



# Requirements List 3rd ed.

🕒 Created	@September 16, 2021 11:33 AM
👤 Created By	
🕒 Last Edited Time	@September 20, 2021 12:34 PM
👤 Minutes Secretary	
📅 Next Meeting	
👤 Participants	
👤 Scrum Master	
🗳 Type	Ad Hoc

## Goals

- Complete the requirements list

## Discussion Items

- Functional requirements
  - MoSCoW division
  - SMART criteria
- Non-function requirements

## Requirements List

### Must have

1. The user **MUST** have the option to create a list of tasks he wants to do during breaks. These tasks can be things the user enjoys doing, chores the user has to do, or anything else the user wants to spend time on during breaks;
2. The application **MUST** have a standard list of tasks. These fall in one of the following categories: healthy, social, personal, and cleaning. These tasks have default settings;
3. The user **MUST** be able to change the settings of his own tasks and the settings of the standard tasks. These settings include:
  - a. The estimated time each task takes;
  - b. The preferred time for each task;
  - c. The priority of the task:
    - i. low priority;
    - ii. normal priority;
    - iii. high priority;
    - iv. must be done today;
  - d. If the task can be done once or if the task is periodic;
4. The application **MUST** be able to select a task for the user. This selection is random, but it takes the priority of the task into account. A task with a high priority is at least two times more likely to be selected than a task with a normal priority and at least four times more likely to be selected than a task with a low priority;
5. The user **MUST** be able to give input about the status of the task. This way, the application always knows the status of the task. A task can have one of the following statuses:
  - a. The task has been snoozed;
  - b. The user is ignoring the notification and not doing the task;
  - c. The task is being done;
  - d. The task is done;
6. The application **MUST** be able to determine a suitable time to take a break, during work time. This can be after a fixed time. The time starts after you've finished your

last break. The default time between breaks is 45 minutes. The user can manually change this time;

7. The application **MUST** give the user a pop-up notification with the selected task when it's time to take a break;
8. The application **MUST** know the current time;
9. The user **MUST** have the option to manually make a basic agenda for the day. The agenda can be changed at any time during the day. There are four possible options in the agenda:
  - a. No work time: This means the functions of the application are turned off and the application does and measures nothing;
  - b. Work time: This is the time the user is working and the time the user will be periodically reminded by the application to do tasks;
  - c. Planned break times: This is the time the user has lunch or some other form of a break, which is scheduled in advance. The user won't get any tasks during this break and the time before the next task, resets after the planned break is over;
  - d. Do not disturb time: The user has a meeting or another work-related activity where he doesn't have time to take a break and do a task, so the user won't get notifications during this time. If it was time for a task during the 'do not disturb' time, the user will get a task right after.

## Should have

1. The user **SHOULD** have the option to redo a task sometime soon, so that they can enjoy that task again soon;
2. The user **SHOULD** have the option to skip a task so that they can do something else that has their utmost priority or continue the activity that doesn't let them do the task;
3. The user **SHOULD** have the option to give feedback to a task so that they could receive the task more often or less depending on how much they like it;
4. The user **SHOULD** be able to indicate (in the settings and through indirect feedback, e.g., via completion, skip, or feedback rates) how often these breaks

happen so that they enjoy doing these tasks rather than receiving too many right after each other or never receiving any of them;

5. The user **SHOULD** be able to see their to-do list so that they know which and how many tasks they can expect;
6. The user **SHOULD** be able to see their agenda so that they know how the day looks like and what time it is;
7. The user **SHOULD** have the option to do a task or take a break before the application tells them to so that they can fit their agenda a bit better and they don't receive a task right after they did a task or took a break.

## Could have

1. The application **COULD** provide two other tasks so that the user can choose to do another task if the first task is impossible to do at that moment;
2. The user **COULD** see an animation or short video after they completed a task so that they will feel accomplished and receive motivation to go about their day;
3. The user **COULD** snooze a task for a couple of minutes for a maximum number of times determined by the user so that they continue with their workflow without any other notifications for other tasks;
4. The user **COULD** snooze to reschedule a task to another time so that they fit their agenda a bit better and can receive another task in the meanwhile;
5. The user **COULD** see their progress on how many tasks were completed, skipped, or rescheduled in form of graphs so that they can reflect on their performance and tweak their personal needs;
6. The user **COULD** often expect tasks at suitable/optimal times so that they do not have to skip, snooze, or reschedule their tasks;
  - a. not too close to do not disturb me times
  - b. not right after a forced break
7. The user **COULD** expect tasks to consider the weather forecast so that they do not perform difficult tasks in the scorching heat, freezing weather, or thunderous rainstorms;

8. The user **COULD** import their external calendar in .ics format so that the application can incorporate the activities from the calendar more efficiently while scheduling tasks.

## Won't have

1. The application **WON'T** be able to add tasks to and/or manage tasks from external calendar services. It will only be able to import information from other calendar services;
2. The application **WON'T** rely on machine learning. Some aspects of the application will be user-specific (i.e. the time between tasks or breaks), but mostly the behavior of the application will not adjust to the behavior of the user;
3. There **WON'T** be a build-in option to export information or results for easy sharing;
4. The application **WON'T** be able to control or inform other programs in any way. It won't block incoming calls when doing tasks or tell other programs the users availability;
5. The application **WON'T** need a continuous internet connection to work.

## Non-Functional

1. At least 3 people (including the coder) must have reviewed the code before any commits are pushed/merged;
2. All uploaded/changed code must be reviewed within 1 week;
  - a. quality of code;
  - b. logic behind code;
  - c. syntax;
  - d. private/protected variables
3. Everyone tests/runs every feature of the application in real-life scenarios;
  - a. invalid input;
  - b. normal (stress) test
4. The code may not slow down the computer significantly while in use.