System Design

NCDOT
SpeedInfo Website

PBS&J Project

17 August 2006

PBS&J

Document Control Panel	
File Name:	NCDOT Website System Design
Version No.:	Draft-1
Created By:	John Hope
Date Created:	March 31, 2006
Version No.:	Draft-2
Updated By:	John Hope
Date Updated:	April 25, 2006
Version No.:	1
Updated By:	John Hope
Date Updated:	August 17, 2006
Reviewed By:	
Approved By:	

Table of Contents

1.0 System Overview 5
2.0 System Architecture 5
2.1 System Architecture Diagram 5
2.2 NCDOT SpeedInfo Databases 6
2.2.1 Configuration Database 6
2.2.2 Archive Data Database 7
3.0 System Interfaces 8
3.1 SpeedInfo Data Server to Java Plug-in 8
3.2 Java Plug-in to NCDOT SpeedInfo Website Data Provider 9
3.3 NCDOT SpeedInfo Website Data Provider to NCDOT SpeedInfo Website 10
3.4 NCDOT SpeedInfo Website Data Provider to NCDOT TIMS 11
4.0 Website Interface 12
4.1 Website Map 12
4.2 Website Table View Page 15
4.3 Login Page 16
4.4 Configuration Menu Page 17
4.5 Sensor Configuration Page 17
4.6 Speed Threshold Configuration Page 18
4.7 User Configuration Page 19
4.8 Archive Data Page 20
5.0 NCDOT SpeedInfo Website Data Provider 24
5.1 Application Overview 24
5.2 Graphical User Interface 24
5.3 Configuration 25
5.4 Alerting Functionality 27

List of Appendices

Appendix A: SpeedInfo Data Feed Specification

Appendix B: NCDOT TIMS Web Service Interface

Appendix C: Hosting Servers Statistics and Configuration

1.0 System Overview

SpeedInfo uses radar technology to determine estimated speeds from 20 speed sensor sites on a portion of I-40 from mile mark 270 to mile mark 297 and 20 speed sensor sites on a portion of I-95 from mile mark 81 to mile mark 180 in North Carolina. Each site contains bi-directional sensors that read traffic speeds for eastbound and westbound traffic on I-40 and northbound and southbound traffic on I-95. PBS&J connects to the SpeedInfo Data Server to receive speed updates once per minute and post speed data to the North Carolina Department of Transportation (NCDOT) SpeedInfo website. The website is accessible via the internet using a standard internet browser at the following URL: http://www.ncdot-traffic.com/

2.0 System Architecture

2.1 System Architecture Diagram

Figure 1 is the system architecture of the NCDOT SpeedInfo website and associated data providers.

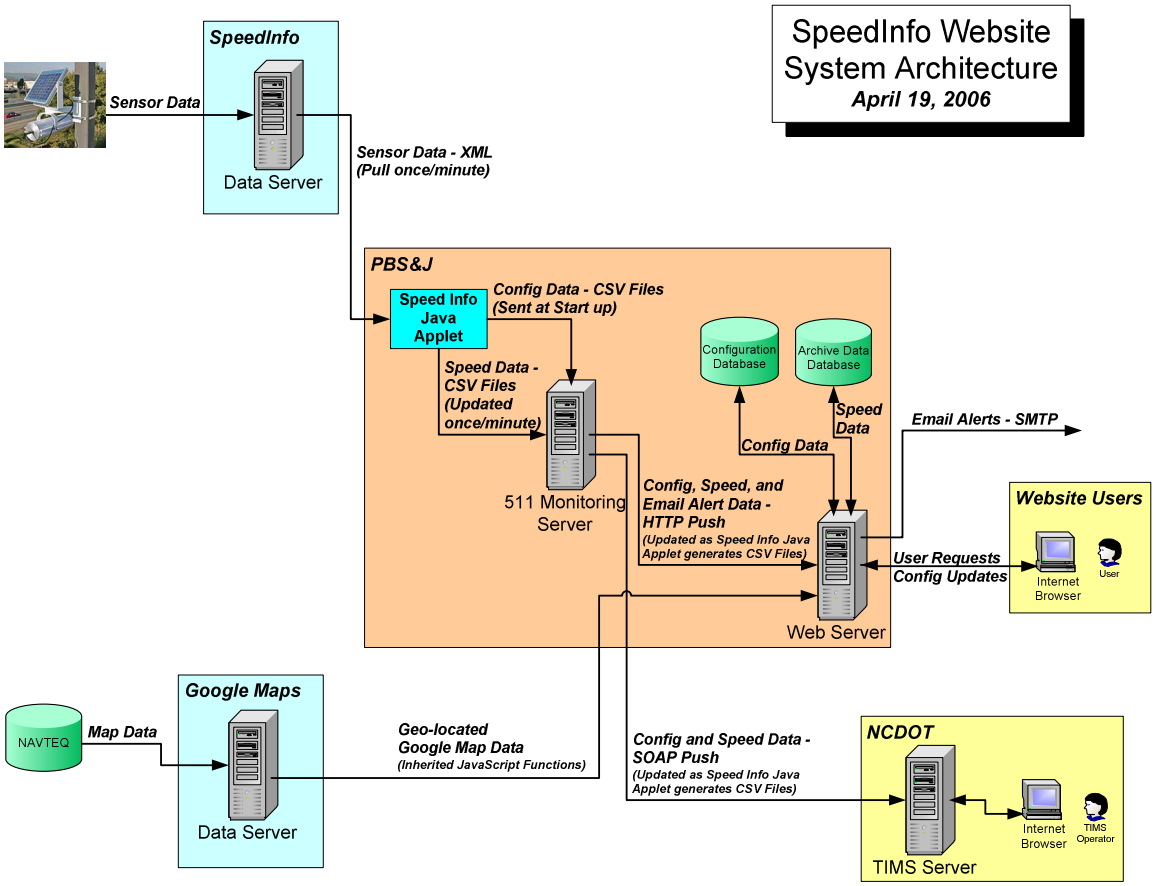


Figure 1: NCDOT SpeedInfo System Architecture

SpeedInfo sensor data is collected by the SpeedInfo Data Server. A SpeedInfo-developed Java plug-in, which runs on the PBS&J 511 Monitoring Server, connects to the SpeedInfo Data Server. The Java plug-in requests speed data for all NCDOT SpeedInfo sensors once per minute. The Java plug-in reformats this into comma-delimited text files. A PBS&J-developed NCDOT SpeedInfo website data provider application reads in these text files as they are made available, determines the status of each sensor, sends out alerts if configured thresholds are surpassed, and pushes sensor configuration and speed data to the NCDOT SpeedInfo Web Server via Hypertext Transfer Protocol (HTTP) and to NCDOT TIMS system via Simple Object Access Protocol (SOAP). All NCDOT SpeedInfo configuration and speed data is stored in an online database on the NCDOT SpeedInfo Web Server. The web server contains the web pages and makes the website available via the Internet. The website uses Google Maps to provide a geo-located map of the area in which NCDOT SpeedInfo sensors are located.

2.2 NCDOT SpeedInfo Databases

Sensor configuration data and speed data is stored in two Microsoft Access databases on the NCDOT SpeedInfo Web Server. All configuration data is stored in a configuration database while all speed data is stored in an archive data database.

2.2.1 Configuration Database

Table 1 describes the tables in the configuration database.

Table 1: Sensor Configuration Database Tables

Table Name	Description
SensorData	Contains configuration data of all sensors.
SpeedThreshold	Contains speed threshold data for green, yellow, and red icons.
Users	Contains configuration data for website users.

Table 2 describes each of the fields in the SensorData table.

Table 2: SensorData Table

Field Name	Example Value	Editable via Website	Required Field	Field Type	Description
SensorID	7771	Yes	Yes	Text (max 50 chars)	A unique identifier associated with a sensor.
RoadName	I-95	Yes	Yes	Text (max 50 chars)	The road name where the sensor is located.
RoadDir	N	Yes	Yes	Text (max 50 chars)	The direction of travel that the sensor is reading.
Latitude	35.3393371	Yes	Yes	Double	Latitude value where the sensor is located in degrees.
Longitude	-78.512015	Yes	Yes	Double	Longitude value where the sensor is located in degrees.
County	Johnston	Yes	No	Text (max 50 chars)	County in which the sensor is located.
MilePost	81	Yes	Yes	Double	Mile mark of the roadway on which the sensor is located.

CrossStreet	Keen Rd	Yes	No	Text (max 50 chars)	Closest cross street to the location of the sensor.
MapIconAngle	315	Yes	Yes	Integer	Angle in which the map icon should be displayed. This value can only be 0, 45, 90, or 315.

Table 3 describes each of the fields in the SpeedThreshold table.

Table 3: SpeedThreshold Table

Field Name	Example Value	Editable via Website	Required Field	Field Type	Description
ThresholdID	green	Yes	Yes	Text (max 50 chars)	A unique identifier associated with the speed threshold. Currently, this field can only be green, yellow, or red.
MinSpeed	55	Yes	Yes	Long Integer	The minimum speed defining the speed threshold.

Table 4 describes each of the fields in the Users table.

Table 4: Users Table

Field Name	Example Value	Editable via Website	Required Field	Field Type	Description
UserID	5	No	Yes	Long Integer	A unique identifier associated with a user.
UserName	john	Yes	Yes	Text (max 50 chars)	A unique user name associated with a user, used to log into the website.
UserPswd	password	Yes	Yes	Text (max 50 chars)	Password used to log into the website.
Config	True	Yes	Yes	Boolean	Determines if user can access the Configuration Menu Page.
ConfigUser	False	Yes	Yes	Boolean	Determines if user can access the User Configuration Page.
Archive	True	Yes	Yes	Boolean	Determines if user can access the Archive Data Page.

2.2.2 Archive Data Database

Table 5 describes the tables in the archive data database.

Table 5: Archive Data Database Tables

Table Name	Description
CurrentData	Contains most current speed data for all configured sensors.
SpeedData	Contains the archived speed data for all configured sensors.

Table 6 describes each of the fields in the CurrentData table.

Table 6: CurrentData Table

Field Name	Example Value	Editable via Website	Required Field	Field Type	Description
ID	20	No	Yes	Long Integer	A unique identifier associated with a speed data entry.
ReadTime	3/27/2006 4:02:05 PM	Yes	Yes	Text (max 50 chars)	The date/time when the SpeedInfo sensor reported a speed.
TimeSlice	3/27/2006 4:02:30 PM	No	Yes	Text (max 50 chars)	The date/time when the speed data was archived to the database.
SensorID	7771	Yes	Yes	Double	A unique identifier associated with a sensor.
Speed	56	Yes	Yes	Double	The speed reported from the SpeedInfo sensor in MPH.
Status	OK	Yes	Yes	Text (max 50 chars)	The reported status of the SpeedInfo sensor. Currently, this value can only be OK or OUT-OF-SERVICE.

Table 7 describes each of the fields in the SpeedData table.

Table 7: SpeedData Table

Field Name	Example Value	Accessible via Website	Required Field	Field Type	Description
ID	20	No	Yes	Long Integer	A unique identifier associated with a speed data entry.
ReadTime	3/27/2006 4:02:05 PM	Yes	Yes	Text (max 50 chars)	The date/time when the SpeedInfo sensor reported a speed.
TimeSlice	3/27/2006 4:02:30 PM	No	Yes	Text (max 50 chars)	The date/time when the speed data was archived to the database.
SensorID	7771	Yes	Yes	Double	A unique identifier associated with a sensor.
Speed	56	Yes	Yes	Double	The speed reported from the SpeedInfo sensor in MPH.
Status	OK	Yes	Yes	Text (max 50 chars)	The reported status of the SpeedInfo sensor. Currently, this value can only be OK or OUT-OF-SERVICE.

3.0 System Interfaces

3.1 SpeedInfo Data Server to Java Plug-in

The Java Plug-in connects to the SpeedInfo Data Server and requests sensor configuration and speed data. Configuration data is requested once upon connection. Speed data is requested once per configurable amount of time. The Java plug-in is currently configured to receive speed data updates once every 60 seconds.

The SpeedInfo Data Server sends all data in Extensible Markup Language (XML) format over a socket connection. For more information about the various XML fields and connection information, reference Appendix A of this document.

3.2 Java Plug-in to NCDOT SpeedInfo Website Data Provider

The Java Plug-in writes a new comma-delimited file to the "C:\Program Files\SpeedInfo\Output" every time it receives XML data from the SpeedInfo Data Server. Upon receiving sensor configuration data, it writes the file "DeviceConfig.csv". Table 8 lists all fields reported in "DeviceConfig.csv" (in the order in which they are reported). This file should include one entry per sensor.

Table 8: Device Configuration Comma-Delimited File

Field Name	Example Value	Description
DeviceID	7545	A unique identifier associated with a sensor.
RoadName	I-40	The roadway on which the sensor is located.
RoadDir	W	The direction of travel that the sensor is reporting.
Latitude	35.75591	The latitude value in degrees of the sensor's location.
Longitude	-78.71641	The longitude value in degrees of the sensor's location.

Upon receiving speed data from the SpeedInfo Data Server, the Java plug-in writes the file "SpeedDataXX.csv". The "XX" represents an integer from zero to one less than the maximum number of files configured (currently set to 60). After the NCDOT SpeedInfo Website Data Provider reads and processes SpeedData0.csv, for example, it will look for SpeedData1.csv, which initially should not exist. Once the Java plug-in writes SpeedData1.csv, the NCDOT SpeedInfo Website Data Provider will read and process it and then look for SpeedData2.csv. Once both applications reach the one less than the maximum number of files configured, then they will write and read SpeedData0.csv. To ensure that the Java plug-in is writing a new file every time, the NCDOT SpeedInfo Website Data Provider will delete old files. A configurable number of speed data files will be kept (currently set to 30). After one more than this configurable amount is written by the Java plug-in and the NCDOT SpeedInfo Website Data Provider reads the newest file, the oldest file is deleted.

Table 9 lists all fields reported in the speed data comma-delimited file written by the Java plug-in. This file should include one entry per sensor; hence, there should be the same number of entries in this list as in the device configuration file.

Table 9: Speed Data Comma-Delimited File

Field Name	Example Value	Description
DeviceID	7545	A unique identifier associated with a sensor.
ReadTime	2006-03-28 05:45:04	The timestamp when the speed was reported by the sensor.
Status	OK	The current reported status of the sensor. This value can be "OK" or "OUT-OF-SERVICE".
Speed	65	The speed reported by the sensor.

3.3 NCDOT SpeedInfo Website Data Provider to NCDOT SpeedInfo Website

Data is sent from the NCDOT SpeedInfo Website Data Provider to NCDOT SpeedInfo Website as text in a Uniform Resource Locators (URL) format via HTTP. The NCDOT SpeedInfo Website Data Provider generates a URL and uses a Microsoft Internet Explorer object to push data to Active Server Pages (ASP) that reside on the NCDOT SpeedInfo Website. Table 10 lists each page used and an example of the URL generated. Note that URL's have a number of "unsafe" characters, such as spaces and colons, which need to be translated into ASCII values.

Table 10: ASP's Used to Push Data to the NCDOT SpeedInfo Website

Active Server Page	Example URL	Description
setdeviceconfig.asp	http://www.ncdot-traffic.com/setdeviceconfig.asp? RoadName=I-95&RoadDir=N&Latitude=35.75591& Longitude=-78.71641&SensorID=7545	Saves sensor configuration data to the website's configuration database. One sensor can be pushed at a time.
setdevicedata.asp	http://www.ncdot-traffic.com/setdevicedata.asp? SensorID1=7545&Speed1=65& ReadTime1=3%2F28%2F2006 %209%3A55%3A02%20AM& TimeSlice1=3%2F28%2F2006 %209%3A55%3A02%46AM& Status1=OK&SensorID2=7546&Speed2=67& ReadTime2=3%2F28%2F2006 %209%3A55%3A02%15AM& TimeSlice2=3%2F28%2F2006 %209%3A55%3A02%46AM&Status2=OK	Saves speed data to the website's archive data database. From 1 to 5 sensors can be pushed at a time. In the example, two sensors are being pushed.
sendalert.asp	http://www.ncdot-traffic.com/sendalert.asp? EmailFrom=alertshandler%40pbsj%2Ecom& EmailTo=johnhope%40pbsj%2Ecom& EmailBody=Time&20Stamp&20&20&20&20&20 &20&20&20&20&20&20&20Time&20Cleared &20&20&20&20&20&20&20&20&20&20Alert &20Type&20&20&20&20&20&20&20&20 &20Sensor&3Cbr&3E ---------- ----------&3Cbr&3E 3%2F28%2F2006%209&3A40&3A36%20AM %203%2F28%2F2006%209&3A43&3A37%20AM %20Late%20Read%20%20%20%20 %20%20%20%20%20%20%207722& EmailSubject=SpeedInfo%20Website%20Alert	Sends an email alert.

Data is pushed from the NCDOT SpeedInfo Website Data Provider to NCDOT SpeedInfo Website when updated data is sent Java Plug-in to NCDOT SpeedInfo Website Data Provider. The only exception to this is when the NCDOT SpeedInfo Website Data Provider detects that no data is being provided past a configurable threshold (see Section 5.4 for more information about alerts and detecting a no data state). When this occurs, all configured sensors are updated once per minute with a status of "No Data". These updates are archived in the Archived Data Database.

3.4 NCDOT SpeedInfo Website Data Provider to NCDOT TIMS

SOAP requests are used to push data from the NCDOT SpeedInfo Website Data Provider to the NCDOT Traveler Information Management System (TIMS) in XML format. NCDOT has implemented a web service that provides functions to insert updated Speed Data and updated Sensor IDs into the TIMS SpeedInfo database. A description of these functions is included in Appendix B of this document. When a SOAP request is sent to the NCDOT web service, these functions are used to parse the incoming data and make the necessary database updates.

The NCDOT SpeedInfo Website Data Provider has implemented two types of SOAP requests, InsertSensorDataList and ChangeSensorId. The InsertSensorDataList SOAP request contains updated speed data from all sensors reporting speed data. Figure 2 is a portion of an example of an InsertSensorDataList XML request. This request is generated when updated data is sent Java Plug-in to NCDOT SpeedInfo Website Data Provider. The "SensorTimeStamp" field is in Coordinated Universal Time (UTC).


```
- <S:Envelope xmlns:a="http://apps.dot.state.nc.us/TIMS/WebService/" xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:XS="http://www.w3.org/2001/XMLSchema" xmlns:XI="http://www.w3.org/2001/XMLSchema-instance">
  - <S:Body>
    - <a:InsertSensorDataList>
        <a:password XI:type="XS:string">{FFC7EBBF-714F-4900-895E-2AC9FA6F8C9F}</a:password>
      - <a:sensorDataList>
        - <a:SensorData>
            <a:SensorId XI:type="XS:int">1610</a:SensorId>
            <a:SensorDataId XI:type="XS:int">0</a:SensorDataId>
            <a:SensorTimeStamp XI:type="XS:dateTime">2006-04-13T10:20:15Z</a:SensorTimeStamp>
            <a:AverageSpeed XI:type="XS:int">66</a:AverageSpeed>
          </a:SensorData>
        - <a:SensorData>
            <a:SensorId XI:type="XS:int">1611</a:SensorId>
            <a:SensorDataId XI:type="XS:int">0</a:SensorDataId>
            <a:SensorTimeStamp XI:type="XS:dateTime">2006-04-13T10:20:15Z</a:SensorTimeStamp>
            <a:AverageSpeed XI:type="XS:int">68</a:AverageSpeed>
          </a:SensorData>
        - <a:SensorData>
            <a:SensorId XI:type="XS:int">7545</a:SensorId>
            <a:SensorDataId XI:type="XS:int">0</a:SensorDataId>
            <a:SensorTimeStamp XI:type="XS:dateTime">2006-04-13T10:20:53Z</a:SensorTimeStamp>
            <a:AverageSpeed XI:type="XS:int">62</a:AverageSpeed>
          </a:SensorData>
        - <a:SensorData>
            <a:SensorId XI:type="XS:int">7567</a:SensorId>
            <a:SensorDataId XI:type="XS:int">0</a:SensorDataId>
            <a:SensorTimeStamp XI:type="XS:dateTime">2006-04-13T10:20:05Z</a:SensorTimeStamp>
            <a:AverageSpeed XI:type="XS:int">73</a:AverageSpeed>
          </a:SensorData>
        - <a:SensorData>
            <a:SensorId XI:type="XS:int">7568</a:SensorId>
            <a:SensorDataId XI:type="XS:int">0</a:SensorDataId>
            <a:SensorTimeStamp XI:type="XS:dateTime">2006-04-13T10:20:05Z</a:SensorTimeStamp>
```


Figure 2: InsertSensorDataList SOAP Request

The ChangeSensorId SOAP request is sent if a Sensor ID changes. Occassionally when Speed Info performs Speed Info server updates, a sensor could be assigned a new Sensor ID without any other configuration data changing. When the NCDOT SpeedInfo Website Data Provider detects when a Sensor ID has changed, a ChangeSensorId SOAP request is sent. Figure 3 is an example of a ChangeSensorId SOAP request.

```
- <S:Envelope xmlns:a="http://apps.dot.state.nc.us/TIMS/WebService/"
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:XS="http://www.w3.org/2001/XMLSchema"
    xmlns:XI="http://www.w3.org/2001/XMLSchema-instance">
  - <S:Body>
    - <a:ChangeSensorId>
        <a:password XI:type="XS:string">{FFC7EBBF-714F-4900-895E-2AC9FA6F8C9F}
          </a:password>
        <a:oldSensorId S:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          XI:type="XS:int">7402</a:oldSensorId>
        <a:nNewSensorId S:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          XI:type="XS:int">5584</a:nNewSensorId>
      </a:ChangeSensorId>
    </S:Body>
  </S:Envelope>
```


Figure 3: ChangeSensorId SOAP Request

The NCDOT SpeedInfo Website Data Provider uses an external application developed by Simon Fell called Pocket SOAP to handle the SOAP requests and responses. Documentation on this application can be found online at: http://www.pocketsoap.com/pocketsoap/docs/default.htm.

4.0 Website Interface

The website is hosted on a web server maintained by GoDaddy.com. The website consists of a series of web pages designed using Active Server Pages (ASP). These pages connect with Microsoft Access databases via ActiveX Data Objects Database connections (ADODB) and uses Collaborative Data Objects for NT Server (CDONTS) to send email alerts.

4.1 Website Map

Speed data is displayed on the website via a website map page. The website map uses mapping data and images from Google Maps, geo-locates representative icons on the map, and displays the area in which SpeedInfo sensors are deployed. Figure 4 is an example of this map.

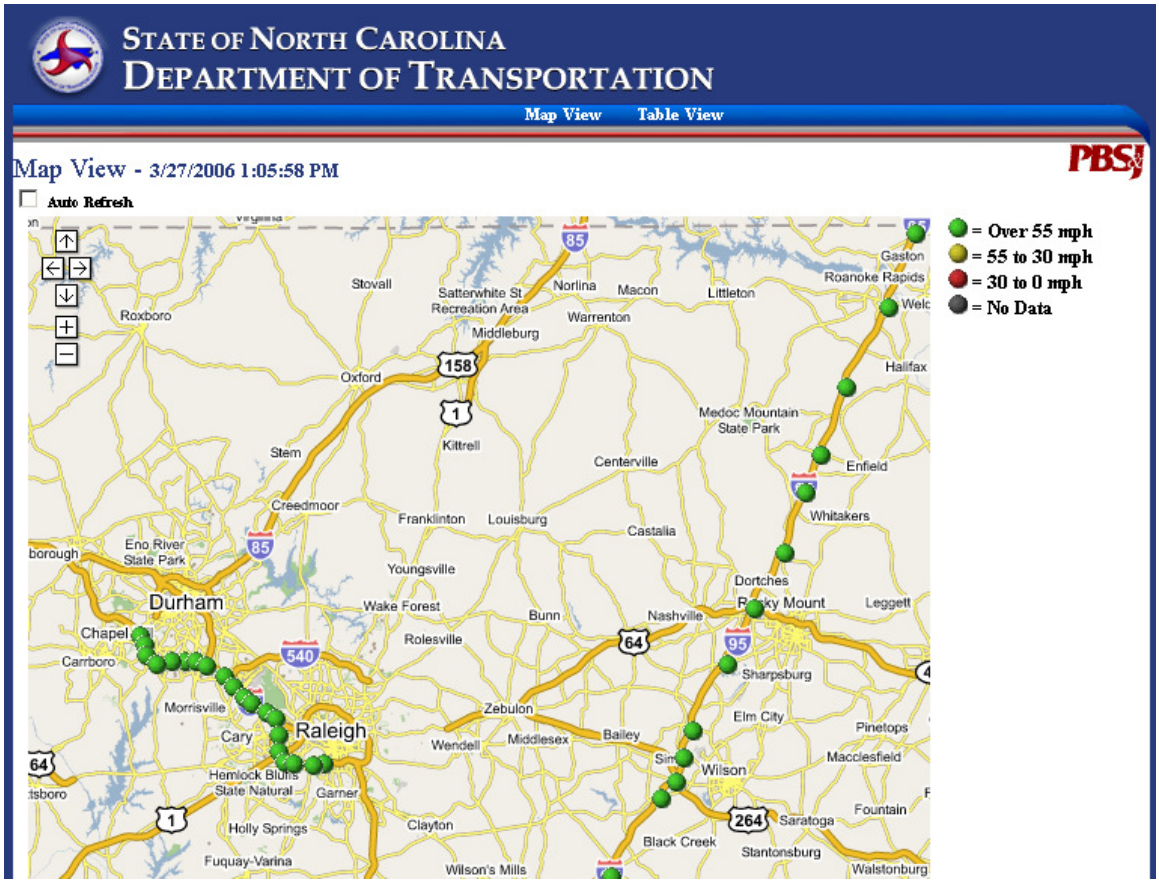


Figure 4: SpeedInfo Website Map Page

The icons on the map are color-coded according to the icon legend listed on the upper right corner of the page. The color-coded ranges are configurable via the website. The user can select "Auto Refresh" at the top left corner of the page to automatically refresh the page once per minute. Data for the website is updated once per minute.

When an icon is clicked with the mouse, a popup window is displayed within the map's view displaying data currently configured for the selected sensor and the current speed data for the selected sensor. Figure 5 shows and example of this popup window.

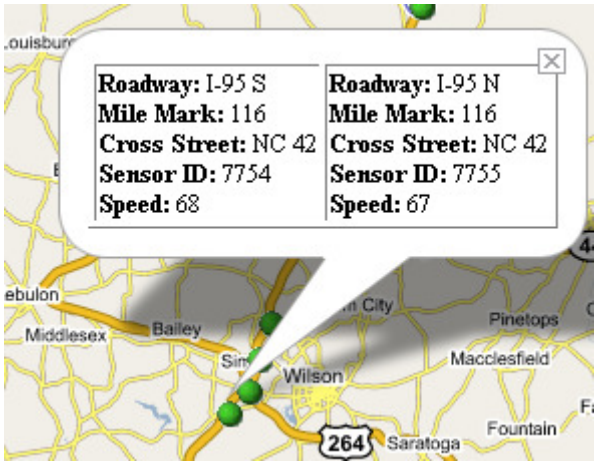


Figure 5: Sensor Popup Window

Each icon on the map represents two sensors from one sensor location (i.e. a single sensor site). The two sensors report data from each direction of travel. For example, the sensors on I-95 report northbound and southbound speed data. Therefore, each sensor site reports a northbound speed and a southbound speed. Half of the icon is colored based on the speed data of one direction and the other half is colored based on the speed data of the other direction. If the icon is completely green, for example, then both directions are reporting speed data within the green threshold. In Figure 6, one the other hand, is an example where half of the icon is reporting green (southbound is at 69 MPH) and the other half is reporting yellow (northbound is at 35 MPH).

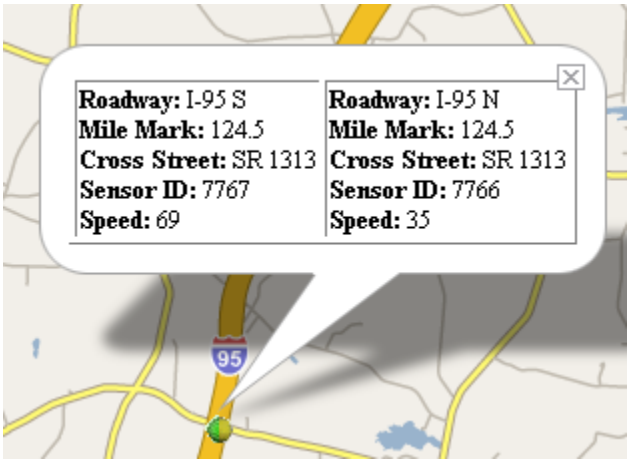


Figure 6: Map Icon Reporting Two Directions

Google Maps provides map navigation tools to pan the map and to zoom in or out. Controls for these tools are located at the top left corner of the map. The map can also be panned by clicking on the map area and moving the mouse while the mouse button remains pressed.

4.2 Website Table View Page

Speed data is also available on the website via a website table view page. This page lists all configured sensors, sorted by roadway, direction of travel, and mile mark where the sensor is located. Figure 7 is an example of this page.

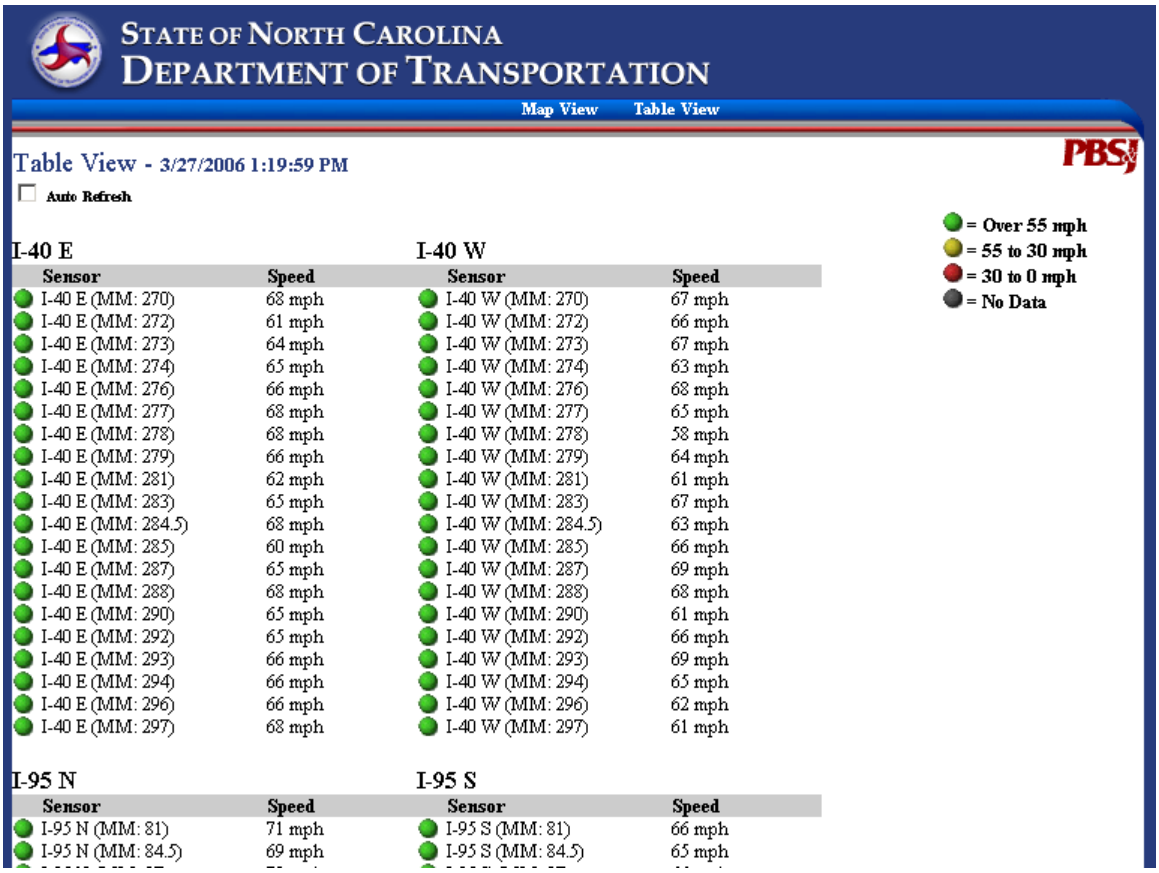


Figure 7: SpeedInfo Website Table View Page

The icons next to each sensor in the table are color-coded according to the icon legend listed on the upper right corner of the page. The color-coded ranges are configurable via the website. When the mouse hovers over an icon, the sensor identifier and mile mark where the sensor is located is displayed in an information popup window. Clicking on an icon or the text describing a sensor launches a sensor information window. This window displays currently configured sensor data and the past 30 minutes of speed data collected for the selected sensor. Figure 8 is an example of this window.

Sensor ID: 7708

Roadway: I-40 (MM: 283)
Cross Street: I-540
County: Wake
Latitude: 35.86925
Longitude: -78.82576

Past 30 Minutes of Reads

Time Stamp	Read Time	Status	Speed
3/27/2006 1:19:59 PM	3/27/2006 1:19:05 PM	OK	67 mph
3/27/2006 1:18:58 PM	3/27/2006 1:18:05 PM	OK	68 mph
3/27/2006 1:17:58 PM	3/27/2006 1:17:05 PM	OK	63 mph
3/27/2006 1:16:58 PM	3/27/2006 1:16:05 PM	OK	65 mph
3/27/2006 1:15:58 PM	3/27/2006 1:15:05 PM	OK	67 mph
3/27/2006 1:14:59 PM	3/27/2006 1:14:05 PM	OK	66 mph
3/27/2006 1:13:58 PM	3/27/2006 1:13:05 PM	OK	64 mph
3/27/2006 1:12:58 PM	3/27/2006 1:12:05 PM	OK	63 mph
3/27/2006 1:11:59 PM	3/27/2006 1:11:05 PM	OK	67 mph
3/27/2006 1:10:58 PM	3/27/2006 1:10:05 PM	OK	66 mph
3/27/2006 1:09:58 PM	3/27/2006 1:09:05 PM	OK	65 mph
3/27/2006 1:08:58 PM	3/27/2006 1:08:05 PM	OK	64 mph
3/27/2006 1:07:58 PM	3/27/2006 1:07:05 PM	OK	65 mph
3/27/2006 1:06:58 PM	3/27/2006 1:06:05 PM	OK	65 mph
3/27/2006 1:05:58 PM	3/27/2006 1:05:05 PM	OK	65 mph
3/27/2006 1:04:58 PM	3/27/2006 1:04:05 PM	OK	63 mph
3/27/2006 1:03:57 PM	3/27/2006 1:03:05 PM	OK	65 mph
3/27/2006 1:02:58 PM	3/27/2006 1:02:05 PM	OK	66 mph
3/27/2006 1:01:58 PM	3/27/2006 1:01:05 PM	OK	64 mph
3/27/2006 1:00:58 PM	3/27/2006 1:00:05 PM	OK	64 mph
3/27/2006 12:59:58 PM	3/27/2006 12:59:05 PM	OK	64 mph
3/27/2006 12:58:58 PM	3/27/2006 12:58:05 PM	OK	66 mph
3/27/2006 12:57:58 PM	3/27/2006 12:57:05 PM	OK	67 mph
3/27/2006 12:56:58 PM	3/27/2006 12:56:05 PM	OK	66 mph
3/27/2006 12:55:57 PM	3/27/2006 12:55:05 PM	OK	65 mph
3/27/2006 12:54:58 PM	3/27/2006 12:54:05 PM	OK	64 mph
3/27/2006 12:53:57 PM	3/27/2006 12:53:05 PM	OK	65 mph

Figure 8: Sensor Information Window

Time Stamp is the date/time that the data was archived, the Read Time is the date/time when the SpeedInfo sensor reported the speed data, Status is the current status of the SpeedInfo sensor, and Speed is the speed that the SpeedInfo sensor reported.

4.3 Login Page

When users navigate to the NCDOT SpeedInfo website homepage, at http://www.ncdot-traffic.com, the website automatically bypasses the Login Page and logs the user in as a guest. Guest logins only have access to the Map and Table View pages. The Login Page is located at http://www.ncdot-traffic.com/login.asp

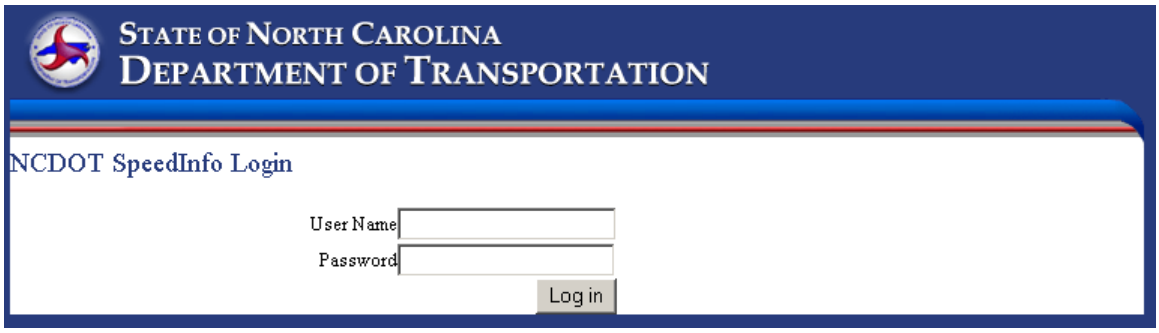


Figure 9: Login Page

Figure 9 is an example of the Login Page. Users must type in a valid user name and password and click "Log in" to log into the website from the Login Page. After logging in, all users expect for guest logins will have a menu option to log out.

Upon login, an internet cookie is saved to the user's computer. ***Users will not have the ability to successfully log in without enabling cookies in their browser configuration for the NCDOT SpeedInfo website.***

4.4 Configuration Menu Page

Users may have access to the Configuration Menu page depending on their current user configuration settings. This page provides access to the Sensor Configuration, Speed Threshold Configuration, and User Configuration Pages. One or more of these pages may not be accessible to the user depending on user configuration settings. Figure 10 is an example of this page.

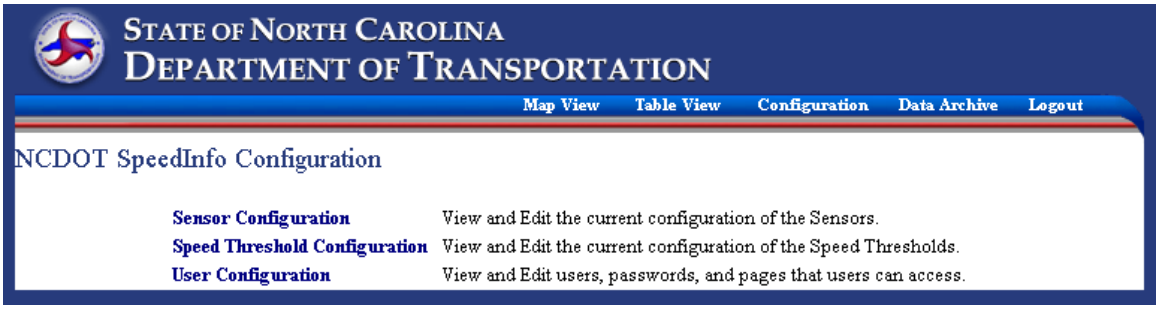
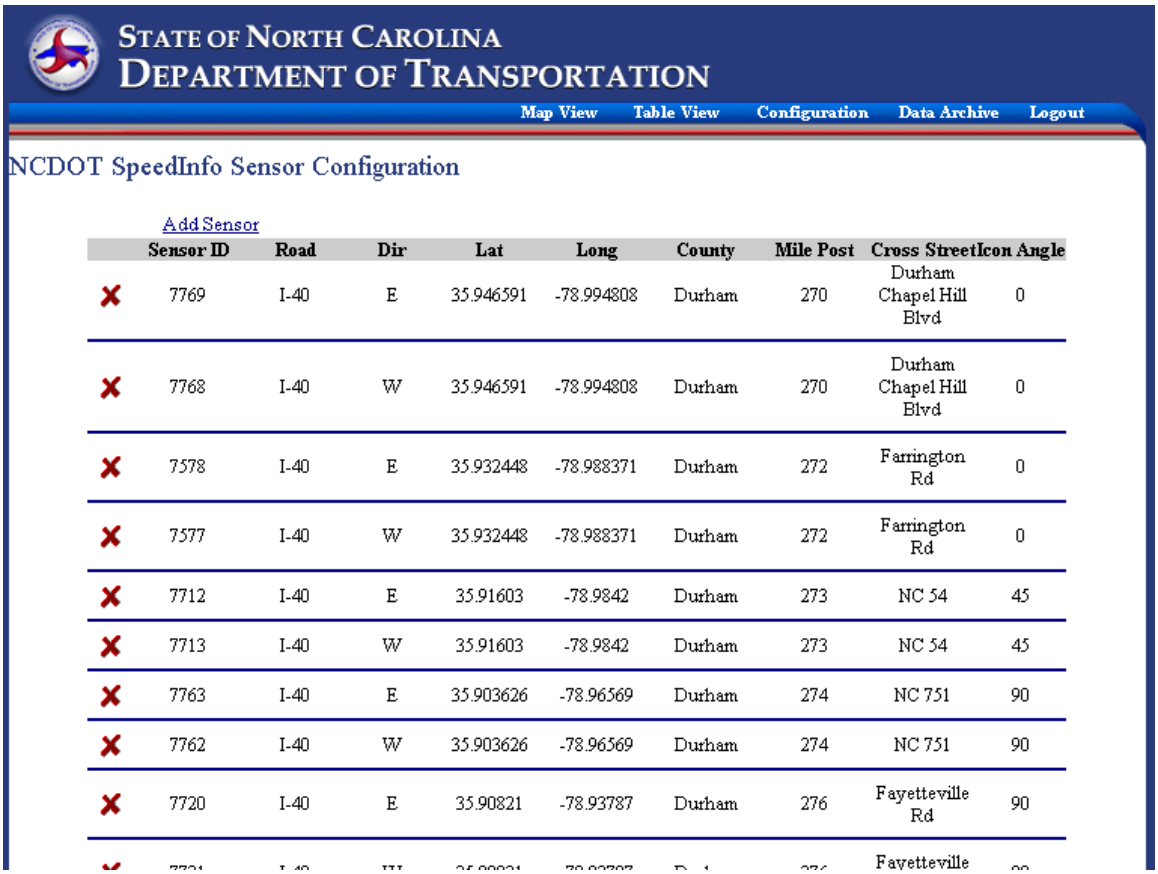


Figure 10: Configuration Menu Page

4.5 Sensor Configuration Page

The Sensor Configuration Page allows the user to view and edit the currently configured sensors. A user can manually add a sensor by clicking the "Add Sensor" link at the top left corner of the page. Sensors can be manually removed by clicking the red "X" next to the selected sensor. Any field of data can be edited by clicking the field value in the page's table, changing the value in the popup text box, and clicking "OK". Figure 11 is an example of this page.



	Sensor ID	Road	Dir	Lat	Long	County	Mile Post	Cross Street	Icon Angle
✘	7769	I-40	E	35.946591	-78.994808	Durham	270	Durham Chapel Hill Blvd	0
✘	7768	I-40	W	35.946591	-78.994808	Durham	270	Durham Chapel Hill Blvd	0
✘	7578	I-40	E	35.932448	-78.988371	Durham	272	Farrington Rd	0
✘	7577	I-40	W	35.932448	-78.988371	Durham	272	Farrington Rd	0
✘	7712	I-40	E	35.91603	-78.9842	Durham	273	NC 54	45
✘	7713	I-40	W	35.91603	-78.9842	Durham	273	NC 54	45
✘	7763	I-40	E	35.903626	-78.96569	Durham	274	NC 751	90
✘	7762	I-40	W	35.903626	-78.96569	Durham	274	NC 751	90
✘	7720	I-40	E	35.90821	-78.93787	Durham	276	Fayetteville Rd	90

Figure 11: Sensor Configuration Page

The "Icon Angle" field specifies the angle in degrees in which an icon should be rotated when displayed on the Map Page. This allows the icons on the Map page to be rotated so that the center of the icons aligns with the centerline of the roadway. This is necessary for accurately depicting speed data over two directions of travel for a single icon. The Icon Angle can only have the values of 0, 45, 90, and 315.

Sensors are automatically added to this list of configuration data if a sensor is added to the SpeedInfo system and is reported from SpeedInfo via the Sensor Configuration data.

4.6 Speed Threshold Configuration Page

The Sensor Configuration Page allows user to view and edit the currently configured speed thresholds. These values define the color-coded ranges of speeds that are associated with each map icon. In other words, the website uses these values to determine which color an icon should be. Figure 12 is an example of this page.

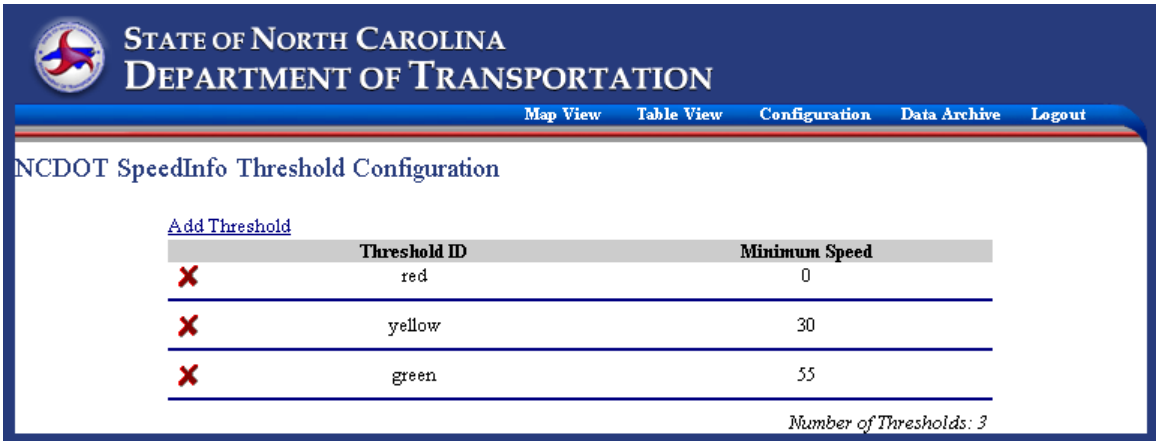


Figure 12: Speed Threshold Configuration Page

The thresholds are checked in the following order: green, yellow, and then red. When a speed is reported for value over the configured minimum speed for green, then the icon for the sensor is displayed green. If the speed is less than the configured minimum speed for green and greater the configured minimum speed for yellow, then the icon is displayed yellow. If the speed is less than the configured minimum speed for yellow and greater the configured minimum speed for red, then the icon is displayed red. If the speed is outside the configured speed threshold range (i.e. the speed is less than the configured minimum speed for red) or if no data is reported for a sensor, then the sensor is displayed as gray.

Thresholds can be edited by clicking the threshold value, changing the value in the popup text box, and clicking "OK".

4.7 User Configuration Page

The User Configuration Page allows users to add and configure website users. Users can add users by clicking the "Add User" at the top left corner of the page. Figure 13 is an example of this page.

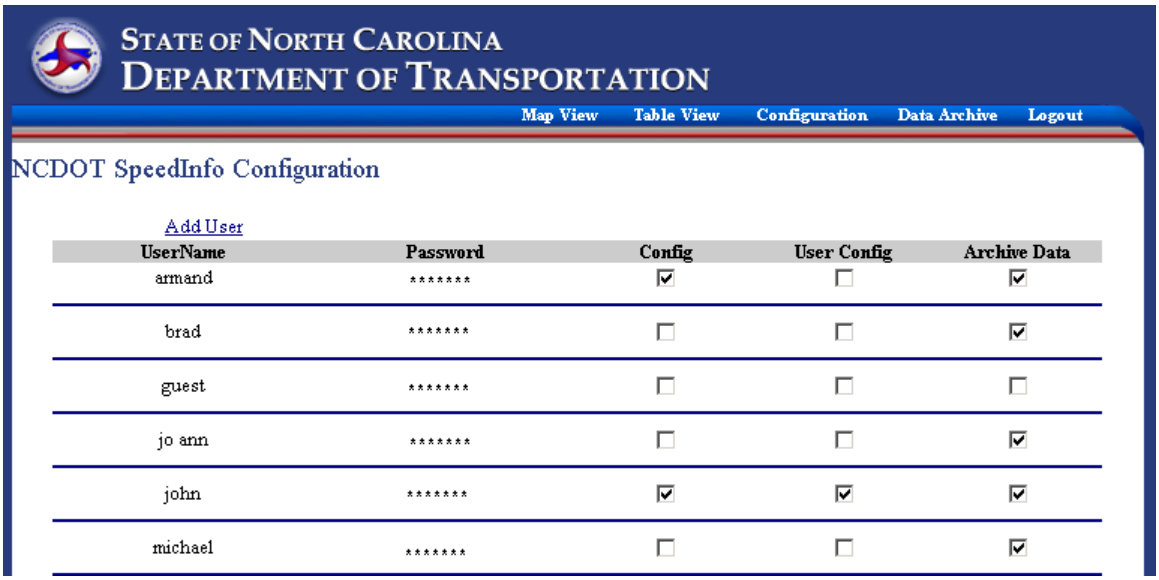


Figure 13: User Configuration Page

The user name and password fields can be edited by clicking the field value, changing the value in the popup text box, and clicking "OK". A user can be configured to have access to the Configuration Menu Page by checking the "Config" checkbox, to the User Configuration Page by checking the "User Config" checkbox, and to the Archive Data Page by checking the "Archive Data" checkbox. The page access checkboxes automatically save once a checkbox is clicked.

4.8 Archive Data Page

Speed data is archived for all sensors once per minute to an online accessible database. Up to three months of data is accessible online through the Data Archive Page. This page allows users to select any or all configured sensors and a date range of when the data was archived. The configured sensors are listed on this page, sorted by roadway, direction of travel, and mile mark. Data will be retrieved for a selected sensor by checking the checkbox next to the sensor information, setting a time range, and clicking on the "Get Data" button. Figure 14 is an example of this page.

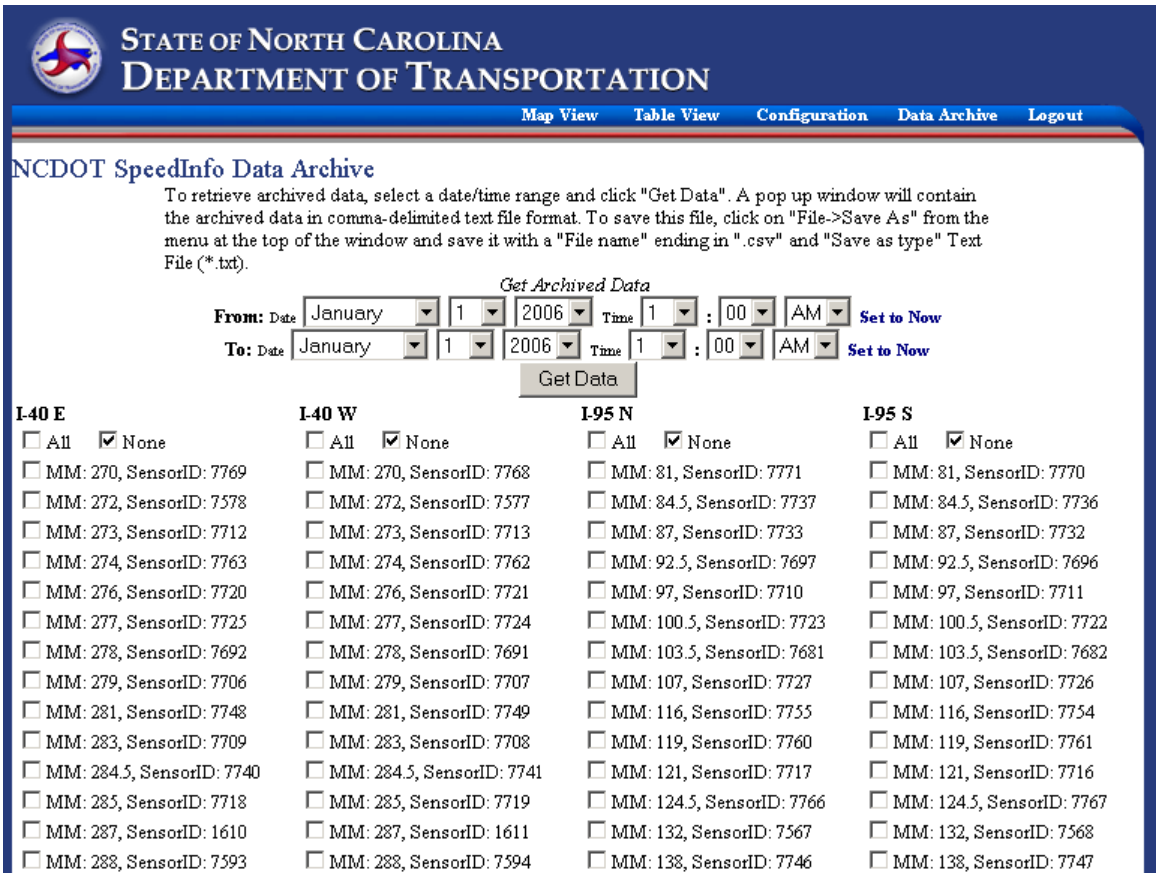


Figure 14: Data Archive Page

Selecting "All" will automatically select all sensors for underneath the selected roadway for the selected direction of travel. Clicking on "Set to Now" will automatically set the From or To date and time next to current date and time currently reported from the user's computer system clock.

The website limits the user to downloading 57,600 rows of data per request. This lessens the possibility of "locking up" the website with data requests. Since data is saved once per minute, this limit is equivalent to requesting 12 hours of data from 80 sensors.

When "Get Data" is clicked, a window will be launched that contains a comma-delimited text file containing the requested data. The data may appear to be jumbled in the user's browser window since the data is formatted as a comma-delimited text file and does not use HTML. Figure 15 is an example of the reported data in the user's browser.

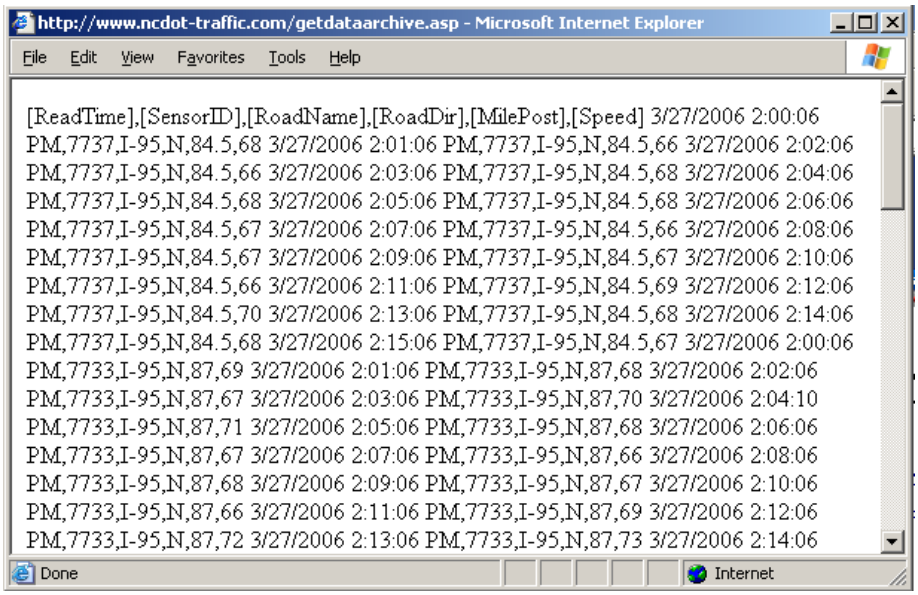


Figure 15: Reported Archived Data in a User's Browser

Archived Data can be saved to the user's computer by using the browser's file saving functions. If Microsoft Internet Explorer 6.0 is used, for example, the user can perform the following steps to save the data to a comma-delimited text file:

1) Click on the "File->Save As…" menu option from the top of the window.
2) Type in a file name appended with the extension ".csv".
3) Select "Text File (*.txt)" from the "Save as Type" menu (see Figure 16 as an example).
4) Click on "Save".

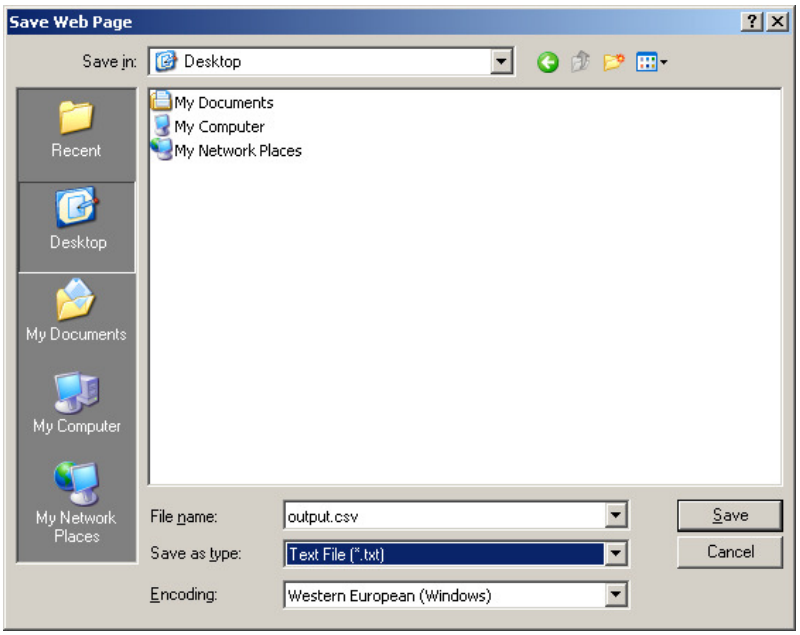
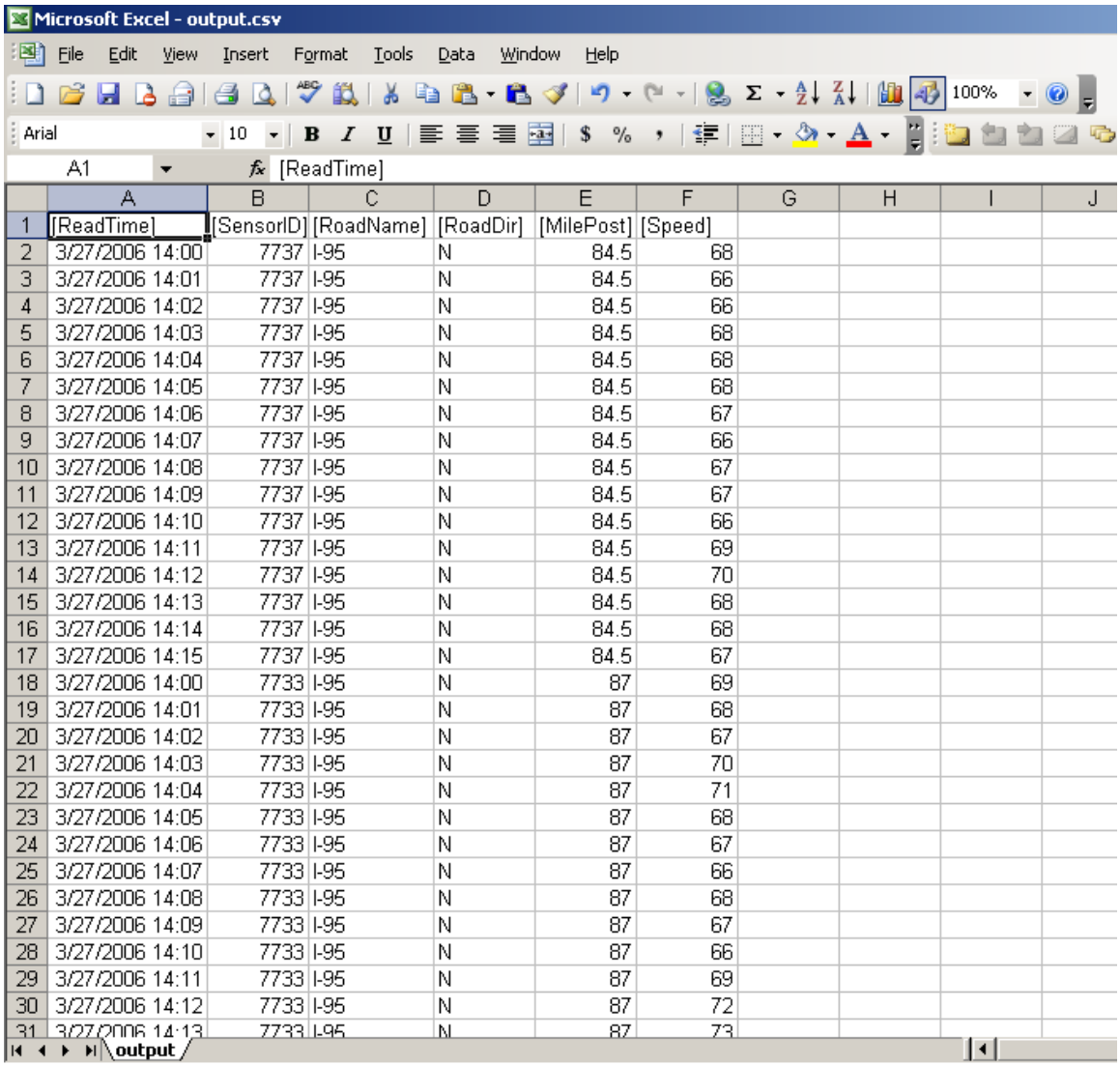


Figure 16: Saving Archive Data to a File via Microsoft Internet Explorer 6.0

After the file is saved to the user's computer, it can be opened as a text file in a standard text editor or opened in Microsoft Excel. Figure 17 is an example of the Archive Data output in Excel.



	A	B	C	D	E	F
1	[ReadTime]	[SensorID]	[RoadName]	[RoadDir]	[MilePost]	[Speed]
2	3/27/2006 14:00	7737	I-95	N	84.5	68
3	3/27/2006 14:01	7737	I-95	N	84.5	66
4	3/27/2006 14:02	7737	I-95	N	84.5	66
5	3/27/2006 14:03	7737	I-95	N	84.5	68
6	3/27/2006 14:04	7737	I-95	N	84.5	68
7	3/27/2006 14:05	7737	I-95	N	84.5	68
8	3/27/2006 14:06	7737	I-95	N	84.5	67
9	3/27/2006 14:07	7737	I-95	N	84.5	66
10	3/27/2006 14:08	7737	I-95	N	84.5	67
11	3/27/2006 14:09	7737	I-95	N	84.5	67
12	3/27/2006 14:10	7737	I-95	N	84.5	66
13	3/27/2006 14:11	7737	I-95	N	84.5	69
14	3/27/2006 14:12	7737	I-95	N	84.5	70
15	3/27/2006 14:13	7737	I-95	N	84.5	68
16	3/27/2006 14:14	7737	I-95	N	84.5	68
17	3/27/2006 14:15	7737	I-95	N	84.5	67
18	3/27/2006 14:00	7733	I-95	N	87	69
19	3/27/2006 14:01	7733	I-95	N	87	68
20	3/27/2006 14:02	7733	I-95	N	87	67
21	3/27/2006 14:03	7733	I-95	N	87	70
22	3/27/2006 14:04	7733	I-95	N	87	71
23	3/27/2006 14:05	7733	I-95	N	87	68
24	3/27/2006 14:06	7733	I-95	N	87	67
25	3/27/2006 14:07	7733	I-95	N	87	66
26	3/27/2006 14:08	7733	I-95	N	87	68
27	3/27/2006 14:09	7733	I-95	N	87	67
28	3/27/2006 14:10	7733	I-95	N	87	66
29	3/27/2006 14:11	7733	I-95	N	87	69
30	3/27/2006 14:12	7733	I-95	N	87	72
31	3/27/2006 14:13	7733	I-95	N	87	73

Figure 17: Archive Data Output Opened in Microsoft Excel

Archive data is sorted by roadway, direction of travel, and mile mark. Section 2.2.2 of this document defines all fields saved in the Archive Data database.

5.0 NCDOT SpeedInfo Website Data Provider

5.1 Application Overview

The NCDOT SpeedInfo Website Data Provider is a PBS&J-developed application that retrieves configuration and speed data from the SpeedInfo-developed Java plug-in, processes the configuration and speed data, and pushes the data to the NCDOT SpeedInfo website. The application relies on the Java plug-in to connect to the SpeedInfo Data Server and provide configuration and speed data. The application is also dependant on the PBS&J network to maintain a constant network connection to the SpeedInfo Data Server and an internet connection to the NCDOT SpeedInfo web server.

The NCDOT SpeedInfo Website Data Provider runs on the PBS&J 511 Monitoring Server. Upon launching the application, the Java plug-in is automatically launched. It is important that these applications are launched together to ensure that applications are in synch.

5.2 Graphical User Interface

Figure 18 is the NCDOT SpeedInfo Website Data Provider Graphical User Interface (GUI).

Figure 18: NCDOT SpeedInfo Website Data Provider GUI

"Last Config Update" is the last time configuration data has been pushed to the NCDOT SpeedInfo Website configuration database. "Last Data Update" is the last time speed data has been pushed to the NCDOT SpeedInfo Website archived data database. "Data File Index" is the next speed data index that the NCDOT SpeedInfo Website Data Provider is expecting. This index corresponds with the index used to name the "SpeedDataXX.csv" comma-delimited speed data file generated by the Java plug-in, as described in Section 3.2 of this document.

Pressing the "Log" button will launch WordPad with the NCDOT SpeedInfo Website Data Provider's log file. The log file contains errors or warnings that have occurred during the application's operation.

Pressing the "Debug" button expands the GUI to show the action list and view buttons. Figure 19 is the expended GUI.

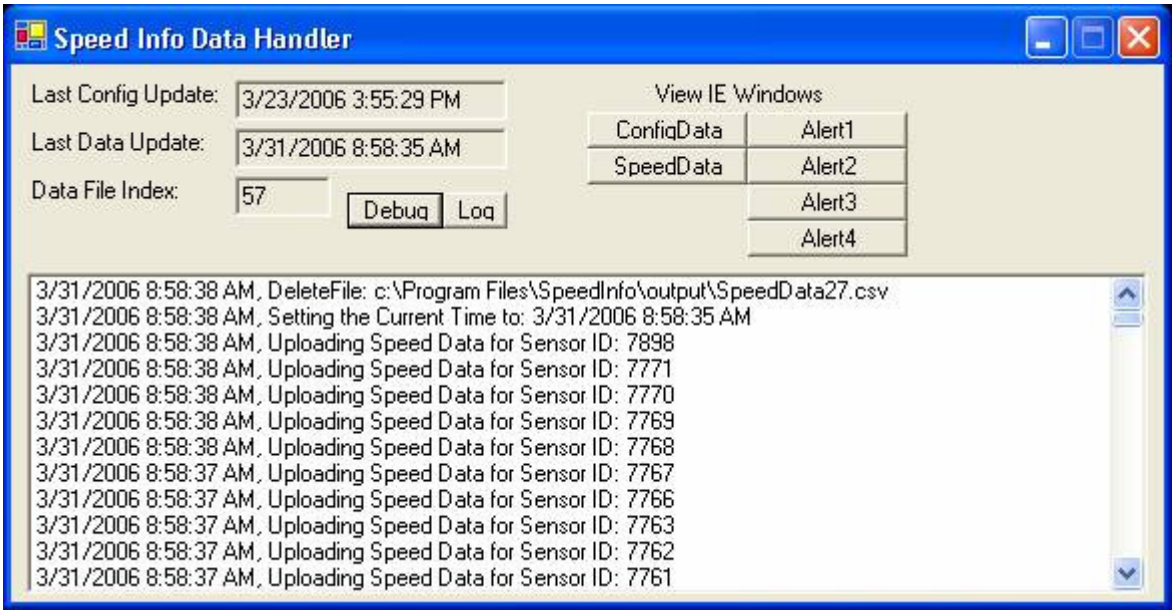


Figure 19: NCDOT SpeedInfo Website Data Provider Debug GUI

The action list is a continually updating list of actions that the NCDOT SpeedInfo Website Data Provider has performed. Each entry has a timestamp of when the action occurred and details about the action. The action list contains the most recent 500 actions.

The NCDOT SpeedInfo Website Data Provider uses Microsoft Internet Explorer (IE) objects to push configuration, speed, and email alert data to the NCDOT SpeedInfo Website. Pressing the "View IE Windows" buttons makes these objects visible for debugging purposes. **Warning: Once the IE objects are visible, the objects should never be closed.** Pressing the selected "View IE Windows" buttons a second time will make the IE objects invisible. For example, the "ConfigData" button will make the IE object open that handles pushing configuration data to the NCDOT SpeedInfo Website. To make this object invisible, the "ConfigData" button should be pressed a second time.

To collapse the expanded GUI back to it's default state, the "Debug" button should be pressed a second time.

5.3 Configuration

Table 11 describes the configuration parameters that can be set for the NCDOT SpeedInfo Website Data Provider.

Table 11: Configuration Fields

Parameter	Example Value	Description
PathSetConfig	http://www.ncdot-traffic.com/setdeviceconfig.asp	URL of ASP page used to save configuration data to the configuration database.
PathSetData	http://www.ncdot-traffic.com/setdevicedata.asp	URL of ASP page used to save speed data to the archive data database.
PathSendAlert	http://www.ncdot-traffic.com/sendalert.asp	URL of ASP page used to send email alerts.

MaxDataFiles	60	The Java plug-in saves speed data in a series of comma-delimited files named "SpeedDataXX.csv" where XX is an incremented value ranging from zero to (MaxDataFiles – 1). If this value is changed, the Java plug-in needs to be updated to maintain consistency between applications.
DataFilesKeepAmount	30	The comma-delimited files written by the Java plug-in are automatically deleted. The application will keep DataFilesKeepAmount. For example, if this number is set to 30, when the application reads the 31st comma-delimited file, the 1st file will be deleted. Likewise, when the 32nd file is read, the 2nd file is deleted. This number must be set to a value less than MaxDataFiles. Ideally, DataFilesKeepAmount = MaxDataFiles/2.
LateReadThreshold	300	Speed data is sent with a timestamp of when a sensor reports the speed. If this reported timestamp is older than the timestamp plus LateReadThreshold (in seconds) then a "Late Read" alert is sent.
AlertThresholdNoData	120	The Java plug-in is configured to write a new comma-delimited file once every 60 seconds. If AlertThresholdNoData (in seconds) time passes without a new comma-delimited file, a "No Data" alert is sent. This alert is typically generated when the Java plug-in is disconnected from the SpeedInfo Data Server for an extended period of time.
AlertFrequency	600	Alerts will not be sent any more frequent than AlertFrequency (in seconds).

TimeZoneOffsetSensors	-8	Time zone offset from GMT that the sensors' clock is set.
TimeZoneOffsetWebsite	-5	Time zone offset from GMT that the website's clock is set.
AlertFromEmail	alertshandler@pbsj.com	When an email alert is sent, AlertFromEmail is used as the "From" email address.
AlertNoData	johnhope@pbsj.com;armandciccarelli@pbsj.com	The list of email addresses (separated by a semicolon) that will receive "No Data" alerts.
AlertSensorFailure	johnhope@pbsj.com;armandciccarelli@pbsj.com	The list of email addresses (separated by a semicolon) that will receive "Sensor Failure" alerts.
AlertSensorMissing	johnhope@pbsj.com;armandciccarelli@pbsj.com	The list of email addresses (separated by a semicolon) that will receive "Sensor Missing" alerts.
AlertLateRead	johnhope@pbsj.com;armandciccarelli@pbsj.com	The list of email addresses (separated by a semicolon) that will receive "Late Read" alerts.
DebugMode	False	This should normally be set to False. When set to True, debug information is output to a debug log file and the website is not updated.

All configuration data is contained within "SpeedInfo.ini" located "C:\Program Files\SpeedInfo\config" on the PBS&J 511 Monitoring Server. The application reloads the configuration data once per minute without requiring an application restart.

5.4 Alerting Functionality

The NCDOT SpeedInfo Website Data Provider has the ability to send email alerts when alerting thresholds are surpassed. These thresholds are described in Table 8. When an alert is generated, the alert is sent to the NCDOT SpeedInfo web server via HTTP and an ASP page creates a CDONTS object to send the email alert. Table 12 describes each of these alerts.

Table 12: Email Alerts

Alert	Description
No Data	The Java plug-in is configured to write a new comma-delimited file once every 60 seconds. If a configurable amount of time passes without a new comma-delimited file, an email alert is sent. This alert is typically generated when the Java plug-in is disconnected from the SpeedInfo Data Server for an extended period of time.
Sensor Failure	If the SpeedInfo Data Server reports a sensor with a status of "OUT-OF-SERVICE" or if a sensor reports a speed of "0", then the sensor is flagged as a failed sensor and an email alert is sent.

Missing Sensor	If the SpeedInfo Data Server fails to report speed data for a configured reader, the missing sensors are flagged and an email alert is sent.
Late Read	Speed data is sent with a timestamp of when a sensor reports the speed. If this reported timestamp is older than the timestamp plus a configurable amount then an email alert is sent.

One email alert is sent when the criteria for that alert has been met. The alert is then considered an active alert and is logged to a file. The alert becomes inactive once the conditions that caused the alert to occur no longer exist. Once this occurs, another email alert is sent with a "Clear Time". The clear time specifies the date/time when the conditions that caused the alert to occur no longer exist. Inactive alerts are removed from the logged file.

Figure 20 is an example of a late read email alert.

SpeedInfo Website Alert

alertshandler

To: Hope, John R.; Ciccarelli, Armand J.

```
Time Stamp            Time Cleared           Alert Type          Sensor
-----------------------------------------------------------------------------
3/28/2006 8:42:18 AM  3/28/2006 8:48:19 AM  Late Read           7722
3/28/2006 8:42:18 AM  3/28/2006 8:48:19 AM  Late Read           7723
```

Figure 20: Example of a Late Read Alert

In the example above, the email alert is reporting that sensors 7722 and 7723 reported a late read on 3/28/2006 at 8:42:18 AM. The alert was cleared (i.e. changed from active to inactive) on 3/28/2006 at 8:48:19 AM.

Appendix A

SpeedInfo Data Feed Specification

SpeedInfo Data Feed Specification

Revision History

File name	Rev	Date	Editor	Page	Description
SpeedInfo Data Feed Specification	01	10/07/05	Aidan Smyth	--	First Draft

Table of Contents

1. Introduction 4
2. Device Description Data 4
3. Speed Data 5
4. Properties File 6
4.1 Measurement Latency 6
4.2 Feed Name 6

6.0 Introduction

SpeedInfo provides information about a given sensor network over an XML feed. The data in the feed falls into two categories:

- device (sensor) description data describes the location and the type of the sensor. This data is typically sent upon connecting to a feed and every 24 hours thereafter
- speed data defines the measured speeds and the times at which they were measured in relation to sensor id tags defined in the device description portion of the specification. This data is sent by default every 60 seconds for every sensor in the feed and can be sent as often as every 15 seconds by customer preference

7.0 Device Description Data

```
<!-- Device (sensor) description -->
<xsd:element name="device-description">
   <xsd:complexType>
      <xsd:sequence>
         <!-- Required Field: Unique sensor identifier -->
         <xsd:element name="Id" type="xsd:string"/>

         <!-- Required Field: Name of roadway on which sensor is located
              eg I-280, CA-85 -->
         <xsd:element name="RoadName" type="xsd:string"/>

         <!-- Required Field: Direction of roadway along which this sensor
              is measuring speeds. The direction is defined in relation to
             the conventions for that roadway rather than compass
             direction -->
         <xsd:element name="RoadDir" type="xsd:string"/>

         <!-- Required Field: Latitude of sensor (decimal notation) to 6
              decimal places -->
         <xsd:element name="Latitude" type="xsd:double"/>

         <!-- Required Field: Longitude of sensor (decimal notation) to 6
              decimal places -->
         <xsd:element name="Longitude" type="xsd:double"/>

         <!-- Optional Field: County in which sensor is located -->
         <xsd:element name="County" type="xsd:string" minOccurs="0"/>

         <!-- Optional Field: Absolute post-mile location of sensor -->
         <xsd:element name="PostMile" type="xsd:string" minOccurs="0"/>

         <!-- Optional Field: Source of data for this sensor -->
         <xsd:element name="DataSource" minOccurs="0">
           <xsd:simpleType>
             <xsd:restriction base="xsd:string">
               <xsd:enumeration value="SPEEDINFO"/>
               <xsd:enumeration value="CALTRANS"/>
               <xsd:enumeration value="PEMS"/>
               <xsd:enumeration value="FIVE_ONE_ONE"/>
```

```
                <xsd:enumeration value="PUBLIC"/>
                <xsd:enumeration value="UNKNOWN"/>
                <xsd:enumeration value="OTHER_1"/>
                <xsd:enumeration value="OTHER_2"/>
                <xsd:enumeration value="OTHER_3"/>
                <xsd:enumeration value="OTHER_4"/>
                <xsd:enumeration value="OTHER_5"/>
                <xsd:enumeration value="OTHER_6"/>
                <xsd:enumeration value="OTHER_7"/>
                <xsd:enumeration value="OTHER_8"/>
                <xsd:enumeration value="OTHER_9"/>
                <xsd:enumeration value="OTHER_10"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
```

8.0 Speed Data

```
<!-- Device (sensor) data -->
   <xsd:element name="device-data">
      <xsd:complexType>
         <xsd:sequence>
            <!-- Required Field: Time of speed measurement in local time
                 eg local time for the Bay Area is UTC-7 -->
            <xsd:element name="DateTimeStamp" type="date-timestamp"/>

            <!-- Required Field: Unique sensor identifier called out in the
                 device description data -->
            <xsd:element name="Id" type="xsd:string"/>

            <!-- Required Field: The current status of this sensor. If this
                 is OUT-OF-SERVICE, the speed data for this sensor at this
                time should be ignored -->
            <xsd:element name="Status">
              <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                  <xsd:enumeration value="OK"/>
                  <xsd:enumeration value="OUT-OF-SERVICE"/>
                </xsd:restriction>
              </xsd:simpleType>
            </xsd:element>

            <!-- Required Field: Degree of confidence in the speed measurement.
                 This field is currently set to 100% for all measurements. It
                is a 'placeholder' for future developments -->
            <xsd:element name="Confidence" type="confidence-factor"/>

            <!-- Required Field: Speed data broken out by lane. The lane type
                 is currently set to 'ALL' for all measurements meaning that
                the speed provided is averaged over all lanes. Again, this
                is a 'placeholder' for possible future developments -->
            <xsd:element name="Lane" type="lane-data" maxOccurs="10"/>
```

```
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
```

9.0 Properties File

Included with the java test client is a properties file that allows the user to customize certain aspects of the data delivery.

9.1 Measurement Latency

Low latency between speed measurement and delivery to a customer is a key benefit of the SpeedInfo Data Service. By default, speed data is updated on a feed every 60 seconds. However, the latency can be decreased further by specifying an **UpdateInterval** of as little as 15 seconds in the properties file.

9.2 Feed Name

A customer may be subscribed to more than one feed. This will require a corresponding number of client invocations. For a particular client instance, the feed is specified by the **FeedName** parameter in the properties file.

Appendix B

NCDOT TIMS Web Service Interface

NCDOT TIMS Web Service Interface

Revised 4/17/06

The test version of the SOAP web service is located at the following URL:

http://appsqc.dot.state.nc.us/TIMS/WebService/SensorInput.asmx

There is a WSDL file available at this URL for generating client-side proxy classes. Also, please do not hard-code this URL, since it will change when we switch to the production servers.

Each of the web service methods is protected by a password parameter that should be set to the following value: "{FFC7EBBF-714F-4900-895E-2AC9FA6F8C9F}". This password will change in the future.

The web service provides the following methods:

InsertSensorData(string password, int sensorId, DateTime timestamp, int averageSpeed)

Store a single data entry from a speed sensor.

Example Code:

```
SensorInput   svc = new SensorInput();

svc.InsertSensorData("{FFC7EBBF-714F-4900-895E-2AC9FA6F8C9F}", 4, DateTime.Now,
45);
```

InsertSensorDataList(string password, SensorData[] sensorDataList)

Store an array of speed sensor data entries all at once.

Example Code:

```
SensorData[]  sensorList = new SensorData[3];

sensorList[0] = new SensorData();
sensorList[0].SensorId = 1;
sensorList[0].SensorTimeStamp = DateTime.Now;
sensorList[0].AverageSpeed = 42;

sensorList[1] = new SensorData();
sensorList[1].SensorId = 2;
sensorList[1].SensorTimeStamp = DateTime.Now;
```

```
sensorList[1].AverageSpeed = 43;

sensorList[2] = new SensorData();
sensorList[2].SensorId = 3;
sensorList[2].SensorTimeStamp = DateTime.Now;
sensorList[2].AverageSpeed = 44;

SensorInput   svc = new SensorInput();
svc.InsertSensorDataList("{FFC7EBBF-714F-4900-895E-
2AC9FA6F8C9F}", sensorList);
```

Currently, the database contains sensors with SensorId values that range from 1-40 as outlined in the Excel spreadsheet that was sent to NCDOT. These SensorId values may be changed in the future so that they will be properly synced with the SensorId values that are used internally by PBS&J.

ChangeSensorId(string password, int oldSensorId, int nNewSensorId)

Change a sensor's ID number without adjusting other information about the sensor. This method should be called when a sensor is assigned a new sensor ID value, but it is not moved to a new location. When this method is called, the old sensor ID will be deactivated and the new sensor ID will become active at the same location.

Appendix C

Hosting Servers Statistics and Configuration

Hosting Servers Statistics and Configuration
Update: 8/17/06

The NCDOT SpeedInfo Website system four separate servers to collect, monitor, manage, and disseminate the SpeedInfo speed data. Figure 1 shows the processes that operate on each server.

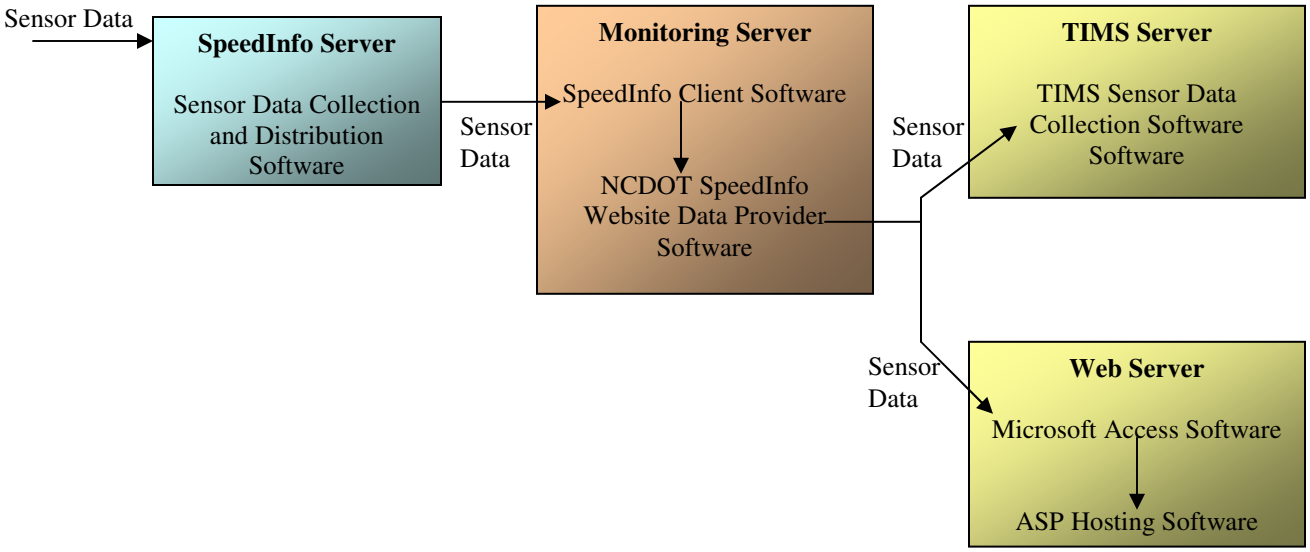


Figure 1: Hosting Servers Data Flow

The following lists each of the servers. The SpeedInfo Server is hosted and maintained by SpeedInfo. The TIMS Server is hosted and maintained by the North Carolina Department of Transportation (NCDOT). The Monitoring Server and Web Server are hosted by GoDaddy.com and maintained by PBS&J. All servers contain contact information. For the Monitoring and Web Servers, login and connectivity information is listed.

SpeedInfo Server

Primary Technical Contact
Aidan Smyth
asmyth@speedinfo.com
(408) 446-7664

Secondary Technical Contact
Ari V Krish
akrish@speedinfo.com

Administrative Contact
Ron Goodall
rgoodall@speedinfo.com

TIMS Server

Primary Technical Contact
Tim Kingsbury
tkingsbury@dot.state.nc.us

Secondary Technical Contact
T. Josh Collins
tjcollins@dot.state.nc.us

Administrative Contact
Jo Ann Oerter
joerter@dot.state.nc.us
(919) 233-9331 x233

Monitoring Server

Primary Technical Contact
John Hope
johnhope@pbsj.com
(407) 806-4147

Secondary Technical Contact
Armand Ciccarelli
armandciccarelli@pbsj.com
(407) 806-4187

Administrative Contact
Michael Berman
michaelberman@pbsj.com
(510) 459-0045

Hosting Technical Contact
Host: GoDaddy.com
Account Number: 4822248
Support Number: (480) 505-8877
Call-in PIN: 7687
Account Management Website: http://www.godaddy.com
Account Management Login: pbsj511
Account Management Password: pbsjjsbp

Server Statistics
Processor: Intel Celeron 2.0 GHz
Allotted Monthly Bandwidth: 500 GB
Disk Space: 240 GB (120 GB usable w/ RAID 1)

Server Configuration
Server Name: 511MonitoringP
Public IP Address: 68.178.194.120
Remote Desktop Login: pbsj511
Remote Desktop Password: Pbsj7319

Web Server

Primary Technical Contact
John Hope
johnhope@pbsj.com
(407) 806-4147

Secondary Technical Contact
Armand Ciccarelli
armandciccarelli@pbsj.com
(407) 806-4187

Administrative Contact
Michael Berman
michaelberman@pbsj.com
(510) 459-0045

Hosting Technical Contact
Host: GoDaddy.com
Account Number: 4822248
Support Number: (480) 505-8877
Call-in PIN: 7687
Account Management Website: http://www.godaddy.com
Account Management Login: pbsj511
Account Management Password: pbsjjsbp

Server Statistics
Processor: Unknown
Allotted Monthly Bandwidth: 500 GB
Disk Space: 50 GB

Server Configuration
Domain: ncdot-traffic.com
Public IP Address: 64.202.189.133
FTP Site: ftp://ftp.ncdot-traffic.com/
FTP Login: ncdot
FTP Password: Pbsj7319