

I3: UI Critique

Zhichun Ye

Good Example #1: Catch The T

Catch The T is a mobile application for iPhone. It provides real time position data for the MBTA's Orange, Red and Blue lines. Data on the Green Line is unavailable due to technical limitations beyond this application's control. You are able to see the trains' current positions, as well as prediction times for when the trains will arrive at other stations. The data there is provided by the MBTA, but the application is developed independently and is unaffiliated with MBTA. (Citations from the details section of "Catch The T" in app store)

Sparkling points make it a good GUI:

1. This application displays the exact information that the users care and need in a very clear layout. The most important feature of a tracking application is the targets' locations and prediction of their arrivals and "Catch The T" provides both of these. To get the location information of a train, the app eases your way by indicating a predefined sequence: select a particular train (e.g. orange, red or blue); select the direction (eg. Forest Hills or Oak Grove) and then select a particular stop (eg. Malden Center). As in Figure 1, on the main screen, it shows three main options with corresponding background colors representing orange, red and blue line, and these colors help users to focuses on the most important function - you need to choose a line to see its schedule. (*Nielson rule 1: Simple and natural dialogue – graphic design and color*)

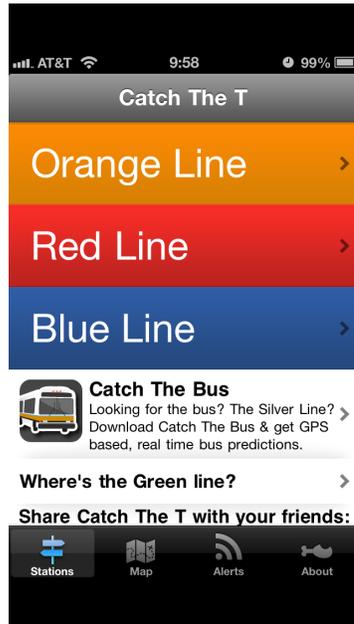


Figure 1 Main Screen Layout of "Catch The T"

After this you are navigated to a screen displayed with the route of this line with a default destination, as shown in Figure 2. You can also change the destination with the two options at the bottom.

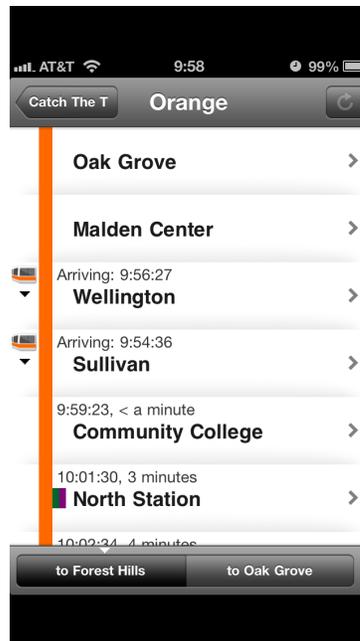


Figure 2 The layout of a specific line schedule

After you set the destination, if you want to know more details about the schedule of a particular stop, you can select that station to see the predicted arrival times to this stop, as shown in Figure 3. This clear hierarchy gives user a clear way to use the software with no obscure descriptions. (*Nielson rules 1: Simple and natural dialogue – ease the user by indicating a suggested or preferred sequence*)

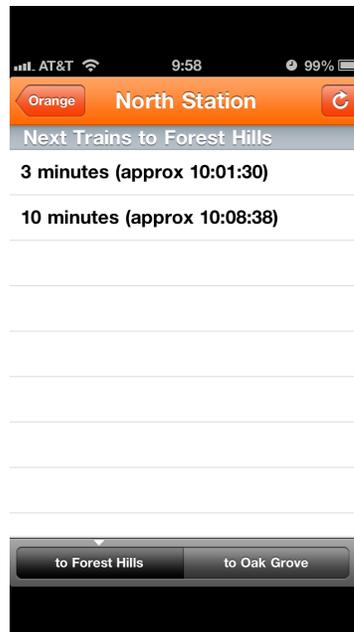


Figure 3 The layout of a specific station schedule

2. When using this application, you do not need to remember the whole map of the subway network, instead the application provides you all the information and you can simply choose from them. Also, within each line's schedule, there are predicted arrival times along with the waiting times with each stop, as shown in Figure 2. In this case, you don't need to check the current time and calculate how long you need to wait; also you can easily know that if the next train will be available before your scheduled deadline. This all minimize the memory load and the work that the user need to perform. (*Shneiderman's 8 Golden Rules 8: Reduce short-term memory load*)

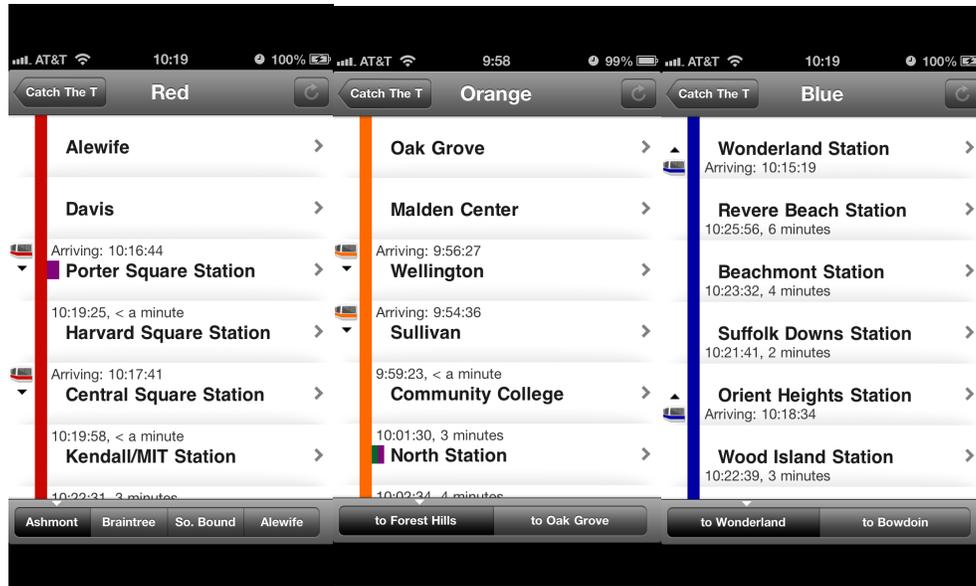


Figure 4 Same layouts in all sub-levels

3. This application has very clear hierarchy in layout and the information is consistent. As mentioned in the first point, the application provides a predefined sequence of searching for particular information. Each level of the hierarchy gives you some particular choices to navigate to the next level. And each sub-level has the same layout thus provides the high consistency across the system, such as the schedule of all these three lines are sharing the same layout expect the color and stops' names, as shown in Figure 4. Same information is usually located at the same place thus facilitate the way to recognize them. (*Nielson rules 4: Consistency*)
4. Users can use shortcuts to navigate to their needed information, as shown in Figure 1. The application provides four shortcuts: 'station' which can provide the schedule information; 'map' which gives you an overall information about the train network layout; 'alerts' which provides you the recent alert; and 'about' provides you information about the FAQ and software. This enables the fast interaction to frequently needed functions. (*Nielson rules 7: Shortcuts – having buttons available to*

access important functions) Users can see each line's route and their alerts as a whole view instead of going to a particular line, remembering that information and switch to another line to check how to change for another train. (*Nielsen rules 3: Minimize user memory load*)

5. The notification and help menu are well located and very useful. At the lower part of the main screen, there are help menus for you to search the bus schedule and notification explains why there is no data for green line, as shown in Figure 1. This can avoid the no-result searching by arranging the help menu at the main screen thus once the user opens this application, he will be notified the information. Also, there is an 'about' shortcut which allows the users to explore more FAQs. (*Nielsen rules 10: Help and Documentation – the need for good, task-oriented search and lookup tools for manuals and online documentation*) The application navigates to a new screen to show the FAQs and arrange them by most frequently asked questions. This again avoids providing the users bunch of unwanted information at the main screen. And at the new screen, it only shows the questions, as shown in Figure 5. So if a user wants to see a specific answer, he can select the question to navigate to a new screen. This enhances the efficiency by hiding the answers while searching for a question since mostly users are just searching for one particular question but not want to read the entire help book. (*Nielsen rules 1: Simple and natural dialogue – provide exactly the information the user needs, no more*)

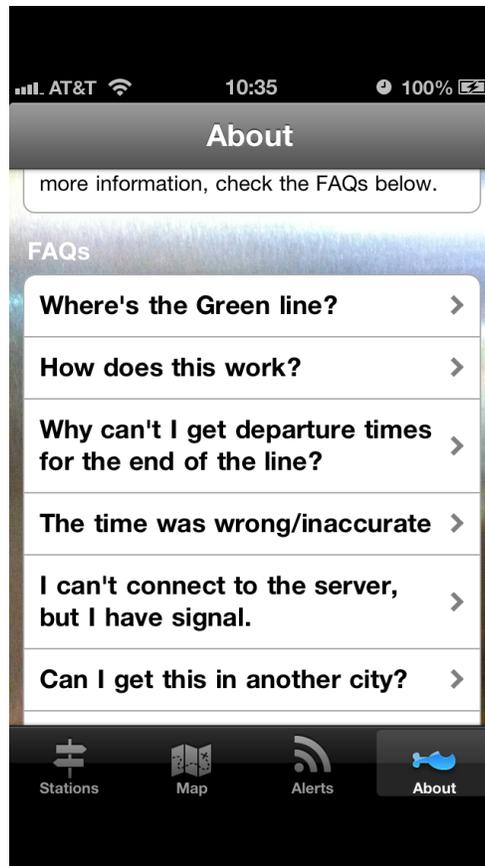


Figure 5 Layout of FAQ section

Good Example #2: Amazon Online Shopping Cart

Amazon.com is a website where you can do online shopping nearly about everything. While you are browsing, you can add item to or delete items from your shopping cart. The shopping cart can preserve all the things you want to buy, just like a real shopping cart in the supermarket. You can also check your shopping cart whenever you want. When you want to check out, you need to type in your address, credit card information and after you confirm your order information you can place your order. After you place your order, you will get a confirmation and you can then continue shopping.

Sparkling points make it a good GUI:

1. When you have added some items in your cart, the confirmation screen is well navigated and informative even with very simple dialogue, as shown in Figure 6.

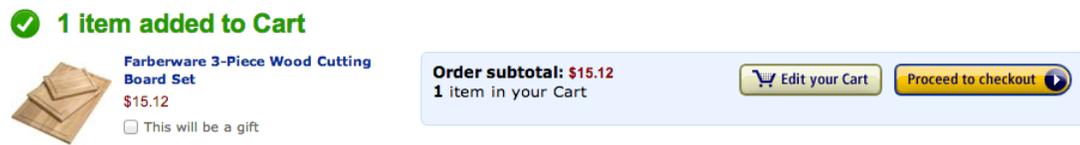


Figure 6 Layout of confirmation screen of adding an item

It first reflects the user's recent action – adding an item to the cart, and shows it using both a correct mark and words in green color. This grasps users attention in higher priority. (*Nielson rules 1: Simple and natural dialogue – graphic and color design*) Also, it provides the information about numbers of items and total cost of the cart, and this is a very helpful way to confirm whether the number of this item is correct. Meanwhile, it also gives user a hint about how many things he has put in the shopping cart. These information can be considered as a part of the positive feedback, and they are expressed in clear way. (*Nielson rules 5: Feedback – the system should provide positive feedback and partial feedback as information becomes available*) Also, considering the user may want to checkout immediately or make some updates about the cart, this page also offers two shortcuts to either edit or checkout. Having these two buttons here will enable the quick shopping since these two functions are important in the shopping process. (*Nielson rules 7: Shortcuts – having buttons available to access important functions*)

2. The sign in page is very concise but supports lots of important and necessary sub-level tasks. As shown in Figure 7, they provides two frequent appeared problems in this process as two help-menu links below the login interface, and the details of the consequences of saving your password on this computer is also provided as a link

below the password field. This gives users a very clear layout as well as an idea about where to find help. The sub-level help books are task oriented thus causes less confusion. And all these tutors about how to reset the password or change email address are available while the users are carrying out these operations in sequence. (*Nielson rules 10: Help and documentation – it is better for the user to be able to refer to the instructions while carrying them out than to have to remember them*)

amazon.com

SIGN IN SHIPPING & PAYMENT GIFT-WRAP PLACE ORDER

Sign In

Enter your e-mail address

I am a new customer.
(You'll create a password later)

**I am a returning customer,
and my password is:**

Keep me signed in on this computer. [Details](#)

[Sign in using our secure server](#)

[Forgot your password? Click here](#)

[Has your e-mail address changed since your last order?](#)

[Conditions of Use](#) [Privacy Notice](#) © 1996-2013, Amazon.com, Inc. or its affiliates

Figure 7 Layout of sign in page

After you click the link of “Forgot your password? Click here”, it directs you to a page which allows you to reset the password in some procedures, and all of them are well articulated and given in a suggested sequence, as shown in Figure 8. This provides users a more clear idea about the steps he needs to carry out and will make him feel more comfortable because he gets to know the problem he is facing is not a big deal, he doesn’t need to worry much about it since there is very clear guidance. (*Shneiderman’s 8 Golden Rules 5: Offer simple error handling – the system should be*

able to detect the error and offer simple, comprehensible mechanisms for handling the error)

Amazon.com Password Assistance

Enter the e-mail address associated with your Amazon.com account, then click Continue. We'll email you a link to a page where you can easily create a new password.

Email address:

Figure 8 Layout of "find your password" page

3. The error messages are informative and indicate you several possible ways to deal with the problem. For example, when you input an existed email address but you think you are a new customer, the system will generate the following error message indicating you already have an account, as shown in Figure 9. It also provides several guessed ways that you want to do: forgot your password, want to sign in or want to register with another email address.

E-mail Address Already in Use



You indicated you are a new customer, but an account already exists with the e-mail **zhichun.ye@gmail.com**

Are you a returning customer?

- [Sign In](#)
- [Forgot your password?](#)

New to Amazon.com?

- Create a new account with [a different e-mail address](#)
- Create a new account with [this e-mail address](#)

Still need help?

- [Contact Customer Service](#)

Figure 9 Layout of error of "email address already in use"

These options are all in very clear words and within limited choices to avoid confusing. (*Nielson rules 8: Good error messages – they should constructively help the user solve the problem, by guessing what the user want to say*) And they can be

considered as the exits of current stage, since users can make mistakes at every step, we need to make sure that users can recover from these errors in a very easy way.

(Nielsen rules 6: Clearly marked exits)

4. At the shipping address page, the default address which is the most used address will be displayed at the top, users can use this address by clicking the “ship to this address” button which is colored in yellow to attract attention as a shortcut.

However, users can also type in a new address. Since address lines are provided with legal input format and the country field has a drop-down menu, users can reduce their memory load and can prevent errors of typos, as shown in Figure 10.

Or enter a new shipping address

Be sure to click "Ship to this address" when done.

Full Name:

Address Line1:
Street address, P.O. box, company name, c/o

Address Line2:
Apartment, suite, unit, building, floor, etc.

City:

State/Province/Region:

ZIP:

Country:

Phone Number: [Learn more](#)

Figure 10 Layout of entering new address form

Also, when you are choosing the shipping method, it gives you the calculated arrival date along with each method, thus you don't need to open the calendar to calculate the date. And after you have typed in your credit card information, a great feature here is to provide you a notification that you still have the chance to review your final order before place it, as shown in Figure 11. We know that users may get

nervous about losing money on unwanted items and usually they are very careful or cautious at this “credit card information” step. So this button and notification at the very outstanding place will give the user a more clear feedback about the whole procedure. (*Shneiderman’s 8 Golden Rules 4: Design dialog to yield closure – the feedback at the completion of a group of actions gives them operators the satisfaction of accomplishment, a sense of relief, and an indication that the way is clear to prepare for the next group of actions.*)

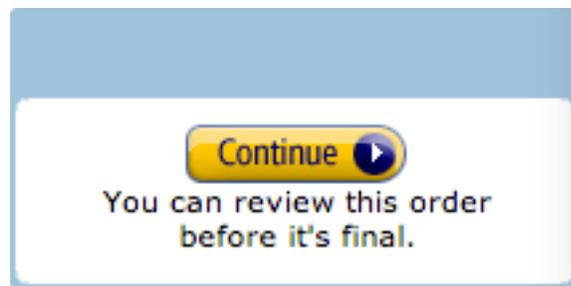


Figure 11 Notification of the chance for final review

5. Across the shopping process, the interface is always carrying out information from the user’s perspective. It uses “your account”, “your cart”, “Enter your email address”, “I am a new customer”, “I am a returning customer and my password is”, etc. This enables the user to feel like the system is reading his mind and thus he can perform the shopping task as his daily in-store shopping. (*Nielson rules 2: Speak the users’ language*) Also, through out the process, there always a progress bar on the very top indicating which state or step the user is in, and this gives the users a very clear overview about how much work of this process has be done, as shown in Figure 12. These can be considered as the positive feedback. (*Nielson rules 5: Feedback*)



Figure 12 Progress bar

Bad Example #1: OpenMBTA

OpenMBTA shows you stop locations, the next arrival times at each stop, and the time schedules for the route you selected. You can switch between a map view and schedule view for each route. The colored pins in the map view means: a red pin is a stop, a green pin is the starting point of a trip, a purple pin is a stop where a subway train is just about to arrive. (Citations from OpenMBTA help menu)

Points violates the principle:

1. The “About/FAQ” button in the main menu isn’t consistent with its name. The “FAQ” section in OpenMBTA only has two questions: 1. Why is the iOS app rated 17+? 2. How can I support OpenMBTA?; and neither of them is relevant with the features of the software. What’s more, it provides information about the mission and credits in the “About/FAQ” but users don’t seem to care much about these. Users expect to find out information about how to use the software and its key features under FAQ section, but in this case it doesn’t provide these wanted information, but include some other tedious ‘non-relevant’ information. This violates the rule “present exactly the information on the user needs, no more.” (*Nielson rules 1: Simple and natural dialogue*) Also, they provide the answers together with the questions. Since answers are quite long and questions are not highlighted, users may feel hard to search for the particular question among lots of contexts, as shown in Figure 13. So overall the documentation of the help menu is not easy to look-up or read. This

violates the basic rule of help and documentation – the need for good, task-oriented search and lookup tools for manuals and online documentation. (*Nielson rules 10: Help and documentation*)

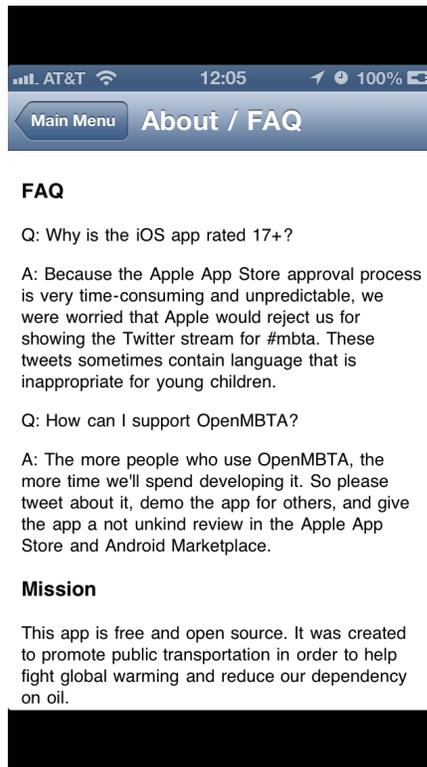


Figure 13 Layout of "About/FAQ" section

I find the answers to both of the questions in the FAQ section are giving some information to let the user using appropriate words to give high rate of the software. Thus I think they may have this kind of issue when users using inappropriate words in public to describe this software. So they give these two questions higher priority than questions about the features. And they may want to honor the people who have put a lot of efforts in developing this software thus they create the mission and credits part. But in my opinion, even though these information is important and should be included in the "About" section, the frequently asked questions and answers should also be included since they are what they user really need, and this

honors the rule of help and documentation: the need for good, task-oriented search and lookup tools for manuals and online documentation. (*Nielson rules 10: Help and documentation*) And the information about the mission and credits can be provided as a sub-level information thus the user do not need to filter from lots of information to find the one they want. (*Nielson rules 1: Simple and natural dialogue – present exactly the information the user needs, no more*) And also in the FAQ part, they should only display the question alone for a better way to search information. Here is a sketch of my design:

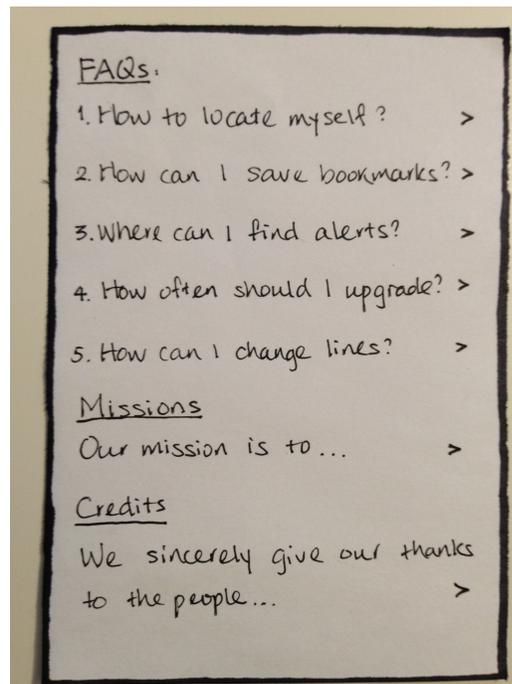


Figure 14 My design of "About/FAQ" section

2. The display of the schedule of a particular line is bad organized and so much information are put together.

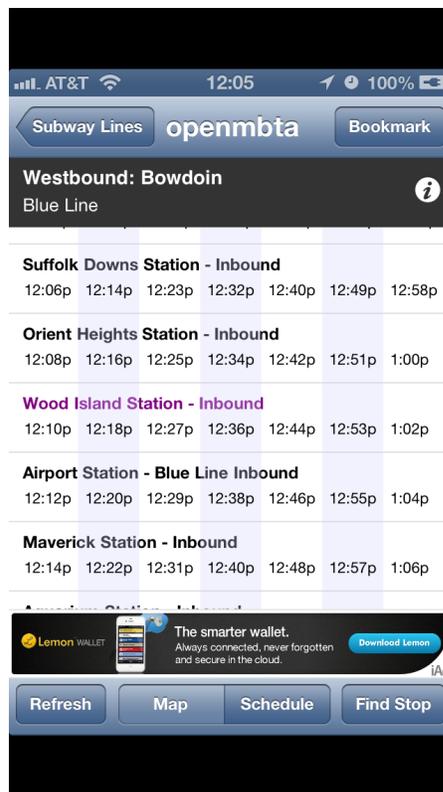


Figure 15 Layout of the time schedule of Blue Line

As shown in Figure 15, the predicted arrival times of each stop are displayed under the station's name and in timeline order, but these times don't give user a good sense of where the train is right now. Users need to go back to the map view of the train to find out the current location. This violates the rule "minimize user memory load" since each time user needs to remember both the time schedule and the location because they are shown in different pages. (*Nielson rules 3: Minimize user memory load*) Another problem is that the schedule still includes the passed trains' arrival times. To users, these are useless information and according to the Nielson rule, we should provide users simple and natural dialogue, no more useless information. (*Nielson rules 1: Simple and natural dialogue*)

One reason I think they want to arrange the time information in this way is because they want to provide the maximum view scope of all the available time schedule. And since they need to provide all the information even it is hours later, they cannot put all the time schedules together with the current train location otherwise the whole view will be a mess. But in my point of view, maybe it is not that necessary to display so much information at the same time because most people who want to check the schedule are going to catch the train in a very short time. Thus the schedules about one hour or two hours later are really not necessary for them. We can only display the most recent line schedules maybe within 30 minutes, and show the current location of each train. This honors the rule of providing user exact the needed information, no more. (*Nielson rules 1: Simple and natural dialogue*)

Here is a sketch of my design:

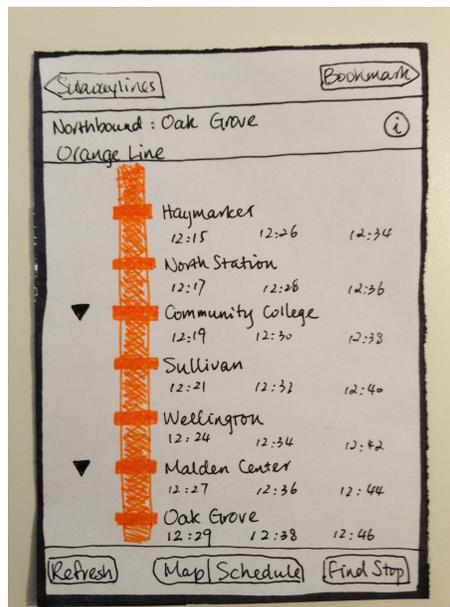


Figure 16 My design of time schedule display of a particular line

3. Across the application, there is no clear hint that if the page we are viewing has been updated or should be refreshed to catch the latest schedule. Since there is no

positive feedback, user may have a wrong feeling about the schedule, and this violates the rule that the system should provide positive feedback as information becomes available. (*Nielson rule 5: Feedback*)

I think they don't give a clear hint about the availability to update information is because the schedule time might be changing every second, and if the screen keeps updating, user don't have enough time to scan over the useful information. But we can rethink it, that we don't need to provide the information every time it has been updated. We can update the information every 2 minutes and we don't have to refresh the screen automatically. Once there is new information about the time schedule in 2 minutes, we can have a signal representing the availability to update. We can also use one of the Kineticons from the research paper we read (Harrison, C. et al. Kineticons: Using Iconographic Motion In Graphical User Interface Design. CHI'11) to demonstrate this information and I think these kinetic motions are more powerful to attract people's attention. This can be a part of the positive feedbacks. (*Nielson rules 5: Feedback*)

Here is a sketch of my design of the icon:

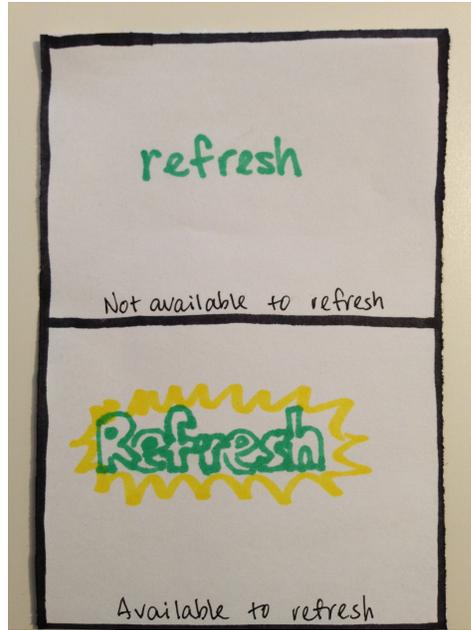


Figure 17 The design of icon representing the availability to refresh

4. The map view of the schedule of a particular line is not consistent with its notation in the help menu.

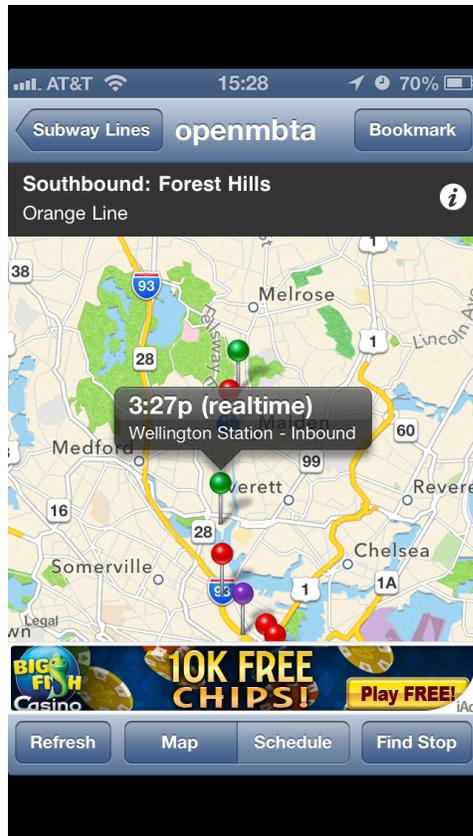


Figure 18 Inconsistency between map view and notations from help menu

As in Figure 18, we can see there are two green pins in the figure but there is only one start point of the train. This violates the consistency rule. (*Nielson rules 4: Consistency*) Also, the first impression of the map view is it is too crowded. Since it tries to show every station, current train location and the destinations, there are too many pins on the limited space thus they overlapped and some even hide the useful route information, as shown in Figure 19.

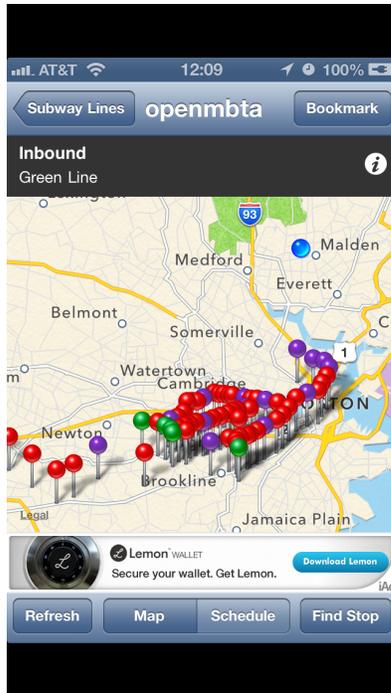


Figure 19 Too many pins in the map view

And once you zoom out the map, the pins will become distinct thus you don't have the overall idea about the route of the train, as shown in Figure 20.

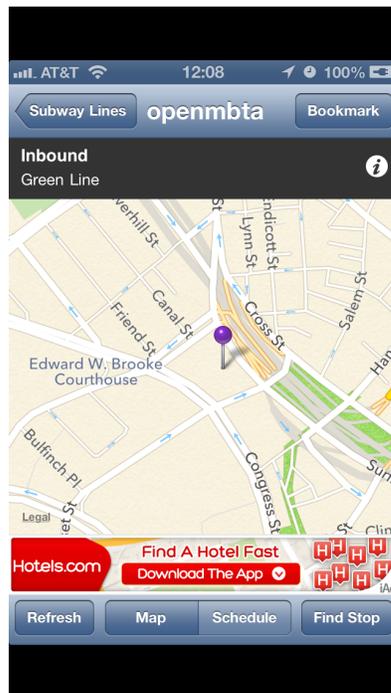


Figure 20 After zoom out we can't have view of the route

On the contrary, when you zoom in the map, pins will be clustered and this makes it hard to be recognized by users. All these violate the rule “simple and natural dialogue”. (*Nielson rules 1: Simple and natural dialogue*)

One reason may cause the inconsistency between the green pins and the start point might be the implementation itself thus we cannot change anything from the GUI part to avoid this problem. The technicians may need to put more effort to check if there is anything wrong with the code. But as to the over-informative screen, which I mean there are too many pins on it, we can definitely improve this by redesign the interface. The original design inspiration of using colored pins to represent different stations might be it's easy to select since each pin has a big circle and using these distinct pins may convey the meaning that the stops are distinct from each other. But we can convey the same meaning but drawing the route on the map using the corresponding color and we use a node represent a station with a darker color. In this way, when the user zooms out, he will see each station as well as the route around that. When he zooms in, he will be able to see the whole route of the line. He may not see each station but I believe when he zooms in he cares more in the overall view of the route instead of each stop. This will give user a clear and simple interface about each line's route and stations and also honors the rule of simple and natural dialogue. (*Nielson rules 1: Simple and natural dialogue*)

Here is a sketch of my design:

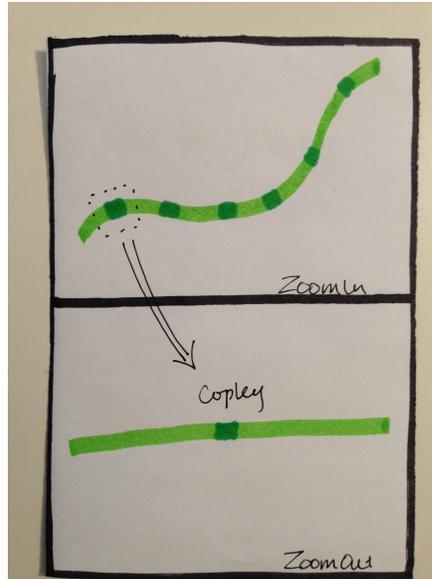


Figure 21 My design of the display after zoom in or zoom out

5. One of the biggest problems is that the software doesn't provide a whole map of all MBTA lines. When user wants to make a connection between two or more lines, the user needs to go back and forth to different lines to see if these two lines have a connection point. This needs a lot of memorizing work and a kind of luck because in the worst case a new user will scan over all the lines to find out the right connecting line. This obviously violates the rule of "minimize the memory load". (*Shneiderman's 8 Golden Rules 8: Reduce short-term memory load*)

I think the lack of the whole map may due to the design team wants this application to focus on the time schedule of each line, but not the connection. But since there is a great possibility that users need to make connections, providing a whole map view becomes very helpful. We can simply add a whole map of all the lines to it and make a shortcut to the map, because shortcut can enable the user to access this feature whenever they need it. As in rule shortcut says: user should be allowed to jump

directly to the desired location in large information spaces. (*Nielson rules 7: Shortcuts*)

Here is a sketch of my design:

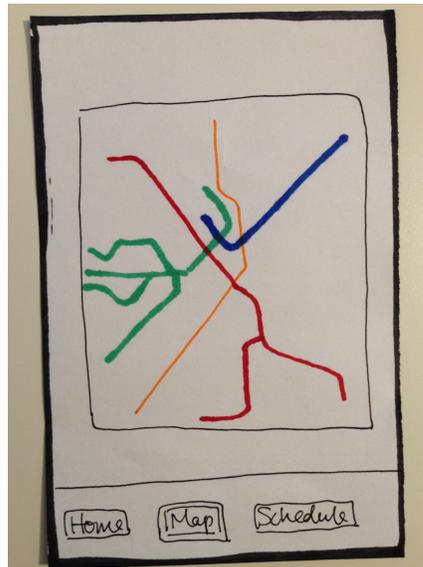


Figure 22 The shortcut linked to the whole view map

Bad Example #2: Shaw's Rewards Card Online Application

Shaw's is a well-know market for groceries or home accessory shopping. Shaw's rewards card provides instant savings on thousands of items and delivered you exclusive offers. The online Shaw's rewards card application enable you to register and get a reward card online, and gain access to shows.com account. Basically you need to fill in several forms online for card information uses. (<http://www.shaws.com>)

Points violates the principle:

1. The information suggested by the blank field in the form is not consistent with the notification.

The image shows a form with the following elements:

- Home Phone Number:** Three input boxes separated by dashes.
- Mobile Phone Number:** Three input boxes separated by dashes, with the third box highlighted in blue.
- Card-less ID:** Two input boxes.
- Savings by Mail:** A label with a yellow tooltip that says "Alternate Id" pointing to the second input box of the Card-less ID field.
- Notification Box:** A box titled "What is my Card-less ID?" containing the text: "This is a secure number you can use at checkout in place of swiping your card. Please choose a 10 digit number that you can easily remember. We recommend you use your 7-digit phone number (without area code) + a 3-digit PIN as your Card-less ID."

Figure 23 Inconsistency between blanks and notification

As this Figure 23 shows, in the Card-less ID field, when you move your mouse on the later part of the blank, it will show you "Alternate Id", but the notification box beside it indicates that the Card-less ID is recommended to be 7-digit + 3-digit PIN. In this way the user may be confused about what exactly should be filled into the later part of the blank since the information is not consistent. (*Nielson rule 4: Consistency*)

I think the designer of this form separates the field of Card-less ID into two parts to indicate a preferred format, and he wants to provide extra information in the field to suggest this Card-less ID can be your alternate ID. But this information or term is not consistent with the term in the notification box. We can change the term in the blank field or in the notification box, making them the same. Also we can provide a sample format in the field thus reduce the confusion caused by obscure description, and this honors the rule of minimize memory load. (*Nielson rule 3: Minimize memory load – the system should describe the required format and an example of legal and sensible input as a default value*)

Here is a sketch of my design:

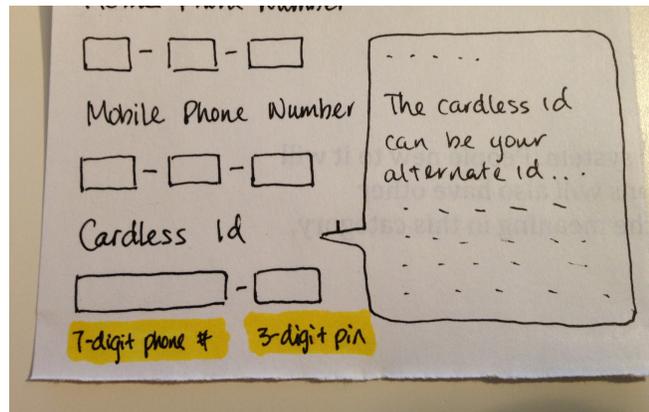


Figure 24 My design of information form

2. There is not clear notification about that you need to obtain an online account before you apply for a reward card. But since you can get easy access to the “apply for rewards card” page from the main page, users may falsely go directly to apply for a new card, at least I do. And he will only find it is not allowed to do so after he fills out all the blanks and press the continue button. Then the error box appears, as Figure 25 shows.

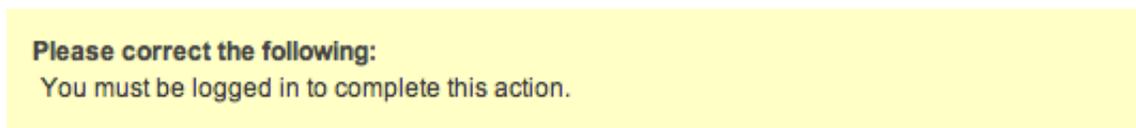


Figure 25 The error message

This violates the rule of both “feedback” and “prevent errors”. The system can provide this feedback once it finds the user is not logged in and still tries to fill in the blanks. As said in rule of “prevent errors”, the system feedback should be sufficiently

varied to provide additional differentiation between modes. (*Nielson rule 9: Prevent errors*)

One reason I think they neglect this important notification is because they think the using of “sign up” instead of “apply for” a rewards card can avoid it happens. Signing up might means you have already got an account thus they take it for granted you don’t need to create a new account. But this doesn’t make sense because even though you have already got an account, you might not be logged in at that time. So once the user comes to the signing up an account page, the browser should detect whether the user has logged in. If not, they should have a signal or popup window indicating the user need to log in first otherwise he cannot fill the signing up form. In this way the system give the user a suggested or predefined sequence to go through the signing up process. And this honors the rule of simple and natural dialogue. (*Nielson rule 1: Simple and natural dialogue – ease the user by indicating a suggested or preferred sequence*)

Here is



Figure 26 My design of the popup window presenting error message

3. After the error message popped up, there is not clear hint or link which will navigate you to the log in or register page. User still needs to find the log in or register button across all elements from the web page. This violates the rule of good error messages which should constructively help the user solve the problem by guessing what the user want to say. (*Nielson rules 8: Good error messages*)

One reason why they don't provide the link is the possibility that they think good error message may just provide clear and precise and polite information, which they have achieved. But to provide better user experience, the system should think ahead of users. (*Nielson rules 8: Good error messages*) Thus we can add two links below the error message: one links to a register page which allow new users to register an account, another links to the log in page which let the user to log in with their current account. My design is shown as Figure 26.

4. The registry for a new account process takes long time to proceed to the next step, but within the long response time, there is no feedback to user whether the process is still on or has already been hang up. As in the rule of feedback, feedback becomes especially important in case the system has long response times for certain operation. (*Nielson rule 5: Feedback*) No feedback will cause the user lose patients and may redo the process or cut the process. Neither of the result is good for improving user experience.

The reason leads to this may due to the designer assumes the transform from one stage to another is very quick thus we do not need to provide any information in the middle of these two stages. However, the real world situation is far more complicated thus requires a more thorough analysis and design. So to improve this situation, we can simply add a notification or progress bar in the transition period telling the user that the system is still processing his information. This honors the rule of feedback. (*Nielson rules 5: Feedback – provide the positive feedback when it is possible*)

Here is a sketch of my design of the progress bar:

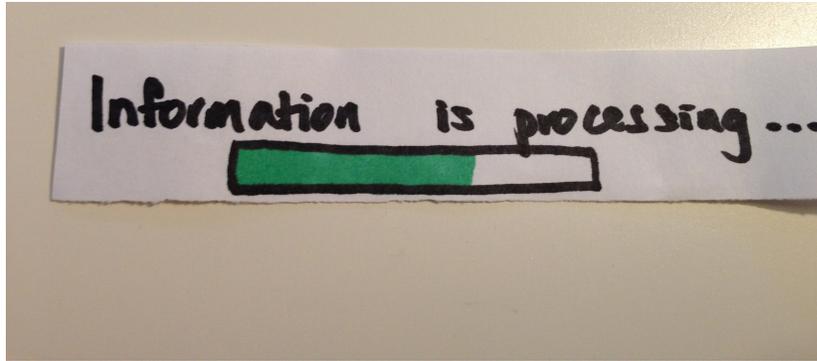


Figure 27 Progress bar

5. The result page of the “search store” operation is confusing and misleading. As Figure 28 shows, after you type in a particular address and zip code, it navigates you to the result page. However, at the first view, most part of the page is empty, but with the title “Store Location” on it.

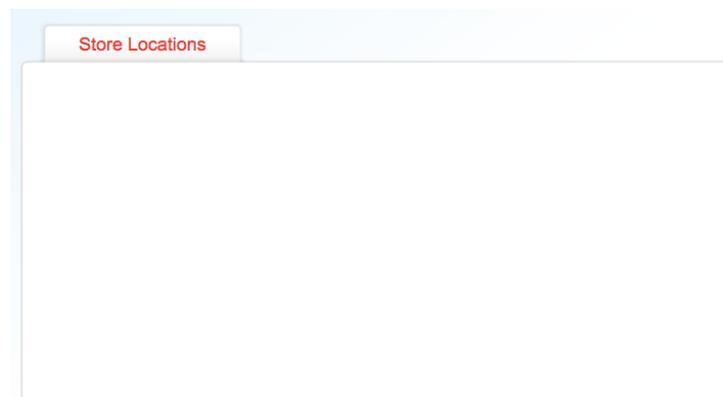


Figure 28 The empty space of the search result

This gives the user a very strong feeling that there are no available search results. However, if you scroll down the page you will find that the results are listed below the empty part, as shown in Figure 29.

Your results for: Hungtington Ave, boston, MA

Results 1 to 10 of 29 Show 10 | 25 | 50

1

Hyde Park (541)
1377 Hyde Park Avenue
Hyde Park, MA 02136

(617) 364-9375 — Main
(617) 364-3161 — Pharmacy Phone
(617) 361-3417 — Pharmacy Fax

1.099 miles from Hungtington Ave, boston, MA

[See Store Details](#)
[Driving Directions](#)

Make This My Store

Preferred Rewards not available at this store.

Gas Rewards not available at this store.

2

Dorchester Burger (586)
4 River Street
Mattapan, MA 02126

(617) 298-1462 — Main

2.407 miles from Hungtington Ave, boston, MA

[See Store Details](#)
[Driving Directions](#)

Make This My Store

Preferred Rewards not available at this store.

Gas Rewards not available at this store.

Figure 29 The real search result

After I click one particular store, the empty space shows the picture of that store.

This is useful for user for better recognition of that store. However, the space for

picture is not necessary before the user choose a particular store. And this can be

considered as the not needed information that violates the rule of providing exactly

the information the user needs, no more. (*Nielson rules 1: Simple and natural dialogue*)

I think the designer leave this space empty incase of the further information display

is reasonable because in that way after the user select a store, the whole layout of

the page will remain the same and keep the information at the same page, which

honors the rule of less is more and consistency. (*Nielson rules 4: Consistency*)

However, we can display the picture right below the selected store and this will not

mess up all the elements in the page but can still represent the user the picture

information. This is support by rule of simple and natural dialogue, which suggest we should provide user exact information when it is needed. (Nielson rules 1: Simple and natural dialogue)

Here is my sketch of the search result page:

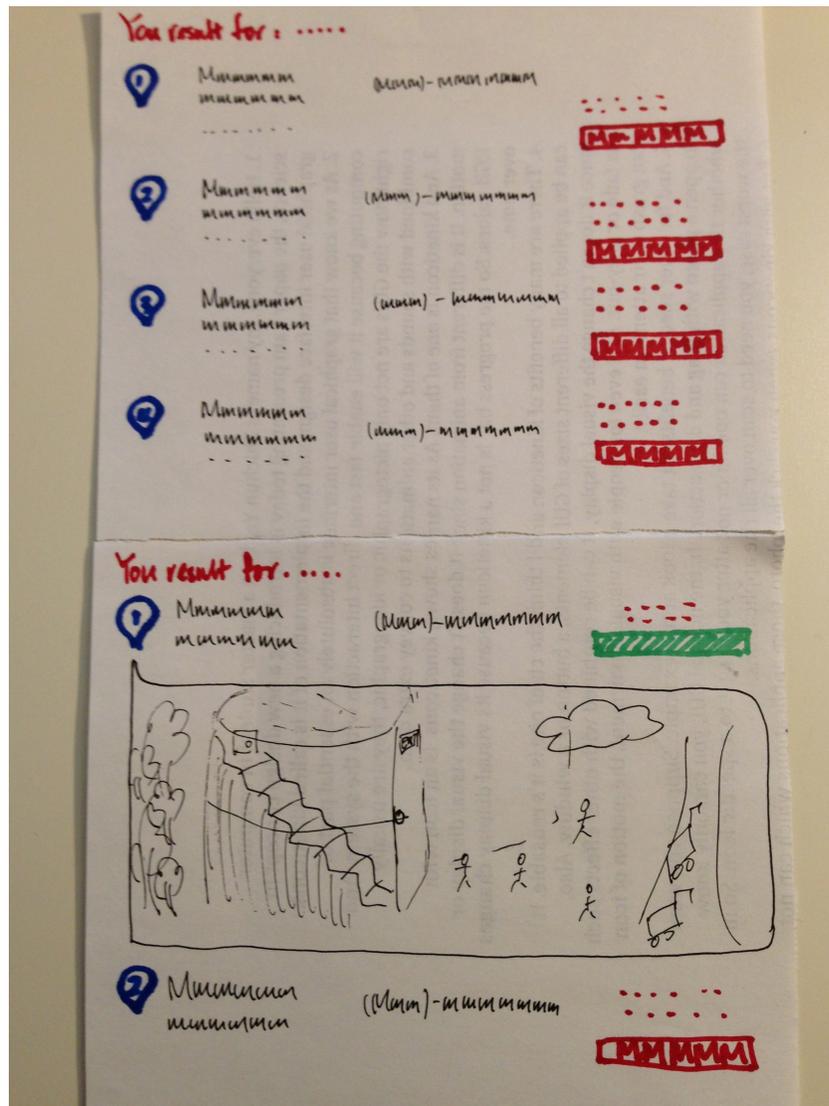


Figure 30 My design of the search result page