

Team: *Don't Forget!*

Andy Hou

Elisabeth K. Olson

Kenneth Kuan

Kevin Chiu

CSE 490F

Assignment #8 – Pilot Usability Study

12 February 2007

Introduction

The *Don't Forget!* system combines RFID technology with a calendar to allow a user to remember not only where and when their appointments are but what they have to bring with them for those appointments. Most importantly, the system will allow the user to check whether they actually have those items before they leave for their appointments. While the calendar can – and arguably should – resemble current calendaring programs, the by-the-door interface has very few precursors that we know about. Both, however, represent a chance for us to examine the process of developing user interfaces that are not just functional but also appealing.

In this phase of our testing, our main focus was exploring the usability of the core features in our interface. The previous iterations of testing only allowed limited exploration of the usability, but with a full-fledged programming environment we were able to fully implement the functionality we wanted to explore. However, we tried to avoid setting our design choices in stone, and purposely shied away from “prettying up” our interface so that we could have our testers focus on the actual usability without distractions such as font and image choice.

Method

Our nine participants were all computer science students, with an exception of one computer savvy engineering major, and all were between 19 and 22 years old. We tested with seven males and two females. Our testers had a wide range of calendar use, from nonexistent to paper calendars to electronic calendar programs. While ideally our testers should have been a bit older and more diverse in gender, concentration, and computer literacy to reflect our wide user demographic, these testers were chosen mostly for convenience.

To test, we started the interface on a personal computer and asked the tester to go through the tasks while we watched. This was done in a quiet setting which varied depending on the person conducting the tests. Due to time limitations, we did not use either a tablet for the touchscreen interface nor an actual RFID scanner, and instead asked the user to pretend the mouse was their finger and mimicked how a scanner would work with what we thought was a realistically slight chance of faulting.

We revised our earlier tasks to better explore our new interface. We kept the easy task the same – using the scanner to verify all needed items are present when leaving the house – but tried to introduce new difficulty into the medium and hard tasks. While the old medium task was to add an item while the hard task was adding an activity, the new medium task is basically a combined version of the two tasks, and the new hard task –

modifying and then deleting an activity – explores functionality we weren't able to implement in earlier iterations.

The first task involved asking the tester to use the system to verify that they had their needed items. This allowed us to assess how well the password feature worked, which is something we couldn't really test in the earlier iterations without too much backend work to be worth the return we'd get. It also allowed us to examine how users interpreted the "Forgotten Items/You Have Everything" page, because we could see their reactions to it and how long they examined it before declaring the task completed.

For the second task, we asked the tester to schedule a dinner with their friends, and told them that they wanted to remember to bring a certain pair of shoes with them. As there were no shoes in the system, this meant that to successfully complete the task the user would have to add an item as well as the activity. We wanted to see whether the testers would pick up on the idea that they had to add an item to the system, and also see how they went about doing so as there were two options. This task also allowed us to examine how users actually input data about their events, and how the structure of adding items to activities worked in general.

The last task involved two parts – first the tester had to modify the activity created in the first task, and then the tester had to delete the activity entirely. We considered this a hard task because these functions are somewhat hidden in our interface – the only way to find them is to actually click on an activity in the calendar. The main thing we wanted to see with this task was whether or not the edit and delete functions were even discoverable, and if so how the edit interface worked for the user.

To implement the testing, we set up the computer with the program running, brought in the tester, briefly explained the project and interface, as well as explaining that they were able to stop testing at any time, and then started them on the tasks (the complete text of the tasks can be found in Appendix 1). While performing the tasks, the users were asked to explain their actions, what they were thinking, and anything that they liked or didn't like. Each person in the group was responsible for having two people test the interface, and while the tester was working through the task the group member took notes on anything significant. Significant events included any time the tester had trouble with a part of a feature, if the tester indicated either orally or through body language that something was counter-intuitive, and times when the tester thought the task was completed when parts were left undone. After all the tasks were completed, the group member asked the tester what they liked and didn't like about the interface to prompt any thoughts that hadn't come out during the rest of the process.

Test Measures

In our tests, we focused on measuring the number and type of errors the testers encountered. We thought these would give us better data than any timing data that we could collect. All of our tasks are very quick to complete (or, if they're not completed quickly, there's a bigger error involved than task time to focus on), so it would be hard to separate statistically significant results from random and systematic errors in timing, while recording where the users had errors and what specifically caused those errors gives us a good place to focus in the next iteration. Also, due to the problems we had implementing full functionality in previous iterations and the fact that many of our features are appearing in a reasonable form for the first time in this iteration, we decided

to focus on exploring the functionality of this iteration of development much like we explored our paper prototypes, with more qualitative than quantitative data.

Results

Among all our nine testers, we had a total of 11 errors for the first task, 19 errors on the second, and 6 on the third, which is an average of 1.2, 2.1, and 0.67 errors/user on each task, respectively. This was a bit higher than we expected, but most of them were clustered on the same type of problems. A full list of the problems for each task, listed by tester, can be found in Appendix 2.

For the first task, the password page was responsible for all the errors users experienced. In particular, the representation of numbers once they've been typed in and the "enter" key were the most confusing, accounting for 6 of the 9 errors users experienced. The other three problems were due to the environment the program was being run in, with testers trying to use the keyboard rather than the buttons on screen and attempting to use the browser "back" button rather than the "Home" button provided by the interface.

The second task was by far the most problematic, with only one tester managing to complete this task without a problem. There were four main problems testers had, with all four encountered by three to four testers. The first problem was that users tried to click the calendar to add the activity, while we only supported adding the activity by clicking the button to do so. After reaching the "Add Activity" window, many testers were tripped up by how the start and end times were implemented – especially when it came to the difference between AM and PM times. One user encountered repeated "time sanity" checks – meaning that the start and end times entered were either exactly the same or out of order such that actually creating the event might have created a singularity in the time-space continuum. When it came to adding items, the testers were confused about the items that were already in the system and had trouble intuiting whether those items were attached to the activity or not. Furthermore, many users thought that adding an item to the system through the "Add Item" window automatically added it to the activity and did not actually select the item when they were returned to the "Add Activity" window.

The "difficult" final task ended up being the one with the least errors. Every tester was able to complete both parts of this task quickly and completely. The majority of the errors came from testers trying to use the calendar as they would use their own calendar, such as drag-and-dropping activities (which isn't supported in our version) or scrolling with the mouse wheel. Another area that was again problematic was the representation of times – showing every hour from 12AM to 12PM and defaulting to 12AM was difficult for testers to use.

Discussion

If we were to base a real user test off of this pilot, the first change we'd make would be rearranging our task hierarchy. Based on the error rate, it seems obvious that the second task is much more difficult than the first, and it makes sense that they should be presented in ascending order of difficulty. Also, it would be helpful to have multiple people watching the tester and merging their observations afterwards, as well as a more standardized idea of what counts as an "error". Furthermore, we now know what

categories of problems our users are likely to experience, and can be on the lookout for them. It's possible that the errors had a higher frequency than we saw because we didn't realize the significance of the tester hovering over a specific part of the interface.

However, there are a lot of changes we could make to the interface based solely on this pilot study, before we started running a larger study. First of all, the "Password" page needs to change. The digits should be separated from the "Enter" and "Clear" buttons – this would both improve the overall aesthetics of the page as well as making the two functional buttons more obvious. Furthermore, each pressed digit should generate only one placeholder digit in the window, and the placeholders should be the same regardless of the number of digits pressed. Finally, some work could be put into improving the look of the page, although that continues to be a low priority for us while we explore the underlying functionality of our interface.

The next thing that needs to be fixed is the way items are represented and added to activities. There should be a clearer difference between adding an item to the system and adding an item to an activity – perhaps a checkbox next to each item indicating whether or not it is attached to the activity. This would separate the two ideas of adding items, because the appearance of the item in the window would indicate that it is part of the system, while the state of the checkbox would indicate whether or not it is attached to the activity. Also, this system would eliminate the confusing method of selecting items by highlighting that is currently implemented. (See Figure 1 in Appendix 3 for an example of our current implementation)

Another change that should be made to the calendar interface is the representation of times and how they're processed. Rather than defaulting to an unreasonable hour – such as 12AM – the time should default to something like 12PM, and there should be some coupling between the start and end times so that the end time defaults to a time a reasonable distance past the start time, such as an hour, and when the start time is changed the end time changes automatically in a similar manner. This would improve not just creating items but editing them as well, and would eliminate (or at least change) a significant number of the problems testers had with this portion of the interface.

The last set of changes would be focused on the calendar image itself, and would make the calendar more responsive to mouse events. Clicking on the calendar to add activities should be supported, and when that is done the start time should be set to the time that was clicked on the calendar. Also, scrolling with the mouse wheel shouldn't require clicking on the calendar to "focus" it – if the calendar is the active window, we can assume that the user is trying to scroll the schedule.

Overall, the pilot test exposed a lot of problems with our interfaces and the assumptions that went into them. Obviously, users want more options on how to complete tasks than the pre-defined paths we were originally thinking of – for example, clicking on the calendar to add an activity seems ridiculously obvious in retrospect, but is something we completely forgot in our development process. Also, while the time of day has very obvious starting and ending points, users are generally only going to utilize a small cluster of those times, and our interface should reflect that as well. This round of testing was very useful for us to point out holes in our development and thinking patterns. If we use this data to correct the functionality problems we discovered, we could easily use the next round of testing to evaluate more of the aesthetic design problems that we've

been avoiding as we tried to get the functionality down while we continue to explore our solutions to the functionality problems we've encountered.

Appendix 1 – Text for User Tasks:

1. You're about to leave the house, and you want to use the system to verify that you have everything with you, although you are sure you have your wallet, notebook, and iPod. Please pretend your name is Andy and your password is '12345'.
2. You just made plans to meet your friends for dinner at Zumeska's Place at 7-9PM. You want to be sure that you remember to bring your dancing shoes, as Zumeska's has an attached dance floor, so you put this event in your calendar.
- 3a. Your friends just called, asking if dinner could be moved to 6-8 instead, as they have to study for midterms. You have to change the event in your schedule to reflect this.
- 3b. Apparently, your friends are huge flakes, and just called to cancel on you. You now have to delete the event from your calendar.

Appendix 2 – Testing Data

Subject 1:

Task 1) Had trouble with password interface - wasn't aware of double clicking

Task 2) Was briefly confused by AM/PM thing

Added item to program without problem, but didn't add item to activity

Thought "Add Item" button added item to activity, not program

Subject 2:

Task 1) Was confused by representation of input in password window

Task 2) Selection of times laggy

Task 3) Wanted to drag/drop the activity in the calendar.

Overall: Wanted to be able to scroll the calendar without using a scroll bar (ie, middle wheel on mouse kind of thing)

Wanted the ending time to change automatically when the starting time was changed.

Subject 3:

Task 1) Asked why display was full of # marks (password obfuscation?)

Task 2) Had difficulty attaching items to activities; wanted an explicit "add to activity" button, or a drag/drop between "available" and "needed" items

Task 3) Found excessive scrolling to select time irritating; suggested that time default to 12:00pm

Overall: On editing existing activity, if items are not selected, they are removed from the activity. Attached items should be selected by default.

Subject 4:

Task 1) Had trouble finding "enter" button on password screen

Task 3) Times should start at a reasonable position, ie whatever earliest visible time on calendar is; took a while to figure out how to select multiple items

Overall: Wanted to be able to add activity by clicking on time slots; wanted an am/pm toggle to set whether times default to am/pm

Subject 5:

- Task 1) asked why display was full of # marks; attempted to hit enter key to enter password; had trouble finding "enter" button on password screen
- Task 2) Used am times instead of pm; made use of tabbing/arrow keys to select time/ didn't attach shoes to activity after adding the item (maybe assumed that creating an item automatically attached it?)
- Task 3) Looked for "Manage Events" button; wanted to delete from "Edit Activity" window

Subject 6:

- Task 1) Tried to use keyboard at password screen.
- Task 2) Tried using add item dialog at first.
Successfully added item, but didn't select it.
Tried to click on calendar to add activity.

Subject 7:

- Task 2) Tried to click on calendar to add activity.
Items shown in add activity dialog are confusing (are they already associated with the activity?).

Subject 8:

- Task 1) Tried to use keyboard at password screen.
Enter button on password screen is hard to see (pressed the home button at first).
- Task 2) am/pm confusion (added as am event first).
Tried using add item dialog after first attempt.
Got several time sanity checks.

Subject 9:

- Task 1) Wanted to go back to previous screen after clicking home (tried pressing browser back button).
- Task 2) Tempted to click on manage items to add activity.
Tried to click on calendar to add activity.
Tried clicking on item several times in add activity dialog (expecting confirmation?).

Appendix 3 – Images

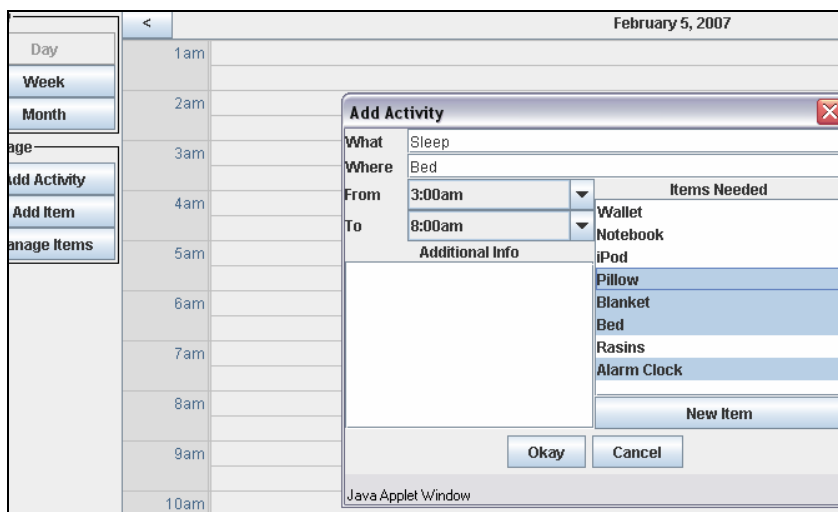


Figure 1 – Example of selecting items to add to activity