BikeQuest Specifications

Danielle L. Karr

February 10, 2005

1. Project Introduction

Premise

More and more cars and trucks hit the roads every day. Cities and states focus all of their attention on improving major highways, while a growing minority of bikers goes unnoticed and discouraged. The most significant problem with this is that paths for bikers to get where they need to go already exist—but it's too hard to find them. Current mapping technology doesn't allow for the restrictions biking places on those paths or for feedback on biking related issues.

Project

Starting with routes in the state of Rhode Island, BikeQuest will provide a resource for searching for bike-appropriate routes from place to place. Through a combination of MapQuest-like route-finding, initial route ranking, and user-feedback ranking, the program will solve the problem of getting a biker from point A to point B, as well as along a recreational route, safely and comfortably. The program would be ideally web-based.

Users

The potential users of this program—people who ride bikes in Rhode Island—would include a wide variety of target audiences. More specifically, this includes cycling clubs and teams, like the Brown Cycling Team, as well as Brown employees that participate in Bike2Brown and strictly recreational riders. Department members Mark Dietrich and Spike Hughes have also expressed interest in the project. The following photos from *conduit!* in the summer of 2003 show the number of department members that enjoy biking. Obviously, if this many people that work in a single department are bikers, then it's likely that a significant user base exists for this project.

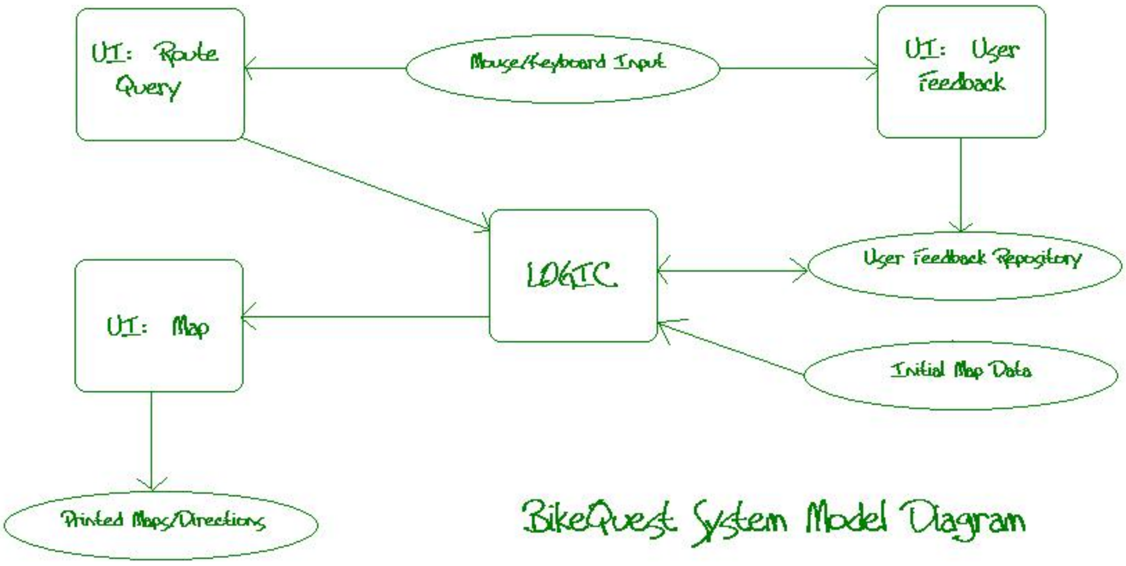
Biking enthusiasts Shriram Krishnamurthi and Mark Dieterich co-organized some great rides on the East Bay bike path over the summer. Students, staff and faculty participated in the 20-mile rides.The start of one is pictured below – Michael Black caught up with them later...

Said Shriram, "We took the photo at Del's because it was unimpeachable evidence that we had indeed ridden at least ten miles each way!"
l to r: Hulya Yalcon, Shriram, David Laidlaw, Tomer Moscovich, Guillaume Marceau, Liz Marai, Thomas Hofmann, Peter Sibley, Mark Dieterich and Michael Black.

Giving users the choice between searching for a direct path from point A to point B or for a more recreational route that loops back to the origin should accommodate all of these users.

2. System Specifications

System Model Diagram

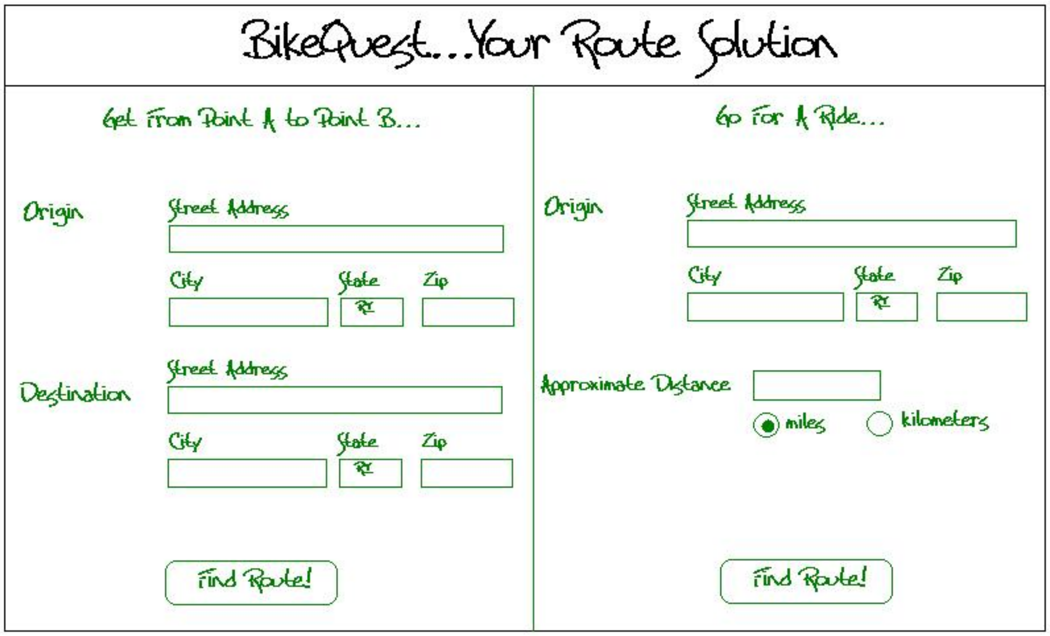


System Component Description

- User Interface: Route Query
  - Provides options to search for unidirectional route given its endpoints or loop route given an origin and a distance
  - Gathers information from keyboard via text boxes
  - Initiates searching via the “Find Route!” button
- User Interface: Map
  - Displays map and directions for following outputted route
  - Allows for map manipulation
  - Shows links to user feedback for roads used in route
- User Interface: User Feedback
  - Allows for collection of user feedback via simple form
    - Check specific problems (dogs, potholes, etc.)
    - Check terrain (hilly, flat, etc.)
    - Type personal comments
    - Select overall numeric rating
  - Initiatiates submission via the “Submit” button
- Logic
  - Provides algorithms necessary for efficient path finding
  - Uses User Feedback and Initial Map Data (in some algorithm proportion) to find reasonable routes for biking given basic traffic maps
- Mouse/Keyboard Input
  - Enter search criteria and rankings into UIs through the mouse and keyboard
- Initial Map Data
  - Includes map data supplied by MapQuest, RIGIS, and GIS
  - Includes Basic rankings (i.e. “worst” ratings on I-95, “best” on an actual bike path)
- User Feedback Repository
  - Gathers and stores user feedback
  - Provides user feedback to the Logic when needed to inform a path-finding algorithm

3. User Interface

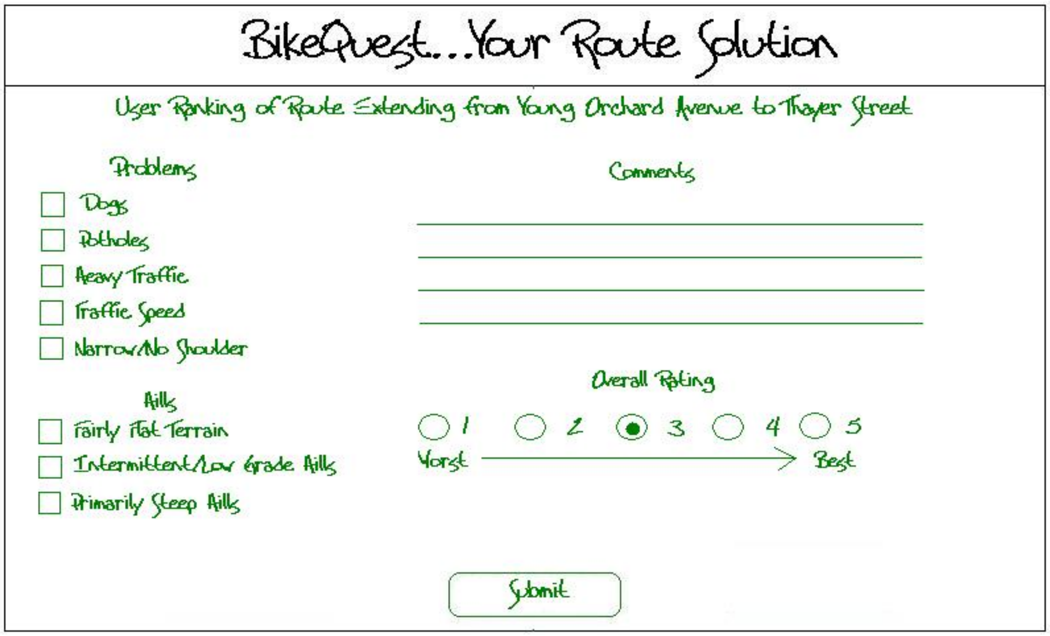
User Interface Diagram: Route Query



User Interface Component Description: Route Query

- Get From Point A to Point B…
  - Includes text boxes for user to input origin and destination addresses
  - Includes “Find Route!” button to initiate the path-finding process incorporating the origin and destination
- Go For A Ride…
  - Includes text boxes for user to input origin and approximate desired length of bike ride, available in miles or kilometers
  - Inclues “Find Route!” button to initiate the process to find a loop of the desired length from and to the indicated origin

User Interface Diagram: User Feedback



BikeQuest...Your Route Solution

User Ranking of Route Extending from Young Orchard Avenue to Thayer Street

Problems

- □ Dogs
- □ Potholes
- □ Heavy Traffic
- □ Traffic Speed
- □ Narrow/No Shoulder

Hills

- □ Fairly Flat Terrain
- □ Intermittent/Low Grade Hills
- □ Primarily Steep Hills

Comments

Overall Rating

○ 1 ○ 2 ● 3 ○ 4 ○ 5

Worst → Best

Submit

User Interface Component Description: User Feedback

- Problems
  - Provides user with opportunity to report on common problems with particular route
  - Allows user to check all the check boxes that apply
- Hills
  - Provides user with opportunity to report on the terrain of a given road in his/her route
  - Allows user to check all the check boxes that apply
- Comments
  - Provides user with a space to comment on any facet of the route they found relevant to the riding experience
  - Allows user to fill in space provided
  - Will not apply directly to route rating (and therefore won't affect routes returned by path-finding algorithm), but still very helpful to riders
- Overall Rating
  - Provides user with a space to rate their overall experience on a given road
  - Will most directly affect the ranking of a section of a route

4. Non-Functional Requirements

Performance

As this program will be web-based, the computing overhead will need to be minimized to compensate for all types of internet connections. Graphics should be streamlined, and the path-finding algorithms will need to be speedy.

Testing

Testing will be somewhat multi-part. The actual path-finding algorithm could be tested fairly easily on a map that only includes initial valuations for bike travel. The user feedback system is basically just a database, which can certainly be tested alone. Testing of the ranking system, which combines initial road values as well as user feedback, will require both aspects of the system to be up and running. Accordingly, some testing dependencies will inevitably exist.

This ignores the concept of biker testing. To ascertain that the path-finding and ranking systems are functional, and that the initial values the coders give certain roads are correct, people will actually have to bike the returned paths.

Reliability

In order to gain and maintain a user base, BikeQuest will need to be reliable. If the system crashes while a particular user is trying to provide feedback, s/he isn't likely to start again. If the system isn't reliable, users might find it less frustrating to find a new route through trial and error. BikeQuest will be dependent on its users to work well, and it will only have users if it's reliable.

Ease of Use

Given the wide variety of technological skills, BikeQuest will need to be intuitive and simple to navigate.

Portability

As BikeQuest is a web-based application, it will need to function on all major web browsers (Internet Explorer, Opera, Mozilla, Fire Fox, etc.).

Hardware/Software

BikeQuest will require at least one node to run off of at all times, and may need SQL or a similar piece of database software to keep track off available routes and their rankings.

Documentation

Documentation will need to be similar to any other web application's documentation. It will need the basic how-to-use instructions and tips for best use. It should also credit the sources of map data and explain, generally, how the parts of the program fit together to produce the bike routes.

Dependencies

One of BikeQuest's internal dependencies is that the path-finding algorithm is dependent on user feedback and initial rankings to work properly. It will also be dependent on outside software for its database, and on external data from Map Quest, GIS, and RIGIS. The project would also be dependent on Zach Schubert and his help with the path-finding algorithm itself.

5. Functionality Requirements

Basic Features

- Bike Route Planning
    - Point A to Point B
    - Loop of a Given Distance
- Numeric System to Rank Sections of Routes
    - Partially attributable to initial rankings
    - Mostly attributable to user feedback
- Mechanism for ranking user feedback
- Means to print out a route so that it will be easily read while riding

Optional Features

- On-line Account with Saving/Loading of Preferred Routes
- Message Boards/Chat Capability
- Bike Traffic Map
    - Indicates bike traffic in a given area (i.e. color-coded map that shows the quantity of route searches in a given area

- Including waypoints in a bike route (i.e. find a path that hits multiple destinations)

6. Additional Resources

In addition to Mark Dietrich and Spike Hughes, Zach Schubert has expressed interest in the project. He is currently working with Amy Greenwald on an improved Map Quest system using autonomous agents, and he would apparently be willing to discuss the use of feedback and algorithms for route planning.

Also, Howard Stone is a catalog librarian at Brown and the author of three books on good bike routes in New England. If the project were to proceed, he would be an excellent source of information on determining good bike routes.

7. Risks

- Dependent on user feedback to be really efficient/useful
- Dependent on outside map data, which will require time to interpret and parse
- Testing components that are dependent on each other (route ranking and path-finding)
- Dependent on people from outside the class for assistance with path-finding algorithm, which is a rather key component of the project