



FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

**Markus Heino**  
**Seppo Pakonen**  
**Jaakko Ikäheimo**  
**Mohammed Alanny**

## **Notification management**

Applied Computing Project 1  
Computer science degree program  
11/2016

## PURPOSE OF THE PROJECT

The purpose of the project is to offer something that helps everyone in their everyday life. Our goal is to ease the stress of our users and give them a chance to fully focus on what they want. That is the reason why we are designing this application, which lets notifications to be delayed in the way user wants.

During the project we want to push to our limits as a group and as individuals, and this way offer as much as we can to our potential future customers. At the same time we want to scientifically study the user response of processing disruptive notifications of their smart devices, and understand our users even better.

While notifications keep an user informed and engaged with events around mobile applications, they do not have the same importance level to an user. Also, the timing of a notification is a factor in the perceived importance. Any kind of modality associated with the notification have been shown to be attended 12 times more likely. [1] Delivering a notification immediately after finishing a phone call make users react them faster. Also transitioning between physical activities have been considered more positive situation to deliver notifications. [2]

Interviews have revealed that delaying group of low priority notifications until a specific time of day is a desired feature. [1] However, classifying content and context information of notifications have shown to lead to more accurate predictions of interruptibility of users, instead of user-defined rules. [3]

Our focus is in the desired user application. The emphasis in the user experience of our application is to allow the most effective processing of notifications in the most user friendly way.

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 Ikäheimo	2380964	j4sk4@hotmail.com
Mohammed Dalanny	2510129	mohammedalanny@gmail.com
<b>The Project Manager</b>	Heino Markus	<u><a href="mailto:Markus.Heino@student.oulu.fi">Markus.Heino@student.oulu.fi</a></u>

## TABLE OF CONTENTS

PURPOSE OF THE PROJECT

GROUP MEMBERS

TABLE OF CONTENTS

GLOSSARY

1.	DESIGN PROCESS .....	5
2.	STATE OF THE ART .....	6
3.	SCENARIOS AND USE CASES .....	7
4.	REQUIREMENTS .....	8
5.	SYSTEM DESIGN .....	9
6.	INTERFACE DESIGN .....	11
7.	USER COMMENTS .....	12
8.	ANALYSIS .....	13
9.	RISK ASSESMENT .....	13
10.	REFERENCES .....	14
11.	CONTRIBUTIONS .....	14

**Comment [1]:** These MUST be manually updated!!!

## 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>would be nice to 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.
NT	Notification filter is an app from play-store that is similar to ours.

## 1. 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.

The user interface desing have been done using Balsamiq Mockups 3 software. Simple prototype have also been made using Java and Java Swing library.

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.

The user interface have to be simple, but the system behind it does not need to be. While the detailed features of the system are being found in the implementation phase, we might want to test them via the user interface. We concluded that advanced features would be available in a separate menu.

The coarse system design is handled in a following chapter, but the more detailed information will be available only in the following implementation document.

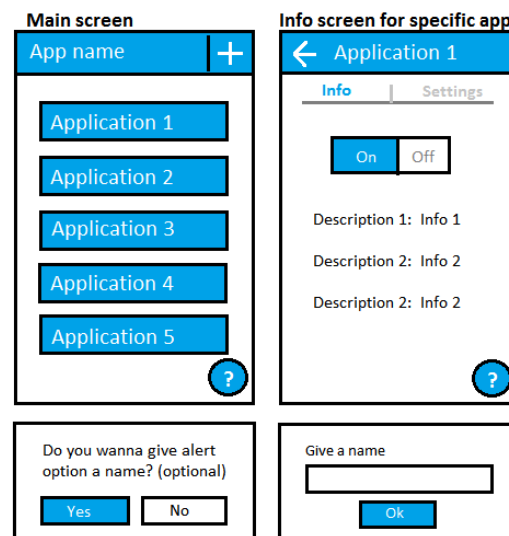


Figure 1. the main menu template

## 2. STATE OF THE ART

In the literature, many studies cover notification interruptions as a source of disruptions. [4][5][6] Users are susceptible to interruption overload. Waiting of an opportune moment is also a social behavior as interrupting a visibly concentrating person is considered rude. The interrupted user performs slower, experiences annoyance, anxiety and feels that the task is more difficult to complete. [5][6]

The general processing of notifications have been available in only in recent years. In Android, the `NotificationListenerService` class have been added in API level 18 (July, 2013). In iPhone, the similar feature is included in the Apple Notification Center Service, which have been available since September 2013.

Several notification manager applications have also been covered by software patents. [7][8] That might cause difficulties in exporting the application to USA. Some of those patents might cover our application entirely, but as a implementation to benefit users from getting overloaded by interruptions, our application is very unique.

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.

### 3. SCENARIOS AND USE CASES

As a prelude to use cases, one user story might be, when Mary have been very annoyed about notifications during school. All she wants is a one click solution to fulfill her need for focused learning. She don't want to go through all settings in all apps to silence all useless notifications. She thinks it is time for action.

The “sunny day” use case is when a consumer wants to delay his notifications while he is working. He goes to the Play-store and searches for notification manager. He finds our application and installs it. He runs the application, set his working hours and clicks 'delay'. All generally annoying notifications are then delayed until his working day ends.

The other use case is to filter working notifications after work. The steps are same as in previous, but instead of working hours he sets evening hours. Also he might want to customize settings and add messages from his employer and coworkers to be filtered.

A religious person with regular praying times might want to delay all notifications during praying time. He just adds multiple times and sets 'delay all'. Also a person going church regularly might want to do that.

The failing use case might be, when the user tries to delay all notifications during his working day, but notices that some important notifications have been delayed, which shouldn't. Also the working day of the user might end in different times and the time for notifications might be wrong too often. Some users might expect the more intelligent predictive opportunistic notification pusher and they would not be impressed about our application. They would turn around at the main menu.

## 4. REQUIREMENTS

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



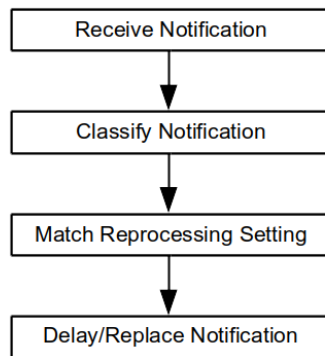
## 5. SYSTEM DESIGN

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.

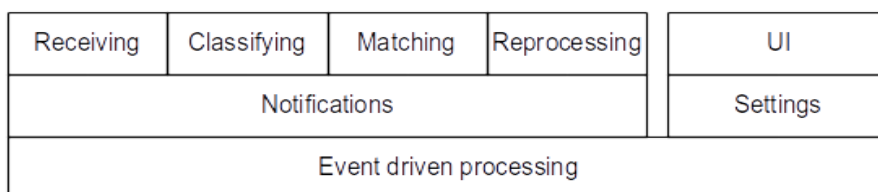
The classified notification is then being matched to existing settings. If it does not match, then it will be forwarded and shown to the user as regularly. If it does match, then it will trigger the processing according to the setting.

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



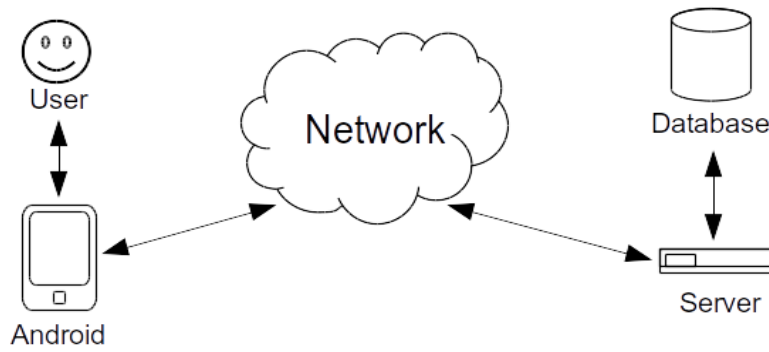
**Figure 2: the pipeline of notification handling**

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.



**Figure 3: the system hierarchy**

While the application itself does not require an 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.



**Figure 4: the network flowchart**

## 6. INTERFACE DESIGN

**Comment [2]:** Jaakko said that he have the text for this

The interface of the application is designed to be as simple and clear as possible so it is accessible to as many users as possible. The structure is also made to be simple and it is formed by two screens: the main screen and the application screen.

The main screen of the application consists of two main parts: the toolbar and the main container. The toolbar only contains the name of the application and a button to access help menu which is detailed later in this chapter. The main container contains application's main functionality which allows user to modify and change notification settings. As the first element main container has delay and date modifiers which allow user to change the amount of delay they want to have between notifications and dates when the notifications should be delayed. In addition the main container has a dropdown button for selecting one of the default delaying options: 30 minutes, 1 hour or 2 hours. The notification delay on/off buttons are switched if the user wants to have notifications delayed or not. The last element for the main screen is the button for app menu which is fully described later in this document.

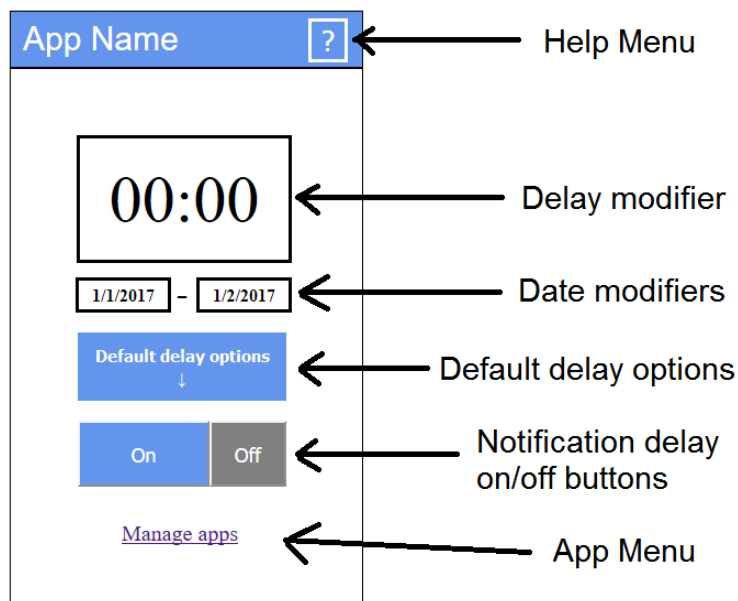
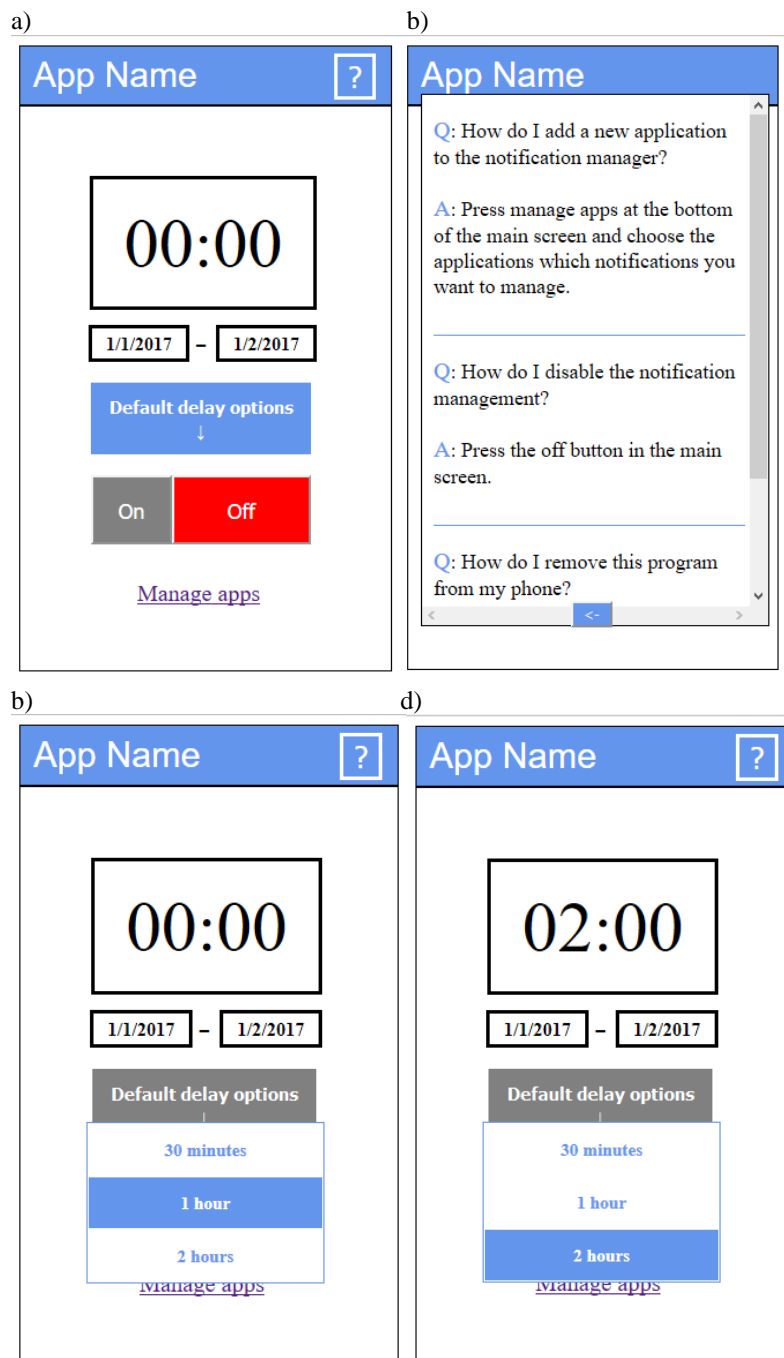
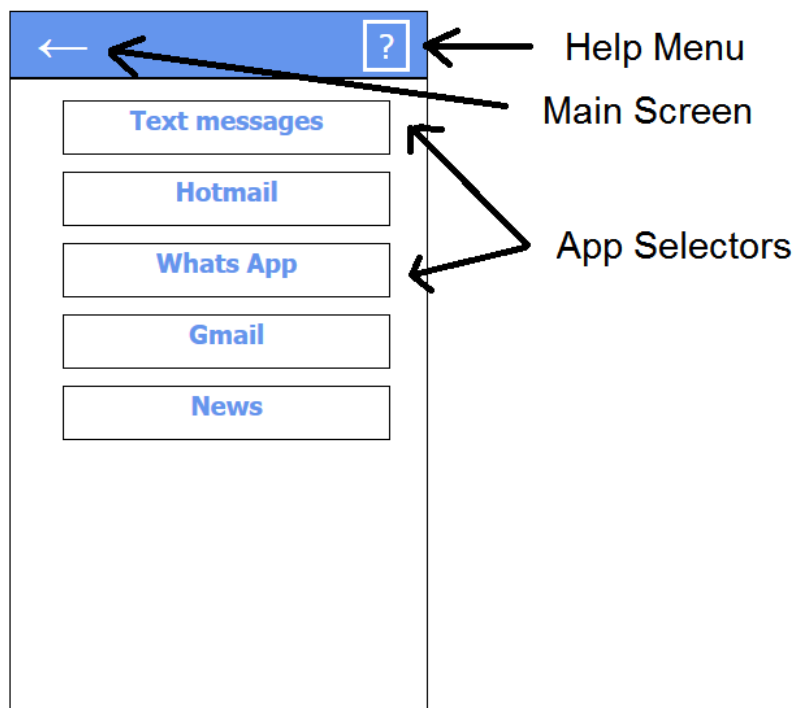


Figure 5: the main screen layout and explanations



**Figure 6: different states of the main screen: a) notifications off b) help menu c) default delay options dropdown-menu d) selecting default option**

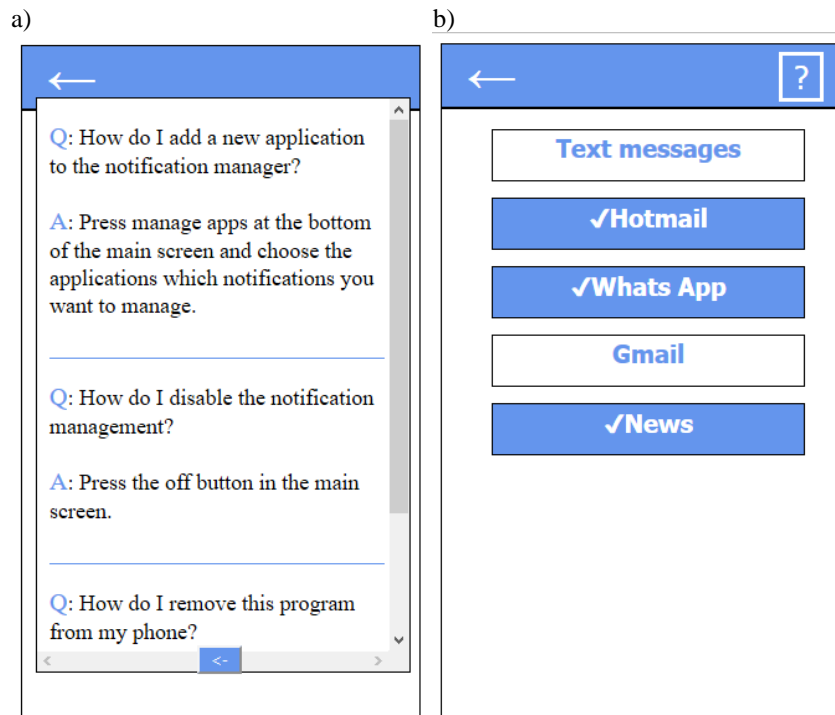
The application selection screen is very simple and contains only three main elements. The most important elements for this selection screen are the application selectors which control the applications user wants to enable or disable for the delaying. Disabled applications are displayed with white background and blue text, and enabled applications with blue background, white text and checkmark in front of the names. The arrow located to the left in the toolbar takes the user back to the main screen and the question mark works similarly to the one located in the main screen.



**Figure 7: the app screen layout and explanations**

The help menu for the application is pretty simple pop-up window which displays the most common problems and answers to them for the user. It is accessible through the question mark button in the toolbar in the both main and application selection screen. User can scroll down the help menu for more questions and answers. When

user wants to exit the menu there is a button with an arrow in bottom of the menu which enables user to exit whenever it is wanted.



**Figure 6: different states of the app screen: a) help menu b) some apps have been selected**

## 7. USER COMMENTS

To fully understand our future customers and their needs, we asked questions from our potential customers. The purpose of these questions was to give us more information about what kind of features users wish to have and what options the app should have to satisfy the needs of our clients.

Conversations with every potential client took only around 10-15 minutes to keep answers neutral. The answers pointed to the most important features in our app instantly, but they can also inspire in our future research.

Basically we asked:

- *“If you had an app that could delay Facebook or Emails for a time, at what time would you use this ?( referred to examples like work and school)”*
- *“Do you think this could help you to focus better on your workday or maybe in a meeting, if you couldn’t be disturbed by your phone unless it is an emergency?”*
- *“Which apps would be most important to delay and why these?”*
- *“Do you have anything you would like to add, now that you have a full view of what you could have”?*

After feedback here are the main points our potential clients had:

- *“It would be actually very nice to have an option to delay my facebook , Email, Whatsapp, notification when I go to class, but I’m worried that I won’t remember to turn it back on”*
- *“The settings I make in the app should be saved so I don’t have to change them every time”*
- *“App should be easy to use, like it could have a button to turn it on and off, maybe widget ?”*
- *“Delaying for an hour or to only and then the app would turn off by itself would be nice so I could just select “delay for 2 hours” and that’s all”*

After research analyses, we are more enlightened as a group of our potential customer needs. We have a full picture of the features people want and how they should work in order to satisfy our clients in a best way possible.

## 8. ANALYSIS

The annoyance of push notifications is scientifically known subject. The response time to notifications have been measured [1][3] and the subjective experience of the user have been polled [2]. Our application will do both of those measurements.

There are some similar application in Play-store [9][10]. Our application have the unique feature of delaying notifications.

Notification Filter [9] was the first similar application we found. The first impression after opening the app is that you have to be using this app and you had to modify your apps. This is not what we want at all. We want that the user has an option to use the app if she/he wishes. There is no need to use it all the time and keep consuming your battery for nothing. Also NT (notification filter) had three steps that you had to do before using the app. We want offer max 3 different side. The front page that includes easy access, on/off, the time for how long to delay. Second page is for settings. Which apps you can delay and the checking is easy as it could be as shown in interface design. The third page is for more detailed settings for certain apps. This page will be for more advanced users so not a mandatory for user.

**Comment [U3]:** differences added

**Comment [U4]:**

**Comment [U5]:** Checking the boxes in a picture is something that should be found from interface design pictures

As overall we want the user feel like this is just one app to help her/him in everyday life without thinking what kind of data will be this app collecting from me and how much it consumes my battery. We want to offer something that is easy to use, don't feel mandatory and can be used only when you want!

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

## 9. RISK ASSESMENT

Brief	The amount of user feedback is not enough
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.

Brief	Functionality is not easy enough
Likelihood	Common
Impact	Moderate
Preventive action	Gathering as much data from potential users as possible. Making the app as simple as it can be.
Corrective action	Changing functionalities after feedback to satisfy customers.

**Comment [U6]:** New risk assessment added



## 10. REFERENCES

- [1] The myth of subtle notifications. In UbiComp '14 (September 2014)
- [2] Investigating episodes of mobile phone activity as indicators of opportune moments to deliver notifications. In MobileHCI '11 (September 2011)
- [3] Designing content-driven intelligent notification mechanisms for mobile applications. In UbiComp '15 (September 2015)
- [4] If Not Now, When?: The Effects of Interruption at Different Moments Within Task Execution. In CHI '04 (April 2004)
- [5] The Effects of Interruptions on Task Performance, Annoyance, and Anxiety in the User Interface. University of Nevada, 2001.
- [6] Notification, disruption, and memory: Effects of messaging interruptions on memory and performance. Microsoft study (January 2001)
- [7] Temporal incoming communication notification management. Patent US 8855723 B2
- [8] Systems and methods for push notification management Patent US 20150120849 A1
- [9] Notifications filter. Android Play-store.
- [10] Filter notifications (beta). Android Play-store.

## 11. CONTRIBUTIONS

Name	Markus	Jaakko	Seppo	Mohammed
<b>Illustrations</b>	9h	20h	3h	
<b>Meetings</b>	8h	8h	8h	8h
<b>Use cases</b>	1h	1h	16h	2h
<b>Document</b>	8h	2h	15h	
<b>Research</b>	12h	4h	12h	10h
<b>Additional</b>		14h		10h
<b>Total</b>	38h	49h	54h	30h

### Markus Heino

I started with mockups with Jaakko. In the first meeting we decided what kind of mockups will be the best for us and Jaakko started to modify mockups the way group decided. I started to work more on the design template and making sure template has everything it should. First a lot of research from design templates from others and then making very coarse version of my own for the group. Version had everything explained what we should have in our template and which ones are optional but from perspective mandatory for us. Template started to change after we added content on it and have been changing during the whole project. My main goals have been cover page, purpose of the project, analysis, risk assessments, mockups and making sure we have everything on the template that should be and even a bit more. Everything been proofread by every member and there for text have been changing a bit from every section.

### Jaakko Ikäheimo

From the beginning me and Markus decided to work on illustration and design of the application's interface. I started to work on the prototype and Markus with research and documentation. After evaluation and some changes I came up with a complete prototype and started to work with documentation and interface design part in the template. The most important part for me in design phase has been trying to come up with a good design for interface and improving the prototype and my designing skills while doing that. I have been also trying to modify the document so it is more readable and adding some stuff here and there. You can see the prototype we used for our design on: [jaskaweb.com/proto\\_main.html](http://jaskaweb.com/proto_main.html)

### Seppo Pakonen

Together with Mohammed we formed the background system team, which planned use cases and system design. We discussed a lot of the content before writing. I wrote most of the design document. I wrote partially or entirely the purpose of the project, design process, state of the art, scenarios and use cases, requirements, analysis and risk assesment. I researched and wrote all references to publications and formatted the text to thesis format. I was heavily corrected, though. Much learning was required for this paper.