# Table of Contents

List of Figures .................................................................................................................................................. iii

1.0 Introduction ................................................................................................................................................ 1
  1.1 Overview of this Document .................................................................................................................... 1
  1.2 Purpose of the project ............................................................................................................................ 1
  1.3 Audience .................................................................................................................................................. 1
  1.4 Glossary .................................................................................................................................................. 1
    Technology-based definitions: .................................................................................................................. 1
    Domain-based definitions: ....................................................................................................................... 2
  1.5 References ............................................................................................................................................. 2

2.0 Overall Description ...................................................................................................................................... 2
  2.1 System Architecture .............................................................................................................................. 2
  2.2 Module Descriptions ............................................................................................................................ 3
    Draw Tool ............................................................................................................................................... 3
    Elevation Graph ..................................................................................................................................... 3
    Weather Module .................................................................................................................................... 4
    RSS feed module .................................................................................................................................... 4
    Article Module ....................................................................................................................................... 5
    User Management ................................................................................................................................... 6

3.0 Complete User Interface Description (as of version 1.0) ........................................................................ 6

4.0 Implementation Remarks ......................................................................................................................... 8

5.0 Appendices ................................................................................................................................................ 9
  ER Diagram ............................................................................................................................................... 9
  Data Dictionary ........................................................................................................................................ 9
  Algorithms ................................................................................................................................................ 11
List of Figures

Figure 1 -- System Architecture (Diagram by Jeff Stephens) ................................................................. 2
Figure 2 -- DrawTool Sub-architecture ................................................................................................. 3
Figure 3 -- Elevation Data Sub-architecture ......................................................................................... 3
Figure 4 -- Weather Data Sub-architecture ......................................................................................... 4
Figure 5 -- RSS Data Sub-architecture ............................................................................................... 4
Figure 6 -- Review Data Sub-architecture ........................................................................................... 5
Figure 7 -- User Management Sub-architecture .................................................................................. 6
Figure 8 -- Start-Up Dialog .................................................................................................................. 7
Figure 9 -- User Interface ..................................................................................................................... 8
Figure 10 -- ER Diagram (Diagram by Jeff Stephens) ........................................................................ 9
Figure 11: Elevation Profile Algorithm (Diagram by Jeff Stephens) ..................................................... 11
1.0 Introduction

1.1 Overview of this Document

The purpose of this document is to summarize the development of the Hiker’s Software Toolkit project. It is meant to be a historical record and self-assessment of the process and outcome.

1.2 Purpose of the project

The Hiker’s Software Toolkit’s purpose is to provide a useful interface for anyone planning a hiking trip. There are modules that: Use Google Maps to allow the user to draw their path on a topographic map; View an Elevation Profile of the path; View the current weather; and view and respond to articles about specific hikes.

1.3. Audience

The audience will be anyone who needs to plan a hike. It can be as short as a day hike of a couple of miles or as long as a hike across a continent.

1.4. Glossary

Technology-based definitions:

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ajax</td>
<td>A style of web programming that utilizes the XHR to create a more robust and user friendly site.</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheet -- A scripting language that can manipulate the view layer of one or many web pages on a web site.</td>
</tr>
<tr>
<td>DOM</td>
<td>Document Object Model -- A representation of a tag-based document (Html or XML) in a tree form that can be traversed and manipulated.</td>
</tr>
<tr>
<td>Google Maps</td>
<td>A service provided by Google that displays information on a street map, topographic map or satellite image.</td>
</tr>
<tr>
<td>JavaScript</td>
<td>A client-side scripting language that allows Html updates without server call backs and has many other capabilities.</td>
</tr>
<tr>
<td>MVC architecture</td>
<td>Model-View-Controller design philosophy describes a scenario in which the data layer, business rule layer and user interface layer are treated as distinct sections of a large software project.</td>
</tr>
<tr>
<td>MySql</td>
<td>A popular multi-user relational database system.</td>
</tr>
<tr>
<td>PHP</td>
<td>A server-side scripting language that allows for dynamic web applications.</td>
</tr>
<tr>
<td>RSS</td>
<td>Really Simple Syndication -- A standardized and specialized XML format that marks up published articles so that they can be located and parsed by specialized RSS readers.</td>
</tr>
<tr>
<td>Web Service</td>
<td>Provides an interface between multiple web-based applications.</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language -- A customizable tag-based language that commonly used to pass data between applications.</td>
</tr>
<tr>
<td>XHR</td>
<td>XmlHttpRequest -- Object used by many scripting languages that transfers text, including content, Html, XML or JSON documents, between a server and a browser.</td>
</tr>
</tbody>
</table>
Domain-based definitions:

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOAA</td>
<td>National Oceanic and Atmospheric Administration -- Provides a web service for gathering weather data</td>
</tr>
<tr>
<td>USGS</td>
<td>US Geological Survey -- Provides a web service that gives elevation data based on latitude and longitude.</td>
</tr>
<tr>
<td>Topographic Map</td>
<td>A specialized map that shows terrain information, most importantly elevation data in the form of contour lines.</td>
</tr>
</tbody>
</table>

*Italicics:* Definitions are based on Wikipedia (http://en.wikipedia.org)

*Bold:* Definitions are based on W3C (http://www.w3.org/)

1.5. References


2.0. Overall Description

2.1 System Architecture

The diagram above shows the overall system architecture, which follows the Model-View-Controller methodology. Real data is held in the units on the right side of the diagram. Data is fed to the units to their left. Those unit handle and process user input. Then next level to
right has the modules, which display the information and the main UI on the extreme left are frames that hold the modules. For data requests just reverse the arrows. Note that for the bottom portion of the diagram the corresponding modules are not in version 1.0.

### 2.2 Module Descriptions

#### Draw Tool

The Draw Tool module allows a user to draw their intended route as a poly line a Google Map window.

#### Elevation Graph

The Elevation Graph Module is a profile view of the path the user entered in using the Draw Tool. The graph view is done with the help of the PhpGraphLibrary from http://www.ebrueggeman.com/phpgraphlib/index.php
This module shows current weather data in the user’s location. The data is retrieved and displayed with the help of http://www.ebrueggeman.com/phpweatherlib/index.php
The RSS module shows available related RSS feeds. This is not part of version 1.0 but the intent is to allow the user to subscribe, retrieve and unsubscribe to RSS feeds. The subscription information will be kept in the server-hosted database.

**Article Module**

![Diagram of Article Module]

Figure 6 -- Review Data Sub-architecture

This module shows pre-written article on specific hikes in Western United States. The review feature, which will be in a future version, will allow the user to submit their reviews submitting comment and rating the hike based on scenery, solitude, trail condition and weather conditions. Aggregated data will be shown when the user returns to that page.
The user management system allows the user to save data to a server–hosted database. As of version 1.0 persistent data consists of a file that holds an Xml files with the most current map settings.

3.0. **Complete User Interface Description (as of version 1.0)**

Hiker’s Software Toolkit is a web-based application. To use it go to [http://www.ecst.csuchico.edu/~jeffst/main.php](http://www.ecst.csuchico.edu/~jeffst/main.php). A dialog will pop up that requests a starting location. The system will attempt to geocode the response using Google Map’s geocoding algorithm. If successful the map will be centered on the requested location. If not successful or the user clicks ‘Cancel’, the map will be centered in Chico, CA. The main user module is now available to use.
The modules on the right side have limited functionality. The middle of the right side is a placeholder for the up-coming RSS feed module. The bottom-right frame has a list of articles. Click on a link to read the article. The list is region based and the region is determined by the current center location of the Google Maps window. The user may notice a change in the list as he/she pan the map around. Also, the Weather Module in the top right frame shows a current weather report.

The most useful part of the toolkit is the Draw Tool. With this the user can draw on the map by left-clicking in the map window. When the user clicks on two points to create a line it is ready for an elevation profile graph to be created, which shows up in the lower left window when the user clicks ‘Gen. Elevation’. The graph also includes calculations for the maximum, minimum and average elevation, path length and path length account for elevation (see Algorithms in the appendix )

Also with the Draw Tool is the ability to clear lines one by one in reverse order in which they were added or clear all lines and start over. Also the ‘Save’ and ‘Retrieve’ features work by save the need map and polyline data to recreate the map in another window. Finally, there are text fields below the map that show the latitude and longitude at the center of the map and the elevation at that spot. The second of the two uses Ajax to communicate with a server-side script that queries a web service for the elevation data.

All of the modules sit in a set of frames with a scrollable feature and resizable windows too accommodate different sizes. Since it uses frames, the toolkit will only be test on newer browsers, like Firefox, IE and Chrome.
4.0 Implementation Remarks

As it turns out, the more difficult parts of this project were not the utilization of Ajax and Web Services. Where these technologies were required, it was accomplished with just a little bit of research. The biggest stumbling block by far came from passing data between each modules. While using the toolkit, the user may notice that some of the modules will react to events in another module. For instance, the region-based listing of the articles will change when the user moves out of a system-defined region and into another. For this I had to capture a movement event in the Google Maps window, grab the new map location and decide which list to reload the frame with.

In most cases I would pass the data on the Url. and use Php’s ‘Get’ method to retrieve it. Far move difficult than the above example was passing points from the Draw Tool module to the Elevation Graph. Data had to be passed using a set of ordered pairs, then carefully parsed and each point coupled with elevation data before being passed to the graph generator. The points go through about six different arrays in the process. The save/retrieve feature was another difficult phase. It required that an Xml file be built from scratch and programatically, then parsed again to get the map and polyline back into Google Maps windows. Calculating the various statistics was somewhat challenging, but having the equations ready to go at implementation time was a help.

Through this project I learned most of what I wanted to learn: Xml parsing, data transfer and web services. Existing libraries made the Xml related items easier to handle but everything was done myself. I will continue working on advancing this website as the number of features I could add is seemingly infinite.
5.0 Appendixes

**ER Diagram**

![ER Diagram](image)

*Figure 10 -- ER Diagram (Diagram by Jeff Stephens)*

**Data Dictionary**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table: Hikes</strong></td>
<td></td>
</tr>
<tr>
<td>name (pk)</td>
<td>char(20)</td>
</tr>
<tr>
<td>trailhead_address</td>
<td>char(50)</td>
</tr>
<tr>
<td>trailhead_latitude</td>
<td>float(10,6)</td>
</tr>
<tr>
<td>trailhead_longitude</td>
<td>float(10,6)</td>
</tr>
<tr>
<td>managed_by</td>
<td>enum('BLM','NPS','USFS','Private','Other')</td>
</tr>
<tr>
<td>trailhead_elevation_ft</td>
<td>smallint(6)</td>
</tr>
<tr>
<td>is_loop</td>
<td>tinyint(1)</td>
</tr>
</tbody>
</table>

| **Table: Users**              |                       |
| user_id (pk)                  | int(10) unsigned      |
| first_name                    | char(20)              |
| last_name                     | char(20)              |
| username                      | char(12)              |
password | varchar(40)
home | char(50)
email | char(50)
isActive | tinyint(1)

**Table: HasReviewed**
- fk_user_id(pk) | int(10) unsigned
- fk_hike_name(pk) | char(20)
date_started | date
hours_to_complete | float(4,1)
rating_overall | tinyint(4)
rating_scenery | tinyint(4)
rating_challenge | tinyint(4)
rating_solitude | tinyint(4)
rating_trail_condition | tinyint(4)
rating_weather | tinyint(4)
rating_comments | text

**Table: RSSChannels**
- Name(pk) | char(20)
- URI | text
- Metadata | XML

**Table: RSSFeeds**
- URI(pk) | text
- Metadata | XML
- fk_channel | char(20)

**Table: HasSubscribedTo**
- fk_channel(pk) | char(20)
- fk_user_id(pk) | int(10) unsigned
- isActive | tinyint(1)

**Table: Routes**
- fk_user_id(pk) | int(10) unsigned
- route_name(pk) | char(20)
- route_points | XML
- last_modified | datetime
**Algorithms**

1. Elevation Gain/Loss from trailhead for non-loop trails
   Difference = Starting elevation – Ending elevation.

2. Average elevation
   \((\text{Sum}(s_1, s_2, \ldots, s_n)/n)\) where \(s\) is sampled from the set of elevations at a set distance apart and \(n\) is the number of data points.

3. Steepest grade
   Max(abs\((s_i-s_{i+1})\) for every \(i\)) to get the greatest rise or fall over a small number of intervals. To calculated the actual angle: \(\text{Angle } a = \tan^{-1} b = \text{(change in elevation/width of interval)}\)

4. Path length (not accounting for elevation)
   \(\text{Sum (Length of all polylines obtained from the Draw Tool)}\)

5. Path length (accounting for elevation)
   Formally, \(\text{Length} = \text{Sum} (h_1, h_2, \ldots, h_n)\) where \(h\) is from the set of hypotenuses from the geometries that make up the profile graph. For a single hypotenuse \(h\), \(h = \sqrt{a^2 + b^2}\) where \(a\) is the fixed interval and \(b\) is the change in elevation over that interval, as shown in Figure 7. The optimal interval width will be determined through experimentation.

---

**Sample Elevation Profile**

![Sample Elevation Profile](image)

**Figure 11 -- Elevation Profile Algorithm (Diagram by Jeff Stephens)**