

Twinlist: Overview and general implementation (prepared primarily for developers)

Tiffany Chao
Johnny Wu
Catherine Plaisant (plaisant@cs.umd.edu)
Ben Shneiderman (ben@cs.umd.edu)

Human-Computer Interaction Lab
University of Maryland, College Park, MD

February 18th, 2014 (Updated from the January 2012 version)

This document complements the Twinlist’s video demonstrations see the main Twinlist webpage: www.cs.umd.edu/hcil/sharp/twinlist

1 Introduction

Twinlist was designed as an **inspirational standalone prototype**. If adopted, it would become part of the user interface of a larger system (e.g. a CPOE or EHR system). Here we will refer to that interface as the “outer” interface.

The entire process of medication reconciliation is a complex collaborative process in which many things can go wrong: the patients may not recall what medications they are taking (or may even be unconscious); the information may not be recorded properly and include a lot of unreported uncertainty (e.g. about dosage); the record of past medication orders may not be complete or not even be accessible at all; not all sources of medication orders for the patient may be known (e.g. they may have consulted a specialist on their own), etc. Eventually the physician is presented with multiple lists of medications from different sources that must be reconciled into a single medication list. **The focus of Twinlist is on this last step of the process: facilitating the task of reviewing and sorting the medications that need to be continued (i.e. kept) from those that need to be stopped (i.e. rejected).**

The configuration page the prototype has menus to give access to several versions (variant designs of Section 5) and test cases (i.e. patient scenarios), see Figure 1. The description of

specific configurations can be embedded in a URL e.g.

http://www.cs.umd.edu/hcil/sharp/twinlist/dev/indev/ipad/index.html?case= _DATASET_APPEN DECTOMY_ &version= _VERSION_FULL_ &animate= _AUTO ANIMATE_ON_



Figure 1: *The prototype configuration page includes menus to select a test case and a version, and lists the basic controls (for desktop and tablet devices).*

Next to the menu of cases is a link to narratives for selected cases (the ones that were used in the evaluation).

2 Assumptions about the data

The Twinlist prototype includes all the user interaction code, but is deliberately independent of the similarity pre-processing algorithm. Our prototype assumes that the similarity information will be generated by a separate algorithm, and that the EHR itself is connected to a drug database.

Such an algorithm has been developed by our SHARPC Project colleagues at the University of Texas in Houston. As of February 2014 the algorithm is available at <https://github.com/jherskovic/MedRec> but PLEASE Contact: Jorge R. Herskovic for possible updates, and read their publication “Automated medication reconciliation and complexity of care transitions”, Bozzo Silva, Bernstam, Markowitz, Johnson, Zhang and Herskovic, AMIA 2011- <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3243161/> . *Note: if you cannot use an existing similarity algorithm you could still use Twinlist by first performing a straight string comparison between drugs. Exact matches would be “identical”, drug name match only would be classified as a “similarity”, and all other drugs would appear as “unique”. This would miss a lot of similarity information, but may be a good way to start implementing Twinlist. When evaluating similarity algorithms you should see if they are able to identify different types of similarities and if they are able to deal with compound drugs (see example of Zestoretic in Figure 2)*

Medications are defined as a set of up to six attributes: name, route, frequency, dosage, drug

class, and diagnosis. Only name is required and is composed of a recorded name (i.e. the default to be displayed, which corresponds to what was actually dispensed), a brand name and a generic name.

The prototype uses an ad-hoc database containing a manually constructed set of test cases with the matching similarity information. This information includes what items are identical to one another as well as (for similar drugs) what attributes are different. Drug-drug similarities will more commonly be between pairs of drugs in separate lists, but Twinlist's layout could also reveal many-to-many similarities, including similarities between drugs in the same list (e.g. if a drug has been ordered twice).

The time-stamp of the last order was not included in the prototype. It was assumed that users would only be asked to reconcile lists of drugs that are current, while a separate interface would be provided to review the medication history with access to both current and past drugs. Still, exact order time information might be useful in some cases (e.g. for antibiotics or steroids). It could be added as an additional attribute, and we would recommend displaying it only in the details area (at the bottom of the screen) so that it doesn't take too much space.

3 Implementation

Twinlist is written in HTML5, CSS3, and JavaScript (using [jQuery](#)): HTML5 for content, CSS3 for polish, and JavaScript for everything else. All datasets are currently hard-coded for demo performance and portability, though it would be trivial to fetch them from an external source if necessary.

As of 2013 the current version performs best in Google Chrome 14+ and runs nearly as well in Mozilla Firefox 5+ and Internet Explorer 9+. Internet Explorer 8 and earlier are not supported. However, the entire application can be downgraded to HTML4 and CSS2 should browser compatibility be at a premium. Early on during the project we produced a downgraded version for IE8 but this required removing several visual design features - which we do not recommend doing.

The code is available under the MIT license (see http://en.wikipedia.org/wiki/BSD_licenses). jQuery also uses the MIT license.

The full copyright text that we use is included at the end of this document.

The codebase (excluding `.min.js` libraries) consists of roughly ,000 lines across six files:

- `twinlist.html` is the base of the application;
- `5col.css` controls appearance and mouseover animations;
- `init.js` bootstraps initialization;

- `utils.js` handles client-side storage (either web storage or cookies, depending on the browser);
- `model.js` encapsulates all view-independent data (including sort order, group order, filter exclusion rules, and so on);
- `controller.js` manages all view-dependent behavior, coordinating all responses to user action (notably `redraw()`).

4 Description of the default Twinlist interface

Twinlist's interface consists of four parts (see Figure 2): the header (top), the list viewer (center), item detail (bottom), and additional controls (right). The header is used as the main navigation. The list viewer is used to interactively make decisions on what medications to keep or reject. The item detail is used to display all the attribute information for medications as needed. Additional controls on the right include conceptual features and navigation controls to return to the introductory page. Fairly large font is used throughout the interface for easy demonstration.

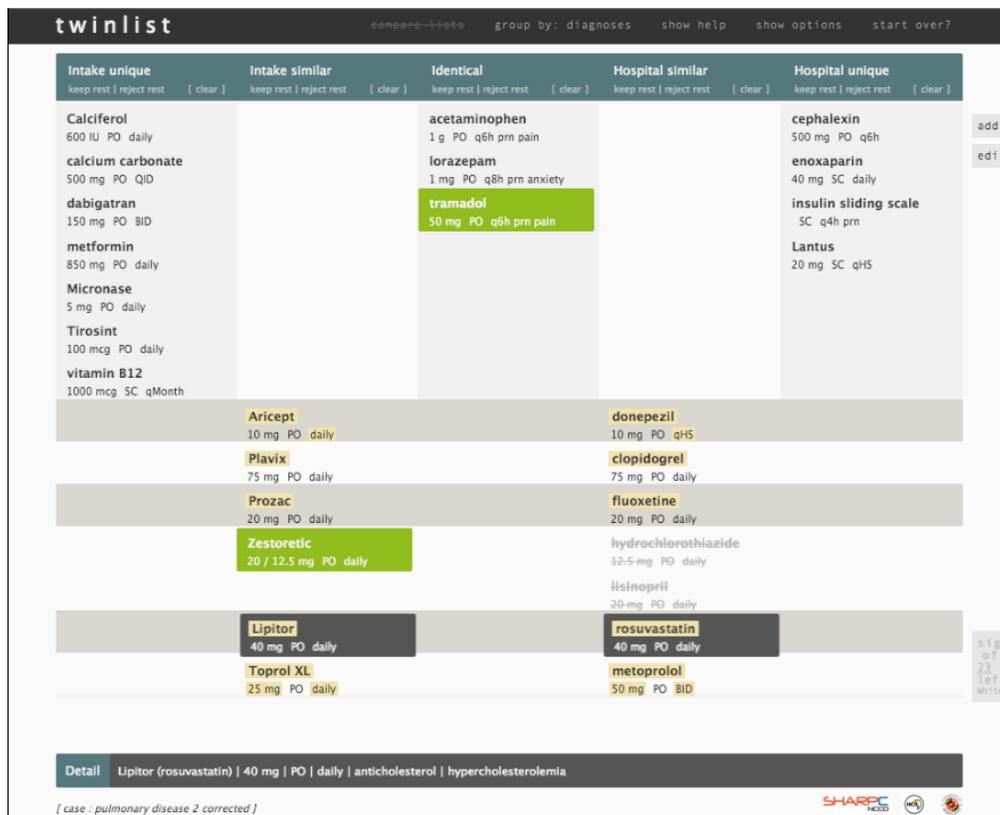


Figure 2. *Tramadol and Zestoretic have already been selected (and changed to green). The Zestoretic is a compound drug so both lisinopril and hydrochlorothiazide have been automatically marked as rejected. Here the pointer is hovering over Lipitor, so the details for lipitor are shown in the detail area at the bottom (here we see first “anticholesterol” for the drug class, and “hypercholesterolemia” for the diagnosis that has been associated with the Lipitor*

prescription. Rosuvastatin is also highlighted because it is the a generic version of Lipitor.

4.1 Header

The header at the top of the page includes the following controls:

- compare lists - used to initiate the staged animation sequence (see below under “Staged Animation”) if in the starting 2 column layout
- group by: used to quickly apply grouping for easy demonstration (currently cycles between grouping by drug class and no grouping) (see below under “Grouping by attribute”)
- show help: used to display a help dialog in case users forget the controls for Twinlist
- start over?: used to reset the drug layout to the original 2 column view
- show options: reveal advanced features and controls

Various advanced features are hidden from the initial page via the options panel (see Figure 2). It includes controls for:

- *display* - to switch between which name is displayed
- *groupby* - to change the grouping of drugs (see section 4.6)
- *item details* - whether to show a few attributes under each name or only show names
- *animation speed* - how quickly items move during the staged animation
- *filtering* - text-based item filtering by name (less useful)
- *steps to jump to* - allows for examining the intermediate states of the staged animation (mostly useful for demonstration, or to set default values)
- *after action behavior* - whether to gray out or remove items that have been acted upon (the gray out mode is strongly recommended so this option is less important)
- *showing the patient tab* - whether to display a patient discharge summary page or not.
Note: This patient tab allows the clinician to review and or annotate the list to be given to the patient (our patient tab is only a proof-of-concept at this point, and would need to be refined).

Most likely developers will have to modify the header to fit the “outer interface”.

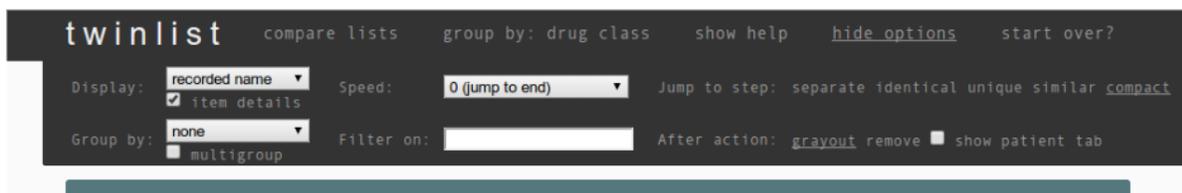


Figure 3. The expanded header (as seen after clicking “show options” in Figure 2). Clicking “hide options” will return the interface to Figure 2.

4.2 List viewer and item detail

The layout and behavior of the list viewer depends on which version is used. Five different versions (layouts and interactions) were designed and implemented. The recommended layout is the 5-column layout. Section 4 describes the alternative layouts.

However, between all versions, many visual elements and behavior are maintained. By default the information is always alphabetically organized with Intake medications on the left and Hospital medications on the right.

Access to details: On mouseover, the hovered over item is highlighted with a dark-gray background and additional information appears in the detail area at the bottom (which also gains a dark-gray background).

We found that tooltips were not practical to display additional drug information due to the possibility that this information could become very long (which would result in occlusion). Additional features are also present on mouseover but these vary by interface version.

Making selections: The general process to make decisions on medications remains consistent between all versions.

To indicate that a medication should be kept i.e. continued, users simply click on the drug, and the background of the item turns green.

To indicate that the medication should be rejected (i.e. discontinued) users can either right-click on the item or use two left clicks. The deselected item is then grayed-out with strikethrough.

Clicking multiple times on an drug will cycle through all the states: undecided, accepted or rejected (e.g. a click on a rejected item changes it to undecided).

In addition, controls are available at the top of each column to keep or reject all the remaining undecided items in a column (see buttons labeled "keep rest" and "reject rest" under the "Intake" and "Hospital" column titles in Figure 2). Practically those buttons were rarely used, probably because it is was easier to make decisions while going down the lists. The selection/deselection of individual drugs is easy because the buttons are large. A "clear" button also resets items to undecided (important for demonstration, probably not for actual use).

Once all items have been decided upon (i.e. either kept or rejected), users can click on "sign off" at the bottom right of the screen to complete the reconciliation The sign-off button remains inactive and grayed-out until all decisions have been made (e.g. in Figure 2 we see that there are still 23 medications left to act upon). We found that this check was important to avoid errors that occur when users forget to indicate their choice on some drugs. By making drugs easy to select/deselect (here we made the whole drug label selectable instead of using small checkboxes next to the drugs - as it is often done) users can specify their choices as they review the list.

Alternate selection method for tablets (e.g. iPad):

- tap on a drug to highlight and see detail information
- left-swipe to keep the drug

- right-swipe to reject it
- swipe again to rotate through all 3 states (undecided, selected or unselected)

4.3 The default 5 columns interface of Twinlist

By default the 5-column version starts with the two columns of medications (Intake and Hospital) and a multi-stage animation reorganizes the display into a 5-column layout. See below for more details on the staged animation.

When the cursor hovers an item the items that are similar are also highlighted. They are spatially shifted and gain dark-gray backgrounds if undecided (see Lipitor and rosuvastatin in Figure 2). When a medication is kept, the similar medications are automatically rejected (see Zestoretic and lisinopril in Figure 2).

During our user testing participants clearly indicated that they did not want the identical drugs to be automatically selected, instead they want to confirm their choice for each drug one by one as they review them (which is easy here because the buttons are large).

Note that when two medications are identical they appear only once on the screen in the middle column, so if users decide to keep that medication we make sure in the backend that only one copy of the medication is kept (we keep the home copy and reject the hospital version).

4.4 Additional controls (sign off, add, edit)

On the right side of the interface are controls to add or edit medications, sign off on decisions and return to the prototype access page. Clicking “add” opens a separate screen to allow users to add new medications to the Hospital list. “Edit” allows users to edit the last medication that was clicked on. The Add and Edit pages are simply placeholders for such functions purely to show how users could access them. When such edits are made we indicate them by adding an asterisk next to the name of the drug next to the drug that was added or modified.

The Sign-Off button remains inactive until all drugs have acted upon. Until then the button is grayed out and the number of drugs that remain to be reviewed is provided. This behavior helps eliminate errors where users do not review the entire list (which may happen when the lists are long and scrolling is needed).

See section 5.4 for additional discussion of the Sign-off method.

4.5 Staged Animation

By default the animation is ON. At the start, we animate the drugs from their initial 2 column layout to the 5 column layout to reveal the similarities and differences between the lists. We

broke a complex overall transition into separate steps, allocating approximately 1 second per step, with up to 750 ms to properly punctuate separate steps.

The animation is comprised of the following stages (see Figure 4-A to 4-E):

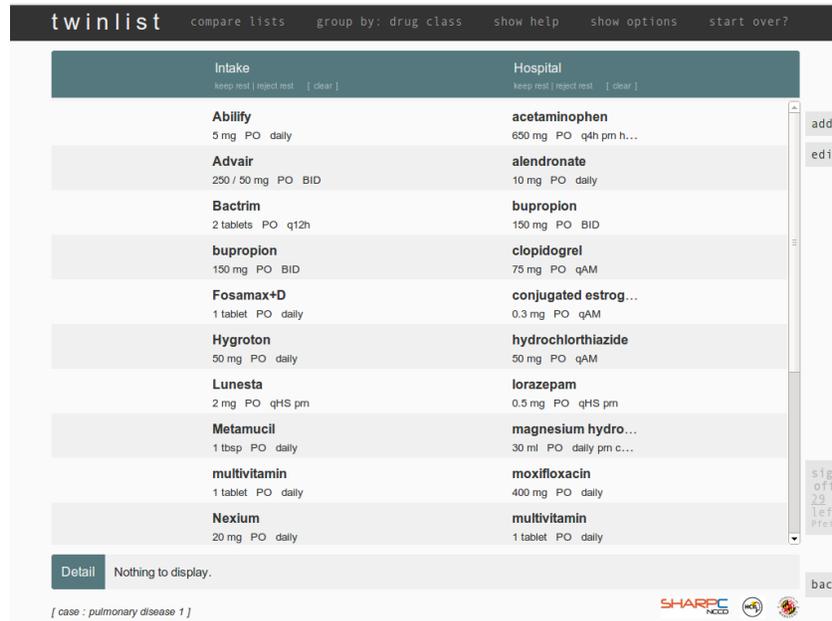


Figure 4-A **Initial stage:** the 2 lists are side by side, arranged in alphabetical order.

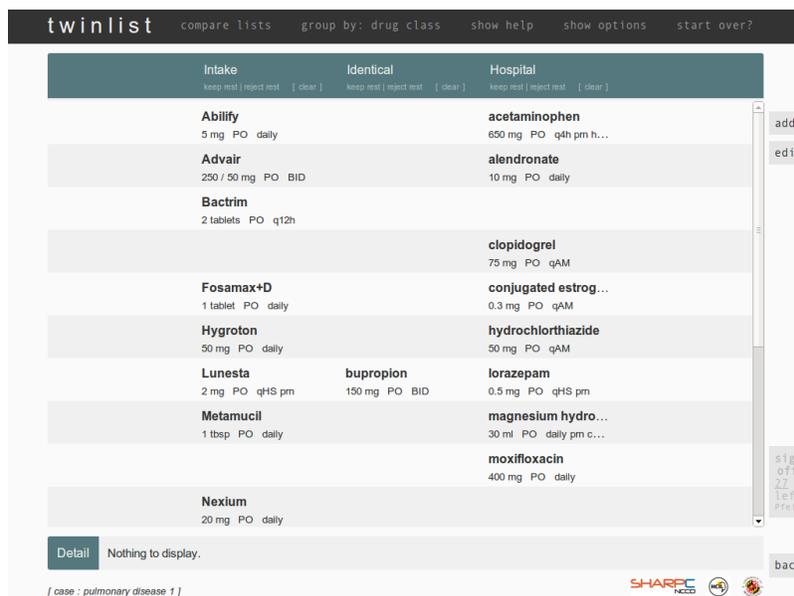


Figure 4-B: **Identical items converge.** Identical items merge in the center column (“Identical” column). Each set waits for its turn: items from the first set fly to the center and converge, then items from the second set fly to the center and converge, and so on. Note that the position of the item from the first list takes precedence.

Note: Animating identical sets one set at a time helps users track individual items.

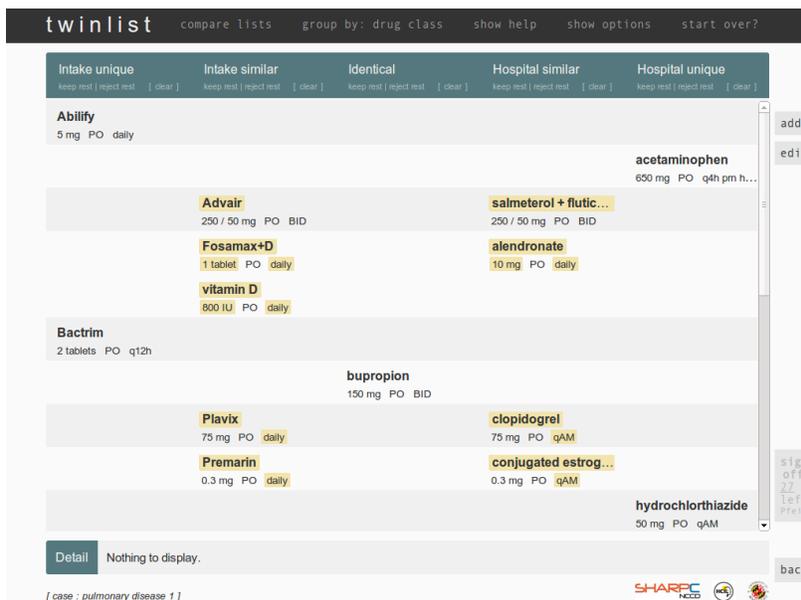


Figure 4-D: **Similar items are aligned** and gain yellow highlights to indicate differences between similar medications. Only similar items move in this stage. Golden-yellow emphasizes the parts that differ, popping a bit from the gray and white background. Notice that only similar items share the same row. That row may be enlarged to accommodate more items
 Note: Highlighting helps. Sometime some attributes may be missing and a block of a different color or a “??” may be used instead to reveal what is missing; use a consistent set of highlighting rules for similar items. Choose an appropriate color: we chose a softer golden-yellow to contrast with our greens, whites, and grays.

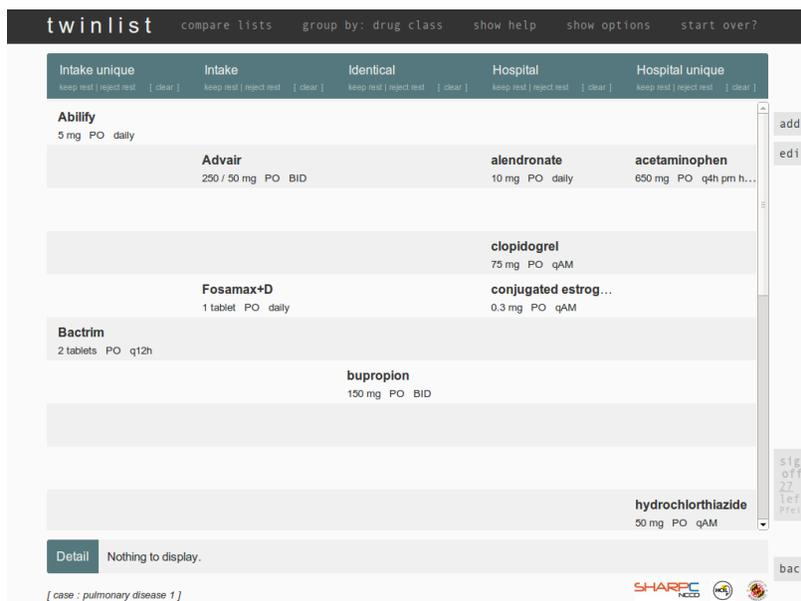


Figure 4-C: **Unique items are pushed aside on their respective side.** This transition is a two-part animation: first all items that are unique to Intake fly to the left, then all items that are unique to Hospital fly to the right. They are moved to their final positions in the precompact

stage, so it is possible for unique items to jump towards the bottom.

Note: The message here is: “these things are different”. Animating one side at a time emphasizes this separation; individual item tracking is less important for this phase.

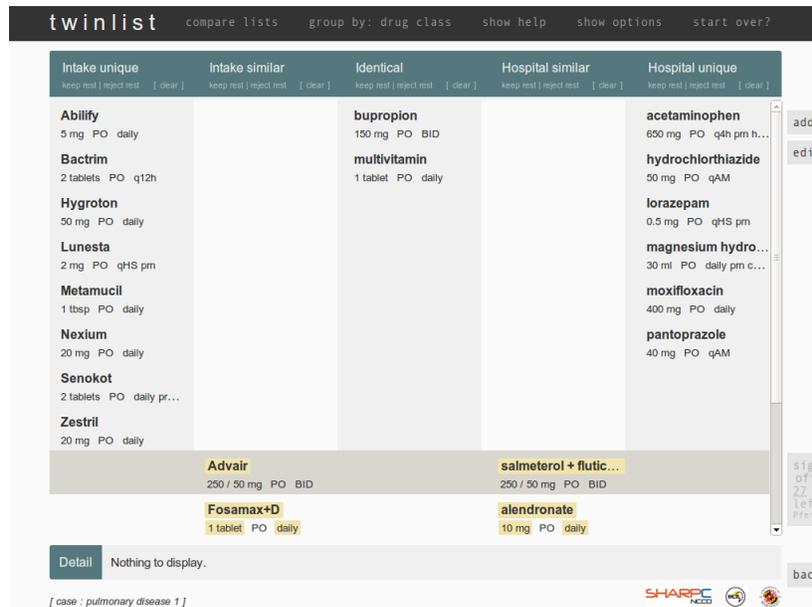


Figure 4-E: **Compact to save vertical space.** All identical and unique items are pushed to the top; similar items grouped at the bottom. Note: This step may not always be helpful. If all items are either “identical” or “similar”, for instance, there is nothing to compact.

Each animation takes 800 ms, follows an ease-in, ease-out trajectory, and is punctuated with a 600 ms pause. The base duration deviates slightly from the 1 second norm [Robertson et al. 2002], shortened to accommodate the 600 ms necessary to properly separate different stages. The ease-in, ease-out pacing -- based on traditional principles of animation [Lasseter 1987] -- provides aesthetic fluidity and eases object tracking [Dragicevic et al. 2011]. We use a simple cubic easing function.

Note: Larger screen resolutions may stray slightly above 1 second for smoother interpolation; smaller screens could accommodate a somewhat shorter duration, since there is less distance to cover.

Per-item animations are dataset-dependent and necessarily custom-tailored.

The current solution opts for precise control at the cost of added complexity, individually positioning free-moving boxes on top of an empty table. A set of “key frames”, expressed as a series of row and column offsets, encode when and where list items move. These key frames are calculated once on page load, and are recalculated only as needed (e.g., when the window is resized).

Note: An earlier version animated table cells directly, but suffered from unintuitive hover behavior: cells that had moved to new locations still raised mouseover and mouseout events when users hovered over their original, HTML-determined positions.

4.6 Option: Grouping by attribute

The 5-column allows for grouping of the drugs by any attribute. When enabled, this will subdivide the list viewer into large horizontal regions, each of which contains only medications that share the same value for the grouped attribute (see Figure 5). Furthermore, it is possible to enable multigrouping (i.e. drugs can belong to multiple groups) which creates shadow duplications of items that belong to multiple groups (see Figure 6). The organization of medications within each group is the same as that without group-by but only 3 stages of the animation are permitted: the initial 2-column layout, the 5-column layout before compaction, and the 5-column layout after compaction. The intermediate stages are not executed. One issue with grouping is that the vertical space needed increases as the medications move from the 2-column layout to the 5-column non-compact layout. This is due to how Twinlist enforces that items are only aligned if they are similar.

Keyboard Shortcuts: users are likely to want to switch back and forth between grouping and non grouping, so providing shortcut keyboard is very useful.

- Type D for group by Diagnosis
- N for no grouping
- C for group by class
- M for multigroup (i.e. showing shadows of the drugs in the secondary groups they belong to)

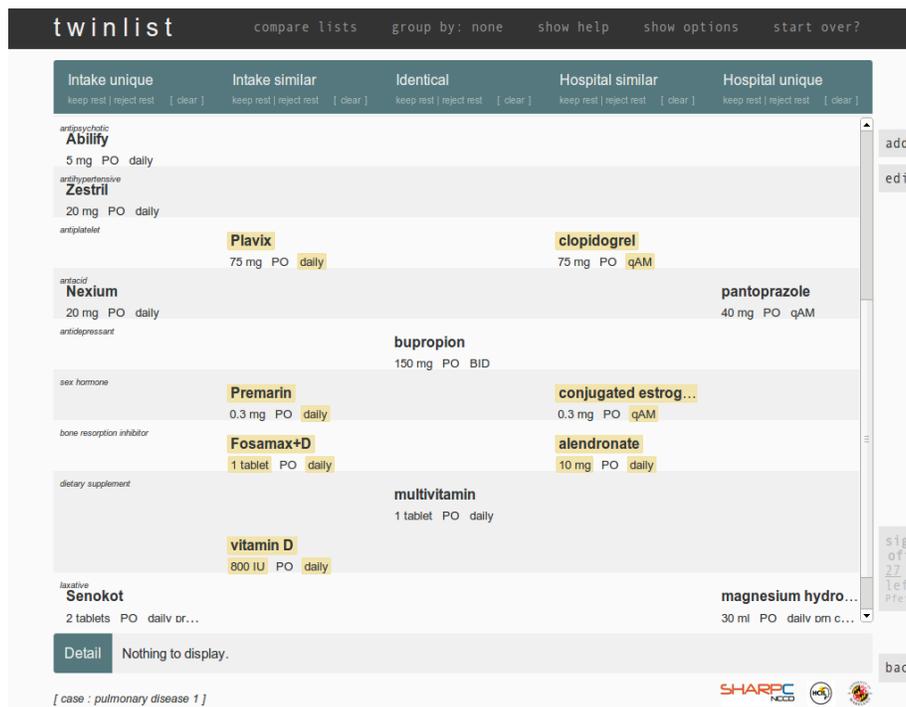


Figure 5: 5-column version with grouping by drug class enabled. Medications are grouped in

distinct rows for each drug class.

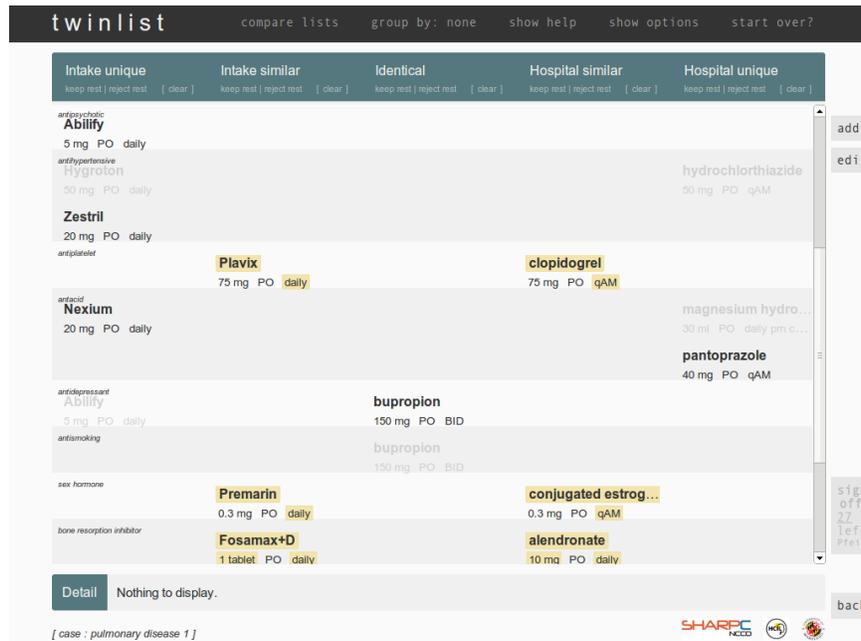


Figure 6: 5-column version with grouping by drug class and multigrouping enabled. Medications are grouped in distinct row by drug class with medications that belong to multiple drug groups appearing as paler duplicates (see bupropion in antidepressants and antismoking).

Impact of grouping on animation:

In order to properly layout items during the intermediate states, three approaches are available. Items could shift vertically down as needed during the staged animation, but this would violate the Common Fate Principle which would reduce the effectiveness of revealing identical or unique items. An additional vertical space adjustment stage could be added, but such a stage would not reveal new relationships and would further increase reconciliation time. Groups could begin with enough space (empty space) so that no vertical adjustment is needed, but this could lead to a very spread out display with the white space serving no apparent purpose at first glance. Given that grouping is expected to be used only in the final layout and that none of the staged animation approaches are appealing, when grouping is enabled only the initial, precompact, and compacted stages of the animation are viewable.

5 Alternative layouts

4.1 Two-column (Baseline)

(NOTE: This baseline was developed for use in our experiment - no similarity information is used. It is NOT RECOMMENDED)

The 2-column version shows the 2 lists side by side a bit closer together. No animation. On mouseover, items gain a dark-gray background to show the details (no similarity information is provided or used during selected).

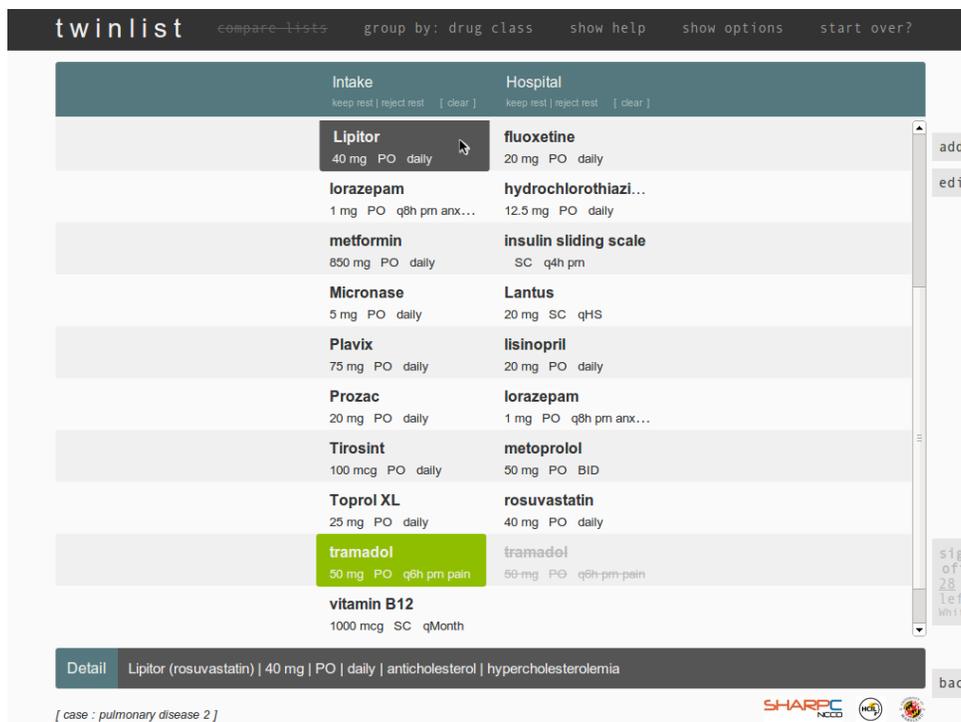


Figure 7. Baseline version. 2-column interface. The pointer is currently on Lipitor. It is highlighted and more detail about this medication is shown at the bottom of the screen but the similar drug is not highlighted. A copy of Tramadol appears in each list. Selecting one requires deselecting the other identical drug manually.

4.2 Two-column with links (i.e. highlight similarity on mouseover)

The 2-column (links) version was designed to maintain the narrow layout of the 2-column version but it uses the similarity information to highlight identical and similar drugs. Grouping the drugs by diagnosis (or class) rather than sorting them alphabetically will “naturally” bring most similar drugs together. Hovering over an item will cause other similar items to be highlighted as well. Yellow highlights indicating differences between drugs also appear on hover. When users select a drug the similar ones are automatically rejected. Since similar drugs can be far away it is important to indicate that a similar drug is offscreen (i.e. need to scroll to see it) so an indicator appears at the bottom or top of the screen to convey that scrolling is needed to see it (see Figure 2.5e for an example).

Figure 8. 2-column (links) version. Drugs can be grouped by diagnosis or class (here we show class). The cursor is currently on Coreg, so the other identical Coreg is also highlighted. We can tell that it is really identical because there is no yellow highlights that normally reveal the differences.

4.3 Three-columns (two lists plus extra diagnosis column)

The 3-column (diagnosis) version is a further extension of the 2-column (links) interface with the addition of the list of diagnosis on the left (see Figure 9). When a drug is hovered over (e.g. Cozear in Figure 9) the similar drug(s) are highlighted as usual, but the linked diagnosis is also highlighted in pale green (here Hypertension and congestive heart failure). This method makes it easy to indicate that a drug belongs to multiple diagnosis groups. The extra diagnosis column also acts as a filter. When the cursor hovers over a diagnosis (e.g. Diabetes in Figure 10) only the drugs that were prescribed for that diagnosis are shown (the others are changed to white and disappear from view, see Figure 10).

This version was never polished entirely and would need some additional work: e.f. the green highlight is a bit too pale. The diagnosis column needs to use a different styling from the drug lists to be easily differentiated so we used a different font size and style (italic) and the line spacing is smaller so diagnoses never really align with the drugs. This could be improved, and the stripping behind the diagnoses should be removed or modified. In fact in this version the stripping can probably be removed entirely since users never need to associate drugs horizontally.

twinlist compare lists group by: diagnoses show help show options start over?

Diagnoses	Intake	Hospital
pain	acetaminophen 650 mg PO q4h prn	acetaminophen 650 mg PO q4h prn...
hypertension	Aldactone 100 mg PO daily	aspirin 81 mg PO daily
diabetes	Amaryl 4 mg PO daily	cimetidine 800 mg PO q12h
insomnia	Ambien 10 mg PO qHS prn	Coreg 6.25 mg PO BID
dementia	Aricept 10 mg PO daily	donepezil 10 mg PO qAM
atherosclerotic va...	aspirin 81 mg PO daily	furosemide 40 mg PO BID
GERD	cimetidine 800 mg PO BID	glimepiride 4 mg PO qAM
constipation	Colace 100 mg PO BID	lorazepam 1 mg PO qHS prn in...
hypercholesterole...	Coreg 6.25 mg PO BID	losartan 50 mg PO qAM
congestive heart f...	Cozaar 25 mg PO daily	magnesium hydr... 30 ml PO daily prn c...

add
edit

sign off
23
left
Jones

Detail Cozaar (losartan) | 25 mg | PO | daily | antihypertensive | hypertension, congestive heart failure

[case : congestive heart failure 1 modified]

SHARPC HCCO MCH

Figure 9 The 3-column (diagnosis) version. The cursor is on Cozaar, highlighting the similar drug and the 2 diagnoses Cozaar was prescribed for. The pale green is fairly subtle but becomes more visible when made to appear and disappear with cursor movement.

twinlist compare lists group by: diagnoses show help show options start over?

Diagnoses	Intake	Hospital
pain		
hypertension		
diabetes	Amaryl 4 mg PO daily	
insomnia		
dementia		
atherosclerotic va...		
GERD		
constipation		
hypercholesterole...		glimepiride 4 mg PO qAM
congestive heart f...		

add
edit

sign off
23
left
Jones

Detail Nothing to display.

[case : congestive heart failure 1 modified]

SHARPC HCCO MCH

Figure 10: The 3-column (diagnosis) version. The cursor is over "diabetes" so only the drugs prescribed for diabetes are visible.

4.4 Three column (plus class)

This version is the same as the 2-column (diagnosis) but uses an extra column for class instead of diagnoses.

5 - Additional suggestions

5.1 Long lists and the danger of scrolling

Because the number of drugs in the lists will never exceed a few dozen items, performance is not likely to ever be an issue. The animation speed may need to be increased in the 1st step where pairs of items are moved one by one, however. One way is to move the 1st pairs slowly and then speed up for the remaining pairs.

With large datasets, all the items may not fit on the screen, requiring scrolling. One way to address the issue is to dynamically reduce the font size so that everything fits (but only up to a point so that readability is maintained). If this is insufficient and scrolling is required, it is critical to make sure users notice the need to scroll with a clear enough scroll bar, possibly supplemented with a visible marker at the bottom of the page indicating there is more below it. The fact that the application checks that all drugs have been acted upon before sign-off is helpful.

The most important thing to handle larger lists gracefully is to detect that similar items that should be highlighted are off-screen. A simple, scalable solution is to simply indicate that some items have fallen out of view, prompting users to scroll to see them. This could be accomplished with an indicator at the bottom or top of each column where more similar items are hidden from view (e.g. an arrow, or a jagged edge, possibly annotated with the number of items).

Also notice that the column header does not scroll, to remain always visible.

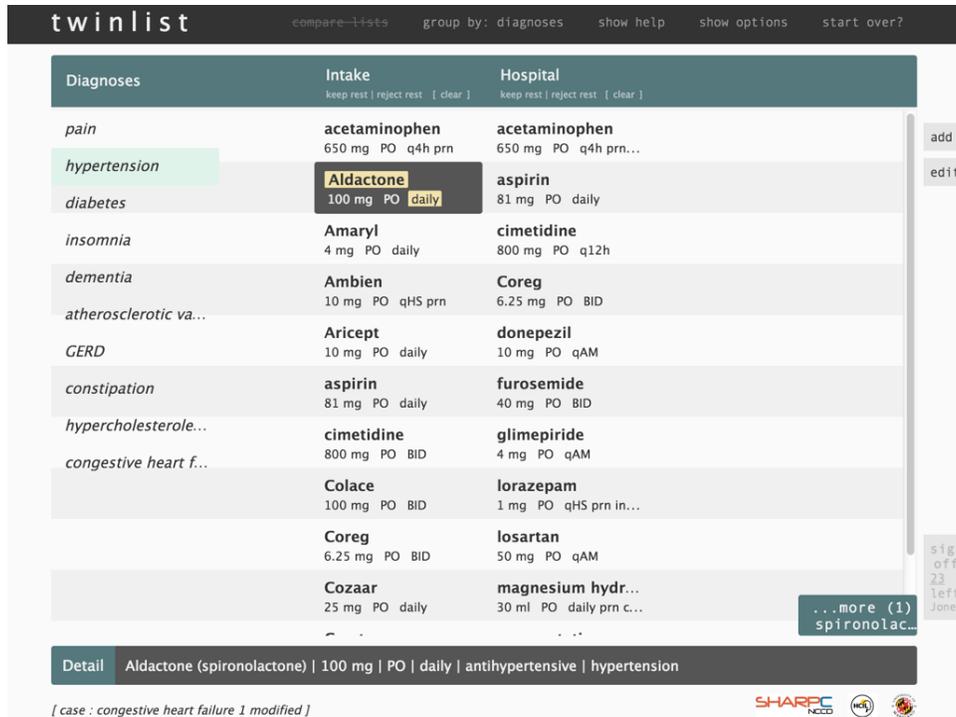


Figure 11: the Cursor is on “Aldactone” but the similar drug is off the screen so a popup appears at the bottom of the screen to alert users of the need to scroll. Animation makes the popup more visible.

While larger screens may allow for alternative layouts, the current prototype was designed to work on a 1024 x 768 pixel screen and used a fairly large font so we could demonstrate it easily. The main menu is thus built into the page header (top), and the item detail occupies a separate stripe below (a fixed position, as it must always be visible). This layout would extend gracefully to larger resolutions as-is, but designers could also consider moving menus and/or the details on either side of the screen. We found that tooltips were not practical to show drug details because that information can be very long and we could not eliminate all cases of occlusion that would hide similar drugs that users needed to see.

5.2 More than 2 lists?

Twinlist focuses on comparing 2 lists. If more than 2 lists need to be reconciled, one could: 1) consider the alternative layouts such as the 2 or 3 column layouts, We believe that those layouts are not as effective as the original 5 column twinlist layout, but may be useful in certain situations

OR

2) repeat the 2-list reconciliation step multiple times: the left list could then be cast as the “current” or “EHR” list, which would be merged with the right list -- the “new” or “foreign” list. If the lists are short enough, the n-list problem could simply be transformed into a 2-list problem by combining n - 1 lists.

5.3 Alternative Selection Methods

Further restricting or tweaking toggle behavior may be preferable, especially if the right-click is needed or desirable for other context-specific actions (e.g. if modifying the order is allowed). For example, small ephemeral buttons for accept and reject could appear below a selected drug.

Explicit accept and reject buttons beneath column headers provide a convenient way to accept or reject entire columns when appropriate. It is unclear whether column-accept and column-reject should overwrite previous decisions; In the current implementation we decided to err on the side of caution and only apply the command to the medications that remained “undecided”.

We decided against manual multi-selection (e.g. using the CTRL key) because the current implementation only requires a simple click to make a decision, thereby eliminating the need to first group drugs before specifying the decision.

5.4 Sign-off alternative designs

Originally we required users to click on a [confirm choices] button to review their final list in a different final summary layout before they could **sign off** on their reconciled list. Accepted and rejected styles are consistent with those used in the main interface, with all kept drugs on the left and all rejected drugs on the right. Such a simple summary of the user’s decisions may provide a better sense of closure. That final summary listed all kept drugs in one list and the rejected ones in the other (see Figure 12). This additional step was removed as it was felt that the final review task could be done just as well in the original layout of the 5 columns (in part because the green color used to highlight of the selected drugs is very visible). Instead we let users sign-off directly when all drugs have been reviewed and acted upon.

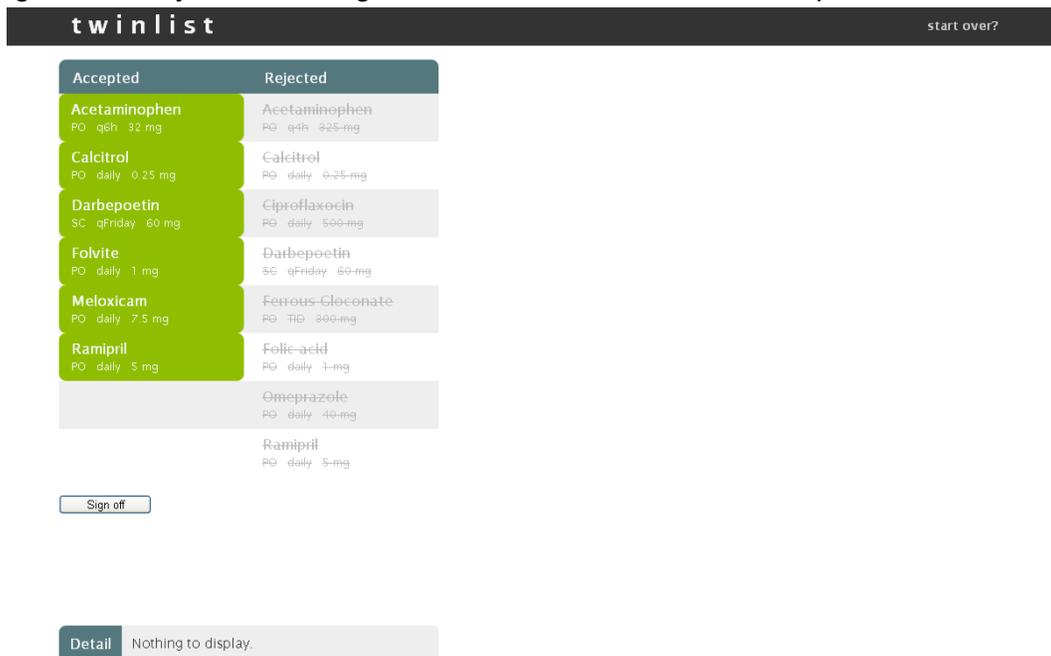


Figure 12: *Example of possible additional summary and sign off.*

Of course the sign-off interface will depend on the “outer” interface and scenario of use. In a discharge setting, for example, it could conceivably feed into patient discharge forms. This was illustrated with the option “Showing the patient tab” available in the option panel. After clicking sign-off a new browser tab is opened showing how the reconciled list could be presented to patients, showing what to continue taking and what to stop, and allowing the clinicians to add annotations as needed.

Doe, David

Please stop taking

Sonata	every hoursS prn	10 mg	PO	<input type="text" value="no notes"/>
--------	------------------	-------	----	---------------------------------------

Please start taking

temazepam	every hoursS	15 mg	PO	<input type="text" value="no notes"/>
-----------	--------------	-------	----	---------------------------------------

Please continue taking

aspirin	daily	81 mg	PO	<input type="text" value="no notes"/>
Chantix	daily	81 mg	PO	<input type="text" value="no notes"/>
Lipitor	daily	20 mg	PO	<input type="text" value="no notes"/>
Capoten	two times as day	25 mg	PO	<input type="text" value="no notes"/>
multivitamin	daily	1 tablet	PO	<input type="text" value="no notes"/>

Figure 13: *Example of patient view (note that this screen was not fully developed or tested in Twinlist)*

5.5 Other options

Users can also choose to hide/remove accepted and rejected medications as they are being acted upon to regain more screen space to show the remaining drugs. This option might be useful for very large lists, but doesn't seem needed for most medication tasks. We do not recommend it.

Font size is controllable using the standard web browser controls (CTRL++ and CTRL--). If those controls are not frequently used in the outer interface, adding explicit buttons might be required.

All options should be saved by the “outer” interface in the user profile (e.g. use of animation or not).

The current prototype also allows users to replay any stage of the list reconciliation animation. This was very important for us to demonstrate the interface and may still be useful for new users (to replay the animation steps one at a time), but those controls should not be the most prominent in the option panel as they are less important, or could be removed entirely.

5.6 Visual design

Design can either break or make the interface. Line, color, texture, form, space -- all of these can make things appear simple and understandable, or overly complex. Meaningless colors and shapes are wasted opportunities and confusing.

In Twinlist particular attention was paid to visual design. Solid colors, used sparingly, define the interface (Figure 8): dark gray anchors the header to the top of the page; bright white lends an illusion of spaciousness; blue-green provides richness and life; golden-yellow highlights important differences between related items; vibrant yellow-green lets users know which drugs have been accepted at a glance. All click-able objects further provide feedback on mouseover: the list viewer in particular uses a slight “nudge” effect to group related medications, exploiting the immediacy of motion and the Gestalt principle of common fate to guide visual queries.

Note: Color schemes and interaction cues need to be consistent with those used in the “outer” application. No matter what, however, there should sufficient feedback to keep users informed as the state of each drug. (Better yet, keep them engaged!)

The screenshot shows the 'twinlist' application interface. At the top is a dark gray header with the 'twinlist' logo and navigation links: 'compare lists', 'confirm choices', 'show help', 'show options', and 'start over?'. Below the header are five columns representing different comparison categories: 'Intake unique', 'Intake similar', 'Identical', 'Hospital similar', and 'Hospital unique'. Each column has a sub-header with 'accept / reject remaining' options. The 'Identical' column is highlighted in green. Drugs listed include Meloxicam, Darbeopetin, Calcitrol, Ramipril, Acetaminophen (two instances), Folic acid, Omeprazole, Ciproflaxocin, and Ferrous Gloconate. A 'Detail' panel at the bottom shows 'Acetaminophen / PO / q4h / 325 mg / analgesic, antipyretic'.

Figure 14 *Color coding. Yellow-green items are accepted, crossed-out items are rejected, and all other items are undecided. The warm gray item in the center is the one the user is currently hovering over, nudged slightly to the right. Notice how solid-colored blocks come to the forefront while light gray items recede. Such visual coding should account for users with color vision deficiency; double encoding is often useful (above, we indicate rejected drugs with a different color and with a strike through the drug name).*

It is often helpful to use colors that users are already familiar with, or colors that are naturally associated with your desired dataset(s). For example our blue-green scheme began with the thought of green scrubs. Dark gray gave the page header some needed weight, and the rest of the color palette evolved from there. Color coding should also account for users with color vision deficiencies; double encoding is often useful. Printing the screen in grayscale is a quick way to verify that all the information is visible without color vision.

5.7 Copy or Copyright notice

Copyright (c) 2011 Human-Computer Interaction Lab, University of Maryland, Tiffany Chao, Catherine Plaisant and Ben Shneiderman, <http://www.cs.umd.edu/hcil/sharp>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

See also the MIT license statement for jQuery:

Copyright (c) 2011 John Resig, <http://jquery.com/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Acknowledgements

We would like to thank Leonardo Claudino, Sameh Khamis, Ran Liu, Ben London, and Jay Pujara for developing an early version of Twinlist. Their work, completed for the Spring 2011 Information Visualization course, can be found on their [wiki](#). A copy of their Workshop on Interactive Systems in Healthcare (WISH) 2011 paper is available on the [HCIL project website](#).

We also thank our main collaborators: Todd Johnson, Jorge Herzkovic, A. Zach Hettinger, Seth Powsner, Elmer Berstram and the many others who have provided feedback and suggestions at all stages of the design and evaluation.

This work is supported in part by Grant No. 10510592 for Patient-Centered Cognitive Support under the Strategic Health IT Advanced Research Projects Program (SHARP) from the Office of the National Coordinator for Health Information Technology.