Requirements and Analysis Document for NNN

Version: 0.5

Date: Friday, March 24, 2017

Author: Felix Nordén

This version overrides all previous versions.

# Introduction

Background explaining why this application is needed (besides mandatory in course). What's the problem addressed (use imagination)? What will it do? Who will benefit/use from this application? In what situation will the application be used? Defining the application. General characteristics of application.

In today’s society, where each individual gets more and more involved, the 24 hours a day we have at hand feels smaller and smaller. Being able to fit everything of interest into our day becomes more difficult and the time we can allocate is becoming smaller by the day.

Sleep deprivation has become a necessity and is a merit when applying for any form of profession.

## 1.2 Definitions, acronyms and abbreviations

Create word list to avoid confusion.

Todo-list – A list of different tasks that the user wishes to get done.

Task – The simple item that the user can add to his/her todo-list

List task – A more complex version of a task, which creates both a category with the List task name and also contains a list of different tasks.

# 2. Requirements

## 2.1 User interface

Sketches, drawings and explanations of the application user interface (possible navigation).

## 2.2 Functional requirements

What will the user be able to do ? Write a list of use case names (id’s) in the language of the customer. The specific flows for each use case is recorded below. Specify a use cases in priority order.

##### First Priority

* Create a new task in the form of
  + General Task
  + List Task
* Check off a task
* Filter the list to a certain category

#### Second Priority

* Update a task’s configuration
* Sort task in another order
  + Priority level
  + Chronological order
  + Alphabetical order
* Update/clean up list of done tasks
* Create new filters/categories through:
  + Task creation
  + Static category creation

#### Third priority

* Change settings
* Check for help
* (Check achievements)
* (Use Advanced tools when creating Items)
  + Set tags for time, priority, category and possibly position

## 2.3 Non-functional requirements

Any special considerations ​besides functionality? Usability, reliability, performance,​ supportability, legal, implementation, … NOTE: ​Testability​ mandatory (must have tests)

* Main Categories
  + Time
  + Tags
* Standard Subcategories
  + ”Today”
  + ”Tomorrow”
  + ”7 days”
  + ”All”
  + ”Home”
  + ”Work”
  + ”Meetings”
* Support for different Android devices from OS ”Lollipop” and upwards
* Support for both tablets and smartphones
* Implementation through Java
  + Test-driven development
  + MVC-focused implementation model
  + Object Oriented implementation
* Usability
  + Simple introduction at first startup of application
  + Focused on simple use
  + Implemented functionality for more advanced usage in ”Creation mode” for tasks
* Entertainment
  + Make use of gamification through Achievements
  + Create interesting representations for statistics

# Use cases

An UML​ use case diagram

## 3.1 Use case listing

Use case texts (using the use case template)

# Domain model

An UML ​class diagram​.

## 4.1 Class responsibilities

Explanation of responsibilities of classes in diagram

# References