

Team: *Don't Forget!*
Andy Hou
Elisabeth K. Olson
Kenneth Kuan
Kevin Chiu
CSE 490F

Assignment #7 – Interactive Prototype
12 February 2007

Problem and Solution

The basic problem continues to be that people are forgetful, and often forget to bring with them the items they need during the day. To solve this, we're trying to develop a computer program that interacts with RFID tagged items and a schedule to ensure that the user has all their necessary items when they leave their home.

Tasks

Easy: Use the scanner interface to ensure that the user has everything they need when they leave for the day.

Medium: Add a new item to the calendar interface.

Hard: Add a new activity to the calendar interface.

Revised Interface

The main change we made to the interface after the Denim testing was to remove Groups altogether. We've started to question the usefulness of groups, and plan to work more on the idea to see if there is some other way to gain the same kind of functionality. We do think there is a benefit to allowing users to group items, especially in situations where the user has a set of items that are required for events that are frequent but not repetitively scheduled.

Removing Groups streamlined our interface for adding items considerably:

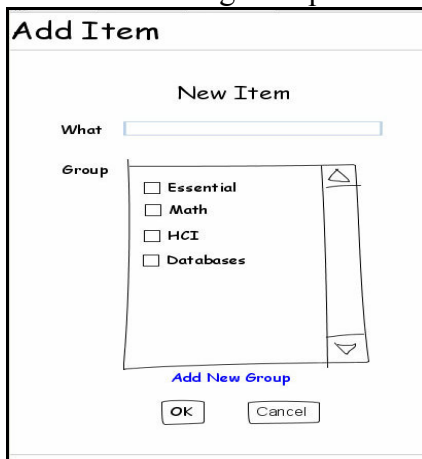


Figure 1. Old "Add Item" window

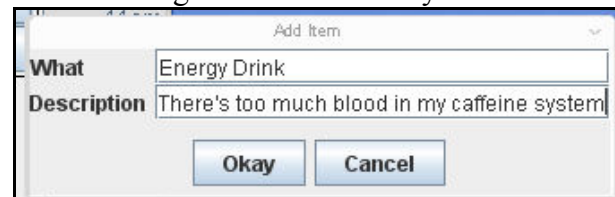


Figure 2. Current "Add Item" window

When we actually add an RFID scanner to the system, we'll have to also figure out how to deal with the errors associated with it. For now, we're assuming that when the user adds items, the reader always finds only the correct tag.

The other major change to our interface is that, in this development environment, we are able to save state. This allowed us to begin to develop the actual functionality we wanted, without creating exponentially large possible paths. We are also able to validate input and give confirmation and error messages.

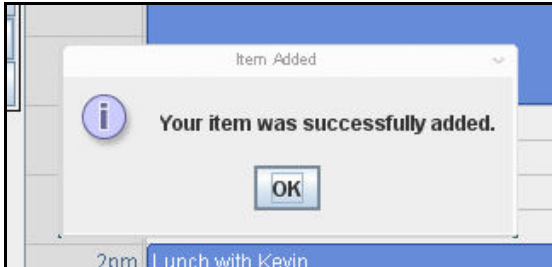


Figure 3. Example confirmation message

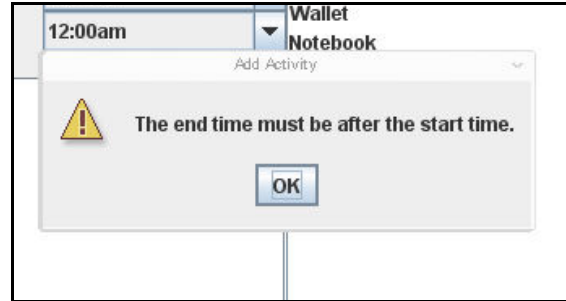


Figure 4. Example error message

We also added the ability to view the details of an activity and delete activities, which aren't necessary for our tasks but are vital to properly functioning scheduling software.

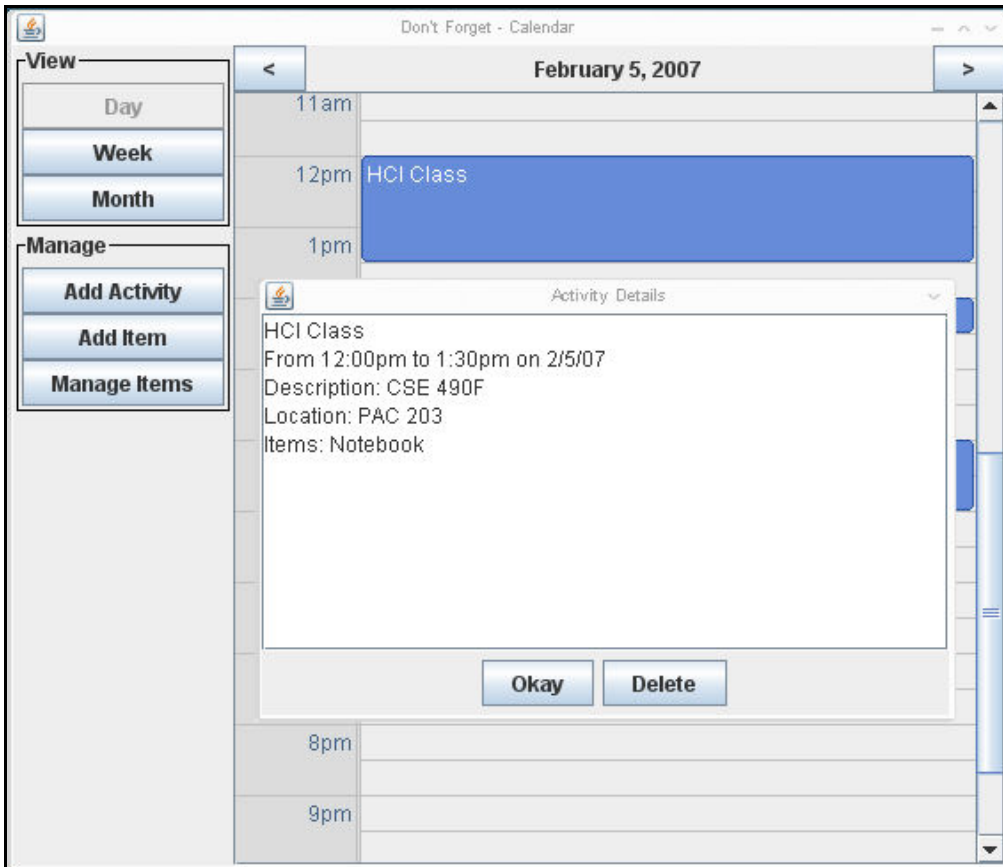


Figure 5. Example of viewing activity details

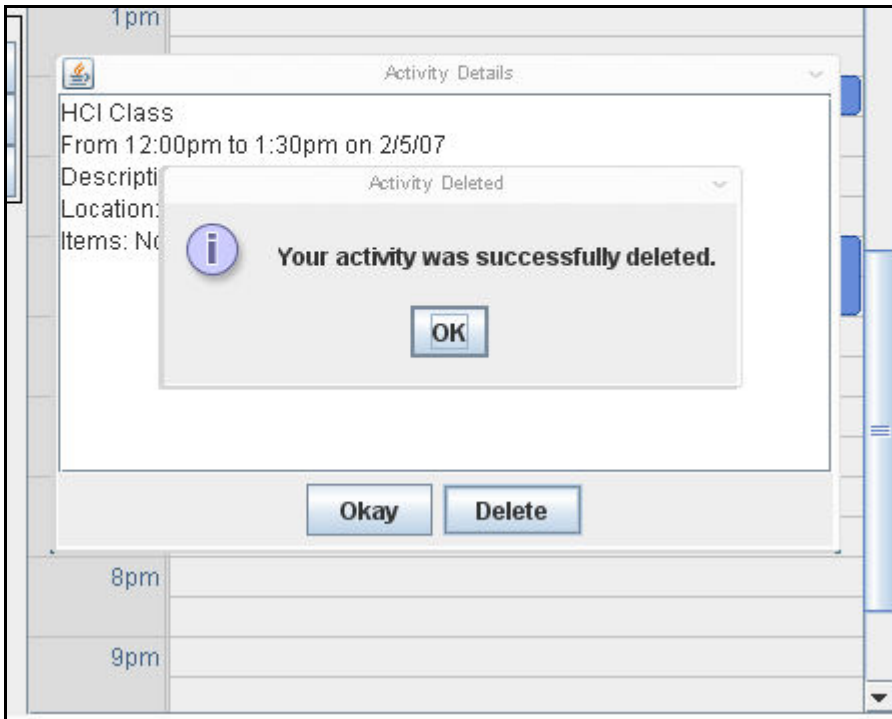


Figure 6. Example of deleting an activity

The paths a user would take to complete the given tasks are virtually the same as they were in the previous versions, except with more responsiveness from the system and more working features.

Task 1

To leave the house, the user would first go to the scanner interface, which is unchanged from previous iterations:

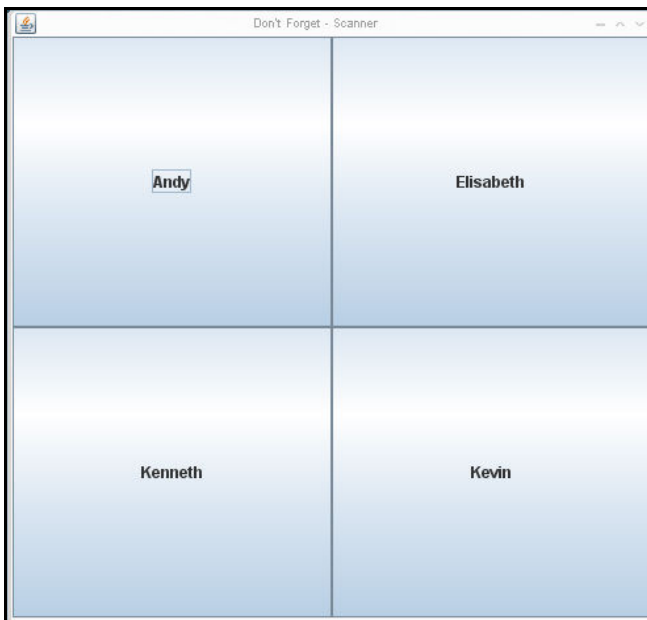


Figure 7. Scanner interface Home screen

From there they would go to the password prompt, which has changed so that now it gives feedback that numbers have been pressed, accepts the valid password and will reject any other passwords:

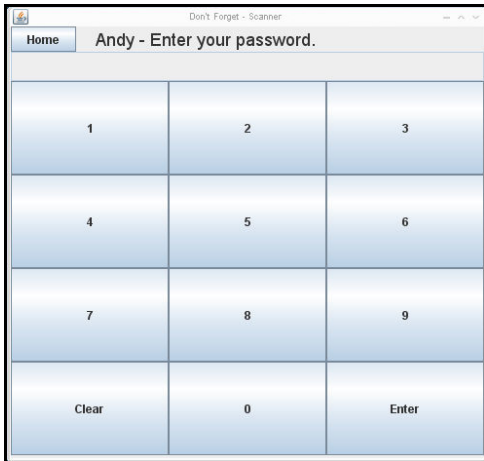


Figure 8. Scanner Password page



Figure 9. Incorrect password error

After entering the correct password, the user will be automatically scanned and taken to the scan page, which tells them whether they are forgetting anything:

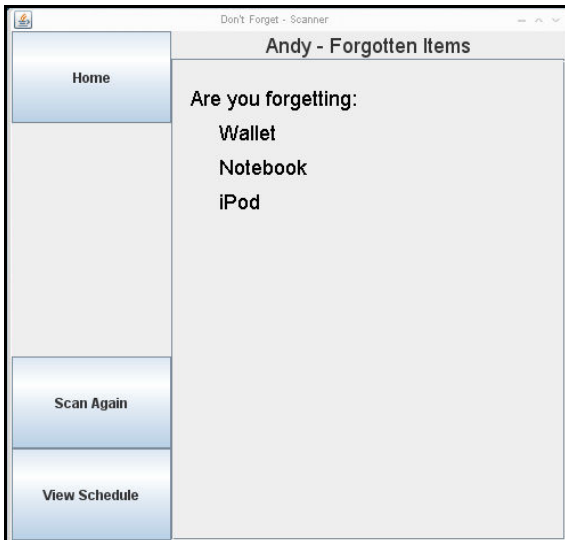


Figure 10. Forgotten Items page with missing items

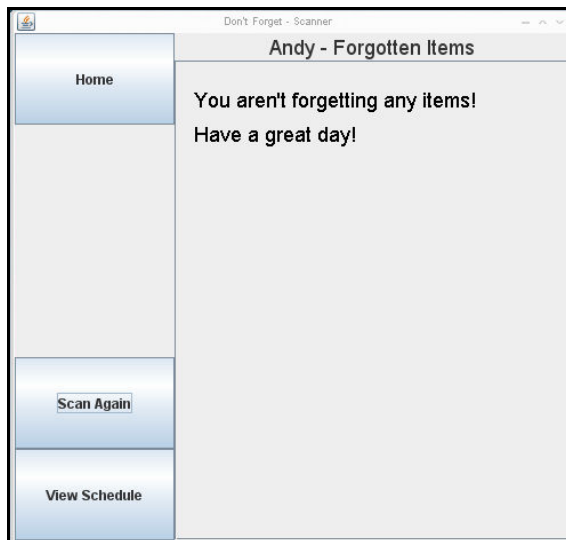


Figure 11. Forgotten Items page without missing items

Task 2

To add an item to the program, the user would start in the calendar interface, clicking on the "Add Item" button:

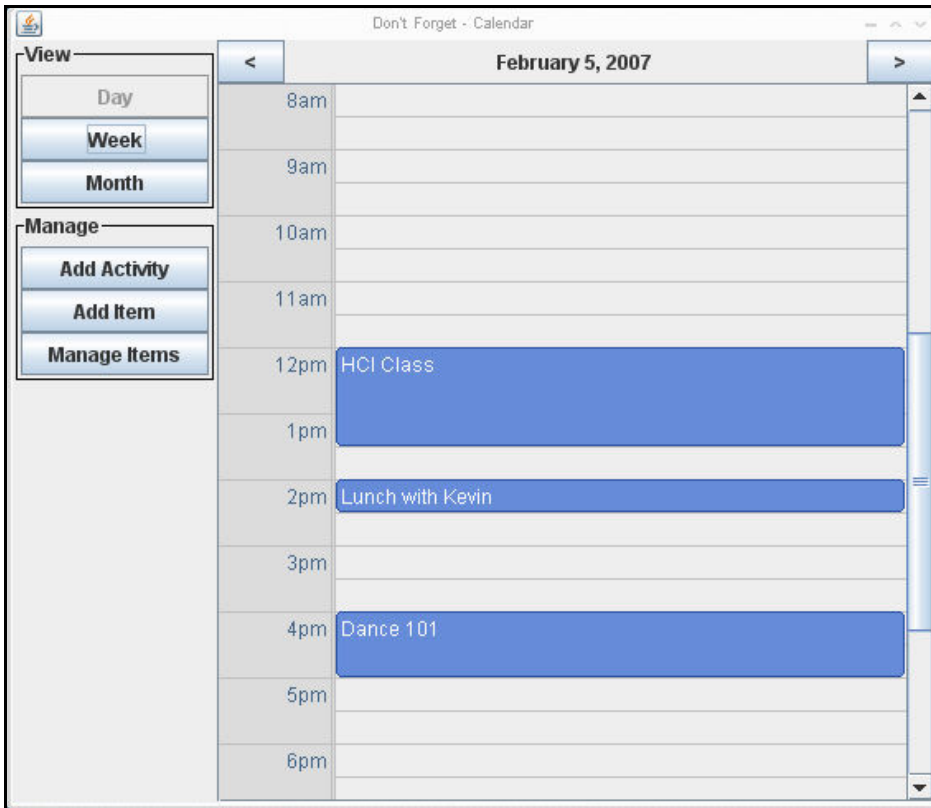


Figure 12. Home screen for Calendar interface

A pop up will then appear, prompting for the name and optional description of the item:

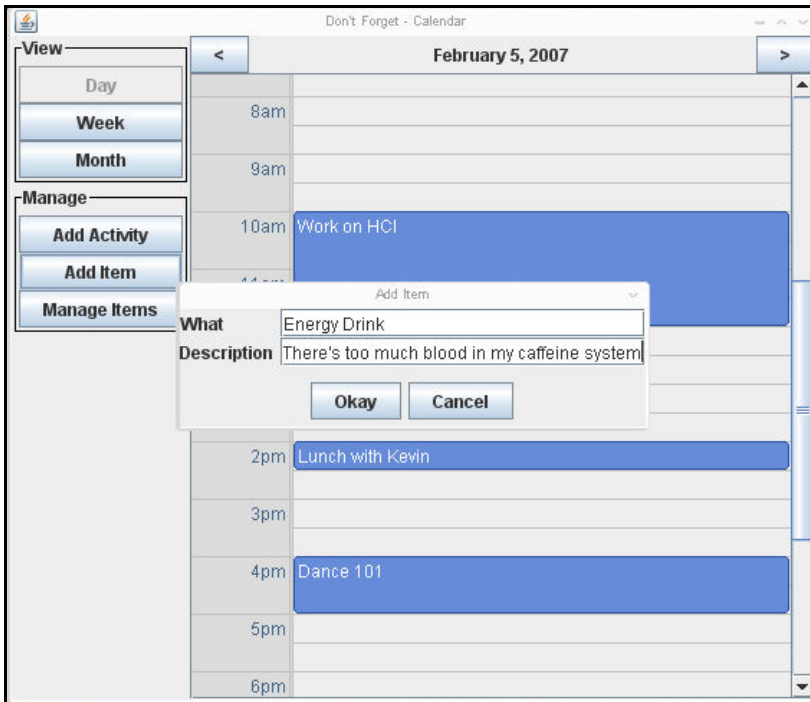


Figure 13. Example of adding an item

Clicking the “Okay” button to add the item will either bring up a confirmation page or an error message, depending on whether the user has entered proper input:

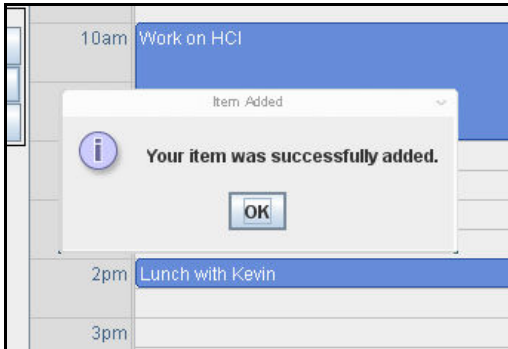


Figure 14. Example of confirmation dialog

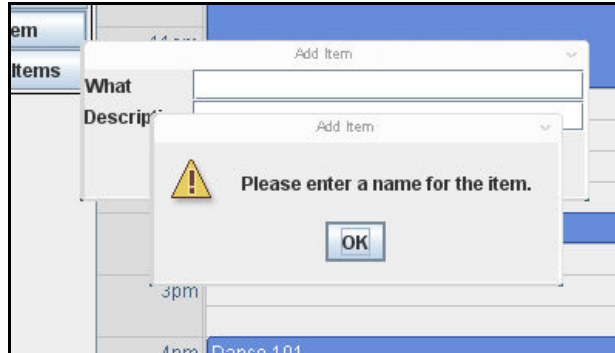


Figure 15. Example of error message

Task 3

To add an activity to the program, the user would click the “Add Activity” button on the calendar interface, which would generate a pop-up window:

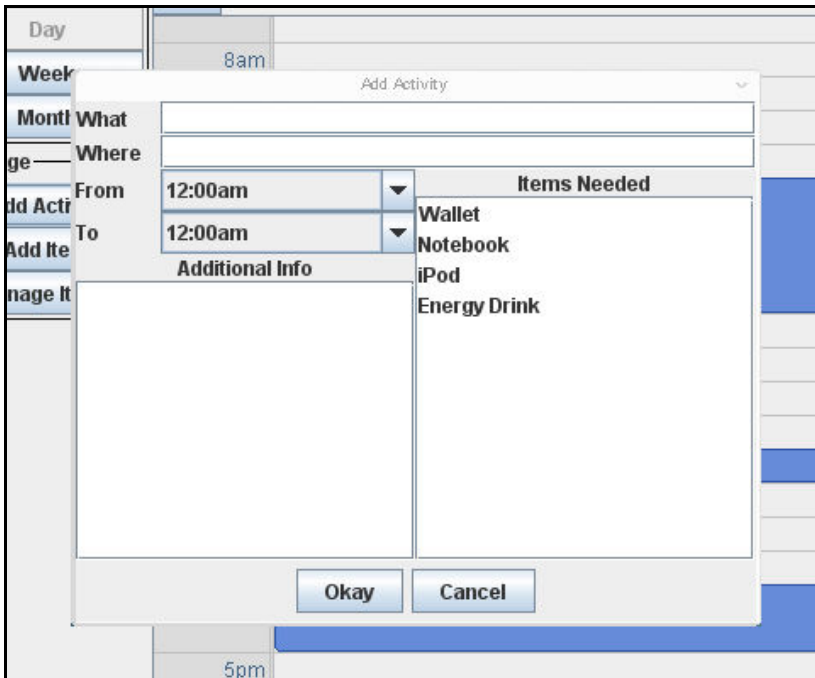


Figure 16. “Add Activity” window in Calendar interface

The user would then type in the information about the event, set the beginning and ending times, and add items to the event by clicking on them. The system indicates that items have been added by highlighting them in the window:

What	Work on HCI	
Where	002	
From	10:00am	Items Needed
To	12:00pm	Wallet
Additional Info		Notebook
All items are selected!		iPod

Figure 17. Example of selecting items when adding activity

Clicking “Okay” brings up either a confirmation box or an error message, depending again on whether the input the user gave was valid.

Prototype Overview

We did our development in Java, and we found it much easier to work with than the prototyping tools we had used earlier. We also think this development environment will make it easier for us to test our interfaces, as we won’t have to force the testers down strict paths but can instead give them broad task descriptions and see how they use the interface to solve them, perhaps in the process discovering new ways of using the interface that we can enhance. Furthermore, many of our features require a remembered state to work correctly, which is a new possibility of our development environment, so our testers will finally be able to actually test our features and see how they are supposed to behave.

At this point, we haven’t added the actual RFID scanner to the interface. We’ve simulated a faulty scanner at the touchscreen interface by randomly deciding whether the scanner recognizes each item the user is supposed to be carrying. Rescanning produces a new set of recognized and unrecognized items, which we think is close to the reality of an RFID scanner (although our scanner as programmed is unacceptably faulty for a real application). We also left the scanner out of the calendar interface entirely in this iteration, but we plan to bring it in for future iterations.

In this iteration, we also removed the idea of Groups entirely. We’re not sure that we’ll add Groups in later, and plan to use our data from this iteration of testing to determine if it’s a concept we should pursue further, and what form they should take.