CSEP510, Spring 2012
Final Project Report

Restaurant 2.0

Team:

Krists Auders
Marc Beitchman
Danny Draper
Dirk Groeneveld

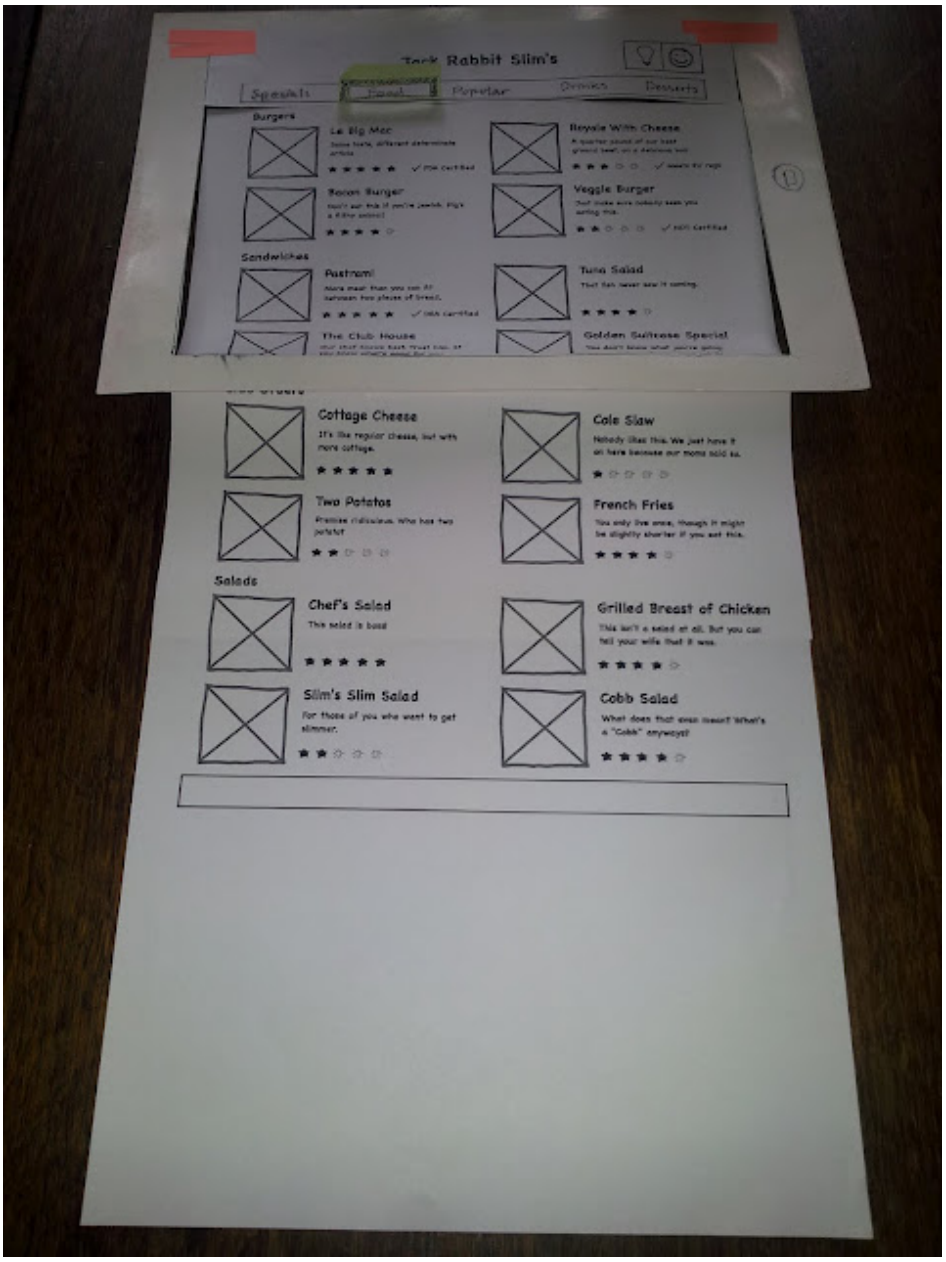
Problem Overview and Solution:

Although restaurants currently make use of software systems and services for some common tasks, there are many areas of operating a restaurant where software has not been considered to improve the dining experience and efficiency of restaurant operations. The goal of the Restaurant 2.0 project is to explore and discover ways that current developments in ubiquitous computing can improve restaurants. The following will report explain the findings of the project and explain the process for the design of the system interface.

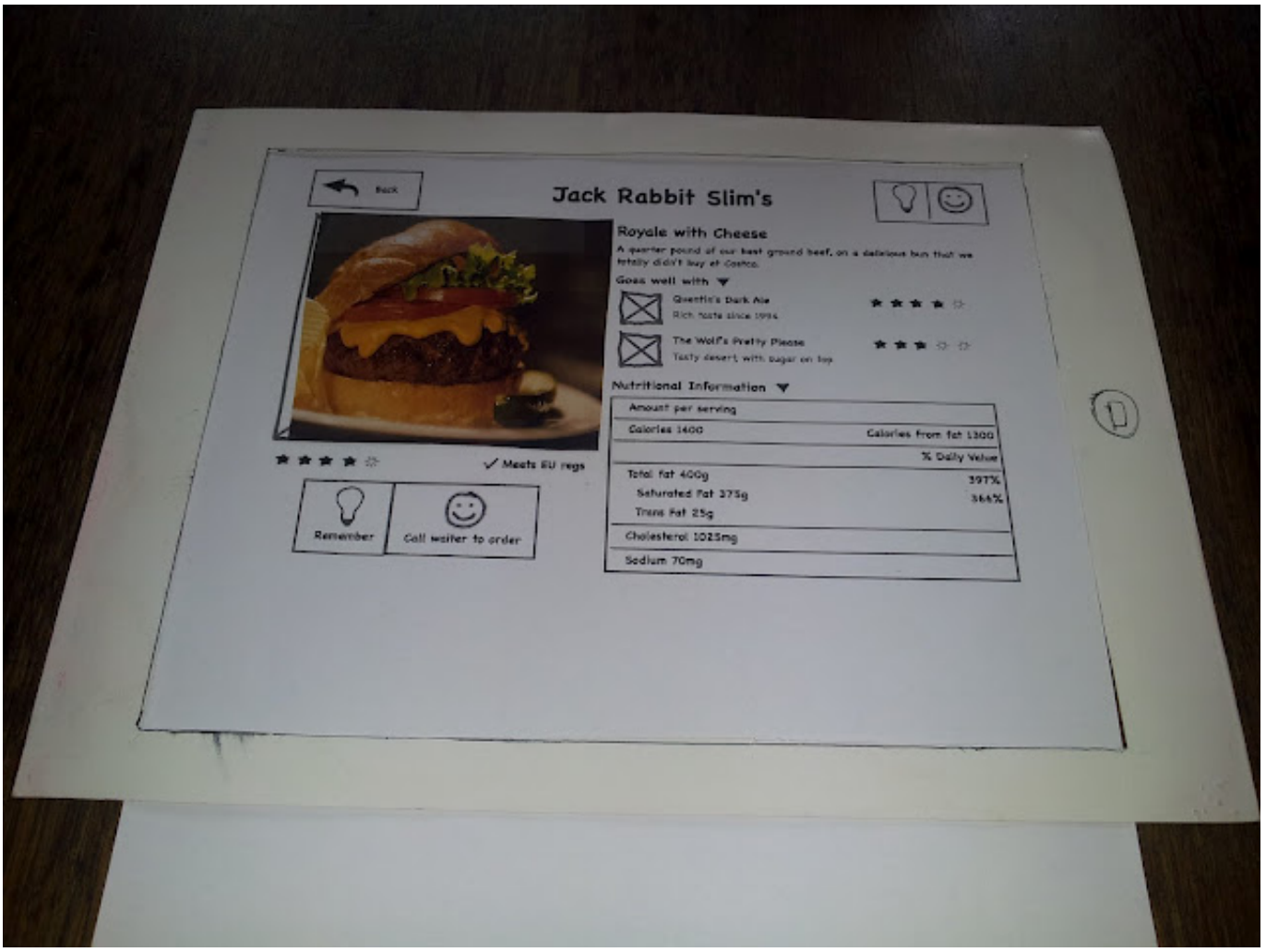
Paper Prototype

The paper prototype completely supports the three tasks that we came up with. The menu has the specials page, the entire food section (with scrolling!), it has detail pages for three items, and it has all the little pages and popups, like the page of remembered items and the “Call Waiter” popup.

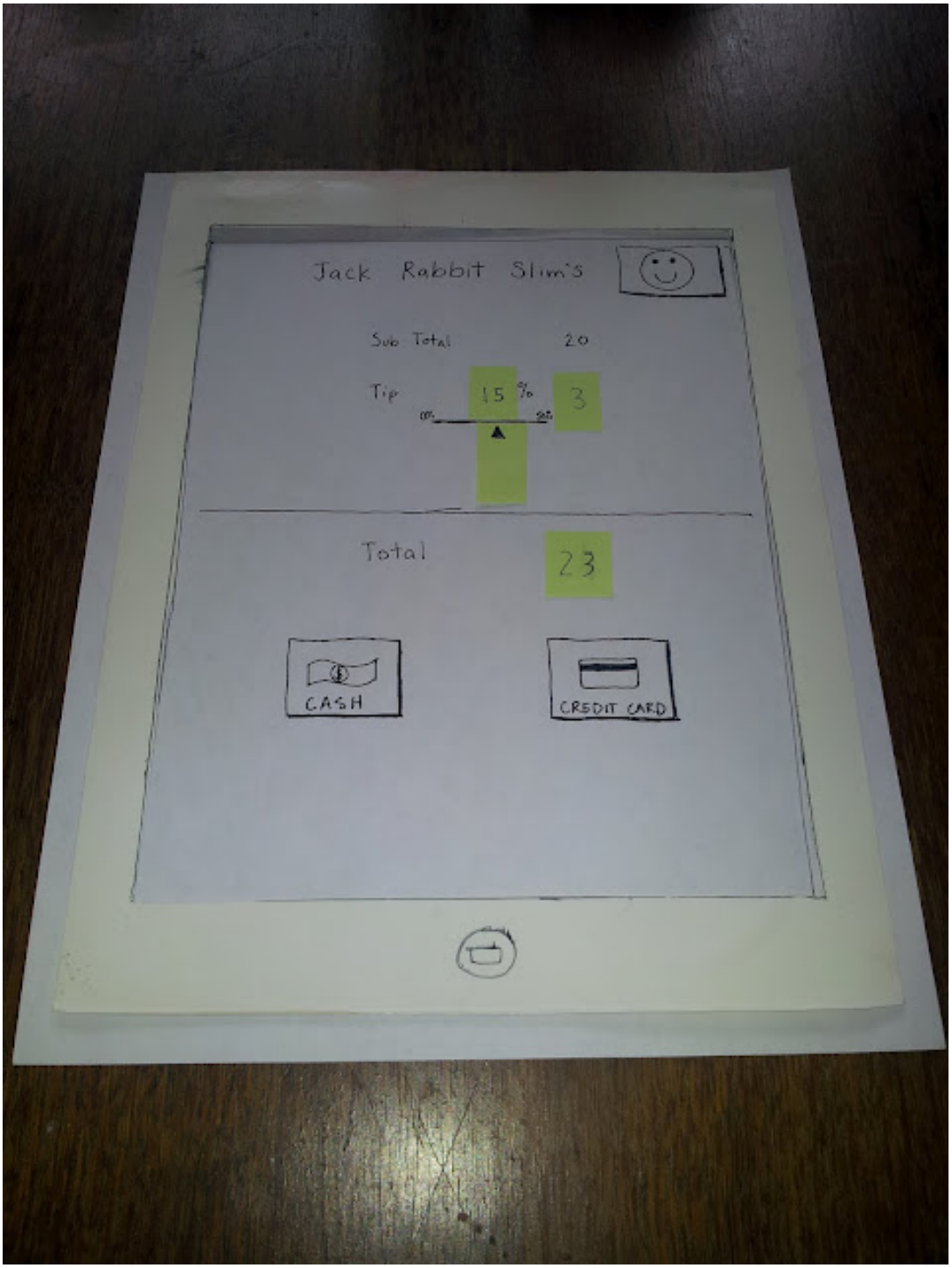
This is the food menu:

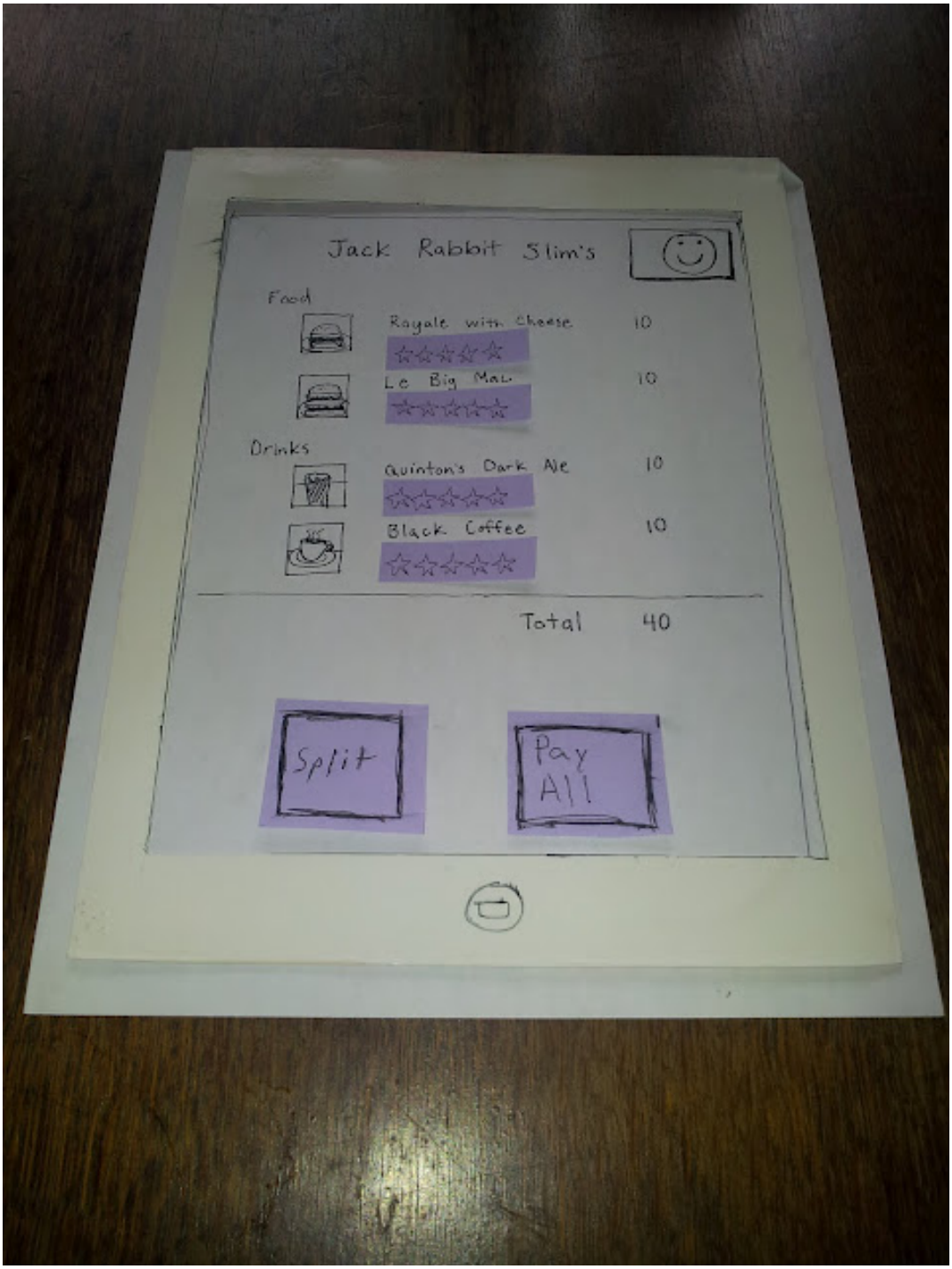


The next picture shows one of our detail pages. This one was printed from Balsamiq, but another was drawn by hand, because that was quicker.

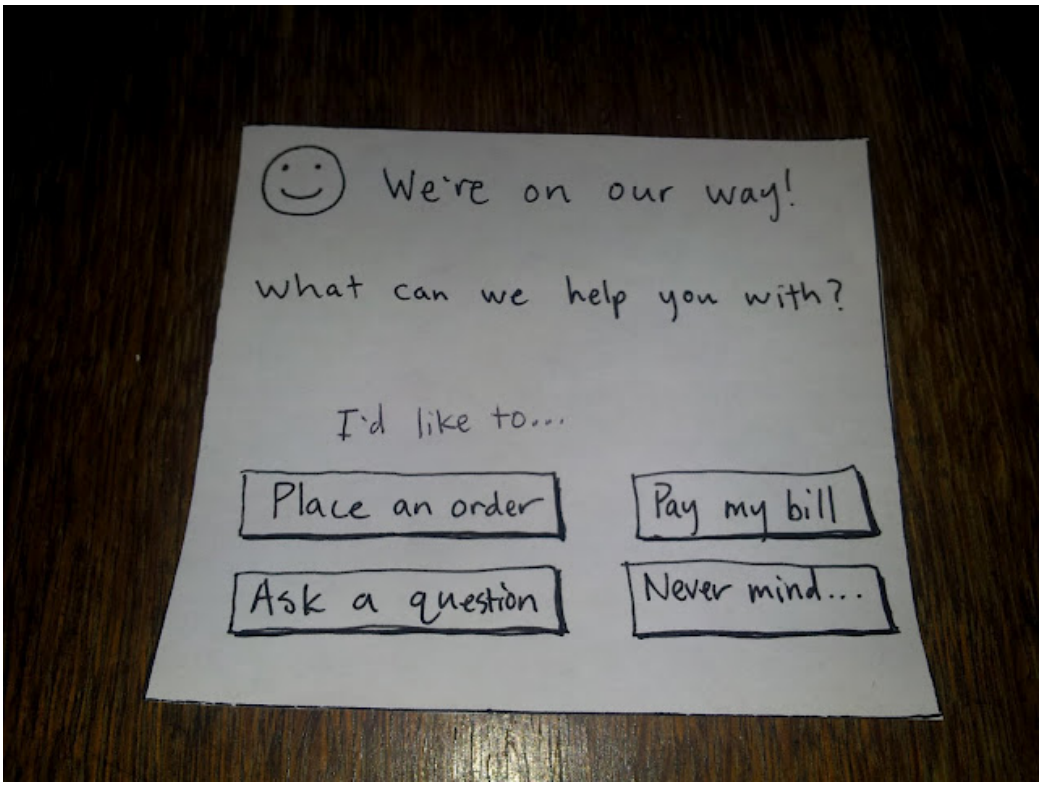


The payment part supports almost all interactions, because our tasks hit upon every aspect of it, and it's not as content-heavy as the menu is. Of the two following images, the left one is the main payment screen, and the right one is the tipping screen.




Jack Rabbit Slim's
Food
Rayale with cheese
10
Le Big Mac
10
Drinks
Quinton's Dark Ale
10
Black Coffee
10
Total
40
Split
Pay All

There were various smaller parts of the prototype, such as popups and little post-its that would make the buttons light up when they were supposed to be lit:



We tried to lay out all parts of the paper prototype in one picture, but it was too large to make out any details, so we skipped that step.

All parts of the prototype involve a cardboard version of an iPad, with the screen area cut out. The content is then put behind the frame to make it look like an iPad. Since most of the content is static, it was either hand-drawn, or printed out of Balsamiq. A few elements, such as changeable buttons, or lists with changing contents, were put together from post-its that were cut to size. Finally, the scrolling main menu was done with a piece of paper that's too large for the frame, and could be pulled up and down behind it. A silent member of the team who would play the computer facilitated the interactions with the prototype.

Testing Method

Participants

For our paper prototype testing, we selected 3 participants to test out our design. All three participants were known by us ahead of time, and were selected as "good fits" for typical users we would expect to be customers at the type of "gastropub" style restaurant we are designing for. Our first participant, Mary, is an accountant living in the Seattle area. She considers herself a reasonable judge of restaurants and eats out several times a week. Our second participant was Will, a local tech worker in his mid-thirties. He describes himself as "very opinionated" when it comes to food and restaurants, and is very picky when it comes to being satisfied with restaurant service. Our final participant was Lisa, a designer living in the center of Seattle. Lisa also frequently eats out at restaurants, but is much less picky when it comes to her dining preferences, often not stating preferences one way or another.

Environment

The environment we chose to administer the paper prototype testing was in a local Seattle café. The café serves coffee, wine and beer, as well as food. The service is customer-driven, so there aren't servers running around the main floor space. We decided on this setting as it provides several large tables for collaborative working (allowing us to actually set-up and use the prototype), but also didn't leave the user in a sterile environment. The sound of the environment is very similar to what we would expect in a target restaurant, and there is also the presence of other people in close proximity, as well as food and beverages. Overall, we felt the location proved very successful in both putting our participants into the desired mindset while also allowing us to successfully operate a prototype successfully while garnering the feedback we were after.

Tasks

Our first task was the most simple. Our goal was to simulate a common experience of being a repeat customer to the restaurant, and using the system to order what you already know you want to order. This task should be able to be performed very quickly with little or no difficulty.

The task was written as follows:

You arrive at your favorite pub, you know exactly what you want, and you want to get it fast. Order the Royale with Cheese.

The second task returns again to the Menu interface. However, this time the task simulates a user who is less familiar with the menu and has more particular preferences in what they are interested in eating. This task requires the user to consider several items, ultimately deciding based on specific details of the item, and then ordering. This also presents the user with a chance to use the "Remembered List."

The task was written as follows:

You have just arrived at a new pub, and you'd like to order something to eat. You're considering both Le Big Mac and the Royale With Cheese. You decide that you want the biggest and baddest burger, so decide on the one with the most calories. You also think that a side of fries sounds really good too, so add that as well. When finished deciding what to order, go ahead and call the waiter to order it.

The final task involved the payment system. This task adds more complexity, as the user has to navigate a more complicated UI to solve a more complicated task. The ultimate goal is to split and pay for a portion of a bill, along with rating a couple of items. We want this more involved task to still have a very natural workflow to it.

The task was written as follows:

You have just finished your meal with two of your friends and you're ready to pay the bill. You wish to split the bill between all three of you, so go ahead and pay for your part first. You loved the Royale with Cheese, so give that a 5 star rating before you pay. Your coffee wasn't very good, so give it a 1 star review.

Procedure

The procedure for testing varied slightly depending on how many members of our group were present during the testing. For our first participant Mary, all four members of our group were involved. While four people administering the prototype evaluation can be a bit overwhelming to the participant, we did our best to minimize the impact.

One of us had the sole role of proctor for the evaluation, handling communicating the tasks, talking with and answering questions from the participant, and generally being in charge of making sure the evaluation moved smoothly. Two of us acted as "The System." This meant we did not interact with the user, and solely focused on working the prototype in response to the user's input. Having two people here allowed us to split portions of the prototype between us, so each of us acting as the system were responsible for a smaller subset of the entire prototype. Finally, our last group member acted as an observer, neither interacting with our participant or the prototype. Additionally, our observer sat slightly farther away from the participant, as to not have too overwhelming of a presence. The observer was responsible for taking notes both of what the participant did, as well as said. Overall, this coordinated strategy proved to be very successful.

For our second and third participants, we used fewer group members at once. With both Will and Lisa, only two group members simultaneously were administering the procedure. In this case, some roles were merged. One of our members still acted as the proctor, being the person directly interacting with the user, but also added the note-taking responsibility of the observer. The other group member then assumed all the responsibility of "The System." This role remained relatively the same, except that now the entire prototype's operation was the responsibility of a single member. We found this operation to also be successful, although it did slow the overall system "performance" some, as the proctor had to stop to take notes on occasion, and the system operator had to manage more paper than before. Despite the slight slowdown in responsiveness of the prototype, we felt that the quality of feedback received from the participants remained high.

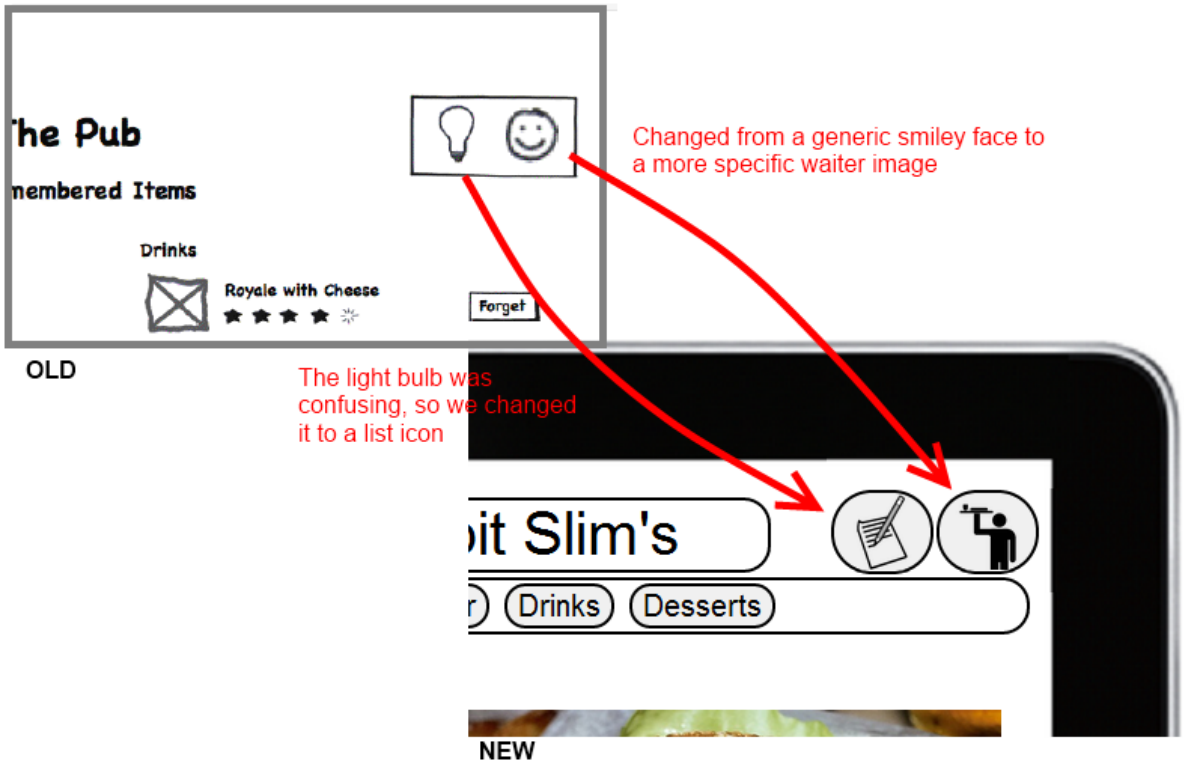
Test Measures

We measured the results of our test in a few different ways. While performing no "hard" measurements like time-per-screen or something along those lines (which didn't seem appropriate for our prototype), we did make more heuristic-style observations. One of our metrics was recording which screens the users went to, and if that screen was essential to the task. Also, we recorded if the user made any mistakes in accomplishing the tasks. Another measure, which we held in very high regard, was actually asking the user (after they completed all the tasks) if they liked the system, particularly if they would frequent a restaurant that used our system. The ultimate goal of our design is to improve the customer experience while dining, and if our prototype users didn't find the design useful or compelling, then we knew we had failed.

Testing Results

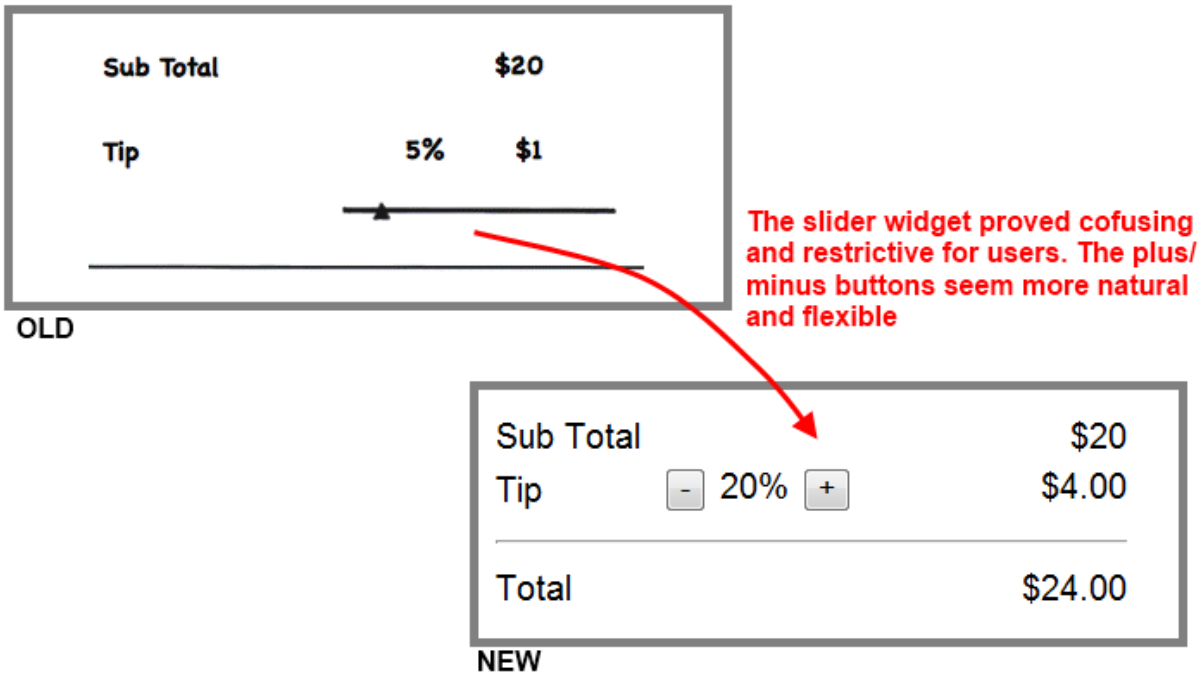
We tested with three people we know, but who hadn't seen any parts of our project before. We gave each of them all three tasks and took notes of the steps they took to accomplish the tasks, along with other notes. We learned that the Remembered list was not obvious enough, or at least it wasn't obvious that it existed, but this might have been our fault for not giving enough context of the prototype itself and how the user would engage with it. It was also mentioned that it's redundant to ask what the user needs, if they're calling a waiter already anyway. We also found out we didn't have prices on the specials pages, and on the main food list.
We also learned that it might be useful to have a back button from the payment success page back to the payment page where the user can rate items. Users wanted to remember items from the main menu without having to click through to the details. The light bulb and smiley face buttons were not obvious to most users, so we should change the icons or describe them somehow. One user requested being able to split the bill by an integer, instead of by item.

Interface Revision Sketches



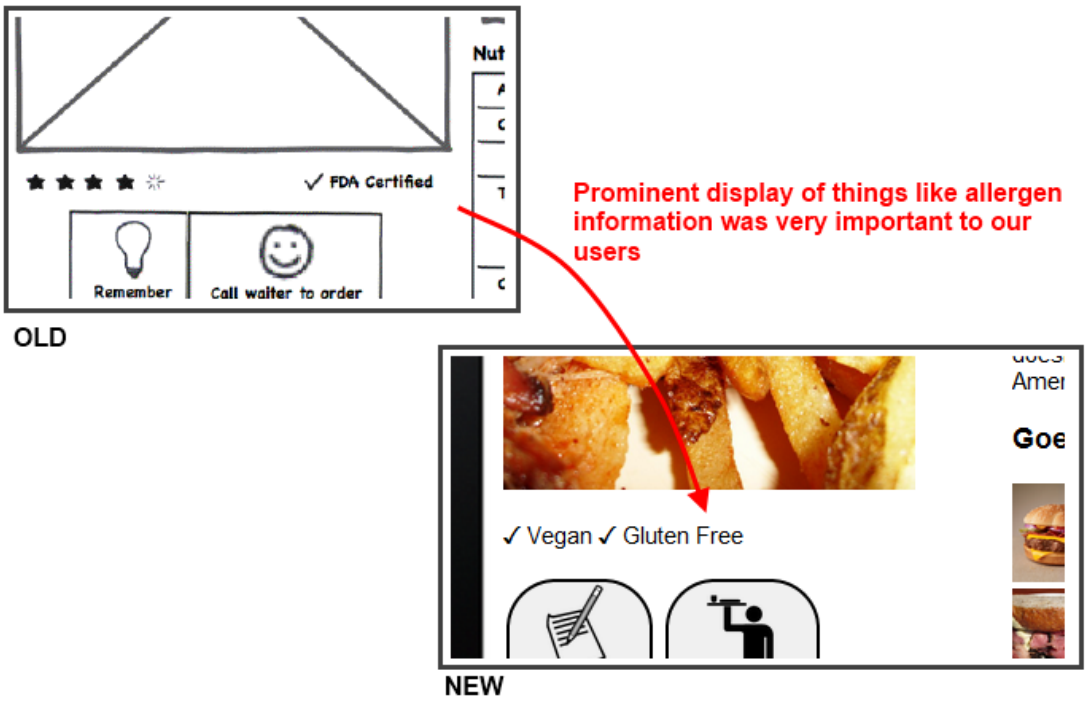
One interface revision we made was changing the icons for both the “Remembered” action as well as the “Request Service” action. Previously, they were a light bulb (representing more of the “idea” association) and a smiley face (trying to associate friendly service). We changed the Remembered List to an actual list with a pencil

writing it, which has a connotation with notes and lists. We changed the Service button icon to a person icon of a waiter, which has a more direct association.



Another interface revision we made was to get rid of the tip slider. In our paper prototype, we didn't find users liked it very much or had much success using it. Additionally, it forces you set minimum and maximum amounts to tip, which users also didn't like. We moved to simple plus and minus buttons, allowing an easier

controlled granularity and no upper limit to the tip amount.



From a recommendation from one of our participants, we added allergen information to the details pages for particular food items. It's becoming more frequently in-demand when eating out for diners to know about allergen information, like if an item is gluten-free. We've added this detail to our prototype.

Interactive Prototype Overview

Our team built an interactive prototype rather than a video. It is based on HTML, JavaScript, and YUI, a JavaScript UI library from Yahoo. A real implementation would probably be written in the native environment for the tablet running the software, but going with web technologies allowed us to build it faster while not committing to a single platform such as Android or iOS. Further, having a web-based version means our prototype can be seen and evaluated easier and by more people.

The prototype splits into two parts, the menu part, and the payment part.

Menu

Broadly, the menu is organized in five categories: Specials, Food, Popular, Drinks, and Dessert. The default page, i.e. the page you see when you first start up the program, is the Specials page. In a real version, running at a real restaurant, the

number and kinds of categories, along with the default category, would be configurable.

Specials

The specials page is meant to show today's special offers. This is the equivalent of the waiter coming to your table and recommending the soup of the day or the freshly caught salmon. In this version, we have three specials, and because of the web-based implementation, we have chosen to expose them via three buttons. On a touchscreen, one could imagine that you'd go from special to special by swiping left or right, but that's not feasible on the web.

Food, Drinks and Desserts

I'm grouping these three categories together since they all work the same way. In fact, in the prototype, only Food is implemented, but it's easy to imagine Drinks and Desserts working the same way.

The Food category shows a long, scrollable list of all items on the menu. Each item has a picture, a title, a short description, and a widget showing the average rating this item has been getting from other diners. Items are grouped into categories, the same way they are on a printed menu. When you click (or tap, in the real version) on an item, you go to the item's detail page. For the prototype, we implemented the detail pages for Le Big Mac, Royale With Cheese, and Fries.

On the detail page, you see much more information about the item. On the left, there is a larger picture of the item. On the right, you get the title and a longer description of it. You can imagine many kinds of information that could be shown here. The prototype shows nutritional information and a "Goes Well With" section that suggests pairings, but different restaurants could add different things here, such as a history or description of the origin of the food or drink, manners of preparation, tags like "Vegetarian" or "Vegan", etc. The detail page also has two other buttons, "Remember" and "Place Order". The "Remember" button puts that item onto the remembered list (see below). The "Place Order" button signals a waiter to take the order immediately.

Popular

The popular page is a placeholder that's not implemented in the prototype since it's not necessary to complete the tasks. However, in a real implementation it would contain features such as "Find the highest-rated food item" or "What people around you ordered", comments people left about certain items, etc. Not all of these have to be based on other patrons. There could be recommendations from the chef, a staff favorite, and so on.

Calling Waiter

Every page on the menu has a button in the top-right corner that calls a waiter immediately to answer questions, take an order, or be otherwise helpful. Once the waiter has been called, the button lights up to let the customer know, but the menu remains otherwise functional.

Remembered

Every page also has a button in the top right corner that shows the list of remembered items. Whenever there are remembered items, this button lights up. Each item is shown with image and description, and there is a button to "forget" the item, which removes it from the list. Customers don't have to follow this workflow, but the idea is that they pick out several things from the menu and put them onto the list of remembered items, and then review the list and call a waiter from there.

Payment

The payment program is a separate prototype, built in the same way as the Menu prototype, but smaller, since it has less functionality. In a real setting, the waiter would bring a tablet to the table, running the software, pre-loaded with the bill, but for the prototype it's just a separate HTML page. Once again, every one of the payment pages has a button at the top-right that calls a waiter to answer questions or settle disputes.

The first page is simply an itemized bill. Each item has a picture, a title, a price, and a ratings control where the customer can rate the item, from zero to five stars. This is how we get the ratings that are displayed on the menu. There are two buttons at the bottom, "Pay All" and "Split". When the customer taps "Pay All", they go to the same payment process that "Split" will eventually take them to, so we'll describe "Split" first.

Once the customer selects "Split", the UI changes a bit. Every item gets a check mark. Tapping an item selects it. At the bottom, there is a total that tallies up all selected items. The buttons change to "Pay Selected" and "Pay All". Both buttons move on to the Tips screen, but the totals are calculated based on the selected items or based on all items.

On the Tips screen, we show the subtotal, the tip, and the total, each one on one line. The line with the tip is interactive: It shows the percentage, and an up or down arrow to adjust the amount of tip. By default, it's set to 15%. At the bottom of the page, there are two buttons, "Cash" and "Credit Card". Tapping "Cash" calls a waiter to receive the cash, but this functionality is not implemented in the prototype.

Tapping "Credit Card" brings up a popup prompting the user to slide their card on the side of the tablet. We imagine the payment tablets to be equipped with card readers, so that paying can be completely self-service for the customers once they have the tablet, and there is no long wait to complete it.

Once the customers slide their card, the transaction is authorized and settled, and the UI switches to the last screen. On this screen, customers can leave their email address to get promotional offers or invitations to events. Once this screen is dismissed, the UI thanks the subject for their participation. A real implementation would switch back to the main payment screen showing all items that were ordered, with the paid items grayed out and not counting towards the total.

Summary and Discussion

Our group delivered an interactive prototype for "Restaurant 2.0", and interactive menu and payment system designed for use in gastropub style restaurants. We began with the idea that restaurant patrons often deal with inefficiencies in the dining process, waiting on servers, looking for recommendations, etc. We believed we could solve these problems and bring a better experience to diners. From that initial thought, we molded and shaped our idea through contextual inquiry, paper prototyping, and finally building an interactive prototype using JavaScript and HTML. At each step, we received critical feedback that guided us in a better direction.

From our contextual inquiry, we learned that customers didn't want a complete overhaul of the dining experience. They didn't want to place orders themselves, they liked being served by a real person. They liked being able to ask for recommendations from both the server as well as their friends. It became clear to us that we didn't want to replace the dining experience, just augment it. We wanted to bring more power, information, and choice to diners.

From our paper prototyping, we learned the importance of how design decisions influence and inform the user. When users found button icons hard to understand, they were unable to easily complete their tasks. Users also brought varying goals and perspectives to the evaluation, which in turn illuminated parts of our design in ways we hadn't conceived. Some users found the ability to remember items for later extremely reassuring, so that they wouldn't have to worry about forgetting something they liked. Others thought it was totally useless, they could just remember it in their mind and didn't use the feature at all. Neither user is wrong, but we just need to make sure neither feels like their needs aren't being met.

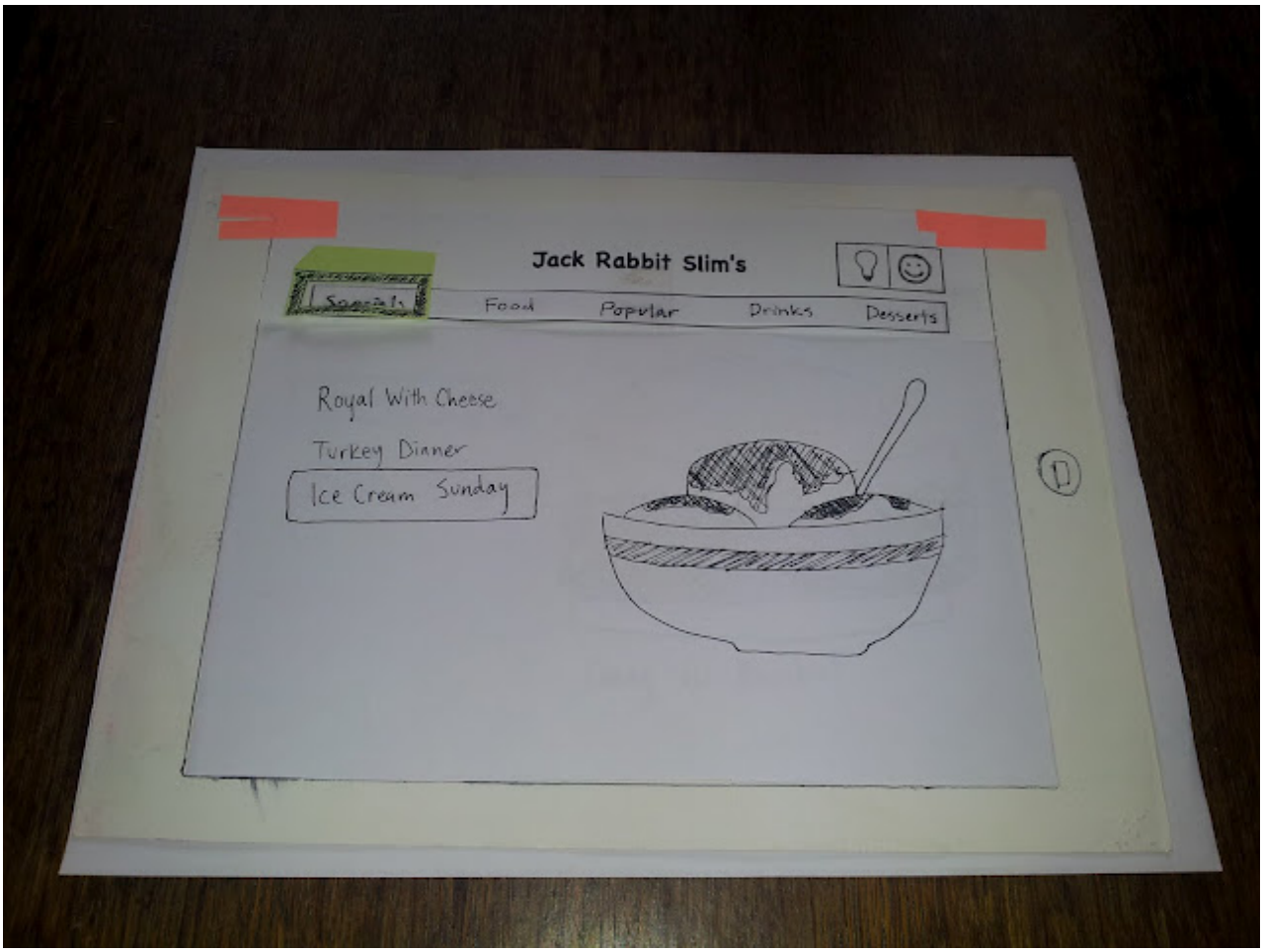
Actually going through the steps of building an interactive prototype also taught us some lessons. Building the prototype at this stage, we found ourselves the most confronted by the reality of implementing a design. While we still were only implementing a subset of functionality, we found that we had to cut corners to "make things work." Paper prototypes allow you to be flexible with things like text size and location, interactive prototypes have real size constraints. The paper prototype allows flexibility in terms of things maybe not working perfectly, and it allows you to do more hand-waving to convince a participant, but the interactive prototype looks like a possible implementation. We found that getting the smaller details working took much more time than anticipated.

The biggest thing our team did not uncover during the design process was what our system looked like from the perspective of restaurant owners. All our design effort (and scoping) was focused on how to present the interface to the dining customer. However, in a truly complete system like this, the interface to the restaurant staff could be just as important. If we were going to continue work for this project, further contextual inquiry would certainly need to be done with servers, cooks, and restaurant managers.

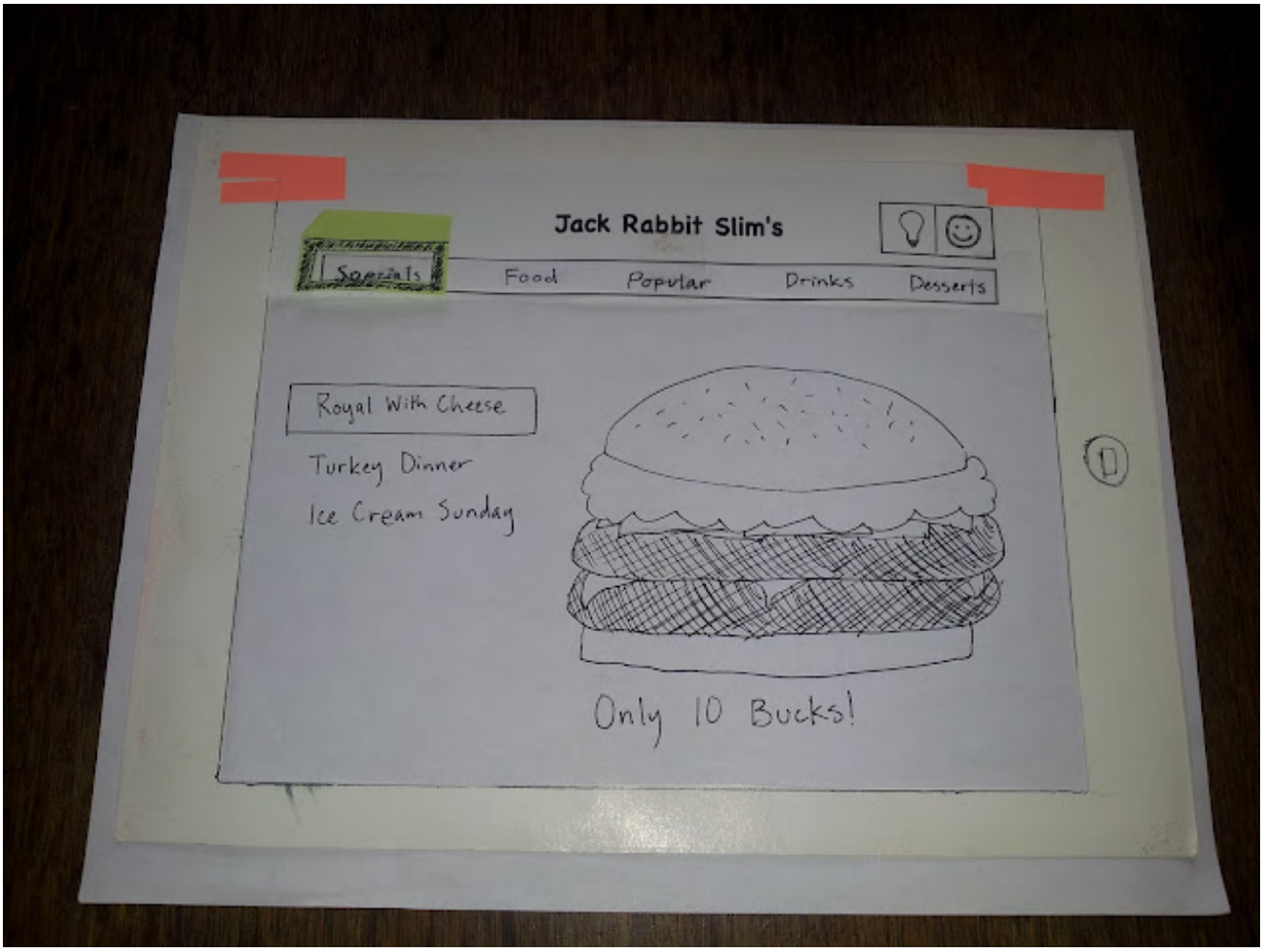
Appendix

Further Pictures of the Paper Prototype

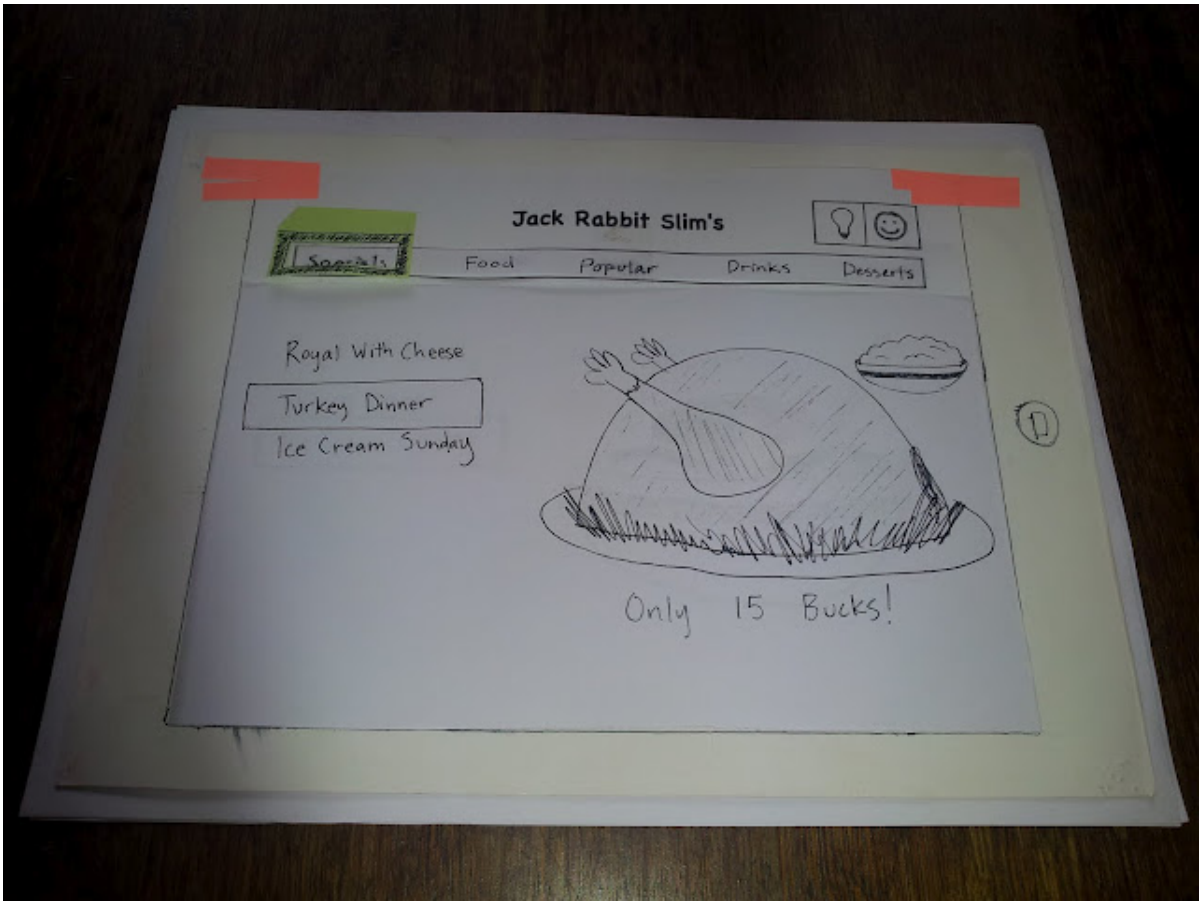
Specials pages:



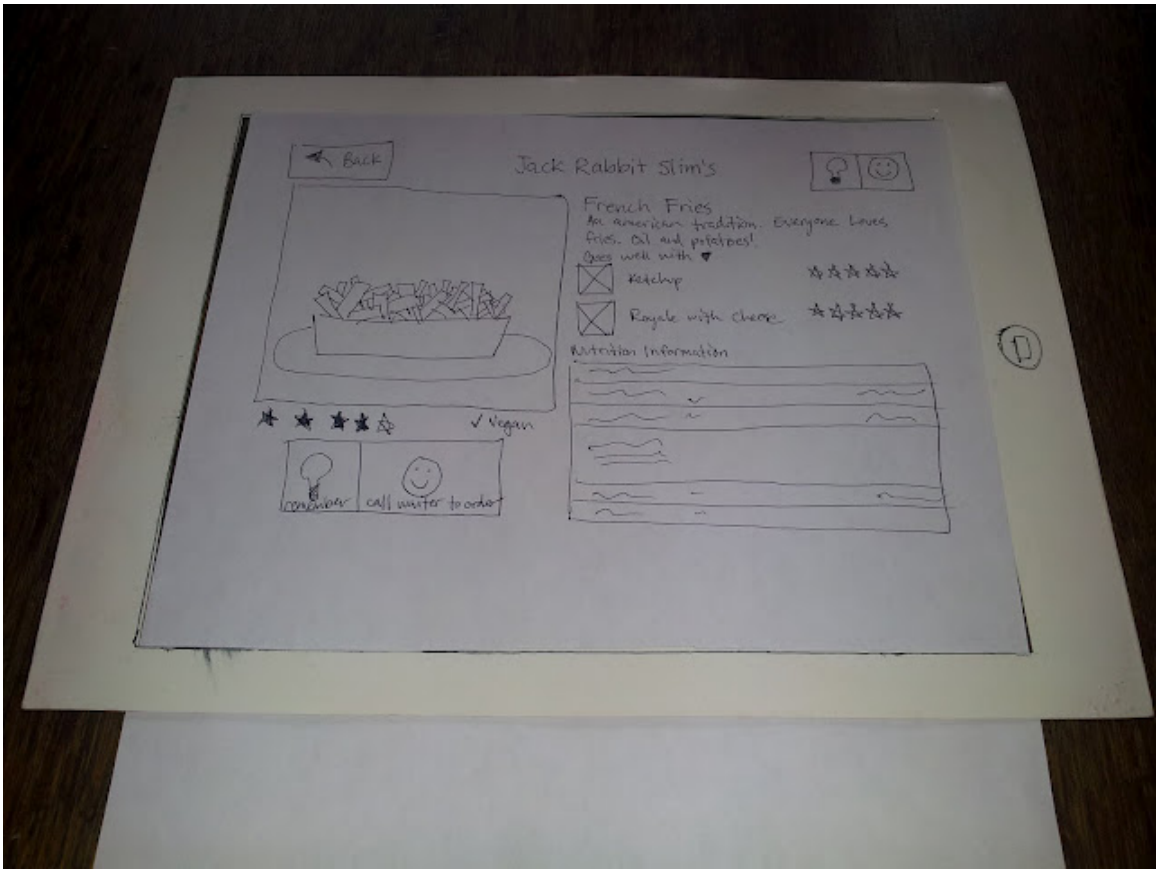
Specials page:



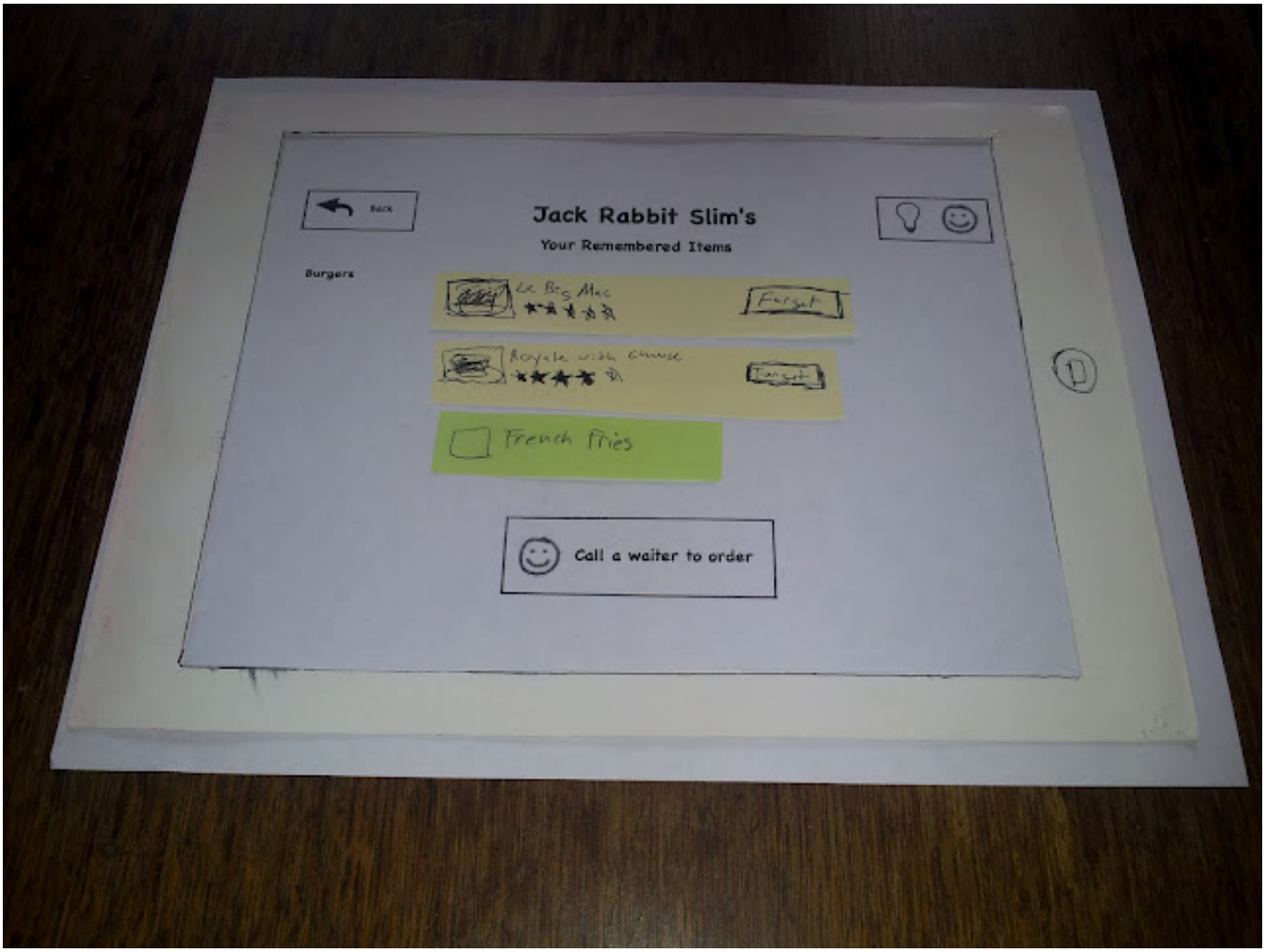
Specials page:



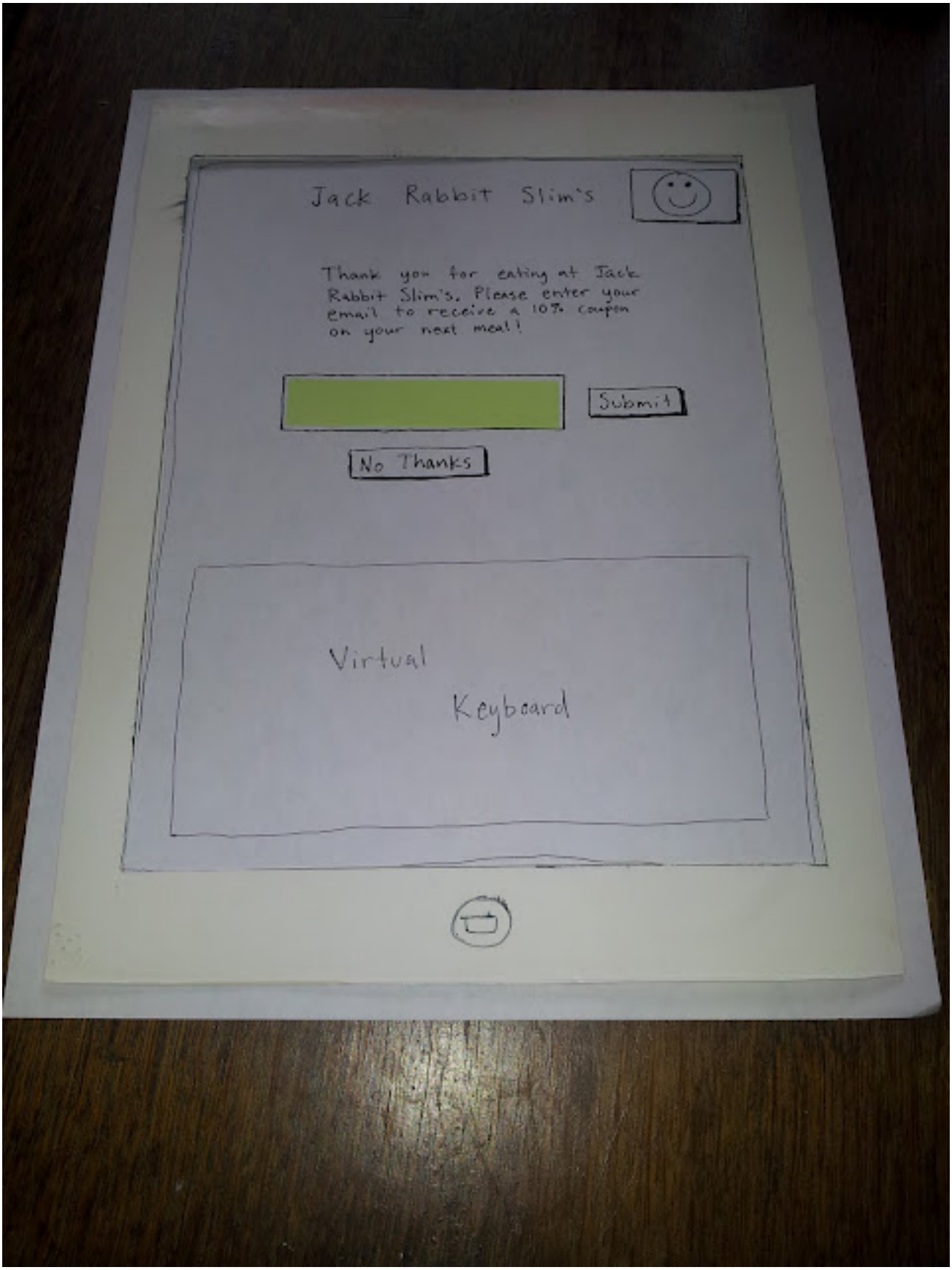
Details page:



Remembered List:



Promotions page:



Reports from testing with Individual Subjects

Subject A
Task 1

- presses the special button right away
- got the point of the specials tab right away
- presses the "call waiter" button, not the remember button
- presses the "place order" button

Feedback: "Call waiter to place an order" is redundant with the popup saying "What do you want from your waiter"?

Task 2

- rates first
- presses split
- clicks her items right away
- clicks "pay selected"
- doesn't notice tip screen, just hits "credit card"
- did give up her email address

Says she did notice the tip thing but ignored it. "How much are you really going to deviate from 15%? Give me a finer-grained interface.". Wasn't sure what sliding would really do.

Task 3

- a bit of confusion at first, hits "food"
- clicks on "Le Big Mac"
- remembers the calories in her head, not in the remember feature
- hits "back"
- hits "Royale with Cheese"
- hits "Remember"
- hits "back"
- gets the scrolling right away
- hits "Fries"
- wants to order the royale and the fries from a single page, because they show up in the "goes well together section"

Didn't use the remembered list. Didn't think to. Found it intuitive once explained.
A bit of confusion about the fact that you can't actually order through the system.
Additional Advice

- Add food allergens to the detail page for those who need it.
- Thinks she has to download the app, rather than being handed it.
- We rely on icons like the smiley face and the light bulb to represent concepts, but that only works if you're familiar with the app, which most of our customers aren't.

Subject B

Task 1

- taps button on special page
- taps "call waiter to order" button under image of the burger
- would assume the waiter would be coming and would still peruse the information on the detail page
- could bypass asking the waiter for a beer since the info is on the screen
- wanted price on the details page to make sure getting special price

Task 2

- taps split button
- taps to select items
- taps pay selected button
- swipes slider to the right to add tip to 20
- taps credit card button
- swipes the card
- at promo page but didn't rate stuff but now can't get back from promo page to rate items
- enters email into promo page
- could still rate items that are paid for already even if grayed out

Task 3

- clicks royale with cheese on specials menu
- clicks remember
- clicks back
- clicks food from main menu
- taps picture for le big mac from main menu
- taps remember under picture of le big mac detail page
- presses light bulb on top right of details page
- clicks to forget the big mac
- would click call waiter from remember page just to add fries but decided to check for fries
- wanted button from details page to main menu
- swipes main menu
- taps fries

- clicks remember on fries detail page
- clicks top right light bulb on details page to get to remembered items
- clicks call waiter to order

Overall

- would want prices on everything especially main menu
- wants remember from main menu if she doesn't want to see details
- light bulb and smiley face don't tell you what they do, had to guess
- question mark instead of smiley face, was guessing what the icons are
- depends how easy it is to use on whether likes it or not; if it's slow in a restaurant, why would you want this
- shortcut to skip the technology for old ladies
- good for making slow ordering process fast especially seattle on a Friday or Saturday night

Subject C
Task 1

- clicks Royale with Cheese on specials page
- clicks smiley face to order
- clicks place order
- asked what would happen if you click the burger from the specials page, show the details page
- details page, likes details hidden

Task 2

- presses split on ordering page
- clicks on items
- clicks to rate items
- clicks "pay selected"
- first instinct was to swipe credit card, but then realized to set the tip with the slider
- clicks credit card
- swipes credit card
- clicks "no thanks" on the promo
- Where do you sign the credit card statement?
- Did you like it? Sure, it was fine, liked being able to choose items to pay.
- suggested a way to split by an integer. They realized our system makes splitting easier but some people share so splitting evenly would be good.

Task 3

- clicks on burger special

- clicks remember under burger image
- clicks back
- hits food to get full menu
- clicks on le big mac
- presses remember on detail page under image
- clicks back
- scrolls and click on fries
- presses remember
- ”I would click on light bulb on top left.” Clicks on it.
- from the page with remembered items, goes to Le Big Mac detail, gets calories, looks at other burger for calories, click forget for burger with lower calories
- clicks “call waiter” to order
- What do you think of the list of remembered items? Like it to remember items when deciding, narrows down the menu.
- Feasibility overall? I think overall I would like it, would be easy for a restaurant that updates often to constantly change, would save paper, specials would be dynamic, system overall would be more dynamic.