Stand Alone Unit Shopping Assistant Gadget Extension v1.0 and Evaluation Plan

Assignment G3

Prepared by Team Discovery Channel:

Phyliss Lee (Phyliss.lee@gmail.com),

Steven Khuu (g7khuust@gmail.com),

Justin Chow (Just.chow@utoronto.ca),

Tong Zou (gofishus@hotmail.com),

Nikola Banovic (nikola.banovic@gmail.com)

Tutor: David Dearman

The Design of Interactive Computational Media - CSC318 - Spring 2009

March 10, 2009

Contents

Project Participation Summary

Project Description: [Nikola 0.5 hours]

Requirements Summary: [Nikola 1 hour]

Design Space & Rationale: [Stephen 2 hours], [Nikola 2 hours]

Prototypes:

- Storyboard Prototype: [Nikola 9 hours] , [Tong 1 hour], [Justin 1 hour]

- Functional/Physical Prototype: [Phyliss 9 hours], [Tong 1 hour], [Justin 1 hour]

- Sketching Prototype: [Phyliss 1 hour], [Stephen 4 hours] , [Tong 1 hour], [Justin 1 hour]

Evaluation Plan: [Tong 4.5 hours], [Justin 4.5 hours], [Nikola 0.5 hours]

Summary: [Tong 2 hours], [Justin 2 hours], [Nikola 0.25 hours]

Other:

- Reviewing G2's and I3a's

[Tong 4 hours], [Justin 4 hours], [Nikola 4 hours], [Phyliss 4 hours], [Stephen 4 hours]

- Editing

[Tong 1 hour], [Justin 1 hour], [Stephen 6 hours], [Phyliss 3 hours]

Total:

Tong 14.5 hours

Justin 14.5 hours

Stephen 16 hours`

Nikola 17.25 hours

Phyliss 17 hours

Project Description

After revising the G2 and I3a's this report summarizes and evaluates the key concepts that we felt were relevant to our envisioned design. Our goal is to utilize these ideas and refine our design of a shopping assistant application. According to the studies and analysis conducted in the I2's and G2's there was a large interest in the process shopping for clothing in comparison to clothing organization. The application will help the user find up-to-date sale event information as well as assist in locating stores based on user preferences and needs. This will eliminate the need for a user to perform the tedious task of comparing prices for similar articles of clothing at different stores. Our intended user group targets young adults, primarily University students. With student loans, hectic work schedules, and limited free time, we felt that this application would have the most significant impact to this user group. By creating this application, we are hoping to make the task of clothes shopping more time efficient, cost efficient, and in general, a more enjoyable experience.

Requirements Summary

Based on requirements listed in select G2 and I3a assignments the proposed design should address the key requirements outlined below:

- **Provide locations and directions to stores of interest**
- **Provide notifications about events, such as sales or limited time offers or availability of items of special interest**
- **Learn user preferences by gathering user shopping habits and preferences**
- **Enable user to specify special items of interest and use the information during shopping sessions**
- **Ensure portability of solution so that it can be accessible to the user at all times**
- **Ensure compatibility with other platforms and electronic devices**
- **Ensure privacy when necessary and filtering of messages based on message priority and user preferences**
- **Allow uninterrupted shopping experience**
- **Provide an affordable solution for individuals with lower income**
- **The solution should minimize the amount of effort a user needs to perform in order to start using the application**
- **Solution should interact with environment (nearby stores, kiosk, and information booths) to collect information relevant to users shopping session**
- **The solution should be able to locate the user in real-time**
- **The solution should provide means of connectivity with the environment**

Design Space & Rationale

After reading over the I3a's, we have selected four submissions that best represent our prototype design.

Nikola Banovic

Nikola's envisioned design involved two main components. The first was a mobile application that could be installed via the internet or through Bluetooth. The second component was a server application installed at every mall utility such as an information kiosk. His motivation for creating two components instead of one was to attempt to enhance the shopping experience of the user by increasing the interactivity between the user and the shopping environment. These interactions include receiving relevant information about nearby stores based on the user's brands of interest. Since the interactions were only intended to enhance the shopping experience of the user, these interactions only occurred when a user was in a shopping center. An example of the interactions he provided talked about a person looking for business suits for an upcoming interview:

"…when Amanda goes to the Eaton Center she would start the application on her iPhone and select her primary interests (if any) for this shopping session, say shopping for clothes for her upcoming interview. The application would in turn gather information about nearby stores that address Amanda's primary interests and would display a map of the environment and mark a path on the map for Amanda to follow to accomplish her task and goals."

For additional convenience, Nikola added some secondary uses such a finding washroom and favourite coffee shops within the shopping center. One interesting feature to note was the application's ability to 'learn' from the user. After every purchase a user makes, the information about each article of clothing purchased such as brand, clothing type, or size are recorded into the application. In turn, with this information, the application would help to find a user's preferred store and brand the next time a user looks for the same type of clothing. In one example, if the user purchased a pair of jeans, then in future cases when the user wants to purchase another pair, the system would be able to make smart suggestions on stores that the user should visit. This would include receiving information from the stores such as quantities of the specified clothing items.

Features We Kept from Nikola's Design

One notable feature that we kept from this I3a was the 'learnability' of the application. Although a program cannot actually predict the thoughts of a user, it can make an educated guess. We believe that this feature serves as a motivational tool for the interactivity between the user and the application. This extends the concept of feedback discussed in the lecture. This feedback can be used to eliminate the need for users to search for the same clothing items in the same tedious and repetetive manner. If the application makes an educated guess about a user's desired clothing item of interest, then this can reduce the amount of work a user has to perform to complete a task.

Features We Did Not Keep from Nikola's Design

We decided not to include the additional locating features such as locating nearby washrooms and coffee shops. Although these features may be considered convenient for other applications, we feel that this feature steers away from our interest in making a shopping helper. We also decided not to include the second component of Nikola's design, the server side application from the mall's kiosks. The problem we had with this was that this would have a dependency on the compliance of other malls. This would also mean that malls that do not have this application would make the shopping helper useless. We want to ensure that the application is fully functional at any shopping facility, not just at the ones with the server-side application.

James Ma

In James' design, he emphasized on the importance of portability and compatibility of the application between different platforms on different mobile phones. These platforms include Nokia's Symbian OS or Google's open source OS, Android. He believed that compatibility with open source software would make development of the application a lot simpler and as well as increasing the amount of support documentation for the application. His application has four main modes of operation: Me, Find It!, What to wear?, and Settings. In “Me”, the user can create a shopping list, bookmark their favourite stores, and rate the clothing in their own wardrobe in terms of likes and dislikes. In “Find It”, the user can manually search for stores in a similar manner to Google Maps. From the “Me” function, there are a list of bookmarks of the users’ marked locations. In addition, there are also stars above locations that contain news about recent sales or events. In “What to Wear”, the application will look over the clothing database of the store that the user is currently in then, based on the shopping list from “Me”, make suggestions for clothing the user could wear. In the case that the user is not inside a store, this function would suggest clothing from Amazon. In “Settings”, the user can switch to another user’s profile and set preferences. As a result, this will change the way in which “Find it!” and “What to Wear” will behave.

Features We Kept from James’ Design

Compatibility with other mobile devices was a design feature we decided to keep from James’ design. By including these two concepts, this will help make our application as accessible as possible and an application that could be available to anyone. In James’ “Find It” the star marker above stores of importance is a notable feature that we want to consider including. From the research conducted in the G2’s we noticed that the price of an item of clothing greatly affected a person’s decision on purchasing the item. Hence, by including sales notifications, this will serve as a motivational tool to encourage a user to shop more often.

Features We Did Not Keep from James’ Design

Although the “What to Wear” feature would be in parallel with the type of feedback that we are looking for (as described in the feature we kept from Nikola’s design), it is a dependent system. James’ form of feedback only works if the user inputs a list of items that they wanted to buy ahead of time. This means that in order for the system to actively offer smart suggestions would require the user to enter in a very specific shopping list every time. In James' design for "Settings", a user can switch between profiles at any given time. This raises a lot of complicated issues such as privacy between two different users and security measures. Although James did not explicitly deal with the privacy issue, it is still something that has to be taken into consideration. For our design, we want to keep it simple by only allowing one user per copy of the application.

Stephen Khuu

Stephen's design describes a mobile device that contains portable software. The software is compliant with other smart phones such as the Blackberry or the iPhone. The device itself is a touch screen device and the menu displayed on the device consists of large buttons each of which contain suitable icons that represent each menu option's functionality (e.g. "Locate Nearby Clothing Store" is represented with an icon that has a magnifying glass over a clothing store). He described two main tasks that his design would accomplish; locating nearby clothing stores and shopping for user preferred clothing. In the "locating nearby clothing stores" menu option, the application displays a map, highlighting nearby stores suggested by the application based on the style and price preferences set by the user. By clicking on highlighted store, the user could find out more information about the place such as sales and new promotions. In "shopping for user preferred clothing", the application offers suggestions for clothing articles in the form of selectable thumbnails. This list of suggestions is populated based on the purchase history of similar articles of clothing by brands and types. For example, if the user makes a lot of purchases from American Eagle, then this menu option will produce suggestions consisting mainly of items from American Eagle. The application allows the user to try on a simulated fitting from the suggested clothing articles. The application uses a picture of the user and places the clothing article over the user's body in the picture.

Features We Kept from Stephen's Design

Similarly to James' design, Stephen also mentions the portability of the application. Although Stephen's design consists of a portable device and the associative software, the application can be transferred onto another device. Again, portability between devices is definitely a feature we want to have for our designed prototype. In the "locating nearby clothing stores" menu option, Stephen brings up an interesting feature where a user can learn about recent events at stores by clicking on any store on the map. On top of James' feature which notifies users of new events, Stephen's feature takes it one step further by showing the user what those events actually are. This is useful for our prototype because we want to enhance the users' shopping experiences by motivating them to shop at stores that would fit their budget.

Features We Did Not Keep from Stephen's Design

Although Stephen has proposed a solution involving portable software, the device mentioned in his design seemed rather unnecessary. The software is operational independently from the device. The device itself can become a useless peripheral to the user if the user can simply install the software onto another device. One of the main considerations we took into account was the idea of wasted consumption. Users are expected to own this device, but not necessarily required to use it. What they are using is the software inside the device and the device just serves as a peripheral that communicates with the software. For virtualized fitting, Stephen argued that this could be a unique feature that online shopping does not offer. However, the key problem with this feature is the fact that it is a virtual simulation. Although a user might get some idea of what the clothing item looks when they wear it, there is nothing that can tell the user how the garment actually fits (e.g. whether the item is too tight or too loose). For the "shopping for user preferred clothing" concept, producing suggestions based on purchase history can be useful in the long term. However, this means that for the first several uses, this feature would be useless and would not be able to produce any educated result. In addition, because it is dependent on purchase history, the user must invest a lot of their money into buying clothing first before the application starts figuring out the preferences of the user in terms of types of clothing that a user prefers.

Tong Zou

With the growth in open-source software developing across the web, Tong's solution was a web based application that could be developed by open source tools such as Ruby-on-Rails or Ajax. Since this is a web based application, Tong's solution requires a user to register for the service and is prompted for a username and password in order to use the service. In the initial registration page, the user specifies their preferences on clothing in terms of style, price range, size, and brand name. When the user logs into his/her account, the application will display the latest suggestive trends based on the preferences set by the user at the time of registration. The application also has a menu which allows the user to configure their preferences, change languages, adjust notifications, find assistance with the application in the help options, and configure GPS options. Since this application is web-based, the application can also be accessed from smart phones such as the iPhone or the Blackberry. If the operating device has GPS, then the web-application offers an addition feature which locates stores close to the user according the preferences of the user. Also, the application would be able to give the user directions to the store. In the case that the current device operating the web application does not have GPS, the user would have to manually enter in the address of the store that he/she is looking for. In either case, the web-application allows users to make online purchases as well as arrange clothing items to be set aside so that the user could try on the articles of clothing when the users arrive at the store.

Features We Kept from Tong's Design

In review with the I3a's as described above, it is clear to see that the concept of portability through different devices on different platform is definitely a feature that must be included in our design. Although Tong's solution to portability is to make a web-based application, it does still demonstrate that the application can be used on any smart phone device. The setup of user specified preferences is something that we decided would be helpful to our system. As we mentioned above, we want our application we be able to make educated guesses and will informed suggestions for the user based on information that the user provides. One flaw that we noticed from other designs was that the accuracy and intelligence of the application would increase over time, not necessarily guaranteeing the initial intelligence of the system. With some initial data, the application will at least have some data to work with in order to make smart suggestions to provide to the user right from the first time the user uses this application. The utilization of GPS in the Tong's design will definitely be useful in our application. Although the automated navigation is only available to mobile devices with GPS, Tong's design does offer an alternative to mobile systems that do not come with GPS. Note that the locating menu option is available for both platform with GPS and platforms without GPS. This flexibility is another aspect of compatibility which we want to include in our design.

Features We Did Not Keep from Tong's Design

Tong's web application extends the convenience of the shopping experience of the user allowing the user to reserve clothing to try on at a store. It can be incredibly inconvenient for the store to hold articles clothing for a large number of users. This feature also does not consider the amount of time a store should hold onto articles of clothing for users. In one case, a user could plan a whole day of fittings at different stores and require certain stores to hold onto the user's items for the entire duration of the day. In another scenario, another user could reserve a set of clothes to try on at a store but forget all about it. It becomes very challenging for a store to differentiate between these two possible scenarios. Shopping online with this application also proceeds in difficult complications. If a user purchases an item online through the web-application, does the store expect the user to pick up the item at the store's location? If this is not the case, then what happens if the user purchases an item online while inside the store? The difficulty occurs when the store has to determine whether a user is picking up a purchased item or expecting a delivery. Even if a user could specify that he/she wanted to pick up his/her purchases, we run into the same complications with reserving clothing as mentioned above. A store could be notified that a user was picking up his/her pre-paid purchases, but there is no notion about how long the store has to have them on hold.

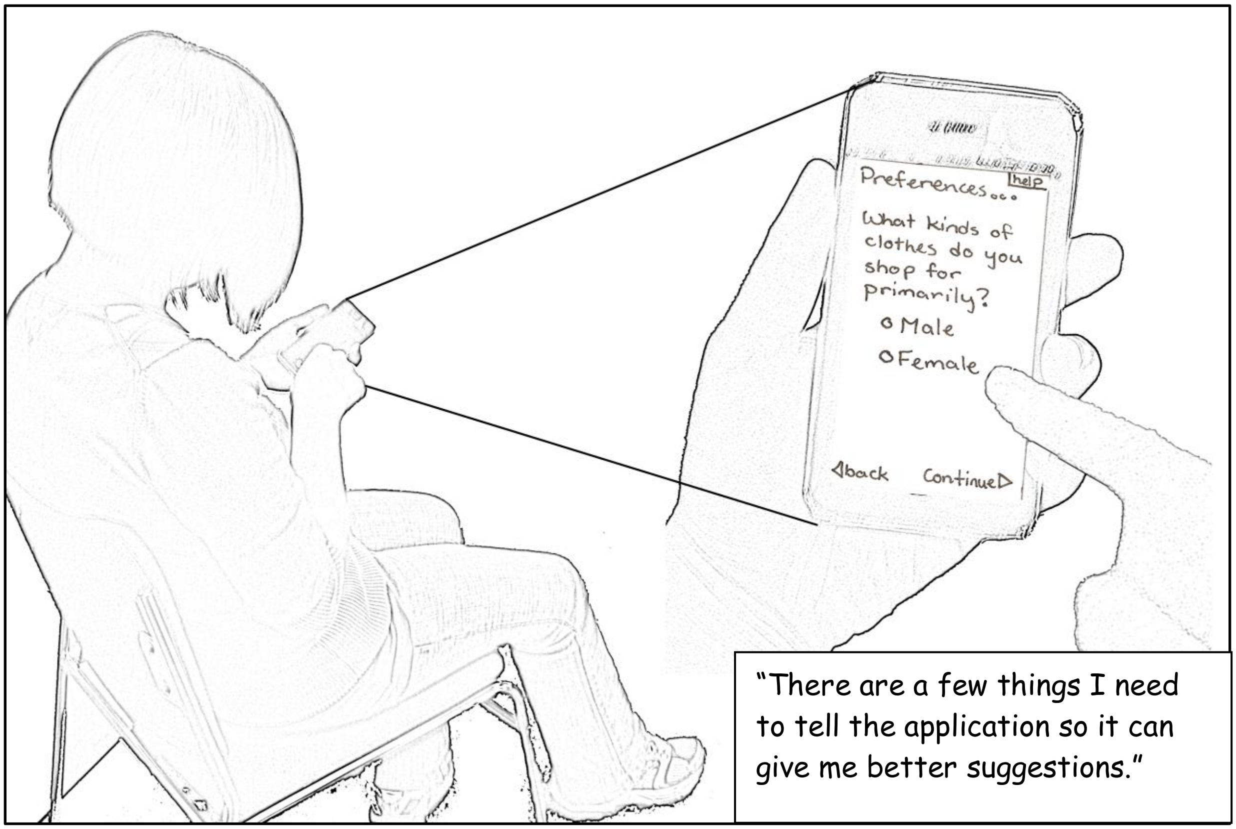
Prototypes

Storyboard Prototype

The two storyboards below illustrate the motivation for using the application and context in which it is used. Considering that this is a mobile application it can be used at any time and virtually anywhere where there is Internet connectivity. However the application is primarily intended to be used while shopping for clothes, or to plan a trip to a shopping center. For example, the user might start the application at her home before she goes shopping to find the closest mall; or while on her way to a shopping mall to make sure that the stores in the shopping center carry clothes she's interested in. Once at the shopping center, the user can start the application and get information about the nearby stores. Additionally, the user can review her preferences at any time before, during, or after she's done shopping for clothes.

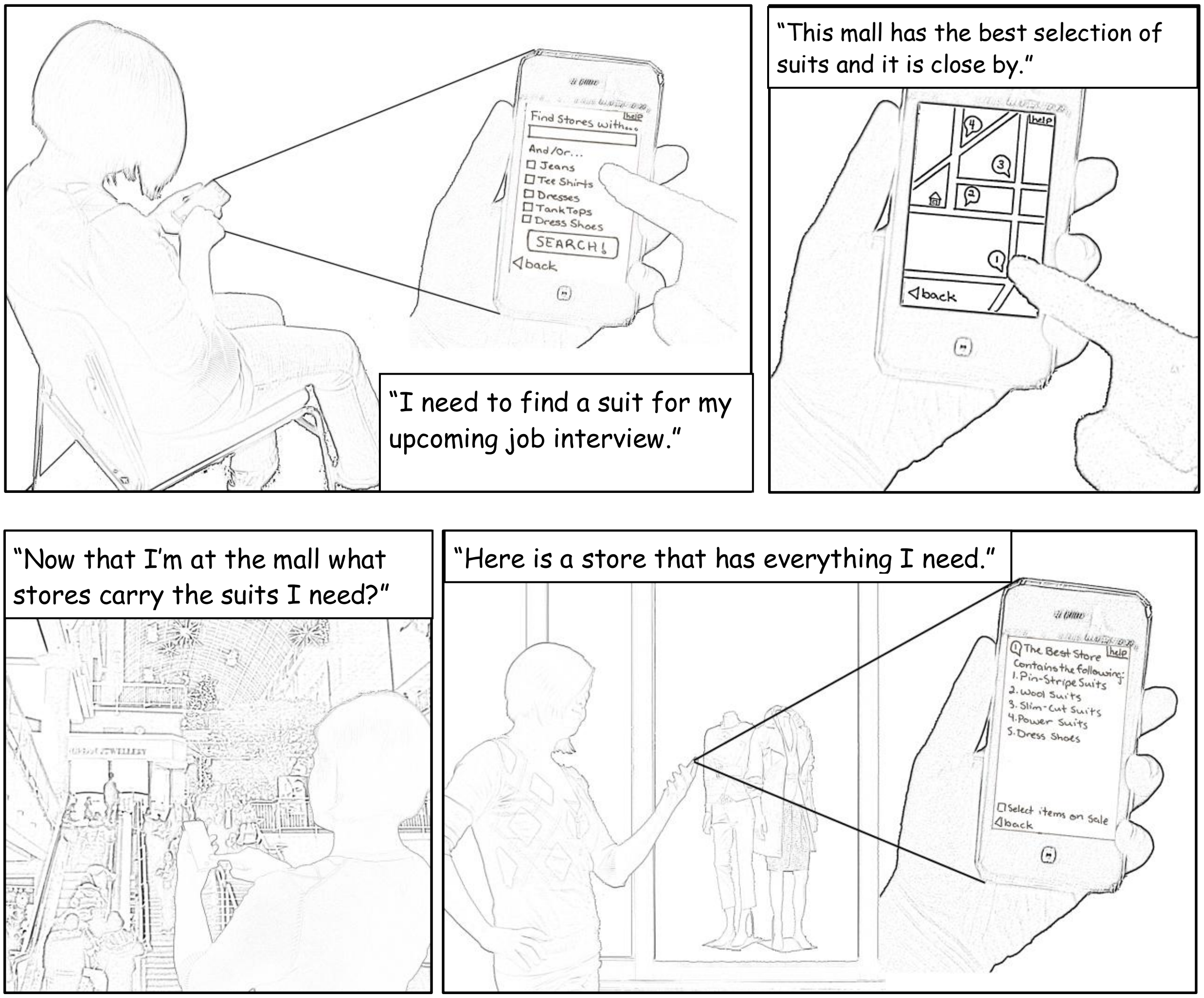
Scenario 1: Setting Preferences

While the application will learn user preferences from user's interaction with the application and information provided by the stores where the user shops, it provides functionality for the user to review and adjust any preferences. Additionally, before the user starts using the application for the first time the application will ask her to answer a few questions so that it can provide relevant information about clothing stores that cater to the user (e.g. user gender, birth date, etc.).


Preferences...
help
What kinds of clothes do you shop for primarily?
○ Male
○ Female
◁back
Continue▷
"There are a few things I need to tell the application so it can give me better suggestions."

Scenario 2: Searching for Clothes

When a user wants to go shopping for clothes she can search for a particular store, item, or brand using the search feature. The feature will perform a keyword-based search and will return the results in form of shopping center locations together with details about the shopping centers and why they were chosen. Additionally, the user can check frequently searched keywords to be included in the search.



Storyboard Prototype Rationale

The storyboard motivates reasons for why a user would want to use this application. Since our design consists of multiple screens, a storyboard is necessary to show the different screens a user would navigate through the application to accomplish a task. Of course, this visualization cannot be performed in just a single drawing. Hence, the frames of a storyboard best demonstrate the progress of the user in terms of completing a task.

Functional/Physical Prototype

The physical prototype is constructed to help simulate what the application would look and feel like on a mobile device, in this case, the iPhone. It is the same size as the iPhone, the only thing it lacks is the weight. However, the main focus of this was to get a good idea of the kinds of user interface challenges we would be facing on such a small screen. The user interfaces for almost all possible screens were also created to make the prototype as functional as possible.

Scenario: Buying Business Clothing

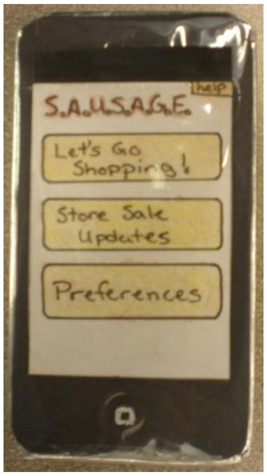


Figure 1

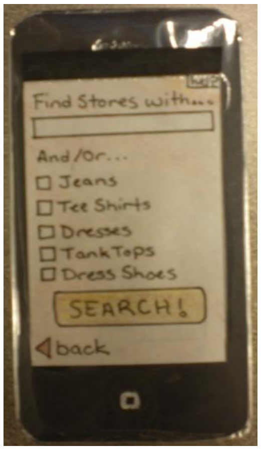


Figure 2

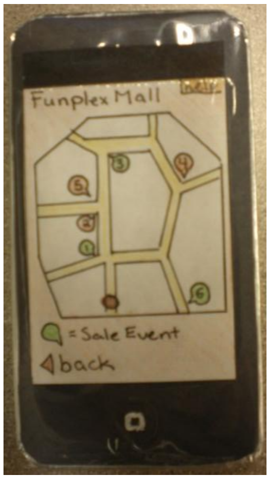


Figure 3

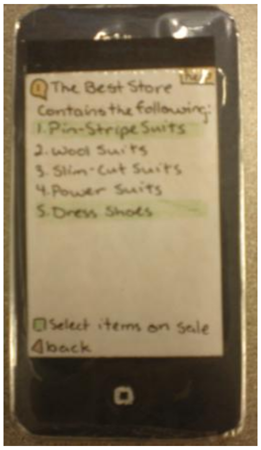


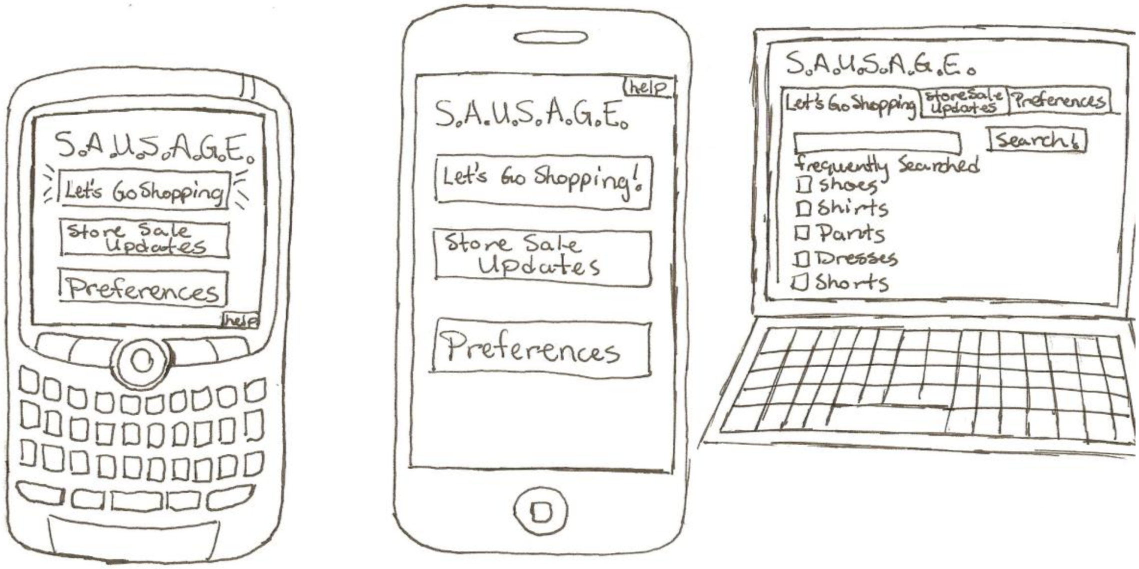
Figure 4

Amanda is at home and has decided that she needs a new suit, but is unsure of where to find one. She goes to her iPhone and opens the Stand Alone Unit Shopping Assistant Gadget Extender (Figure 1). Upon opening, she is taken to the main menu where she taps on "Let's go Shopping!" She smiles to herself about the button label while being redirected to the search page. Looking at her list of most frequently searched items, she realizes that a suit is not on there (Figure 2). She opens her keyboard application on the phone and types into the search box "suit". She also realizes that she needs a pair of dress shoes to go with her suit and that it is conveniently in her list of frequently searched for clothes. She checks off "Dress Shoes" and hits the search button. Next she is taken to a map of the area with certain locations marked off where she could go to find what she needs (Figure 3). She taps on the "1" bubble to see why the application chose that to be the most relevant (Figure 4). She is taken to a more detailed description of what is at that location and learns that that location is Funplex Mall and contains many stores that meet her current needs. She decided that she would like to go there so continues to get the directions. Upon arrival, she taps on “We’re there!” to continue to mall map view. From this map, she can see where each relevant store is located as well as which ones have sale events going on at the moment. She selects on the “1” bubble again to see why this is the most relevant store. From here she can see what kinds of clothes the store has available as well as which ones are on sale. She goes to the store and finds everything she needed. She purchases a suit and shoes and heads home happy about how efficiently that shopping trip went.

Functional/Physical Prototype Rationale

We decided to create this prototype over another because it was important to find the limitations of a mobile application in terms of usability. We chose to focus on the iPhone, mainly for its touch screen capabilities. The touch screen greatly improves usability compared to other smart phones such as the Blackberry. It also has a larger screen than other smart phones, which allows us to create better interface mock-ups. Since mobile devices often have small screens, it was important to prototype this because it greatly affects the user interface design. This kind of paper prototyping also allows us the low fidelity we need in the early design stages while still being able to convey functionality.

Sketching Prototype



The sketch shown above was made to show how our application would look like on different devices. In this case, the sketch shows the application on a Blackberry, an iPhone, and a laptop. The purpose of this sketch was to show to variation in size of the interface on different devices. For example, the screen size on a Blackberry is sufficiently smaller than that of a laptop. This sketch shows that the application can be supported on different platforms but does not compromise in terms of how much a user can see. Notice that on all three platforms, we can see the full menu. Since the laptop has the largest screen size of the three types shown in the sketch, the software uses a secondary visual layout instead. Instead of having three large buttons for the main menu, the application represents each menu option as a tab in a window, similarly to Google Chrome. On the mobile devices, the application requires the user to select one of the three menu buttons before the application goes to the next respective screen. On the laptop, when a user clicks on one of the three menu tabs, the controls for the menu options are displayed within their respective tabs.

Scenario: Looking for Sales Updates from Three Perspectives

i) After a tiring week of long assignments, Amanda finally gets some free time to herself. She takes out her BlackBerry from her pocket and selects the S.A.U.S.A.G.E. application. Feeling the need to find deals, she uses her Blackberry's trackball to navigate to the "Store Sale Updates".

The “Store Sale Updates” button is highlighted, notifying Amanda that the BlackBerry’s trackball is currently hovering over this menu option. Amanda selects the menu option and she sees a list of RSS feeds regarding sales events about her preferred stores (as specified by her settings in the preferences menu option). Using the trackball, she scrolls down the list of RSS feeds and finds a headline about a clearance sale at one of her favourite stores. She selects the headline and the application displays the flyer of the clearance sale. Excited about the sale, she grabs her purse and heads out the front door.

ii) Sam wakes up to a phone call from his friend who asked him about plans for shopping. The end of the summer is the time when businesses hold seasonal clearance sales. Sam reaches for his iPhone and loads the S.A.U.S.A.G.E application. He touches the “Store Sale Updates” option and sees a list of RSS feeds regarding sales events at his favourite stores. With his finger on the scrollbar, Sam scrolls down the list, dragging his finger along the scrollbar. He notices that one of their favourite stores is having a 2-for-1 sale. With his finger, Sam selects the 2-for-1 headline and the application displays the associative flyer. Sam reads on the flyer that the 2-for-1 sale in on jeans. Sam tells his friend the details about the sale and arranges to meet with him at the store.

iii) Jimmy is getting prepared for the new school year and has a tight budget. He turns on his computer and loads the S.A.U.S.A.G.E application. Jimmy uses his mouse and navigates the cursor to the “Store Sale Updates” tab. He clicks on the tab and the application loads list of RSS feeds inside the tab. With his mouse wheel, Jimmy scrolls down the list and notices a 2-for-1 sale on t-shirts at a store that he visits often. He clicks on the headline and the application displays the flyer of the event. Jimmy notices that the sale is still in effect. He quickly rushes out the door and heads to the sale.

Sketching Prototype Rationale

The sketch allows us to demonstrate the visual adaptation the application takes on different mediums. We chose to display our application on the BlackBerry, iPhone, and a laptop because each of these devices uses a different peripheral to navigate (i.e, the BlackBerry uses a keyboard and trackball, the iPhone uses a touch screen, the laptop has a keyboard with a touchpad or a mouse). The reason we chose to demonstrate this concept on a sketch is because the sketch is capable of providing a visual aid to show the application on all three devices. We are not trying to show the usage of the application on each of these devices. Rather, this sketch shows the visual-adaptation capabilities of the application. Hence, by showing the application on different devices ranging in screen sizes and peripherals, this demonstrates the extendable compatibility of our application on different platforms.

Evaluation Plan

The design requirements cover the basis of our usability specifications. Our design goals and prototype features focusing on 4 main features: portability, locator, storing user preferences, and making suggestions. We have chosen to focus on evaluating two of the main features in our list, which are locating and storing user preferences. Our evaluation plan is centered on these two aspects since portability is easy to assess and will rely on the device the software is installed on.

The evaluation techniques we chose to use are cognitive walkthrough, think aloud, and heuristic evaluation. We believe that these three techniques will help us evaluate the behaviour of users when using our application. This will allow us to review how intuitive the design of our application is. As such, the feelings of users are best expressed using the think aloud approach, as opposed to a questionnaire or interview. A cognitive walkthrough provides feedback based upon knowledge of HCI experts and is much more valuable to us than a simple usability experiment. Finally, a heuristic evaluation is in depth and comprehensive, which is necessary as our other two approaches only cover evaluations on the user end.

Think Aloud

For this technique, users from our primary stakeholder group (18-29) will be asked to perform a set of tasks and to describe his/her thoughts after each action is performed. Each instruction references the respective storyboard scenario and makes use of the UI frames on the functional prototype whose form factor is shown in the sketch prototype. Users will be asked to perform the following tasks:

- Users will be given the task of setting up user preferences (brand new users) and searching for a particular item in the area.
- Users will be asked to locate a store after inputting their location.

Setting up preferences (first time user):

1. Software boots up.
2. Screen shows introduction of software.
3. User first sets his/her gender and clicks continue.
4. User sets his/her birth date using drop down menus and clicks continue
5. User types the stores he/she frequently shops at to add it to the software, but it's optional and the user can choose to click skip instead.
6. A thank you screen shows up and the user clicks continue.

Finding an article of clothing (suit):

1. Software boots up
2. (Assume user set preferences already) User clicks on lets go shopping.
3. (User doesn't have suit in the suggestions) User types in suit and checks shoes (dress shoes are already listed).
4. User clicks on search button with items already checked.
5. A map comes up with numbers (bubbles). User clicks on 1 (Funplex mall).
6. User clicks on get directions which gives directions a la Google Maps.

7. Application navigates user to the mall, the user clicks “We’re there” upon arrival.
8. Map switches to the mall view with numbers. User clicks on 1 and is given articles of clothing the store carries.

We will write down the user’s thoughts as they perform these tasks. After the evaluation is complete, we will have the two main features of our design tested by end-users: the locator which helps users navigate to the store they want, and the storing of the user preferences and how the device learns from the user’s shopping habits. This feedback allows us to assess how our prototype functions from the viewpoint of the average user.

Cognitive Walkthrough

A cognitive walkthrough will be performed because it utilizes the experience and knowledge of HCI experts to evaluate design for the ease of learning especially via exploration.

Being on a small mobile device, the interface must be easy for the user to learn to navigate.

The HCI experts evaluating the design will be given our prototype and two tasks to complete: setting up preferences and finding an article of clothing. They will be provided with the list of actions sequences necessary to complete each task.

The HCI experts will be given a description of our persona (Amanda Lee) and users in the same demographic (18-29; Students/Recent Graduates, limited budget, hectic work schedules, etc.). Each instruction references the respective storyboard scenario and makes use of the UI frames on the functional prototype whose form factor is shown in the sketch prototype

The following tasks will be administered:

- Evaluators will be given the task of setting up user preferences (brand new users) and searching for a particular item in the area.
- Evaluators will be asked to locate a store after inputting their location.

Setting up preferences:

1. Software boots up.
2. Software displays introduction screen.
3. Evaluator first sets his/her gender and clicks continue.
4. Evaluator sets his/her birth date using drop down menus and clicks continue
5. Evaluator types the stores he/she frequently shops at to add it to the software, but it's optional and the user can choose to click skip instead.
6. A thank you screen shows up and the evaluator clicks continue.

Finding an article of clothing (suit):

1. Software boots up
2. (Assume user set preferences already) Evaluator clicks on lets go shopping.
3. (Suit doesn't appear in the suggestions) Evaluator types in suit and checks shoes (dress shoes are already listed).
4. Evaluator clicks on search button with items already checked.
5. A map comes up with numbers (bubbles). Evaluator clicks on 1 (Funplex mall).
6. Evaluator clicks on get directions which gives directions a la Google Maps.
7. Evaluator navigates to the mall, clicks on We're there.
8. Map switches to the mall view with numbers. Evaluator clicks on 1 and is given articles of clothing the store carries.

HCI Expert will then have to formulate a believability story by answering four questions;

- Will the user be trying to produce whatever effect the action has?
- Will user be able to notice that the correct action is available?
- Once user finds correct action at interface will she know that it is the right one for the effect she is trying to produce?
- After action is taken, will user understand feedback given?

The two main features of our prototype are tested from the standpoint of advanced experts: The locator which helps users navigate to the store they want, and the storing of the user preferences and how the device learns from the user's shopping habits. This feedback allows us to analyze strengths and weaknesses that were not caught or noticed by the average user.

Heuristic evaluation

A heuristic evaluation provides us detail and depth into our design that 'hands-on' evaluations such as the think aloud and the cognitive walkthrough misses or covers insufficiently. We decided to use Nielson's set of heuristics as it provides the most in-depth coverage of our design, each point will have sub bullets explaining how we test each heuristic.

All heuristics are carried out using the functional prototype with clicking and navigating through the software simulated using the different UI frames:

- Visibility of system status (is there appropriate feedback?)

  -Click different buttons, set options and navigate around to see if the system always responds in the way that we want it to (go to a different screen, changed settings, etc).

- Recognition over recall (are the features of each button easily recognizable by the user?)

  -Navigate around and look for a specific feature (such as editing preferences or setting connection/GPS options) and see how long it takes to find that feature or instructions for it.

- Aesthetic and minimalist design (are information and features always relevant to the goals of the user?)

  -Go through different screens while trying to locate a store or set preferences to see if options, features on each screen are relevant to what we are trying to accomplish.

- User control and freedom (does the system supports undo and redo?)

-If the software makes wrong suggestions, try to find a way to edit preferences such that the software improves its suggestions.

- Consistency and standards (do system actions follow conventions and are they context-independent?)

  -Click on buttons, use the navigation and location tools to make sure the buttons always trigger expected events and navigation/location tools provide navigation and location in a way that we expect it to.

- Flexibility and efficiency of use (does the system cater to novice to experienced users equally?)

  -Use the software from the standpoint of an inexperienced user who is computer-illiterate, and from the standpoint of a computer programmer.

- Recognition, diagnosis and recovery from errors (does the system recognizes the problem and suggests solutions?)

  -If the software crashes or has to abort, see if it provides the reason or explanation for it. If the software makes wrong suggestions, recognize if any way to improve its suggestions can be made.

- Help and documentation (are there options or a tutorial to help troubleshoot problems the user might have?)

  -Use the software for locating a store, setting user preferences and obtaining smart suggestions to see if any options detailing how to accomplish these tasks are available.

- Error prevention (is it difficult to make errors?)

  -Click buttons, navigate through screens, set random preferences to see if the software will crash or make wrong suggestions.

- Match between system and real world (is the system predictable and easy to communicate with?)

-Check that the software provides support for commonly spoken languages and that the words (terms) used in the software are not too simple or technical.

Each of these criteria will be analyzed by us and given a different grade based on; frequency, impact, persistence, and market impact.

The grading scale:

- 0 (not a usability problem)
- 1 (cosmetic problem)
- 2 (minor usability problem)
- 3 (major usability problem)
- 4 (usability catastrophe)

Ideally, we would like all of our criteria to have a 0 grading, but when time and budget constraints are considered, we would like to have a detailed set of functions that we implemented well and others that we need to improve. All of the features of our prototype and design are tested and provides us in-depth feedback from the back-end, while the think aloud and cognitive walkthrough provides feedback from the front-end. Using all three of these evaluation techniques is the most optimal with our design and hence contributes to our evaluation plan. At the end, we will have all the benchmark tests ready for the next stage of executing our evaluation plan.

Summary

The main goal of this project was to further develop our vision of helping users maximize their efficiency and satisfaction with their shopping experience. Our prototype tries to achieve this through a software based application that assists the user in locating and finding clothing. Taking into consideration the research and analysis completed in G2 and I3's we decided on three main features for our prototype: portability, locator, and storing/learning user preferences.

We initially had decided upon having a feature which suggested clothing to the user, but decided to drop this feature since it would require a lot of input on the user's part in order to get better at suggesting clothes. For example a user may reject suggested clothes for a variety of reasons and would need to specify why every time. The feature may also been ineffective in areas where the user doesn't make a lot of purchases such as buying a suit for a special occasion.

Another requirement discarded was the ability for users to put clothing on hold at the stores. This was dropped since this requirement was out of the scope of helping users find clothes. It also wasn't something expressed by users in our research. Lastly, it would also cause problems for the stores such as tying up inventory in the event another customer wants to purchase the item.

The requirement of an uninterrupted shopping experience was initially created with the intent of maximizing the user's efficiency and overall experience. Upon further inspection we decided to drop this feature due to the fact that emergencies do arise and it is important for the user to still be able to receive calls to their phone.

Our final usability criteria focus on the following core concepts:

- Portability; the software works with multiple devices.
- Location and navigation; the software locates the user (if GPS is supported) and provides directions using Google maps to the specified store/shopping mall in his/her vicinity.
- Setting and making use of user preferences; the software stores user specified preferences and preferences based on the users' past purchases.
- Making suggestions; the software makes suggestions on what to search for based on the frequency of the user's past searches.

In order to test these features we have created a physical mock-up with UI frames, sketch and storyboards. Using our prototypes, we outline and display the main features mentioned, as well as show how to execute an action such as setting user preferences and finding clothes. As outlined in our evaluation plan a cognitive walkthrough, think aloud, and heuristic evaluation will be used in order to test our prototype. Given the time constraints and limited resources we feel these give us the best coverage and testing available to us.

Our design is software based and as part of our usability criteria, available to multiple platforms and devices. The software is implemented as a program with a GUI and making use of several different forms of connectivity (WiFi, Bluetooth, GPS, 3G). It integrates navigation software similar to Google maps and has an internal database for storing searches, preferences, past purchases, etc. The software is low-cost and designed to be available as a free download for mobile devices. Help and documentation is implemented as a FAQ/Email form on the software website.

With the physical mock-up, UI frames, sketch and storyboards created, we can now move on to the next phase of product development. In G4 we will be able to put our evaluation plan into action which we have laid out in order to further test and develop our Stand Alone Unit Shopping Assistant Gadget Extension.

References

Banovic, N., Assignment I2. Retrieved from CSC318 Website:
http://www.cs.toronto.edu/~khai/classes/csc318-spring2009/submissions/I2-Banovic.pdf

Cooper, D., Considerations of the Tast of Buying Clothes. Retrieved from CSC318 Website:
http://www.cs.toronto.edu/~khai/classes/csc318-spring2009/submissions/I2-Cooper.pdf

Jalali, M., Assignment I2. Retrieved from CSC318 Website:
http://www.cs.toronto.edu/~khai/classes/csc318-spring2009/submissions/I2-Jalali.pdf

Lam, Y. Observing People Performing a Task. Retrieved from CSC318 Website:
http://www.cs.toronto.edu/~khai/classes/csc318-spring2009/submissions/I2-Lam_Y.pdf

Team Discovery Channel, Assignment G2: User Needs and Analysis. Retrieved from CSC318 Website:
http://www.cs.toronto.edu/~khai/classes/csc318-spring2009/submissions/G2-Team_Discovery_Channel.pdf

Mo and the Economists, Assignment G2: User Clothes Organization Needs and Analysis. Retrieved from CSC318 Website:
http://www.cs.toronto.edu/~khai/classes/csc318-spring2009/submissions/G2-Moe_and_the_Economists.pdf

Grumpy, Assignment G2: User Needs and Analysis. Retrieved from CSC318 Website:
http://www.cs.toronto.edu/~khai/classes/csc318-spring2009/submissions/G2-Grumpy.pdf

Banovic, N., Assignment I3a. Retrieved from CSC318 Website:
http://www.cs.toronto.edu/~khai/classes/csc318-spring2009/submissions/I3a-Banovic.pdf

Chow, J., Assignment I3a. Retrieved from CSC318 Website:
http://www.cs.toronto.edu/~khai/classes/csc318-spring2009/submissions/I3a-Chow.pdf

Khuu, S., Assignment I3a. Retrieved from CSC318 Website:
http://www.cs.toronto.edu/~khai/classes/csc318-spring2009/submissions/I3a-Khuu.pdf

Lee, P., Assignment I3a. Retrieved from CSC318 Website:
http://www.cs.toronto.edu/~khai/classes/csc318-spring2009/submissions/I3a-Lee_P.pdf

Ma, J., Assignment I3a: Designing Interactive Support for a Task. Retrieved from CSC318 Website:
http://www.cs.toronto.edu/~khai/classes/csc318-spring2009/submissions/I3a-Ma.pdf

Zou, T., Assignment I3a. Retrieved from CSC318 Website:
http://www.cs.toronto.edu/~khai/classes/csc318-spring2009/submissions/I3a-Zou.pdf