“Memedicine”: Interactive Prototype

CSE 440 Spring 2011

Mariko Hope - Poster creator + Report writer
Katherine Tung - Hi-fi prototype image creator
Sean Mack - Prototype support + Report writer (some)

Problem & Solution Overview

Taking prescription medication is a routine part of everyday life for nearly half of all Americans. Modern pharmaceuticals are tailored to treat an expansive variety of ailments including potentially life-threatening conditions such as high cholesterol, depression, heart disease, and diabetes. However, even when faced with the dangerous consequences of failing to take these essential medications, patients of all ages can still fall prey to simple forgetfulness and accidentally neglect to follow their regimen correctly. Memedicine liberates patients from the memorization and guesswork of tracking their prescriptions themselves and instead organizes it for them on their mobile devices. Memedicine composites multiple distinct dosing schedules, quantities, and methods into a single, easy-to-follow timeline, and presents patients with just the information they need when it’s time for the next dose. With Memedicine, patients can finally get back to doing what their prescriptions were supposed to afford them all along: living happy, healthy lives.

Tasks

The tasks that we had our users perform were carried over from our paper prototype and refinement. Though the order of the difficulties were not performed from low to high, the order that we designated--add medication, set alarm, take medicine-- reflected the most natural order of tasks that a user would perform when using our application, especially for the first time. Since our users had not been exposed to our interface prior to testing, we hypothesized that this task order would therefore result in less confusion.

Through our task analysis,video prototyping, and paper prototyping, we worked with these three tasks and concluded that they were distinct and would be commonly performed by the great majority of the projected user base of our application. As we solidified the design for our paper prototype, we naturally had to have example medications that we would use for the user testing stage of our design development. We created three fictional medications for paper prototype testing, which carried over for our final interactive prototype: Loratadrine, Eyedroppi, and Nerpazone. Our most recent refined tasks involved the first of these three. Task 1 was to add Loratadrine to Memedicine as a new prescription medication(hard). Task 2 was to set two alarms a day for Loratadrine: 9am and 3pm (the default being 9am and 2pm)(medium). Task 3 was to take all necessary medication when Memedicine’s alarm goes off at 10:40am (easy).

Interface Revision Sketches

A number of minor changes were made to our interfaced based on the feedback that we received from user testing with our paper prototypes. The issues with the lo-fi prototype that users had brought to our attention were that the camera screen was confusing, and that some of the wording on the buttons could be improved upon.

Camera Screen Improvements

On our paper prototype, the second screen of the “add medication” was intended to act as a way of identifying a specific medication’s detailed information by scanning the QR code on its container. Through testing, we discovered that users did not know what to do at this screen, as they could not understand what the image at the top of the screen represented, nor did they know whether or not they had to press a button to continue on to the next screen.

For our interactive prototype, we included text which explained to the user in words what he or she had to do at that screen: “Frame the medicine’s QR code in the viewfinder. It will automatically be detected.” This text has the dual purpose of explaining what the camera screen is for (to scan a QR code from the medicine container), and how to continue to the next screen (automatically detected). In addition, at the bottom of the viewfinder we included text saying “detecting...” and a loading symbol, to show to users that their actions were producing feedback.

Fig 1a.

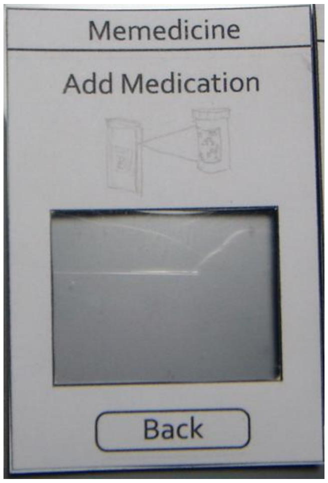
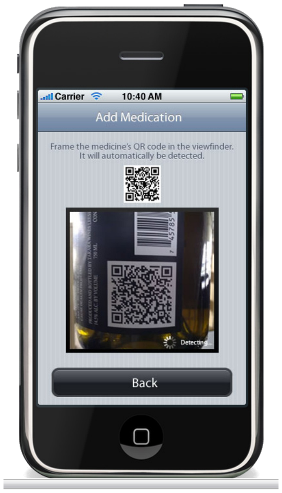


Fig 1b.



Frame the medicine's QR code in the viewfinder. It will automatically be detected.

Fig. 1a: Paper prototype – No clarifying directions or state transparency
Fig. 1b: Interactive prototype – Concise directions and “loading…” text

Miscellaneous Button Improvements

In our paper prototype, users pointed out that on the add medicine screen, we should change the buttons from “Add”, “Retry” and “Back” to “Confirm” and “Back”. The wording issue was mainly to clarify what the users suspected, and did not cause major confusion; however, we nonetheless found it to be easy enough to change, and for a good reason.

In our newest prototype, the add medicine screen wording was changed to “Confirm and add” and “Back”. We phrased the first button in this way to show that this screen validated the previous screen’s scan, while at the same time completing the entire “add new medicine” task.

Fig 2a.

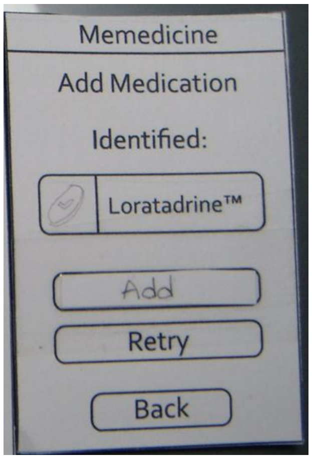


Fig 2b.

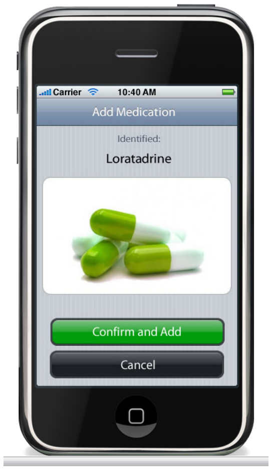


Fig. 2a: Paper prototype – Difference between the three buttons was confusing
Fig. 2b: Interactive prototype – “Confirm and add” and “Cancel” better exemplifies the meaning of the action

Prototype Overview

The interactive prototype consists of *N* different pages for *N* possible states of the prototype. A separate image is prepared for each possible state and embedded in the appropriate page. For example, the Manage Alarms screen for Loratadrine has two corresponding images: one for the second alarm set at 2:00pm and a second image for 3:00pm. HTML <map> tags are used to specify clickable areas within each interface image that link to the page corresponding to the next state (generated automatically with the tools we used).

Overview of implementation

Our interactive prototype was created with nearly the same images and paths as our paper prototype. Due to time limitations, we decided to make ours fixed-path, which eliminated the need for some screens, which would lead to more in-depth information about the other fictional medications that we included in our prototype (Nerpazone, Eyedroppi). For the most part, we found that users liked the images and layout of our application, so we tweaked the wordings in a few locations, but otherwise kept our paper prototype design. In fact, our interactive prototype's main menu looks very similar to one of our initial sketches, back when we did our contextual inquiries. We have a simple but clear implementation using high-quality images of what an actual iPhone application of Memedicine might look like, and laid location-based links on top of the image to direct the user to the next screen.

Scenarios for 3 tasks

For our first scenario, Deanna has just received new medicine her nephew Derek will add it to Memedicine for her. In our second scenario, Deanna asks Derek to set or change the times that the alarms for her medication goes off, to better fit her schedule. In the final scenario, Deanna must now take her medication on her own by navigating the app.

Scenario 1: Derek Navigates "Add New Medicine" screens to add Loratadrine to Memedicine

Deanna has just returned home from the doctor's office. She has been prescribed Loratadrine, which she must take twice a day. Her nephew Derek hears about Memedicine from a coworker and downloads it onto her phone, to remind her to take her new medication. He opens the app, and selects the "Add Medicine" icon from the main screen. Derek places the pill bottle under the camera, and a red box appears superimposed on the camera image on-screen, lining the QR code on the bottle, along with the text "Please hold phone still..." The camera reads his medicine, and the information for Loratadrine appears on screen. There are two buttons on the bottom of the screen, "Yes, this is the correct information" and "No, rescan." Derek presses the first button, and the screen lists the words "Loratadrine added to system." He presses the "OK" button, and is taken back to the menu screen.

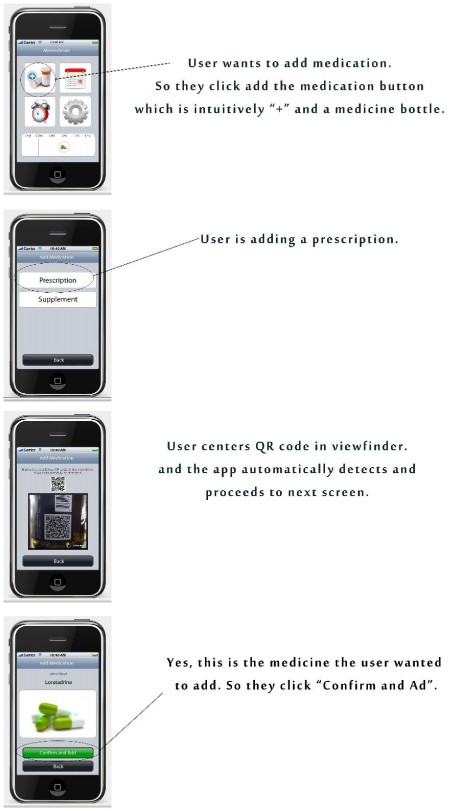
Scenario 2: Derek Navigates "Set Alarms" screens to remind Deanna to take Loratadrine

Derek has already entered the medicine information in Memedicine to help his aunt Deanna remember to take it. The default alarm times for Loratadrine were 9am and 2pm. Now, Deanna tells Derek that it better fits her schedule if she were to take her medicine at 9am and 3pm. To set up the alarm, Derek opens the app and presses the "Schedule" button. He is shown a list of the medicine that he has already entered for Deanna as having to take. He presses the name of the medicine that Deanna would like him to change the times for, "Loratadrine", and is taken to a time setting screen. He acknowledges that the first time is accurate according to Deanna's plan, but wants to change the second time, so he taps on the "2pm" alarm. He drags his finger to set the time to 3pm. He presses the "Save changes" button and is done.

Scenario 3: Deanna Navigates "Take Medicine" screens to take Eyedroppi and Loratadrine

At 3pm, Deanna's phone alarm rings. She picks up her phone and sees text on the screen that says "Eyedroppi" and "Loratadrine". She presses the question mark button on Eyedroppi, which takes her to an instructions screen which directs her how to use the eye drops, and how many drops to do per eye. She presses the button that says "Taken". She is taken back to the first screen that popped up, which now only displays “Loratadrine”. Knowing how to take Loratadrine, she simply presses the “Taken” button on that initial screen, and is then taken to a completion page saying “All medication taken.” She presses the “Back” button and is taken back to the main menu.

Storyboards of scenarios



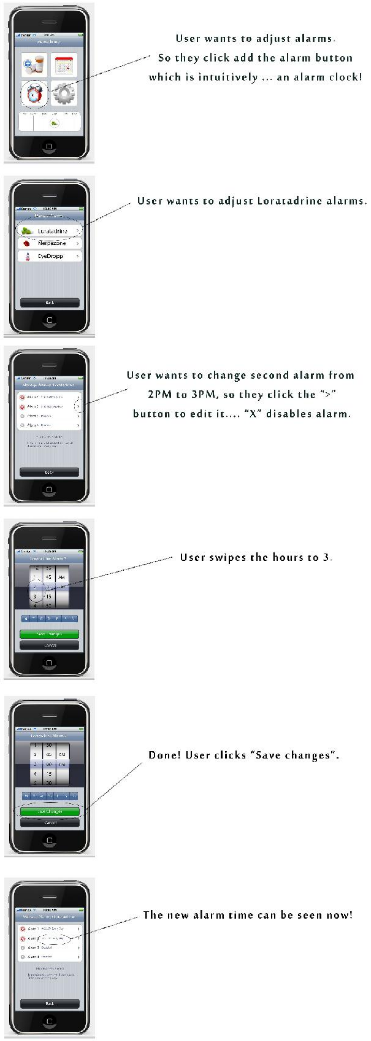
User wants to add medication. So they click add the medication button which is intuitively “+” and a medicine bottle.

User is adding a prescription.

User centers QR code in viewfinder. and the app automatically detects and proceeds to next screen.

Yes, this is the medicine the user wanted to add. So they click “Confirm and Ad”.

Storyboard 1



User wants to adjust alarms. So they click add the alarm button which is intuitively ... an alarm clock!

User wants to adjust Loratadrine alarms.

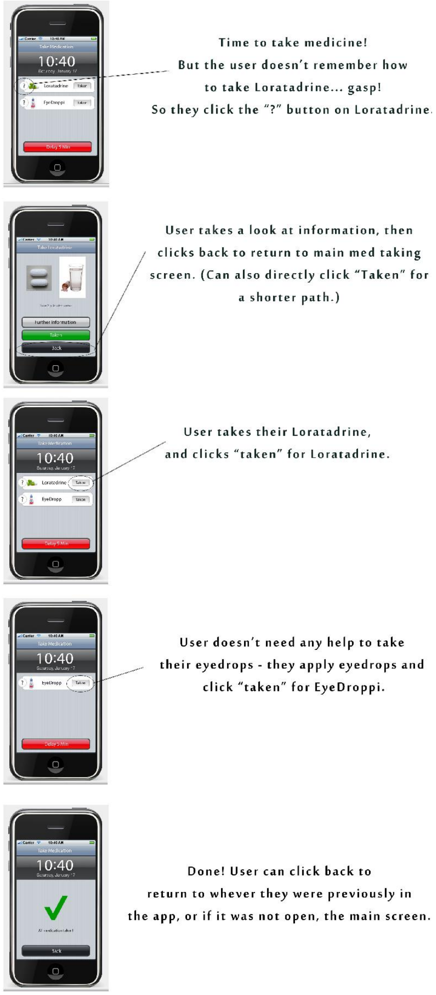
User wants to change second alarm from 2PM to 3PM, so they click the “>” button to edit it.... “X” disables alarm.

User swipes the hours to 3.

Done! User clicks “Save changes”.

The new alarm time can be seen now!

Storyboard 2



Time to take medicine!
But the user doesn't remember how
to take Loratadrine... gasp!
So they click the "?" button on Loratadrine.

User takes a look at information, then
clicks back to return to main med taking
screen. (Can also directly click "Taken" for
a shorter path.)

User takes their Loratadrine,
and clicks "taken" for Loratadrine.

User doesn't need any help to take
their eyedrops - they apply eyedrops and
click "taken" for EyeDroppi.

Done! User can click back to
return to whever they were previously in
the app, or if it was not open, the main screen.

Storyboard 3

Description of tools used

For our interactive prototype, we used Adobe Fireworks, Microsoft Expression Web, and Adobe Photoshop. We started out with a Fireworks file we found online that provided a basic iPhone UI template. We made mockups with this, then inserted the images of our application into the iPhone base in Photoshop. Then, we added them to Expression Web, where we linked the images. Finally, we used WinSCP to make them live.

How the tools helped

The tools helped us to make sleek, beautiful graphics that would be very close to what the UI for an iPhone version of our application would look like. Specifically, Fireworks made it easier to manipulate the base template, since it works with vectors. Photoshop was useful for editing some of the stock photos we used for the interface (i.e. images of pills, alarm clocks), and placing the UI pieces into the fake iPhone. Expression Web made linking easier since we could drag hotspots right onto images rather than defining them through code.

How the tools did not help

There was a lot of confusion at the beginning caused by the fact that we had not worked with Fireworks before. It behaved differently than the usual editing software that we used (Photoshop), so we made a lot of mistakes at first. We essentially learned it on the fly, which wasted time that perhaps may have been better spent on other parts of the project.

What was left out of the prototype and why

Since the nature of our application is to provide easy, step-by-step directions to users on how to take their medication, the final implemented product would be more or less fixed-path. Each subsection of our prototype's menu follows a linear path, so not many of the buttons and screens were left unimplemented.

The one screen that we knew would be difficult to implement was the second screen in the "add medication" task, in which the user would use the built-in camera on his or her mobile device to take a picture of a QR code on the medicine or supplement container. Since our interactive prototype was to be posted online and would be most likely accessed via non-mobile devices, the most intuitive way to implement this would be to use a webcam. However, our team decided that this was too much of a hassle, and it would be simpler to just substitute this process by having a static image of a QR code appear on screen, and then waiting for a couple seconds before the website would automatically forward to the next screen.

Also, we did not include the details of our three fictional medications. In our previous prototypes, we had an "Information" screen which would list the details of a medication including the name of the doctor who prescribed it, and the dosage, method of intake, side-effects, and precautions for the medicine itself. We felt that it was unnecessary to make up details for our sample medicines, and that the "Information" button would speak for itself.

Sketches of unimplemented portions of the interface

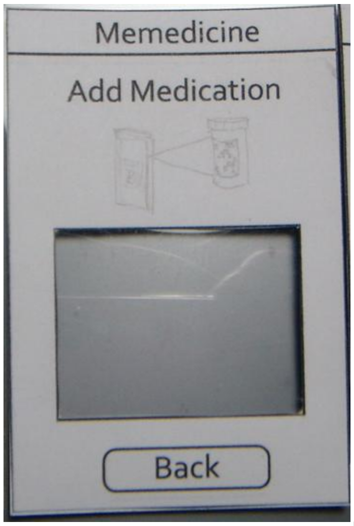
Fig 3.


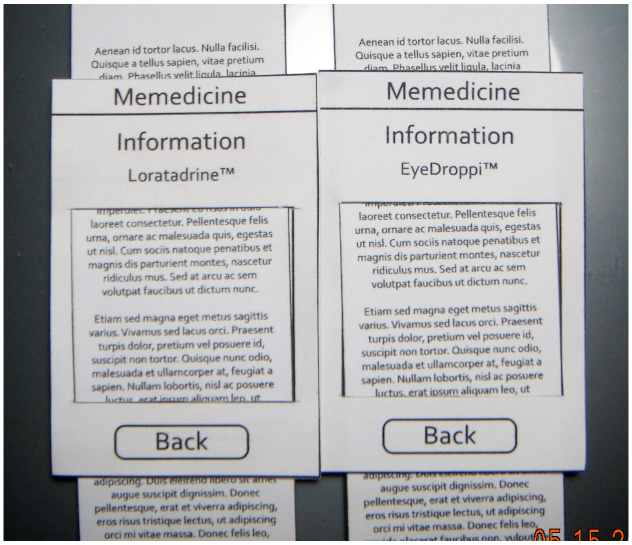
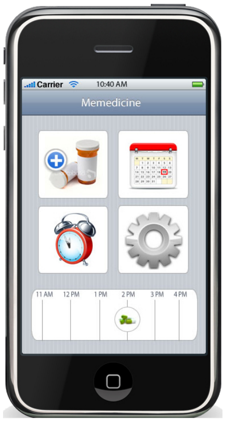
Fig 4.


Fig 3: Unimplemented “camera” screen which would show the viewfinder of the mobile device’s camera
Fig 4: Detailed information that would be provided for each medicine by the scanned QR code

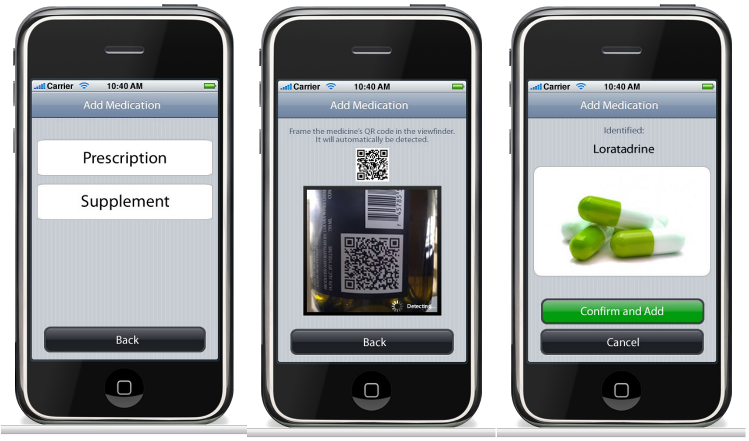
Wizard of Oz Techniques

Our prototype was created using images and transparent links, so there are no simulations of real-time system responses. The application is designed to prevent error, as there are no screens at which the user must manually input information. There was no user input to discern, other than the scanning of the QR code, which is much better regulated (issued by pharmacist/doctor) than what would be typed in directly by a patient.

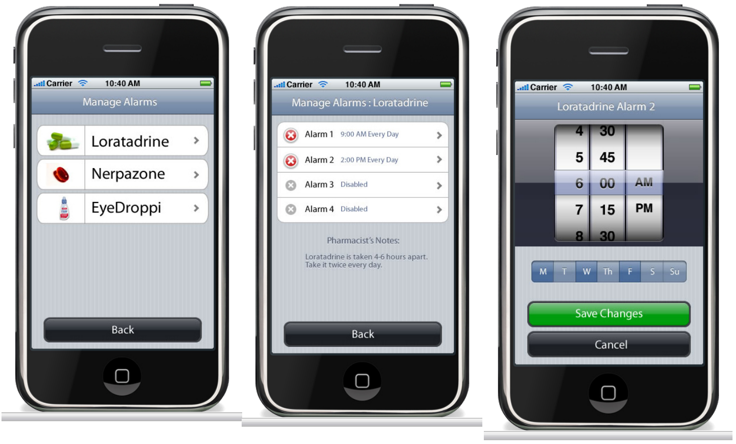
Prototype Screenshots



Add meds->



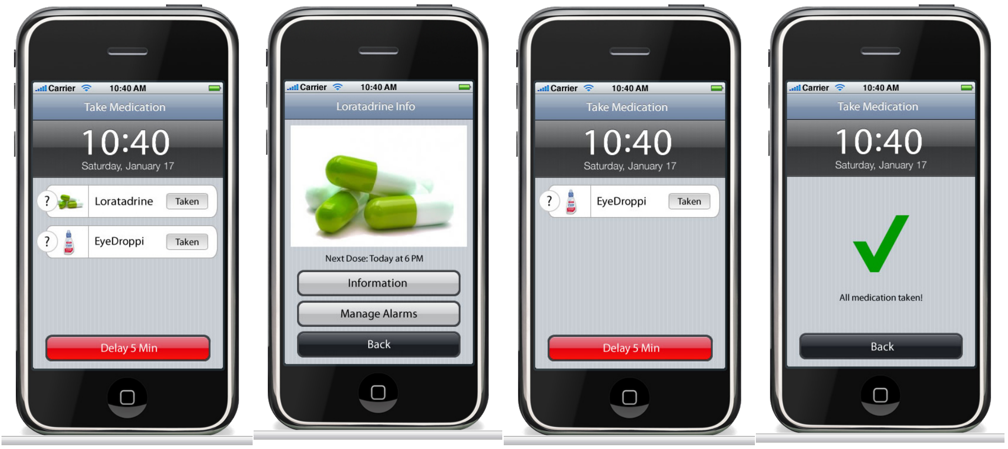
Set alarms->



Pharmacist's Notes:

Loratadrine is taken 4-6 hours apart. Take it twice every day.

Take meds->



Next Dose: Today at 6 PM

All medication taken!