University of Padua
2016/2017 MSc Degree Computer Science
Mobile Programming

AppSpan

1 Introduction

The Project is based on the Proposal #5: PhoneUsage.
The application is named “AppSpan”, it is an Android App for monitoring applications time usage.

Today Smartphones are used for accomplish numerous tasks. We rely on apps for communication and productivity, but also for recreational purpouses, in fact there is an high risk to spend too much time on mobile games and social networks.

Keeping track of the time spent on our smartphone could aid us in making decisions, such as to inhibit the time spent on our favourite apps to increase productivity or to free up some space by deleting the least used ones.

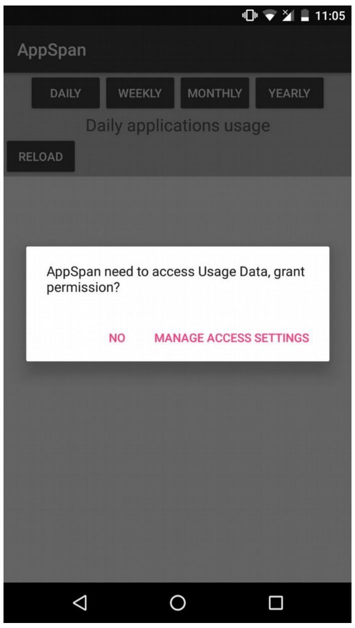
AppSpan monitors the time usage of all the applications installed on the user device and lets the user set time limits for the applications. The app also warns the user when a limit is reached.

2 Features

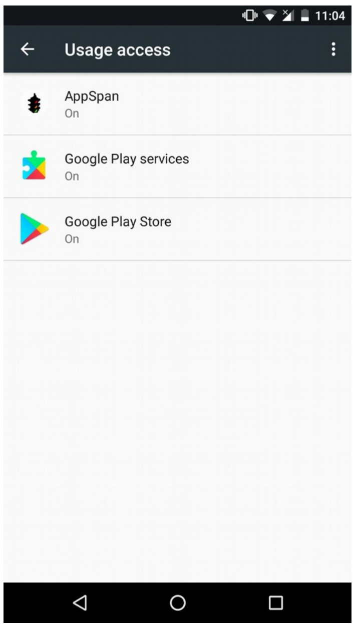
2.1 First Installation

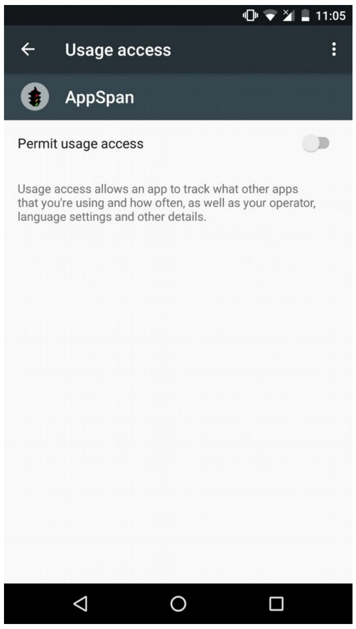
After the installation it will be necessary to grant the Usage Access Permission to the application in order to make the app be able to monitor the other applications and read usage data. It is a system level permission so the user must personally permit the access.

A message will be shown, asking for Usage Access Permission, if the user agrees, the application will open the Usage Access Settings   on the device Settings application.



AppSpan need to access Usage Data, grant permission?





Usage access allows an app to track what other apps that you're using and how often, as well as your operator, language settings and other details.

2.2 Apllications Usage Data

In the main layout it is possible to examine some usefull usage data, for every app it will be shown: icon, name, last usage time and date, total time usage out of the total time permitted if a limit has been set.

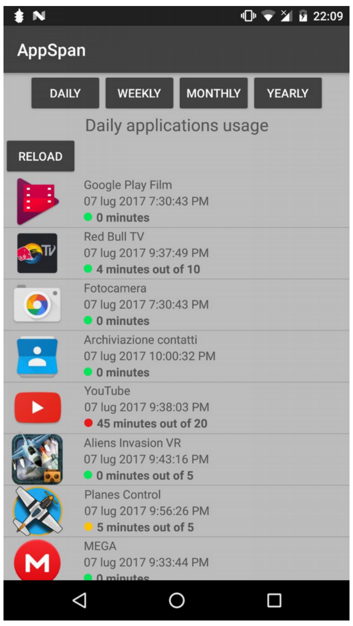
The usage status is higlighted with a colored dot:

- green, usage under the limit, or no time limit is set
- yellow, the limit is reached
- red, the limit is exceeded

The user can choose the preferred time period: Daily, Weekly, Montly, Yearly.
The maximum duration that the system keeps this data is as follows:

- Daily data: 7 days
- Weekly data: 4 weeks
- Monthly data: 6 months
- Yearly data: 2 years.

Depending on the selected period of time it will be listed every day usage for the last 7 days(a single app will have 7 different entries), weekly usage for the last 4 weeks, montly usage for the last 6 month, yearly usage.



Daily applications usage

Google Play Film
07 lug 2017 7:30:43 PM
● 0 minutes

Red Bull TV
07 lug 2017 9:37:49 PM
● 4 minutes out of 10

Fotocamera
07 lug 2017 7:30:43 PM
● 0 minutes

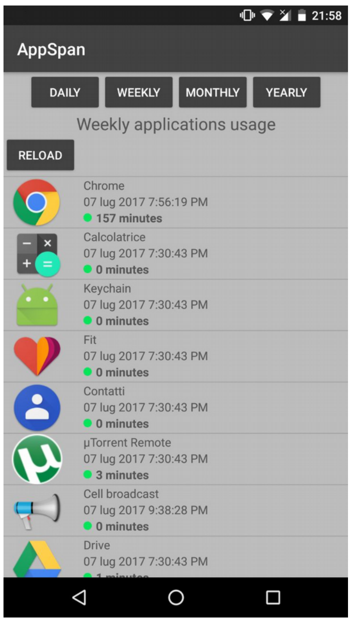
Archiviazione contatti
07 lug 2017 10:00:32 PM
● 0 minutes

YouTube
07 lug 2017 9:38:03 PM
● 45 minutes out of 20

Aliens Invasion VR
07 lug 2017 9:43:16 PM
● 0 minutes out of 5

Planes Control
07 lug 2017 9:56:26 PM
● 5 minutes out of 5

MEGA
07 lug 2017 9:33:44 PM
● 0 minutes



Weekly applications usage

Chrome
07 lug 2017 7:56:19 PM
● 157 minutes

Calcolatrice
07 lug 2017 7:30:43 PM
● 0 minutes

Keychain
07 lug 2017 7:30:43 PM
● 0 minutes

Fit
07 lug 2017 7:30:43 PM
● 0 minutes

Contatti
07 lug 2017 7:30:43 PM
● 0 minutes

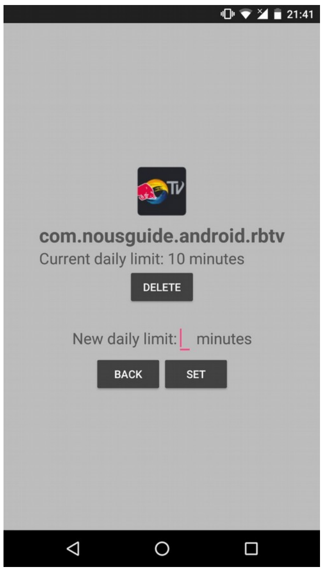
µTorrent Remote
07 lug 2017 7:30:43 PM
● 3 minutes

Cell broadcast
07 lug 2017 9:38:28 PM
● 0 minutes

Drive
07 lug 2017 7:30:43 PM
● 1 minutes

2.3 Limit Setting

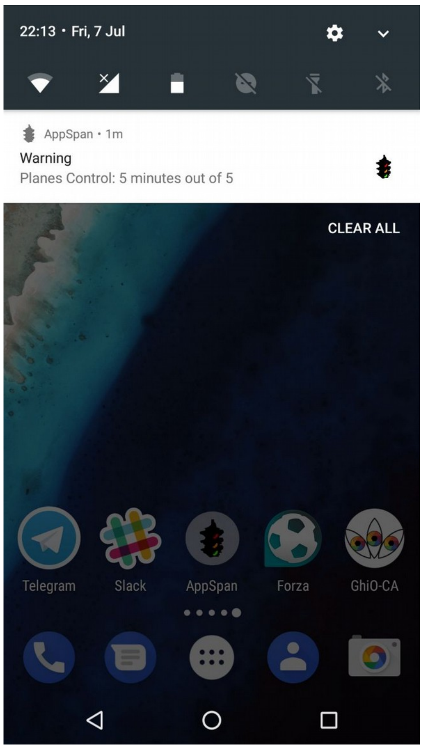
Every item on the list is clickable. When clicked, an item will open another activity where it is possible to add a time limit, delete the previous limit or update the current limit.



com.nousguide.android.rbtv
Current daily limit: 10 minutes
New daily limit: minutes

2.4 Notification

When a limit is reached or the user keeps using an app beyond the limit that has been set, a notification will warn the user. The notification specifies the app name, the usage time and limit.



AppSpan • 1m
Warning
Planes Control: 5 minutes out of 5

3 Project Architecture

3.1 SDK
API level with which the app is compatible: Minimum SdkVersion: 23, Target SDK version: 25.
The application has been tested with the following devices:

- Nexus5 - Android6.0.1 Marshmallow
- Nexus 5X - Android 7.1.2 Nougat

The choice depended mainly on the available testing devices.

3.2 Structure
3.2.1 AndroidManifest.xml

```
<uses-permission android:name="android.permission.PACKAGE_USAGE_STATS"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
```

In the manifest are declared the required permisisons. PACKAGE_USAGE_STATS, a system level permission to make the app able to read usage data. RECEIVE_BOOT_COMPLETED used to enable a monitoring service at device boot.

```
android:configChanges="orientation|screenSize|keyboardHidden|keyboard"
```

Configuration Changes Management inhibits the activities resetting on orientation changes and keyboard actions.

3.2.2 MainActivity
The main activity checks for PACKAGE_USAGE_STATS permissions, and eventually ask the user to enable it since is a system-level permission.
MainActivity renders the main view, it lets the user choose the desired time interval, through UsageStatsManager gets device usage history and statistics with wich renders the applications list using StatsAdapter to build the single items of the list.

3.2.3 StatsAdapter
This class gets name, icon, total time in foreground and last time usage and limit for all the applications currently installed on the device using the PackageManager. It renders these data for every list item.

3.2.4 OptionsActivity
By clicking any item in list it is possible to open the Options Activity. It displays the app package name, the icon and the current limit, if a limit is already set.
It lets the user set a new time limit, update the limit or delete the current limit.

3.2.5 DataBaseHelper
The class DataBaseHelper extends SQLiteOpenHelper, this class creates a sql database for the device and implements some useful database operations. Give a package name it is possibe to add, update and delete a time limit.
The DataBase is a SQLite database, version 3.9, stored in the device internal memory. It contains a table formed by a “package” column and a “minutes” column. The operations are implemented using SQL queries.

3.2.6 MonitorBootReceiver

MonitorBootReceiver extends BroadcastReceiver, a class capable of receiving broadcast messages from the Android system.
MonitorBootreceiver detects the device booting up and then starts a Monitor class service for monitoring real time apps usage.

3.2.7 Monitor

Monitor is a service that triggers the NotificationService class every minute. It is implmented as a service since it can run in background when the application is not active and does not need an interface.
Uses TimerTask, useful to shedule a repeated execution. To offload the TimerTask from the real time usage detection it calls a NotificationService.

3.2.8 NotificationService

This service manage information about the application with which the user is interacting. Runs in background and does not have an interface.
If the app time usage reach or is exceeding the limit it builds and shows a notification to the user.

3.3 Layouts

Three layout files are provided:

- activity_main.xml renders the applications list and buttons, used by MainActivity
- row_layout.xml renders an item of the list, used by StatsAdapter
- activity_options.xml lets the user interact with a specific app limit, used by OptionsActivity.

3.4 Notes

Javadoc documentation is provided.

4 Bibliografy

4.1 Links

- Mobile Programming http://www.math.unipd.it/~abujari/mprogramming.html
- Android Developer https://developer.android.com/
- Javadoc http://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html
- Android Nougat https://www.android.com/versions/nougat-7-0/
- Nougat for Developers https://developer.android.com/about/versions/nougat/android-7.1.html

4.2 Sources

Source Code https://github.com/aliceblack/AppSpanAndroid