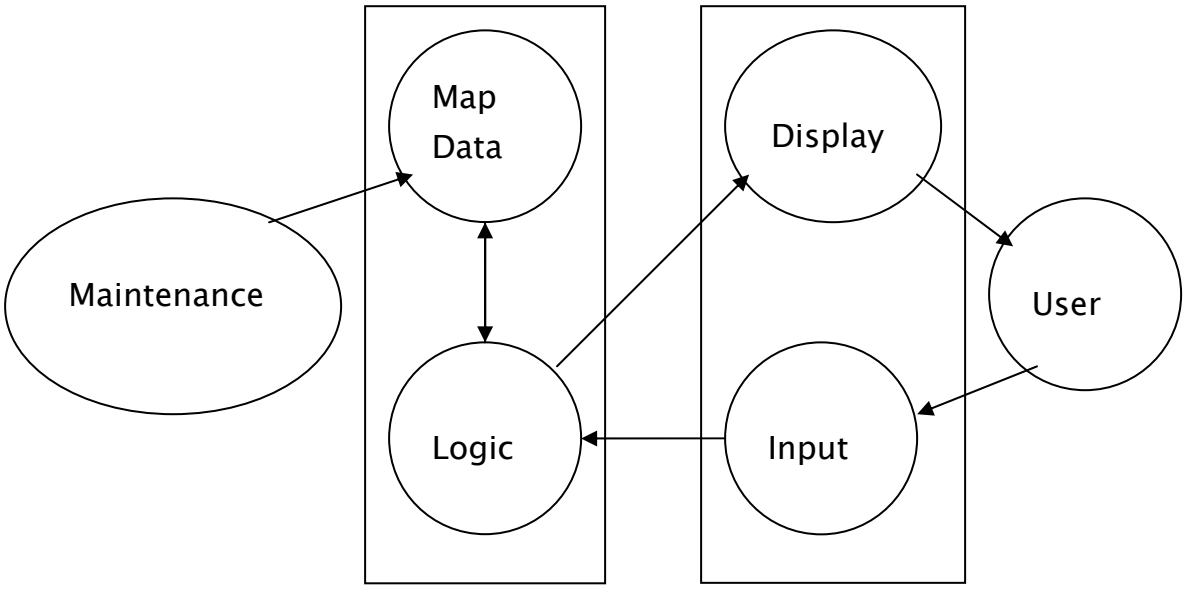
BikeQuest Specifications

Aldo Avila
<aavila>

Description:

Not all Roads are made to accommodate bicyclists. Some roads have heavy traffic, others don't allow bicyclists to ride on them at all, and some roads just have really bad terrain that is inconvenient for the riders. Due to these problems, bicyclists are having a difficult time trying to figure out routes to get to their destinations; be it a completely different place(commuting) or where their ride starts(looping).
We need to implement a program that will give bicyclists a route to their destination. This route will not only be bike-friendly, but it will also accommodate the preferences of the rider. If the rider wants a hilly and scenic route with minimal traffic, the program should be able to generate the best route that fits those preferences.

System Model Diagram:



Description of Diagram:

User- The user should basically be able to sit at his computer and use a GUI that will be very intuitive. He/She will be able to type input into the GUI such as their destination, preferences for routes, or to give us user feedback on certain routes.

Input- The input part of the front end will obviously be able to determine what type of input is being put in. Is the user requesting a route, or is the user simply rating a route that was already used? We will be able to determine what type of input is being put in and will be able to tell the logic what it is the user wants.

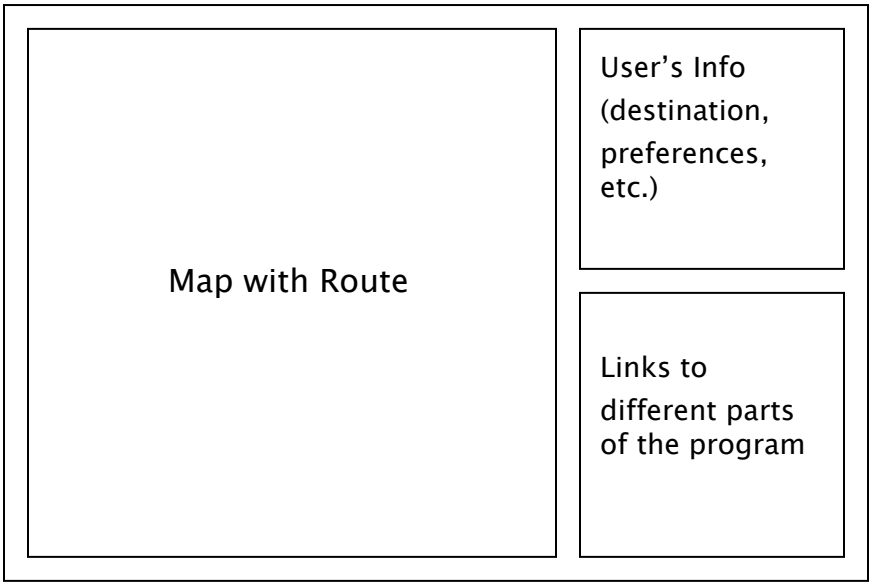
Display- The display should basically be very easy to use and read. It will receive information from the logic as to what it should be displaying. Once the logic gives it a route, it should show clean and clear maps with the route outlined. I will go into a little more detail as to what the User interface should look like in a bit.

Logic- The logic will use the information from the input to perform the action that it needs to do. If the user is simply ranking routes, it will go into the map data and rank the streets accordingly. If the user is asking for a route, it will retrieve all the information from the map data to create a route according to the preferences of the user. It will then get this route and feed it to the display so that it can be shown to the user.

Map Data- This part of the system will basically serve as a library. It will hold all the information about the maps. It will have all the different roads and all their rankings (toughness, scenic, traffic, etc). It will be able to give this information to the logic every time it asks for it and the logic should also be able to update it, as well as maintenance.

Maintenance- Every once in a while, someone will have to go into the library of roads and make sure that roads that are unavailable are actually not being used by the routing algorithm. Also, roads may sometimes be added to the map, which will mean that someone will have to go in and update the library. It's not something that will be used a lot, but there still needs to be a way for someone to have access to the map data to be able to update it, which will be the maintenance person's job.

UI Diagram:



Description of UI Diagram:

-The main component of the UI is going to be our ability to display the map and the route in a clean and clear manner. Most of the Display should be used to show the map.
-The user's information should also be displayed once it has been typed in and the logic has generated a map. We should also be allowed to let the user edit the information from this window, in case the route suggested isn't to the user's liking.
-There must also be links to the different parts of the program from this main window. The links could take you to a window where you can rate the route you were given. They might also be able to take you to other routes that might suite the user's taste (like Amazon.com does with books), or maybe even to our "create-a-route" option, if we have enough time to implement it.

Non-Functional Requirements:

Performance- The logic should be able to come up with a route pretty quickly after the user asks for one. It shouldn't have to take 5 minutes to generate a route to fit the user's taste.

Testing- Aside from making sure that the actual program doesn't crash and generates routes, we need to make sure that the routes that are being generated are actually correct. If the user wants a scenic route with minimal traffic, the program should not be generating something that goes through downtown. The only way we might actually be able to test this when the problem isn't as obvious, is to actually have someone ride the route.

Reliability- The program should really only have maintenance done once a week to make sure that all the roads that are unavailable aren't being used. It shouldn't be braking down on its own too much.

Easy to use- We need a very intuitive GUI to be able to compensate for the bicyclists that are not too computer literate.

Documentation- No documentation is actually really needed. Just a very simple Read Me should do the trick, since the user really only needs to know how to look up a route, which the GUI should make very obvious how to do.

Updated Requirements:

High priority = *** Mid priority = ** Low priority = *

Basic Features-

-Data of all possible bike roads ***
-Ranking parameters for roads ***
-Determine routes for commuting ***
-Determine routes for looping ***
-User input on roads **
-User friendly GUI ***
-Printable Maps **

Optional Features-

-Emergency notices *
-Emergency routes *
-Weather *
-Recommendation of routes *
-Create-a-route *

Risks:

There are a couple risks that come with trying to create this program. The biggest risk is actually not being able to obtain the map data. It might also be pretty difficult to create a useful database once we actually get the map data. Another huge risk will be testing that the routes actually are chosen according to the rider's preference. Like I stated earlier, the only way to actually test this is to actually go out and ride the routes. Also, with all the bells and whistles, this program has the potential to be a huge project and we might not have enough time to finish. The last big risk that I can see coming is actually getting user feedback. If we don't obtain user feedback, we might not be able to keep the maps updated. If a user doesn't let us know that a road suddenly has a huge pot hole, we might still send riders down that road which will be a big problem. We will need user feedback to really make this program work.