Location-Based Classifieds Mobile App

KEEPS TRACK OF GEO-LOCATION-BASED ADVERTISING OR OPPORTUNITIES FOR YOU

By: Edwin Li
Interaction Designer
Email: hci@edwinli.com
Phone: 510-371-485
Website: edwinli.com/ux
LinkedIn: www.linkedin.com/in/edwinli
Last updated: April 5, 2013

CONFIDENTIAL

Problem Statement: Manual Scanning of a Stream of Ads

Some websites offer a stream of advertisements posted by members, and present them in a chronological fashion. The website may or may not provide RSS feeds for their advertisements.

Key standard features of the advertisements for applicability of the app are location, time of posting, and advertiser. There should also be programmatically parse-able extractible other features of the .

Two primary determinants of relevance to users of the website looking for advertisements are location and recency of posting (within the last few days or hours). To keep abreast of which advertisers have advertised what recently, users must periodically scan the stream of ads, scanning each advertisement going back to the time of their last scan, mentally keeping in mind the features in which they're interested and matching each scanned advertisement against their mental list.

This scanning of ads requires a lot of time, and is often fruitless or inconveniently time-consuming.

Proposed Solution: A Mobile App That Scans for You with Notifications, etc.

One potential solution for this problem is an app for a mobile platform like a smartphone. With a list of advertisers in whom the smartphone's user is interested or just a set of features in which the user is interested and a data connection, the app can poll the advertising-stream, each time going back to the last scan's timestamp and scan each advertisement for any and all features of interest.

When matching advertisements are found, they are stored locally on the smartphone for reference whenever the user is interested in looking at the advertisements.

In local storage, the app will replace older ads with newer versions for the same item and/or by the same advertiser, so that the phone will always have the most recent version of each ad.

Users will also be able to search for particular favorite items to find when they were most recently offered.

Users should be able to set preferences for downloading all ads in the stream, only those that match certain criteria, or only those from favorite advertisers. The app will then basically behave as a more or less selective TiVo for the ad stream, showing users the "recorded" ads by favorite advertisers by default, but also offering users the opportunity to filter and search that locally "recorded" ad stream.

For this design, I have chosen to use HP webOS as my target platform due to my personal familiarity with it, its innate support for true multi-tasking, ability to support multiple simultaneous "cards"/"windows" for each app, and its unobtrusive, non-interruptive notification system. Devices running webOS are capacitive touchscreen devices, with or without a touch-sensitive "gesture area" below the display screen, and with various combinations of wireless technologies and sensors, including GPS location sensors, wi-fi, Bluetooth, and 3G cell data. The primary focus of this app's first iteration design would be for an HP webOS device with 3G, wi-fi, and GPS, most likely a phone.

Process

Even as a personal project, the creation of this mobile app will follow a user experience-focused process, incorporating best practices and iteration to create and develop an app that will be increasingly useful, usable, and delightful to use.

Here are steps along the journey:

1. **Domain research.** Are there aspects of the domain that may be identified as domain structure or constraints that may influence or be taken advantage of in the ultimate design?
2. **User behavior research.** How can users current behavior in the current domain-state be described? Does that behavior suggest positives or negatives with respect to accomplishing their goals within the domain? Are there regularities that may be taken advantage of when positive or done away with when negative?
3. **Digesting the data into design-useful artifacts.** This process may include diagramming, card sorting, re-representing the notes and data in various ways. Products of this process may include diagrams, personas, scenarios, sketches, lists of problems, lists of requirements, etc. This is a quote by Herbert Simon of particular relevance to me at this stage: "solving a problem simply means representing it so as to make the solution transparent." As such, this stage may very well blend and flow back and forth with the next stage, Design.
4. **Design.** Of course, this is an iterative and circular process with sub-processes. It can involve casting about for inspiration and models, lists and re-lists of requirements, content models, navigation models, task models, sketches, Balsamiq mockups, HTML, even Photoshop--whatever's quick, easy, easily let-go, and supportive of reflection.
    a. Sub-processes: Idea generation, Idea exploration, Make choices and "go deep", then repeat recursively as design problems are identified.

CONFIDENTIAL

5. **Develop.** This will interact with the Design processes as constraints and interactions between elements are identified.
6. **Usability/User Experience testing.** What's it like to use the product at its various stages of being designed and developed? It should be tested, evaluated all along the way, with lessons learned from testing adding to the mix of Idea generation, Idea exploration, "going deep", etc.
7. **Wrap up version and release…**
8. **Repeat the processes looking towards future versions.**

Process Steps Represented In This Document

This document represents a snapshot of the state of the described Location-Based Classified Ad Mobile App in terms of the process outlined above and how it got there.

1. **Domain research.** Done as a user of various relevant websites. While self-observation and introspection can often be suspect as a source of research, as a personal project with the self projected as a primary user, I judge this acceptable. **Artifacts:** notes.
2. **User research.** As with domain research, the data collected here comes from self-observation and introspection. **Artifacts:** notes, lists of tasks, lists of goals, lists of problems, requirements, etc.
3. **Digesting the Data. Artifacts:** lists of requirements, problems, potential solutions, functionality in service to problems, etc.
4. **Design.** Part of the design comes from the inspiration/choice of a mobile platform/app as the basis of the solution. Further design constraints and structure come from the choice of HP webOS as the platform of choice. Artifacts: content models, navigation models, sketches, Balsamiq mockups.

This is as far as my current process has gotten, and I am still in the design stage, with forays into testing via paper/sketch/Balsamiq-based prototyping and "back" into digesting data.

Artifacts Included in This Document

Included in this document are sample artifacts from the unfinished process of creating the application:

- Brainstorms: Lists of scenes, screens, functionality
- Navigation models: what screens there may be and how might they connect
- Content models: what functionality, types of widgets, and interactions might be needed on what scenes/screens.
- A usage scenario, drawn as a navigation path through abstract screens.
- Sketches for some screens
- Mockups created in Balsamiq Mockups, exported as PDF pages.

CONFIDENTIAL

Data, Digesting the Data, and The Beginnings of Design

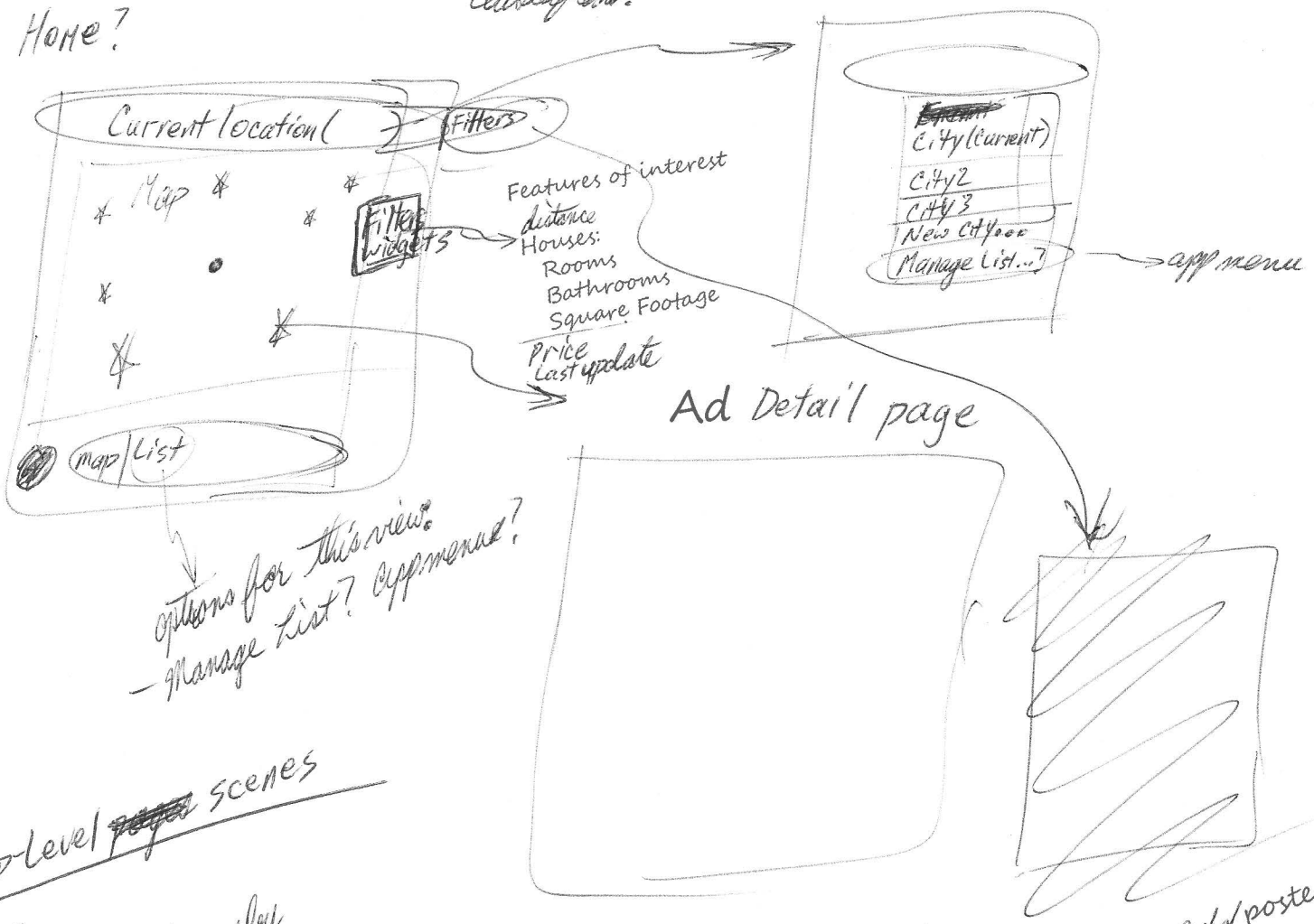
In this section: Lists, Content Models, Navigation Models, Scenarios, and Actual Sketches

CONFIDENTIAL

RB App

11/6

- also may apply to housing/rental apps for classifieds!



options for this view:
- Manage List? App menu?

Top-Level ~~Pages~~ scenes

~~Home~~
Map w/ results nearby
List of results near you
RB Ads list, current? } maybe just directories, fan out to browser page

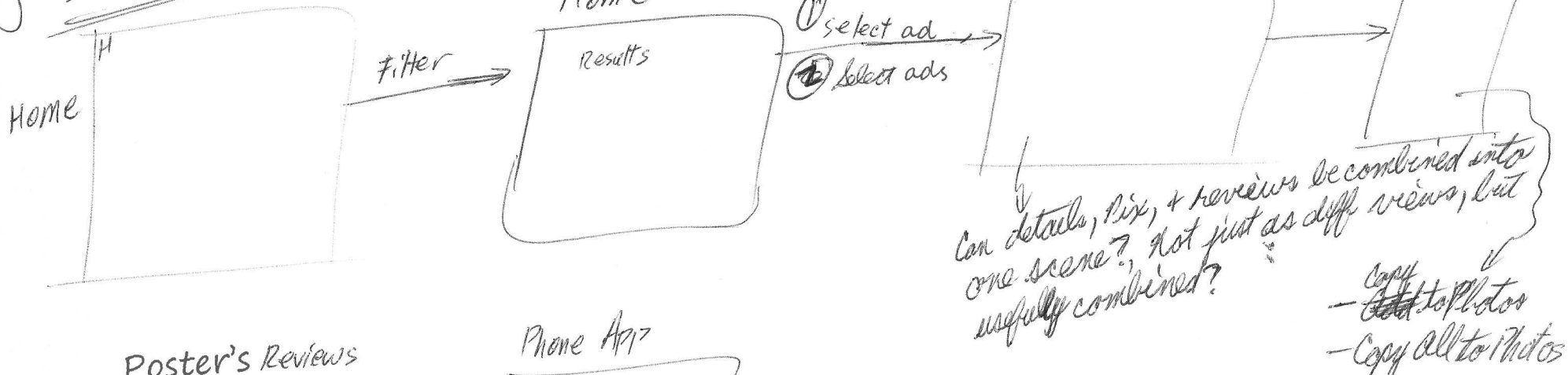
Functionality

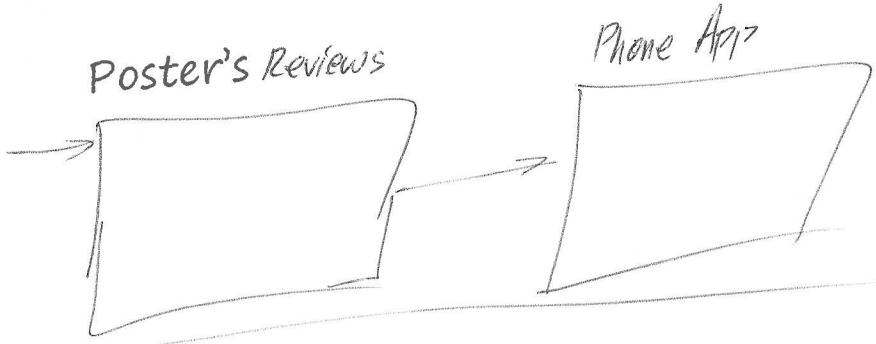
- Add to Contacts
- Select what info to add.
- Select account to add to.

Scenes, Other

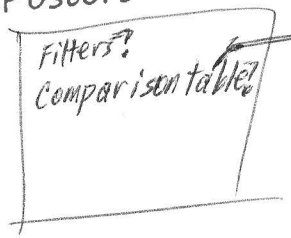
- RB ad search + results ◦ add poster
- RB Classified search + results ◦ add poster
  - favorites list
  - Inbox
- Cities of interest
- Standing filters?
  - default
  - eg, rooms, bathrooms, specs
- Preferences
  - Polling freq.
  - Notifications
    - specific favorites?
    - attributes?
    - prices, etc?

(2) Scenario





(2) List of Posters of Current Interest



Backupdated

Reviews
- Average rep
- Average condition
- Configurable
- Simple...
- Pickup etc.

View Multiple ad details. Separate cards for ea?

Home: Content Model

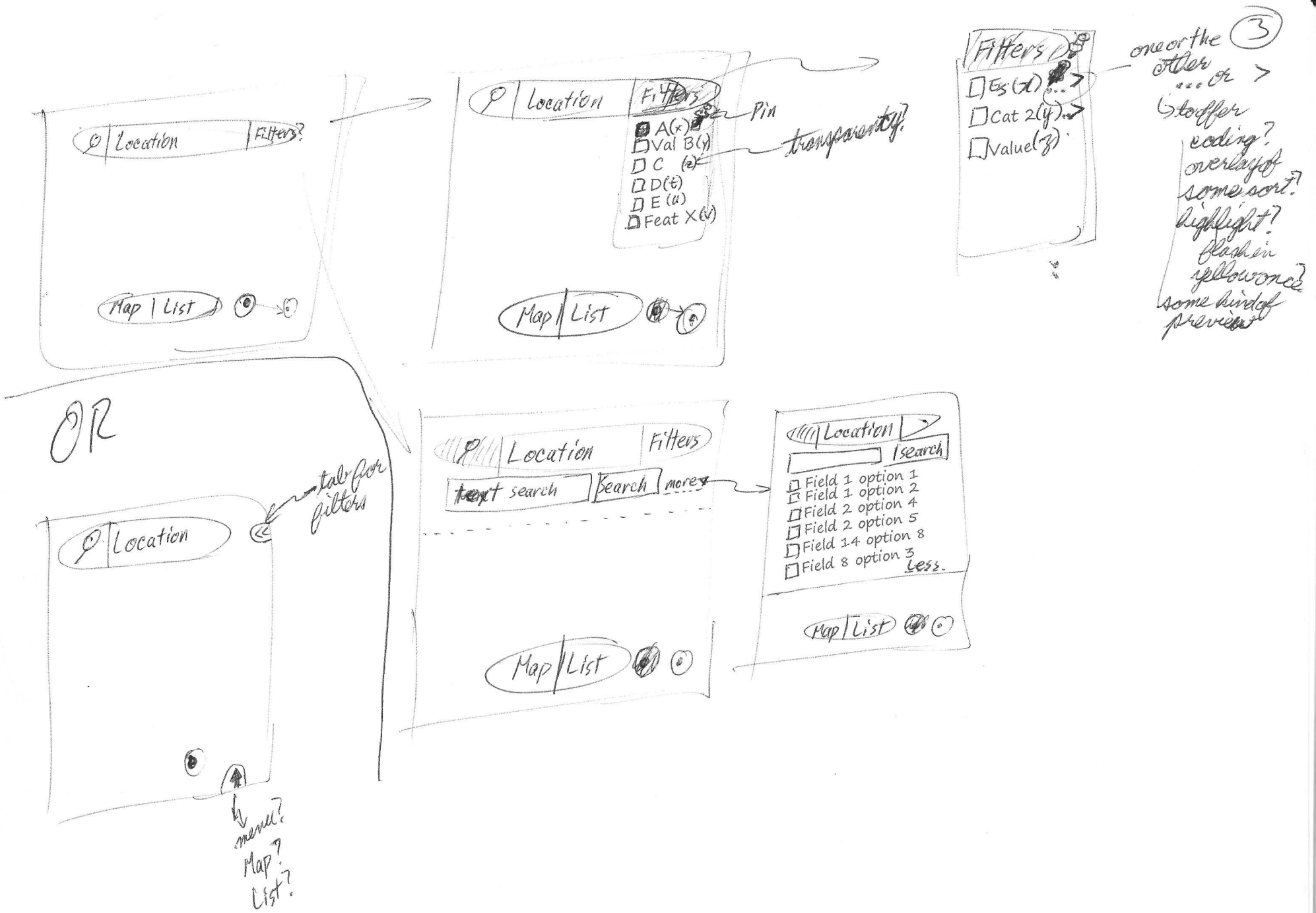
- Map
- ~~Radius~~
- Results — on Map (name) → Color?coded for ~~fast~~ update
  - List view → Coded for: Main cats, Target feature values
    - Pick small # of interest features + define/collect coding via Preferences
    - Seen + ~~repeat~~
    - Seen +
- Filter list → gateway
- Filters of greatest interest
- Location being viewed
- Location selector
- New location add-er
- View list directly/current
  - Content nave → drawer?
- Search for /name
  - Options? Areas? City? Feature(s)?



Result1, Result2, Result3...

Result

Map | List | Filters


Location
Filters?
Map | List
Location
Filters
A(x)
Val B(y)
C (z)
D(t)
E (u)
Feat X(v)
Pin
transparency.
Filters
Es (x)
Cat 2(y)
Value(z)
one or the other ... or >
3
tooltip
coding?
overlay of some sort?
highlight?
flash in yellow once?
some kind of preview?
OR
Location
tab for filters
Location
Filters
text search
Search
more
Location
search
Field 1 option 1
Field 1 option 2
Field 2 option 4
Field 2 option 5
Field 14 option 8
Field 8 option 3
Less.
Map | List
Map | List
hamburger menu?
Map?
List?

RB App 11/7/2010

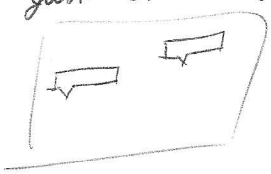
Just Type → Quick Actions for RB App?

- Provider name filter / search?
- Provider search by attribute? ~~filter~~ apply particular filter?
- Location setting?
- Allow some sort of syntax for greater complexity in searching or filtering?
  - ~~Too~~ Too complex?

↳ Ex <attribute-value> in <location> → Top App's Quick Action

<Name> Quick action → [name] [search] OR just the results

[search] ?



Mockups

In this section: Mockups via Balsamiq Mockups

About the Balsamiq Mockups

In this document is a set of mockups done in “Balsamiq Mockups for Desktop” (from http://balsamiq.com/) for the app described above. It covers only a few of the primary screens and is focused primarily on the experience of finding and seeing what advertisements of interest have recently been advertised.

Navigating the Balsamiq Mockups

Start on the first page of the mockups in this pdf. Comments that look like “sticky notes” on each page indicate what should be tap-able/clickable.

There are also comments included that provide some insight into the questions and considerations faced as part of the design process.

CONFIDENTIAL



This is the first screen of the application, what users first see on opening the app.

ASSUMPTIONS

- Users have previously configured the app. This includes a list of cities in which they might be interested, features they are looking for in the advertisements (whether product features or advertisers).

- Users have set an application preference for the app to show them relevant advertising around their current location (Palm Pre has a GPS, and so is location-aware), if it is one of the locations they've previously defined for interest.

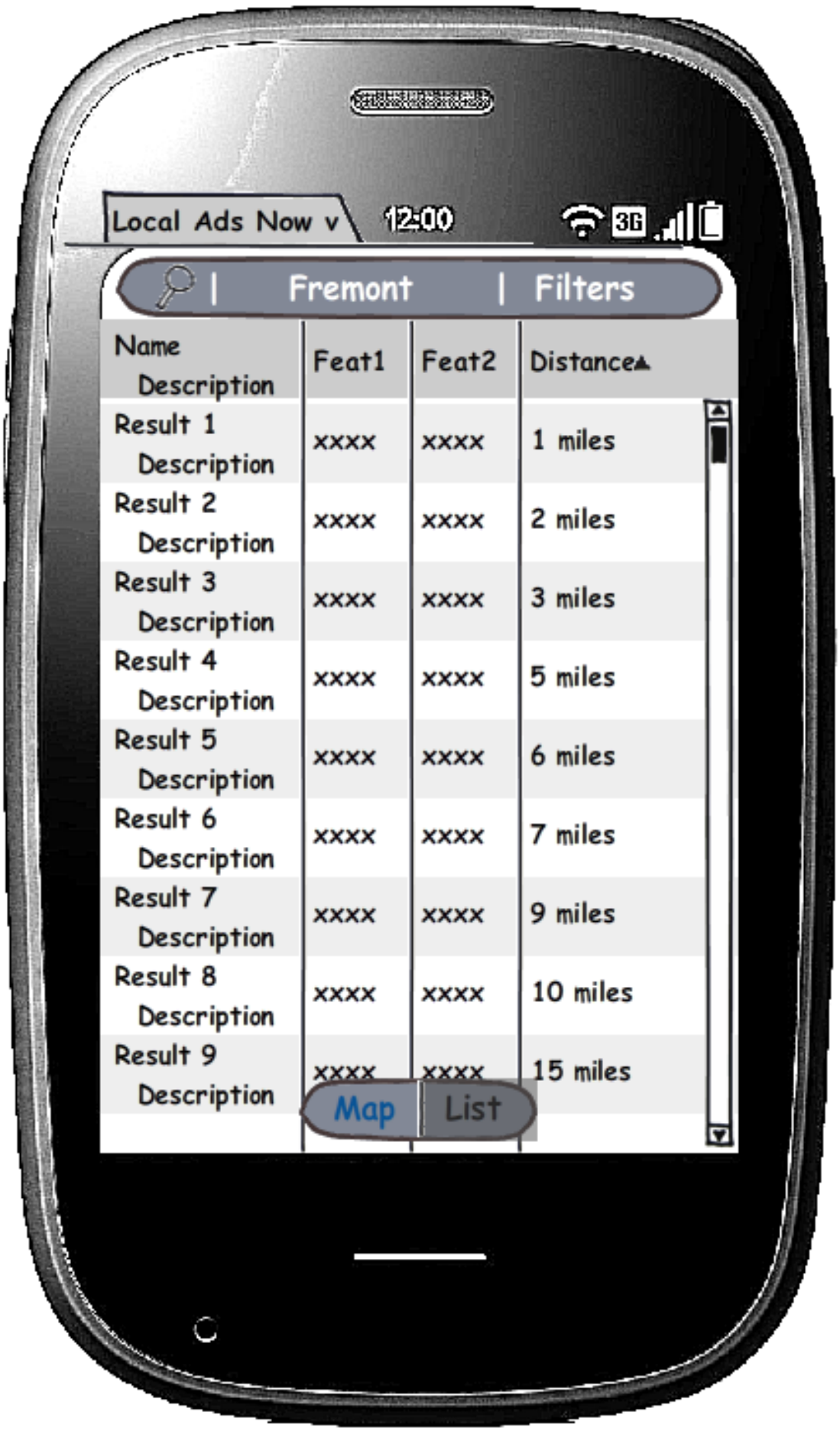
INTERACTING WITH THE MOCKUPS

Hotspots have been defined and linked, so that clicking the hotspots can approximate the usage experience of the application. Many of these hotspots are recognized as tap-able via convention in webOS. For those unfamiliar with webOS and even for those very familiar with it, here is the subset of potentially tap-able UI elements that have been linked/made tap-able here on the main "scene"/screen:

- In the header: magnifying glass, Fremont, Filters

- Results in the main body of the screen

- View control: Map or List, whichever is not currently viewed.



Tap-able UI elements:

- Header items
- Results table contents



This mockup is a work-in-progress, included as an illustration of a minor design trade-off and the considerations there-of.



New Functionality: Label results with name/text, picture, or both.

Design Problem: Where, how to show it.

Solution candidates:

- Buttons for layers, a la Google Maps' traffic, satellite, terrain, etc.
- Buttons, essentially, but hidden behind a tab for a drawer, indicated by the "rewind"-looking icon.
- Buttons, accessed via a menu that comes up when you tap the Map view on the bottom.



Tap-able UI elements:

- Header items
- Map
- View control



Tap-able UI elements:

- Header items
- Results table contents



Tap-able UI elements:

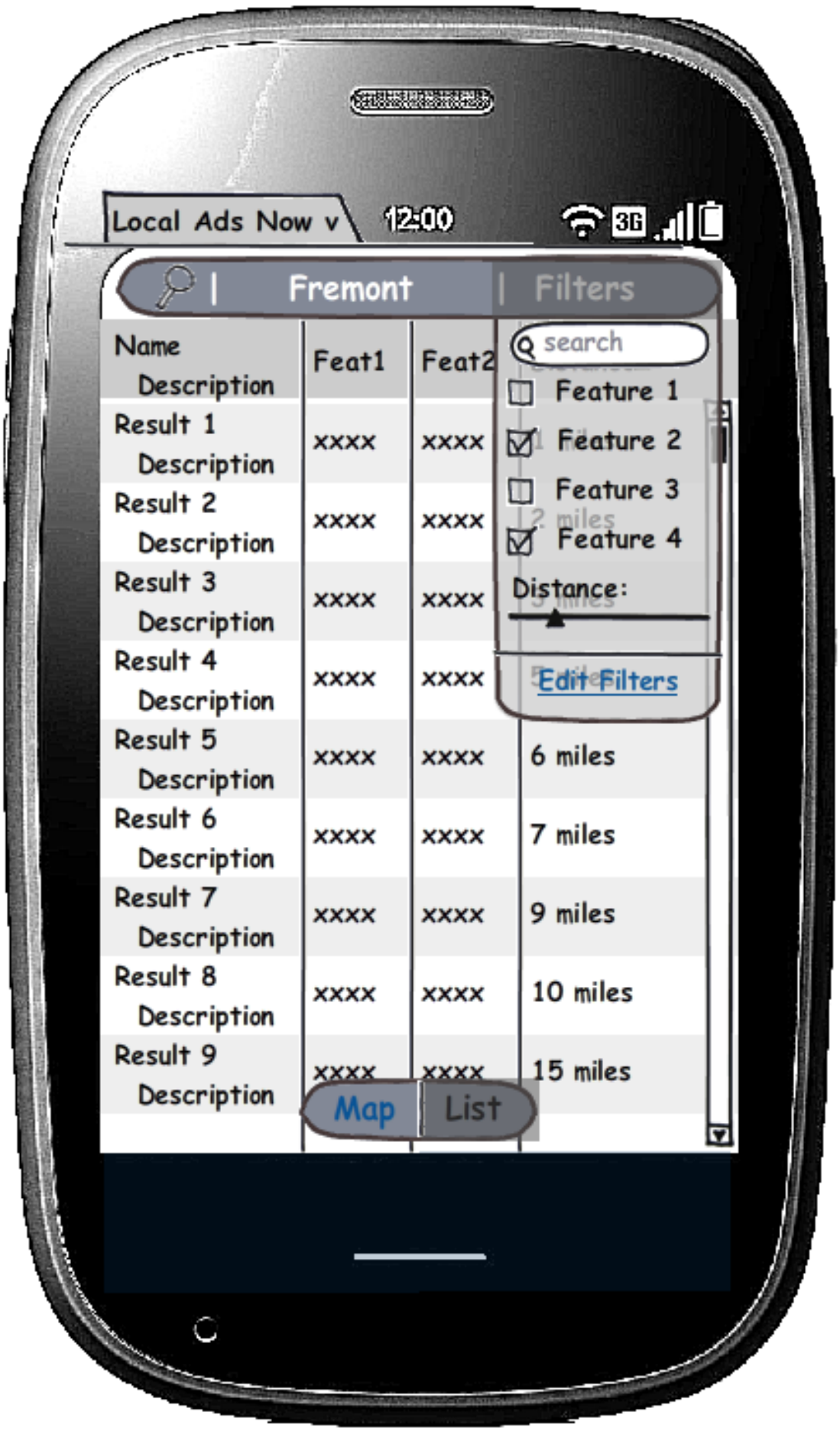
- Header items
- Slider drag-point, to simulate what users will see as they hold the slider control.
- Map
- View control



Tap-able UI elements:

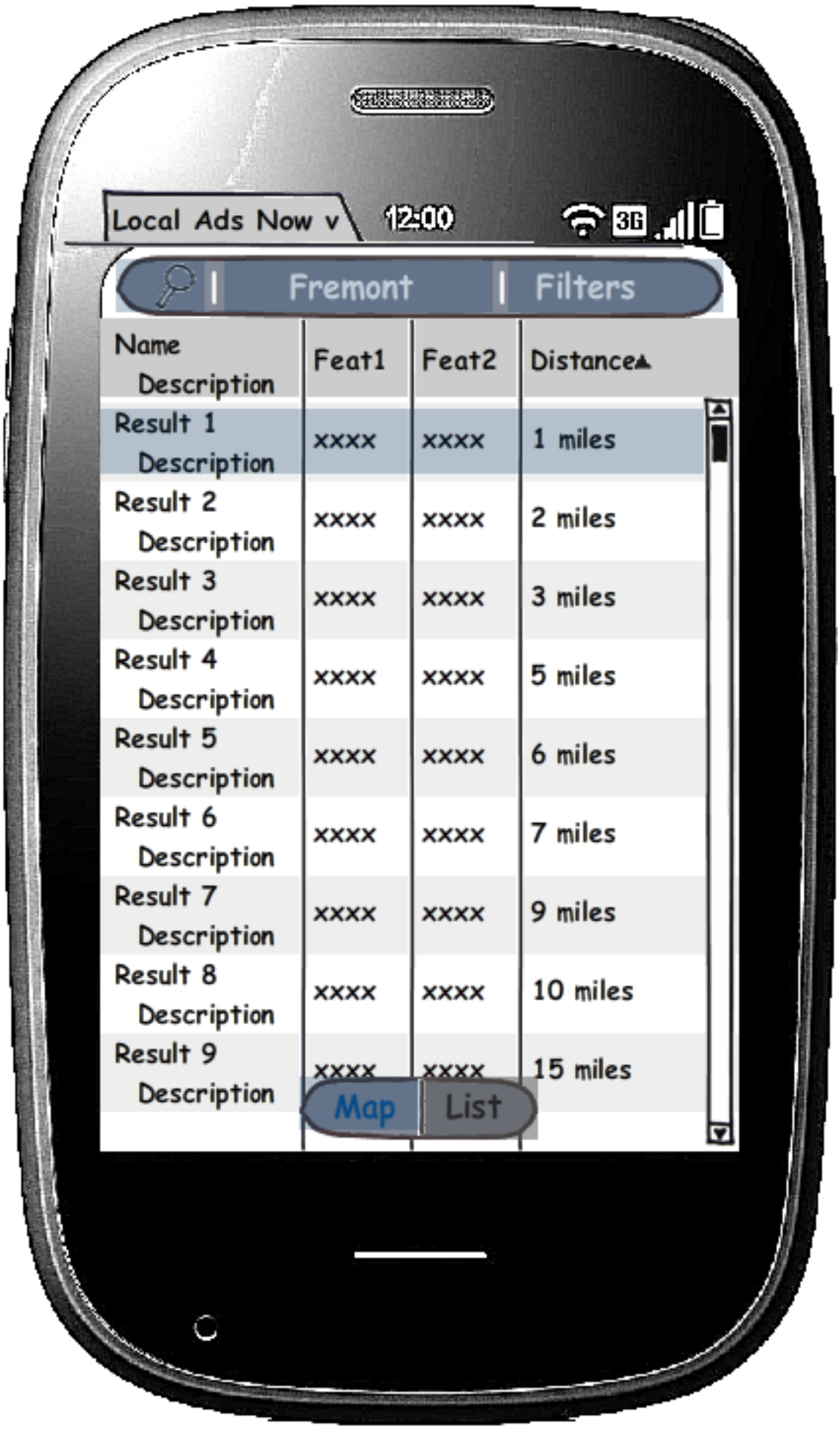
- Header items
- Map
- View control

Note the pop-up above the slider control to show users what value they're at with the slider.



Tap-able UI elements:

- Header items
- Results table contents

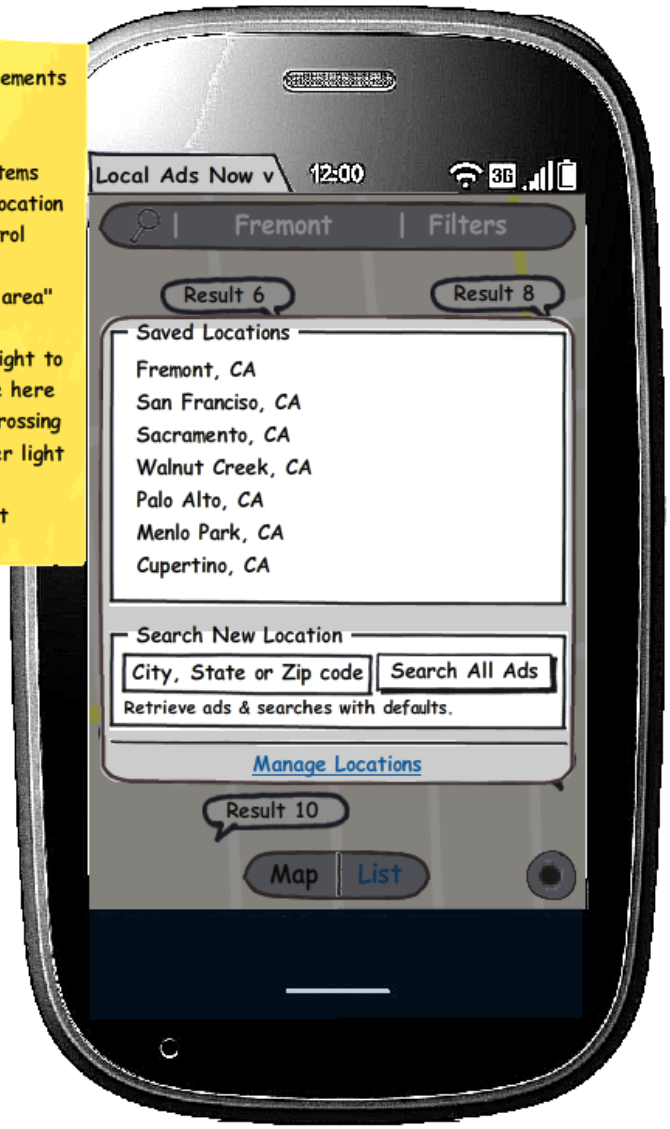

Local Ads Now v
12:00
3G
Fremont
Filters
Name
Description
Feat1
Feat2
Distance
Result 1
Description
xxxx
xxxx
1 miles
Result 2
Description
xxxx
xxxx
2 miles
Result 3
Description
xxxx
xxxx
3 miles
Result 4
Description
xxxx
xxxx
5 miles
Result 5
Description
xxxx
xxxx
6 miles
Result 6
Description
xxxx
xxxx
7 miles
Result 7
Description
xxxx
xxxx
9 miles
Result 8
Description
xxxx
xxxx
10 miles
Result 9
Description
xxxx
xxxx
15 miles
Map
List



Tap-able UI elements on left mockup:

- Header items
- Fremont location
- View control

- "Gesture area" below the screen: right to left swipe here without crossing the center light is used throughout



City Selector & Quick New City Search: 2 Possibilities

LEFT: A drawer
Pros: matches the behavior of the search and filters drawers.
Cons: Takes up a lot of the screen. Inconsistent with some other apps (e.g., Accuweather). Is it too big or complicated for a drawer?

RIGHT: A popup...
Pros: With the overlay, there's no need to feel self-conscious about taking up so much of the screen.
Cons: May be too big for a popup. Certainly, it's much more complex than the "dropdown selector" pop-up of Accuweather's city changes.
Cons: May be violating UI idioms. WebOS has pop-up menus that show up in the middle of the screen, but those are essentially for dropdown selectors, very simple 1-of-n choices. WebOS also has a pop-up modal dialog that pops up from the bottom of the screen, but that's much more for "dangerous" actions.

SOME OTHER SOLUTION???



Tap-able UI elements:

- WebOS "gesture area" below the screen

In layout and overall scene behavior, use Contacts details cards as a model.

IDEA: The app's bookmarked or favorited results/advertisers should be accessible as the app's own "address book". Each advertiser should have an option in their card to add their contact details into the phone's Contacts app.

Group Name