

# Project 7

## Evaluation

Christopher Carrassi, Tony Huynh, Jared Storts, and Amanda Tait

Department of Electrical Engineering and Computer Science, Oregon State University

Corvallis, Oregon, United States

carrassc@oregonstate.edu, huynhton@oregonstate.edu, stortsj@oregonstate.edu, taita@oregonstate.edu

**Abstract -- This paper presents the results of an analytical and empirical evaluation for a prototype replacement application process for the Nation School Lunch program. It discusses the core findings of these two evaluations, and derives from them several design implications.**

**Keywords -- evaluation; prototype; heuristics; Nielsen; users; analytical; empirical; usability; user experience;**

### I. INTRODUCTION

Our group is designing a mobile-first web interface as a potential replacement for the National School Lunch Program paper-based application form. We have now conducted four independent Nielsen Heuristic Analyses, and three empirical testing sessions with target users. We will review our evaluation goals and methodologies, and then explore our key findings. The core data we gathered, along with the prototype which was used for evaluation, is presented in appendices.

### II. ANALYTICAL EVALUATION

#### A. Goals

As stated in project 6, our goals for the interface are: that it provide a friendly and welcoming user experience, that it reduce the error rate in applications, that it should tailor prompts and application flow in response to user input, and that users should feel comfortable answering all questions honestly and completely. Now some of these goals, particularly the first and last, are more about user experience rather than core usability. Consequently, our empirical testing with users is far better suited towards determining if we met those UX goals than heuristic analysis. Instead our analytical evaluation focuses primarily on our usability-oriented goals.

#### B. Method

We had planned to begin our evaluation process with a Nielsen Heuristic analysis, and then, once we had compiled our findings, to proceed to conduct user tests. As it worked out, we each independently conducted a heuristic analysis at the same time that user tests were being conducted. We deviated from our evaluation plan because our prototype was not ready for testing until fairly late in the week. We had

underestimated the amount of time it would take to get a detailed myBalsimiq pdf ready, and by the time it was ready, we didn't have time to wait for everyone to finish their heuristic analysis before starting user testing.

We selected heuristic analysis for our analytical evaluation primarily because we believe it offers a better benefit vs time-invested ratio than a cognitive walkthrough, and because we did not have ready access to the tools necessary for a GOMS/KLM analysis. A cognitive walkthrough would offer the benefit of allowing us to more explicitly empathize with our target user base during the analysis, but the fact of the matter is that our group does not much resemble our target user base, and so the results of a cognitive walkthrough would likely only be best guesses as to how our users would respond to the interface. Further, a full cognitive walkthrough can be quite time consuming, and given that we are uncertain that our assessments of the interface would correspond to our users' assessments, we feel that a cognitive walkthrough does not merit the expense in time. In contrast, a Nielsen heuristic analysis will be much quicker, and will provide a more general assessment of the interface, without trying to immerse ourselves in a user perspective we may empathize with but not fully appreciate or understand. Further, a GOMS/KLM analysis is not indicated because most users of our interface will be first-time, not skilled, users.

In contrast with our initial plan, we did not divy up usage scenarios between team members, but instead chose to make multiple passes through the application, trying out different options with each pass. We made this modification to our evaluation plan because once we had actually constructed the myBalsimiq prototype, it became clear that certain usage scenarios curtailed the amount of interaction with the interface, whereas others required more extended interaction. As a result, we felt that scenarios would just limit how much time some of us would spend analyzing the interface, and that it would be better for each of us to just fully explore the interface. However, despite this deviation, we did still use the questions we listed in Project 6 -- and which we present again in the Materials section -- to help guide our heuristic analysis.

To reiterate, each team member independently conducted a Nielsen Heuristic analysis of a myBalsimiq pdf prototype of our interface. Instead of using scenarios, we each did multiple passes through the interface, trying out different

options and application flows. We used a standard set of questions to guide, but not necessarily dictate, our analysis and response.

### C. Materials

A copy of our myBalsimiq linked pdf is in appendix H.

The guiding analysis questions are listed below:

#### Visibility of system status

- Does the user know how much progress they have made in the application?
- Does the user know when an error has been made?
- Does the user know when a response is optional?

#### Match between system and the real world

- Is the language friendly and welcoming?
- Is the language easy to understand?
- Are the questions clear?
- Does the help text answer all questions and concerns?

#### User control and freedom

- Can the user easily correct mistakes as soon as they are made?
- Can the user go back and correct mistakes in the form?
- Does the application support undo and redo?

#### Consistency and standards

- Are the controls placed consistently?
- Is the use of key words and prompts consistent?

#### Error Prevention

- Are dropdowns, checkboxes, and autofill used whenever possible?
- Are the confirmation screens used often enough?
- Are the confirmation screens in the right places?

#### Recognition rather than recall

- Is the location of the help button intuitive?
- Can the user navigate easily using the buttons?

#### Flexibility and efficiency of use

- Does the application logic and fast-tracking work?
- What happens if a user changes a response?

#### Aesthetic and minimalist design

- Are the questions clear and brief?
- Is the help text clear and brief?
- Are the fonts easy to read?
- Are the colors friendly and welcoming?

#### Help users recognize, diagnose, and recover from errors

- Do all of the inputs have real-time error messages?
- Can users correct mistakes without losing other information?

#### Help and documentation

- Is the help text complete and useful?
- Is the inline text complete and useful?
- Does the application need additional help screens (i.e. FAQs)?

### D. Analysis

See appendices A - D for the individual heuristic analyses.

### E. Results and Insights

As noted in the goals section, our primary interest for the analytical evaluation was determining if our interface was likely to reduce user application errors and if our interface offered a responsive path through the flow of the application. In general, the answer to both questions is yes, however there are a few problem areas, and the interface as a whole could still use more polish and fleshing out.

First, we are concerned about the completeness of all of our prompts. Specifically, some prompts do not give enough detail to allow users to fully understand why a specific piece of information is being asked for at that specific moment. Without such context, a user may fear that he or she, for example, has made a mistake when in fact all they did was trigger fast-tracking logic. Accordingly, more detailed prompts would improve user confidence that they are doing what they intend.

Second, we need a consistent way in which to deliver error messages to the user. Our current lack of error messages results from myBalsimiq's limited input entering capabilities: it's hard for a user to make a mistake when they can't enter input. As a result, our prototype currently ignores where and how to report errors to the user. But we could, even with little in the way of actual user input, decide on a consistent location and format for our error messages, and other system status updates, and fill the location with placeholder text. In fact, having some sort of system messaging area would allow us to also address some of the concerns behind needing more detailed prompts: we can inform a user when they've triggered a special condition, such as fast-tracking, which may not be immediately evident from prompts alone.

Finally, the implementation of our interface is not yet fully and consistently realized. Not all screens have help text, certain constraints are not consistently applied, and some parts of the interface are less responsive/reactive than others. These deficiencies primarily result from the prototype only being about 2/3s finished at the time of testing. That said, going forward, we must ensure that a given aspect of the interface behaves consistently throughout the entire application process.

### III. EMPIRICAL EVALUATION

#### A. Goals

Our goals for our empirical evaluation are similar to those for our analytical evaluation with one key difference: since we are testing with actual users, we can truly assess the user experience of our interface, and not just its usability. Accordingly, we are particularly interested in whether or not users find the interface friendly and welcoming, and whether or not they felt comfortable with and able to share the information asked of them.

#### B. Method

Our empirical user testing plan called for a minimal pre-test briefing of the user, a talk-aloud walkthrough of the interface, in which the user would attempt to fill out the application as though they were applying for the school lunch program, and finally a post-walkthrough questionnaire.

Our testing plan called for a myBalsamiq pdf prototype with “80-90% of the behavior of the final interface”. As it stands, our prototype probably has closer to 2/3s functionality. As noted in the section on analytical evaluation, we underestimated the amount of time necessary to build a full myBalsamiq prototype, and so we were unable to get all the functionality we had had hoped for in time for testing. Nevertheless, we were able to capture the big picture application flow, along with the general look and feel of the interface, and many, but by no means all, of the actual functional details. The result was a prototype that was less complete than we wanted, but was still completely suitable for user testing and much, much more sophisticated than what a lofi paper prototype, or something similar, would have offered.

As performed, our user testing otherwise followed our testing protocol without noticeable deviation. Below is a quote from our Project 6, describing our testing protocol:

“Our pre-test briefing for the user will be quite minimal. We will explain what the prototype is, an interface for a school lunch program application form, and that most of the interface can be clicked on. We will further encourage the user to talk through what they are thinking as they navigate the interface, providing a brief demonstration of the kind of thinking aloud we are looking for. Then, we will ask that the user begin exploring the interface in an attempt to complete an application for the user's family. If time permits, we will ask the user to imagine being a foster parent, or being an active recipient of government assistance, and then to go through the application again, so that we can see how the

different application flow affects their response to using the application. Finally, we will conduct a post evaluation interview, in which we will ask several questions to be detailed shortly.”

“As the user uses the interface, we plan to record the evaluation through observation and note taking. We do not have ready access to reliable recording equipment that could be unobtrusively used in the testing environment, and so simply watching and writing is our best option. We should note that the testing environment will be in the user's home. Further, although we will prompt the user if the user should go quiet, and although we will encourage the user if she mentions concern that she's letting us down by not understanding part of the interface, what we will not do is explain the interface to the user, or how to proceed with the application. While we do not anticipate the user getting stuck or lost, if that does happen we absolutely want to record where it occurs, and see if the user is able to work her way out of it. With that in mind, as we observe the user, we will pay special attention to any issues the user has with navigating the application, to confusion regarding instructions, to user concern about providing sensitive information, to the user's ability to recover from errors, to the amount of time the user takes to complete the application, to the number and type of obstacles/errors the user encounters, and finally to the nonverbal feedback which the user offers.”

Details on the users themselves occur in testing reports in appendices E – G.

#### C. Materials

See appendix H for the the myBalsamiq pdf prototype used during testing.

The contents of that questionnaire are as follows:

- What's your initial reaction now that you're done using this applications?
- On a scale of 1 - 10, with 1 being extremely easy, and 10 being extremely hard, how would you rate the ease of use of this application? Why?
- Did you ever feel lost or stuck? When? Were you able to get unlost/stuck?
- Did you ever feel uneasy about providing some of the information that was asked of you? When?
- How did you feel before you started the application? After?
- Did you feel prepared to complete each step of the application?

#### D. Empirical Data

See appendices E – G for the data gathered in user testing.

### *E. Results and Insights*

Although we had expected our user testing to primarily reveal user experience issues, we discovered in our testing that our interface's problems mostly center around usability, and any negative user experiences primarily result from deficient usability considerations. The following is a quote from appendix G, detailing what those usability problems are and how we might go about addressing them:

“Text prompts on some of the screens are either not clear or not helpful. User did not know what was expected particularly in the Sign and Date, Income, and SSN screens. The prompts should do more to communicate the intention of the screen. User should not have to click the help text every time. These prompts need to be re-written in clear, plain language.”

“Assistance Program response "I don't know" leading to Program Contact Info was confusing. It was not clear to the user why that screen was being presented or what the response "I don't know" meant in that context.”

“The welcome screen text should include a brief overview of the process, including what will be asked and what information it requires. This could eliminate some of the confusion about what was next and why it was being asked.”

“The sign and date screen should be at the end of the application. It was placed at the beginning because users of the paper application often forgot to sign and date the application. However, moving it to the beginning caused the user to be confused about why they were being asked to sign and date an application they had not filled out. It should also include text explaining its purpose.”

“The SSN help text is potentially confusing. It was written to alleviate concerns for users who do not have an SSN but it can also lead a user to wonder why they are being asked for that information when the help text says it is not required.”

“The application fast-tracking that causes some sections to be skipped should be explained at the beginning. Users who go back and change responses may be confused when they are taken to sections that were previously skipped. The navigation is also less intuitive when they are taken back to a section that was skipped.”

## Appendix A

Christopher Carrassi  
Project 7 - Heuristic Analysis

### *Visibility of system status*

#### **Does the user know how much progress they have made in the application?**

The user can track progress visually by using the progress bar along the top of the screen.

#### **Does the user know when an error has been made?**

Not in the prototype as it stands, but there's also no real way to make an error yet either.

#### **Does the user know when a response is optional?**

It could be made more explicit in some circumstances. Namely the social security screen and the contact screen, since it may not be obvious to someone they don't need to include a street address.

### *Match between system and real world*

#### **Is the language friendly and welcoming?**

Yes. The application uses friendly language, such as the help screen for the contact info page, which assures the applicant that it's OK if we can't provide all of the information up front.

#### **Is the language easy to understand?**

Yes, for native English speakers. We would need to add alternate language support otherwise. But for now, it uses very simple and straight forward English.

#### **Are the questions clear?**

I think the income screens could be a little more clear about the fact we'll need to ask this information for all of the children at once, and then all of the adults one at a time. I think we should also make explicit that the signature is not the end of the process, as that isn't currently clear to users. Otherwise I think the questions are simple and direct, focusing on just one thing at a time.

#### **Does the help text answer all questions and concerns?**

I don't think it helps answer all questions and concerns, but it addresses some of the most common ones, such as not having a SSN or not understanding what is meant by "income". The

application includes some links to program descriptions, although they don't currently work, but would be helpful. The limitations of the smaller device mean we can't exactly address everything, however.

### *User control and freedom*

#### **Can the user easily correct mistakes as soon as they are made?**

In theory, yes, although the current prototype doesn't allow for actual input (and thus does not allow for actual errors). However, the user can easily go back through the application if they realize they made a mistake to fix it.

#### **Can the user go back and correct mistakes in the form?**

Yes they can by simply clicking the back button, which is always in plain sight.

#### **Does the application support undo and redo?**

Not currently, but I don't think it would necessarily be appropriate to this sort of application.

### *Consistency and standards*

#### **Are the controls placed consistently?**

Yes, there are few controls and they are always in place at the bottom of the screen.

#### **Is the use of key words and prompts consistent?**

Yes, the app uses consistent phrasing and prompts throughout the process.

### *Error Prevention*

#### **Are dropdowns, checkboxes and autofill used whenever possible?**

To the extent that it is possible in our prototype, yes, these objects are used to simplify the experience of filling out the form for the user.

#### **Are the confirmation screens used often enough?**

It may make sense to add confirmation screens after the child info input procedure as well as the adult info input procedure, as right now there is only one confirmation at the end.

#### **Are the confirmation screens in the right places?**

Yes, having it at the end of the process currently makes sense.

*Recognition rather than recall*

**Is the location of the help button intuitive?**

Yes, it's right in the middle at the bottom the entire time.

**Can the user navigate easily using the buttons?**

Yes.

*Flexibility and efficiency of use*

**Does the application and fast-tracking work?**

It seems to send any user who qualifies for food stamps or other programs, or any family with a foster child, straight to the confirmation screen so I would say yes. Every other type of applicant routes through the application normally.

**What happens if a user changes a response?**

Right now, not much, because a user cannot enter or change their input beyond simple checkboxes. Changing those simply checks or unchecks the box, as expected.

*Aesthetic and minimalist design*

**Are the questions clear and brief?**

Absolutely yes.

**Is the help text clear and brief?**

In some cases yes, in others the help text is really wordy. Have to find the balance between information needed and information that can actually be digested.

**Are the fonts easy to read?**

Yes. Simple clean, although very comic sans-like, which we wouldn't use in the final product.

**Are the colors friendly and welcoming?**

Being a prototype, there aren't many colors, but it's welcoming enough in black and white.

## *Help and documentation*

### **Is the help text complete and useful?**

What is complete is pretty useful, but not all of it is complete. Need to add help screen for SSN and signature pages, namely.

### **Does the application need additional help screens?**

I don't think so -- there should be a way to bake that into the process organically, so that at each step the necessary help can be found. Additional details in the help buttons or other on screen prompts should address that.

*Visibility of system status*

- Does the user know how much progress they have made in the application?

Yes, there is a progress bar on top (no numbers though)

- Does the user know when an error has been made?

At this point in the prototype, it is hard to determine any errors made.

- Does the user know when a response is optional?

Not really. The user might be able to see a check-box and realize that they can choose one or more options. Other than that, there haven't been a lot of indications for optional answers.

*Match between system and the real world*

- Is the language friendly and welcoming?

The layout is currently only in English and is just a simple black-white text, so it is hard to say. It certainly doesn't feel unwelcoming, though.

- Is the language easy to understand?

For English users, yes.

- Are the questions clear?

From our perspective, yes, the questions appear to be clear. The myBalsamiq prototype is simplified and lacks some functionality, so there may be some area where the questions are unclear.

- Does the help text answer all questions and concerns?

For the ones that are available, they do a decent job of explaining the page's current situation. A little more information wouldn't hurt, but they work as it. Maybe they could have a link to another page where more detail is given.

*User control and freedom*

- Can the user easily correct mistakes as soon as they are made?

There is a back button that should work in case mistakes were made. Something that is not implemented- maybe we could give the users an option to go back and fix their form, even if they already signed and submitted the application.

- Can the user go back and correct mistakes in the form?

There is a back button for the users to use during the application process.

- Does the application support undo and redo?

There is a back button, which the users can return to a previous page and fix some information. At this point, there is no redo button. Maybe our app can support a redo option if the user's phone has that option.

*Consistency and standards*

- Are the controls placed consistently?

Yes, the back/help/next button are always at the bottom.

- Is the use of key words and prompts consistent?

The current prototype is simple with not much fleshed out. In its current state, things seem to be consistent.

*Error Prevention*

- Are dropdowns, checkboxes, and autofill used whenever possible?

Dropdowns and checkboxes are noticed in the prototype. Autofill could be implemented in places where typing is required.

- Are the confirmation screens used often enough?

We have only one confirmation screen, which is at the end. At the moment, this seems to be all that is necessary, but we could definitely implement more if it is necessary.

Are the confirmation screens in the right places?

Yes, I believe so. We have a confirmation page at the end so that the user could check their inputted answer one last time before doing an official submission.

#### *Recognition rather than recall*

Is the location of the help button intuitive?

I think it is. It is located where some of the main buttons are (at this point, just back and next).

Can the user navigate easily using the buttons?

The buttons are easily located and easy to use, but the current prototype does not support jumping from one page to another easily. Traversing through the application requires the constant use of the back/next button.

#### *Flexibility and efficiency of use*

Does the application logic and fast-tracking work?

The logical flow of the application isn't flawed, but could always use more work. It is not easy to use fast-tracking given that all we have are back/next buttons.

What happens if a user changes a response?

Any changes in responses will be finalized once the form is confirmed/submitted.

#### *Aesthetic and minimalist design*

Are the questions clear and brief?

Yes.

Is the help text clear and brief?

Yes.

Are the fonts easy to read?

Yes.

Are the colors friendly and welcoming?

Yes – color is just black and white. It could maybe change to look more welcoming.

#### *Help users recognize, diagnose, and recover from errors*

Do all of the inputs have real-time error messages?

At this stage of the prototype, this cannot be determined.

Can users correct mistakes without losing other information?

I believe the users should be able to make changes without losing information.

#### *Help and documentation*

Is the help text complete and useful?

I think the help isn't incomplete, but could maybe have more content to it. I believe it is currently useful, but could be even more so when the final product comes out.

Is the inline text complete and useful?

Yes.

Does the application need additional help screens (i.e. FAQs)?

I think it would help for the application to have additional help screens, but we currently do not have that implemented in our prototype.

## Appendix C

### Project 7 Heuristic Analysis by Jared Storts

#### Visibility of system status

The progress bar is visible and consistent across the top of all screens except the initial login screen. It may not be clear that some of the contact methods are optional. There is a help screen indicating this. The signature screen might need some help text. There will be real-time input validation on all form fields to give users immediate feedback about errors.

#### Match between system and the real world

The language is clear and simple. The tone of is welcoming. The organization of the form might be confusing. The presentation of the signature screen early in the form could be confusing because it is usually at the end of the paper form.

#### User control and freedom

Navigation through the application is consistent and intuitive. One of the controls ("I don't know" checkbox in Assistance Program screen) is "stuck". If selected, it cannot be deselected.

#### Consistency and standards

Navigation and help buttons are placed consistently in each screen. There are a few help text items that are accessed through linked text or a question mark icon. These are conventional devices. One of the screens in the pdf prototype is significantly smaller than the others.

#### Error Prevention

The application might require more than one confirmation screen. The purpose of the screen is to allow users to confirm data that is inaccurate but not otherwise invalid. This data would not be flagged by real-time validation.

#### Recognition rather than recall

The button placement seems intuitive. The use of colors or shapes might further improve user perception of their functions. The modular nature of the form might lead users to be confused about where they are in spite of the progress bar.

#### Flexibility and efficiency of use

The fast-tracking logic is functional. The prototype cannot adequately capture the back-tracking necessary if users change a response that had previously allowed them to skip a section of the application. User stories involving corrections and fast-tracking should be developed and tested.

## Aesthetic and minimalist design

Many aesthetic features such as color and typography cannot be evaluated in a prototype at this fidelity. The design features that are present such as prompts and inputs are minimal and focused.

## Help users recognize, diagnose, and recover from errors

Confirmation screens and real-time input validation are the primary methods for recognizing and diagnosing errors. It will be necessary for users to navigate to previous screens to correct input without confusion or loss of data.

## Help and documentation

The help text screens have definitions of jargon and in some places clarify the purpose of the screen. Additional help text

### Visibility of System Status

The progress bar gives the user an indication of how far into the application they are, however it does not currently handle the fact that different users will have different length applications, depending on their family circumstances, very well. As implemented, the progress bar indicates the users progress through the application but it assumes they have to complete every screen. If a user only ends up needing to fill out part of the application, then the progress bar jumps around pretty wildly as they skip over unnecessary sections.

### Match Between System and the Real World

The interface currently behaves like a dialog between the user and the application, rather than like filling out a form, but the prompts are more in line with what you see in a form than in a dialog. Forms allow someone to see an entire application at once, and so therefore contextualize each form field as a part of the whole. This contextual information allows a form to stick to fairly terse prompts. In contrast, a dialog, like in this interface prototype, loses a lot of the contextual information, because the user's attention is drawn to the specific question being asked in isolation, and cannot even see the entire application. This is not necessarily a flaw, but dialog based prompts need to be richer in context and more conversational, so that the user can understand what's going on. As implemented, this interface's prompts do not have enough contextual detail to make up for the lack of complete application visibility, and could further stand to have a more conversational tone that matches the dialog-like interaction.

### User Control and Freedom

The interface keeps a pretty tight leash on user actions, allowing them to go backwards, but only allowing them to go forwards after they've completed the next step in the application. The interface does not have native undo/redo support, however the ability to move backwards and forwards again through previous prompts, along with the end of application review, somewhat obviates the need for undo/redo functionality.

### Consistency and Standards

The current prototype has some inconsistently implemented interface conventions, such as its use of graying out currently unavailable buttons. Further, it uses some web form elements in ways that are counter to how they are typically used. For example, checkboxes are used for a mutually exclusive choice of Yes | No | I don't know, when radio buttons would be more appropriate.

### Error Prevention

The current interface design limits the potential for error by preventing the user from proceeding until they've entered valid input. However, as currently designed there is no review of what the user has entered until they've completed the entire application, making it more likely that while reviewing they will skip over any errors due to information overload. On the other hand, since the total quantity of information collected is relatively small, it would likely feel overly intrusive to review input throughout the application.

### Recognition Rather than Recall

The memory load imposed by the interface is fairly minimal. The only information the user has to actively recall is their own personal information for entry into the application. The limitations

## Appendix D: Heuristic Analysis, Amanda Tait

imposed on what the user can actually do on any given screen, along with prompting for what's being asked, reduces, if not eliminates, any need for the user to remember what to do.

### Flexibility and Efficiency of Use

As designed, this interface offers very few opportunities for a user to become faster and more proficient in its use. However, since a given user is only likely to use the interface once, this lack of room for growth as a user is less of a problem. Further, the fast-tracking logic appears to be triggered at the correct moments to move the user more rapidly to the end of the application. That said, as currently implemented, the fast-tracking comes off as rather abrupt, and the user isn't really informed why they're already at the end of the application.

### Aesthetic and Minimalist Design

The interface is uncluttered and clean in its visual aesthetic, removing nearly everything but the most important information. However, sometimes this has been taken a bit too far, and the user isn't given adequate context for what they're filling out and why. The visual design may need to be reorganized to accommodate more descriptive and contextualizing text prompts for the various form fields.

### Help Users Recognize, Diagnose, and Recover From Errors

The current interface design is a bit sparse on error messages. Partly this a function of the limited interactivity of a myBalsimiq prototype: it's hard to implement error messaging when the user can't actually enter erroneous information. A consistently styles and layed out messaging area should probably be set aside on each page, even if it's only filled with a non-interactive placeholder.

### Help and Documentation

As currently implemented, not all screens of the interface have help information. The design currently features a centrally located help button on every screen, however it only actually provides help on a handful of the given screens. Further, on some screens, the help button is grayed out when its unavailable, and on others it is active but non-clickable. At the least, screens without help need to have the help button greyed out, but it would be preferable for all screens to have relevant, clear help information.

## Appendix E

questionnaire:

What's your initial reaction now that you're done using this applications?

The phone app was easier to fill out than the form.

On a scale of 1 - 10, with 1 being extremely easy, and 10 being extremely hard, how would you rate the ease of use of this application? Why?

3 - There weren't a lot of words to read, and it was fast to fill out.

• Did you ever feel lost or stuck? When? Were you able to get unlost/stuck?

I never felt stuck. The phone app was easy, just answer the questions and hit "next.

If you made a mistake you could hit "back" or "help," so you can't get lost.

• Did you ever feel uneasy about providing some of the information that was asked of you?

When? •

I didn't know why it asked for my social security number at first, but since it was only the last four digits it was okay.

How did you feel before you started the application? After? •

When I started I didn't know how long it would take but it went by really quick so it was good.

For me maybe a real phone would be too small.

Did you feel prepared to complete each step of the application?

Yes, it didn't ask for anything I didn't know. I could fill it out off the top of my head.

info:

Name: Andres Interiano.

Age: 46

Occupation: construction worker.

lives in pacoima, ca.

wife is nancy, part time office assistant. 2 kids, andres and monica. andres is sophomore at valor high, monica a fifth grader valor middle.

she has filled out form before but not him, he is familiar with it.

"clean, simple"

(I explain that the fuzzy text talks about what to expect and how to prepare.)



Paused for a second and then just selected e-mail without prompting.



Again just clicked on next without seeming to worry about needing to provide anything else.



Asked "do I need to enter my email?"  
(I explained that he didn't literally, but that we would pretend he did.)



Had a puzzled look when he asked “Why do I have to sign this now?” (I said to pretend he was at home, what would he do?)

He answered “I would click on help”, which he did. Nothing happened.

(I provided an explanation about why we designed it this way--research, better accuracy, etc -- and tried to provide roughly what the help text would say)

“The explanation should say why you want me to sign now. Otherwise I think this is the end already.”

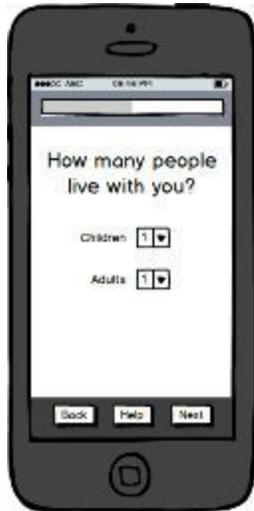


Read the text. Tried to tap on the links. (I explained that they would be links to the program descriptions and I provided them to him instead.)

He said “Oh we don’t have those so I will click no” and he clicked no.



Clicked next without prompting.



Asked “Do I have to fill it in?” (I explained we would pretend again)

“OK, I would put in 2 adults 1 child.”

(I informed him that this tally would include his wife as well, so it should be at least 2 adults.)



Asked “Would I do this for both of my children?” (I explained yes, but we’ll just simulate it for now).

He did not attempt to check any of the categories listed or ask about the name fields.



Said “My children don’t make any money. They’re job is to go to school.”

(I asked, so what would you do?)

“Put zero.” (I explained that we would pretend he did.)



“I would enter my weekly income?” (I explained yes, if you knew it) He said he did.

(I explained we would pretend he did enter it, but that he would need to do this again for the other adult in the household)



Some confusion about what “Other” means: “Is this for my wife?” (I explained it was other type of income, that he would have been prompted to enter the income for every adult in the household)



“So I pretend to enter it in?” (I said yes.)

Selecting "Yes" confirms information.  
Selecting "No" allows users to reselect information as they see fit.  
Selecting "Next" takes user to subsequent screen. (You will need to make this?)



(Explained that this was a summary screen of what he would have put in so far)

“OK. I like that, that’s helpful.” (I explained he would then click yes if it was correct and that it would take you to a success page).

## Appendix F

Tony Huynh

CS 352 Winter 2016

Project 7- Testing Notes

User Testing Notes

Tester: David Calhoun

---

This isn't a verbatim copy of my raw notes. Questionnaire is at end.

- “Having to click twice can be annoying” –In response to clicking on e-mail and then clicking next.
  - Advice- Add in a “Click all that apply” button for clarification.



2.0.0 Initial Contact Info ▾

- Let “Address” come in before “zip code”



3.3.0 Address Contact Info ▾

- Next button is nice. Calendar is nice.



4.0.0 Sign and Date ▾

- Asked if there were links. Does paper form require info on SNAP, TANF, and/or FDPIR?
  - Suggested this layout for answer:
    - Does anyone in the household get help from:  
[we might want to take “from” out to make the suggestion make more sense]
      - Yes (SNAP, TANF, FDPIR. Click link for more information)
      - No
      - I don’t know



5.0.0 Assistance Program Prompt



- Asked “Is what correct?” Likely asked this because it was a prototype.  
“Will info be imported or inputted?” Asked this because our first run skipped a lot of the address/mail input.
- Suggested that we either combine the signature page with the confirmation page or place the signature right after the confirmation page.



Confirmation ▾

## Post-Testing Questionnaire

- What's your initial reaction now that you're done using this application?  
Good start, but needs work.  
Number 1 suggestion: Get copy of form, then follow the form as closely as possible.
- On a scale of 1 - 10, with 1 being extremely easy, and 10 being extremely hard, how would you rate the ease of use of this application? Why?  
7, 7.5 – understanding that the prototype is still a work in progress. Difficult because it doesn't closely follow the paper form. Some things are reversed [*talking about the order of information that was input*]
- Did you ever feel lost or stuck? When? Were you able to get unlost/stuck?  
No.
- Did you ever feel uneasy about providing some of the information that was asked of you?  
When?  
No more than filling out a paper form. If SSN is optional, make a note about that for the users.
- How did you feel before you started the application? After?  
No change in feeling from before to after.  
*[He didn't know how to respond, so I had to direct him a bit (I one of the interview rules a few times.), so I told him how some people felt nervous/shame filling out the form.]*  
He said he would feel the same when doing it on paper.
  - There may be a concern that hackers can gather personal information through electronic form.
- Did you feel prepared to complete each step of the application?  
Yes.

## Appendix G

### Empirical Observation

User: Single Mother, age 34, with three children ages 2, 5, and 8. Does not have any prior experience with the application for free-and-reduced lunch.

Observer: Jared Storts

#### Introduction

I gave the user a brief explanation of the purpose of the application, explaining that there was placeholder text in some areas and I would read example text in these areas. Otherwise, the user should think out loud as they go through the application and pretend that I was not there.

#### Transcript (by Screen)

Login:

Ok, I'm clicking the button to begin.

Contact Info:

It should say something like what is your preferred means of contact. What if I want to select more than one of these? Do I select one of these? Can I select all of them? What happens then?

Sign and Date:

I'm kind of irritated. I don't really know what I'm signing and dating. That seems like something that would be at the end of the application. Let's see what help says. Help doesn't say anything. I don't like the sign and date. It's confusing. What are you signing and dating? I haven't done anything yet.

Assistance Programs:

Does anyone in the household get these things. I checked I don't know. Why am I going here? If I don't know or if the answer is No... I was expecting the same screen as if I said No. Are you expecting me to go contact these people and find out? It isn't clear what I should be saying.

Social Security Number:

Social Security Number. I'm confused by this. What does this mean? Social security number. I'm assuming this is for the adult but it should be more clear. When I click the help button, it says "It's OK if you don't have one of these." Why are you asking me then? What is it doing here?

Household Members / Child Info:

This is fine. This makes sense. So, wait, if I say I have three children, I'm going to see three of these screens?

Income:

Children? How much do my children make? You know, what I think is confusing about modern technology is that you have to click a help button to understand what it is asking you to do?

I feel like there should be more text at the beginning that says what you are supposed to be doing. Income, this is not clear at all. Why is everything in the help thing? Why doesn't it just say "Do your children make anything?" and then "Adult. Do you have any money coming in?"

Confirmation Screen:

Is this correct? What happens if I click Yes? What if I click No? [I explain the placeholder text] I clicked back from the confirmation screen. This is very confusing now. Maybe it's not working right. Now it's asking for income. Now I'm lost.

### Post-Observation Interview

Q: What's your initial reaction now that you're done using this application?

A: I don't know if I qualify or not. What do you mean?

Q: On a scale of 1 - 10, with 1 being extremely easy, and 10 being extremely hard, how would you rate the ease of use of this application? Why?

A: Umm (long pause), I would say 5. It's not entirely built, but it's not entirely clear some of the steps but I'm pretty confused with most applications. I think it should say something at the beginning that says something like "If you qualify for one of these things already, this will be easy. If you don't that's ok, we're going to get some information." It seems like there's different paths and it's not clear which one you're taking.

Q: Did you ever feel lost or stuck? When? Were you able to get unlost/stuck?

A: Yes. Yeah, sometimes the help button was functional and gave some information about what was being asked when it wasn't obvious.

Q: Did you ever feel uneasy about providing some of the information that was asked of you? When?

A: Yes, my social security number. It wasn't clear why you needed it when the little question mark said you didn't need it to qualify. Although it was just the last four digits but why do you need any of the digits?

Q: How did you feel before you started the application? After?

A: What do you mean? I don't know how to answer the question. I felt frustrated at some points.

Q: Did you feel prepared to complete each step of the application?

A: No, I did not. The children's income. If my kids did work, I don't think I would know. I don't think my mother knew when I was working or how much I made. I think she had an approximation. If I was an hourly worker, I don't know if I would know that figure off the top of my head. The income screen is confusing. The children's income is confusing too. I don't know what benefits means.

### Insights

1. Text prompts on some of the screens are either not clear or not helpful. User did not know what was expected particularly in the Sign and Date, Income, and SSN screens. The prompts should do more to communicate the intention of the screen. User should not have to click the help text every time. These prompts need to be re-written in clear, plain language.
2. Assistance Program response "I don't know" leading to Program Contact Info was confusing. It was not clear to the user why that screen was being presented or what the response "I don't know" meant in that context.
3. The welcome screen text should include a brief overview of the process, including what will be asked and what information it requires. This could eliminate some of the confusion about what was next and why it was being asked.
4. The sign and date screen should be at the end of the application. It was placed at the beginning because users of the paper application often forgot to sign and date the application. However, moving it to the beginning caused the user to be confused about why they were being asked to sign and date an application they had not filled out. It should also include text explaining its purpose.
5. The SSN help text is potentially confusing. It was written to alleviate concerns for users who do not have an SSN but it can also lead a user to wonder why they are being asked for that information when the help text says it is not required.
6. The application fast-tracking that causes some sections to be skipped should be explained at the beginning. Users who go back and change responses may be confused when they are taken to sections that were previously skipped. The navigation is also less intuitive when they are taken back to a section that was skipped.