

Tangelo Product Evaluation

Section I: How well does it work? Does it serve user desire?

This section aims to motivate the sections to follow by providing a brief summary of each of Tangelo's initial UI/UX goals, and the outcome and current state of each goal.

Self-Contained Dashboard: In terms of accomplishing its goal of providing a one-page interface where users can easily create, access, and manage widgets, the current version of Tangelo does its job. Users indicated the integration of all functions onto the single page was properly convenient.

User Customization: User feedback indicated that further customizability, in the form of multiple dashboard color schemes, personal widget styling, widget ownership transfer, making a widget commentable, and viewing multiple posts from a single widget, would improve overall user engagement.

Data Management and Display: Back-end data management (and data sourcing) works as expected. However, there is a mismatch on the front-end widget display, where user created widget posts display a netid as the author handle, whereas custom widgets such as Princeton News utilize the same orange block to display the publish date.

Front-End Functions: Most features respond to user actions as initially intended. One apparent issue for our beta testers has been using the "drag-to-unfollow" feature on the left sidebar; for very large widgets, deletion does not always function as expected.

App Intuitiveness: There definitely did seem to be a gap that we consistently saw throughout the beta tests that showed that many features' intended purposes weren't as obvious as intended, including a lack of general understanding of what a widget even is and what Tangelo's purpose is. In other words, we seemed to not effectively front load information regarding the important aspects of the application.

Section II: Testing the System

Programmer Testing:

Over the course of development, as features were added or removed, the key programmer for the feature, as well as the rest of the team, would immediately test the new function in various ways. This generally held true over the course of the entire development process, and will likely continue to occur if the team chooses to continue development moving forward.

User Testing:

We conducted one-on-one beta tests with 8 Princeton students, from a range of backgrounds. Among these students were also COS majors with prior experience building three tiered projects; these individuals provided particularly useful feedback. We crafted a basic cognitive walkthrough of the system that we requested they follow.

There was slight variability from user to user, but the basic sequence of requested actions was the following:

- 1) Login
- 2) Go through the Welcome Widget demo.
- 3) Dashboard Widget Moving
 - a) Move your first widget.
 - b) Resize your first widget.
- 4) Add the Princeton News Widget.
- 5) What's the description for the Poem-a-Day widget?
- 6) Who created the Clock widget?
- 7) Create your own widget.
- 8) Make your first post.
- 9) Update your post.
- 10) Clear your widget post.
- 11) Remove the Princeton News Widget.
- 12) Remove your widget.
- 13) You have two minutes to build your custom dashboard and make a widget that you care about.

Users were requested to openly talk through each of these actions, and make us aware of any concerns, surprises, thoughts, that came to their minds. This immediately allowed us to get a sense of what stuck out in good or bad ways to users, and what functions seemed obvious, what could have been better explained, etc. We noted down these plethora of observations in our discussions with each of the users, the culmination of which is highlighted under Section III.

Section III: Test Results

System Performance Review:

The following is a **list of features that seemed unintuitive to users**, or less obvious than desired by system developers, in ascending order (top is least obvious to user, bottom is most obvious):

- Click to add widgets from sidebar (rather than dragging)
- Widget movement/resizing
- Description viewed by hovering over 'i' icon (rather than clicking)
- Post making is actually post updating (rather than being able to make multiple posts)
- Created widgets are public for all to see

The following is a **list of components of the system that the users enjoyed** or could see themselves enjoying:

- Following and playing/joking with widgets of friends
- Crafting a personal dashboard
- Complete freedom to move and adjust widgets on the dashboard
- Easy access to automated news headlines that can be clicked to view the source article
- Using Tangelo as a replacement for email listserv spamming
- All information on one interface
- All functionality on a single page

System Feature Suggestions:

In addition to an evaluation of overall system performance, we have garnered a comprehensive list of features that users asserted would make the product more attractive to them, in no particular order:

- Login screen displaying a blurb that indicates more holistically what Tangelo is
- Some sort of to-do list or planner, custom to each user
- Night mode/user customizability of dashboard color scheme
- Categorization of widgets in the search bar, and under widget creation (i.e. widget tags)
- Allow widget creators (admins) to custom style their widget. For example, the calendar widget displays information linearly. Users expressed interest in alternative forms of information display.
- Ability to create private invite-only widgets
- The creation of 'official' widgets for clubs for guaranteed official content
- Ability to add pictures, animations to dashboard backgrounds
- Platform availability as a mobile application
- Multiple dashboards; the ability to 'save' a dashboard configuration
- Google calendar integration

- “Next event” widget
- Communal widgets that multiple users can post to
- Make the current shape of the dashboard stable upon adding new widgets (currently widgets get shifted around when a new widget is added)
- Click deletion of widgets (on right sidebar under My Subscriptions)
- Multiple posts or post history on widgets

Bugs:

The following bugs listed are still present in the system. They are listed in no particular order.

- Drag and drop widget deletion functionality is not consistent: sometimes, widgets are not removed from the dashboard if they are too large, relative to the deletion area
- Widget deletion is possible if a widget is dragged over the area where the trash can icon would be if the Follow sidebar were opened, even if follow sidebar is collapsed
- Widget follow list not having a scroll bar for some users until they actually begin to scroll on it
- Widget follow list bottom widget’s description not showing
- Date and time widgets don’t wrap when minimizing widgets/resizing
- Subscription list does not update when deleting widgets until the page is refreshed

Section IV: Heuristic* Analysis of Results

*Breakdown of Nielsen’s Heuristics from *Usability Engineering* retrieved from Piazza. Bold text is original retrieved text.

(1) Visibility of system status. The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

a. System Alignment

- Widgets: Currently, the “Weather” is the only widget with a functional delay between page load and completion of the weather API request. While retrieving this information, the widget displays the message “One moment, we are grabbing some weathery goodness”.
- Action Info: We integrated Flask’s Flash module to flash messages at the top of the dashboard upon various user actions, to keep the user informed of the status of their committed action. These messages are removed from the dashboard after 4 seconds. Messages appear after each of the following actions:
 1. Post Creation
 2. Widget Creation
 3. Widget Deletion
- Admin Drag Deletion: We do not allow users to delete admin widgets by dragging them to the trash icon, and we inform users of this when we reject the deletion, so the user understands this rule and can then properly go to the “Admin” tab under the “Manage” sidebar if they truly want to delete their widget.

b. System Misalignment

- i. Widget Status: Currently, if a widget creator has cleared their post, the widget remains empty for all subscribed users with no indication as to whether the widget is not properly working or the author has not made a post. A fix for this would be to clarify widget status. When empty, a message of the likes of “The author has not yet made a post for this widget” would be a quick solution.

(2) Match between the system and the real world. The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

a. System Alignment

- i. Iconography: The trash icon, Make New Post icon, and source URL link icons all match standard icons used for similar functions.
- ii. Grid Drag and Drop: On widget drag, a grid boundary clearly indicates the limits of the grid to the user. The concept of organizing a physical space by moving objects around is aligned with the drag and drop feature.
- iii. User Handles: When a post has been made to a widget, it has the author's name with an “@” before it, which is the conventional way of signaling that the username that follows is the ‘agent’ of an action.

b. System Misalignment

- i. Terminology: There is unclear and inconsistent terminology used in the product.
 - 1. “Widget” was demonstrated to not be self explanatory to beta testers and may need a more common replacement term to indicate the different packets of information users can drag, post to, read from, etc.
 - 2. The act of adding a widget onto your dashboard is called ‘follow’ in one location, ‘add’ in one location, and ‘subscribe’ in another. More consistency would be beneficial in this case.
 - 3. “Manage” title of the right sidebar does not fully indicate what it's function is. Some questions that arose: Do we manage a widget or our dashboard? Why is widget creation under management?
- ii. Widget Search ‘i’ Icon: Usually, if a search result or any button has an ‘i’ next to it, you have to click the ‘i’ to see extra information, not hover. Currently we have hovering implemented, but this confused many testers.
- iii. Add Widget Drag: Currently, we have implemented clicking to add widgets from the “follow” sidebar, but it is more intuitive, particularly because the rest of the interface is drag and drop, for the widgets to be draggable from the search results onto the Dashboard. All testers naturally tried to do so but couldn't.

(3) User control and freedom. Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

a. System Alignment

- i. Admin Deletion: If a user accidentally drags and deletes a widget they are an admin for, the deletion will fail; must delete admin widgets from secure admin tab in “Manage” sidebar. Admin deletion from the appropriate place also has a confirmation prompt to protect accidental deletion
 - ii. Post Edits: If a user makes a mistake on a post to a widget, it is easy to fix the mistake as “Make New Post” populates the entry area with the most recent post. For example, updating the time for an event is simple.
- b. System Misalignment
 - i. Post Change Regress: If a user changes a post and saves the update, there is no means of retrieving the most recent post.
 - ii. Widget Name: Upon widget creation, there is currently no way for the creator to change the name or description of their widget. Admins have to delete the widget and create a new one, which is inconvenient.
 - iii. Widget Ownership: Widget admins cannot transfer ownership of their widget, or add additional administrators.

(4) Consistency and standards. Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

- a. System Alignment
 - i. Widget: The term “widget” is used consistently in all locations on the app
- b. System Misalignment
 - i. Many misalignments carry over from Heuristic #2
 - ii. New Post: “Make new post” feature replaces the previous post with new one, and in dialogue box says “save post” rather than “make post”, so small disparity between creation and updating
 - iii. Calendars: Calendar dates are styled the same way author names are elsewhere, which is inconsistent

(5) Error prevention. Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

- a. System Alignment
 - i. Admin Deletion: Users cannot delete an administered widget without confirming they want to continue. They are clearly reminded of the consequences of deletion: specifically that deletion also results in a cascade deletion of the widget for all subscribed users.
 - ii. Widget Creation: The widget creation form has back-end validation, and expressive error handling and display.
- b. System Misalignment
 - i. Post Creation: When a user makes a post, there is no confirmation; posts are immediately displayed on the widget upon one click of ‘Save Post’.

(6) Recognition rather than recall. Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

a. System Alignment

- i. Welcome: The “Welcome to Tangelo” widget with the site demo is always accessible, so the user can revisit the demo if necessary.
- ii. One Stop Dashboard: All Tangelo functionality is contained within a single, minimalistic page, promising an inherently smaller memory load for the user.

(7) Flexibility and efficiency of use. Accelerators—unseen by the novice user—may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions

a. System Alignment

- i. Widget Deletion: Widget deletion is instantaneous and efficient; users can delete widgets very quickly through the drag and drop feature.

b. System Misalignment

- i. Multiple Adds: The “follow” sidebar closes and the page reloads each time a user adds a widget to their dashboard, which makes adding multiple widgets at once inefficient and frustrating.
- ii. Layout Shift: A user’s dashboard setup shifts upon addition of a new widget, requiring the user to modify their setup, which is inefficient and frustrating.

(8) Aesthetic and minimalist design. Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

a. System Alignment

- i. Minimalist Design: The dashboard design is fairly minimalistic; there is limited use of excess color, fonts, or components that are devoid of integral function. The design is also consistent across the site.

b. System Misalignment:

- i. Subscriptions: Under the “Manage” sidebar, the “My Subscriptions” tab details the various widgets a user is subscribed to, but offers no additional function. This feature is currently redundant, as the user can easily view all current subscriptions on their dashboard.

(9) Help users recognize, diagnose, and recover from errors. Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

a. System Alignment

- i. Admin deletion: Users cannot delete widgets they are admins of by dragging the widget to the trash can icon. We remind the user of this enforced protection.

- ii. Error Handling: All errors flashed to the user never display code. In the event of an error, messages are often of the form “something went wrong with this action”.

(10) Help and documentation. Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

a. System Alignment:

- i. Welcome: The “Welcome to Tangelo” widget is simple, compact, with limited steps, and always accessible to users for a quick walk-through of Tangelo's features.

b. System Misalignment:

- i. Login Impression: The Tangelo login page does not motivate the role or purpose of Tangelo. New users may be left wondering as to the nature of the app. An info section, or short blurb, would remedy this issue.