Assignment 2: Voice-only, system-driven user interface
Scott Fabini
Thomas Van Klaveren

Phone Number to call: (360) 489-6055
Backup SIP Phone: sip:9996179281@sip.voxeo.net

Usability Test Script

1) Start the app
    a) Intro to app and prompt for order
2) Order sandwiches (user may have 0 or more sandwiches)
    a) Verify Correct items reported as sandwich is built
    b) Verify Correct sandwich reported before more options
        i) Prompted for review, edit, remove, continue
    c) Review the sandwich (current)
        i) Verify correct sandwich reported
        ii) Verify correct amount reported for current sandwich (assumes user knows pricing)
        iii) Prompted for review, edit, remove, continue
    d) Edit the sandwich
        i) Choose what to edit and what to replace
        ii) Verify Correct sandwich reported before more options
        iii) Verify Correct amount reported for current sandwich (assumes user knows pricing)
        iv) Prompted for review, edit, remove, continue
    e) Remove a Sandwich
        i) Verify Sandwich reported to be removed
        ii) Prompted to add to order or checkout (Done)
    f) Continue order
        i) Prompted to add to order or checkout
3) Order drinks (user may have 0 or more drinks)
    a) Verify correct drink reported as added
    b) Verify correct drink order reported before more options
        i) Confirm prompted for review, edit, continue
    c) Review drinks (all)
        i) Verify correct drinks reported
        ii) Verify correct amount for all drinks reported (assumes user knows pricing)
        iii) Prompted for review, edit, continue
    d) Edit drinks
        i) Choose what drink to edit/change and what that drink is changed to
        ii) Verify correct drink edit reported
        iii) Prompted for review, edit, continue

e) Continue
    i)   Prompted for add to order or check out (Done)
4) Check out
    a) Amount should be stated before disconnecting

Reference Table of Item Costs:

Item	Cost
Bread (Wheat, Rye, Sour Dough)	$2.00
Meat (Chicken, Meat Balls, Tuna Fish)	$3.00
Veggie (Lettuce, Tomato, Olives, Pickles)	$0.50
Drink (Coke, Pepsi, Seven-Up)	$1.00

Project Estimations vs. Actual

	Estimated Research	Estimated Design	Estimated Implementation	Estimated Testing	Actual Research	Actual Design	Actual Implementation	Actual Testing	Total
Scott	3	1	6	3	4	1	10	3	18
Thomas	5	2	8	2	4	2	10	2	18

User's Point of View:
The biggest hindrance to usability was the Aspect/Voxeo SIP Phone's sensitivity, resulting in an application with no barge-in capability. This means that the User must wait to listen for the whole message before speaking, which can become tedious once the User is familiar with the interface. Additionally, the grammar is quite limited and the user must speak the exact word. This is cumbersome, robotic, and impersonal. If there is any variation, the speech is not detected and the user is reprompted. If the user were to say "I'd like rye bread," the input is rejected even though the user believes he/she made a correct choice. It is also frustrating from a user standpoint to go several prompts rather than give all selections at once. For example, it would be nice to say "I'd like a meatball sandwich on rye with lettuce". The multi prompt approach eats up the user's valuable time.

Developer's Point of View:
The largest challenge of this project was learning the syntactical nuances of VoiceXML. Specifically, the use of <goto next="form">, <goto nextitem="#field"> , <clear namespace="#field">, and which XML elements/attributes could be children/parents of one another, was the most difficult part. After the lightbulb went off (after reading Voxeo's tutorial), the project started to gain steam.

Another difficulty was the declarative nature of VoiceXML. This led us to regretfully resort to <goto>s and <clear>s to drive the algorithm to jump between forms & fields, which felt like a bit of an outdated way to program. Our understanding is that we could supplement the XML with Javascript (1.1), but didn't want to further complicate the program by going in that direction as it seemed beyond the scope of this small application.

From the developers' standpoint, we opted to implement a multi-form setup. We thought this would be the easiest way for the user to add sandwiches and drinks to the order. We tried to remain cognizant of the user's experience when implementing the prompts. We tried to give enough information in the initial prompts and "tapered" off as the order progressed. For example, the user is asked to select veggies with a selection listed in the initial prompt. When adding more veggies, we simply ask "any other veggies" assuming the user will remember the list. The reprompt gives a reminder of the veggies. Even though this is a multi-form application, we tried to maintain consistency between the forms so that the user would not be easily confused jumping between forms. From the user's point of view, it should feel seemless.

Finally, the Aspect/Voxeo SIP phone's sensitivity to noise was quite frustrating from the testing side. This meant we had to add the bargein="false" to every prompt, which reduced the usability from a user's perspective.