access the application again. As the user is an administrator, he or she can make it further accessible, but it acts as a reminder for him or her about the time of app usage that in turn makes continuous addiction reduced. This application also provides the time usage details of each app on the device. This application helps the user to manage time properly and contributes to avoid falling addicted to any particular app.
TABLE OF CONTENTS

ABSTRACT................................................................................................................................. ii

TABLE OF CONTENTS................................................................................................................. iii

LIST OF FIGURES ........................................................................................................................ v

1. BACKGROUND AND RATIONALE ....................................................................................... 1
   1.1 Android ........................................................................................................................... 2
   1.2 Android Development ..................................................................................................... 3
   1.3 Existing Applications ....................................................................................................... 4
   1.4 Solution to the Problem ................................................................................................. 5

2. NARRATIVE ............................................................................................................................. 6
   2.1 Problem Statement ......................................................................................................... 6
   2.2 Motivation ....................................................................................................................... 6
   2.3 Project Objective ............................................................................................................ 6
   2.4 Functionalities of the Project ......................................................................................... 7

3. SYSTEM DESIGN .................................................................................................................... 8
   3.1 System Architecture ......................................................................................................... 8
   3.2 Design Flow ..................................................................................................................... 9
   3.3 Use Case Diagram .......................................................................................................... 10
   3.4 Class Diagram ............................................................................................................... 11
   3.5 User Interface ................................................................................................................ 12
3.5.1 Navigation Drawer ........................................................................................................ 13

4. IMPLEMENTATION ........................................................................................................... 17

4.1 Retrieving the list of applications present on the device .............................................. 17
4.2 Starting a Service to periodically check for usage ...................................................... 18
4.3 Setting Soft and Hard Restrictions .............................................................................. 19
4.4 Getting Foreground Application Information ............................................................. 20
4.5 Setting up the database ............................................................................................... 21
4.6 Calculating the Time Spent on Foreground Application by the User ......................... 22
4.7 Deleting the History ..................................................................................................... 23

5. TESTING AND EVALUATION ......................................................................................... 25

5.1 Setting Hard Restriction ............................................................................................. 25
5.2 Setting Soft Restriction .............................................................................................. 26
5.3 Start and Stop Service .................................................................................................. 28
5.4 Delete History of any Particular Application ............................................................. 30
5.5 Delete Entire History .................................................................................................. 31

6. CONCLUSION AND FUTURE WORK .......................................................................... 33

6.1 FUTURE WORK ............................................................................................................. 33

7. REFERENCES ..................................................................................................................... 34
LIST OF FIGURES

Fig. 1.1 Market share of different mobile operating systems by Bloomberg L.P. .......................... 2
Fig. 3.1 System architecture .................................................................................................................. 9
Fig. 3.2. Design flow ................................................................................................................................ 10
Fig. 3.3 Use case diagram ........................................................................................................................ 11
Fig. 3.4 Class diagram ............................................................................................................................... 12
Fig. 3.5 Password screen of the application ............................................................................................... 13
Fig. 3.6 Navigation drawer of the application ........................................................................................... 14
Fig. 3.7 All Apps Fragment of the application .......................................................................................... 15
Fig. 3.8 Usage Fragment of the application .............................................................................................. 16
Fig. 4.1 Retrieving the Package info list .................................................................................................. 18
Fig. 4.2 AppInfo class ............................................................................................................................... 18
Fig. 4.3 Setting up a service that runs periodically .................................................................................. 19
Fig. 4.4 code snippet for setting the soft and hard restrictions ................................................................. 20
Fig. 4.5 Getting the foreground application ............................................................................................. 20
Fig. 4.6 Code snippet for creating tables ................................................................................................. 22
Fig. 4.7 Code snippet for tracking foreground application ..................................................................... 23
Fig. 4.8 Code snippet for deleting history ............................................................................................... 24
Fig. 5.1 Setting up hard restriction .......................................................................................................... 25
Fig. 5.2 Hard restriction ............................................................................................................................. 26
Fig. 5.3 Setting soft restriction for Calculator app ................................................................................... 27
Fig. 5.4 Soft restriction notification displayed for Calculator app ............................................................ 27
Fig. 5.5 Navigation Drawer ....................................................................................................................... 28
Fig. 5.6 Monitoring Service started ........................................................................................................... 29
Fig. 5.7 Monitoring Service stopped ........................................................................................................ 29
Fig. 5.8 Usage of Calculator app .............................................................................................................. 30
Fig. 5.9 Usage after deletion of history for Calculator app ................................................................. 31
Fig. 5.10 Usage history and entire usage history deletion ................................................................. 32
Fig. 5.11 Usage after deletion ............................................................................................................... 32
1. BACKGROUND AND RATIONALE

“The smartphone revolution is under-hyped, more people have access to phones than access to running water. We have never had anything like this before since the beginning of the planet” - Marc Andreessen, founder of Netscape.

These days’ smart devices, especially mobile phones have become part and parcel of one’s life [1]. The main reason of this is the usability and affordability of smart devices. Smart devices come with features like camera, GPS, Bluetooth, Wi-Fi, sensors and many more. The advantages of smart devices are, it allows to browse, play games, chat, know the weather, and many more due to different applications that are available. The hypothesis of optimal flow [2] says that Information technology can make one fall into dependence. One can say a person is addicted if they feel they are highly dependent on it [3], they do repeated tasks on it [4]. Information Technology creates several kinds of addiction some of the most prominent ones are work addiction, media addiction, and smart device addiction. Work addiction [5] is something that makes people neglect the important commitments like health and family. Media is something that helps individuals to know different kinds of information like news, read different opinions on a topic and watch videos. Example for media addiction is excessive use of the Internet that may lead to loneliness [6]. Smart device addiction is something that may lead a person to an uncertain attitude. The most prominent example of this kind of mentality is the usage of smart devices in restricted places (workplace) and dangerous situations such as while driving. Mobile phone addiction can be noticed when a person feels irritated if they cannot find their smart device. It may lead to so many health problems like depression, anxiety, and time management problems [7], [8].

The hardware used in every smartphone is almost similar, but the software used may vary. The most popular kinds of software (operating system) available in smart devices are Android
provided by Google, iOS provided by Apple, Windows provided by Microsoft. The market share for each operating system from the year 2008 to 2013 is shown in the fig. 1.1. From fig. 1, it is clear that Android users have highly increased when compared with other operating systems. The reason behind this is, it is an open-source operating system, and there are wide ranges of applications developed for this operating system where most of them are available free of cost.

![SMARTPHONE OPERATING SYSTEM MARKET SHARE](image)

**Fig. 1.1 Market share of different mobile operating systems by Bloomberg L.P.**

**1.1 Android**

Android is an open source operating system developed on the Linux kernel. Initially, it was being developed by a company Android, but later in the year 2005, it was bought by Google. Android operating systems was officially launched in the year 2008 as version 1.0. The various versions of Android released by Google up to the date are as follows:

- i) Android 1.2 (API level 2)
- ii) Android 1.5 (API level 3) known as Cupcake
- iii) Android 1.6 (API level 4) known as Donut
- iv) Android 2.0, 2.0.1, 2.1 (API level 5,6,7) known as Éclair
v) Android 2.2 to 2.2.3 (API level 8) known as Froyo
vi) Android 2.3 to 2.3.2, 2.3.3 to 2.3.7 (API level 9,10) known as Gingerbread
vii) Android 3.0 to 3.2 (API level 11,12,13) known as Honeycomb
viii) Android 4.0 to 4.0.2, 4.0.3 to 4.0.4 (API level 14,15) known as an ice cream sandwich
ix) Android 4.1,4.2,4.3 (API level 16,17,18) known as Jelly bean
x) Android 4.4 (API level 19,20) known as KitKat
xi) Android 5.0 to 5.0.2 (API level 21) known as Lollipop

Since it is an open source operating system, Google has provided various tools for developing applications for Android. Every tool can be downloaded for free of cost from the website [9]. Google also has provided an Integrated Development Environment for developing the applications known as Android Development studio. There are a lot of open source developers developing apps for Android platform as it is user-friendly. Moreover, the apps developed can be easily submitted to play store [10].

1.2 Android Development

Android applications are mostly developed in Java language, and most people use Eclipse [11] as IDE because it is easy to use. For developing apps in eclipse, one has to add ADT (Android Development Tool) plugin to the IDE and install Android SDK. The charm of the Android development environment is that it also provides a virtual device known as the emulator for testing the application developed. Based on the user’s target device, they can execute their app in the respective emulator. One can also use the Android device for testing the app by installing the necessary drivers for the device [9]. Android provides the following four building blocks for developers:
i. Activity (Helps in creating windows and takes care of the user instructions)

ii. Service (Long time running process in background)

iii. Content Provider (Helps in sharing the data present on the device among the applications)

iv. Broadcast Receiver (Helps in broadcasting the messages or events among the applications)

1.3 Existing Applications

Currently, there are only a few applications available for the smart device users that help the users to get reminded about the time usage of the device. Some of the popular applications are:

1.3.1 Moment

Moment tracks the time spent by the user on the smart device, and it also maintains usage history of the apps for the device.

Advantages

i. Provides an option of setting a time limit on the usage of the device.

ii. The user will be notified when the usage exceeds the time limit.

iii. It also provides an option of automatically turning off the device when the usage exceeds the limit.

Disadvantages

i. Available only for iOS users.

ii. Tracks only the time usage of the device.
1.3.2 Checky

Checky gives the count of how many times the user has checked the device on the current day and the previous day if available. It is available for both Android and iOS users.

Advantages
i. Provides the user to set an option of reminding the usage count at the end of the day.

Disadvantages
i. Limited features.

ii. Doesn’t track the time.

1.4 Solution to the Problem

The developed app helps the user from falling addiction by letting the user set the time limit on any particular app. It also tracks the usage time of each application present on the device, and maintains daily, weekly and monthly usage statistics of the applications.
2. NARRATIVE

2.1 Problem Statement

Up to date, Android market has provided only a few applications that minimize addiction towards the smart device. But there is no proper application in the market regarding the user concern and limit the access to a particular application that makes the user addictive. These addictions may lead to various problems.

2.2 Motivation

The technological advancement made the change of the human interaction and communication with others. These smart devices have been a major part in the technological advancement which may result in major human behavioral damage. For instance, behavioral addiction leads to many behavioral disorders.

Therefore, there should be an application that takes care of the time management skills and other relative aspects of the user. There should be some kind of warning to the user whenever he/she operates the smart device and its applications excessively.

2.3 Project Objective

The main objective of this project is to enforce efficient usage of the smart device so that it can also help in reducing the behavioral addiction in users. The primary part of this is that it should allow the user to know about the usage of all applications and set the time limit on any particular app.
2.4 Functionalities of the Project

The functionalities of the project are:

- Display all the apps present in the smart device.
- Allow the user to select the apps for restriction and tracking the usage.
- Allow the user to set time limit on selected application.
- Allow the user to set either soft or hard restrictions on any particular app.
- After the completion of time limit either the user should be reminded or restricted access to any particular app.
- Allow the user to set restriction completely on some apps based on user request.
- Ask the user to set a password at the time of the initial launch and subsequent accesses are password protected.
3. SYSTEM DESIGN

3.1 System Architecture

The system architecture is shown in fig. 3.1. It has three layers:

1. Presentation Layer

2. Logical Layer

3. Data Layer

   Presentation Layer – It consists of two parts, namely themes and layouts. Themes have styles used for layout design and fonts, strings and menus defined in the XML files are used in the layouts. Layouts are the views that are seen by the user while running the application.

   Logical Layer – It consists of two parts, namely foreground and background. The foreground subdivision has the files, which are going to be executed whenever the application is active. The background subdivision has the files, which are executed in the background as a service even when the application is not active.

   Data Layer – It has a database and the files which help in accessing the database. The database has tables which store the application data and usage data.
3.2 Design Flow

Fig. 3.2 shows the design flow of the system. Initially, when the app is opened for the first time, the user is requested to set the password and security question. From next time onwards, the user needs to enter the password to get access to the application. After getting access for the application, user can select “AllApps” or “Usage”. In “AllApps”, application provides two tabs: “AllAppsTab” and “SelectedAppsTab”. The “AllAppsTab” shows all applications present on the device, from which the user can select applications for restriction. The “SelectedAppsTab”
contains selected applications that are chosen from “AllAppsTab”. If the user selects “Usage”, the app provides three tabs: “Daily”, “Weekly” and “Monthly” which provides time usage details of all applications present on the device.

![Design Flow Diagram]

**Fig. 3.2. Design flow**

### 3.3 Use Case Diagram

Fig. 3.3 shows the use case diagram for the Temporal Lock application. In the use case diagram, all the ovals represent the functionalities of the user. The Select Apps functionality provides the user to select the applications for setting the time based restriction, edit the restriction options (time, and soft or hard restriction) and remove the application from the selected apps list. The Usage functionality provides application usage statistics of all apps present on the device, delete usage, and edit restriction options. The Start Monitor functionality starts the service that keeps track of the foreground application. The Stop Monitor functionality stops the monitoring
service. The Delete Entire History functionality will delete entire usage history of every application present on the device.

Fig. 3.3 Use case diagram

3.4 Class Diagram

The class diagram consists of the interfaces, methods, variables and relationships between them. Fig. 3.4 shows the class diagram for the application. It shows the main classes of the application. MainActivity class create all the views and triggers the events according to the user choice. DatabaseHeler class does all the CRUD (Create, Read, Update, and Delete) operations of the application. MyApp and SelectApp classes help in displaying all apps and selected apps on the device. AppInfo class is a model which helps in sending the devices apps’s information such as app’s name, app’s package, and app’s version among the different classes. DailyFragment,
**WeeklyFragment**, and **MonthlyFragment** classes display the daily, weekly and monthly usage statistics of the applications. **AlarmReceiver** class is a service that helps in tracking the foreground app’s information.

![Diagram](image)

**Fig. 3.4 Class diagram**

### 3.5 User Interface

The main user interfaces of the system are **PasswordActivity**, **NavigationDrawer**, **AllAppsFragment**, and **UsageFragment**. Fig. 3.5 shows the **PasswordActivity** screen with buttons numbered from zero to nine. These numbers are randomly arranged to avoid smudging attacks.
3.5.1 Navigation Drawer

Fig. 3.6 shows the Navigation Drawer that has the app menu. It will be loaded whenever the user swipes right starting from the left edge of the device and also by clicking the app icon on the action bar of the application. The user can choose any of the options from the menu that triggers the respective event of the application.
3.5.2 AllAppsFragment

Fig. 3.7 shows All apps fragment screen, which consists of two tabs: SELECTED APPS and ALL APPS. In these tabs, the user is allowed to select applications for setting, editing and removing the restriction on the apps present on the device.
Fig. 3.7 All Apps Fragment of the application

3.5.3 UsageFragment

Fig. 3.8 shows the Usage fragment, which consists of three tabs: DAILY, WEEKLY, and MONTHLY. In these tabs, the user can see each application usage, delete usage history, and set restriction.
Fig. 3.8 Usage Fragment of the application
4. IMPLEMENTATION

The Temporal Lock application is mainly developed in seven separate modules and later all of the modules are integrated. The seven modules are as follows:

i. Retrieving the list of applications present on the device.

ii. Starting a service to periodically check for usage.

iii. Setting soft and hard restrictions.

iv. Getting foreground application information.

v. Setting up the database.

vi. Calculating the time spent on foreground application by the user.

vii. Deleting the history.

4.1 Retrieving the list of applications present on the device

Android provides a framework known as Package Manager, which helps in retrieving all packages (apps) information present on the device. From the list of package info, only the user interacting packages should be stored and displayed. This can be achieved by checking whether the respective package has the launcher intent. Fig. 4.1 shows the code for retrieving the list. AppInfo class is used as a model for referring to the package info details such as app’s name, app’s package name, app’s icon which is shown in fig. 4.2.
List<PackageInfo> apps;

apps = getActivity().getPackageManager().getInstalledPackages(0);

appInfo = new ArrayList<AppInfo>();

PackageInfo p = apps.get(i);

ApplicationInfo ai = p.applicationInfo;

Intent in = ac.getPackageManager().getLaunchIntentForPackage(p.packageName);

if (in == null)
    continue;

else

    // the details are stored using app info class

        Fig. 4.1 Retrieving the Package info list

public class AppInfo{

    private String appname;
    private String pname;
    private String versionName;
    private String restrict;
    private Drawable icon;
    private long totalTime;
    private long remainTime;

    // has getter and setter methods

}

Fig. 4.2 AppInfo class

4.2 Starting a Service to periodically check for usage

To generate a service that runs at a regular interval, Android provides a framework known as Alarm Manager. Alarm Manager broadcasts the event on the device so there should be a receiver to catch the event. AlarmReciever is the class written to receive the event. Since the
*AlarmReceiver* acts as a receiver, it should be mentioned in the *Android Manifest* file of the application. Fig. 4.3 shows the receiver part of the *Android Manifest* file and the code snippet of setting up the event using *the Alarm Manager*.

```java
//receiver part of Android Manifest file
<receiver android:name=".AlarmReceiver" />
//code snippet of setting up the event using alarm manager
//getting the alarm manager
alarmMgr = (AlarmManager) context.getSystemService(Context.ALARM_SERVICE);
//setting the event to trigger
Intent intent = new Intent(context, AlarmReceiver.class);
pendingIntent = PendingIntent.getBroadcast(context, 0,
                                     intent, PendingIntent.FLAG_UPDATE_CURRENT);
//Repeating the event for every five seconds
alarmMgr.setRepeating(AlarmManager.ELAPSED_REALTIME,
                      SystemClock.elapsedRealtime(), (5000), pendingIntent);
```

**Fig. 4.3 Setting up a service that runs periodically**

**4.3 Setting Soft and Hard Restrictions**

Soft restriction will notify the user by displaying up a notification in the notification bar of the device. The hard restriction will block the foreground application with *timeUp* screen, fig. 4.4 shows the code snippet of setting the soft and hard restrictions.
//notifies the user
showNotification(context, helper.getAppDetails(app).getAppname() + " TimeUp!!");

// blocking the current foreground app
Intent in = new Intent(context, TimeUp.class);
in.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

Fig. 4.4 code snippet for setting the soft and hard restrictions

4.4 Getting Foreground Application Information

The Activity Manager Framework helps in retrieving the information about the foreground application and the services. Fig. 4.5 shows the code snippet for getting the foreground application information.

```java
String[] getActivePackages() {
    final Set<String> activePackages = new HashSet<String>();
    final List<ActivityManager.RunningAppProcessInfo> processInfos = am.getRunningAppProcesses();
    for (ActivityManager.RunningAppProcessInfo processInfo : processInfos) {
        if (processInfo.importance == ActivityManager.RunningAppProcessInfo.IMPORTANCE_FOREGROUND) {
            activePackages.addAll(Arrays.asList(processInfo.pkgList));
        }
    }
    return activePackages.toArray(new String[activePackages.size()]);
}
```

Fig. 4.5 Getting the foreground application
4.5 Setting up the database

Android provides SQLite database for storing and retrieving the data. Android has hidden the setup or administration from the developer [11]. The developer has to define only the SQL statements and need not bother about the setup.

In temporal lock application, four tables are created for storing all application information and their usage. `SelectedAppDetail` table is used for storing the data related to selected applications only, `AllAPPS` table stores all the application names and their respective packages, `Tracking` table stores the time for the foreground application, and the `History` table stores the usage history of all applications present on the device. Fig. 4.6 shows the code snippet for creating all four tables.
db.execSQL("CREATE TABLE " + selectedAppDetailsTable + " (" + appName + " TEXT," + packageName + " TEXT," + appVersionCode + " TEXT," + appVersionName + " TEXT," + totalTime + " INT8 NOT NULL," + remainTime + " INT8 NOT NULL," + today_date + " TEXT," + restrict + " TEXT, PRIMARY KEY("+appName+"))");

db.execSQL("CREATE TABLE " + TrackingTable + " (" + trPackageName + " TEXT," + trTimeInMillis + " TEXT," + isTrackingStarted + " TEXT)"");

db.execSQL("CREATE TABLE " + allAppsTable + " (" + usageTime + " INT8 NOT NULL," + appName + " TEXT," + packageName + " TEXT," + appVersionName + " TEXT," + restrict + " TEXT, PRIMARY KEY("+appName+"))");

db.execSQL("CREATE TABLE " + historyTable + " (" + historyAppPkgname + " TEXT," + historyApptime + " INT8 NOT NULL," + historyAppdate + " TEXT)"");

**Fig. 4.6 Code snippet for creating tables**

### 4.6 Calculating the Time Spent on Foreground Application by the User

Temporal lock application tracks the time of the foreground application and stores in the database. Since it has to be done in regular intervals, it will be placed along with the services discussed in section 4.2. Fig. 4.7 shows the corresponding code segment for tracking the time of the foreground application.
if (helper.getRemainingTime() == 0) {
    helper.startTracking(app, System.currentTimeMillis() / 1000);
} else {
    if (helper.isTrackerChanged(app)) {
        long newRemainTime = System.currentTimeMillis() / 1000 -
            (helper.getRemainingTime());
        String previousPkgName = helper.getTrackingName();
        helper.updateAppDurationTime(previousPkgName, newRemainTime);
        helper.startTracking(app, (System.currentTimeMillis() / 1000));
    } else {
        long systemTime = System.currentTimeMillis() / 1000;
        long newRemainTime = systemTime -
            (helper.getRemainingTime());
        String previousPkgName = helper.getTrackingName();
        helper.updateAppDurationTime(previousPkgName, newRemainTime);
        helper.updateTracker(app, systemTime);
    }
}

Fig 4.7 Code snippet for tracking foreground application

4.7 Deleting the History

Temporal application supports four kinds of deletion on application usage:

i. Deleting the present day usage.

ii. Deleting the weekly usage.

iii. Deleting the monthly usage.

iv. Deleting the entire usage.

Fig. 4.8 illustrates the code snippet for deleting the history.
//Daily

db.execSQL("delete from "+historyTable+" where "+historyAppPkurname+" = "+pack_name+" AND "+ historyAppdate +" = "+todDate+");

//Similarly the statement will be repeated 7 times for week and 30 times for month deletions respective

//entire history

db.execSQL("DELETE FROM " + historyTable);

Fig. 4.8 Code snippet for deleting history
This chapter deals with the functional evaluation of the application. The application is tested by installing on a Nexus tablet with Android version 5.0.2. The application supports minimum Android version 4.0.4 to recent update 5.0.2. Each module of the application is tested with all possible test cases.

5.1 Setting Hard Restriction

The hard restriction restricts the app completely after completion of the assigned time to the particular application. Fig. 5.1(a) shows the setting of hard restriction on the chrome app for one minute and fig. 5.1(b) shows the apps which are set for restriction. Fig. 5.2(a) shows the launch of application chrome and fig. 5.2(b) shows the time up restriction for the chrome application after one minute of usage.

![Setting up hard restriction](image1)

![Chrome app](image2)

**Fig. 5.1 Setting up hard restriction**
5.2 Setting Soft Restriction

The soft restriction gives a notification after completion of the assigned time for any particular application. Fig. 5.3(a) shows the setting of soft restriction on the calculator application for one minute and fig. 5.3(b) shows the apps which are set for restriction. Fig. 5.4(a) shows the launch of application calculator and fig. 5.4(b) shows the notification of time up for the calculator application.
Fig. 5.3 Setting soft restriction for Calculator app

Fig. 5.4 Soft restriction notification displayed for Calculator app
5.3 Start and Stop Service

The navigation drawer of the application shown in fig. 5.5 provides options for starting and stopping the service. Whenever start monitoring option is selected, the service will start and displays a toast message “Monitoring is Started” as shown in fig. 5.6 and similarly when stop monitoring is selected, the service will stop and displays a toast message “Monitoring is Stopped” as shown in fig. 5.7.

Fig. 5.5 Navigation Drawer
Fig. 5.6 Monitoring Service started

Fig. 5.7 Monitoring Service stopped
5.4 Delete History of any Particular Application

The application tracks the usage information of the apps after starting the service, and provides details about the applications usage for daily, weekly and monthly. Fig. 5.8(a) shows the usage time of the calculator app on a daily basis. From the figure it can be seen the calculator app has been used for three minutes. Fig. 5.8(b) shows the user interface when user clicks on the calculator app. Fig. 5.9 shows calculator app’s usage time after deletion of the history by clicking on “Delete History” button from fig. 5.8(b).

Fig. 5.8 Usage of Calculator app
Fig. 5.9 Usage after deletion of history for Calculator app

5.5 Delete Entire History

The user can also select an option of deleting the entire usage history any app. Fig. 5.10(a) shows the usage time of all the applications before deletion of entire usage history. Fig 5.10(b) displays a dialog when “Delete Entire History” is clicked as shown in fig. 5.5(b). The displayed dialog provides options to “Delete” or “Cancel”. Fig. 5.11 shows the usage history after deletion of the entire history.
Fig. 5.10 Usage history and entire usage history deletion

Fig. 5.11 Usage after deletion
6. CONCLUSION AND FUTURE WORK

Temporal Lock helps the user in tracking the phone usage, setting up the time limit on each application present on the device, improves from falling from addiction. It also helps the parents to restrict their child’s usage. The limitations of the application are the user can stop the background service by going into the settings of the device, and the user can also delete the data of the application.

6.1 FUTURE WORK

The app can be improved in the future by adding the following functionalities:

- Ask password when uninstalling the app
- Restricting the user to stop the service and delete the data
- Extending it for iOS users
7. REFERENCES


http://developer.android.com/guide/topics/data/data-storage.html#db