

OutWrite

Marisa Yamasaki -- Writer, Graphic Designer
Alisha Meherally -- Writer, Prototype Developer
Kevin Jung -- Writer, Outreach Coordinator
Fengjun Dong -- Writer, User Researcher

While writing fiction requires detailed development of plots and characters, there are few applications to help writers organize the details of their stories. Most authors use word processors, but many prefer to handwrite their initial ideas for better creative stimulation. Authors often write notes throughout their day on notebooks or mobile phones when inspiration strikes. These factors result in divides between digital composition, notes on a mobile device, and physical notes. Furthermore, common notation technologies such as Microsoft Word and OneNote provide no organizational assistance specific to developing plots and characters. OutWrite proposes a way to store notes, as well as organize plot and character development, that could be easily tailored to an author's routine. OutWrite's QuickNotes feature enables authors to enter notes in their preferred format from a mobile device, either typing on the screen, writing with their finger or a stylus, or embedding a picture of a handwritten note. All input formats are recognized by the application as text, so authors can store their notes in a centralized platform without losing the creativity of hand-writing their thoughts. OutWrite would also provide the OutlineIt feature for organizing characters and plots in a more specific way than a generic word processor. Authors could add character bios with links to instances in their story's plot. The plot would be mapped on a plot graph, with plot points storing detailed information and containing links to related characters and notes. Since most authors write on a computer word processor, OutWrite would have a desktop extension that would automatically sync information with the mobile application.

Initial Paper Prototype

Our initial paper prototype had two buttons on the opening screen, QuickNotes and OutlineIt, each leading to a solution for one main task. The QuickNotes button opened a new note, allowing people to jot down their ideas quickly without bypassing many layers of interface. From the new note, writers could view all notes to see and edit their old notes.

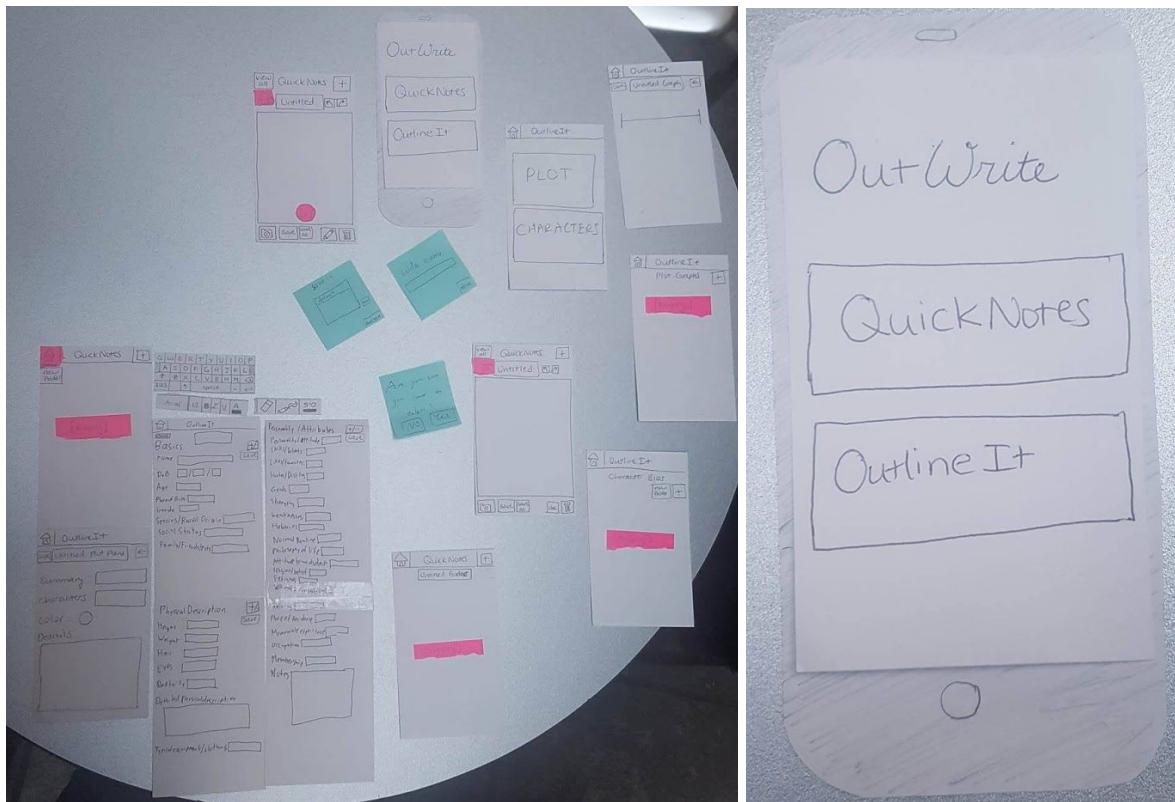
We allowed a one level hierarchy of folders so that people could sort and classify their notes based on categories of their choosing. Notes autosaved into a top level default folder, but if people wanted, they could click a "save as" button to move the note to a different existing folder or create a new folder. They were also able to create folders and move notes into different folders in the notes library tab, which allowed for easy movement between sections.

The other major task supported by our initial prototype was the outlining tool, opened using the “OutlineIt” button on the main screen. It opens to a menu that asks the user to tap one of two buttons, one for characters and one for plot. Either button you click on will lead you to the library of either the characters or plot graphs, with a small button to create new ones. Unlike note taking, creating plots and character descriptions are not as frequent or time-sensitive, so we made the libraries central to this part of the application rather than creating a new plot or character immediately.

The character bios page had a large number of scrollable and collapsible fields, none of which were required to be filled out. The only required field was the name. The plot graphs page has a plot graph where people were able to click on the graph to add a plot point, which had its own fields with customizable colors and characters.

Our first main task was to allow writers to be able to quickly take notes and be able to edit those notes at a later point in time. Our other main task was to allow them to keep track of their characters and different points on their outline. The OutlineIt tool takes care of the second task, and the first task is covered by the QuickNotes tool.

Overview of our initial prototype:

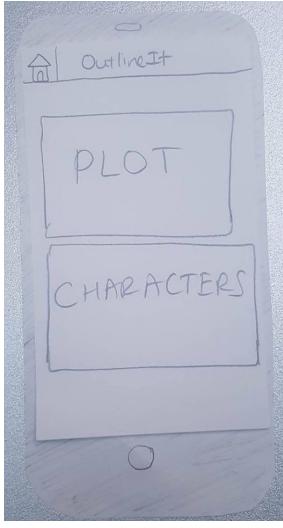
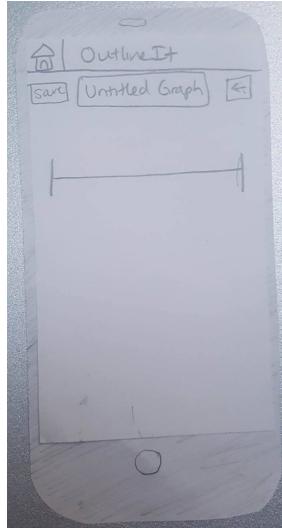
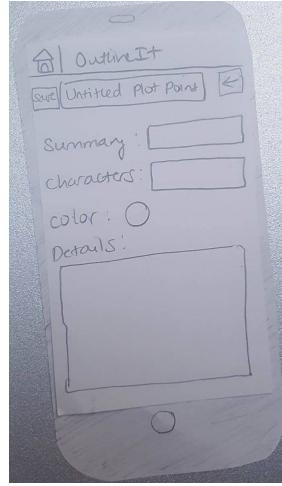


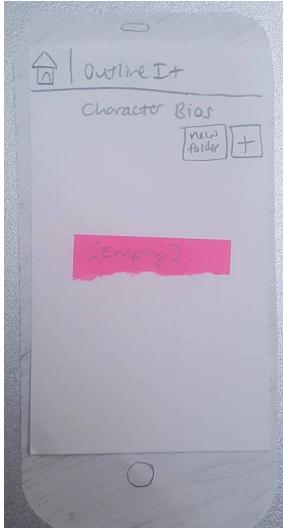
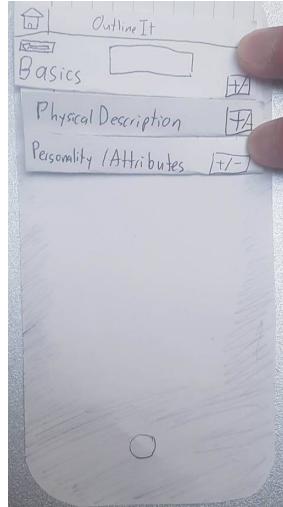
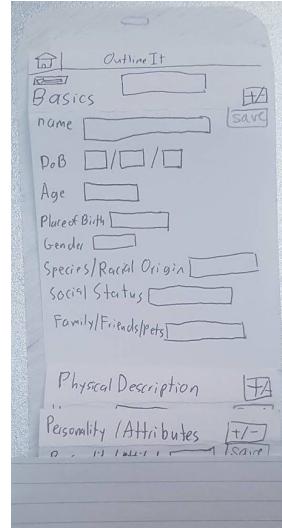
Task 1: Creating a new note and embedding a photo

Home screen of the application.	Tapping the QuickNotes button opens a new note in draw mode	Changing the mode to typing (abc button) brings up a keyboard	Tapping the camera icon brings up a camera to embed a photo.

Tapping "save as" allows the note to be saved in a different folder	Tapping "View All" shows all notes.

Task 2: Creating a plot graph and adding a plot point, and creating a new character

			
Tapping OutlineIt opens the outline menu	Clicking plot opens the plot graphs library	Tapping + creates a new empty graph	Tapping on the plot line creates a new plot point

		
From the OutlineIt menu, tapping character opens the character library	Tapping + creates a new character	Tapping +/- on the right opens character fields

Testing Process

Our testing process consisted of two heuristic evaluations and three usability tests. We used our initial paper prototype in the heuristic evaluations, then completed a major re-design of our interface before beginning usability tests. In heuristic evaluations, we had two teams of evaluators go through the app, and use the different functions. We helped them along the way as they needed it and answered questions they had about the app. These people were other members of the class, and were very familiar with the design experience. They had no writing background, but they had a computer Science background. Their comments and feedback had a major impact on our design, prompting a full redesign before we began our usability testing. We asked our Heuristic evaluators to explore the app and ask questions about what each feature did, and they brought up many inconsistencies and illogical features that we, as designers, had been unable to see. Their feedback is included in the Appendix.

After conducting heuristic evaluations, we redesigned our app to follow the feedback we received before beginning usability testing.

Our first usability test was conducted with two participants – one each for tasks 1 and 2 – due to our first participant having to leave early. Both participants were students in the computer science building lobby.

At the beginning of each test, we provided some background on our project and explained to our participants what the goal of each task was. Task 1 was to navigate from the home screen to create a new note with an embedded photo. Task 2 was to navigate from the home screen to create a new plot outline with one plot point, as well as create one character. We found that it was very important to explain the tasks very thoroughly in the beginning to ensure that the participant has all the information they need to complete their task. We discovered that when a participant was not familiar enough with the purpose of the application, they struggled to complete the task in ways that were unrelated to the interface. Since we were testing the application, not their competence, we realized the importance of explaining the concept of our application before beginning testing.

We conducted two more usability tests, this time on students in the computer science labs, using the same interface as the prior test, and the same tasks. We explained the details of the tasks and gave them scenarios so it was easier to understand how we wanted them done. Prefacing the tasks with scenarios helped the participants better understand the concept of our application, and they quickly accepted their role as hypothetical writers planning a novel. When testing with the first participant, we did not explain the concept of creating notes to jot down

ideas well, resulting in the participant wandering off into the outlining tool to look for the camera. Once we gave the participant a brief review of what the task was intended to accomplish, they realized their confusion and immediately returned to the correct section of the application. Learning from our mistake, we explained the idea of an author jotting down notes in greater depth to our second participant, still without revealing information about the interface. As a result, the second participant was more confident in completing the tasks.

Reviewing our first usability test, we decided that we were providing too much help to the participants. We refined our process by framing our tasks in scenarios, giving the participants more background on what they were intended to accomplish. These, along with the design changes made using the advice from the heuristic evaluation helped later usability tests be more successful.

Testing Results

We had several refinements we had to make in our process, and we have three paper versions as well as our digital mockup. In our first version, we had no real standardization. We sketched out what each screen we would need would look like, and what dialogs we would need based on our storyboard and how we thought it would work best. This wasn't a perfect approach, but we were able to get an idea of how we wanted the app to flow and approximately what we wanted each screen to have. Our heuristic evaluation got us the feedback we needed for our major overhaul, and we not only fixed the issues we were told about, but we standardized the whole interface. We conducted three usability tests with this new interface and received some feedback, but a more minor amount of criticism as compared to what we had before, and we did make several changes and refinements, but nothing as drastic as our prior overhaul. This final version takes the feedback from both versions of the app and is clearer to understand and more streamlined for both new and experienced users.

We had many problems with our first prototype, mainly since participants in the heuristic evaluations recognized many flaws in the interface that did not occur to us as designers. One problem was lack of handling whether multiple notes could have the same name. Ultimately, we decided to not allow two notes with the same name, and required unique names. Another issue we had was that our text boxes were too small, and it was hard to tell how they would expand or how people would be able to read or add text to them, so we decided to ensure that our design had bigger fields and a scrollbar would appear in the box when the text was bigger than the design. We also standardized the back button to be on every screen, added help menus, added an undo and redo to screens where users were able to add text. A major standardization change we made was to keep the layouts between the notes and outlines features the same, so that tapping

the note button took users directly to the notes library, as tapping the plot or characters button would do in the outline section. However, we have made the “new” buttons very large so writers could quickly make a new note when they had an idea. We also refined the plot summary portion so that people can add characters related to the plot. This does, as a consequence, require people to create character biographies for each of their characters. On this point, we also refined the character biography page to be less cluttered and allow people to add whichever fields they feel like to each biography, as well as allowing them to import standard fields from a template. Another major part of our redesign was to get rid of our folder system, because people kept wanting to make subfolders, which we did not support. As a result, we now have a system of tags, allowing people to add as many or as few tags as desired to a note. We have a top level book, defaulting to “MyBook,” that allows people to create separate versions of the application if they are writing multiple books. Data is compartmentalized, but is able to be moved between books in the settings menus. This also prompted some redesign changes for the save feature, allowing for a removal of a confusing “save as” feature to change folders and adding a Done button with an autosave feature. A final feature we changed was to add a settings pop up menu so as to lessen the clutter and small buttons, and allow for larger, more important buttons to be more large and prominent in the app.

When we got into the usability tests, we saw that people had an easy time figuring out how to create a new note with a giant button. However, the buttons on the bottom were a little hard to see, and the embed photo button was a bit of a trouble spot, because it was text, rather than a camera icon that seemed to be what they expected. As such, we changed it to a camera icon and created large buttons. We also discovered we needed a function to be able to delete a photo, by tapping and holding on it until the option popped up. We also found out that tags are a more advanced feature, as only one our three users was able to explain what it meant. Additionally, we discovered that the keyboard either needed a done button or should disappear when the person clicks outside a text field. Our new bottom panel was more easily maneuverable than the first design, other than the photo button, and the help tips were useful. We added a popup that prevents users from exiting a character profile without giving it a name, since characters are added by name to plot points. Another feature we changed was the templates for character biographies. We allowed users to be able to add and remove character traits, which we initially called “fields,” from the template in settings, and let them check/uncheck all on the selection screen. We also realized we needed to implement a drag and move feature for our plot points, so that users can change the locations events on the timeline. We also discovered that our eraser function got lost between versions, and we needed to add it back in. Additionally, we found out that users wanted to sort notes by not only tag, but other features, such as date and time created.

The changes made were far more drastic the first time as compared to the second. We discovered that even though we had to make a major overhaul before usability testing, we still had a lot more minor issues to sort out later, including a few changes that we had to revert, such as how we had a camera icon before, but we changed it to “embed photo”, which only wound up being more confusing. We learned though that sketches of screens do really help figure out how a screen should look before actually starting at it, and that different users see the app differently. Some users are far more experienced with using digital applications, and, as such, may be better able to determine what each part should mean and do. One of our users breezed through the app, even using the tag feature, yet another user went off on a totally different track, and made a new plot graph when trying to embed a picture in a note. This happened in the same iteration, and users were both found in the same place, and presumably both have a computer science background. This is something we have to worry about in the real world as well, as our user base may not have as much experience with using applications and may not have the computer science knowledge.

Final Paper Prototype

Our final prototype is a lot more like our second iteration than our first. We kept a lot of the general features and appearances the same, but made a lot of changes to the app as a whole. We kept our design to our two primary tasks, and our refinements streamlined the process significantly as a whole

A major feature we implemented was the book organization feature, which allows users to separate their notes, characters and plots based on which book they belonged in, thereby eliminating the need for a folder system. The app defaults to a MyBook for users, but allows them to rename it and create more as needed. The app also has tags for additional classification of notes.

As our goal is to help writers keep track of their ideas and quickly write notes on the go, the QuickNotes feature allows people to click on the quick notes button and easily see their existing notes. They can also add a new note by clicking on the new note button. This button is purposefully very large so that people are able to easily add new notes to the app. Users are able to add text, embed photos and draw in a note, and they are also able to add tags for easier discovery in the future.

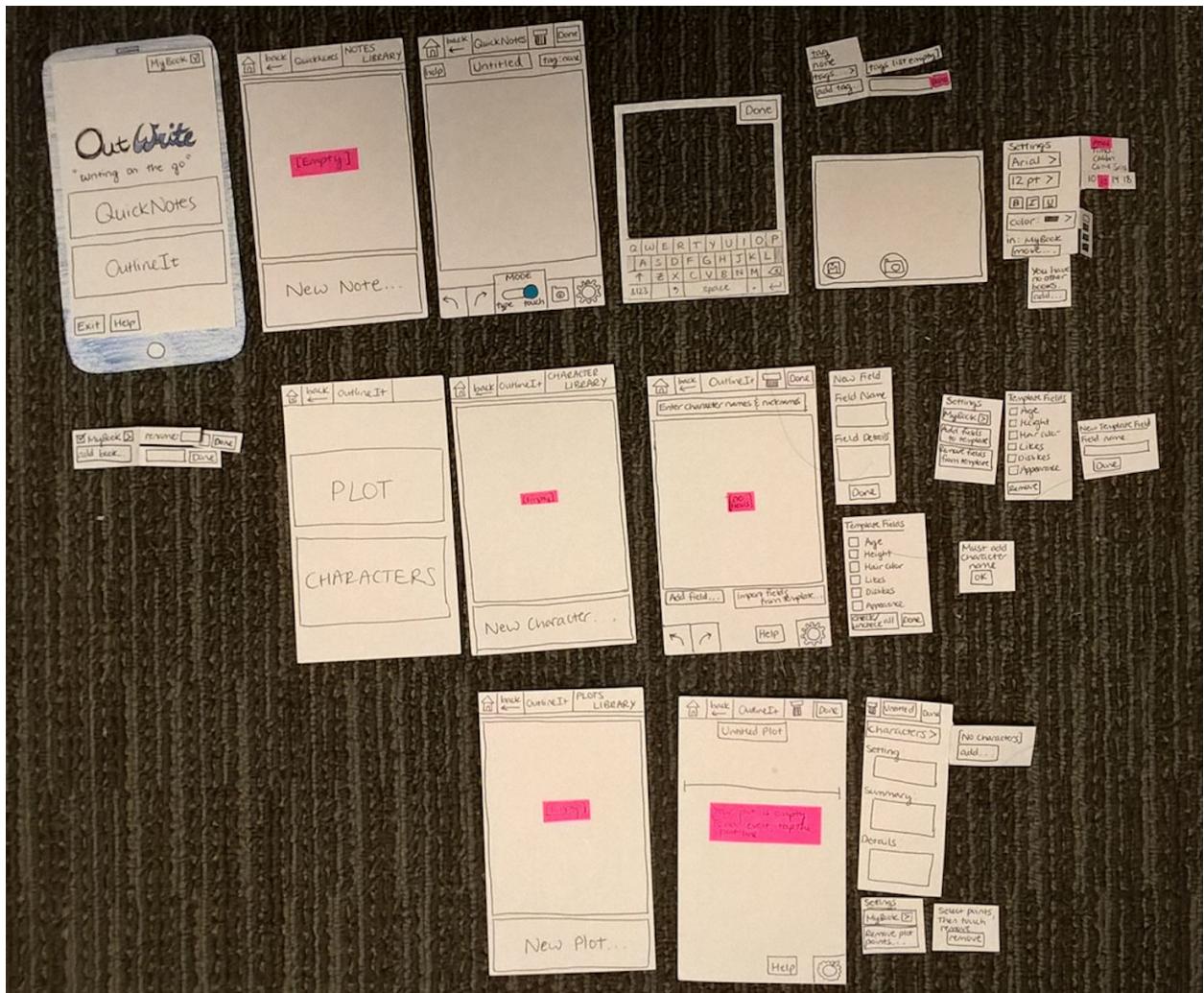
The other major feature is that people are able to click on the OutlineIt tool to be able to access the character biography and the plot summary page. In the biography, they are able to add characters and view existing characters. Once they create a new character, they are able to add

fields to that character, to give the character descriptors. There is also a template they can use if they want to take some of the common fields that they would use. This template can be modified, so that the users are able to customize their biographies if, for instance they live in an ageless society and don't want it on their template but want everything else. The only thing that is required of a character is a name or nickname, entered in the top title bar.

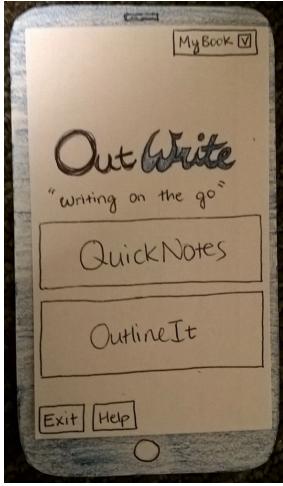
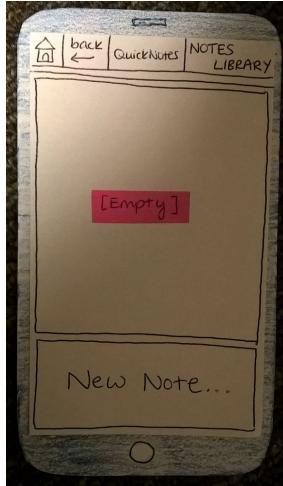
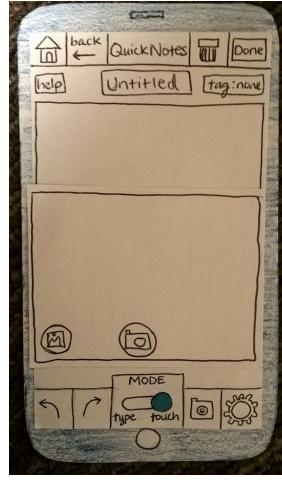
In the plot section, users are able to make plot graphs and add various points to the plot graphs. The plot points are the main events of the story and can be as descriptive or vague as the user wishes. They also are able to add characters that show up at each point, from a drop down checklist. The characters are hyperlinked to the biographies and users are able to add new biographies from the plot points screen.

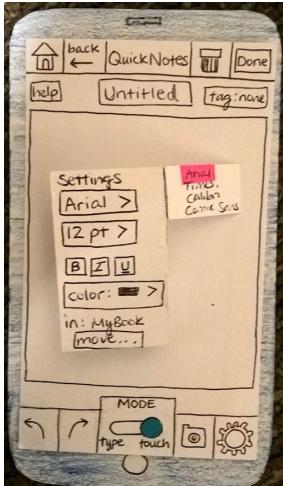
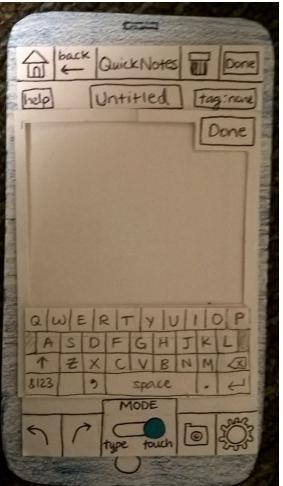
Our two main tasks were to allow users to quickly take notes on the go and store those notes as well as to allow them to keep track of their plot and their characters. This app fulfills both of these tasks. The first task happens in the QuickNotes tab, and allows for the user to easily compartmentalize their notes and take them in a variety of formats. The second task happens in OutlineIt under characters and plots, and allows users to give a detailed plot description and character biographies and store and organize these for later use.

FINAL PROTOTYPE



Task 1: Creating a new note with an embedded photo

			
When the app is opened, the home screen is displayed.	Clicking the “QuickNotes” button opens the Notes Library, where all notes are stored.	Clicking “New Note” creates a new untitled and blank note. Input mode can be changed between touch and type with the bottom slider.	Clicking the camera icon allows a picture to be taken and embedded in the note.

	
Clicking the settings icon enables changing font style, size, and color.	Clicking the title bar to add a title brings up the keyboard.

Task 2: Creating a plot graph, adding a plot point, and creating a character.

Clicking the home button returns the application to the home screen.	Tapping the “OutlineIt” button brings up the plot and character menu.	Tapping the “Characters” button opens the character library.	Selecting “New Character” creates a new unnamed character.

Clicking the “Add field” button brings up a pop up window to add a field to the character description.	Tapping “Import fields from template” brings up a pop up window to add pre-programmed fields.	Selecting “Settings” allows adding and deleting fields from the template, as well as changing the book to which the character belongs.	Adding a new field to the template.

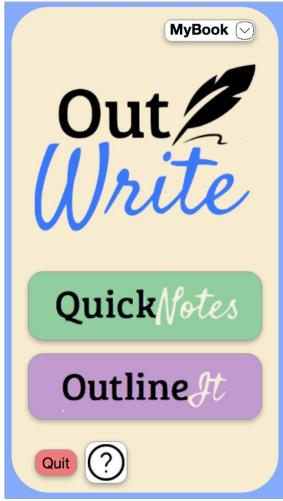
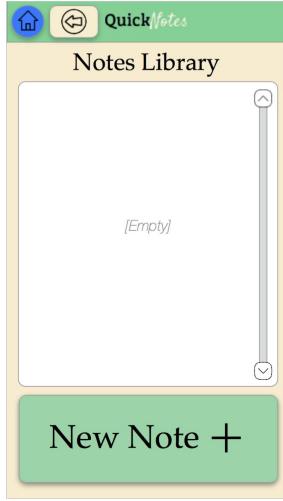
Deleting fields from the template.	Navigating home->OutlineIt->Plots opens the plots library.	Selecting "New Plot" creates a new plot timeline.	Click the timeline to add a plot point; select characters involved and add descriptions.

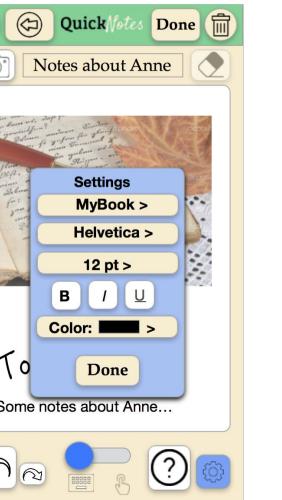
Clicking the plot point fields brings up a keyboard for typing.	The settings icon brings up a menu to change the book in which the timeline is located or delete a plot point.	Delete a plot point by tapping the point and then hitting "remove."

Digital Mockup



Task 1: Creating a new note with text and an embedded photo

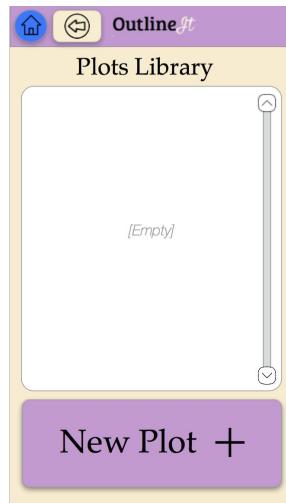
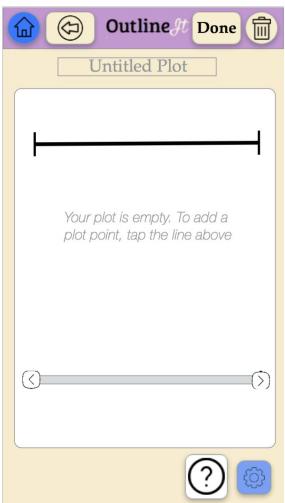
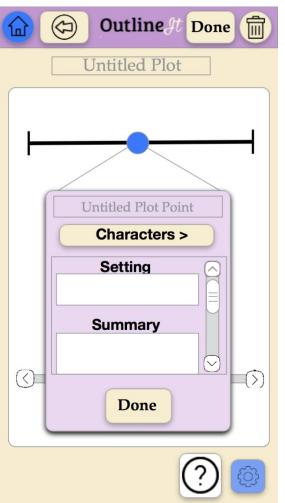
			
When the app is opened, the home screen is displayed.	Clicking the “QuickNotes” button opens the Notes Library, where notes are stored. The library is initially empty.	Clicking “New Note” creates a new untitled and blank note. Input mode is initially finger-touch mode.	Clicking the camera icon brings up a camera to embed a photo.

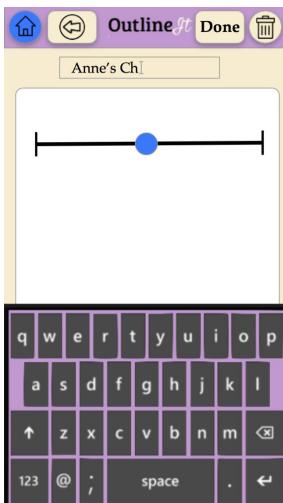
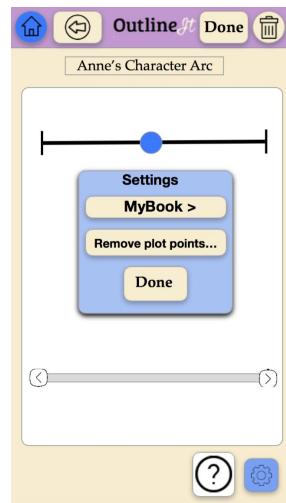
			
The photo taken is embedded, then drawings or words can be added.	Changing the mode to type using the slider brings up a keyboard.	A title can be added on the title bar.	The settings icon enables changing font style, as well as the book where the note is stored.

Task 2: Creating a plot graph, adding a plot point, and creating a character.

Clicking the home button returns the application to the home screen.	Tapping the “OutlineIt” button brings up the plot and character menu.	Tapping the “Characters” button opens the character library.	Selecting “New Character” creates a new unnamed character.

Clicking the “Add character trait” button brings up a pop up window to add a trait to the character description.	Tapping “Import character traits from template” brings up a pop up window to add pre-programmed fields.	Appearance after traits have been added.	Clicking the name bar enables typing names and nicknames.

			
Selecting “Settings” allows adding and deleting traits from the template, as well as changing the book to which the character belongs.	Navigating home->OutlineIt->Plots opens the plots library.	Selecting “New Plot” creates a new plot timeline.	Click the timeline to add a plot point; select characters involved and add descriptions.

	
Clicking the plot point fields brings up a keyboard for typing.	The settings icon brings up a menu to change the book in which the timeline is located or delete a plot point.

As we interpreted our design digitally, we had to make decisions about graphics and color that we did not include in our paper prototype. In accordance with the Category Pages pattern described in *Design of Sites*, we used separate header colors for the two parts of our application, QuickNotes and OutlineIt. To ensure internal consistency, we maintained a blue and pale yellow color scheme throughout the application. We also had to replace hand-drawn symbols with more polished icons, which we chose from the Noun Project website. Other decisions we made included text size and button style.

In response to critique, we added an eraser button to erase drawing and handwriting. We included an eraser in our initial prototype, but forgot to add it to later versions after our interface changed. We did not catch this oversight during usability testing, but we noticed it during design critique and added it to our mockup. From feedback during critique, we realized that the fundamental buttons (home, done, etc) were not in uniform locations across the application, so we standardized these features on each page.

Our digital mockup application supports our tasks in the same fundamental ways as our final paper prototype, but with greater graphical clarity to make the interface more intuitive. Different colors indicate different sections of the app, and greyed out letters and scroll bars show when a box is empty. Pop up windows are displayed more clearly using graphical shadowing, and the digital icons are more recognizable than their sketched counterparts.

All of our icons are from the Noun Project website, and the photograph depicted in the camera feature is a stock photo ([Autumn Diary — Stock Photo © andrejad #6736342](#)).

Discussion

As we developed our design, we learned that many iterations are necessary to refine an interface. We started designing our prototype by sketching a state machine with the pages as nodes, to explore navigational flow. The results of this exploration were captured in our first prototype. Despite our efforts to create a logical design, participants identified many inconsistencies, holes, and logical fallacies, and we realized the importance of having an outside perspective of our work. As designers, we were inclined to imagine our interface was logical and unaware of problems that were obvious to outside participants.

Feedback from heuristic evaluations helped us understand the importance of logical organization. Since participants got lost in the many layers of our interface, we decided to centralize each part of the application around a library and allow creating only one new document at a time from that library. Initially, we based our design on a Windows-like folders system, but participants were confused when they expected a complete Windows interface. We then explored the interfaces of other existing notepad applications, such as Apple and Samsung

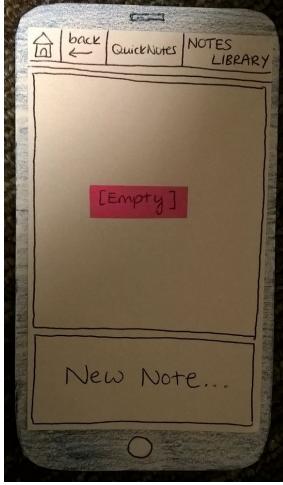
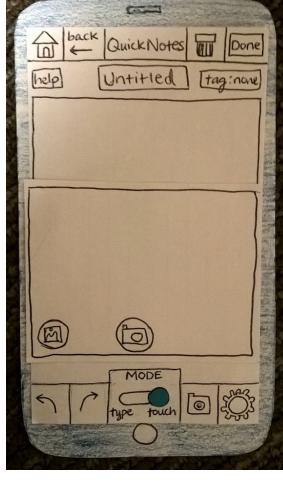
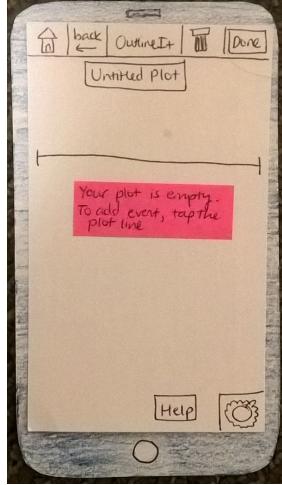
note features. We adapted features from these existing applications that were logical for our design, such as a Done button and a tagging system. We realized that we had been subconsciously clinging to the Windows notes interface, which was hindering our design. Incorporating successful aspects of other interfaces made our design more intuitive, and later participants navigated it more easily.

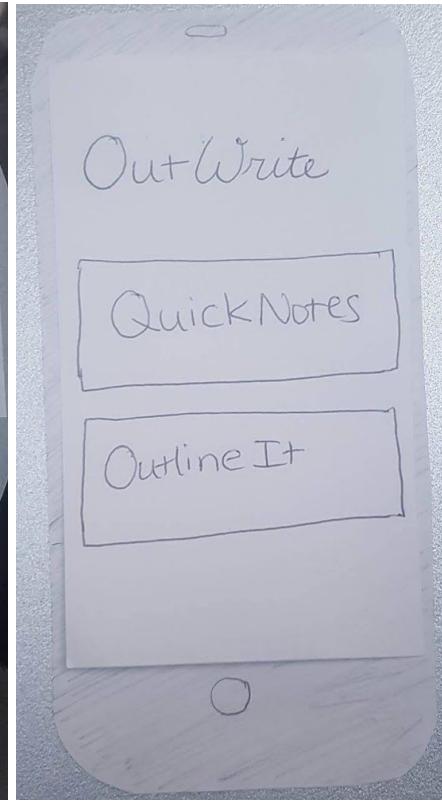
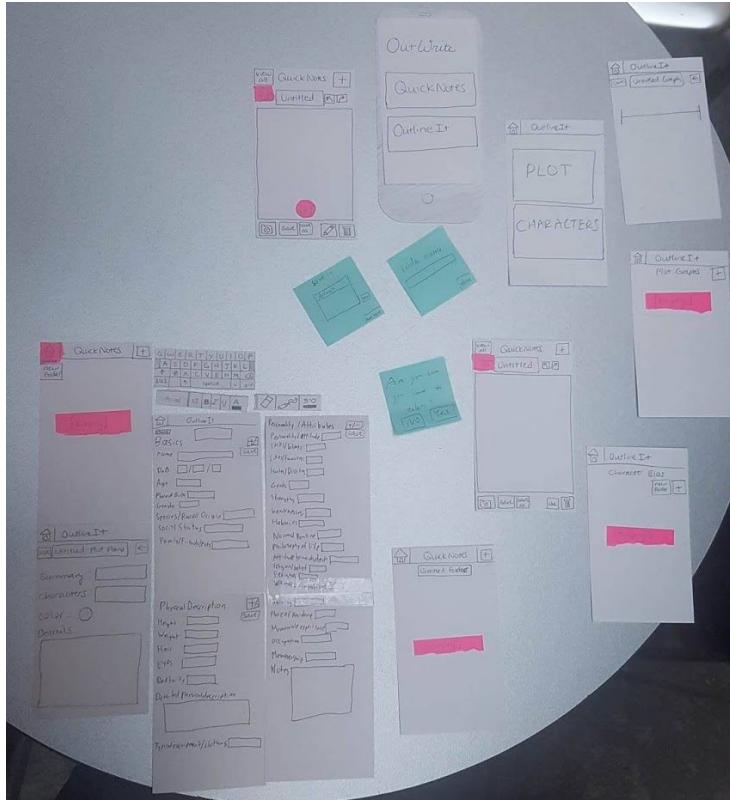
We were also surprised that the processes of developing some features led us back to the original interface. For example, we initially supported embedding photos with a camera icon, but participants had trouble locating it. To make its function clearer, we changed the button label to the text “embed photo.” Despite our idea that the text would clarify the function, a later participant said that the text implied that an existing photo would be imported from a gallery. They told us that they had been looking for a camera icon. After this feedback, we changed the icon back to a camera, but placed it in a more logical location for our digital mockup. Seeing the users be more confused about a button that says embed photo than a button that has a camera icon when told to embed a photo further drives home the point that we are not the customer. We see the world differently and we have our own set of biases and beliefs, especially since we spend so much time working on the app.

Our tasks have changed slightly as we redesigned our app with our feedback. Originally, we wanted to make sure that we catered to writers who wanted to write down their notes immediately, and sacrificed standardization between the two sides of the app to do so. As we started to receive feedback, we quickly realized that this part of the task was not as important as standardizing the organization between the two parts of the application. On this point, we changed our task slightly so that we were more focused on keeping track of notes while still keeping a very large new note button so that users are able to quickly add notes without having to deal with inconsistent organization.

If we had more time for more testing, we might have used more iterations, and we might have tried to go for a wider audience of people to test on, but as is, with the limited time, we made do with what we had. It would have been nice to test the prototype on our target audience of fiction writers. We wound up with only one major overhaul, that being after the heuristic evaluation. Judging by the feedback we received from the audience we did have, we may have not actually made very many more drastic changes. After we figured out a way that worked and that people generally seemed to like and use, we only needed smaller refinements to the process. If we had time to develope more iterations, we probably would have done so, since changing small details could still greatly improve the user experience. We would have liked to test our digital mockup with participants as well, to see whether they found navigation easier without the barrier of handwritten visuals. This would have allowed us to refine our graphic design and color scheme, as well as the size and placement of features on the screen.

Appendix

			
We conduct the usability test as if the user is using the app for the first time, all libraries are empty and every thing set to default.	We gave our users least instruction possible. One user identified a critical error of not able to erase.	Instructions are given in the tool, and we let our users to read instructions in his/her own and see if he/she will be able to figure out the procedure.	After the user has completed an action, we use sticky notes to reflect what happened.



Evaluation 1

Facilitators: Kevin Jung, Fengjun Dong

Evaluators: Kim Lum, Harrison Kwik, Yujia Liang

delete photo and text.

user ~~control~~ control

V: 2

F: 4

H-4, V:1, F:4

Folders/make folder s not
consistent across things
(plots don't have folders)

H4, V:1, F:4

back button should be upper left,
save at bottom
(Outline it)

~~APP~~ H-3 V:3, F:3

Since everything autosaves, ~~deleting~~
deleting is hard/confusing

H6, V:1, F:4

Can't see current folder

H-~~4~~, V:1, F:3

Save-as doesn't let you rename
file like most systems.
name-changing too

H-8, V:2, F:3

The character bio options are
overwhelming

feedback for "saved"

V: 1

F: 4

H-3, V:2 F:3

Cannot remove/change photo
after taken.

H3 & H7

Search Notes, filters

V: 2 F: 3

Hb, V:1, F:4

Previous note → view all → new note

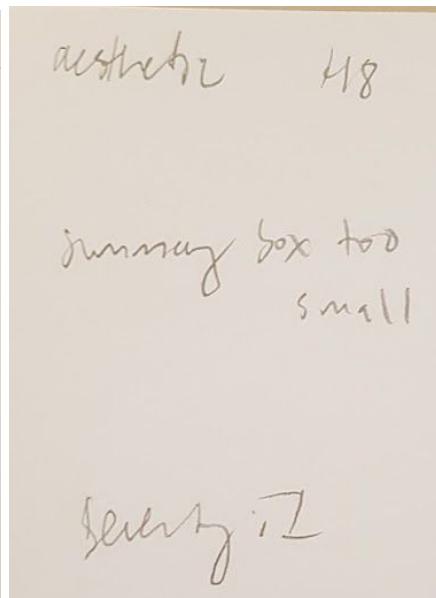
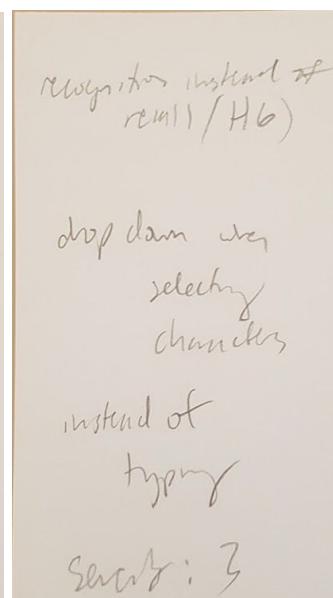
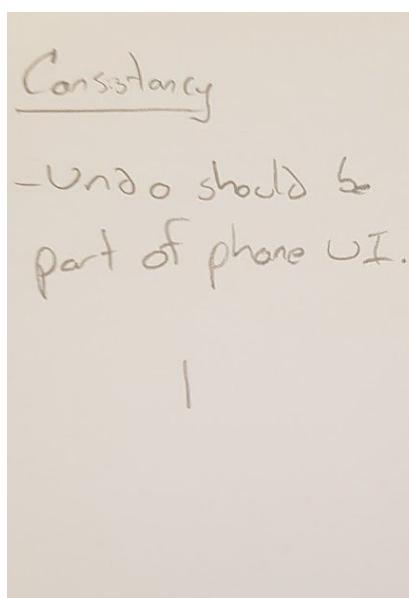
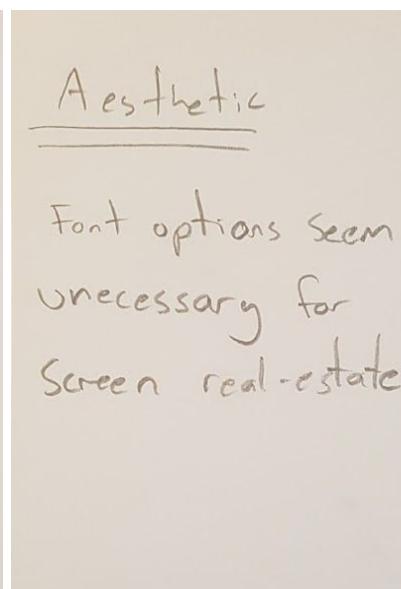
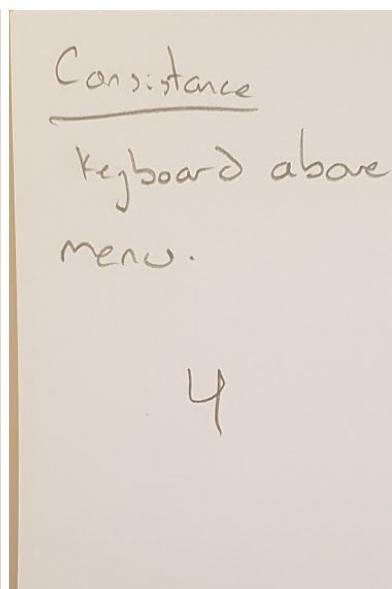
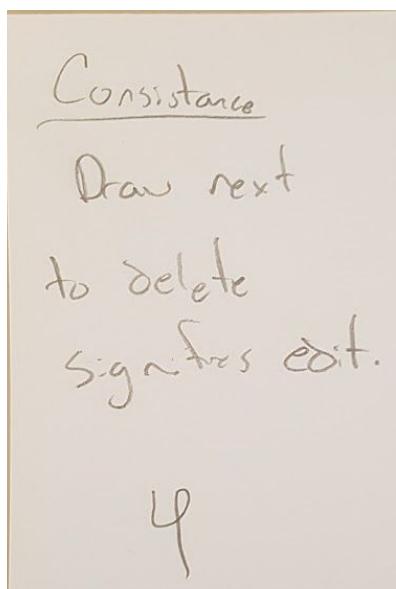
↑ this is lost, instead of allowing
us to go back to it

→ would like a back button
instead of home on
View All for QuickNotes
and in Outline It

Evaluation 2

Facilitators: Marisa Yamasaki, Alisha Meherally

Evaluators: Tracy Tran, Matthew Chun, Carson Wilk, Carl Ross



Design

Dropdown on mobile

1

Design

Bad use of real-estate.
Requires very fine control

3

consistency + standards

(H14)

write button or draw button

Severity: 2

visibility of system status

which parameter in?

character vs plot vs ...

Severity 3

use control + freedom
(H13)

no delete
folder option

Severity: 3

help + documentation

H10

how to navigate graph?

Severity: 3

user control +
freedom (+13)

how to go back
from camera
mode?

severity: 1

error prevention
(+15)

multiple w/
same title?

severity: 2

user control +
freedom
(+13)

I want to draw
on the
plot part screen

severity: 2

grey out undo-redo

details on plot pt

make "new folder" screen for OI

"making character" + add
button

dropdown

severity: 2

Match b/tw system &
real world

- usually navigation is one
place, need more state
change buttons

Consistency/Standards

- undo button should go back

3