Project Proposal

Robert Konell, Blake Wellington

February 17, 2014

Summary

The application will use data from Tri-Met's database. (Tri-Met is a the public transit system in Portland, Oregon). The database contains route information for all of Tri-Met's trains and buses.

This proposal implies a joint project to be developed by Robert Konell and Blake Wellington. We recognize that the intent of the CS457 project is to have student work individually. However, we believe that, by working together, we can accomplish a greater level of sophistication in the end product. Each student has skills and interests that can be leveraged to complete a more robust application with a richer user experience.

Since we are new to Haskell and have limited time to work on this project, we have broken the project down into manageable, intermediate goals.

Project Goals

The aim of this project is to use the Haskell programming language to create a web-based application. It is intended as a learning exercise with the following project goals:

- Learn functional programming with Haskell
- Learn how to use a REST API as a web service
- Learn how to use Haskell libraries to present the data in a meaningful way

Initial Goal

The initial goal of the project is to successfully pull a single, small piece of information from the database, massage it in Haskell, and present it on a web page. This may be something as basic as showing the next bus/train to arrive at a given stop number.

Secondary Goal

Once the initial goal is in place, we will expand the user interface to contain ever more complex query abilities. For example, we might add the ability to select a certain route, list the stops of a route, estimate the travel time between two stops, plan a trip (including multiple transfers), etc.

Final Goal

Since this is a web-based application, it would be nice to have it hosted on a public server. The final goal is to be able to pack all the code up and install it on a server in the cloud (such as Amazon EC2). This will involve installing Haskell (or just shipping the compiled version) on a server equipped with Apache web server.

Project Motivation

The authors of this application are Computer Science students at Portland State University. We are relatively new to programming and would like to gain experience in building applications of greater complexity than those developed for small academic projects. As part of our requirement to use functional programming, we will use Haskell and associated libraries to create an integrated piece of software with 'real world'-like behavior.

This includes using graphical libraries and/or HTML wrapper libraries for the front end, and calls to an exposed web API for the backend. The final user interface will be presented on a web page using HTML (and possibly some Javascript).

Collaboration Plan

A git repository hosted on Github (https://github.com/blakewellington/cs457project) will be used. One benefit of using Github is that it will provide a public way for each student's work to be individually tracked.

In the first week a high-level architecture will be designed, with modules and their sub-tasks identified and broken down into manageable chunks. Each student will take ownership of certain tasks, but not entire modules, that way both students will have a hand in every part of the project.

Weekly in person meetings will be conducted to discuss progress, share advice, dicuss difficulties, etc. Frequent communication through online chat programs will be used as well. Occasional pair programming will be utilized as time allows.

Schedule:

Week of Feb 17th:

- High level design, discussion of major modules, identification of bite sized tasks.
- Research and analyze Trimet web API.
- Research and analyze web server options.

Week of Feb 24th:

- Functions to call Trimet API created.
- Work begins to parse responses, including design of data types to hold parsed data.
- Web server option chosen, work begins on having server accept GET and response with Haskell generated minimal HTTP page.

Week of March 3rd:

- Parser is working.
- A complete cycle of: HTTP Get, haskell calls API, parse response, generate page and respond
- For at least 1 API function.
- Work begins on designing the front end more with a complete cycle complete.

Week of March 10th:

- More functions created for more complex queries.
- Heavy work on the front end/HTML in order to make it robust enough to use backend.
- Begin cleaning up code by end of week.
- Work begins on final paper.

Week of March 17th:

- Final touches are made. Paper turned in. Celebration ensues.