

Tascked

The Sanity Promoting Task Manager

By Jake Tobin

California Polytechnic State University

Advisor Dr. Philip Nico

December 6, 2013

Abstract

Personal task managers or various forms of to-do lists are abundant in our modern computing age. With the explosion of mobile computing technology, it is easier than ever to take notes digitally and make the data seemingly instantly available anywhere on the Internet. There is a fairly well defined core set of features in personal task managers available for public consumption, but it seems nothing that is publicly available provides feedback to the user or suggestions based on user history. Tascked is a task management solution, which records user history and solicits user feedback on progress. This allows the system to generate trend reports for the users consideration and provide “what’s next” suggestions to giving users rapid access to productive tasks they can take on at any given time. Tascked.com is the work in progress implementation of the design outlined in this document.

Contents

1	Introduction	4
2	Background	4
2.1	Producteev	5
2.1.1	Features	5
2.1.2	Limitations	6
2.2	Version One	6
2.2.1	Features	6
2.2.2	Limitations	7
2.3	Google Tasks	7
2.3.1	Features	7
2.3.2	Limitations	7
2.4	Reflections	8
3	Core Design Principles	8
3.1	Problem Areas	8
3.2	Design Principles	9
4	System Design and Implementation	9
4.1	User Accounts	9
4.2	Adding Tasks	11
4.2.1	Task Usage History	11
4.2.2	Adding a Task User Interface	11
4.3	Managing Tasks	14
4.3.1	Individual Task UI	14
4.3.2	Task Recommendation - "What Next"	14
4.3.3	Sleeping Tasks - Allowing Users To Hide Results	17
4.3.4	Filtering Tasks	17
4.3.5	Editing Tasks	18
4.3.6	Completing Tasks and Tracking Time	19
4.4	User Reports	20
5	Implementation Notes	21
6	Future Work	22
6.1	Task Suggestion Improvements	22
6.2	Mobile Development	22
6.3	Task Timer Improvements	23
6.4	User Interface Overhaul	23
6.5	Data Liberation	23

6.6 Exploring Caching, No-SQL and other Database Solutions	23
7 Conclusion	24
Appendix A Full Page Mock Up “Give Me” Filter Tool	26
Appendix B Full Page Mock Up “What Next” Task List	27

1 Introduction

Given modern responsibilities, it is improbable that anyone gets along without a to-do list or task manager of some type. It might be written down in your daily planner, your calendar, a post-it note, or simply in your head. A growing group of people also find themselves using their phone as a digital post-it note or utilizing services to create and manages lists that are online or on their local computers. The recent and immense boom in smart phones, tablets, and all mobile computers has created the opportunity to digitize many aspects of our lives, including our to-do lists. Unfortunately, both in my experience and the casual consensus from conversations with friends, colleagues, and peers, many of the offerings fail to retain users and surpass simple paper solutions. Calendars have been quite successfully migrated to a digital realm, but personal task management faces different challenges.

A key instigator for choosing this project was a personal experience in trying to find a task solution that meshed with my work style. After years of trial and error, I discovered a service which I particularly enjoyed using, one which I could work into my day-to-day life with with minimal overhead. This was Producteev.com. Producteev was pleasantly simple to use, allowing inline text parsing to get things like due dates, categories, and task priority. You could simply view a list of all your tasks and sort it by deadline, category, priority, date created, among other metrics. Unfortunately, Producteev was purchased by another company, Jive. While the acquisition was not bad from the outset, after this merge was completed, they announced an updated version of the service would be released. When the update arrived, I was disappointed to find the tool that I had relied on was quite broken. Aside from a painful multi-day user migration, the process of adding a task was made more difficult, the inline processing commands were made more complex, the display of the tasks no longer could be defaulted to sorting by deadline and instead only listed items by last modified date. Despite a week or two of trying to adjust, the inability to easily access the features I became accustomed to, which I had sought out in other task managers, lead me to abandon it and be manager-less.

In the months since, Producteev has improved to fix many of the glaring problems (except, perplexingly, the ability to set a default sorting order consistently). Nevertheless, the forced migration from their service caused me to seriously reconsider what I expected from a task management solution, what features I had been using, and what features I wish I had access to. With this dream in mind, I decided to enter the world of productivity tool development.

2 Background

The long and varied history I have with task managers is not well suited to be explored in-depth for the purpose of this project; however to better justify my design choices in

Tasked, we will first be examining three very different styles of task management as implemented in Producteev, Version One, and Google Tasks.

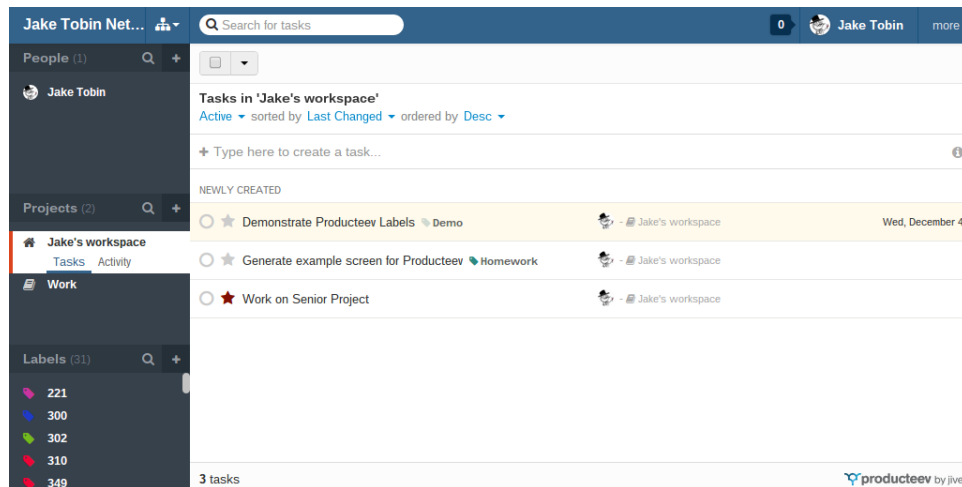


Figure 1: Producteev Task Window

2.1 Producteev

As a full featured team task manager, Producteev has an emphasis on storing data to be shared and categorizing tasks in workspaces and with labels[2, 3, 1].

2.1.1 Features

Although the interface is not as simple or minimal as you can get, it is pretty easy to use and provides significant functionality on top of basic list keeping. There is a balance between showing all of the information a user might need and ensuring users are not buried in data, options, and interfaces. For this most part, this is achieved with Producteev. It also has a very simple, easy to use category system in the form of labels. The user can create new labels right from the add task tool and attach these labels with ease to a task on creation or after the fact. You can then quickly view all the tasks with a given label.

Not only can you quickly attach labels on creation, but the whole task creation process is very well executed. Labels, other users, deadlines, and task priority can all be added with a keyboard shortcut character, so you almost never need to leave the task creation text box.

When using Producteev, users get some feedback on the form of an activity feed. This is most useful in the team environment where users can see what fellow team members are working on at any given time, but they can also review, to some extent, their won task

history. This and almost all other features for Producteev are available on most smart-phones and as desktop applications in addition to the web client, making it difficult to find an excuse not to access your tasks.

Did I mention it is free?

2.1.2 Limitations

As I mentioned before, the product has improved significantly since the shock of the update caused me to jump ship and build my own, but there are improvements which I would wish to see in the system.

One of the seemingly minor issues which contributed to my decision to leave is the unexpected default sorting behavior. This is tied in to a larger limitation of the way the system handles large collections of tasks. By default, tasks were sorted by when they were last changed and shown in a large list. There are menus on the task page which allow you to order tasks by deadline, priority, or view a single category, but this is not conducive to rapid access and immediately moving on to the next productive item. Additionally, I could not find a way to set a default sorting view or to filter the displayed tasks by deadline or other properties aside from labels.

Producteev is a powerful management tool, but these flaws are serious enough, in my mind, to justify attempting to build an improvement.

2.2 Version One

Version One is an Agile development management tool. Although it is used primarily for features, bugs, and deadlines within the context of Agile sprints and stories, it is not wildly different from a personal task manager in its core goals. For the purpose of this evaluation, I will exclude discussion regarding features that do not relate to a personal task manager.

2.2.1 Features

With far more features and metrics than Producteev, Version One runs deep in track-everything territory. There are dozens, sometimes closer to a hundred or more points of meta-data which can be attached to tasks (called Stories in Agile). You can track the difficulty of a task, a rough estimate of the time it will take, priorities, categories, authors, contributors, subtasks, deadlines, dependencies, and much more. Additionally, all of this data can be queried internally to find all tasks of a particular priority, author, size, etc.

Users log their progress incrementally by updating the amount of time spent and approximate completion progress of a given task. Various visualizations of individual and team performance are available to evaluate. You can examine metrics such as how long certain tasks take, how estimates and deadlines shift, and how much work you can take on the next week.

2.2.2 Limitations

In addition to the enterprise cost of using this system, the cost of this abundance of data is an abundance of tedium. The process to create and assign a task requires at least 2 page loads, usually more, to save the various settings you change them. The add task interface never has any less than a dozen text boxes and drop-down menus. Once you learn what you use and what data you want in it, the process can be fairly fast, but the overhead to learn these steps is nontrivial and filling out these forms is not simple even after establishing your habits.

2.3 Google Tasks

Abruptly jumping about as far to the other end of the spectrum as possible, Google Tasks is among the simplest forms of task management or more aptly to-do-list management that you can find. It requires a Google Account to use, and it is helpful if you are already a user of Google Calendar, but if that is not the case it is easy to get started.

2.3.1 Features

There are three options in Google Tasks: Title, Deadline, and Description, and none of these are required. You can create a task with no data associated with it, or with any combination of these attributes. There are no categories, no sharing, no prioritizing, notifications, extra meta-data, feedback, history, or any bells and whistles. The two differences between Google Tasks and a simple text file list is the ability to set a deadline on your calendar and the cloud hosting.

The beauty is this is enough to be tremendously useful. Adding a task is trivial and the ability to set a deadline is a simple and excellent way to allow users to have truly huge task lists spanning weeks or months which are available in bite-size or day-size chunks.

2.3.2 Limitations

The implementation of Google Tasks is quite hard to criticize since it really has so few features. This system was, in fact, my fall-back after Producteev drove me away, and I have used it quite successfully since. What has finally limited my usage of it is the development of my own solution. Google Tasks left me wanting for more. I would often supplement the task lists I had with either local computer notes or post-it notes on my desk to organize subsets of tasks, get a sense of what order to do things in, and estimate how long things should / did take. Google Tasks is missing the capability to tie additional meta-data to some of my tasks and the organization features and automation which digitizing data should allow but is not utilized in this instance.

2.4 Reflections

There needs to be some middle ground between the drop-dead simplicity of Google Tasks and the meta-data saturated computer-aided environment of Version One. Producteev is a partial solution, but it needs more. This careful balance is what I am exploring with my version of a task manager, Tasked.

3 Core Design Principles

3.1 Problem Areas

To make the better task manager, I worked on improvements to a selection of problematic traits in various task managers including those discussed in the Background section.

Too Many Options

It is far too easy to overwhelm or frustrate a user with options they do not want, need, or understand. The options available to the user need to be minimal and asked for.

Too Few Options

The flip side of overwhelming is frustrating users by not providing enough control to use a task manager they way they need it. A generally usable solution would be flexible enough to use as many or as few options as the user needs and is willing to enter into the system.

Barrier to entry on setup

New users need immediate access to the system, so the necessary overhead to create an account and start managing productivity needs to be minimal.

Barrier to entry for adding and managing a task

At least as important as the setup is retaining a user in a system they feel comfortable using and do not have to spend much time thinking about. Adding a task should be fast with as few hoops and requirements as possible, updating and editing existing tasks needs to be equally fast although it may be less crucially simplistic or minimal, and completing a task and providing any extra information needs to be easy to accomplish with minimal effort.

Task Paralysis

Anecdotally, one of the major factors in breaking down a work-flow is a feeling of being lost in your to-do list. I know once I have more tasks than I know what to do with, I avoid my task manager and fall further behind. To break this messy spiral of missed deadlines, there must be a way to view only an intelligently sorted subset of tasks.

User Feedback

Version One is the only task management solution I have used which incorporates real user feedback and analysis. I have only speculated this would be useful in a personal task manager as opposed to a development tool, but this project is really an experiment to see how it works. Whether or not the user sees it, collecting information about how tasks are completed could improve the intelligent sorting discussed in the paralysis avoidance requirement.

3.2 Design Principles

Keep the Interface Simple A good task management system should allow the user to provide as much or as little information as they like to promote the flexible usage of the platform. Wherever possible, hide extra data until the user asks for it or the system has a good reason to suggest it.

Provide Broad Meta-data Options Give users the opportunity to record as much information as they feel is worth including with a broad selection of flexible meta-data fields.

Automatically Filter Data First, Then Allow a User Complete Access Users should always have the option to view all of the information they entered into the system, but this should not be the first view they experience. By default, wherever possible, limit options and data to a small subset of what the user had told the system is important.

4 System Design and Implementation

4.1 User Accounts

In Tasked, users are stored in a user table which includes all of their login information. Passwords are salted and hashed and standard login information is stored including last login, a log of login history, email, and name. A separate preferences table is setup with defaults that are connected with a foreign key to the users table. These preferences are loaded into the users session data on login and used in the various views for rendering the application. The system supports password recovery, updating preferences, and account deletion.

At this time, user data is entirely private (tasks, user information, and preferences), but it was developed with the possibility of including shared data in the future.

Column	Description
taskid	A unique ID for the given task that the user does not have control over and effectively never sees or uses.
owner	Foreign Key for the user who created the task
title	The task itself. Most user accessed data about the task is the title of the task, a short description of what needs to be done.
location	Stored as a foreign key to a locations table containing the user-defined locations, each task can point to a single location or no location at all.
privacy	String representation of the privacy level based on an application level enum.
status	String representation of the status of the task, for example 'queued', 'inprogress', or completed
sleep	Date / time to sleep until. If sleep is non-null, a task will only be shown in the sleep listing until the sleep duration expires or is "woken up" by the user.
softdate	Soft or ideal deadline date.
softtime	Soft or ideal deadline time. This may be null at any time but must be null if softdate is not set.
harddate	Hard deadline date for a given task.
hardtime	Hard deadline time. As with softtime, hardtime must be null if harddate is null.
priority	Arbitrary integer task ranking. This is used in breaking ties and sorting tasks, higher numbers indicate greater priority.
estimate	Number of minutes the user thinks it will take to complete the task
actual	Number of minutes the user reported to have spent on the task.
progress	Number of minutes logged so far on the given task.
fragmenttime	Size of 'chunk' of the task, in minutes. If there is a minimum amount of time the user should spend on a task before it is useful time it can be reported here.
created	The date/time when the task was entered into the database
completed	The date/time when the user reported to complete the task or when the system marked the task as completed.

Figure 2: Primary Task Data

4.2 Adding Tasks

Task data is stored in a collection of tables. First there is the **primary task table** which is mostly reproduced in Figure 2. This table focuses on the core features each task is likely to use. A **secondary task meta-data table** has each row linked to a primary task table entry. The secondary table contains additional, less common meta-data. The goal of this table is to accommodate easy explanation for future arbitrary meta-data fields.

Task data is also stored in both primary and secondary log tables. Whenever the primary or secondary task table is changed a log entry is generated with old and new values and the timestamp. Review Section 4.3.5.1 for more details on this structure.

4.2.1 Task Usage History

Whenever a task is added, a log of the user's history for tasks is updated for use in future suggestions. These logs are tied with category, location (categories with geospatial coordinates attached to them), and the keywords within each task title.

4.2.1.1 Category and Location History When a user inserts a task, a log containing the category or location, and a true or false for each meta-data property is inserted into this table. If new meta-data is added, the value is defaulted to null and ignored in future operations.

In order to make recommendations from this table the system gets the total number of entries which have each attribute (non-null), then calculates the percentage of insertions which use that property. If this percentage is above a certain tunable threshold that meta-data attribute is returned as recommended by the system for that category.

4.2.1.2 Keyword History Similar to the category system, a keyword based model has two layers of logging that need to be done. The first layer identifies how common a given word is within all tasks across all users. Whenever a task is added this table is updated by incrementing the entry for each word.

The second layer records a similar record for each category but with a link to the given keywords. Again NULL is used for meta-data added in the future. Recommendations are then made utilizing this second layer in exactly the same way as the categories recommendations are generated.

4.2.2 Adding a Task User Interface

To reflect the goals of simplicity and minimal barriers to entry, the add task UI needs to be a simple text box. Unless the user specifies otherwise, a non-intrusive information bubble appears near the text box with the information on the available additional features. Namely:

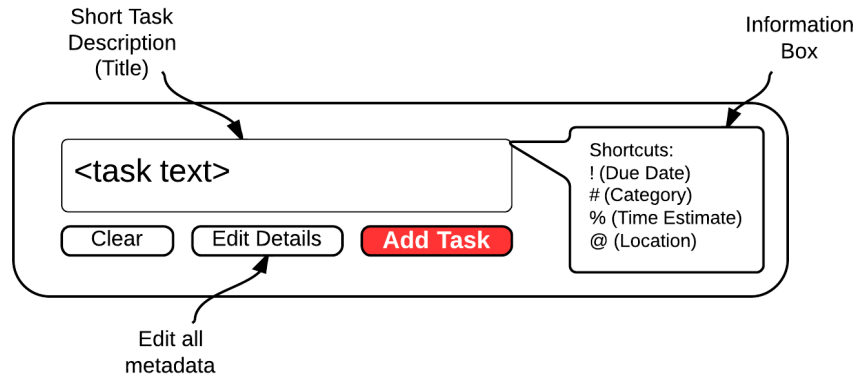


Figure 3: Add Task UI

- !(date) to specify due date
- #(category) to enter category
- %(estimate in minutes) to give the estimated amount of time a user will for the given task.
- @(location) to enter predetermined location.

As each of these inline commands are entered the information box changes to interpret the input. For date the information box will display the full date the user expects to have it done, for example "Monday", the information box might show "Dec 2, 2013". For category and location the helper will provide auto-complete listings as they type. Estimate will allow the user to enter a time such as '%45' and the information box will show '45 minutes', or another time format such as '%4 hr' and the information box will show '4 hours'.

This processing is done periodically as the user types using JavaScript and jQuery keydown listeners. When one of the special characters is detected the script will begin processing between the special control character and the next space character or 'blur' event. For as long as the input can be parsed as a valid value for the given special character it will update the information box. Once input becomes invalid or if another control character is entered the length of the entered value string, the position of the command character, and the interpreted value itself are stored in a JavaScript object.

When command characters are present the system will check for edits made within the value string and will attempt to process updates accordingly. If the user attempts to add a second instance of any of the command characters it is ignored with the info box warning of "You have already specified [command]!".

Input	Process
"Finish Project"	(nothing)
"Finish Project !"	Info Box displays "Due Date: " prompt
"Finish Project !January "	"Due Date: Jan 1, 2014"
"Finish Project !January 18 "	"Due Date: Jan 18, 2014"
"Finish Project !January 18 For 2014"	Since "For" ca not be processed as part of the date, the date parsing ends. Due date noted as Jan 18, 2014 in an object which also includes start position 15 length 10.

Figure 4: Input Processing Example

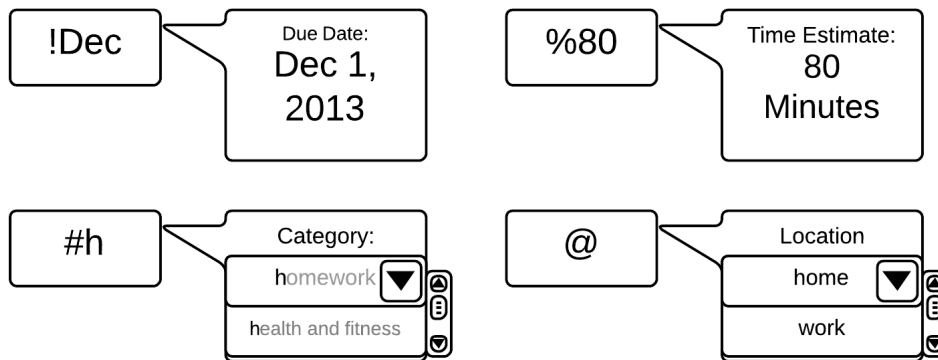


Figure 5: Information Box Examples

4.2.2.1 Suggested Meta-data Once the user enters any of the data tracked for meta-data menus an asynchronous query is made to see if any recommendations can be made. If the query returns a suggestion the client-side JavaScript determines if the field is already being show. If not, the recommendation loads quietly under the add task window and the user can choose to use it, ignore it, or hide it using a "close" (X) button (see Figure 6).

If no suggestions are found for a task entry, on occasion a suggestion box will be loaded below instead with text informing the user of the option for additional meta-data. The dialog, reading something like "Add additional information about your task:" includes a drop-down box with a list of available options.

As a backup, a full meta-data menu is also available through the "Edit Details" button on the Add Task UI element. This opens a menu which lists all of the meta-data options for a given task.

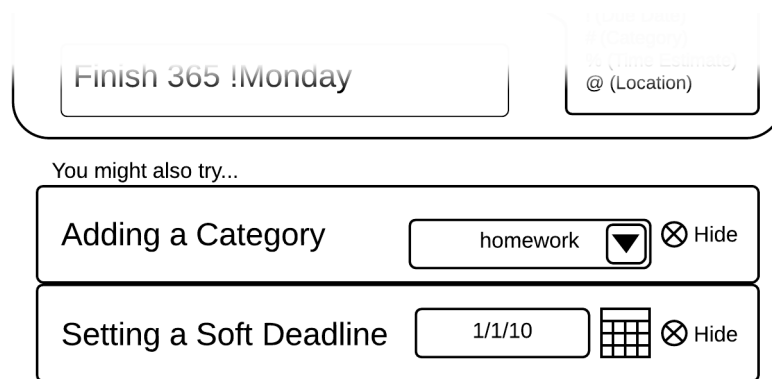


Figure 6: Sample Meta-data Suggestion Layout Under Add Task (See Figure 3)

4.3 Managing Tasks

How tasks are accessed after they are created by the user is a key area where the Tasked system differs from other popular task managers. As stated in the core design principles, users will always have the option to view all of their tasks, both completed and queued, and all of the meta-data associated with each task. In addition to the everything all at once view I complained about earlier, Tasked offers several ways to access tasks in a more manageable way.

4.3.1 Individual Task UI

To start off with, the way each task is displayed is with a somewhat large row that has key information about the estimate, the due date, and the task itself. This UI element draws a lot of attention to these three attributes as they are key to the priorities the system anticipates the user to have. Having the task, deadline, and estimate clearly visible make it fast for the user to pick a task they can do out of a list.

4.3.2 Task Recommendation - "What Next"

If a user has already created tasks, the first thing they will see when logging in is the "What Next" box. This box represents the aggregation of multiple types of task prioritization. The goal of this interface is simple:

1. Quickly access urgent tasks
2. Make long term goal tasks visible
3. Make tasks without a deadline visible

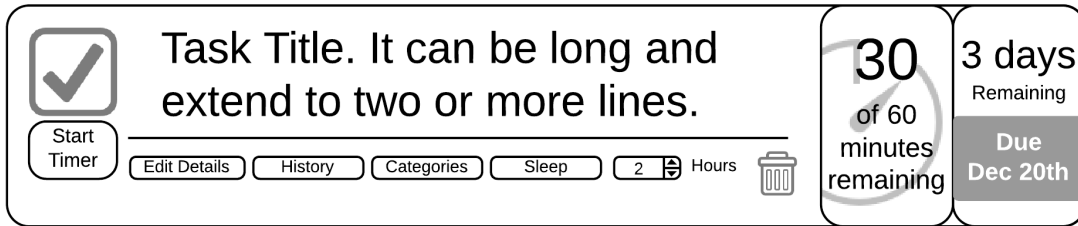


Figure 7: Layout For a Single Task Row.

4. Limit the number of tasks the user sees at one time

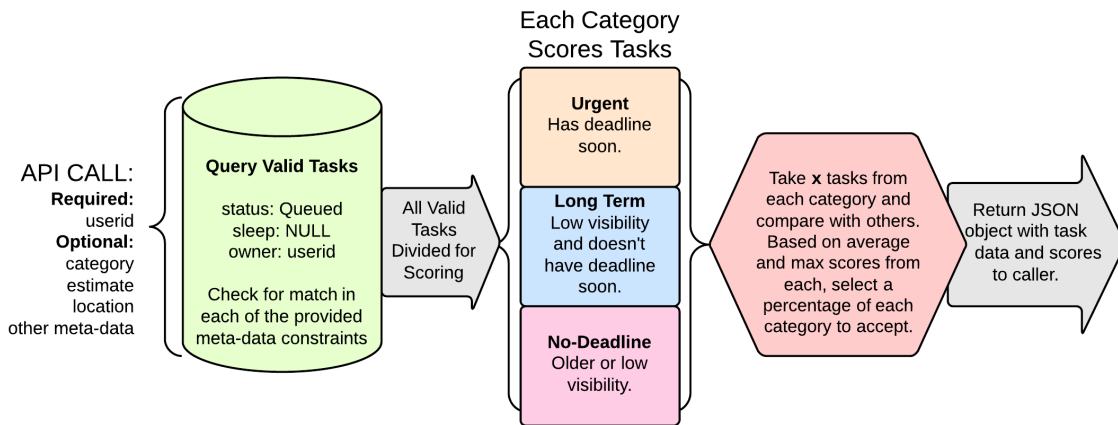


Figure 8: Recommendation Engine Overview

4.3.2.1 Scoring Tasks “What Next” is generated by an algorithm which sorts tasks into these categories then prioritizes them in a semi-random way. Each task has a visibility metric which counts the number of times it shows up in a result. The goal of the system overall is to balance visibility with task age and deadlines so tasks do not get lost and to encourage users to finish on time or early.

The ranking in the recommendation engine involves breaking a set of tasks into three groups, then calculating the score specific to each group. These groups are:

- **Urgent Tasks:** Ranked first by hard deadline, then soft deadline.
- **Long Term Tasks:** Encourage users to meet soft deadlines.

- **No-Deadline Tasks:** Protection against these tasks from getting lost.

Score Title	Task Group	Basis
Visibility Count	All Groups	Number of times a task has been seen. This factor is balanced by the creation date of a task to prevent tasks that have existed for a long time from being hidden due to a large number of views.
Last Progress	All Groups	If the task has been worked on recently. Tasks that have been started already in the same day will rank lower than tasks which have not been started.
Creation Date	No-Deadline	For tasks without a deadline, the longer it has been since the task was created the higher chance it has of being suggested.
Soft Deadline	Urgent, Long Term	Used as a percentage of "golden years". Tasks are scored based on what percentage of the time between the creation date and the soft deadline has passed.
Hard Deadline	Urgent, Long Term	Ranked higher than soft deadline scores, hard deadline scores work on task "lifespan". Tasks that are late must be put to sleep to hide. 'Later in life' tasks will always have a higher score in this category than tasks earlier in their lifespan, however those tasks in the last quarter of their lifespan will have a considerably higher score than others.

Figure 9: Scores Used in Suggesting Tasks

After a set of tasks is broken down into the three groups, each group generates it's own score based on different scoring criteria. All groups use visibility to some extent, and last progress heavily affects all groups such that tasks already recently worked on will be mostly ignored unless other tasks are significantly less important. Aside from the shared attributes, scoring is broken down for each group as follows:

Urgent tasks weigh hard deadlines most heavily, followed by soft deadlines and then visibility count.

Long term tasks are focused primarily on the differences between creation date and the hard or soft deadlines, whichever is relevant when requested. Long term tasks will downplay the hard deadlines to encourage meeting soft deadlines. The visibility of any of

the long term tasks is weighted comparable to soft deadlines, ensuring tasks with similar deadline range are seen.

No-deadline tasks primarily use creation date balanced with visibility to compete with tasks that might have hard deadlines.

4.3.2.2 Combining Group Tasks Three objects containing tasks and scores are passed to the next stage of recommendation. The scores will be on roughly the same scale so they can be directly combined into an array ranked by score. The system refrains from doing this though in order to ensure that at least one task from each of the groups.

The recommender has a number of tasks, x , which it is expected to return. This is determined by either an API call parameter or a recommender default. Then, using the average and max scores of each group the recommender assigns a percentage of the x returned tasks to be from each so it selects the top scores.

4.3.2.3 Task return and display A JSON object is then constructed and returned to the API call with the top X results. When the recipient displays the tasks it will callback to a visibility count API with the taskids and increment their visibility counter.

4.3.3 Sleeping Tasks - Allowing Users To Hide Results

If a user knows they do not wish to review a task at this time they can put it to sleep. This functionality sets a date and time in the future specified by the user as an offset from now. The task will not be returned in any query except for the "View Sleeping Tasks" page until that sleeping time has expired and the task is woken up. Tasks may be woken up at any time by the user when they visit the sleeping tasks page.

4.3.4 Filtering Tasks

4.3.4.1 Give Me What I Want An extension of the "What Next" tool, "Give Me" was designed with mobile in mind but is useful for rapid access on any platform. The Give Me tool rapidly filters tasks down to what the user can do immediately. The user provides the system with a keyword, location, category, time-frame, or another piece of meta-data and the filter is applied to the What Next results. See Appendix A for a full UI design.

The tasks that are returned can be fragmented tasks if the specified estimate requirement is larger than the fragment time for the task. This is useful if tasks with larger estimates have urgency. If no tasks can be found which match the criteria specified by the user, the results allow for some flexibility to suggest similar tasks. This is done on a per metric basis by choosing one or more metrics to ignore, or if a metric is numeric, the range can be broadened.

4.3.4.2 Sorting and Filtering Besides querying directly from tasks, you can filter and sort task results. By selecting a category it is possible to only view tasks which are in that category. Similarly you can filter on location, keywords (search), and deadline. Within the context of a filter or the list of all tasks you can also sort by any task metadata with the option to either hide tasks without that attribute or display them at the end.

Within user preferences the user can choose what their default view sort option is although in most instances the user will be starting on the “Give Me” or “What Next” pages.

4.3.5 Editing Tasks

Once a task is created, users have the ability to change any of the details. Although this technically opens the possibility for the user to transform any task into a completely different one over time the user is permitted this power. Such broad changes may be beneficial for the users purposes and while the metrics will be somewhat confused by blending tasks in this way there is potentially greater frustration created by limiting users seemingly arbitrarily.

In an effort to preserve metrics, if a task is edited frequently, surpassing a threshold for total number of edits and number of edits in 24 hours or so, the system will present a non-intrusive suggestion: ”It looks like you’ve changed this task a lot. Would you like to break this down into multiple tasks or create a new task?”

4.3.5.1 Logging Task Updates Edits to any task meta-data are logged in an updates table for tasks. This table consists of the taskid, event type, event time-stamp, and a counterpart for each column in the primary tasks table as well as an “old.value” version of each. Logs are generated whenever a task is added, updated, or deleted.

Adding or deleting a task generates a row with all of the columns.

Updating a task stores an old value and new value for each column which is updated.

4.3.5.2 User Interface The task UI focuses on the estimate, title, and deadlines. Each of these is immediately available to be edited by clicking on respective element.

A given task’s title, the primary text on the task row, allows the user to edit inline. Simply clicking on the text will make it editable. Clicking outside the text box (that is, on a ‘blur’ event) the change is sent to be committed to the database. If the title was simply clicked but nothing was changed the update will be stopped before calling to the database logging a change.

Task time estimates and progress are edited by clicking on the clock button. The progress can be recorded automatically using the tracking tool outlined in 4.3.6.1, or you can manually edit it. You can enter your progress so far in minutes, or if you like, you

may edit the task estimate. When editing the estimate itself the user will get a notification asking whether or not they really want to change the time estimate. Task estimates, progress, and any updates to them are subject to user reports (outlined in section 4.4). While estimate changes are tracked, the user is reminded that changing an estimate to match the actual time to complete a task provides less benefit from reporting. Users can disable this warning with a "do not show this again" option which updates the user preference on estimate warnings.

Both soft and hard deadlines can be edited by clicking the calendar deadline button to the far right of the task. A similar warning is displayed if a user attempts to edit these deadlines with the same "do not show this again" preference.

4.3.6 Completing Tasks and Tracking Time

The process of completing a task is a balance of two key principles. First and foremost the process must be simple, minimal, fast, and painless. Users should go through no hassle to get their well earned strike-through and fadeout on a task. At the same time, user feedback is an important part of the system. To aid the system in recommendations and reports for the users consideration, Tascked attempts to collect the actual time taken to complete the task, when the task was completed, and some idea of how the work was divided.

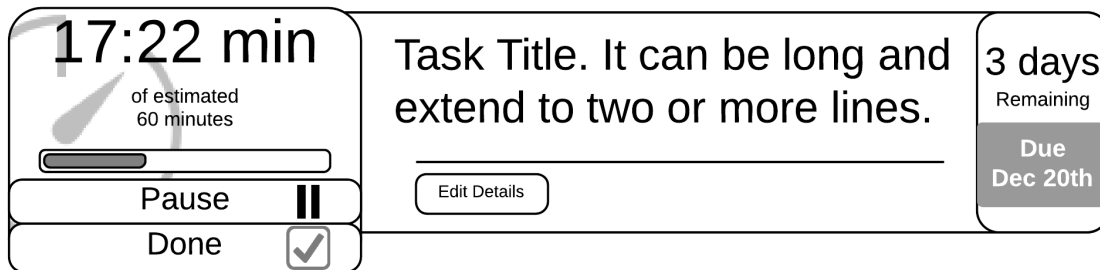


Figure 10: Tracking Task Time

4.3.6.1 Tracking Time To aid in the collection of actual task completion time and to help users stay focused on a single task, Tascked is designed with a timer which you can start for each task. When you press the "Start Timer" button on the task, the task you selected will be moved up to a "What Now" section just below add task. User can now close the browser window or open the task manager on another computer and the timer will persist until it is stopped by pausing or clicking "Done" to complete it.

Timer information is stored in a timer log table. Each row consists of a sessionid, taskid foreign key, start timestamp, pause / done timestamp, and an active boolean column. When a page loads, the system checks if the user has any active timers. If so, this is recorded in their session until they dismiss the timer with pause or done. Upon dismissal the counter stops, and the database is updated with the end time and the new active value of false. When the database update completes the timer is cleared from the “What Now” section and the task is returned to the task list if applicable.

See Figure 7 for Task UI. See Appendix B to view the “What Now” timer UI in context.

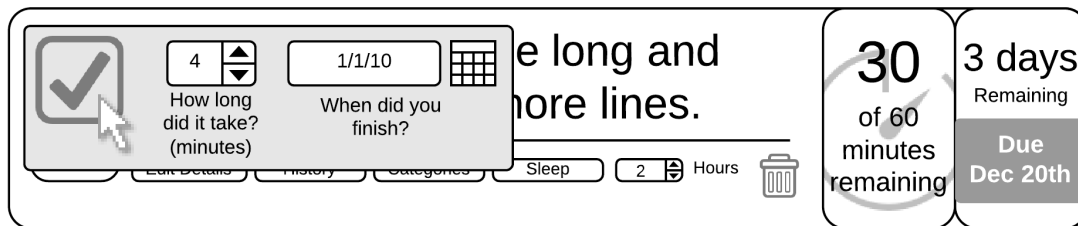


Figure 11: Completing a Task

4.3.6.2 Complete Task Task completion involves some optional meta-data collection which the user can choose to ignore if they are in a hurry. To complete a task, a user either dismisses the timer tool with the “Done” button or hovers over the checkbox in a normal task row. When the user’s cursor enters the checkbox an additional menu pops out from underneath revealing an **actual time** and **completion date** form. The actual time field is filled automatically with the estimates and the completion time field

4.4 User Reports

In addition to making suggestions for users, a core element of Tascked is the option for users to examine and visualize the data that is collected. Some of the currently designed reports are described below.

Activity Feed Users can see all of the actions they have taken on their account with newest first. This report can be filtered by type of action, or value updated. The user will see a number of rows with the timestamp of the action and a brief human readable summary. For example an entry might read

“ Dec 4, 2013: Changed title from “Wash the cat” to “Wash the car” ”

“ Dec 3, 2013: Added Task “Wash the cat” ”

Estimate Values Another report outlines how task estimates change and how well they are observed. For the former, a chart with the number of tasks whose estimates were changed by certain increments (+10 min, +20, -10, -20, 0, etc) would provide the user with insight to trends in increasing or decreasing estimates.

The user will also be able to see actual time vs estimated time. This consists of two charts: by percentage of task estimate and how the actual time related. For example, if I have three 15 minute tasks that actually took 20 minutes, each would be counted as off by +33%. The second chart would be off time in minutes, so for the 15 minute task example they would each be counted as off by +5 minutes.

Completion Examination Using either the completion time timestamp or the timestamp for the update where the task was completed, Tascked will generate a daily view of the number of tasks completed each day and how far off, if at all, the tasks over a given time period were.

Aggregation Aside from distinct reports Tascked can also provide aggregate updates to the user periodically. This might involve a static widget on the page with stats like average number of tasks per week, estimated workload last week, actual workload last week, estimated workload this week, percent of tasks completed on time, and more. There has not been sufficient data generated in the system to determine what metrics would actually be of interest, but over the course of continued development as that becomes clear the options for displaying various widgets can be made available to the end users.

5 Implementation Notes

The implementation of the Tascked design is a work in progress. Tascked.com, the domain hosting the current developments, has implemented the following key features.

- Adding Tasks
 - Supports priority, title, estimate, hard deadline, and soft deadline
- What's next / give me
 - Recommendations use only deadlines, priorities, and creation dates. Visibility logging is not complete yet
 - Categories and location are not accessible to the end user.
- Viewing tasks
 - Tasks can be put to sleep. At this point sleep defaults to 2 hours.
 - Sleeping tasks can be viewed and woken.

Completed tasks can be viewed and un-completed.

View all tasks loads all user tasks that are queued.

- Editing tasks

Task title can be edited inline.

- Completing tasks

Supports actual time, date and time completed, and the capability to un-complete a task in case of accidental completion.

While not specifically a feature, a component missing from the current development version of the project is a cleanly implemented user interface. The application supports asynchronous calls and all of the data is available on the page but it is not quite as sharp as I had hoped to make it. It is possible time could have been made for improving the UI but to finish on time focus was instead directed at solidifying the design and implementing portions of the project for testing and proof of concept.

6 Future Work

6.1 Task Suggestion Improvements

The task suggestion algorithm is really an experiment. It produces interesting results but how well it scales and how it reacts to a broader set of user data than I was able to produce is unknown. A major component of future development will be adjusting this algorithm, experimenting with different options, metrics and weightings, and attempting to evaluate the results.

6.2 Mobile Development

To allow this application to reach a broader audience and become an option for users who might not be working at a computer during most of the day a mobile option is necessary. The “Give Me” interface would be a logical starting point to allow users rapid access to an action item. Soon after the ability to add a task, edit a task, and view all tasks would also crucial to prevent user frustration. It may not be possible to easily port all features from the web application to a mobile device, but the most commonly used and the mostly likely to need mobile attention features such as adding, completing, filtering and sorting tasks should be a high priority.

Other features could be improved by having a native-like mobile experience. The task timer system could be started on your phone and stopped or managed on any other device. This would free you from the necessity of being near a computer when using this feature.

Location features would also benefit. If you opted to attach approximate GPS coordinates to certain locations your phone could automatically load tasks that specifically require your current location.

6.3 Task Timer Improvements

Taking the action of starting a task timer is a large incentive to stay on task and focused. It's easier to definitively be working on something when time is being calculated while you work. However, as designed the system does not offer forgiveness if you forget to stop a task timer. A time log should be implemented to allow the user to enter start and end times as they see fit, as well as edit the automatically captured times to correct timer overruns.

6.4 User Interface Overhaul

The system could benefit from consulting a user experience designer. An important element of this project is simple design with enough user functionality. While in some respects this is achieved there is certainly room for improvement.

6.5 Data Liberation

Liberating data is important. Aside from calendar events or outlook tasks I could not find a consistent standard to personal tasks. It might be worth attempting to create one so that users can free their data. It would be difficult to ship the history and logs but the completed task list and the queued tasks could easily be exported at least as JSON, CSV, or Excel.

It would be possible to create custom formats as well. Providing users with the ability to map a schema to the data stored could be a powerful way to make users comfortable trusting the system. Power users could generate these schemas for various systems (Producteev, outlook, custom applications) and users could browse a schema market for their particular needs.

6.6 Exploring Caching, No-SQL and other Database Solutions

By exploring caching and other database solutions there are potentially huge performance improvements to be gained. The current system design presumes all data is generated on the fly and pulled from a SQL database. Many operations may be improved by caching key/value pairs for tasks, categories, or logging. Meta-data suggestions could be cached in an in-memory or other fast database to rapidly access suggestions as users type. This alleviates strain on the database from large numbers of users and give the end user a faster experience. There are plenty of avenues to explore that should be examined as the plan to provide the service to more than one user develops

7 Conclusion

This project started as an experiment on a hunch. Over the course of the project I have had the exciting opportunity to try some of the features which I had dreamed of having and evaluate them for myself. The features I was able to play with most are the time estimates and a version of the “What Next” recommendations. Somewhat to my surprise they functioned in an interesting and useful way. I have been using the system as it is for several months now as my primary task manager and it has addressed several of the features I had long wished for, and as more of the design is implemented it will become even easier to use. I do not yet know if the interest in using Tascked comes from the novelty of a slightly different system or merits of the ideas, but as I continue to use the system and invite others to do the same I plan on finding out.

In reflecting on the development of this project I would change some aspects of my approach. The idea for the project was quite exciting and interesting for me so I spent a lot of time jotting down half worked out ideas which sounded neat and somewhat related. When I began implementation I started with basic functionality and then moved on to experimenting with ideas. It was not until after a few experiments that I realized some of the features I had planned were not compatible with the somewhat haphazard, spur of the moment initial design. At that point I was able to take a step back and work through how different systems tied together, what made sense, what could be modified to fit, and what was not suitable for the project in its current state.

Doing it again, and continuing in development, I will approach even exciting experimental ideas with hard hitting design questions about where they fit in with the system. Despite the redesign somewhat late in the development in the system I find I am very pleased with the design of the project and the prospect of continuing work in implementing the now better planned design. Here’s to open enrollment in Tascked.com in the near future.

References

- [1] ilanewyork, “Producteev by jive (aka 4.0): The birth of a giant,” Jive News Blog. [Online]. Available: <https://community.jivesoftware.com/community/newsroom/blog/2013/05/21/producteev-by-jive-aka-40-the-birth-of-a-giant>

Announcement of the Producteev release under the new Jive management.

- [2] Producteev, “Producteev by jive,” Producteev.com. [Online]. Available: <http://www.producteev.com>

The Producteev product homepage with features and signup.

- [3] victoria.blake, “Fall release preview: Jive + producteev,” Jive Products Blog. [Online]. Available: <https://community.jivesoftware.com/community/products/blog/2013/10/26/fall-release-preview-jive-producteev>

Announcement of new Producteev features.

Appendix

A Full Page Mock Up “Give Me” Filter Tool

Add Task

The "Add Task" form consists of a rounded rectangular container. At the top is a text input field containing the placeholder text "<task text>". Below the input field are three buttons: "Clear", "Edit Details", and "Add Task" (which is highlighted in red). To the right of the input field is a tooltip box with a pointer to the input field. The tooltip contains the text "Shortcuts:" followed by a list of symbols and their meanings: "!(Due Date)", "# (Category)", "% (Time Estimate)", and "@ (Location)".

Give me something...

The "Give me something..." filter tool is a vertical stack of five rounded rectangular boxes. The first box contains the text "to do in" followed by a small input field with an "x" and a spinner control, and the word "minutes". The second box contains the text "in the category" followed by a dropdown menu with the placeholder "<category>" and a downward arrow. The third box contains the text "with the words" followed by a text input field with the placeholder "<keywords>". The fourth box contains the text "that I can do at" followed by a dropdown menu with the placeholder "<location>" and a downward arrow. The fifth and final box is a large rounded button containing the text "Give me anything!".

B Full Page Mock Up “What Next” Task List

Add Task

Clear Edit Details Add Task

Shortcuts:
! (Due Date)
(Category)
% (Time Estimate)
@ (Location)

What Now

17:22 min
of estimated 60 minutes

Task Title. It can be long and extend to two or more lines.

0 days Remaining
Due Dec 6th

Pause

Done

Edit Details

What Next

View All Completed Sleeping Filter

Proin tempus dolor non nulla sollicitudin cursus.

Start Timer Edit Details History Categories Sleep 2 Hours

15 of 90 minutes remaining

No Deadline Specified

Fusce id mauris a ligula commodo.

Start Timer Edit Details History Categories Sleep 2 Hours

30 of 30 minutes remaining

Late By 1 Day
Due Dec 5th

Nulla eu mi sit amet mauris rhoncus tincidunt. Etiam non augue ac erat lacinia .

Start Timer Edit Details History Categories Sleep 2 Hours

75 of 90 minutes remaining

15 days Remaining
Due Dec 21st