

“The Brand Name”

A notification manager.

Purpose of the project

The purpose of the project is to scientifically study the user response of processing disruptive notifications of their smart devices.

The another objective is to offer something in the marketplace, that helps people in their everyday life. As a group of developers, we want to build something that has an impact to us as well. Every group member has their own idea how they want to satisfy a customer and we will combine most important features into the application.

Many studies have shown, that while notifications keep an user informed and engaged with events around mobile applications, they do not have the same importance level to an user and the timing of a notification is also a factor in the perceived importance.

The emphasis in the user experience of our application is to allow the most effective processing of notifications in the most user friendly fashion. However, advanced options in our application is meant to be accessible as well. For example, the user may select the delaying of notifications for the action, instead of just skipping them.

In the process of development we use a prioritization technique called the MoSCoW method, which allows the dynamic planning of the implementation. We are gathering all user feedback, which may affect also to minimum requirements. Getting enough user feedbacks is challenging, but we will put the effort to it in the early phase of the development to gain as much information as possible.

Group members

Name	Student number	Email
Markus Heino	2433882	Markus.Heino@studen.oulu.fi
Seppo Pakonen	2265199	mail@sppp.heliohost.org
Jaakko	-----
Mohammed	-----
The Project Manager	Heino Markus	Markus.Heino@student.oulu.fi

Table of contents

1. Glossary
2. Design process
3. State of the art
4. Scenarios and use cases
5. Requirements
6. Interface design
7. Analysis
8. Risk assessment
9. References
10. Contributions

Glossary

Push notification	A push notification notifies the user of new messages or events in the application. The devices might show an icon and a message in the status bar. Tapping the notification opens the main view of the cause.
The MoSCoW method	This is a prioritization method, which has four categories for requirements. Categories are <i>must have</i> , <i>should have</i> , <i>could have</i> and <i>won't have</i> .
The AWARE framework	AWARE is an Android framework dedicated to instrument, infer, log and share mobile context information, for application developers, researchers and smartphone users. AWARE captures hardware-, software-, and human-based data.

Design process

The group have been divided to the user interface team and the background system team.

The user interface team have created illustrations of menus of the application. They estimate different ways to show the most important aspects of the system while keeping the ease of use and the current state of smartphone applications in mind.

The background system team have focused to the system design and how to allow complex use cases later without making major changes to it. They have been reading publications about other similar studies and searching existing applications.

Together the group have been negotiating about the connections between the user interface and the system. Many practical issues have also been as subjects of discussions.

State of the art

In the literature, many studies covers notifications as a source of disruptions. They examine groups of people and their reactions to different notifications. In the results they take a stand on how people perceived notification importance or how to recognize automatically good times to push a notification.

The designed application as a notification manager have also been covered by many software patents. That might cause difficulties in exporting the application to USA.

The market place offers various similar solutions already. The most common feature is to allow blocking notifications of an application. Other features might include the customization of the sound and vibration or keeping the log of all notifications. Upcoming version of the Android operating system will also have a similar built-in feature.

- Notifications filter
<https://play.google.com/store/apps/details?id=com.tadevel.notification>
- Filter notifications
<https://play.google.com/store/apps/details?id=in.portkey.filter>
- New built-in adroid feature
http://www.phonearena.com/news/How-to-block-or-prioritize-an-apps-notifications-in-Android-5.0-Lollipop_id64612

Scenarios and use cases

Use Case	Delay annoying notifications.
Primary Actor	An user, who downloaded the app from the Play-store for free.
Scope	Android OS with NotificationManager running in the background or the foreground.
Level	User Goal
Brief	The user edits a setting which makes annoying notifications to be avoided.
Importance	This have to be implemented.
Postconditions	-
Minimal Guarantees	<ul style="list-style-type: none">• User gets the most important information about success or failure of the usage
Success Guarantees	<ul style="list-style-type: none">• The setting is saved and an updated version is loaded in the engine• The version snapshot of the setting is created by the system
Preconditions	<ul style="list-style-type: none">• The user have installed the application• The application is launched and the main menu is visible
Triggers	<ul style="list-style-type: none">• The user edits options which affects to initial settings• User enables advanced editing mode and adds more custom settings
Basic flow	<ol style="list-style-type: none">1. The system provides the menu, which shows the most relevant options about what and how notifications are processed.2. The user modifies options until satisfied.3. Settings are saved and loaded to the engine automatically
Extensions	-

Use Case	Make notifications less annoying
Primary Actor	An Android OS user with a Play-store account or with F-Droid installed.
Scope	The whole userspace of the Android OS with a Internet connection.
Level	Summary
Brief	By installing our application, people get a optimized dispatcher for notifications automatically or with very little configuring.
Importance	This could have been implemented.
Postconditions:	
Minimal Guarantees	<ul style="list-style-type: none"> • Users gets a unique footprint with their usual notifications
Success Guarantees	<ul style="list-style-type: none"> • Users gets a positive feeling about the smartphone taking their experience into account in unique fashion.
Preconditions	<ul style="list-style-type: none"> • The application is available in the Play store or in the F-Droid • The application can be installed and configured with less or equal steps as in average Android application
Triggers	<ul style="list-style-type: none"> • The user searches for ”notification” in the Play store • The user reads the list of system applications in the F-Droid.
Basic flow	<ol style="list-style-type: none"> 1. The market place connects an Android user to the application 2. The user installs and configures the application 3. The user feels the benefit of not being distracted without good reason
Extensions	<ol style="list-style-type: none"> a) Tune the user experience by customizing settings <ol style="list-style-type: none"> 1. The user selects advanced settings option 2. The system allows detailed tweaking of the setting b) Show public feedbacks about the application <ol style="list-style-type: none"> 1. The user selects Statistics menu 2. The system shows anonymized results of polls and other performance measurements c) Connect to the cloud <ol style="list-style-type: none"> 1. The user creates a account to our service 2. The system allows storing, loading and viewing existing settings

Requirements

Functionality	<p>For users:</p> <ul style="list-style-type: none">• Access to the Play-store, to the F-Droid or to our github frontpage• Android 2.3.3 (Android API 10) <p>For developers:</p> <ul style="list-style-type: none">• Android-Studio• Amd64 architecture PC or USB-connected Android device• Access to GitHub
User interface	<ul style="list-style-type: none">• Most important settings first
Usability	<ul style="list-style-type: none">• Application must be working as expected without additional unwanted features
Security	<ul style="list-style-type: none">• The feedback system must be anonymized• The connection to our server must be encrypted• The account service may not ask full name or other detailed credentials as they are completely unnecessary• Connecting the application with social media services is worse option than using the device ID as a login, because highly anonymized data and avoiding unnecessary liabilities is the primary objective.• Passwords must be checked by hash, not by plaintext
Management	<ul style="list-style-type: none">• Our group have meeting with our supervisor once in every two weeks• The group chat. Currently WhatsApp.
Standards compliance	<ul style="list-style-type: none">• Compliance with Play-store terms of service
Portability	<ul style="list-style-type: none">• The code must be written entirely in Java to use it with other architectures than ARM.

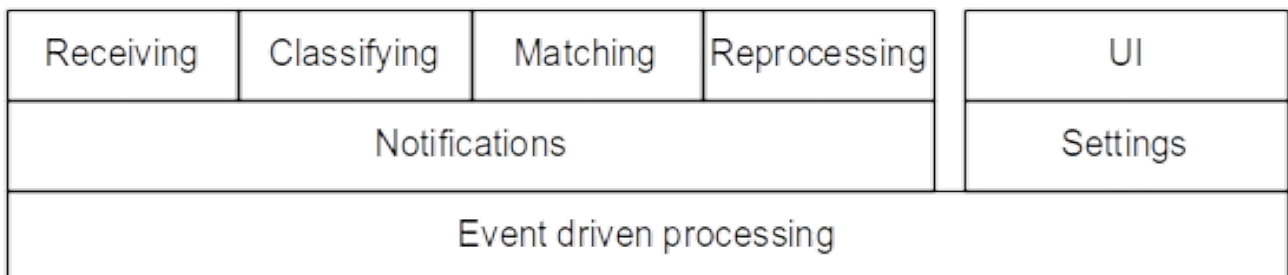
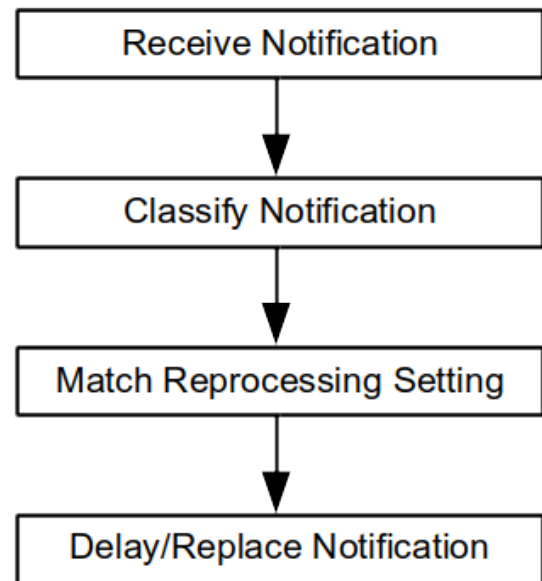
System design

The main task is to intercept the normal way of flowing of notifications and use our configurable processing instead.

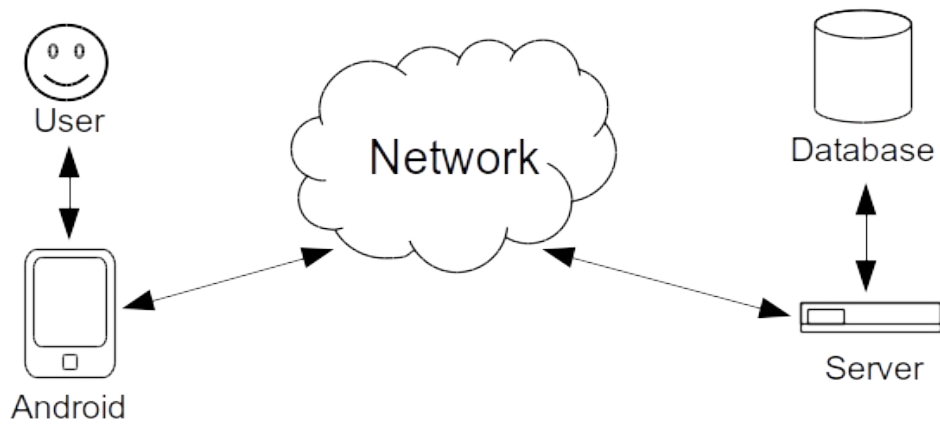
Notifications are being analyzed and classified. All information about the notification is collected and stored structurally in the memory.

The classified notification is then being matched to existing settings. If it does not match, then it will be forwarded and shown to the user as regularly. If it does match, then it will trigger the processing according to the setting.

The processing might cause the notification to be delayed, or modified or blocked entirely.



The whole application will be running from event based function calls instead of a fast loop, which is found e.g. in many games. If there is a need to call some function periodically, then scheduled system calls will be used.



While the application itself does not require a internet connection, the user feedback system requires. Gathering the evaluation data is one of our primary objectives, and it requires a separate server and database connected to the Internet.

Interface design

Please put here:

- *illustrations*
- *user comments*
- *why this*
- *what similarities it have to other android apps (visual references)*

Analysis

The annoyance of push notifications is scientifically known subject. Our project matches that in the analysing of the feedback data.

Our application is similar to the existing solutions in Play-store. Being at the same level or competing with them is in the reach of our implementation. Also, with more complex processing system this may be technically more advanced than them.

This project is a combination of known parts, both in marketing and scientific perspective, which makes it a good design.

Risk assessment

Brief	Not getting enough user feedback
Likelihood	Common
Impact	Minor
Preventive action	Making a working beta version in early phases of the development, and outreaching friends and relatives as test users.
Corrective action	We focus to data which doesn't require user interaction.

References

Publications used:

- Understanding Mobile Notification Management in Collocated Groups
http://link.springer.com/chapter/10.1007/978-1-4471-5346-7_2
- The myth of subtle notifications
<http://dl.acm.org/citation.cfm?id=2638759>
- Designing content-driven intelligent notification mechanisms for mobile applications
<http://dl.acm.org/citation.cfm?id=2807544>
- InterruptMe: designing intelligent prompting mechanisms for pervasive applications
<http://dl.acm.org/citation.cfm?id=2632062>
- Push or Delay? Decomposing Smartphone Notification Response Behaviour
http://link.springer.com/chapter/10.1007/978-3-319-24195-1_6

Patents regarding our application:

- Contextual Alarm and Notification Management
<https://www.google.com/patents/US20140253319>
- Temporal incoming communication notification management
<https://www.google.com/patents/US8855723>
- Systems and methods for push notification management
<https://www.google.com/patents/US20150120849>

Softwares used:

- AWARE framework
<http://www.awareframework.com/>
- Android Studio,
<https://developer.android.com/studio/>

Contributions

Name	Illustrations	Meetings	Use cases	Document	Research	Additional	Total
Markus	9h	8h	1h	8h	12h		38h
Jaakko	3h						
Seppo	3h	8h	12h	6h	12h		39h
Mohammed		8h	2h				