RouteShare

Find.map.share bike routes

Specification Document

Vanessia Wu (vywu)

02/10/05

Description

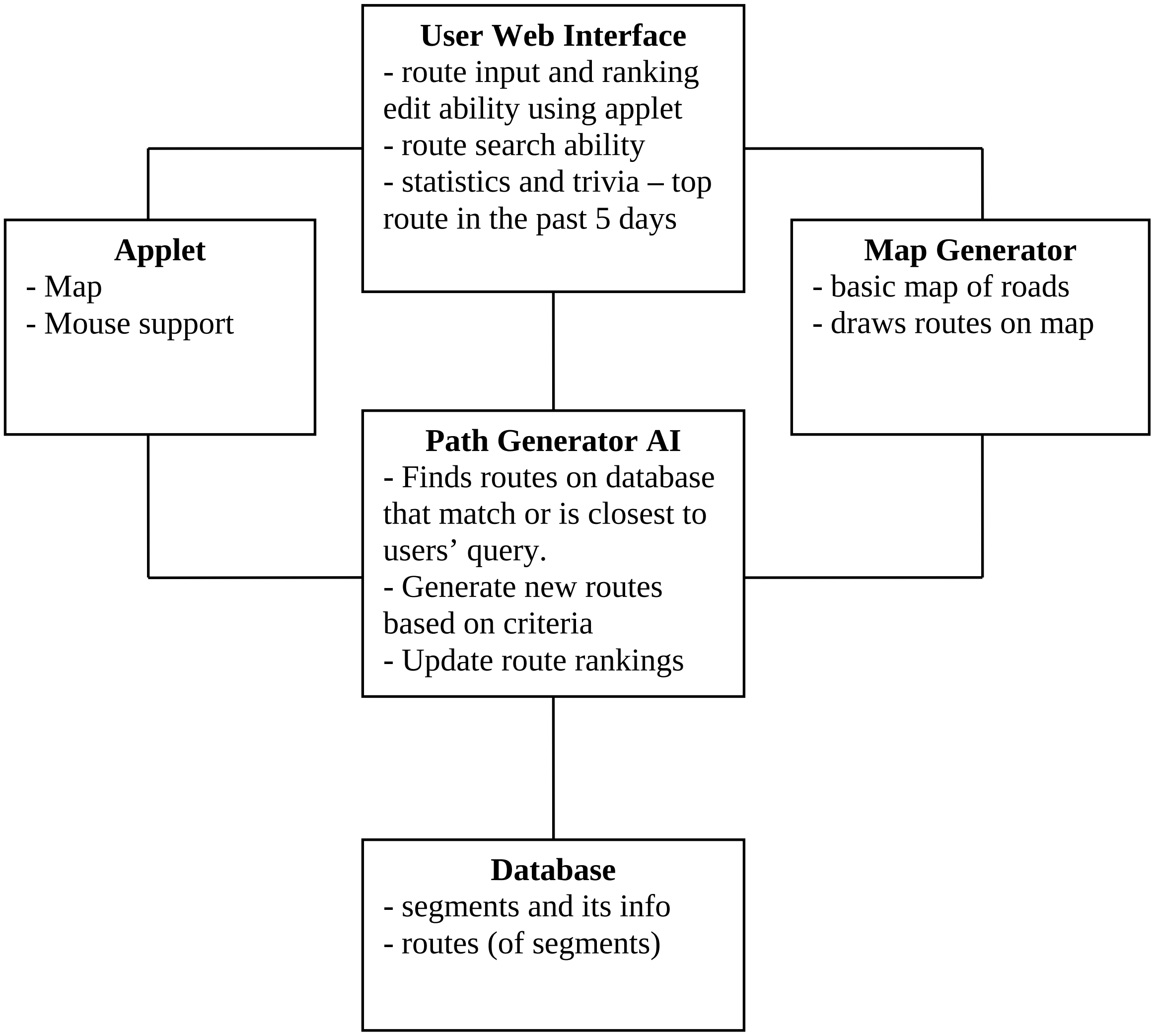
RouteShare (formerly known as BikeQuest) aims to facilitate the sharing of useful bike route information collected by cyclists. This includes finding, mapping and rating of bike paths. Existing mapping service providers such as Mapquest are designed for automobile drivers and are useful for mapping shortest and fastest routes between two points. However, they do not have the ability to provide information along the route that are extremely useful to cyclists, such as terrain, pavement conditions, scenic-ness and amount of automobile traffic on the route, that are particularly useful for cyclists' needs.

RouteShare will provide an interface for users to input new routes or loops (referred together as routes below), a search engine to search for the best-matching existing routes given a starting point, a map generator function that displays the resulting routes, and a feedback system for cyclists to rank routes and flag potential dangers on the routes. Routes will be stored by segments as defined by each turn or junction or intersection. Each segment will have a fixed distance and dynamic ratings of terrain type, pavement conditions, scenic-ness and amount of traffic. The dynamic ratings will have numerical (e.g. on a scale of 1-5) or descriptive (e.g. flat, rolling, hilly, very steep terrain) scales. The dynamic ratings can be pre-determined by available GIS topography, road type and traffic data and adjusted by users. Users will have the ability to edit the rating after their or someone else's initial input to reflect changes on the road, such as flagging dangerous potholes and blind spots.

Once the database is populated by routes, users can search for their desired routes by distance, terrain and other factors mentioned. If no matching route was found, a path-generator AI will help the user find the closest matching route. If the user is starting from a destination far away from all routes in the database, then the path generation AI would ask the user to put in a constraint (e.g. the least traffic 10 mile loop from A) and find a new route for the user.

Finally, RouteShare will encourage rider's feedback by rating routes they have recently visited, such that the average score for the routes (which will be the final route rating) will constantly change based on rider's feedback. It will have features showing the most popular routes, ask users to vote on the worst hills and other trivia questions designed for sharing useful knowledge on biking in the locale. The CS190 version of RouteShare will focus on Rhode Island cyclists and bike routes.

System Model Diagram



User Web Interface
- route input and ranking edit ability using applet
- route search ability
- statistics and trivia – top route in the past 5 days

Applet
- Map
- Mouse support

Map Generator
- basic map of roads
- draws routes on map

Path Generator AI
- Finds routes on database that match or is closest to users’ query.
- Generate new routes based on criteria
- Update route rankings

Database
- segments and its info
- routes (of segments)

System Model Diagram

User Interface (Web)

- As the name suggests, it is the Web platform through which the user interacts with the program. Users interact with it in several ways:
    - Users who wish to search for a route selects whether or not they want to search for a loop (begins and ends at the same point) or a route (begins and ends at different points). They select the distance they want to travel, and can opt to specify with pull-down menus the average terrain, pavement conditions, traffic and scenic-ness that their routes should have.
    - Users who select to provide a new cycling loop or route will have an applet opened for them to input them graphically.
    - Users who wish to provide feedback for the loop they last visited (which is known by means of a cookie or by registration) will have an applet of the route opened for them to select a segment to input their ratings or provide further comments on.
    - Users can also see some statistics and interact with trivia questions relating to cycling and bike routes.

Applet

- Allows users to input new routes and give feedback on existing ones.
    - Input new routes: The applet will display the base map (map with all possible, bike-able roads) and users simply have to click on the start and end of a road segment to join them. The graphics is instantaneously updated and they can unclick a segment to deselect it. The users can then click a button to submit the route to the server.
    - Give feedback on routes: Users can click on a road segment, and a side menu will show up with pull-down menus for users to give ratings on terrain, pavement conditions, traffic and scenic-ness.

Path Generator AI

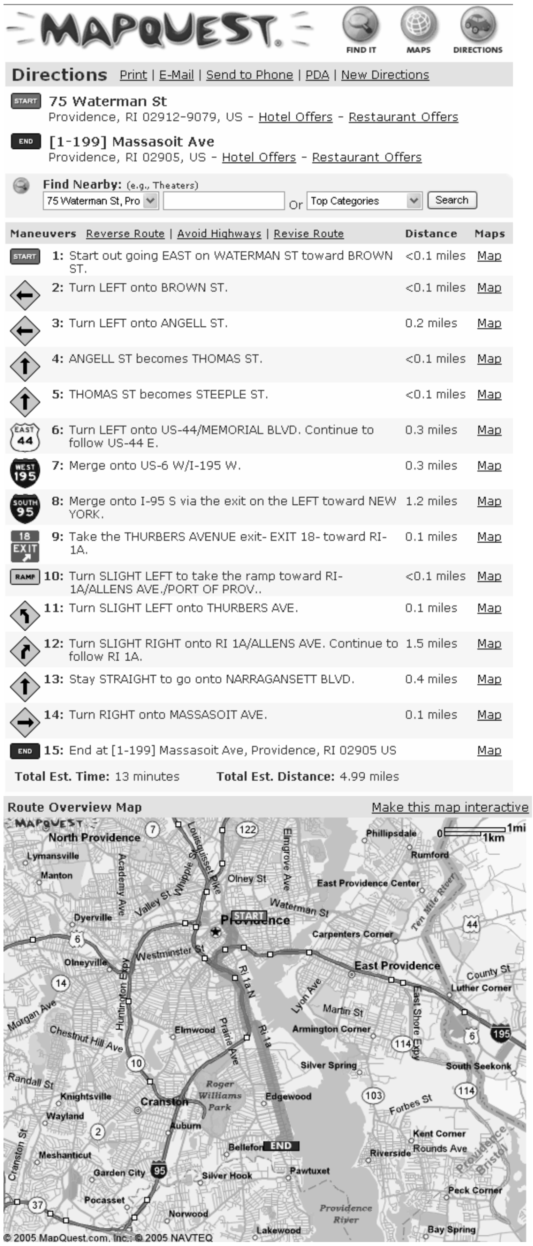
- Handles all user path queries, and seeks to find the best route, either by finding the best-matching one in the database, or if there is no close match, generating a new route and storing it.
- Determines the average rankings of certain attributes of the routes, and based on the user’s weightings (priority), return the best-fitting route to the user.

Map Generator

- Given a route (list of road segments), draw it out on the base map.

User Interface

Current programs:



Directions Print | E-Mail | Send to Phone | PDA | New Directions

START **75 Waterman St**
Providence, RI 02912-9079, US - Hotel Offers - Restaurant Offers

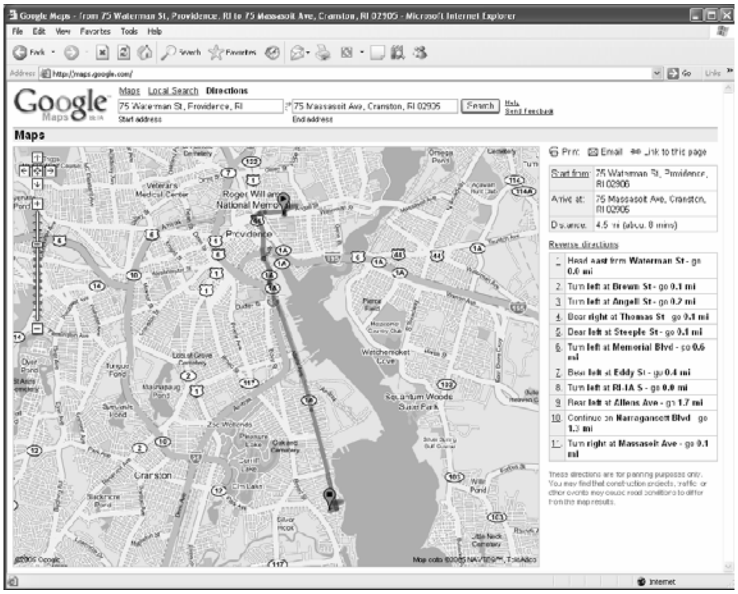
END **[1-199] Massasoit Ave**
Providence, RI 02905, US - Hotel Offers - Restaurant Offers

Find Nearby: (e.g., Theaters)
75 Waterman St, Pro Or Top Categories Search

Maneuvers Reverse Route | Avoid Highways | Revise Route	Distance	Maps
START **1:** Start out going EAST on WATERMAN ST toward BROWN ST.	<0.1 miles	Map
2: Turn LEFT onto BROWN ST.	<0.1 miles	Map
3: Turn LEFT onto ANGELL ST.	0.2 miles	Map
4: ANGELL ST becomes THOMAS ST.	<0.1 miles	Map
5: THOMAS ST becomes STEEPLE ST.	<0.1 miles	Map
EAST 44 **6:** Turn LEFT onto US-44/MEMORIAL BLVD. Continue to follow US-44 E.	0.3 miles	Map
WEST 195 **7:** Merge onto US-6 W/I-195 W.	0.3 miles	Map
SOUTH 95 **8:** Merge onto I-95 S via the exit on the LEFT toward NEW YORK.	1.2 miles	Map
18 EXIT **9:** Take the THURBERS AVENUE exit- EXIT 18- toward RI-1A.	0.1 miles	Map
RAMP **10:** Turn SLIGHT LEFT to take the ramp toward RI-1A/ALLENS AVE./PORT OF PROV..	<0.1 miles	Map
11: Turn SLIGHT LEFT onto THURBERS AVE.	0.1 miles	Map
12: Turn SLIGHT RIGHT onto RI 1A/ALLENS AVE. Continue to follow RI 1A.	1.5 miles	Map
13: Stay STRAIGHT to go onto NARRAGANSETT BLVD.	0.4 miles	Map
14: Turn RIGHT onto MASSASOIT AVE.	0.1 miles	Map
END **15:** End at [1-199] Massasoit Ave, Providence, RI 02905 US		Map

Total Est. Time: 13 minutes **Total Est. Distance:** 4.99 miles

Route Overview Map Make this map interactive



Google Maps - from 75 Waterman St, Providence, RI to 75 Massasoit Ave, Cranston, RI 02905 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://maps.google.com/
Go
Links
Google Maps BETA
Maps Local Search Directions
Search
Help
Send Feedback
Maps
Print Email Link to this page
Reverse directions
Roger Williams National Memorial
Providence
Veterans Medical Center
Cranston
Pawtuxet
Map data ©2005 NAVTEQ™, TeleAtlas
©2005 Google
Internet

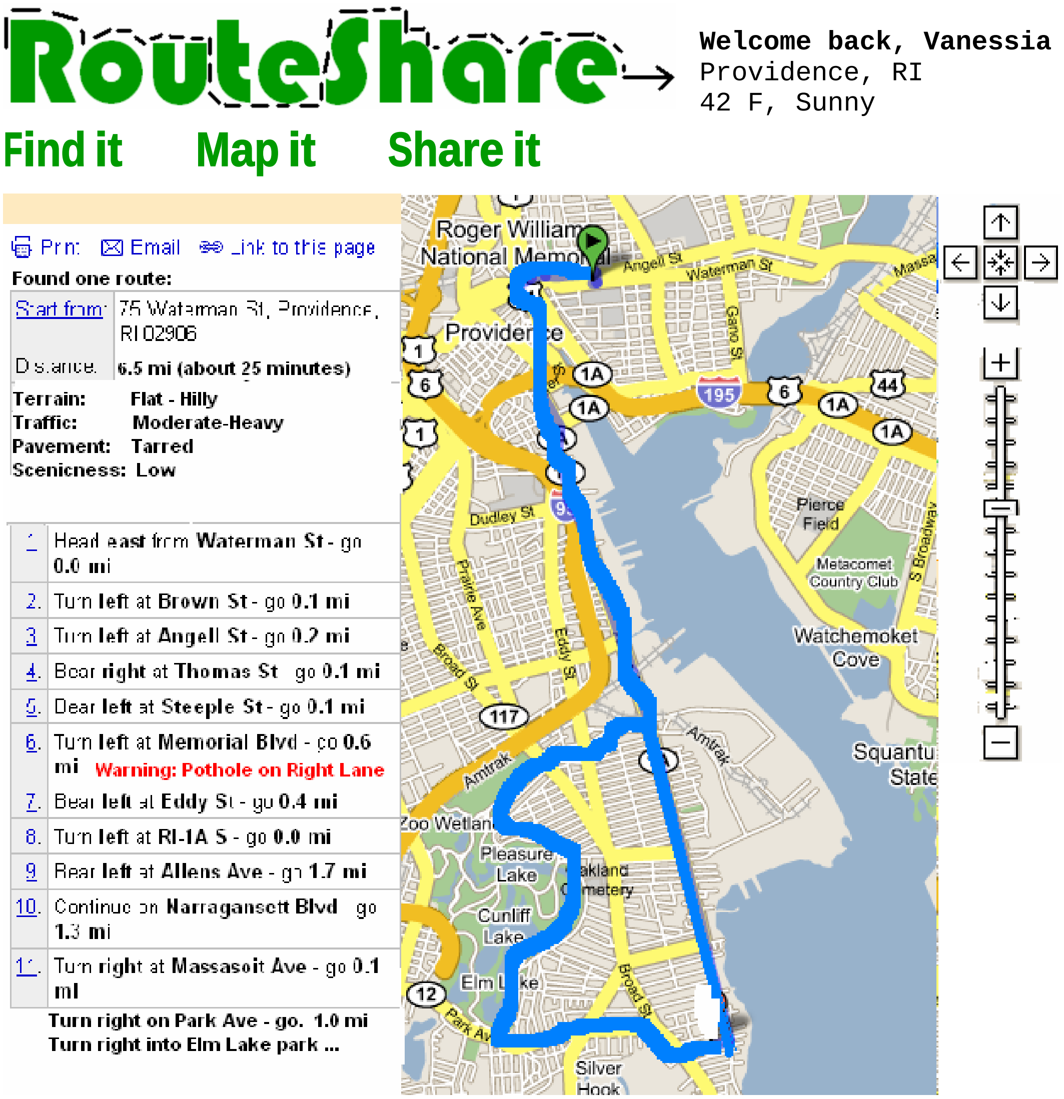
Start address: 75 Waterman St, Providence, RI
End address: 75 Massasoit Ave, Cranston, RI 02905

Start from:	75 Waterman St, Providence, RI 02906
Arrive at:	75 Massasoit Ave, Cranston, RI 02905
Distance:	4.5 mi (about 8 mins)

1.	Head east from **Waterman St** - go **0.0 mi**
2.	Turn **left** at **Brown St** - go **0.1 mi**
3.	Turn **left** at **Angell St** - go **0.2 mi**
4.	Bear **right** at **Thomas St** - go **0.1 mi**
5.	Bear **left** at **Steeple St** - go **0.1 mi**
6.	Turn **left** at **Memorial Blvd** - go **0.6 mi**
7.	Bear **left** at **Eddy St** - go **0.4 mi**
8.	Turn **left** at **RI-1A S** - go **0.0 mi**
9.	Bear **left** at **Allens Ave** - go **1.7 mi**
10.	Continue on **Narragansett Blvd** - go **1.3 mi**
11.	Turn **right** at **Massasoit Ave** - go **0.1 mi**

These directions are for planning purposes only. You may find that construction projects, traffic, or other events may cause road conditions to differ from the map results.

The Mapquest and the Google maps here demonstrate their driving maps' emphases on highways as a means of transportation and disregard for traffic, terrain, pavement conditions and other criteria that cyclists choose to select their bike route.



RouteShare

Welcome back, Vanessia
Providence, RI
42 F, Sunny

Find it Map it Share it

Print Email Link to this page

Found one route:

Start from:	75 Waterman St, Providence, RI 02906
Distance:	**6.5 mi (about 25 minutes)**

Terrain: Flat - Hilly
Traffic: Moderate-Heavy
Pavement: Tarred
Scenicness: Low

1. Head **east** from **Waterman St** - go **0.0 mi**
2. Turn **left** at **Brown St** - go **0.1 mi**
3. Turn **left** at **Angell St** - go **0.2 mi**
4. Bear **right** at **Thomas St** go **0.1 mi**
5. Bear **left** at **Steeple St** - go **0.1 mi**
6. Turn **left** at **Memorial Blvd** - go **0.6 mi** **Warning: Pothole on Right Lane**
7. Bear **left** at **Eddy St** - go **0.4 mi**
8. Turn **left** at **RI-1A S** - go **0.0 mi**
9. Bear **left** at **Allens Ave** - go **1.7 mi**
10. Continue on **Narragansett Blvd** go **1.3 mi**
11. Turn **right** at **Massasoit Ave** - go **0.1 mi**

Turn right on Park Ave - go. 1.0 mi
Turn right into Elm Lake park ...

The user interface for RouteShare will focus on providing useful “pre-ride” information. The route-mapping format part is not too different from driving maps except that it will only generate routes that are bike-friendly (i.e. no highways). Like Mapquest and Google maps (which was re-arranged here as demo), it will also show turn-by-turn instructions for the user. Both the map and the directions will be printable.

Non-functional requirements

Performance

- Query process needs to be efficient – the user is not going to wait all day before getting on the road.
- Java applet and graphics engine needs to be efficient as well.
- Should be able to handle a reasonable amount of Web traffic.

Testing

- Needs to be tested thoroughly for route accuracy (or at least no incorrect non-accurate routes).
- Needs a larger number of people to input test data.
- Needs to go out on the road to test these routes.

Reliability

- Must be absolutely reliable. One unreliable experience (wrong route that took user up to highway) could have detrimental results.

Ease of Use

- Should be intuitive to use. Users can be assumed to have experience with online maps, or if not, paper maps. Since we rely on user-input data and feedback, this is very important that more people use it often.

Portability

- The program as a whole is not very super portable as it relies on a central database.

Documentation

- Documentation will talk about how to use the different features of the Web site, but hopefully they it will be intuitive enough because unlike games, people just want quick information and do not want to have to read the documentation.

Dependencies on other systems

- Dependent on user’s machines if we would use Java applets.

Updated requirements

In each section, from the highest priority to the least.

DATABASE
- A database table of bike routes (lists of road segments)
- A database table of route segments with information on each segment's
    - o Distance
    - o Terrain (numerical scale of 1-5. 1 is flat, 3 is rolling and 5 the most hilly, but displayed to users as descriptive scale, i.e. flat, not 1)
    - o Traffic conditions (same as above. Use 1-3 but displays as light, heavy, etc)
    - o Pavement (same as above. Use numerical scale but displays as tarred, gravel, etc)
    - o Scenic-ness (optional parameter)

SEARCHING
- A search engine that searches bike routes by a combination of the above criteria. If search yields no result, return the closest results and rank them. Display the map of the highest ranking route.
- Ability to present the search result by a creating a printable map of the route.
- Ability to present the search result by detailed route information at each turn (could also implement it such that it highlights each significant change of terrain, traffic conditions or pavement conditions).

INPUT
- Ability to allow users to easily input a route by using their mouse to trace or click on a map.
- Ability to allow users to rank each route segment by one of the above criteria.
- Must have an easy to use, simple, non-cluttered interface.
- Ability for users to flag any segment of any route to alert each other of emergency situations.
- Ability for the administrator to manually add, edit and remove routes.

ROUTE GENERATION
- Ability to generate new routes by joining a new segment to an already established route nearby.
- If there is no established route nearby, the program should be able to generate a new route based on user's weightings of the different criteria and distance.

OTHER FEATURES
- Have a weather icon which shows the weather of the bike route that the cyclist has chosen.
- The home page shows the most popular bike routes, asks users to vote on the worst hills and other trivia questions designed for sharing useful knowledge on biking in the locale.

Risky Parts

Some of the greatest threats to using this program for cs190:

- With the new route generation function, the project is too big and complex to be completed in the amount of time given; yet without which, it might just be a large database and not challenging enough for CS190.

- Too dependent on users for route data input and rating – part of the success of the program will be out of our control due to the possible variability in the quality and accuracy of user input.

- Project is not very divisible – there are not many atomic parts for a large group of people to work on at the same time. The structure of the program has few components but each is highly complex.

- The reliability of the program is very important, yet it will be difficult to test this program unless we or our cyclist friends go out and test the routes themselves. This is possible but takes time and external manpower.

- There could be were a billion ways to generate a 10-mile route from Faunce Arch if no data is available initially.

- Obtaining map data and translating the map data into something we can work with. We need to look for a free or low-cost GIS program and data.