

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

SMART TO-DO LIST

Version 1.0

Abdul Ahad (CS-17022)

Syedah Nawal Munif (CS-17024)

Sanjna Rajani (CS-17089)

*Department of Computer and Information Systems
NED University of Engineering and Technology*

Submitted to
Ms. Fakhra Aftab

Novemeber 21, 2019

Contents

1	INTRODUCTION	3
1.1	DOCUMENT PURPOSE	3
1.2	PROJECT SCOPE	4
1.2.1	DESCRIPTION	4
1.2.2	BENEFITS	4
1.3	INTENDED AUDIENCE	4
1.4	DOCUMENT OVERVIEW	4
1.5	DEFINITIONS, ACRONYMS AND ABBREVIATIONS	5
1.6	DOCUMENT CONVENTIONS	5
1.7	REFERENCES AND ACKNOWLEDGEMENTS	5
2	OVERALL DESCRIPTION	6
2.1	PRODUCT PERSPECTIVE	6
2.2	PRODUCT FUNCTIONALITY	7
2.3	USERS AND CHARACTERSTICS	7
2.4	OPERATION ENVIRONMENT	7
2.5	DESIGN AND IMPLEMENTATION CONSTRAINTS	8
2.6	USER DOCUMENTATION	8
2.7	ASSUMPTIONS AND DEPENDENCIES	8
3	SPECIFIC REQUIREMENTS	9
3.1	EXTERNAL INTERFACE REQUIREMENTS	9
3.1.1	USER INTERFACES	9
3.1.2	HARDWARE INTERFACES	9
3.1.3	SOFTWARE INTERFACES	9
3.1.4	COMMUNICATION INTERFACE	10
3.2	FUNCTIONAL REQUIREMENTS	10
3.3	BEHAVIOUR REQUIREMENTS	13
3.3.1	USE CASE VIEW	13
4	OTHER NON-FUNCTIONAL REQUIREMENTS	14
4.1	PERFORMANCE REQUIREMENTS	14
4.2	SAFETY AND SECURITY REQUIREMENTS	14
4.3	SOFTWARE QUALITY ATTRIBUTES	14
4.3.1	EFFICIENCY	14
4.3.2	USABILITY	15
4.3.3	ECOSYSTEM	15
4.3.4	OPERATIONAL	15

Chapter 1

INTRODUCTION

Smart to do list is an open source to-do manager specially designed to help users manage their tasks and projects without going through the process of typing. It is available in English and works majorly on voice commands. The project has been developed in C#.

1.1 DOCUMENT PURPOSE

A Software Requirements Specification (SRS) is a document that describes what the software will do and how it will be expected to perform. This Software Requirements Specification provides a complete description of all the functions and specifications of the smart to do list. Todo apps are widely used, but different users use the apps differently depending on their particular needs. The main purpose of this smart to-do list is that it provides ease to users as it helps users to set their tasks or any important meeting to be reminded of by using their voice commands and the algorithm then processes the natural human language into computer understandable language and set the reminder of the particular task for the user. It reduces human efforts of typing and without going through the process of providing every detail to the system, the user can just say the sentence along with date and time and the system will automatically set it for user.

1.2 PROJECT SCOPE

1.2.1 DESCRIPTION

To do list is a practical organizing tool for all types of users. For common users or business oriented users , as it has the capability to work as a plain manager or as a professional to-do manager.

1.2.2 BENEFITS

As a plain to do manager, user's tasks can be saved and reminded in hierarchical form. While, as a professional to do manager, timers are set which can aid to remind of the meetings schedules and also of the approaching deadlines. here are many benefits of To Do lists, from helping you prioritize your time to helping you remember what it is you meant to do! Manage tasks effectively. You can see all of the items at a glance and prioritize, what most needs to be done according to timelines and importance.

1.3 INTENDED AUDIENCE

The SRS document is addressed to:

- Developers who want to extend the program with new features.
- Testers who are interested in discovering possible flaws of the program and want to report them for improvement.
- All users of the program, who are interested in being informed about the capabilities, which smart to-do list, gives to them.

1.4 DOCUMENT OVERVIEW

The rest of this SRS

- chapter 2 gives a general description of the project. It gives what level of capability is anticipated from the client, some broad requirements while making the product and a few suppositions and conditions that are accepted.
- chapter 3 contains most significant highlights of the project portraying detail description and requirements. It explicitly states the requisites which the product is required to convey. Functional requirements are given in this section. This area is composed essentially for the developers and portrays in specialized terms the functioning of the project, security performance and execution.
- chapter 4 consists of non-functional requirements describing them in detail.
- chapter 5 comprises of database information.

1.5 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

SQL Server

SQL stands for Structured Query Language . SQL is a database computer language designed for managing data in relational database management systems (RDBMS), and originally based upon relational algebra and calculus. Its scope includes data insert, query, update and delete, schema creation and modification , and data access control.

.NET framework

.NET Framework is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library named as Framework Class Library and provides language interoperability across several programming languages.

1.6 DOCUMENT CONVENTIONS

In general this document follows the IEEE formatting requirements. font size 12 is used throughout the document for text. Document text is single spaced .no special formatting techniques are used.

1.7 REFERENCES AND ACKNOWLEDGEMENTS

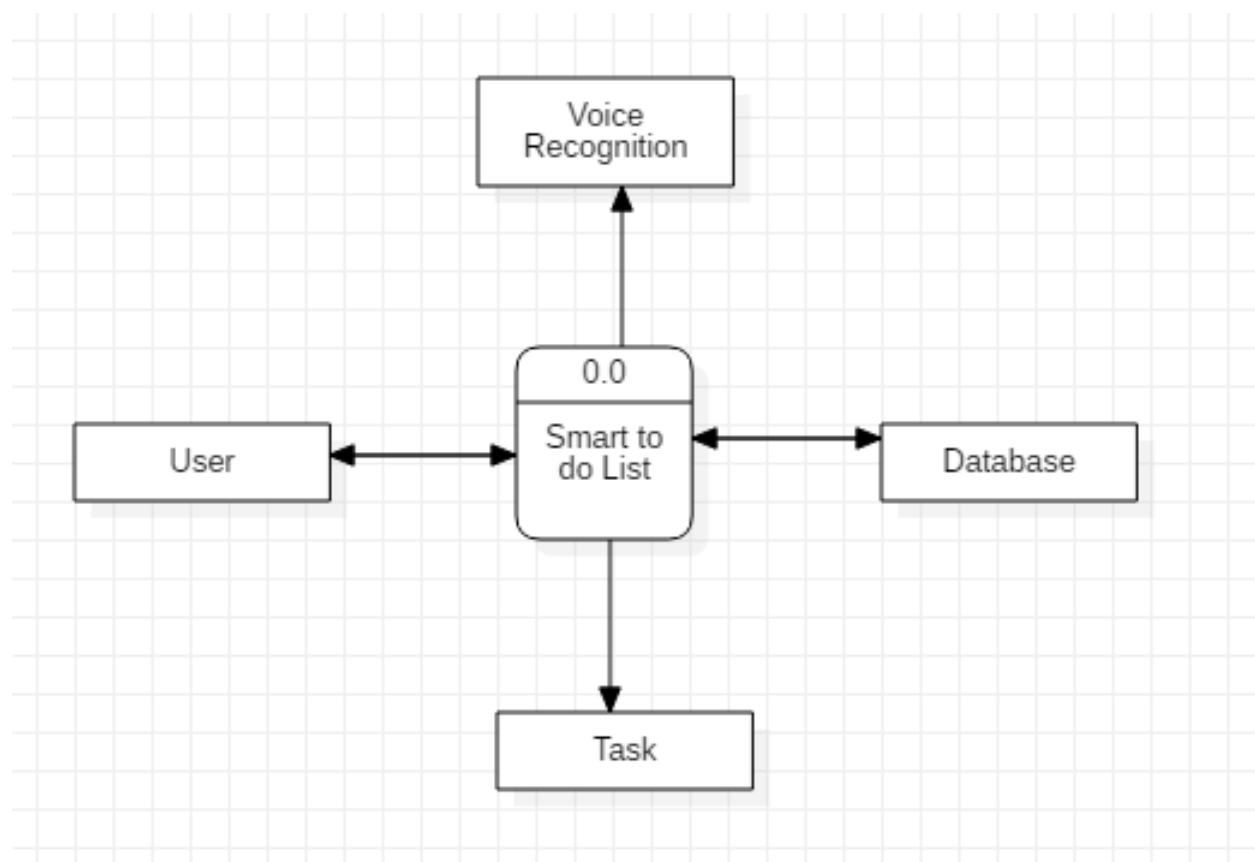
- <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>
- <http://todomoo.sourceforge.net/>
- <https://todoist.com/app>

Chapter 2

OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

The product is supposed to be an open source. It is a desktop based system enhancing user to make their lives productive and manage time. This product is a follow-on and improved version dealing with certain ML algorithms which helps user to just create a reminder by voice recognition without the need of typing anything. It reduces the efforts of typing and will provide user delightful experience. Be it a meetup or some random office work, it will help in reminding every scenario with just human voice.



2.2 PRODUCT FUNCTIONALITY

The Intelligent ToDo System provides a simple mechanism for users to organize their daily routine by saying something like “Remind me to bring vegetables at 2pm” and it will smartly set the reminder and will prompt it to the user.

- Set a reminder to organize your day
- Set an alarm for meetings, milestones etc
- Reduce the effort of typing
- Suggests time for the completion of task
- Prioritize the most important task first
- Constantly reminding you to complete pending tasks (if any)
- Works with voice recognition

2.3 USERS AND CHARACTERSTICS

This product will cover all the end-users whether a student or a company manager. Anyone can handle their tasks on the basis of their daily life functionalities. This will generally help people like manager or CEO, it basically works like a secretary as it will remind all the important meetings and milestones and focus on the most important conferences and piece of work first, prioritize on the basis of the company’s benefit. Students and people at home can find it useful to manage their daily life routine by setting an organized plan on the product with the ease of using their voice only to set the tasks. Anyone from the company’s CEO to a teenage student, everyone can find this easy and handy who wants to make their lives organized and productive by keeping track at all times.

2.4 OPERATION ENVIRONMENT

This will work in any desktop environment, whether be it Windows, Linux , MacOS etc . There’s no explicitly system requirements as it is a very easy and not a hardcore software program. It is a local product so it doesn’t require any sort of server to store database. Everyone has their independence to work in the environment on their local PCs and the database is stored in the PC locally so only you can access it. As this product is developed on C# and .NET Classes so it’s more of Windows compatible along with Linux and MacOS support.

Minimum System Requirement

- Processor: Intel Pentium 4 2.0GHz / AMD Athlon XP 2500+
- RAM: 512MB
- OS: Windows XP

- 3D: No
- Microphone: YES
- Installed at least .NET Framework 3.50

2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

This system is provisioned to be built on the .NET Classes which is highly flexible . Decision regarding which database to use should be taken considering the fact that data being stored is not large and stored locally.

The assumed factors that will affect the product is the use of microphone. In any case if the microphone is not in working condition, the whole idea of the project will go in vain as the main reason of using this intelligent ToDo is to just say your task and it will know what time to set and when to remind you. Another constraint that may arise is voice recognition API, we're using voice recognition API to fetch the human natural language and then after applying some algorithms translated into text and then applying machine learning to set the task to be reminded so if the API crashes due to some error or bug, the whole product will be of no use. Also the factor of noise in the environment is also a major issue if the software can not listen properly what the end user is trying to say, it may end up messing up things. The overloaded of tasks is another thing to keep in mind if the user keeps on adding task without finishing the old one so the system will prompt user to constraint for now and complete old tasks.

2.6 USER DOCUMENTATION

There's not such user documentation needed for this product as this product is pretty simple and everyone of every age group can easily hold a grasp on it. Adding the task you wish to work on and provide the deadline is all you want to do for this product to work. The rest will be handled by the ToDo program. All you need to know is what task you should be reminded of and when and the ToDo assured everything for you.

2.7 ASSUMPTIONS AND DEPENDENCIES

It purely depends on the voice recognition API which is doing most of the work and is the base of this system. However, ML algorithms to set tasks efficiently is used later in the system to make this ToDo not any ordinary ToDo software product but an efficient and smarter one. You need to make assumptions in a project to be able to move forward with it. As we've established, planning is never certain and there are many external factors you can't control or anticipate. The dependency on the .NET Framework classes is another thing to take into account which helps us to develop the product without the need of starting and defining each and every thing from scratch. The documentation by Microsoft is also being considered when working with C# and .NET Classes to develop this system.

Chapter 3

SPECIFIC REQUIREMENTS

3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 USER INTERFACES

The first interface available to user will be of home page displaying a button that proceeds further.

For example

The user will then be directed to another screen in which he/she can add tasks through voice commands, schedule them or delete them.

Example:

3.1.2 HARDWARE INTERFACES

The logical characteristics of each interface between the hardware and the user of the system is all handled by C# and .NET Library as it helps the computer in computing the instructions provided by the developer and the compiler processed that instructions and provides information to the computer to perform the following tasks. It plays a big role in human-computer interaction as it reduces one of the main barriers of language between human and computer.

All ToDo system needed is a Windows machine since C# is developed by Microsoft so it provides a great means of ecosystem for the environment.

3.1.3 SOFTWARE INTERFACES

Not a lot has been going on in software interface, all it needs is a Windows machine and MS Visual Studio and all the work is being carried out by the developer. Since this product is mostly used by the users for personal or professional attitude of work, so it doesn't need to have large or distributed database. To keep things simple, we're going with local database because anyone using ToDo system is using it for its own productivity and simplicity and for example: if a company makes it obvious to use ToDo system, Even so everyone will be using it in their personal computer organizing their personal or professional life.

The preferred OS used for this product is Windows just because of the ecosystem although you can also run this on Linux or MacOS without any issue.

3.1.4 COMMUNICATION INTERFACE

The Smart TO-DO project will be locally available. neither the cloud service is involved nor the cross platform support.

3.2 FUNCTIONAL REQUIREMENTS

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish.

1. TASK

In this menu there are many features regarding the task in the to do list. Features in this category are about creating, editing and managing tasks.

(a) NEW TASK

USER REQUIREMENT

User will add a task. User can set a new task either manually or through voice command. User can say set a task “meeting” on “2nd october 2019” at “3:00pm” or Task name, date and time can be set up manually. User can also add reminding option in which he/she will be reminded of the task completion prior to the deadline.

SYSTEM REQUIREMENTS

- i. The user will select the add task option.
- ii. Add task page will be opened consisting of minimum three fields.
- iii. First field will be add task name.
- iv. Second field will be date of task completion.
- v. Third field will be time of task completion.
- vi. When the user presses add, system must check if the “name”, “date” and “time” fields are completed.
- vii. The fields can be typed-in or can be entered through voice command.
- viii. The task will be stored in database.
- ix. If they are not, it should warn the user that the particular field is mandatory, and allow the user to continue completing.

(b) EDIT TASK

USER REQUIREMENTS

The user must select any task for the previously existing tasks and can edit its name, date or day.

SYSTEM REQUIREMENTS

- i. The user will select the edit task option.

- ii. Edit task page will opened.
- iii. User may change name, date or day of the existing task.
- iv. Edited task cannot be saved if the new name of it is empty.
- v. If there is not a task selected before user chooses edit task option, the task cannot be saved.
- vi. The edited information will be updated in the database.

(c) DELETE TASK

USER REQUIREMENTS

In this feature, it is described how the user can delete a task.

SYSTEM REQUIREMENTS

- i. User selects a task Then chooses “Task: Delete task” .
- ii. the system will then ask/confirm if the user wants to delete the selected task.
- iii. If user answers positive, then this task will be deleted, else the delete command will be canceled

2. VIEW

In this menu there are features, which select the view of the tasks.

(a) COMPLETED TASKS

USER REQUIREMENT

in this feature, it is described, how users can show the completed tasks. User chooses “View: Show completed tasks”.

SYSTEM REQUIREMENTS

- i. User chooses the option to view tasks.
- ii. User will select the option of view completed tasks.
- iii. System must show any task that is completed in the task list.
- iv. User must not be able to choose show completed tasks, if these tasks are already shown.

(b) INCOMPLETED TASKS

USER REQUIREMENT

In this feature, users can view uncompleted task.

SYSTEM REQUIREMENTS

- i. User chooses the option to view tasks.
- ii. User will select the option of view incompleted tasks.

- iii. When the user chooses to utilize this feature, system must show any task which is pending.
- iv. If all the tasks are already completed, no taskk will appear in this view option.

3. SMARTLY DIVIDED IN CATEGORIES

- **FUNCTION**

The smart To Do system can smartly defines the category of task and combines all the task of the same category in one place.

- **DESCRIPTION**

This would help the students or businessmen to manage their household and school/office tasks easily. Also, the completed and pending tasks are pre-defined in different categories so you can always take a quick glance on the completed or pending tasks.

- **INPUTS**

the user will give input by selecting any of the available categories.

- **SOURCE OF INPUT**

user will give input in this feature by selecting the category.

- **OUTPUTS**

The output of this feature will be in the form that tasks will be displayed or can be viewed in different categories.

- **ACTION**

as the user selects a particular category for a particular tast, databse will be updated i.e the particular category will be assigned to that task.

- **PRE-CONDITION**

There is no pre-condition to utilize this feature.

- **POST-CONDITION**

after selecting a particular category, user must enter task name, date and time to store it in database.

- **EXCEPTION**

if the task name, date or day is not assigned, then the task will not be updated under the selected category inn database and hence cannot be viewed.

4. ALARM TONE

USER REQUIREMENTS

In this feature, user can select any of the alarm tone provided by the system.

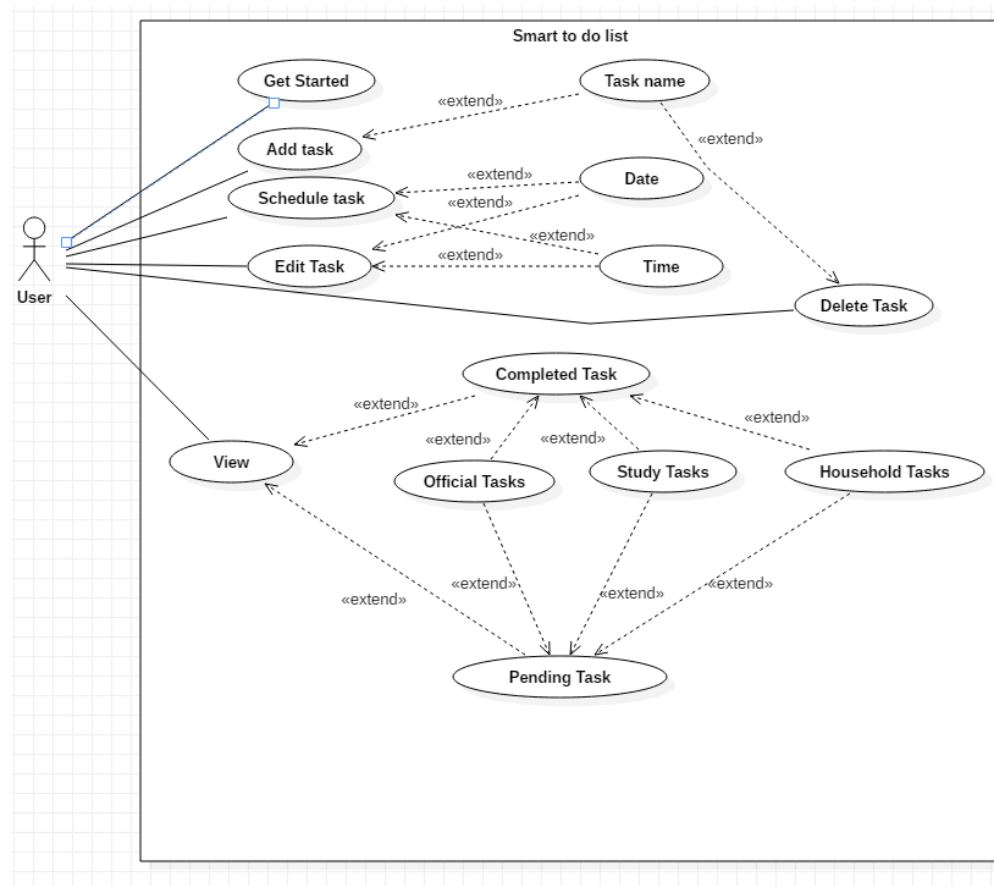
SYSTEM REQUIREMENTS

- (a) User can choose any of the available alarm tone.
- (b) The system should remind the user of the task by ringing the specific tone selected by the user.

3.3 BEHAVIOUR REQUIREMENTS

3.3.1 USE CASE VIEW

Here there's only one actor but we can further specify it like Student, Manager and Teacher. For the sake of simplicity, we're considering User who can add tasks with respective date and time using their voice commands and also edit , view and delete the tasks. Moreover, tasks are then classified into categories depending on the type of tasks.



Chapter 4

OTHER NON-FUNCTIONAL REQUIREMENTS

4.1 PERFORMANCE REQUIREMENTS

Smart to do list is a program with :

- Minimal memory requirements,
- Disk space and
- Processing power.
- However to use the voice command feature, the microphone should be working.
- Any task can be edited provided that it is not completed.
- Any task whether completed or not, can be deleted.

4.2 SAFETY AND SECURITY REQUIREMENTS

The program should be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute.

- To do so, it is safest to each file at least the “copyright” line.
- Also add information on how to contact you by electronic and paper mail.
- Database should be protected by a username and password.

However, .NET framework classes are open to all and can be used by any user.

4.3 SOFTWARE QUALITY ATTRIBUTES

4.3.1 EFFICIENCY

The program aims to provide the user ease and functionality without any charge. It lets you keep track of everything in one place, so you can get it all done and

enjoy more peace of mind along the way.

4.3.2 USABILITY

Our software is user friendly. It can be used for the professional employees as well as by the unemployed people and students. It can be used in diverse ways.

4.3.3 ECOSYSTEM

This software provides a great means of ecosystem. By developing this product on native platforms, you can work on cross platforms. You can update, delete and edit tasks from your Windows PC or mobile phones and even from car multimedia control system but our main focus is towards windows platform so we only implement that. We can further make it work on cross platforms across different OS.

4.3.4 OPERATIONAL

The smart to-do list also offers an essential non functional requirement highlighting the operation of the software. The software will only be operational when a user is registered. No guest accounts are allowed.

Chapter 5

DATABASE REQUIREMENTS

Keeping in mind the database is saved locally, it's not a large or distributed database. Tables and entities are divided in terms of categories like Pending tasks and Completed Tasks and further subdivided into Household chores, Official Work etc. The preferred database system we're using is sql since it's minimal and handy to use for such a small database. Moreover, no specific or hardcore server is required for this purpose as it's consuming very little memory and can easily fit in any PC.

Index

- .NET Framework, 5, 8, 14
- Add Tasks Support, 10
- Alarm Support, 12
- API, 8
- Constraints, 8
- Cross Platform Support, 10
- Database, 16
- Database Requirements, 16
- Delete Tasks Support, 11
- Document Scope, 3, 8
- e.g, 7
- Edit Task Support, 10
- End Users, 7, 8, 12
- Functional Requirements, 10
- Functionalities, 7
- Hardware Interfaces, 9
- Intended Audience, 4, 7, 8
- ML algorithms, 8
- Operating System, 7
- Overview, 4
- Product Perspective, 6
- Project Scope, 4
- references, 5
- Safety, 14
- schema, 5
- Security, 14
- Software Interfaces, 9
- SQA, 14
- SQL, 5, 16
- System Requirements, 7, 9, 10, 14
- Use Case View, 13
- User Interfaces, 9