

To-Do List

(SRS)

Submitter by: Bharathwaj S

Roll No: 201016

Members: Surthishwaran (201015)

Vignesh (202008)

Sasikumar (202006)

Team Guide: Ashraf



College Name: Rajiv Gandhi College of Engineering

Department: CSE

To-Do List

1. Introduction

To-Do List is an open source to-do manager specially designed to help users manage and regulate their day to day task by scheduling in advanced. It is available in English. This project has been developed in PHP.

1.1 Purpose

This is a Software Requirements Specification (SRS) document for To-Do list. Its purpose is to describe functional requirements, features and other important requirements for this system's function.

1.2 Document Convention

To-Do list is a program which will soon get released. Therefore, the SRS document intends to make clear implementation features of the software, and to write down requirements for future extensions. This document may be used as a guide for those who want to understand the features of the program and for those who want to develop future components of the program.

1.3 Intended Audience and Reading suggestions

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 To Do list, gives to them.






1.4 Project Scope

To-Do list is a practical day to day organizing tool for all type of users. It can be used as a to-do manager, for common users or business oriented users, as it has the capability to work as a plain manager, in which user's tasks can be saved and organized in hierarchical form, or as a professional to-do manager, which offers extra features like storing payments details, saving multiple comments for each task and timers, that compute the time spent on each task. Beside the hierarchical organization of the tasks, user can organize his tasks in different categories, which he previously can create. Also, tasks can be sorted according to their attributes, like creation time, priority, due date, etc. To-Do List is easy to use program and is available in English language. It does automatic backups and it is fully portable (it uses MySQL).

1.5 References

The official To-Do List webpage we created in local hosting environment, including downloads of the latest version of the program, change log of the latest version, the source code and available contacts of the developers (imbharathwaj@gmail.com)

To-do list using PHP and MySQLi

#	Task	Status	Action
1	This is my new task	Completed	
2	This is my second task	Pending	 
3	This is my third task	Pending	 

This project is hosted in local environment, where source code of latest and older versions and some screenshots of the program are available. There is also a forum, so users can report bugs, propose new features of the program or any improvements.

2. Overall Description

2.1 Product Perspective

To-Do List is a free software program suitable for all users interested in organizing their tasks and can be run on any web browser. It is free and open source program, which means that it is suitable both for those, who are interested in offering its development by expanding its functions and features, enriching the code, and for the most demanding users who can adapt it to their own personal needs by modifying the existing source code. Additionally, for whatever problems that may arise.

2.2 Software Requirement

- Front end:
 - ◆ HTML
 - ◆ CSS
 - ◆ JavaScript
- Back end:
 - ◆ PHP
 - ◆ MySQL

2.3 Hardware Requirement

- Web Browser
- 2.5GHz Processor
- 4GB RAM
- Intel i3
- Unix/Linux or Windows Operating System

2.4 Functional Requirement

2.4.1 New Task

Description: In this feature, it is described the procedure of creating a new task, completing task's fields and adding the task in an existing category.

The user: User has to choose "Task: New task" and then a pop up window shows up, with the empty fields of the task and the category, in which the task will be added in. If there is not already a category, user should create a new one by pressing new category. User must definitely complete the field with the name of the task, as it is mandatory. User can choose the color, in which the task will be shown, complete the description, set the priority of the task, define the due date, set the task as completed, set the task timer, reset the task timer, sum the subtasks' timers, edit the payment details, add a note.

The system: System must show up a pop-up window, when the user chooses "Task: New task". When the user presses add, system must check if the "name" field is completed. If it is not, it should warn the user that this field is mandatory, with a new pop up box, and allow the user to continue completing in the first pop up window. also, system, must not allow the user to add a new task if there is not an available category. Finally, the system must save the new task and its attributes, and then add it nested in the category.

Constraints: A new task cannot be added if the "name" field is not completed and if there is not a category to add the new task.

2.4.2 Edit Task

Description: In this feature, it is explained the procedure of editing an existing task and its fields.

The user: Firstly, user must select an existing task by left clicking on it and then choose "Task: Edit task". Then a pop-up window will show up, same one as

the new task's, in which the user can edit task's fields. User must not delete the name of the task and leave it empty, as it is mandatory. User can also change the color, in which the task will be shown, edit the description, reset the priority of the task, redefine the due date, set the task as completed, edit the task timer, reset the task timer, sum the subtasks' timers, edit the payment details, add, edit and delete a note.

The system: System must show up a pop-up window, when the user chooses "Task: Edit task". When the user presses edit, system must check if the "name" field is completed. If it is not, it should warn the user that this field is mandatory, with a new pop up box, and allow the user to continue completing in the first pop up window. If user presses sum subtasks, system must sum the timers of each subtask of the current task and return the result in the current task's timer. also, system, must not allow the user to be able to press "Task: Edit task", if a task has not been already selected. Finally, the system must save the edited task and its edited attributes, and move it in a new category or parent task, if the user has changed the task's location.

Constraints: Edited task cannot be saved if the new name of it is empty and if there is not a task selected before user chooses edit task.

2.4.3 Delete Task

Description: In this feature, it is described how the user can delete a task or a subtask.

The user: User selects a task or a subtask. Then chooses "Task: Delete task" and a dialog box shows up, asking if he wants to delete the selected task. If user answers positive, then this task and any subtask in it, will be deleted, else the delete command will be canceled. **The system:** When user chooses "Task: Delete

task”, system shows up a dialog box asking the user, if he wants to delete the selected item. If user answers positive, system deletes the selected task, and any other subtask there is in it, else system cancels this action.

Constraints: There must be a selected task before user chooses delete task.

2.4.4 Mark Task as Completed

Description: In this feature, it is described the procedure of marking a task as completed. Every task must have a checkbox on its left, to make it clear if the task is completed or not.

The user: User selects a task and then chooses “Task: Mark task completed”. After completing this action, selected task’s checkbox should be checked. Alternatively, user can straight check the checkbox by himself.

The system: When user chooses “Task: Mark task completed”, the system must check the checkbox of the selected task and save this task as completed.

Constraints: There must be a selected task, which is uncompleted, before user chooses mark task completed.

2.4.5 Mark task Uncompleted

Description: In this feature, it is described the procedure of marking a task as uncompleted. Every task must have a checkbox on its left, to make it clear if the task is completed or not.

The user: User selects a task and then chooses “Task: Mark task uncompleted”. After completing this action, selected task’s checkbox should be unchecked. Alternatively, user can straight uncheck the checkbox by himself.

The system: When user chooses “Task: Mark task uncompleted”, the system must uncheck the checkbox of the selected task and save this task as uncompleted.

Constraints: There must be a selected task, which is completed, before user chooses mark task uncompleted.

2.5 Non-Functional Requirements

Usability Requirement

The system shall allow the users to access the system from the phone using web application. The system uses a web browser application as an interface. Since all users are familiar with the general usage of mobile app, no special training is required. The system is user friendly which makes the system easy.

Availability Requirement

The system is available 100% for the user and is used 24hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

Efficiency Requirement

Mean Time to Repair (MTTR) - Even if the system fails, the system will be recovered back up within an hour or less.

Accuracy

The system should accurately provide real time information taking into consideration various concurrency issues. The system shall provide 100% access reliability.

Performance Requirement

The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.

Reliability Requirement

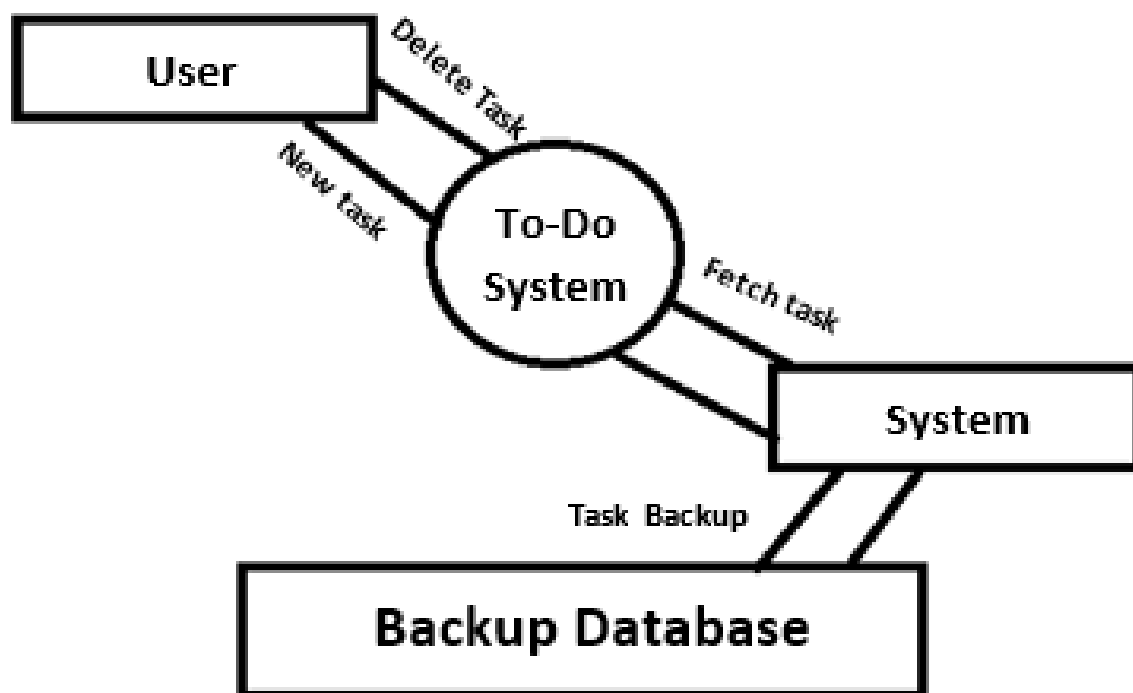
The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week, 24 hours a day.

2.6 User Characteristics

Here we have 2 levels of users:

- User module: In this module, User will manage their tasks
 - Create new tasks
 - Edit existing tasks
 - Delete the task created
 - Mark tasks as completed
 - Mark task as uncompleted
- System module: In this module, System will take backup
 - Check for task updates
 - Take backup is tasks updated
 - Will not take backup if tasks are not updated

2.7 Flow Diagram



2.8 Use Case Diagram

Todo List Use Case diagram

