Nick Winters & Mandy Askew
Cse 403 Sp03

FineDinerFinder

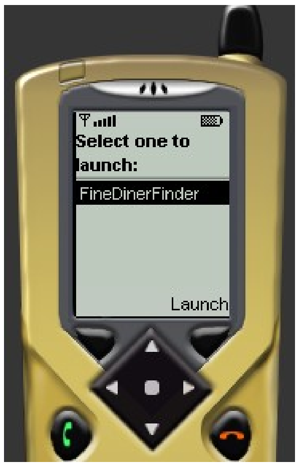
Life Cycle Objectives:

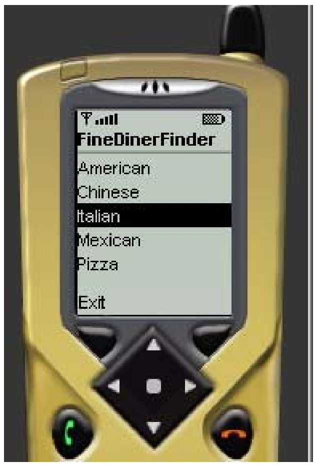
1. Operational Concepts – What is it?

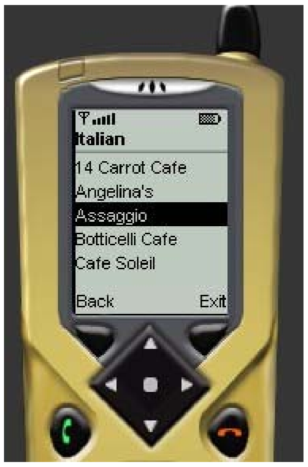
The purpose of FineDinerFinder is to give users a convenient way of finding a restaurant in Seattle without having to be in front of a computer to look one up. Users can select the type of food they want and will see a list of possible restaurants. It will give them names, addresses and phone numbers. It will also keep track of restaurants they have visited and ratings and comments they have entered for those restaurants.

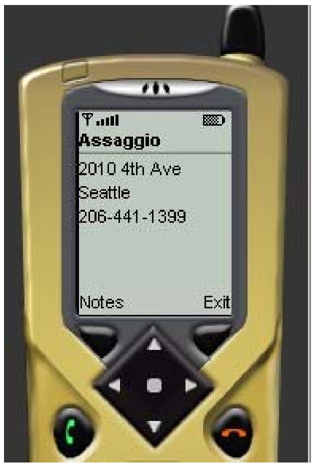
Scenario:

A couple are out driving home from a movie and decide they want to stop somewhere for dinner. They know that they want to eat Italian food but they're tired of all the usual places they go to. Normally they'd look online but they don't want to drive home to look something up. So one of them gets out their cell phone and opens up FineDinerFinder. They choose the Italian food option and are given a list of possible restaurants. They scroll through the list and see checkmarks by the restaurants they've already visited. They select the restaurant "Assaggio" then view the address and phone number. They drive there and have dinner. Afterward, they decide this is a restaurant they want to remember so they go back to FineDinerFinder, find the restaurant and go to the "notes" section. They enter a rating of '8' and possibly write a short comment. The next time they're looking for Italian food they'll see a check mark by this restaurant to indicate they've already been there.







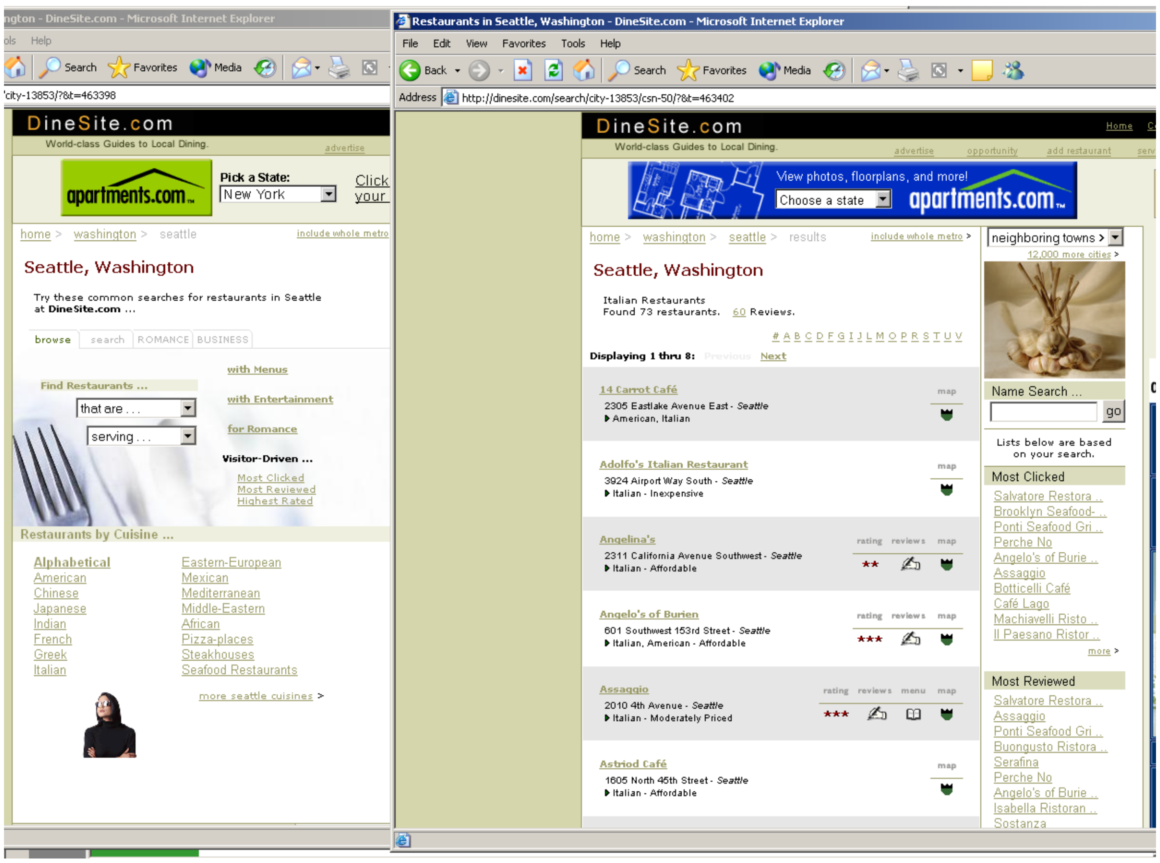


2. System Requirements – What does it do for us?

The interface for FineDinerFinder will look like the picture above. It will have a start screen, then a screen with a list of types of food. When a type is selected the new screen will have a list of restaurants with visited restaurants having a small icon to the left of their name, possibly a check mark. When a restaurant is selected, the screen will show the address and phone number of the restaurant. If the phone number is selected you can automatically dial the restaurant.

If you then select the notes command, a screen allowing you to specify a rating, some notes about the restaurant, and an option to mark that you have visited this restaurant will be shown. When you mark somewhere as visited an icon will show up next to the restaurant's name to denote it as visited. Press send to submit these changes and they will be saved to the database. Next time the application is used these changes will be loaded from the database and if the restaurant is selected the user will be able to view them.

FineDinerFinder will use the website http://www.dinesite.com for its data. This website provides a list of cities, types of food, and all the restaurant information needed. This website also contains other useful information like reviews, ratings, and maps. Possible enhancements to FineDinerFinder would include displaying some of this other information to the user.



Seattle, Washington

Try these common searches for restaurants in Seattle at DineSite.com ...

Seattle, Washington

Italian Restaurants
Found 73 restaurants. 60 Reviews.

A B C D F G I J L M O P R S T U V

Displaying 1 thru 8: Previous Next

14 Carrot Café — map
2305 Eastlake Avenue East - *Seattle*
▸ American, Italian

Adolfo's Italian Restaurant — map
3924 Airport Way South - *Seattle*
▸ Italian - Inexpensive

Angelina's — rating ★★ reviews map
2311 California Avenue Southwest - *Seattle*
▸ Italian - Affordable

Angelo's of Burien — rating ★★★ reviews map
601 Southwest 153rd Street - *Seattle*
▸ Italian, American - Affordable

Assaggio — rating ★★★ reviews menu map
2010 4th Avenue - *Seattle*
▸ Italian - Moderately Priced

Astriod Café — map
1605 North 45th Street - *Seattle*
▸ Italian - Affordable

3. System and Software Architecture – How?

The components of this project consist of a midlet, a servlet, and a database.

Midlet

To implement this project we will need to write a midlet to display the restaurant information sent to the phone from the server. It will use basic menus and some textboxes and checkboxes. We will also need an image file to use as our "visited" icon.

The following are the four different screen objects:

1. Cuisine List
   The cuisine list will have only one menu called Exit.
2. Restaurant List
   The restaurant list will have only two menus: Back and Exit. The default Select key from the phone will be used to select the restaurant.
3. Restaurant Detail
   The restaurant list will have only two menus: Notes and Exit. The name of the restaurant will be the caption for the screen. The address and phone number will be displayed here.
4. Notes
   The Notes menu is used to display the Notes screen object. This gives the rating and previously entered notes.

Servlet

We will write a servlet to get the webpage for the information the midlet requests. It will need to parse the html and extract the needed information. This servlet or possibly another one, we will also need to query the database and check for restaurants that match. The servlet will send all this information back to the cell phone. Finally, the midlet will display the information for the user.

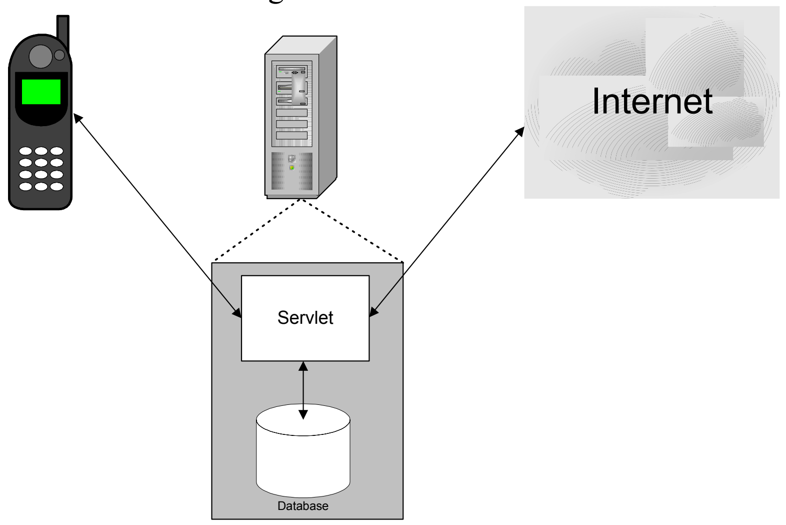
The following is the functionality of the servlet based on the current screen of the midlet:

1. Cuisine List
   The servlet will download the main web page containing all possible types of cuisine.
2. Restaurant List
   On the website, there is a link to navigate to the list of restaurants in the particular cuisine. The servlet will need to follow this link to get this list of restaurants. The servlet will also need to check the database to see if a certain restaurant has already been visited.
3. Restaurant Detail
   The servlet will follow the above process to scrape the web page and return the detailed information for that restaurant.
4. Notes
   The servlet will need to make a request to the database and return the notes previously entered and rating (if any) of that restaurant to the midlet.

Database

We will use a database to store relevant information from the user such as all the names of the restaurants that the user has denoted as "visited", any notes the user made about that restaurant, and their rating for it.

Below is a diagram of the architecture:



4. Lifecycle Plan – Who wants it? Who'll support it?

FineDinerFinder is directed toward people who want to eat at a new restaurant and are not able to look one up on their computer. It is targeted toward adults who eat out frequently.

Some maintenance will be needed. Someone will need to continually check the website (www.dinesite.com) to make sure that any updates or changes do not affect the way FineDinerFinder works. If the site is changed the servlet that parses the site may also need to be changed. Allowing restaurants to advertise through this service could offset the cost of this person. Restaurants could pay to have a special icon next to their name or have their name in a different more eye-catching color.

We will not be targeting a specific phone type or phone service provider for this application. As far we know, it should work with all phone types.

5. Feasibility Rationale – Is this really going to work?

This application can work successfully with the above design. However, a few known issues are dependencies on the website from which we scrape the data. If the website was offline during a request from the user, it would not return the results desired. A solution to this could include several different websites to serve as a backup data provider in case one was to fail.

Another issue is the number of round trips to the website. Currently the design calls for three requests to the website. This could increase the time to serve the user's request. A solution to this could an additional servlet scheduled to scrape the website at a non-peak time and store the results in a database.