Project 3 - Shirk: A Forgiving Task Manager

6.170 Software Studio | FA14

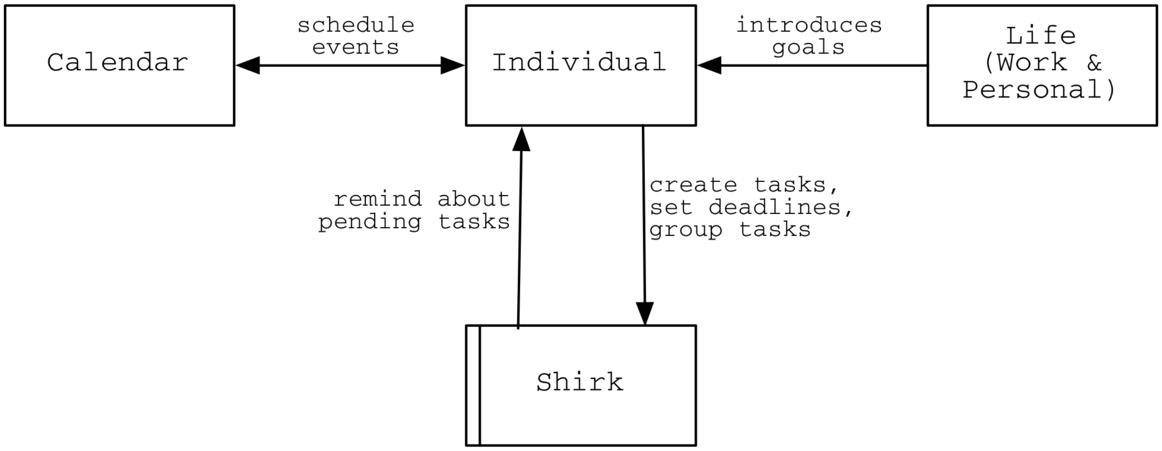
Motivation

Shirk is a web application designed to help individuals stay organized and productive. It aims to replace the paper todo list by providing a platform for users to record, view, and complete tasks. It also supports grouping, scheduling, and prioritizing tasks.

The purposes of *Shirk* are:

- **Capture incipient tasks in detail.** - For *Shirk* to be as fluid as a paper list, users must be able to record their thoughts as they form. It must provide a clean method of entering a new task, with the option of associating deadlines and 'scribbling' notes.
- **Organize and tack pending tasks.** - *Shirk* allows users to categorize tasks and assign relative priorities to each. It also displays uncompleted tasks by various importance metrics (priority, deadline, grouping).
- **(Potentially) Reflect on productivity.** - *Shirk* can provide productivity statistics based on task completion. This includes indicators such as the fraction of tasks completed on time or longest completion streaks.

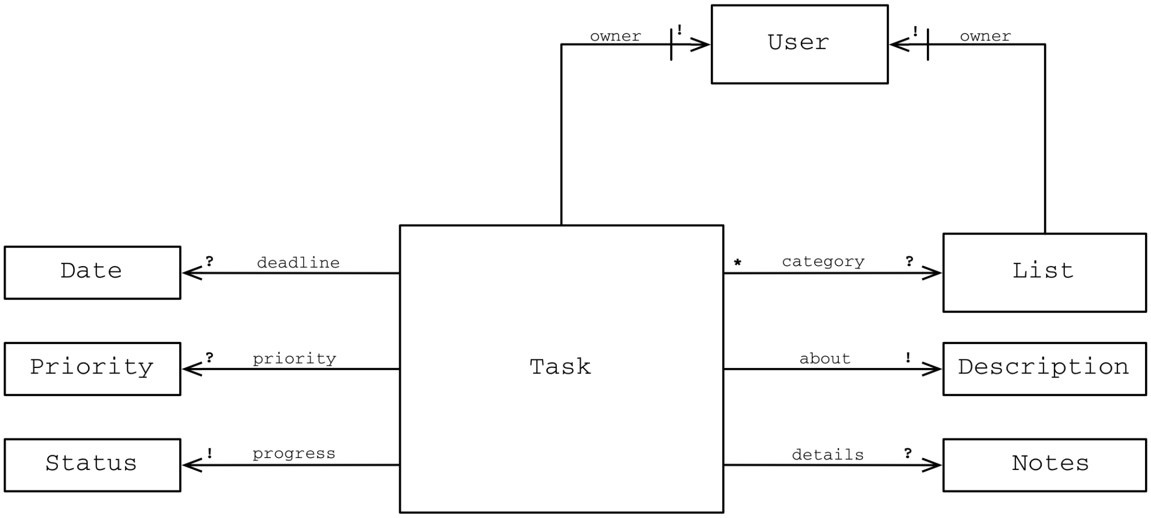
Context



Concepts

- **Task** - A specific chore or undertaking, constituting a single unit of work. Each task must have a concrete description representative of the work; the description acts in place of a title. Each task may, optionally, have corresponding free-form notes, although these are only to provide supplemental information for the user.
- **List** - A grouping that contains individual tasks, usually for a similar purpose. Each task can belong to at most one list, and can be moved from one to another. If a task is not within any list, it is referred to as "uncategorized".
- **Deadline** - The date by which the user aims to complete a specific task. Assigning a deadline is optional; tasks without dates are considered "open-ended". Tasks are considered "for" a specific day if their deadlines falls on that day. For instance, users would typically view tasks for today, meaning the application will have a means of displaying all tasks whose deadlines are today.
- **Shirking** - The action of delaying a task that is due today by a single day. If the user is unable to complete a task, the task can be *shirked* until the following day. Changing the deadline of a task is not considered shirking the task.
- **Finishing** - The action of completing a task. A *finished* task is flagged as completed, although this does not affect how the task can be mutated -- its list and deadline can still be changed. The task would no longer be displayed when viewing upcoming and overdue tasks, essentially removing it from the working set.
- **Priority** - The importance of a specific task, as assigned by the user. Each task can be optionally classified as "high" or "low" in priority, distinguishing it from neutrally pressing tasks.

Data Model



Additional Constraints:

- All ***Tasks*** referencing a ***List*** must have the same *owner* as the ***List***.

Challenges

Shirking Tasks

The concept of shirking, or delaying, a task can be represented in varying ways. It can signify either follow a conscious refusal to complete a task or simply neglecting it.

- Users consciously shirking tasks on their due date is not a very feasible option. For one, if the user forgets to shirk a task, there would emerge a distinction between *delayed* and *overdue*. For instance, the task may be several days past its deadline, whereas the user has only requested to shirk it a single day. The number of shirk days taken from the deadline also needs to be recorded, and would be complicated to adjust for a modified deadline. The positive for this approach is that the user is aware of the work left undone, since each task must be individually shirked.
- [Selected] A better approach is to make *overdue* tasks synonymous with *shirked* ones, meaning that all neglected takes would be automatically shirked. This would require no intervention from the user, and would reflect the number of days passed since the deadline, a usable indicator of urgency. Changed deadlines would not be an issue, since the number of shirked days would adjust correspondingly. However, since the process is automatic, the user may be unaware of the overdue tasks.

Determining Task Importance

Pending tasks may have an associated priority or deadline, or both. The idea of relative "importance" becomes relevant when comparing tasks across these two indicators. The problem lies with ordering and representing these tasks corresponding with this importance metric.

- The tasks can all be combined into a single listing and ordered by the accepted 'urgent and important matrix'. Urgent/high tasks would be most important, followed by urgent/low, not urgent/high, and not urgent/low. Since we represent priority as high/neutral/low and deadlines can be early/timely/late, we would need to extrapolate on the accepted standard to accommodate for these added values. This is further complicated by the fact that tasks can omit having a deadline. In any case, the user is unlikely to agree with the formulated ordering for a subjective view of ordering.
- [Selected] Since priority and deadline are not directly comparable, it is possible to leave them separated. Within each date period -- overdue, due today, next seven days, etc. -- the tasks can be ordered by priority. With this simpler representation, the analysis of importance is delegated to the user. The only shortcoming is that viewing the wrong date range (ex. next month) may allow the user to overlook what is urgent/high. However, the UI can be designed to warn the user if this occurs.