UI Critique

1. Good Design: Twitter for iOS

Twitter is a social networking site where user can post 140 characters status update, known as a "tweet", so that it can be seen by their followers. Followers can reply the tweet by mentioning the original poster username, or simply resharing the tweet (known as "retweeting"). The Twitter for iOS is a mobile app designed to help Twitter user to access the feature through their mobile devices.

1.1 Feature organization

The application is organized into four sections: Home, Connect, Discover, and Me. This structure maps exactly to the structure of the web version, as well as other platforms (Figure 1). Twitter also uses the same icon for every sections.
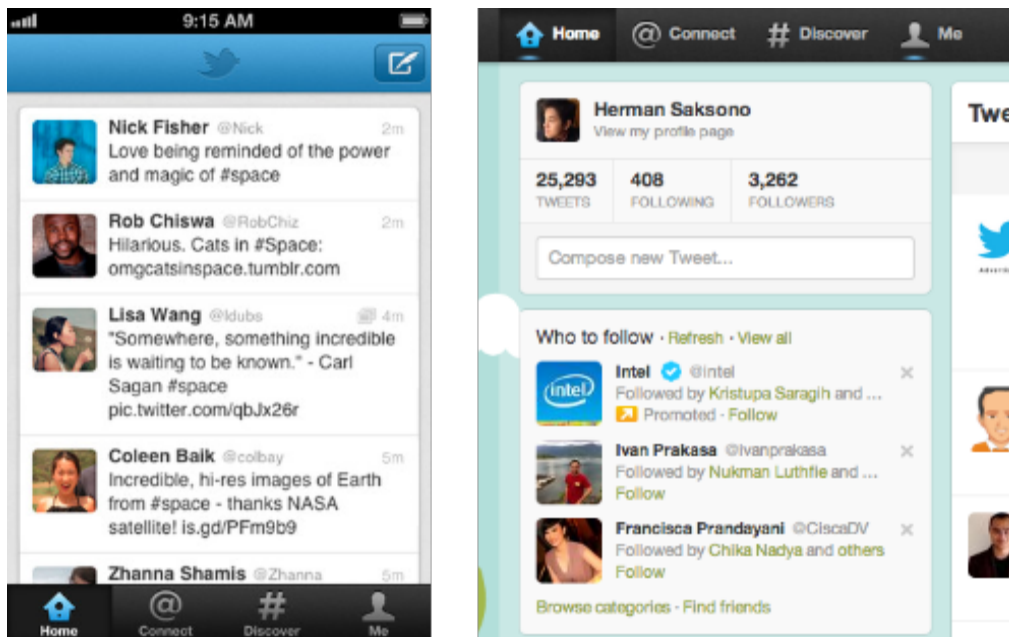

Figure 1. Feature organization in Twitter for iOS (left) are identical to the web version (right)

By employing the same organization and consistent icon across multiple platforms, Twitter user could apply their past knowledge under new environment (Nielsen's 4th heuristic). This is a learnability principle known as familiarity.

1.2. Refreshing Twitter timeline

To refresh their timeline, user is required pulls the timeline to maximum and release it. After that the app will display a rotating icon indicating that the app is currently loading the latest updates through the network (Figure 2). After the latest timeline have been retrieved, new tweets are displayed, and the icon disappeared.

Figure 2. Rotating circle indicate that Twitter is currently retrieving the latest update.

The rotating icon (and clear labelling) provides a visual confirmation that user's request to refresh the timeline have been received (Norman's 1st and 3rd principle, and Nielsen's 1st heuristic). It also made the system transparent to user, by indicating that the application is currently in the process of communicating with Twitter's server (Norman's 3rd principle and Nielsen's 1st heuristic). Furthermore by placing the rotating icon on a non-modal state, users can perform other tasks in the application, such as replying other status updates. The principle of flexibility that allows user to perform more than one task at a time is known as interleaved multithreading.

1.3. Preventing incorrect mention

When a twitter user mentioned other user's name on their status update, Twitter automatically notifies the mentioned user. To help user finding the correct username, Twitter provides automatic username suggestions that appears when a he/she typed "@" (Figure 3).

Figure 3. Twitter offers username suggestions that is initiated when user typed "@"

This feature also reduces the likelihood of user typing the wrong username (Norman's 6th principle and Nielsen's 5th heuristic). Since this feature is non-modal, user had the flexibility to ignore the suggestion and continue typing usernames that are not offered in the suggestions.

1.4 Preventing user exceeding 140 characters

A status update in Twitter is limited to 140 characters. To prevent user from having their status update truncated, Twitter constraints the user by disabling the "Tweet" button when there are more than 140 characters (Norman's 5th principle).

Furthermore, Twitter anticipate user from composing a status update longer than 140 characters by providing character count at the right side of the screen. When there are 10 characters left, the counter turns to red (Figure 4), signalling that the user need to reduce the length of their update (Norman's principle 6).
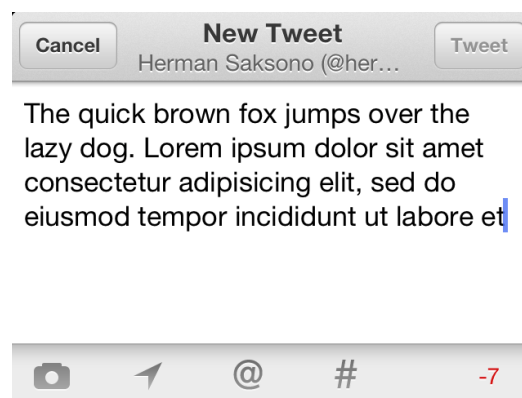

Figure 4. Twitter provides information how many characters are left

When user exceeds the character limitation, Twitter did not stop the user from composing their update. Instead Twitter allows them to continue recomposing and reducing the characters until it reaches the 140 limit. This is a feature of user *preemptive* dialog, where user can control the dialog with the system, which is a usability principle in flexibility.

2.5. Timeline gap

When a user left the Twitter iOS app and returned several hours later, it's likely that a large number of status updates have not been retrieved. At this point there will be a timeline discontinuity. To ensure this discontinuity is visible to user (Norman's 3rd principle), Twitter displayed a jagged gap (labelled "Load more Tweets") between the last retrieved timeline and the recently retrieved timeline (Figure 5). To retrieve the timeline between the gap, user is simply required to click on the gap.
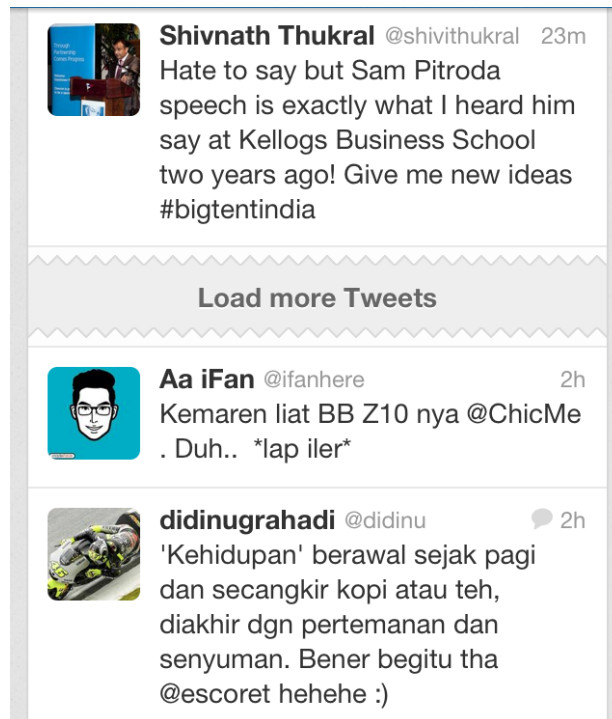

Figure 5. Jagged gap in the timeline indicates a discontinuity in the timeline, and needs user action to refresh it.

2. Good Design: Google Maps for iOS

Google provided an iOS version of their mapping application. This iOS application offer most of the features available in the web version of Google Maps, namely as places search, direction, street view, satellite view, reviews and pictures.

2.1. Flattened Structure

Since there are many features in Google Maps, in the iOS version, Google organized the functionality into horizontal hierarchy. Place's address, photos, street view, reviews, direction, etc, are displayed after a place is found. Driving direction, public transport, and walking direction is available after user selected the Direction menu. Search result, places search, and favorites appears after user initiated a search. This flattened structure simplifies the clutter of the initial screen (Norman's 2nd Principle), so that there is only 1 search box, 3 buttons, and 1 mini button (Figure 6).
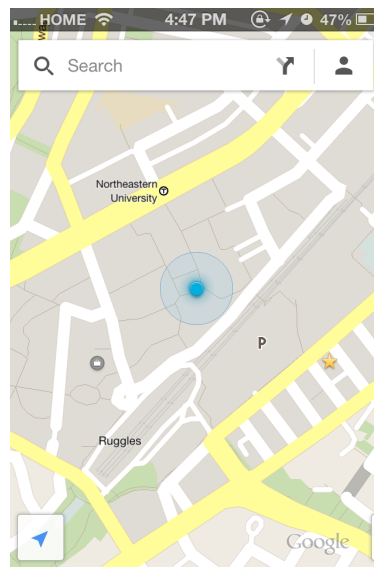

Figure 6. Initial screen of Google Maps only displays one searchbox, three buttons, and one small button.

2.2. Getting Started Guide

Google introduced a significantly different interface and interactions for the iOS version of Google Maps. To help user understanding the new application, Google provides a getting started guide when a user installed the application for the first time (Nielsen's 10th heuristic). However, rather than using standard text, images, and video to explain how the application works, Google created an interactive tour where the user will be guided through experiential learning. Therefore rather than dictating user, the interactive tour encourages user to learn the application directly under a "safe" and guided environment.
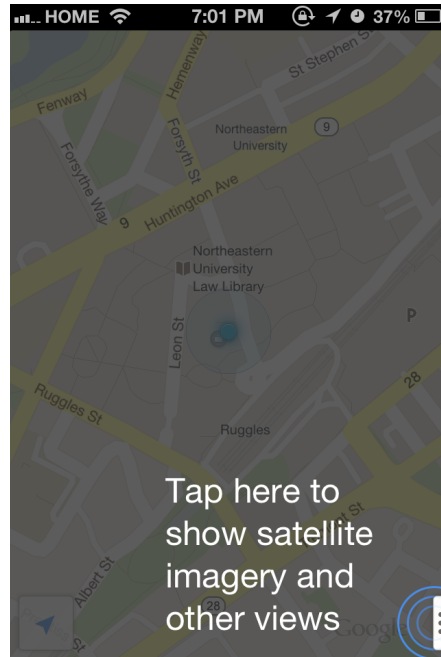
Figure 7. Guided tour encourage user to learn by experience

2.3. Details Card

Google present a large amount of information in the place details card. Initially only the topmost part of the card is visible (Figure 8a). Clicking on the card slides the card up to partially covers the map (Figure 8b). This design implies that the card is on top of the map. When the user slide the card up, the card covered the map entirely (Figure 8c). This metaphor matches how things works in real life (Nielsen's 2nd heuristic), and it helps user to synthesize that the map is under the card and can be retrieved back by sliding the card down.
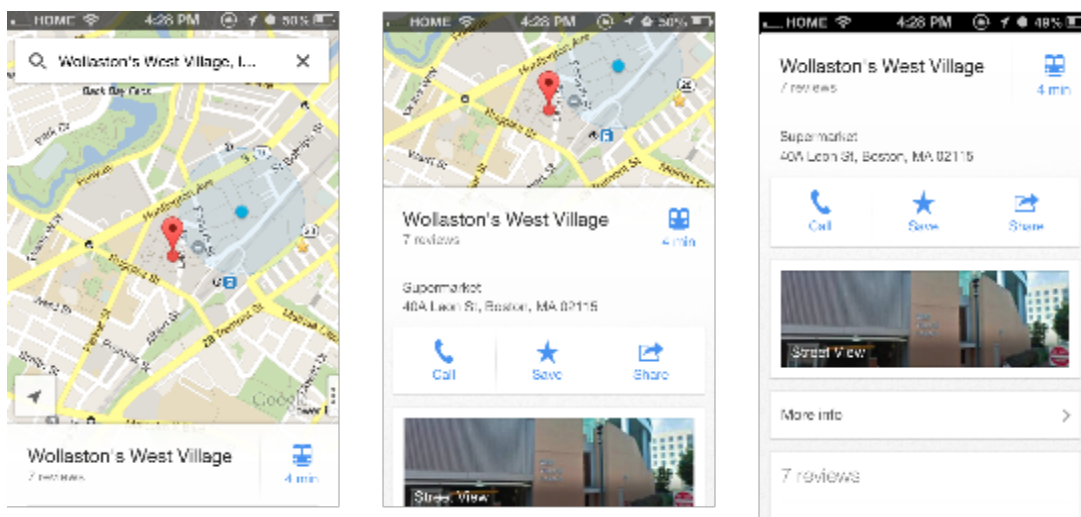

Figure 8a (left) Details card initially displayed at the bottom of the screen. Figure 8b (center) Details card after user click. Figure 8c (right) Details card fills the screen after user slide the

card up.

2.4. Directions

In the web version, finding direction is performed by selecting the starting point and destination, and then pressing the "Get Directions" button. Since mobile devices generally have limited screen space, Google discarded the "Get Directions" button (Figure 9). The route search is automatically initiated after the user have set both the start and destination. This approach simplifies the task (Norman's 2nd principle) as well simplifies the graphic design and saving the screen space for other relevant information (Nielsen's 8th heuristic).
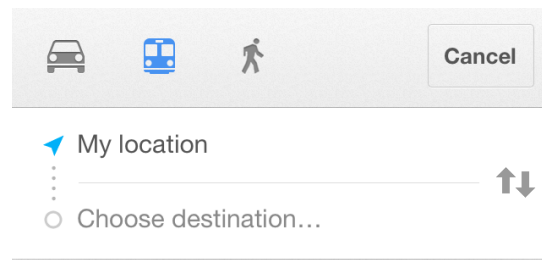

Figure 9. Direction search is automatically initiated when user set the start and destination.

2.5. Search Result

There are many places with duplicate names in Google's mapping database. To prevent user from selecting the wrong place, the places result is complemented with addresses (Nielsen's 5 heuristic). However, the additional information are likely to cause more items in the dialog to compete for user attention. Therefore Google deemphasize the addresses by using smaller non-bold text with less contrast (Nielsen's 8th heuristic).
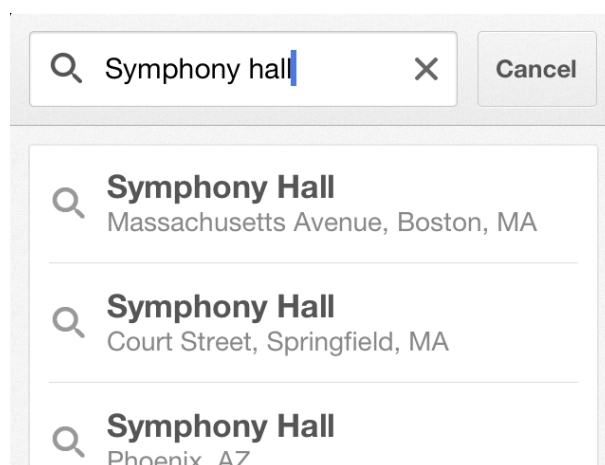

Figure 10. Addresses are displayed using smaller non-bold text with lower contrast.

3. Bad Design: Text Selection and Editing in iOS

iOS is an operating system for Apple iPod, iPhone and iPad. It offers various text selection and editing tools designed for touch based interface, such as text selection, magnification, and auto-correction. User entered text using the virtual touch keyboard that appears at the lower part of the screen. In version 3, iOS introduced text selection features in the form of range selection widget, box selection widget, and selection magnifier.

The range selection widget (Figure 10) consists of start widget and end widget. Start widget appears a blue line placed at the beginning of the selection with a circle on top of it. The end widget is also a blue line but placed at the end of selection with a circle at the bottom of the line. To change the selection, user drags any of this widget.
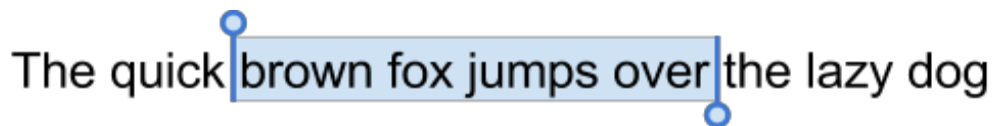

Figure 11. Range selection widget.

The box selection widget (Figure 12), highlights the currently selected paragraph(s) as blue and appears as four circles placed on every edges of the box. To modify selection, user drags one of the circle.

3.1. Selection Inconsistency

When a user select a text, he/she could drag the range widget to cover the entire paragraph. At this point the range widget is transformed to box selection widget. On normal text, user could drag the box selection down and the the selection will be transformed back to range widget. However the behavior is not always consistent in every website. On certain website, after user expands the text selection to box, user cannot drag the the box widget down so they are "stuck" with box selection. iOS did not clearly provide any way to undo the selection.
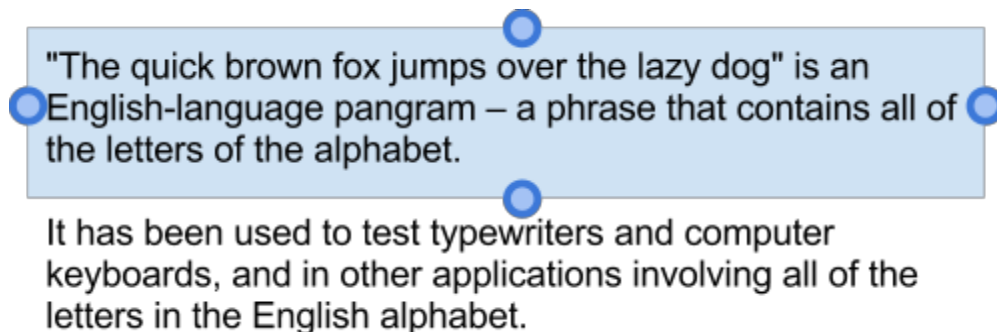

Figure 12. Box selection widget on a Wikipedia article.

This is a compound issue. iOS did not provide consistent behavior on text selection on every website, and when a user experienced an inconsistent result there is no clearly marked exit (Nielsen's 4th heuristic) to recover from the error.

3.1.1. Why it might have been designed that way

This problem occurs on densely organized website. It is likely this problem occurred when iOS could not properly parse the website.

3.1.2. Better design

While parsing issue can be attributed to issues outside of iOS developer's control, this problem can be addressed by providing a means to undo undesired selection, for example by shaking the phone to evoke the undo menu.

3.2. Text Selection Magnifier

To aid range selection, iOS provided selection magnifier (Figure 13) that enlarges the currently selected text. This magnifier appears as a rectangle that appears on top of the circle in the range widget.
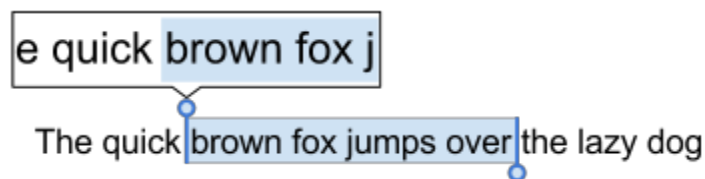

Figure 13. Selection magnifier

Problem arises when a user attempted to select text on top part of the screen, for example in iOS browser's address bar. In this situation, because of the limited vertical space, iOS placed the selection magnifier below the range widget's circle. As a result, some of the text in the magnifier become partially obstructed by user's fingertip (Figure 14). The visibility of selection magnifier decreased further due to the parallax effect when user's eye is perpendicular to the screen's center.

Figure 14. Selection magnifier is obstructed by user's fingertip[1]

User can avoid obstructing the magnifier by selecting the lower part of start range widget (the line). However, Fitts's law indicates that it takes less time to reach larger target than smaller one. Therefore, since the circle's area in the start range widget is larger than the line, it is easier for users to reach for the circle.

This is a physical gulf of evaluation caused by screen size limitation and the nature human's visual system. iOS did not apply Norman's principle to bridge the gulf to ensure that every things are visible, nor constraint the system to ensure that the selection magnifier is always visible (Norman's 3rd principle).

3.2.1. Why it might have been designed that way

The magnification widget generally works well on other circumstances. Since this problem occurred on limited occasion, this bug could be there because of insufficient usability testing.
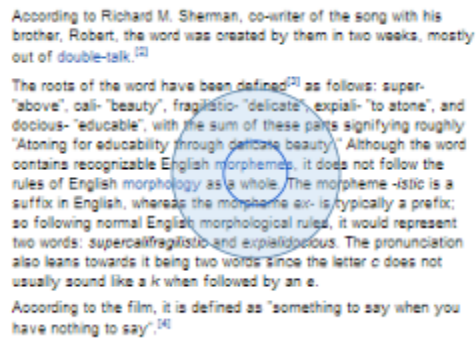
3.2.2. Better design

To help user avoiding placing their finger above the magnifier, the start range widget could be modified so that the circle is placed under the line. Since the circle area is visually larger than the line, user is more likely to tap on the circle and avoids obstructing the magnifier.

3.3. Selecting Small Text

To select a non editable text, user may touch and hold over the text. However, this approach can produce inconsistent result on websites that are not designed for small screens where the text are rendered using small font. When a user is touch-holding on a paragraph with small sized text, a hyperlink and a normal text can fall under touch's center. As a result the resulting behavior depends on iOS's decision whether to prioritize the hyperlink or normal text. Figure 15 shows user touch area (blue transparent circle) over of a hyperlink ("morphemes") and a text ("whole").

---

[1] Pointing hand graphic is a public domain image from clker.com uploaded by Kristen_K. http://www.clker.com/clipart-pointing-finger-5.html

Figure 15. On a paragraph with small sized text, the center of user touch can contain both hyperlink and normal text[2]

This problem is likely to occur. Survey conducted by Google on 1,088 adult smartphone users indicated that 96% of respondents encountered websites that are not designed for small screens [2]. While website owners are also responsible to provide mobile friendly pages, iOS did not made user mapping transparent whether it is prioritizing hyperlink or normal text (Norman's principles 3 and 4). As a result, user experienced an inconsistent behavior (Nielsen's heuristic 4).

3.3.1. Why it might have been designed that way

This is a corner case that occurred when a very small text text is at close proximity with a very small hyperlink. This limitation might occurs due to insufficient usability testing, or ignored due to low occurrence.

3.3.2. Better design

This issue can be solved by having the touch algorithm prioritizing either hyperlink or normal text. While this approach might not satisfies every user, this produces consistent behavior, so the user does not need to wonder what might happen after they tap-hold on a text.

3.4. Text Selection Shortcut

Text can be selected using double tapping, this is a feature that was carried from desktop computing. However this convention is only applicable to text inside a text field. Double tapping on a non editable text (such as in the built-in browser), will zooms in the text instead of selecting it.

This is another inconsistent user mapping. iOS did provide consistent behavior with desktop system when a user double taps a text, however this behavior is inconsistent with users who are familiar with desktop computing behavior.

---

[2] Text is a Wikipedia article licensed under Creative Common.

3.4.1. Why it might have been designed that way

This feature might initially be incorporated to create consistency with the desktop. However due to the nature of touch based interface, double tapping on a normal text was later assigned to zoom feature. As a result double-tapping to select was limited to textbox.

3.4.2. Better design

Double tapping is desktop computing convention, it is better to reserve that feature in the desktop's domain and remove double tapping selection to ensure consistency across the interface.

3.5. Autocorrect

iOS offers automatic spelling correction as user typed in their words. The correction is offered to user as an auto-correct tooltip in the shape of a yellow tooltip with an "x" and placed near the currently typed word (Figure 16). After user presses the space button, iOS will automatically "correct" the word. To prevent iOS from performing the correction, user may tap on the tooltip.
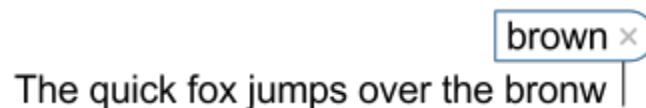

Figure 16. Autocorrect tooltip in iOS

In Windows computing (and desktop computing in general), tooltip is a piece of information that reflects possible change in system's state in the future. A tooltip does not require user action. As a result, Windows user will assume that the tooltip is offering a word suggestion that will not alter system's state. Therefore they are more likely to ignore the autocorrect tooltip. When a user chose to ignore the autocorrect, they will experience an unexpected result as iOS will modify the typed word without any user initiation.

iOS did attempt to provide dialog initiative to user through the tooltip. However, on user with Windows experience, the knowledge about tooltip that is available externally is not consistent in iOS's autocorrect tooltip (Norman's 1st principle). As a result, on some user, iOS can produce unexpected result (Nielsen's 4th heuristic).

3.5.1. Why it might have been designed that way

iOS "corrects" unwanted word when user typed a word not in the dictionary. This usually occurred when a user used english dictionary and typed a non english word. It is likely that iOS developer did not found this defect because they did not test it on bilingual group of user.

3.5.2. Better design

12

To reduce the recurrence of this defect, iOS should allow user to use multiple dictionary for spell correction.

4. Bad UI: Blackboard

Blackboard is a web-based learning management system that have been deployed on more than 6,000 institutions. In Blackboard an instructor can share learning materials and syllabus, hold online discussions, and organize online tests.

4.1. Browsing on a Discussion Thread

To view a discussion, user need to (1) click on the Discussion menu in the sidebar, (2) click on one of the discussion forums, and finally (3) click on one of the discussion thread. After performing the three clicks, user will be presented with Thread Detail page that shows a list of posts in that thread, and he/she need to scroll down to see the first post. To view replies, user need to scroll up and click on one of the replies. Figure 17a shows the Thread Detail page before scrolling, and figure 17b shows the Thread Detail after scrolling.
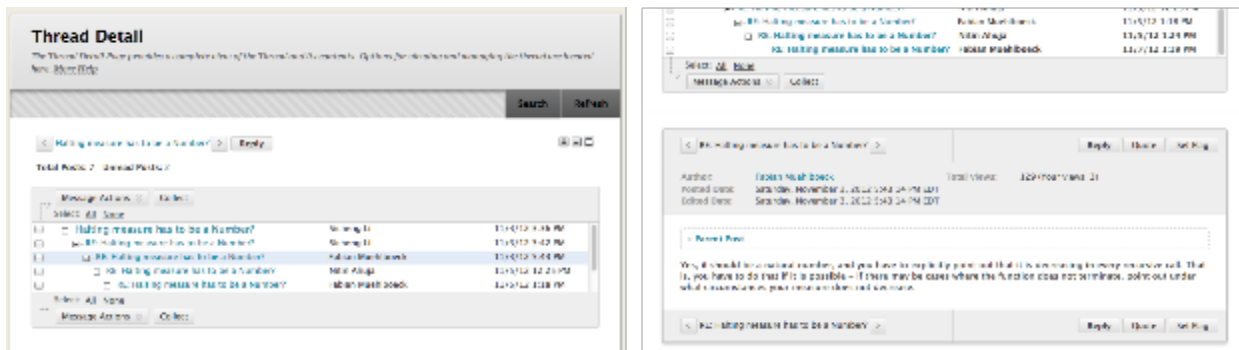

Figure 17.a (left) thread detail page when it was first presented to user. Figure 17.b (right) the same page after user scrolls down to view the first post.

To go back to previously viewed reply, users familiar with web browser convention are likely to press the back button. However pressing the back button did not direct user to the previously viewed page, but direct user to the thread list page.

This is a compound problem. First, the structure of task requires three steps just to view a post. Therefore, a user require more time and memory to accomplish a task (Norman's 2nd principle). Second, the behavior of thread detail page is inconsistent with most websites when a user presses the back button (Nielsen's 4th heuristic).

4.1.1. Why it might have been designed that way

Blackboard seems to employ this interface to mimic traditional email reader such as Microsoft Outlook, to create familiarity with their users. However internet discussion forum is not generally

organized this way. This problem might occur due to insufficient design research.

4.1.2. Better design

A better way to present the thread details is by adopting flattened conversation commonly used in online forums. Since interface is used by almost every online forums, users are more likely to expect discussion organized this way. This also simplify the task by reducing the number of click to browse the discussion.

4.2. Navigation When Searching for Posts

To search for posts, user needs to open the search box, enter the keyword, and press the "Go" button. Blackboard needs a significant amount of time to search for posts on a large forum. Searching the "object" keyword on a forum with 2778 post takes an average of 5.28 seconds. After the search query is complete, user will be presented with a search result page. To view the entire conversation, user clicked on the thread title, and he/she will be presented with the thread page. User familiar with web convention will likely press the "Back" button to go back to the search result page. However user will be presented with error page instead (Figure 18).

Error

An error occurred. Contact the System Administrator.
For reference, the Error ID is bc9350ed–6116–4197–8a49–06d318cf5f15.
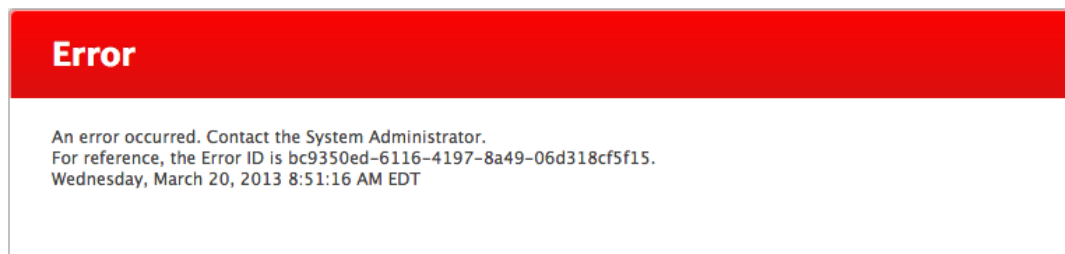Wednesday, March 20, 2013 8:51:16 AM EDT

Figure 18. Error page when user presses back button during search.

This is a compound problem. First, the error message is not transparent. It did not provide any information on the error in the language of user's domain and it did not specifically indicate where to seek help (Nielsen's 1st, 2nd, and 9th heuristic).

Second, Blackboard did not provide any path nor information how to exit from this unwanted state (Nielsen's 3rd heuristic). Third, Blackboard did not prevent user from making this error (Norman's 6th principle), although the error does not destruct user's data.

4.2.1. Why it might have been designed that way

The generic error message implies that the error is unique and Blackboard cannot determine what kind of error it is, so therefore it cannot provide sufficient information for user.

4.2.2. Better design

Blackboard need to build a categorized database of errors, so that every time an unexpected error occurred, the system could tell what kind of error they are having. To improve the error

message, Blackboard should (1) provide system administrator's email so that user can contact them for help, and (2) provide a closure button to send user to the "safe" state. For example, an "OK" button that sends user to the previous page is sufficient.

4.3. Optimizing Search

While searching the entire discussion board takes more than 5 seconds on a large forum, the search time reduced considerably if a user performs search on specific subforum, sometimes to less than one second. Blackboard did not made this visible to user nor provide user the option to search on specific sub forum on the main page (Figure 19).

Figure 19. Blackboard did not tell their user that searching on all subforum takes more time than searching on a subforum.

While long query time is not considered as an error, it is still an undesirable character of a software. There is obviously a gulf of execution when Blackboard cannot deliver search result faster, but Blackboard also did not made their user aware that a search might take time (Norman's 3rd principle). Blackboard also did not made any attempt to prevent user from executing slow search. For example, by setting the default to subforum (Norman's 6th principle).

4.3.1. Why it might have been designed that way

Perhaps due to insufficient usability testing, Blackboard was not aware that the current UI causes user to experience slow search.

4.3.2. Better design

The best solution design would be to optimize the database and search query on the application side. On the user interface side, to prevent user from searching on the entire forum, Blackboard could put all the available subforum in the filter drop-down list and set the default to the first forum (Figure 20)

Figure 20. To prevent user from searching on entire forums, Blackboard could list all subforums in the filter drop-down list.

4.4. Thread list page

The list of thread discussion is presented as a table that shows date, title, number of posts, and number of unread posts. Blackboard attempts to assist their users to spot unread post by highlighting them with blue oval (Figure 21). However, Blackboard also highlight zero unreads.

Figure 21 . Thread list that highlights zero unread post

While this is a forward step to simplify user's task to find unread posts by providing graphical highlights, Blackboard also maps the same highlight to threads with no unread posts. Therefore both read and unread threads competes equally for user attention and as a result diminishing the value of graphical highlight (Nielsen's 8th heuristic).

4.4.1. Why it might have been designed that way

Blackboard designers seem to be aware that making unread count more visible could help user decide which post to read. However the designer doesn't seem to aware that using the same color and shape communicates that both unread and read posts are equal.

4.4.2. Better design

Threads with no unread post should have not been highlighted in blue circle, so that it does not compete with other threads thats needs user action.

4.5. Gradebook

Research by Seldow found that the availability of online gradebook enable students to confirm their course performance and potentially a bolster their sense of self competence [4]. However, by default, in Blackboard the online gradebook is by default hidden under the "Tools" page (Figure 22). Given the opportunity of this feature to improve student's academic performance, students needs to navigate on 11 submenus to discover the gradebook. Sometimes the gradebook is available on the course dashboard, but after several days, the link disappeared.

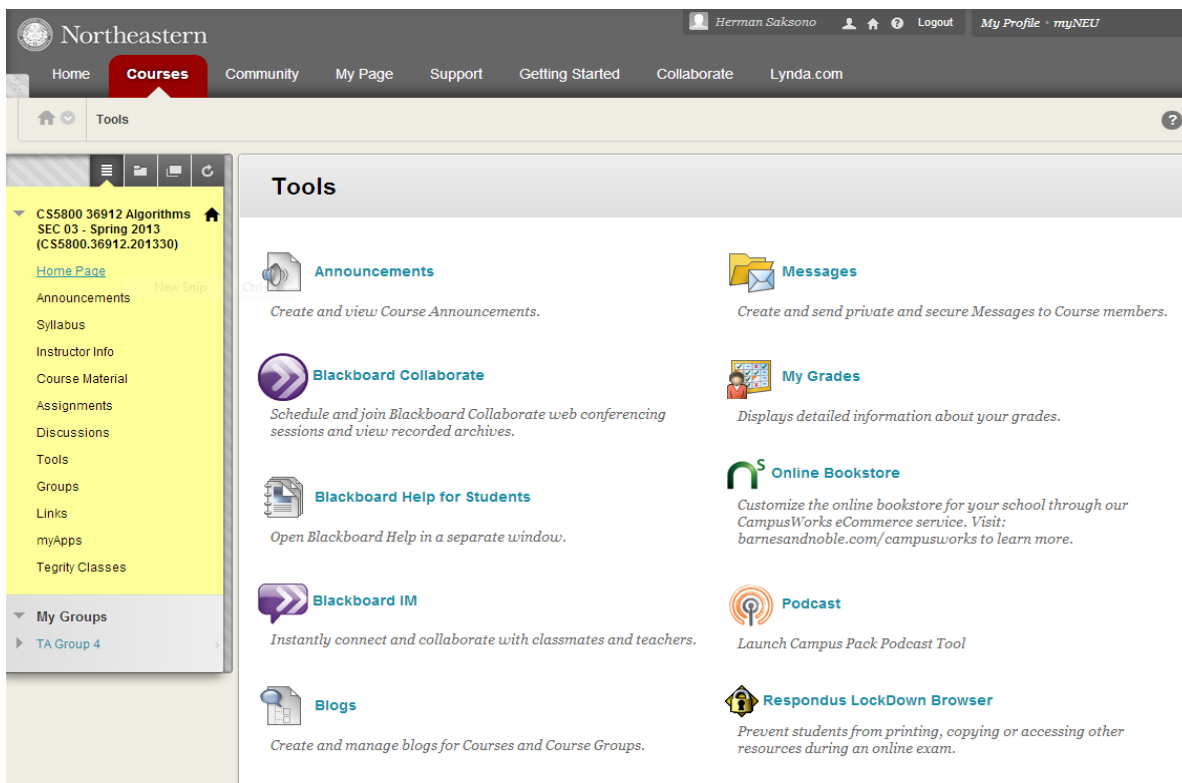A "blackboard my grades" web search on Google shows many people are having problem discovering the gradebook.

Figure 22. Blackboard's gradebook, presented as "My Grades" can be found under the "Tools" page. To discover this, user needs to browse 10 submenu in the left sidebar.

This is an issue of task complexity (Norman's 2nd principle). To discover the gradebook, user needs to navigate 11 submenus at most. This exceed the limitation of human's working memory which is generally at 7±2, based on a paper by Miller. After a user finds the location of the gradebook, he/she will need to remember its location out of 11 submenu (Nielsen's 6th heuristic).

4.5.1. Why it might have been designed that way

My grades seems a feature that was not designed as the main feature of Blackboard. As a result, the link is hidden in another layer.

4.5.2. Better design

To improve feature visibility, the link to My Grades should be placed on the main sidebar by default, and it should be at most the fifth link from the top. This way, the link is always visible to users and user require less memory to discover the link.

References

1. Dix, Alan; Finlay, Janet; Abowd, Gregory D.; Beale, Russel (2004). Human-Computer Interaction: Third Edition. Essex, England: Pearson Education Limited.
2. Fisch, Masha. "Mobile-friendly sites turn visitors into customers". *Google Mobile Ads Blog*. 25 September, 2012. Web. Retrieved 19 March 2013.
    http://googlemobileads.blogspot.com/2012/09/mobile-friendly-sites-turn-visitors.html
3. O'Carroll, Lisa. "Twitter active users pass 200 million". *The Guardian*. 18 December 2012. Retrieved 20 March 2013
    http://www.guardian.co.uk/technology/2012/dec/18/twitter-users-pass-200-million
4. Seldow, Adam Lowell. "On-Demand Grades: The Effect of Online Grade Book Access on Student Mastery and Performance Goal Orientations, Grade Orientation, Academic Self Efficacy, and Grades." *Harvard University*, 2010. United States -- Massachusetts: ProQuest. Web. 20 Mar. 2013.