Software Requirements Specification

for

Todomoo

Requirements for Version 0.8

Prepared by Panagiotis Melidis

10/9/2011

Table of Contents

1. Introduction

Todomoo is an open source to-do manager specially designed to help users manage paid projects without advanced scheduling. It is available in English, Italian, French and Portuguese. The project has been developed in C#.

1.1 Purpose

This is a Software Requirements Specification (SRS) document for Todomoo (version 0.8). Its purpose is to describe functional requirements, features and other important requirements for this system's function.

1.2 Document Conventions

Todomoo is a program which will soon get released. Therefore, the SRS document intends to make clear already implemented 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 existing 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 Todomoo, gives to them.

1.4 Project Scope

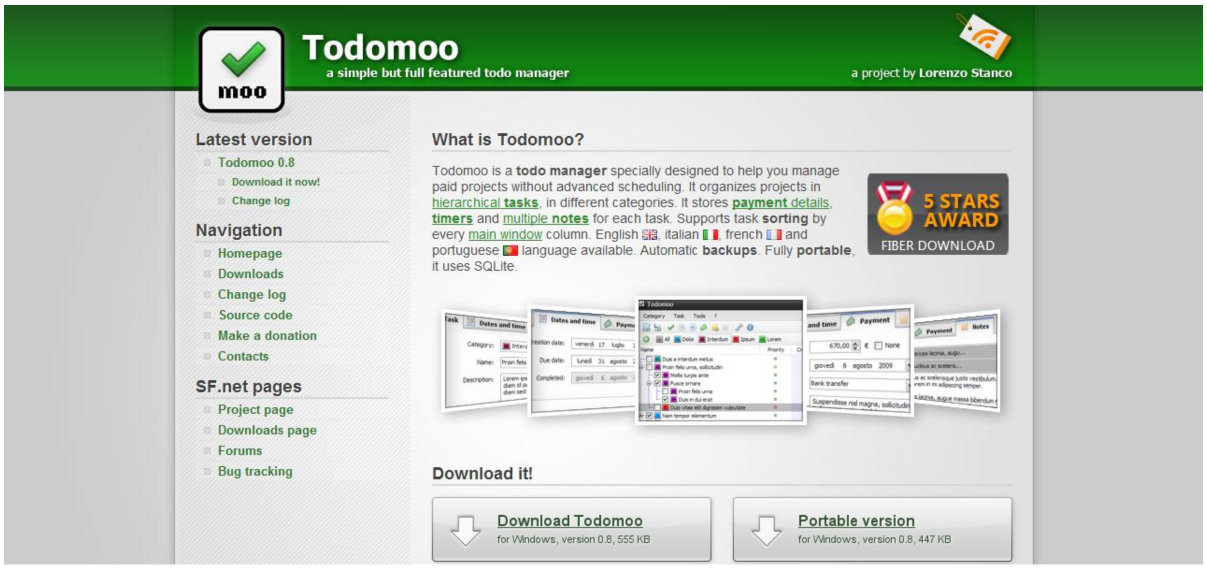
Todomoo is a practical organizing tool for all type of users. It can be used as a to-do manager, for common users or business oriented users, as it has the capability to work as a plain manager, in which user's tasks can be saved and organized in hierarchical form, or as a professional to-do manager, 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.

Todomoo is easy to use program and is available in four different languages (English, Italian, French, and Portuguese). It does automatic backups and it is fully portable (it uses SQLite).

1.5 References

The official Todomoo webpage is at http://todomoo.sourceforge.net/, including downloads of the latest version of the program, change log of the latest version, the source code and available contacts of the developers (lorenzo.stanco@gmail.com).



Todomoo
a simple but full featured todo manager
a project by **Lorenzo Stanco**

Latest version
- Todomoo 0.8
  - Download it now!
  - Change log

Navigation
- Homepage
- Downloads
- Change log
- Source code
- Make a donation
- Contacts

SF.net pages
- Project page
- Downloads page
- Forums
- Bug tracking

What is Todomoo?

Todomoo is a **todo manager** specially designed to help you manage paid projects without advanced scheduling. It organizes projects in hierarchical **tasks**, in different categories. It stores **payment** details, **timers** and multiple **notes** for each task. Supports task **sorting** by every main window column. English, italian, french and portuguese language available. Automatic **backups**. Fully **portable**, it uses SQLite.

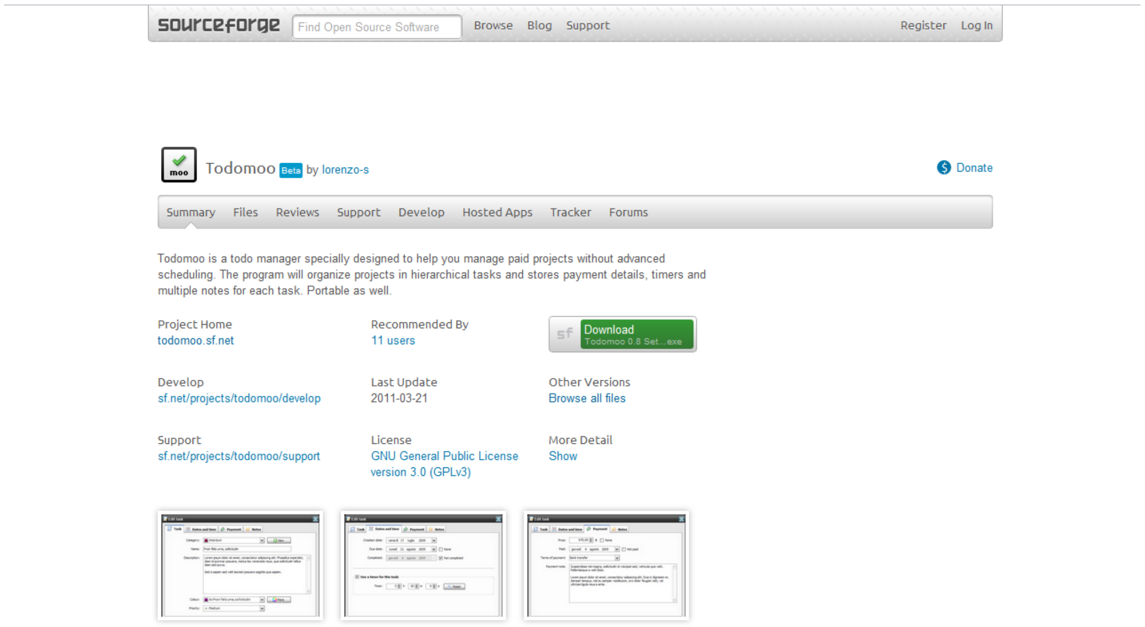
5 STARS AWARD
FIBER DOWNLOAD

Download it!

Download Todomoo
for Windows, version 0.8, 555 KB

Portable version
for Windows, version 0.8, 447 KB

This open source project is hosted by sourceforge, at: http://sourceforge.net/projects/todomoo/, 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.



sourceforge | Find Open Source Software | Browse | Blog | Support | Register | Log In

Todomoo Beta by lorenzo-s
Donate

Summary | Files | Reviews | Support | Develop | Hosted Apps | Tracker | Forums

Todomoo is a todo manager specially designed to help you manage paid projects without advanced scheduling. The program will organize projects in hierarchical tasks and stores payment details, timers and multiple notes for each task. Portable as well.

Project Home
todomoo.sf.net

Recommended By
11 users

Download
Todomoo 0.8 Set...exe

Develop
sf.net/projects/todomoo/develop

Last Update
2011-03-21

Other Versions
Browse all files

Support
sf.net/projects/todomoo/support

License
GNU General Public License version 3.0 (GPLv3)

More Detail
Show

2. Overall Description

2.1 Product Perspective

Todomoo is a free software program suitable for all users interested in organizing their tasks and can be installed on any operating system. 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, users can be addressed to the forum in order to resolve any questions or problems.

2.2 Product Features

Some of the main features of Todomoo are:

- Creating new tasks and subtasks.
- Editing existing tasks and their attributes, like notes, creation time, priority, due date, payment details.
- Creating new categories of tasks.
- Editing existing categories.
- Delete tasks and categories.
- Move tasks between categories.
- Sorting tasks by their attributes.
- Manage the view of the tasks (expanded, collapsed).
- Start and pause a task timer.
- Mark a task as completed.
- Export all tasks in a CSV file, or in a database (SQlite).
- Automatic backup.

2.3 User Classes and Characteristics

The program is built on two basic categories of users:

- Ordinary users can use Todomoo for specific needs. These users are the main beneficiaries of Todomoo and are not required to have a specialized instructional and academic background. In addition, users can be of any age. What is needed is these users have some basic knowledge of computer use without requiring special technical skills and experience.
- Developers, testers and all those who participate and contribute in their way on open source groups. These users can be of any age but must have enough specific knowledge of computer usage. Especially in cases where they wish to contribute to the development of the program by writing code and implementing new functions, is essential to have important knowledge of programming language and experience. This, of course, is not necessary, where users want to contribute to the community with constructive comments and ideas.

2.4 Operating Environment

Todomoo is a fully portable program without special requirements, and it can be used on any Unix-like operating system and Windows.

2.5 Design and Implementation Constraints

As the licensed software is GPL (General Public License), this means that this project is 100% open source and 100% free. Anyone interested in contributing in whatever way, he wants in its development, it is necessary to accept and agree to the terms and conditions of this license as it is the only limitation.

2.6 User Documentation

The elements of this software are available through the official site at source forge. However, this documentation is superficial. For more information, the user may approach the members of the community and more specifically the project manager or by sending e-mail or write to the forum's project through the source forge. Finally, this document is intended to be an indirect documentation from now on.

2.7 Assumptions and Dependencies

There are no conditions for using the Todomoo beyond the existence of a graphics card and a graphical interface to use the program.

3. System Features

In this section system features are described analytically. Both features, that have been already implemented and features, which have been proposed for future implementation, are described.

3.1 Task

In this menu there are many features that must show up when user right clicks on a task. Features in this category are about creating, editing and managing tasks.

3.1.1 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.

3.1.2 New subtask of selected one

Description: In this feature, it is described the creation of a new subtask, which will be added in an existing task, and the completion of its fields.

The user: Firstly, user must select an existing task by left clicking on it and then choose “Task: New subtask of selected one”. Then a pop up window will show up, almost same one as the new task’s, in which the user can complete the subtask’s fields. User must complete the name of the task, as it is mandatory, and can also 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 subtask of selected one”. This pop up window is almost the same as the new task’s pop up window, with the only difference, that it has not the category field. 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. 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: New subtask of selected one”, if a task has not been already selected. Finally, the system must save the new task and its attributes, and then add it nested in the parent task.

Constraints: A new task cannot be added if the “name” field is not completed and if there is not a task selected, so that the new subtask can be put nested in it.

3.1.3 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.

3.1.4 Move task

Description: In this feature, it is described how the user can move a task or a subtask, from a category or a parent task to other categories or parent tasks.

The user: First of all, user selects a task or a subtask. Then chooses “Task: Move task” and a new pointer shows up. User left clicks with the new pointer on a category or a task that he wishes the selected task to move in.

The system: When the user chooses “Task: Move task”, the system changes the pointer of the mouse to show to the user that he is going to move a task. As soon as, the user has left clicked on a category or a parent task, the system must move the selected task from its former location to the new location user has chosen. System must save the new location of the selected task.

Constraints: In order to move a task, there must be at least one different category or task to move to and a task must be selected before user chooses move task.

3.1.5 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.

3.1.6 Mark task 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.

3.1.7 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.

3.1.8 Start task timer

Description: This feature refers to how starting a timer of a task works.

The user: User selects a task and then chooses “Task: Start task timer”. After completing this action, selected task’s timer will start running.

The system: When user chooses “Task: Start task timer”, the system must start running the selected task’s timer. If the task timer has already run before, system must continue from the last saved timer of this task, else it starts from 00:00:00 (hh:mm:ss format).

Constraints: There must be a selected task, before user chooses start task timer.

3.1.9 Pause task time

Description: In this feature, it is described how a task’s timer can get paused.

The user: User selects a task and then chooses “Task: Pause task timer”. After completing this action, selected task’s timer will be paused.

The system: When user chooses "Task: Pause task timer", the system must pause the selected task's timer.

Constraints: There must be a selected task, before user chooses pause task timer.

3.1.10 Edit payment details

Description: In this feature, it is described how users can edit the payment details of a task.

The user: User selects a task and then chooses "Task: Edit payment details". After this, a pop up window shows up with the payment fields of this task, where user can set the payment per hour of this task, or payment for completing this task, check if the worker got paid or not, set the terms of payment, or add a payment note. Finally, user can sum the costs of all subtasks, of the current task to see the total cost.

The system: When user chooses "Task: Edit payment details", the system must show up a pop up window with the payment details. When user completes the form and presses edit, the system must save the new payment details. In case, user presses sum subtasks, system must sum the payment of each subtask of the current task, and return the result in the payment of the current task.

Constraints: There must be a selected task, before user chooses edit payment details.

3.1.11 New note

Description: In this feature, it is described how users can create a new note.

The user: User selects a task and then chooses "Task: New note". After this, a pop up window shows up with a note form, where user can add a new note. User can create as many notes as he wants and also delete already existing notes.

The system: When user chooses "Task: New note", the system must show up a pop up window with a form to add new notes. When user chooses add, system must save the new note and when user chooses delete note, system must delete the selected note.

Constraints: There must be a selected task, before user chooses new note. Also there must be a selected note, in order user to be able to delete a note.

3.1.12 View notes

Description: In this feature, it is described how users can view a task's notes.

The user: User selects a task and then chooses "Task: View notes". After this, a pop up window shows up with the available notes, where user can also add and delete a note.

The system: When user chooses “Task: View notes”, the system must show up a pop up window with the available notes. When user chooses add, system must save the new note and when user chooses delete note, system must delete the selected note.

Constraints: There must be a selected task, before user chooses View notes and also the selected task must have at least one note. Also there must be a selected note, in order user to be able to delete a note.

3.2 Category

In this menu there are many features that must show up when user right clicks on a category. Features in this category are about creating, editing and deleting categories.

3.2.1 New category

Description: In this feature, it is described how users can create a new category.

The user: User has to choose “Category: New category” and then a pop up window shows up, with a form, in which user must complete the category name and choose the color of that category. After completing the fields, user must press add.

The system: System must show up a pop up window, when the user chooses “Category: New category”. 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. Finally, the system must save the new category and its attributes.

Constraints: A new category cannot be added if the “name” field is not completed.

3.2.2 Edit current category

Description: In this feature, it is described how users can edit an existing category.

The user: Firstly, user must select an existing category by left clicking on it and then choose “Category: Edit category”. Then a pop up window will show up, same one as the new category’s, in which the user can edit the category’s fields. User must not delete the name of the category and leave it empty, as it is mandatory. User can also change the color, in which the category will be shown.

The system: System must show up a pop up window, when the user chooses “Category: Edit category”. 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. Also system, must not allow the user to be able to press “Category: Edit category”, if a category has not been already selected. Finally, the system must save the edited category and its attributes.

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

3.2.3 Delete current category

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

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

The system: When user chooses "Category: Delete category", 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 category, and any other task there is in it, else system cancels this action.

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

3.3 View

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

3.3.1 Flat view

Description: In flat view, every task and subtask of each category, show up in the same list allowing the user to handle them together. For example, instead of only sorting the top tasks, user can also sort the subtasks at the same time.

The user: User chooses "View: Flat view".

The system: When user chooses "View: Flat view", system must unravel the nested subtasks, and show every task and their subtasks in the same list. When user tries to sort the tasks by any attribute, system must also sort the whole list, including the subtasks.

Constraints: There are not any constraints.

3.3.2 Expand all

Description: In this feature, every task, which has at least one subtask, expands and shows its nested subtasks.

The user: User chooses "View: Expand all".

The system: When user chooses "View: Expand all", system must show the nested subtasks of each task. However, if user tries to sort the tasks by any attribute, system will only sort the top tasks.

Constraints: There are not any constraints.

3.3.3 Collapse all

Description: In this feature, every expanded task, which has at least one subtask, must hide its own subtasks. It is the exact opposite function of Expand all.

The user: User chooses “View: Collapse all”.

The system: When user chooses “View: Collapse all”, system must hide the nested subtasks of each task.

Constraints: There are not any constraints.

3.3.4 Minimize to tray

Description: In this feature, it is described how user can minimize Todomoo, into a system tray on the operating system’s taskbar.

The user: User chooses “View: Minimize to tray”.

The system: When user chooses “View: Minimize to tray”, system must hide the open window of Todomoo and create a small tray icon on the operating system’s taskbar.

Constraints: There are not any constraints.

3.3.5 Show completed tasks

Description: In this feature, it is described, how users can show the completed tasks.

The user: User chooses “View: Show completed tasks”.

The system: When user chooses “View: Show completed tasks”, system must show any task that is completed in the task list. User must not be able to choose show completed tasks, if these tasks are already shown.

Constraints: Completed tasks must be hidden, so that user can choose show completed tasks.

3.3.6 Hide completed tasks

Description: In this feature, it is described, how users can hide the completed tasks.

The user: User chooses “View: Hide completed tasks”.

The system: When user chooses “View: Hide completed tasks”, system must hide any task that is completed from task list. User must not be able to choose hide completed tasks, if these tasks are already hidden.

Constraints: Completed tasks must be shown, so that user can choose hide completed tasks.

3.3.7 Exit

Description: In this feature, it is described, how users can exit the program.

The user: User chooses “View: Exit”.

The system: When user chooses “View: Exit”, system must save anything has not being saved yet and then terminate the program.

Constraints: There are not any constraints.

3.4 Tools

In this menu there are features, which allow users to save their tasks into different file formats, like .csv or .sqlite, and manage the configurations of Todomoo.

3.4.1 Export all tasks to CSV

Description: In this feature, it is described how users can export all tasks to a CSV (Comma Separated Values) file.

The user: User has to choose “Tools: Export all tasks to CSV”. Then a pop up window will show up and user must choose the location and the name of the file. After that user must press save.

The system: When user chooses “Tools: Export all tasks to CSV”, system must open a window, where the user can complete the name of the file and choose the location, in which the file will be saved. If user has not completed the name, system must not allow him to save the file, because the name field is mandatory.

Constraints: User must complete the name of the file, so that the system can create the file.

3.4.2 Backup the database

Description: In this feature, it is described how users can backup their tasks in a .sqlite file.

The user: User has to choose “Tools: Backup the database”.

The system: When user chooses “Tools: Backup the database”, system must save all tasks in a default sqlite database.

Constraints: There are not any constraints.

3.4.3 Backup the database in

Description: In this feature, it is described how users can backup their tasks in a .sqlite file and choose the name of it, and its location.

The user: User has to choose "Tools: Backup the database in". Then a pop up window will show up and user must choose the location and the name of the file. After that user must press save.

The system: When user chooses "Tools: Backup the database in", system must open a new window, where user can choose the location of the sqlite database and complete its name. When user presses save, system must save all tasks with the right name in the location user chose. If user hasn't completed the name, system must not allow him to save the file, because the name field is mandatory.

Constraints: User must complete the name of the file, so that the system can create the file.

3.4.4 Options

Description: In this feature, it is described how users can configure some options of the program.

The user: User has to choose "Tools: Options". Then user can configure the appearance of the menu, the toolbar and the category-bar, the language of the program, the currency in which the payments are calculated, the behavior of the program when the program is closed, or options like, whether to backup the tasks every time the program is terminated or not. When user finishes, he must press ok to save the configurations.

The system: When user presses "Tools: Options", system must show up a window with the configuration options to the user. When user presses ok, system must save all the new changes, that user made in the configurations.

Constraints: There are not any constraints.

3.5 Help

In this menu there are functions, to help users learn more about Todomoo and check for available updates.

3.5.1 Check for updates

Description: In this feature, users can check for available updates.

The user: User must choose “Help: Check for updates”. Then a pop up window will show up and inform the user if he uses the latest version of Todomoo, or if he must update the program to the latest version.

The system: When user chooses “Help: Check for updates”, system must show up a window to the user to inform him about the version of his program and if there is any update. System must connect first to Internet in order to know if there is an available update.

Constraints: There must be an Internet connection.

3.5.2 Visit Todomoo Website

Description: In this feature, users can visit the official Todomoo Website.

The user: User must choose “Help: Visit Todomoo Website”.

The system: When user chooses “Help: Visit Todomoo Website”, system must open a browser with the URL of Todomoo Website, if there is an available Internet connection.

Constraints: There must be an Internet connection.

3.5.3 Visit project page on SourceForge.net

Description: In this feature, users can visit the project page of Todomoo on SourceForge.net.

The user: User must choose “Help: Visit project page on SourceForge.net”.

The system: When user chooses “Help: Visit project page on SourceForge.net”, system must open a browser with the URL of Todomoo’s project page, if there is an available Internet connection.

Constraints: There must be an Internet connection.

3.5.4 About Todomoo

Description: In this feature, users can be informed about the license and the creators of Todomoo.

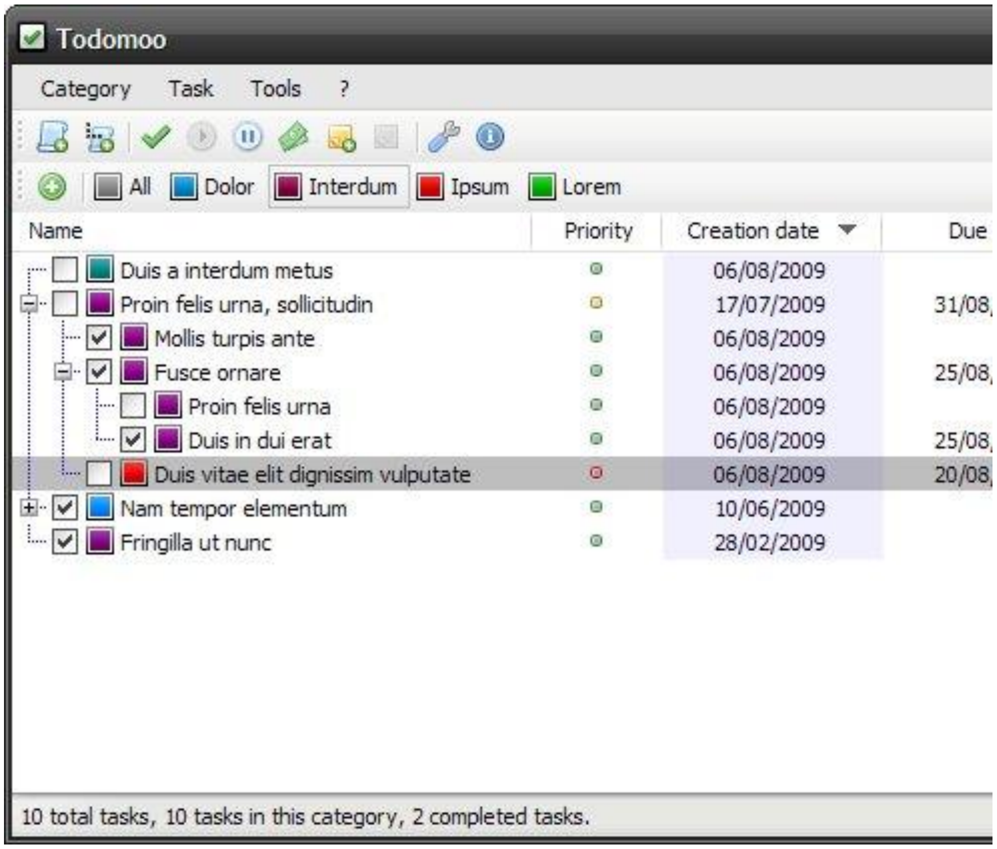
The user: User must choose “Help: About Todomoo”.

The system: When user chooses “Help: About Todomoo”, system must show a pop up window with information about Todomoo’s license, creators, testers, translators, etc.

Constraints: There are not any constraints.

4. External Interface Requirements

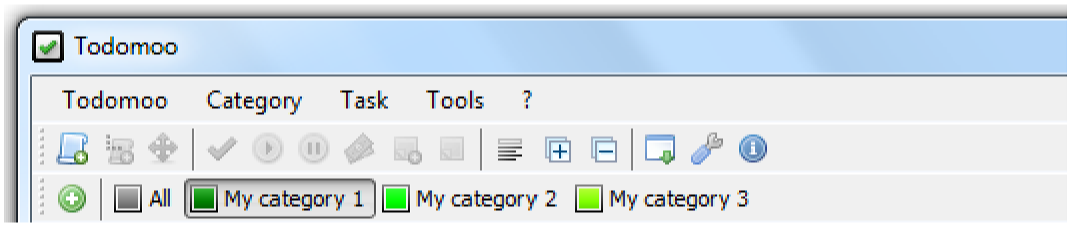
4.1 User Interfaces



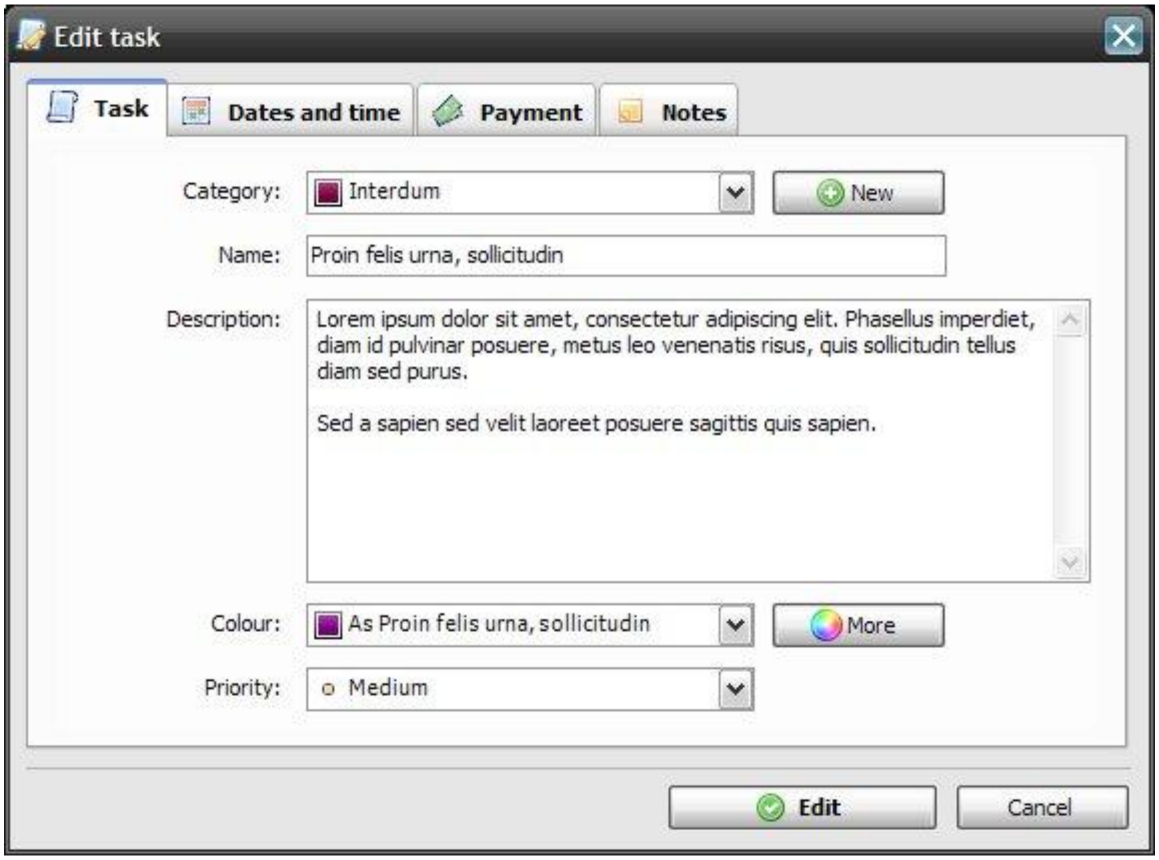
Name	Priority	Creation date	Due
☐ Duis a interdum metus	●	06/08/2009	
☐ Proin felis urna, sollicitudin	●	17/07/2009	31/08,
✓ Mollis turpis ante	●	06/08/2009	
✓ Fusce ornare	●	06/08/2009	25/08,
☐ Proin felis urna	●	06/08/2009	
✓ Duis in dui erat	●	06/08/2009	25/08,
☐ Duis vitae elit dignissim vulputate	●	06/08/2009	20/08,
✓ Nam tempor elementum	●	10/06/2009	
✓ Fringilla ut nunc	●	28/02/2009	

10 total tasks, 10 tasks in this category, 2 completed tasks.

Main screen of Todomoo



Menu Bar of Todomoo



Category: Interdum

Name: Proin felis urna, sollicitudin

Description: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, diam id pulvinar posuere, metus leo venenatis risus, quis sollicitudin tellus diam sed purus.

Sed a sapien sed velit laoreet posuere sagittis quis sapien.

Colour: As Proin felis urna, sollicitudin

Priority: Medium

Editing a task

4.2 Hardware Interfaces

Todomoo works without communicating at all, with any external hardware device. Only a network interface card is required, so that Todomoo can connect to Internet.

4.3 Software Interfaces

In order to work properly, Todomoo uses sqlite, which is a software library that implements an SQL database engine, so that user can save his tasks and backup them in SQL databases.

4.4 Communications Interfaces

Todomoo needs to have connection with an internet provider, for some of its functions.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Todomoo is a program with minimal memory requirements, disk space and processing power. The program aims to provide the user ease and functionality without any charge.

5.2 Safety and Security Requirements

Todomoo's safety and security requirements are specified by GNU GPL license: http://www.gnu.org/licenses/gpl.html.

5.3 Software Quality Attributes

Todomoo is mainly described by the ease of use, the handy organization tools, the portability and the practical interface.