Milestone 4

Digital and Interpersonal Communication

CPSC 4140 - section 001
4/22/18

Austin Youngblood

Noah Axelrod

Walter Thompson

Carson Sallis

Alexander Stone

https://github.com/StoneHub/4140

Study Protocols

Heuristic Evaluation:

The goal of a heuristic evaluation is to identify problematic user interface design with the help of UI experts. For our heuristic evaluation, we gathered a group of 5 UI experts who are part of our class and had them use our application in its current level of functionality. We decided to use 5 participants because of Jakob Nielsen's research where he found that 5 participants were able to find around 70-75% of the UI problems in an interface [3].

Each expert participant was given an excel spreadsheet and asked to record UI problems as they were found. To determine if a certain aspect of the interface was a problem, participants had to use Jakob Nielsen's ten usability heuristics for interface design. If a participant found an aspect of design to be in violation of Nielsen's heuristics, then they were asked to record the problem and identify which heuristic principle was violated. Then, users were asked to rate the severity of the violation on a scale of 1 to 4. A rating of 1 indicated that the problem was of low significance, while a rating of 4 was of the greatest level of significance. Finally, participants were told to offer potential solutions to the problems that they found.

Predictive Evaluation:

For our Predictive Evaluation, we chose to use an adaptation of the Keystroke-Level Model called the Touch Level Model. The touch level model measures the time it would take for an expert to do tasks on a mobile device. Since the KLM model is mostly used for computers, we had to use the adaptation that would work best for a mobile device. The reason we chose this model was because of the amount of interaction with the device you have when you are working with a mobile device. The GOMS model didn't quite fit since we were looking for something that could give us more quantitative results. We also didn't want to look at Fitts' Law because we know that our interface uses more swiping than just hitting targets since you have to interact with a map. The buttons that correspond to composing a message or finding your old conversations are also easy to find and use the principle of recognition. With all this in mind, we felt confident in our decision to use the TLM model. We took all our tasks that we were using for our usability testing and put them into the TLM model. We assigned operators for all of those tasks and used them to calculate approximate times that it would take an expert to complete the tasks given. One operator that we did not measure in our evaluation was the distraction operator. The distraction operator is a multiplicative operator that indicates an outside distraction that would cause the length of time to complete the task to increase. Since we didn’t plan on having distractions in our usability testing, we didn’t feel the need to include distraction operators on our predictive evaluation. Another interesting feature of the TLM model is the I operator which stands for Initial Action. This operator is used to designate an action like unlocking your phone or opening the app and is traditionally put at the front of your task breakdown.

Think Aloud Evaluation:

In Jakob Nielsen’s 1993 book, *Usability Engineering*, he states that “Thinking aloud may be the single most valuable usability engineering method…” [1] In addition to this supporting statement, Think aloud evaluations have remained as one of the most popular usability methods throughout the past 20 years. This is due to the test being inexpensive, extremely informative, and a thorough evaluation of the produce. Due to these impressive factors, we have decided to employ the Think-aloud method to evaluate our existing product.

The first step to set up the environment for the assessment was to decide on a group of tasks which summarize normal usage of our application but also remain broad enough to not bias any results. To do this, we decided on 12 central tasks which summarized all major activity involving the use of our application. These 12 tasks were able to encompass all normal usage of the application while remaining broad, but still specific enough so users would not be confused. The following tasks were listed in the following order: Post a note at a location, read unread messages, reply to friend, view old notes, find an unread note, post note at current location, find a read note, view who left a note, find current location, post note to multiple people, find settings, and use navigation menu to return. The next step was to determine volunteers who would be willing to perform the Think-aloud evaluation. This step came with some difficulty due to it being necessary for completely new volunteers who had not used the application before. To remedy this issue, we were able to bring together a group of 5 individuals comprised of group members’ roommates and friends who were willing to dedicate their time to perform the evaluation. This number was adequate by most recommendations, including in-class recommendations. For the next few steps, the group attempted to be as hands-off as possible. We assigned each user one task at a time and then instructed them to be honest regarding their experience and to speak their thought process aloud; including any assumptions that they make. Our group then recorded major data and only intervened if a user was completely derailed from a task. Finally, after all, data regarding the tasks were reported, the users were given a questionnaire to gather additional information regarding their experience.

Usability Testing:

The usability testing was done throughout the week while taking multiple factors into consideration. Our main goal was to get a diverse group of people to benchmark the application. To do this, we attempted to ask a diverse group of people that consisted of friends and acquaintances to test the application. They all had a few things in common. First off, all users were accustomed to using apps on their phone. Whether it was Android or iPhone, they all had at least minimal experience with applications. This also encompassed our target group of users. Also, we tried to maintain usability testing within this target group, so the results would be akin to real life usage. The testers, our group members, were very hands-off regarding the process to get the most accurate results. For the process, the group read from a script to tell all the users, in the same way, what tasks to perform. The tasks were completely identical to the tasks from the Think-aloud section. The most common issue was the very first benchmark test. No matter the trial, the first was always the longest as the user had to get accustomed to the application. As the testing went on, the process increased in speed due to the users becoming familiar with how the application functions. A noticeable difference in the testing was recorded when the user had some experience with google maps or not. This skewed the results and therefore made an impact on how quickly someone was able to pick up the application. The resulting factor is may relate to the fact that we built the application with google maps in mind.

Demographics of Users:

As stated earlier the demographics of the users varied far and wide. This, however, was limited by two things. It did not include the very young or the very old. The reason behind this is because those would not be the people that would normally use the application. That data is useful, and we would like to collect it later down the road, but it is not the most important part of the information that needs to be considered right now. The biggest user base for our application would be current social media users. We felt like that group of people would be the best to test in the first round of usability tests for the system to get the best feel for how the application would be used in the real world. Another restriction to the data set was geographical restraints, this means that we only had people that were available to us near our location. That means that the data is skewed towards a mainly college level demographics. If there was more time available to the group, then a more throughout study could be conducted with a wider range of available people.

Justification of Tasks:

The tasks we asked our users to perform in the Usability evaluation were all functions and processes that a user must go through to use and test every feature of our application. We presented each task to the user in an order that allowed a natural discovery of interactive elements and actions. For each evaluation, the Usability, and the Think Aloud, we issued the same task to the users. These task can largely be grouped into two categories. Task that focuses on the map interactions and task that focuses on submenus and other UI sections. Since the application opens to the map view, the first tasks we had users focus on would be the task related to the map.

First, we issued a task to post a note at any location the user would like. This lets a user explore the map by panning with their finger in the search for some location. Generally, this lets the total time for the task to be large and vary wildly between users, as they might be looking for a specific place they want to leave a note at. Next, once a note was placed by the user, we ask the user to find a read or unread note close to their general location. This lets the user get used to the map interaction, and also lets them start understanding the color coding for quickly identifying a read vs unread note on the map. An example of this task would be one where the user is instructed to find out who left an unread note for them. Since there were a mixture of read and unread notes on the map, they had to either already know from experience, trial and error, or interpret from context clues that the brighter color (Green) was one they have not opened, compared to the brunt orange for read notes

The next category for our user task focused on the navigation of the applications different screens and menus. The navigation drawer allows access to each of the different screens of the applications. The first task we asked users to perform in this category is to open and close this navigation drawer. This action is not so much specific to our application as it is a general action and design implementation across the Android platform. Still, we needed to get usability testing on this because it is crucial to the navigation of the software. Next, the user is tasked with finding old or archived notes. This requires the user of the navigation drawer to switch to the message list view. Once in the message list view screen, the users are given a task to reply directly to an old or achieved note. This task is important to ask once the user is in the message list view as from there they have a direct button with textual context to perform this reply to an archived note. For one of the last task for our study, we ask the user to post another note on the map This time they need to enter multiple comma separated recipients for the note. This test the functionality of the contacts autofill service and the visibility of this feature.

Analysis and Discussion of Results:

The results were actually better that we initially thought for the system. We expected that we would have a lot of confusion on what the app is trying to accomplish. However, participants in our evaluations typically did not have much confusion about the overall purpose of the app and it. It was a great idea to base it off of google maps so that way many of the users felt familiar with the system already. Though some of the users had some comments on what should change and what should stay the same. They did comment that there should be a public feature in the application in the future. The note icons should be more customizable. The message list should be cleaned out a bit and small features like that. These are items that will be flushed out in a later development process. The great part is that there were no complaints about the system in a way that we know will not be cleaned up in a later process.

Our predictive evaluation showed that our predictions for how long it would take experts to navigate our tasks were generally ahead of how long it took our users in the usability tests. Most of the bias in our timing data came from the fact that users were not yet acclimated to the application so it took them longer than what we predicted. If these users were to keep using the app on a more regular basis then we are sure that they would eventually lower the time that it takes them to complete these tasks. The biggest discrepancy comes from the first task because in our usability testing we did not give them any prior time to work with the app and get acquainted to it. We just immediately gave them the tasks and had them fulfill them to the best of their ability. We could also use these results from comparing our predictive times and the actual times to help guide us to the areas that will need some more work. There should be a gap between what an expert user should be able to do versus a beginner but we can at least lower that gap a little bit to make the user seem more like an expert in a shorter amount of time.

Our graph on the average difficulty of usability tasks also will help show us what tasks caused us the most problems with our testers. The two highest rated tasks were the posting note at current location task, and the finding a read note task. By contrast, finding current location and viewing who left a note were rated the easiest out of our tasks. For posting a note at the current location, the reason we believe this one was rated highly is because this the first task that we gave all of our users. They were still brand new to using the application and getting to grips with how it worked. Perhaps if we had another round of usability testing and varied which order we gave the tasks in we would potentially see a reduction in the rated difficulty for this task. Our highest rated task was the finding a read note task. This task immediately points us to the color coding that we used to designate a read versus an unread message. A lot of our participants commented on how there was no context to differentiate the two messages and that if we could improve that then we could make that task much easier. The results were better than we initially thought for the system. We expected that we would have a lot of confusion on what the app is trying to accomplish. However, participants in our evaluations typically did not have much confusion about the overall purpose of the app and it. It was a great idea to base it off google maps so that way many of the users felt familiar with the system already. Some of the users had some comments on what should be changed. They did comment that there should be a public feature in the application in the future. Also, they note icons should be more unique and obvious as to their intent, instead of just color variation. The message list should be flushed out a bit to encompass standard messaging features. These are items that will be flushed out in a later development process. The good news from our data is that there were no complaints about the system in a way that we know would not be cleaned up in a later development process.

Our predictive evaluation showed that our predictions for how long it would take experts to navigate our tasks were generally ahead of how long it took our users in the usability tests. Most of the bias in our timing data came from the fact that users were not yet acclimated to the application, so it took them longer than what we predicted. If these users were to keep using the app on a more regular basis then we are sure that they would eventually lower the time that it takes them to complete these tasks. The biggest discrepancy comes from the first task because in our usability testing we did not give them any prior time to work with the app and get acquainted with it. We just immediately gave them the tasks and had them fulfill them to the best of their ability. We could also use these results from comparing our predictive times and the actual times to help guide us to the areas that will need some more work. There should be a gap between what an expert user should be able to do versus a beginner, but we can at least lower that gap a little bit to make the user seem more like an expert in a shorter amount of time.

Our graph on the average difficulty of usability tasks also will help show us what tasks caused us the most problems with our testers. The two highest rated tasks were the posting note at current location task and the finding a read note task. By contrast, finding current location and viewing who left a note were rated the easiest out of our tasks. For posting a note at the current location, the reason we believe this one was rated highly is because this the first task that we gave all our users. They were still brand new to using the application and getting to grips with how it worked. Perhaps if we had another round of usability testing and varied which order we gave the tasks in we would potentially see a reduction in the rated difficulty for this task. Our highest rated task was the finding a read note task. This task immediately points us to the color coding that we used to designate a read versus an unread message. A lot of our participants commented on how there was no context to differentiate the two messages and that if we could improve that then we could make that task much easier.

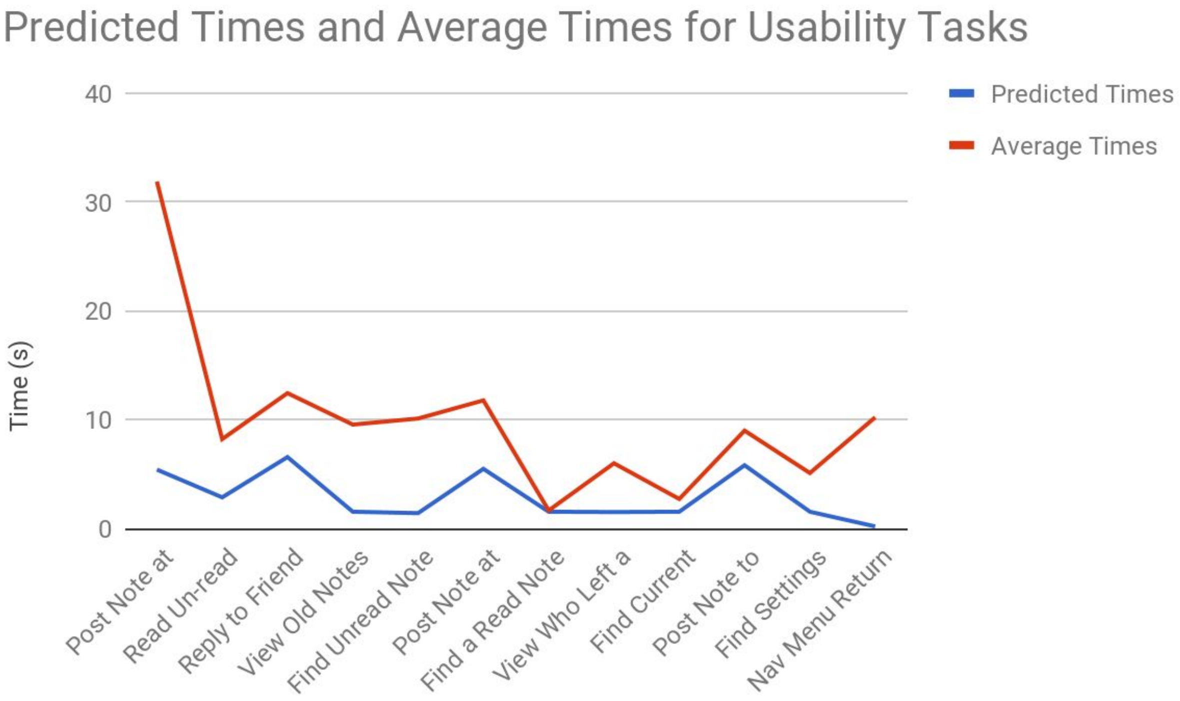

Figure 1. A graph showing predicted average times versus actual average times to complete evaluation tasks.

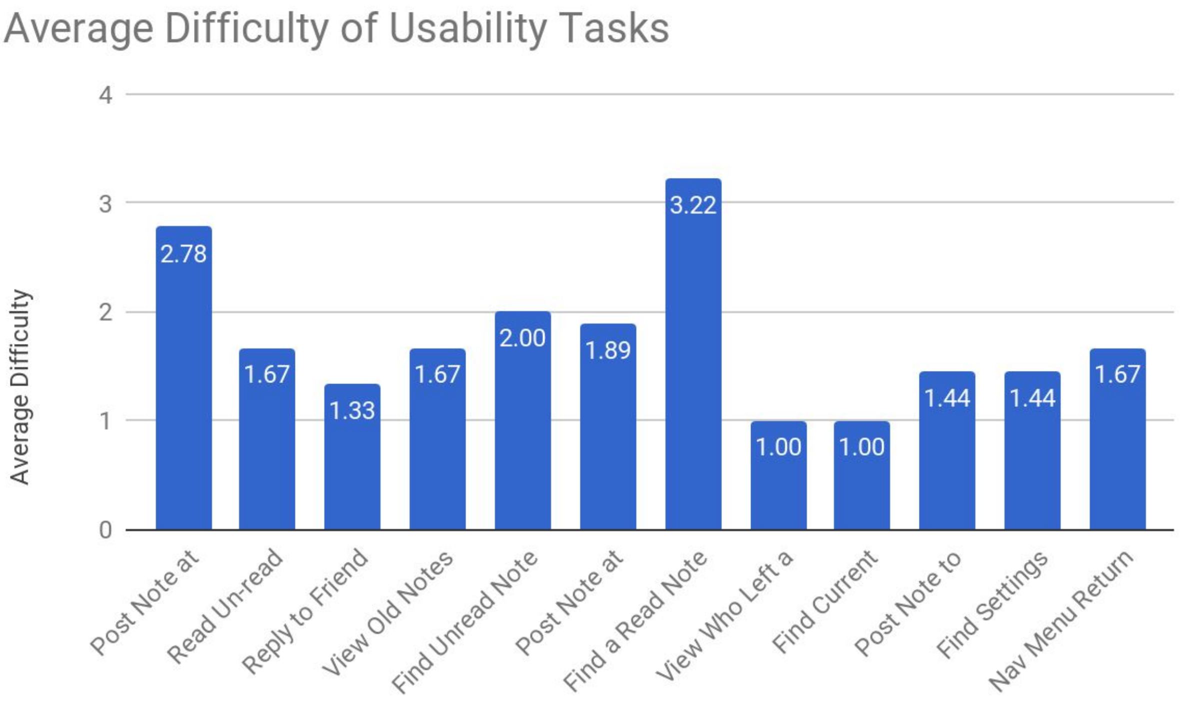

Figure 2. A bar chart showing difficulty ratings that users assigned the tasks in our usability evaluation. A

rating of 1 indicates no difficulty and a rating of 5 indicates an extremely difficult task.

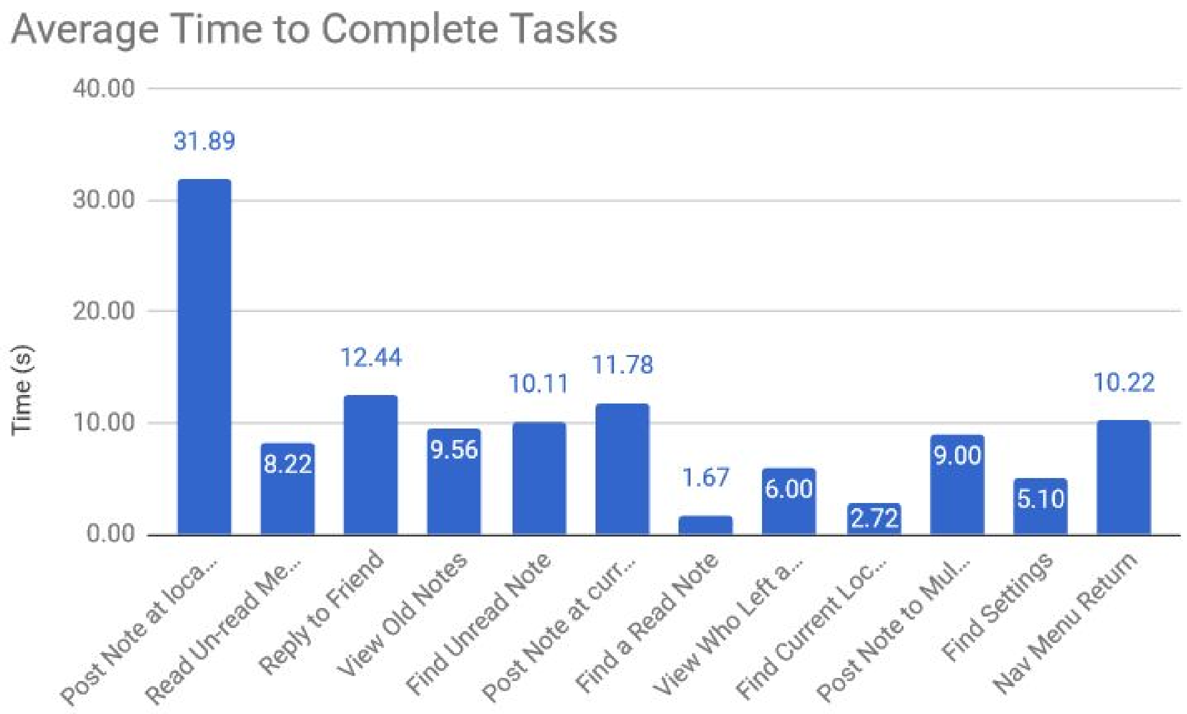

Figure 3. A bar chart showing the average time (in seconds) to complete our evaluation tasks.

Road Path to the Future:

The heuristic and usability testing that we performed at the end of this development cycle left us with lots of feedback and data on our application and its features. Think aloud data from the usability testing with users gave critical insight into what users expect in control and freedom. If we as a development team had another agile cycle to work on this application, we would focus on making the user experience more fluid and intuitive. Towards this goal we would focus on Android system issues, many uses express difficulty with the interaction of keyboard and other device specific issues. One of the most requested features to implement would be better in-context tutorials or help functions to tell users how to interact with elements and the process involved with posting or viewing notes. In part of this, color coding was not enough for new users to easily understand what each note on the map status was.

After all the data was gathered, we concluded that our application would be a new concept that can add to the benefit of the current trends of the social messaging experience. The goal that we set out to accomplish was to create a more personalized messaging experience where people will be more involved in messaging as a whole.

Our core values encompassed giving users a greater depth in there messaging experience and, to do this, we developed specific features with the sole purpose of adding to this experience. After our testing cycle, we came away with insightful critiques and ideas that we could apply towards our application. Given another cycle of development and testing, we could implement these supplementary ideas and improve on the existing application.

References

1. https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/

2. https://www.nngroup.com/books/usability-engineering/

3. https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/