Reflective Journal:

**April 9th 2018:**

Most of the group has made contact on the discussion board by now, I assisted Brent in finding out how to get the email addresses of our teammates by typing their names into the ‘to’ box when crafting an email so that he could send out instant notifications instead of relying on people to see it in canvas updates.

I’m not too incredibly confident in my project proposal, I think the base is a nice, and not too complicated item, but I can’t really think of a unique feature for it, or some way to expand it past something that already exists. Should I make it look like Tinder for recipes? Reddit for recipes? How should I approach the display, what do the users need, are we going to even use my idea? I guess we’ll find out as we go.

I did some pre-emptive reading from the book, as well as the slides, and took some notes on a few key concepts as I went along some examples were:

* Usability has many goals associated with it, they include:
  + Effectiveness
  + Efficiency
  + Safety
  + Utility
  + Learnability
  + Memorability

Memorability and Learnability were two distinct ideas that I would conflate in CS361, but my current understanding of them is:

Memorability - Plays off what prior experience a user may have, and uses it within the application. A user will not want to re-learn an entire application if they are frequently using it, so making an interface that a user can memorize for future usage is important.

Learnability - A learnable interface is one that a user may not use very often, but they can learn or re-learn the interface with little to no issue. This could be something that you infrequently use, like a website to get your W2 tax information, or a tourist booth at an amusement park.

* Design Principles also appear to be a major point of this chapter, I’ve written what they are, and an abridged definition to try to memorize some of their aspects:
  + Visibility - Can I see the state of a device and possible actions?
* Are the controls positioned in a manner where they can be easily found and used?
  + Feedback - what action is being performed now? This action needs to be immediate and synchronized with the user action.
  + Affordance - Relationship between the properties of an object and the capabilities of a person that determines how the object can be used.
  + Constraints - restricting the kind of interactions that can take place within the environment.
  + Consistency - similar interface for similar interaction and similar task, for similarity increased the learnability of the environment.

The design principles seem to be much more important with the actual design of the interface, and less about a conceptual idea. We’re going to want to reflect on these ideas much more when it comes to actually designing an interface, as we will need the user be able to learn how to use it as they’re using it, along with what they can and cannot do.

**April 13th 2018:**

I took Quiz 1, and was surprised I retained enough from the to get 100% on my first attempt! I’ve written down the questions and answers to start making a list of notecards for future quizzes, or even the final.

On a side note, we ended up choosing my project for our group proposal. I’m still not too confident in it, but most everyone else seemed to think it was a good idea. We’ve also started a Slack channel so that we have a more convenient form of communication than the discussion boards, so far it's pretty good to be able to contact everyone, but hopefully everyone checks it.

We’re also looking into assigning who gets what role, I’m not too keen on the communication or leadership roles, but I believe I am able to work with any of the others. It's too early to tell how well we’ll do as a group, but I’m still confident we will be able to do well.

**April 15th 2018:**

Working with the group to finish up our Project documents 1,2 and 3. I think this was actually a very good assignment for drilling in some of the core concepts that create a foundation to the rest of the class concepts.

From this week’s information, I’ve gathered a few new perspectives on a user’s role in designing an interface, They provide key information on how usable your product actually is, a new perspective on your product is always important. To gather information on what a user wants, or needs from a product, you can directly ask groups or survey for new features to see if some feature is more desired than another.

The work on the interface is not done at that point though, did you really implement what the user wants? Is it usable by the user? Do you need to make adjustments? These and more questions need to be asked before finalizing the product. A product may continue to evolve over time, and as technology enters new realms, software and applications will need to adjust themselves to be accessible to the users.

Personally, I had written up a section for our users that we will access to. Initially I had glossed over this, and began writing. Before I finished my section, Alec pointed out that it is about users that are within our target users list that we have access to.

Some Important ideas for this section seemed to be centered around the PRICPE concepts, to try to solidify my understanding of each of the steps, and try to display my knowledge of it, I’ll recite what I understand them to be here: PRICPE stands for:

**Predispositions** - What do you know before you start? What do you believe you know? What don't you know? What do you believe you don't know? Also prepare a list of facts and questions around user experience.

**Research** - Study your potential users, study potential resources like books, and surveys. As well as study competitor products to gain more information about what you are building, and things you can do to make it your own.

**Insights** - Do we know enough? Do we need to do more research? Has our research had an effect on our predispositions?

* While we may have results of how our predispositions were affected, we don’t iterate back to that step, we can iterate back to any other steps though.

**Concepts** - This is a somewhat rough implementation of what your project may look like, It is normally a sketch or drawing, but not an actual model of the product.

**Prototypes** - A working model/example of the product, somewhat of a MVP (Minimum Viable Product) where different concepts and approaches can be explored, as well as the product being able to undergo changes at little to no cost.

* This can be either Low, or high fidelity meaning:
  + Low Fidelity: Very few details, more focused on core product
  + High Fidelity: bells and whistles included, detailed build of what product may look like.

**Evaluation** - Did your product do what users wanted? How well did your team handle the process? Are there issues you need to approach for future iterations?

* You can evaluate at any step, and it may be beneficial to do so.

Some key takeaways from this that aren’t immediately in the definitions is that PRICIPE is an iterative process, where you can go through your steps and look over them again and gain new perspective to apply to the future of your product. Another Very important idea seems to be the most important part of predispositions discussion: ‘what don’t you know?’ It helps highlight blind spots that the project has to give some ideas of what to research going forward.

Applying these to our project through the project documents 1,2 and 3 seemed a little challenging, but this challenge may stem from my lack of communication skills. We know the basics of what a user will need to go through with the website, they will need to:

* Be able to view a recipe
* Be able to Search for a recipe
* Be able to add a new recipe

Those are the absolute mandatory items that the user will need to be able to do, I’ll continue conferring with my group, but I’m not sure what else can be placed in the absolutely know section.

**April 20th 2018:**

Our TA expressed concern for our writeup, I touched base with group on slack asking if we can think of a unique feature to add to make it more complicated, or should we scrap and try something else. Waiting on a reply from the group, but I sent this friday morning, so it's probably not their highest priority at the moment. Brent proposed 3 ideas on how to improve it, I plan on contacting the TA’s for some advice.

I’m also mildly concerned about my understanding of the contents of the book, not that the concepts are difficult, but because I’m having difficulty retaining interest in the passage I’m reading, and as a result I don’t believe I’m learning as well as I could be. I’m going to try to give myself smaller chunks to read at a time, instead of trying to read a chapter at a time, and then quizzing myself over the contents of that section. Maybe the smaller bites of content will be more approachable for me.

I took some inspiration from the key questions for the week, and decided to respond somewhat indirectly to them. The human mind enjoys familiarity, sameness and likeness is something most people are comfortable with. Which is why memorability and learnability are very important aspects, if your interface is learnable, and memorable, users will feel more comfortable with your product. To gather data on how to improve an interface, an ethical approach would be to try to survey the user on if they are satisfied with the application, or the specific user that they’re using. Whether this is within the application, or an email afterward, you may receive important information on what users believe about your product.

My workload for other classes is also increasing, as midterms are fast approaching. Hopefully I won't be in a position where I have to neglect our project to concentrate on another project/assignment.

**April 29th 2018**

Me and Brent managed to knock out the majority of the document for our interviews, and conclusions we have come to based on them. We’re nearly finished, but *somebody* is waiting until the last second to add their interview to the pile. I’m not a big fan of last minute work, especially when my ability to contribute hinges on their work, but whatever, they said they’re having a family emergency, and i’m not here to start drama.

Our reflections from the interview seemed interesting, I expected our interviewees to be interested in a more picky custom search feature, but I was slightly taken aback when finding out that users were interested in the idea of our event feature, but did not regularly plan out, or attend cooking events. Could this be something that users think they want, but don't really want? Could it be something that users will love and it will revolutionize the world’s approach to cooking? Or could it just be that this may be a neat little feature that occasionally will be used? Who knows, at least, there would need to be more extensive research and testing than we have access to.

Something else that stuck out from interviews was that an interviewee was interested in submitting their own recipes, but didn't want them to be public. It was a somewhat interesting idea that never crossed our minds when planning this, privacy settings and filters! Allowing users to create content, and then decide if they want others to see it, or do they just want it in their own personal digital cookbook? This might be something we should bring up in group conversations.

**May 1st 2018**

Things are a tad hectic for our group today, we just started on our early prototypes, and writing up the documents which are due tonight. We decided it would be best if everyone bites off a chunk and approaches it from that way, for my chunk, I got:

* Friends list
* Event Page

They didn't seem to be too difficult to sketch up, they just required some thinking of positioning the elements. What I found more difficult was attempting to do some of the writeups for the individual pages for our application, as we need to be concise, which I’m normally great at, but I was encountering some difficulty keeping it simple when describing my events page, but thankfully this is just a draft with participation credit.

As far as reading, I’ve kind of been slacking on that front, i’ve taken peeks at the important questions to touch on the journals, but not much more than that. Hopefully I can get back on the horse before this journal is due! Looking at some of the posed key questions for this section, I believe that I am able to answer them from the studying that I have done though.

For instance, Mental models are an important aspect of designing a user experience, because it will enhance the usability of your product. If you’re designing a light switch, users (at least in the US) will have a mental model in their head of down being off, and up being on. If you were to design it the opposite way, it would lead to users getting confused, because their prior knowledge of similar light switches is different than this one instance. Confusing the user often leads to frustration, and you do not want users to be frustrated with your product.

User perception vs designer perception seems to be a fairly important issue, and a large issue surrounding this topic is that the designer perception is partially a meta analysis on the users, and how the designers think the users think. Designers may perceive that a feature is needed, when a user would think that it is not. Just as well, designers may think that a user is capable of navigating an interface that the end user would have no clue how to approach. Users have the same issue, but in the opposite direction, this is due to perception being shaped by perspective. In other words, an individual will apply what they know on to what they see, to take the model posed in the previous paragraph, a user will instinctively flick the light switch down to turn it off, as this is what they know, they are applying their experience to the given model.

These concepts are important for designing a user interface, because it makes your model much more easily learnable, as the user doesn’t need to experiment as much to figure out how to use your application, as you are designing it with information that they may have previously had from another concept/idea and applying your model to the formula, reducing as many variables for confusion as possible.

**May 3rd 2018**

Going into this journal entry, I’m a tad nervous for our group. I’m seeing a lot of other group’s progress, and their ideas, and I’m feeling like I’m finally understanding what the TA’s had been getting at for some of our group’s issues.

Beyond that though, I wanted to give some groups some ideas on their interfaces, and what things could possibly be missing, I tried to search for a couple that I would know something about, and I settled on giving feedback to the following two:

Group 12’s Game Streaming service - I had noted that they seem to have some focus on speedrunning aspects of games, I tried to apply some of my knowledge about the topic to ask a question that they may want to make an additional feature/page to sort out. That feature being the ability to choose what type of speedrun the user will want to be attempting with their run.

Group 7’s Recipe App - I thought their approach was a neat and different way of looking at a problem of a recipe database than our group is taking. Nonetheless I tried to offer some insight that our group had gained from our interviews, where some users were interested in viewing comments on the recipe to make it a discussion instead of just being told the recipe.

It was pretty fun getting to see everyone's progress so far though, as well as seeing other’s ideas on how to tackle specific design choices like friends lists, and navigation menus. At the same time though, it made me wonder how our groups hamburger menu would shape up in user testing, it may be mildly thematic to have a hamburger menu in a recipe application, but do non-tech savvy users know how to navigate them? Time will tell I guess.

Some of the key questions that had been posed for week 5 seemed a bit more difficult to research, but I can apply what I know to this item, starting with answering: Why do you need a formal process for evaluating Prototypes? From what I understand, this is to create a structured environment, where the user the user has to navigate through the interface by actually using the prototype rather than just saying what to click. This is to make sure that the user can find their way through the menus, and end up where they wanted to be. This can also allow you to get immediate feedback on what the user thinks, which will be important to know what their experience with using your interface is.

There are a few advantages of developing prototypes in stages, you can see individual features that users are having difficulty navigating, and design a new ‘stage’ to see what users would prefer. It's also an agile part of development, where it does not cost too much to create a new prototype, and present it to the user, this will make it easier for a company to research their users and figure out what is wanted from the interface.

Developing in stages does seem to have a few downsides, one of them specifically being that as the product is prototyped more and more, the scope of the product may continue to increase based on user feedback, this will also make the project more and more complex until it is no longer a feasible application.

**May 4th 2018**

I’m hopping on to do my portion of this week’s group project today with Brent to make wire frames for our submission to Project 5, I mostly jumped because I’m going to have a really busy day on Sunday, and didn't want to have my team suffer because of it. I notified the team of this restriction, and then started my part. Brent and I decided to divvy up the work as following:

* Brent and I do wireframes.
* Sue does any additional that are needed.
* Alec and Alex do the rest of the writeup.

Brent was generous enough to make a template and share it with me to make creating the wireframes much easier, altogether we created 9 pages that we think will get the job done. We are able to navigate between them and we think it makes sense how we move from page to page, but the step after this seems to be actually applying and testing these interfaces with potential users who will be seeing it for the the first time and have to make inferences on how to conduct themselves. It looks like its gonna be fun!

**May 12th 2018**

I know it's been a week since the last update, but not much has happened in the meantime. I took the quiz for the week yesterday, and somehow I lucked out with a 10/10, I managed to fill out the navigation chart activity before it got DDOS’d, and I got a lot of work done towards the next project!

Something a bit unexpected was the grade we received for project 5, I only worked on the wireframe models, which got 13/14, but seeing the writing grade stung a bit. I was only able to work on the wireframes do to extenuating circumstances occupying my time over the weekend, but seeing that the written got 3/10 blew my mind, Who’s to blame? That's not anywhere near as important as the question: What's to blame? What happened, and what can we do to fix it are two questions we need to ask. I’m hoping that we can gather everyone and talk about it, but it seems to be getting more and more difficult as we go, where instead of everyone attending meetings, we only have 3 in attendance now.

To shift gear after that sour note, I am pretty thankful there was no new reading that was due this week, so I can catch back up with the pack, but we still had some key questions to pound through, so I’ll try to explain my way through them.

As for why we need to plan for an evaluation, we’re the ones that determined all of the other ideas and scopes for our product, we now need to evaluate on our project with information that will be given with our empirical testing, is the user able to navigate our interface? How cumbersome is it? Do we need any major changes. We need to evaluate our product and determine if it is what we want, and then determine if what we have built satisfies what we want.

For why the evaluation plan needs to be formal, my own research came to the conclusion that it needs to be presentable. We need to be able to clearly impart and communicate all parts of our evaluation to other individuals. It should be easy to understand, and not be lengthy or overly technical. Effectively making it readable and presentable for review. A simple google document isn’t going to cut it, we need a clear and organized approach.

**May 19th 2018**

Brent had me take the lead for the project 6 write-up, and it looks like I did not disappoint, We received 20/20 for the project which our team needed the morale from, but it was still nice that Brent was very attentive in emailing the TAs with concerns and queries.

For this next project, it looks like theres gonna be much less writing for it overall. We will have to do our analytical and empirical write ups, but after that we’re just appending our information. While doing this journal entry, I had finished up my portion, and realized that we’re missing a number of pages/features that we’ve outlined that we would like to implement. We also have some features that are of questionable use, I’ll consult with the group about some of these issues, but they mostly sum up to:

* No ways to view events you’ve been invited to
* Is ‘meh’ option really needed in can’t/wont eat?
* No real way to search friends
* Several page inconsistencies

In order to resolve these issues, the group will consult with each other before making the next level prototype. Fingers crossed everyone can make it, or at least have some input before the last second.

Some key concepts from this section were:

Why is it important to understand user problem solving behaviors on computers?

It is important to understand their behaviours because they will likely apply those skills onto your application. If you design your application in such a way that they can troubleshoot using these established skills, the user will often feel more confident in navigating your interface.

Why must you make user-related decisions at design time? Why not just sort it out in alpha and beta testing?

It is much easier to make these big application design decisions at a stage where it's much less expensive on time and resources to uproot a design, and swap it for another. If it was to be sorted out later in the cycle, it would become much more expensive to create a new alpha interface, because it would involve much more development resources all working to make it a bigger bill in the end.

What are the advantages of enticing users to behave in a certain way? Are there any downsides or difficulties in doing that?

Enticing a user to behave in a certain way has many advantages in developing an application. It can assist in retention time, making the users more likely to use the application for longer amounts of time. Getting users to use an application in a specific way can also help streamline their experience, making them able to flow between more actions in a less clunky way.

It can also introduce users to a new method of usability, some users will need to be coaxed in one way or another into using special features, AI is an example of this, where Amazon, Apple, Google, and Windows all have an AI that they push you towards to help a user navigate through the interface (it is also really useful for collecting user information, but that seems a bit scummy).

**May 26th 2018**

I realized that i’m a bit dumb today, I’ve been heading each of these journal updates with ‘march’ instead of ‘may’ I’ve fixed it since, but still found that funny.

So for this week, our group got to have some fun with myBalsamiq, brent did a huge portion of setting up the application within the program, so huge props to him. I ended up working more on the written segment (even though at the beginning of the term I was assigned graphic design, but whatever I guess) Which ended up pretty swell by the looks of it, I do feel like I need to add more justification for each feature that we had added or removed from the project, but we will see what the TA thinks.

As far as the key questions the instructor poses for this week, I understand these to be the justifications behind the segments:

How can well thought out navigation make your application a winner?

Well thought out navigation can make your application a winner by making the user interface to help guide the user without having their hand held. A user will enjoy an interface that they found memorable, and easy to learn. Having to look up how to access a feature is never a good thing, but for some items like an operating system, it is a necessary evil.

How can you build that kind of navigation experience?

We can apply skills and tools that this class has introduced us to. Using the PRICPE methods, and maintaining contact with users can help determine what approaches are winners and losers, and can help us get a better idea of what implementations will provide the best experiences.

Can the tools and skills presented in this course help you do that?

Yes, during my explanation of the previous segment, i touched on PRICPE, but there are other possibilities: usability goals, design principles, Prototype construction methods, as well as Analytical and Empirical evaluation are all useful tools for going forward and designing interfaces.

**June 2nd 2018**

It feels like we’re getting into the crunch cycle of our project, which is unfortunate because CS444 and CS331 are both keeping me really busy. I’m trying to dedicate a good amount of time to this project though, it looks like we did get feedback about our justifications being lacking, so we will need to edit the existing ones, and more thoroughly flesh out the new ones. The TA did like our prototype a lot though, but they still had some suggestions to make the interface more easily navigable. Things like implementing a back button amongst other ideas. It seems like Xu is taking responsibility for making the back button, hopefully all things with that go well.

As far as the key sections for this unit, we only seem to have one. here it is, along with my response:

Would that make computers better for everyone?

What wouldn’t make them better? A more serious reply though, something that would make them better would be a more direct way to design with usability in mind. Not in the sense of taking as many controls away from a user as possible, but instead in the sense that we could make more frequently used features and functions more immediately accessible to the user, then make the less commonly used features further away. This could help reduce screen clutter, help guide a user to do their daily activities more easily, and prevent confusing the user. The user should still be able to forage for information somewhat easily, but at the same time, uncommonly used items should not be at the forefront.

A big direction that has potential, but is still in its infant stages is AI, Cortana, Siri, and Alexa all help the user experience immensely, but still need a lot of work. These AI can help guide a user, can give a quick answer to specific questions, and can make things like amazon purchases in an instant with no clicks involved, just a few spoken words.

**June 7th 2018**

It’s getting kind of exciting for the term to be over, not because I dislike any of my classes, but because it’s been an incredibly stressful few terms for me this year. CS325, CS444, CS381, ST314, CS331, etc. This has been a rough school year for me, but thankfully all I have left is capstone, and some somewhat easy applied classes.

But let’s get down to business, the project. It’s looking good, it feels pretty good to navigate, you can go back one page which a brand new feature. I’ve been pitching ideas to Brent about features to add that could help QoL for the application, but would not need to much work to implement. I’ve also been helping him hash out what exactly people mean in some of their conflicting feedback.

To touch on some of the key questions:

Did we do our best work on the project?

I believe that there is always room for improvement, no matter what it is, it will never be perfect, but can strive to be as close to perfect as it can be.

Our team had some rough spots starting, namely getting an email from the teacher saying we’d fail if we didn't contribute. Then came problems with the project scale not being large enough, then some work schedules had some conflicts, some people worked more on specific submissions than others, drama happening, and on and on.

Despite all of these hurdles, I believe we were able to create a Solid application interface that is navigable to our intended audience. I believe each of our group members had a solid contribution to the project, and I believe that thanks to feedback from the TAs, we were able to create a better interface that what we initially had planned.

My only gripe with the group at the moment is that I am asking yes/no questions that people seem to be ignoring, its kind of annoying, but I understand that they may be busy with other classes.

How can I help my classmates with constructive comments on the projects?

In order to help out our classmates, we need to give some direct criticism, or some ‘tough love’. A compliment sandwich works wonders for preserving how people feel about their performance, but there still needs to be direct feedback of what you believe they can improve on. I know I’m not the best at it, but I do notice that the direct feedback from the TA of what they liked and didnt like has helped us improve our project immensely.

Do I have my proctor scheduled for the final exam?

I signed up for an open session for the final exam many weeks earlier, so I’ll arrive for it as early as the lab opens. So I believe that I’m good to go in that regard.

Reflection on term:

Something I had believed in pretty thoroughly up till now was the three click rule, users should be able to navigate wherever they need to go on your website within three clicks. While I still believe in this rule somewhat, I also understand that this can lead to too much information being presented to the user, and through making prototypes that this goal is very difficult to do while also helping guide the user on where they need to go. While speed of access is a thing in retention, so is attention investment, if they need to spent a lot of time to memorize, or even re-learn your application, they may not be using it for much longer. I believe this class has given me a number of tools to reflect on graphical interfaces, as well as other forms of interfaces in order to help create a more user friendly, and user centric design.

How did the rubric help?

The rubric gave some general guidance on how to do the journal updates, we need to display some ability to intake, interpret, reflect, and apply the lessons that we are taught. This gave me a bit more confidence on what I needed to do for these journals as well as some general ideas on what I should be getting from the class.

Why did you not use the rubric?

I didn’t find myself referencing the rubric very often after the first journal, as it seemed like they were more like check-ins after reading segments about vocabulary and the like, as well as updates about how your group interacts.

What would you change to make it better?

I think an addendum could be to have the individual talk specifically about progress on applying lessons from the reading and lectures into the group project. I mentioned it a few times in my updates, but I think it could be a more core component of the journal in order to help reinforce the skills and how they are to be used.