Requirements and Analysis Document for NNN

Version: 0.1

Date: Thursday, March 23, 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.

# 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 Item
  + List Item
* Check off a task
* Filter the list to a certain category

#### Second Priority

* Update an item’s configuration
* Sort task in another order
  + Priority level
  + Chronological order
  + Alphabetical order
* Update/clean list of done tasks
* Create new filters/categories through:
  + Item creation
  + Static category creation
* Delete a task, instead of checking it off

#### Third priority

* Change settings
* Check for help
* (Check achievements)

## 2.3 Non-functional requirements

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

# 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