

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

**Markus Heino   
Seppo Pakonen  
Jaakko Ikäheimo**

**Mohammed Alanny**

**Notification management**

GROUP MEMBERS

|  |  |  |
| --- | --- | --- |
| **Name** | **Student number** | **Email** |
| Markus Heino | 2433882 | Markus.Heino@student.oulu.fi |
| Seppo Pakonen | 2265199 | mail@sppp.heliohost.org |
| Jaakko Ikäheimo | 2380964 | j4sk4@hotmail.com |
| Mohammed Alanny | 2510129 | mohammedalanny@gmail.com |
|  |  |  |
| **The Project Manager** | Heino Markus | [Markus.Heino@student.oulu.fi](mailto:Markus.Heino@student.oulu.fi) |

Applied Computing Project 1

Computer science degree program

2/2017

TABLE OF CONTENTS

GROUP MEMBERS

TABLE OF CONTENTS

INTRODUCTION

GLOSSARY

[1. Implementation PROCESS](#__RefHeading___Toc179888587)

2[. Software architecture](#__RefHeading___Toc179888587)

[3. data structures](#__RefHeading___Toc179888587)

[4. hardware documentation](#__RefHeading___Toc179888587)

[5. user interface](#__RefHeading___Toc179888587)

[7. third party materials](#__RefHeading___Toc179888587)

[8. security and privacy](#__RefHeading___Toc179888587)

[9. project risk assesment](#__RefHeading___Toc179888587)

[10. REFERENCES](#__RefHeading___Toc179888587)

[11. Contributions](#__RefHeading___Toc179888587)

Introduction

Our goal is to offer an application, that helps smartphone users in their everyday life. We want to ease the stress of our users by giving them a chance to delay their notifications, time chosen by the user. We want people to focus on what is important on given time and focus on notifications afterwards. On the way we want to collect user data and focus on their behavior, and this way improve the app to satisfy customer needs best way possible.

We are focusing a lot on interface to make user feel like they can rely, that app does its job (Delays notifications the way user sets). Functionality is simple and therefore app is very easy to use. Interface has only few buttons and setting time/date is easy as clicking few buttons. We are offering a tool that can help everyone, and they have the option to use it in their own way.

We do believe that this application can help more or less everyone. Collecting data will help us improve the application and its functionality during the project and afterwards.

Our app can be used in various ways and therefore not all usage might not be as good as others. We have our concerns that app might be used by father/moms to set delay notifications on their children phones without them knowing. Also we fear that app might be used for cheating and covering it up by delaying notifications.

Unfortunately our app can’t fix everything, but for sure it helps to organize your thoughts as well as chores better. It gives user and opportunity to focus on something (kids, meetings, lessons, etc.) 100% and not get disturbed by smartphone notifications.

We don’t judge anyone and we simply just offer a tool to help user in their everyday life. It’s up to user to choose how he wants to use it. Just to remind delaying does have its affects in your life since an important e-mail or phone call might be missed. Make sure you know the impact and on whom it affects.

1. Introduction (mandatory)

* A succinct overview of the project’s purpose.
* Brief description of the system and its organization.
* Major assumptions and constraints.

1. Glossary (mandatory)

* Definitions, acronyms, abbreviations and symbols.

1. Implementation process (mandatory)

* A description of the steps that you have taken in your implementation, possibly in form of a flow chart.
* If you have followed a specific development paradigm, please describe it.

1. Software architecture (optional, mandatory for projects involving substantial software development and/or reuse)

* Implemented software architecture.
* Overview of the logical organization and interaction of software components.
* Other aspects, for example:
  + - Subsystem decomposition (description of the subsystems comprising the system, along with a description of the responsibilities of each subsystem).
    - Software/hardware mapping.
    - Persistent data management.
    - Access control and security.
    - Software control (synchronization of subsystems, concurrency control).
    - Boundary conditions (startup, shutdown, error situations).
* Subsystems (design of each major subsystem identified above).
  + Architecture (e.g. a known pattern or a new design).
  + Interfaces or APIs within subcomponents.
* Use applicable diagrams in documenting software architecture, e.g. the UML notation provides many candidate diagrams:
* Structural diagrams: class diagram, object diagram, component diagram, deployment diagram.
* Behavioral diagrams: use case diagram, sequence diagram, activity diagram, collaboration diagram, state diagram.
* Model management: packages, subsystems, models.

1. Data structures (optional, mandatory for projects involving substantial software development and/or reuse)

* Description of data formats you have used or implemented.
* Database schemas.
* File storage documentation (if you are using a non-standards file type).

1. Hardware documentation (optional, mandatory for projects developing or relying on specialized hardware)

* Description of the specialized hardware’s purpose and functionality.
* Include blueprints if you built the hardware.
* Description of important operational parameters necessary for the operation of the hardware.
* Description of variables affecting the hardware’s performance and data generation.
* Description of important APIs that are necessary to use the hardware.

1. User Interface (optional, mandatory for projects involving highly interactive systems)

* Interaction: show in graphical form the interaction flow of the system, the various screens or states it haves, and how users transition/navigate the system.
* Screenshots: provide a small number of screenshots that demonstrate how they system looks to the end user.
* Describe any tools that were used to implement the user interface.
* Summary of design guidelines and/or design patterns used.

1. Third party materials (optional, but contributes to grading)

* Description of commercial software, hardware, or other material used in your implementation
* Document the version numbers, series numbers, API used, etc.
* Describe the material in enough detail so that it is possible to identify a replacement for that component. Do not give more details than necessary.

1. Security and privacy (optional, but contributes to grading)

* Identify security weaknesses of your system. How can a malicious entity break your system?
* Identify privacy weaknesses of your system. Does it store sensitive data that could be leaked/exposed?

# **PROJECT RISK ASSESMENT**

|  |  |
| --- | --- |
| Brief | The amount of user data collected is not enough |
| Likelihood | Common |
| Impact | Minor |
| Preventive action | Making sure we have enough test users and data collected to evaluate. |
| Corrective action | We focus to data we have enough collected for evaluation. |

|  |  |
| --- | --- |
| 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. |

|  |  |
| --- | --- |
| Brief | Project member leaving |
| Likelihood | Unlikely |
| Impact | Major |
| Preventive action | Keeping the atmosphere in a good level during the whole project. |
|  |  |
| Corrective action | Trying to change the mind of leaving person or to keep going without one (depending what is the reason for leaving) |

|  |  |
| --- | --- |
| Brief | Implementation |
| Likelihood | Common |
| Impact | Moderate |
| Preventive action | Making sure the whole group does have enough information and research. Helping each other the best way possible. |
| Corrective action | ---- |

|  |  |
| --- | --- |
| Brief | Project doesn’t finish on time |
| Likelihood | Unlikely |
| Impact | Major |
| Preventive action | Setting goals and keeping group informed. |
| Corrective action | --- |

# **References**

# **Contributions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | Markus | Jaakko | Seppo | Mohammed |
| **Interface** | h | h | h | h |
| **Functionality** | h | h | h | h |
| **User Data** | h | h | h | h |
| **Meetings** | h | h | h | h |
| **Document** | h | h | h | h |
| **Research** | h | h | h | h |
| **Additional** | h | h | h | h |
| **Total** | h | h | h | h |