

## HOMEWORK 2: HEURISTIC EVALUATION

Ekaterina Ob'yedkova

*katya\_@mit.edu*

*Collaborators: None*

### 1) Add a task to the to-do list: Household Dashboard Metaphor

*Principle: Match the real world, Learnability*

*Severity: Positive, Good*



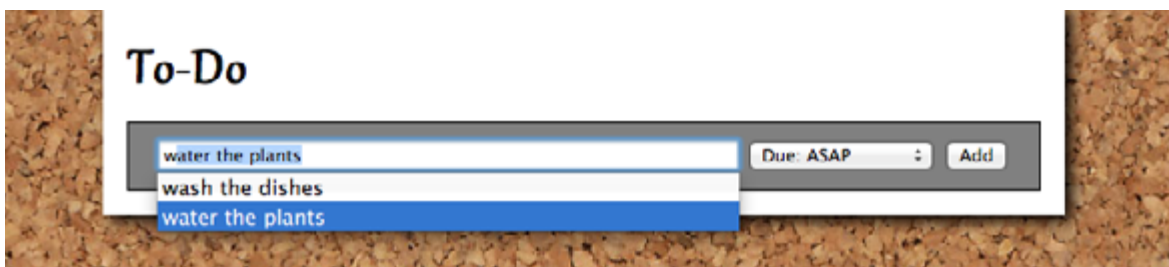
The interface mimics a dashboard, which immediately makes the purpose of the interface extremely clear. Analogy with sticky notes and pins communicates well the main purpose of the application: making it quick and easy to distribute household chores. To-Do list pops out against the grainy texture of the board and the light shadow also helps to visually bring the list to the front. Nevertheless, I found that after a while the texture map makes eyes feel tired. The texture contains a lot of details that may increase cognitive load. The main information is located in the middle with large empty space that is currently mapped with texture. I would suggest limiting the area where the board's texture is applied in order to minimise eyes load while maintaining the dashboard metaphor.

Although the application is designed to be viewed on a desktop, current design will work well on iPhone/iPad.

### 2) Add a task to the to-do list: Autofill

*Principle: Anticipation, Efficiency*

*Severity: Positive, Good*



The application saves the tasks entered previously making it efficient to add items to the list. The items are listed based on alphabetical order which is also convenient. On 'enter' the element gets added to the list with the due date automatically chosen from the current state.

### 3) Add a task to the to-do list: No Confirmation Dialog

*Principle: Error Prevention, Safety*

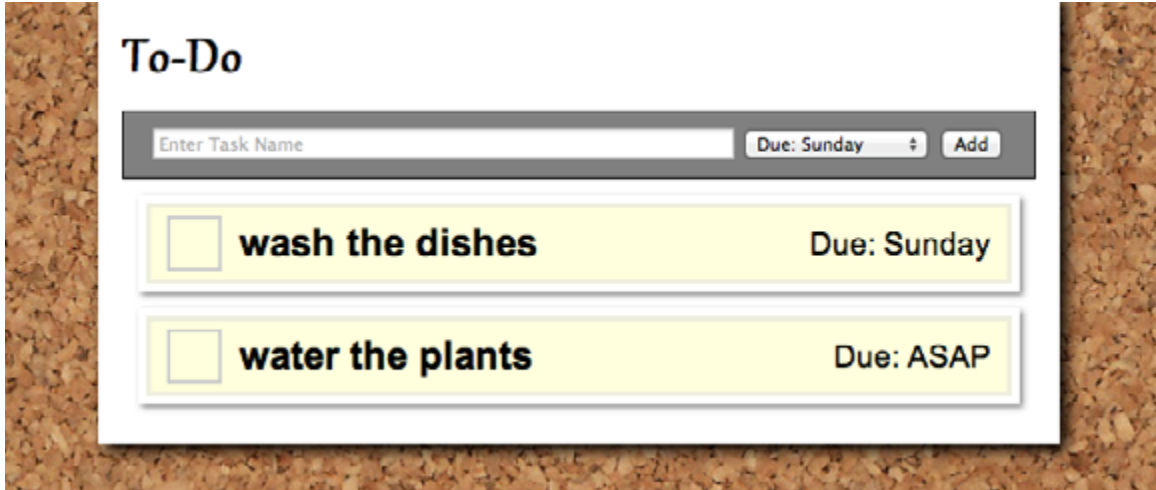
*Severity: Negative, Minor*

When the user hits 'enter' the due date is automatically chosen from the current state. I suggest having a pop up window that confirms the due date prior adding the chore to the list would be helpful.

### 4) Add a task to the to-do list: No Undo

*Principle: Reversible actions, Safety*

*Severity: Negative, Major*



Once the task is added there is no undo option.

### 5) Add a task to the to-do list: No Delete

*Principle: Reversible actions, Safety*

*Severity: Negative, Major*

Once the task is added it is impossible to delete it.

### 6) Add a task to the to-do list: No Edit

*Principle: Reversible actions, Safety, Efficiency*

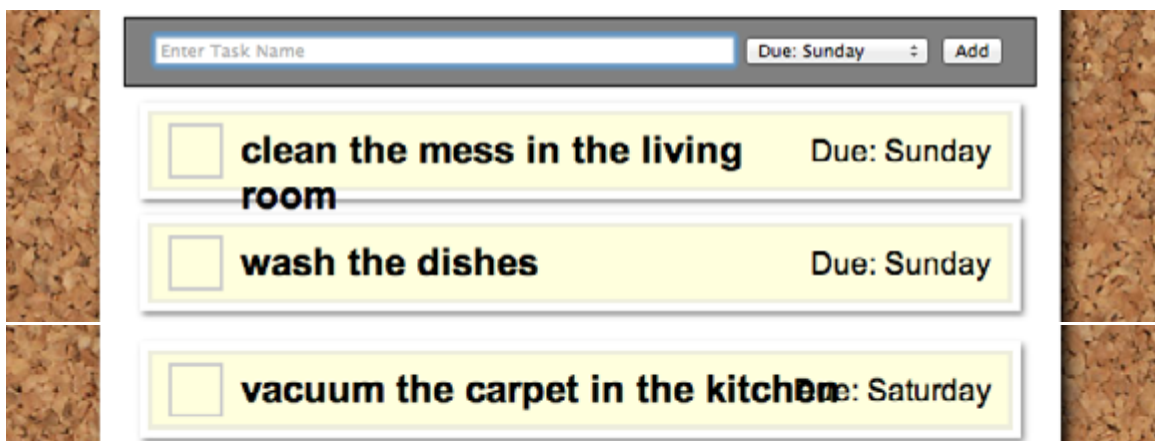
*Severity: Negative, Major*

Once the task is added it is impossible to edit the due date. The urgency of the problem can often change. Having an edit option will give the user more flexibility and control. Also if nobody has volunteered for the task the status may need to be changed to ASAP or have some other special highlight.

### 7) Add a task to the to-do list: Window size is not adaptable

*Principle: Consistency*

*Severity: Negative, Major*



Window is not adjustable, therefore text appears to be outside the boundary or overlaps with Due.

## 8) Pick a task and mark it's status: Insufficient Information Scent

*Principle: Learnability*  
*Severity: Negative, Minor*

When the user hovers over a chore, the mouse cursor shows up as a hand, which gives an indication that an item is clickable. However, there is no indication as to what is going to happen when the item is clicked. I would suggest that either having a text box appearing with a hand or a button 'pick this task' could improve learnability. Adding a button could, however, compromise the simplicity of the interface.

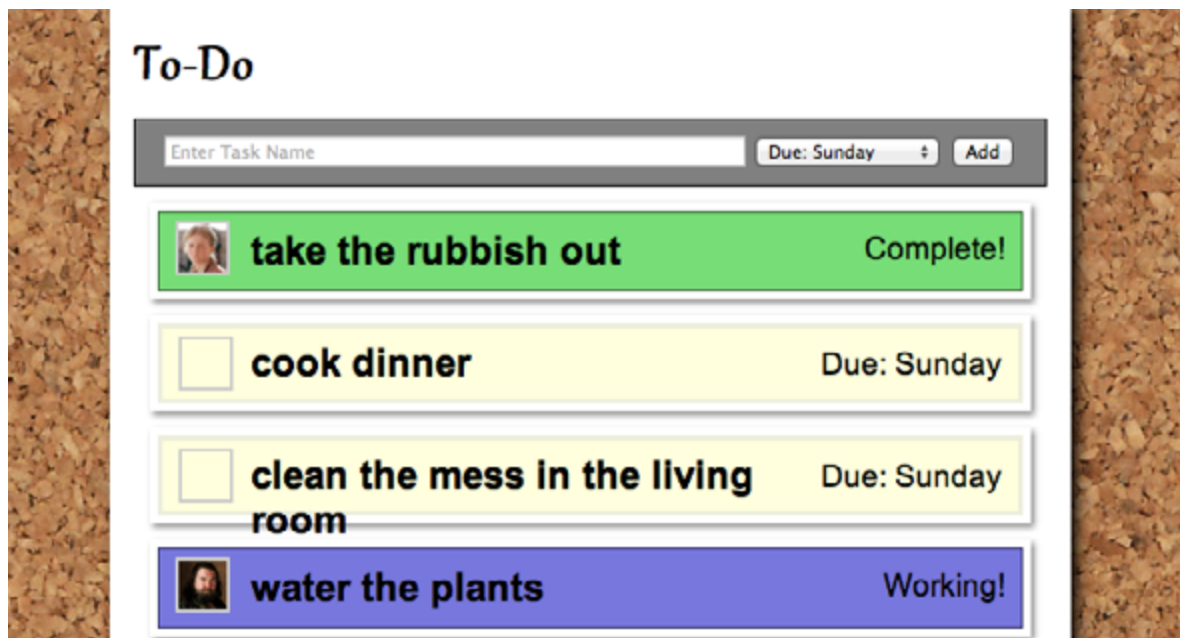
## 9) Pick a task and mark it's status: Overwriting an already chosen task.

*Principle: Protect User's work*  
*Severity: Negative, Minor*

If a task has already been chosen, it can be easily overwritten if another person tries to choose the same task. Also, a task may require multiple people working on it. The only way to signify this with the current implementation is to add it to the To-Do list multiple number of times.

## 10) Pick a task and mark it's status: Ordering To-Do list.

*Principle: Track state, Efficiency*  
*Severity: Negative, Major*



The tasks that are already picked do not get shifted to the bottom of the list. Additionally, the list is not sorted based on the due date. As a result, additional attention is required from the user to estimate the urgency of the tasks as well as visually find vacant tasks. Items do not get removed from the list regardless of how many items there are or how old they are meaning that scrolling through an unsorted list may become time-consuming.

## 11) Pick a task and mark it's status: Color highlight.

*Principle: Track state, Efficiency*  
*Severity: Positive, Good*

Having different color highlights to indicate whether the task is completed or already taken helps to maximize efficiency. The chosen colors are also externally consistent. Green is often used to indicate a positive status of a system and blue is often used to indicate work in progress.

## 12) Pick a task and mark it's status: On reloading data is not saved.

*Principle: Protect User's Work*  
*Severity: Negative, Major*

All the items get deleted when the page is reloaded.

### 13) Pick a task and mark it's status: Window size is not adaptable.

*Principle: Visual Consistency*

*Severity: Negative, Minor*

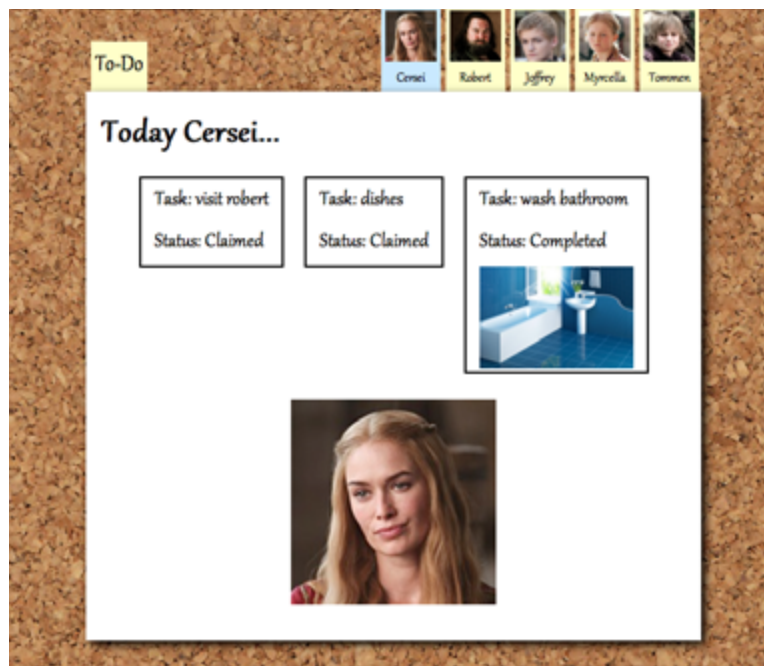


The window size does not scale based on text length, therefore, appearing inconsistent graphically.

### 14) Check what another person has done: Only today's tasks are visible.

*Principle: Track state, Learnability*

*Severity: Negative, Major*



Although tasks can be scheduled a week in advance, it is only today's status that appears on personal notebook page. It may be that not only the user is interested in other people's activities but needs to track his/her own. It will also be useful to show the due date together with the status and the task.

### 15) Overall design: Simplicity.

*Principle: Aesthetic and Minimalist design*

*Severity: Positive, Good*

In overall, the design is simple and aesthetically well-thought through. Short-term memory is reduced by having simple visual clues and powerful use of metaphors. There is no inessential variation in visual variables or features. Contrast is well used to establish hierarchy and bring attention to the To-Do list.