TRAUMATIC BRAIN INJURY OPTIONS
Web application

By

SRUTHI NAGULAVANCHA

A REPORT

Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2010

Approved By:

Major Professor
Dr. Daniel Andresen

Abstract

According to the Division of Injury Response, Centers for Disease Control and Prevention, approximately 1.4 million Americans sustain a traumatic brain injury each year. The aim of the project is to create a web interface to link survivors, family members, and caregivers of individuals suffering from traumatic brain injuries (TBI) to potentially helpful agencies or service centers within their local communities. Often the TBI service centers located in the remote places are difficult to trace hence this website mainly concentrates on small rural centers which are located in Kansas State.

The portal will offer two-dimensional and basic information about traumatic brain injury centers and specifically about access of resources. Within the portal, a link to an interactive map will be provided. A form for data entry helps the service centers to publish about their presence and the regions they serve. A search distance feature is also added into the website which interactively searches the nearest latitude, longitude values (TBI service center) to the user's location by using the haversine formula.

Timeline:

I plan to complete this project by the end of Summer2010 semester.

Tools and Technologies:
Microsoft Visual Studio for .NET 2008 (For using ASP.NET and C#.NET)
Oracle 10g
Geo Coder services
JMeter

Table of Contents

Acknowledgements	IV
1. Introduction	1
1.1 Goal	1
1.2 Scope	1
1.3 Need of the application	2
1.4 Related work and problems	2
1.5 Motivation	3
2. System Requirement Analysis	3
2.1 Information Gathering	3
2.2 System Feasibility	3
2.2.1 Economic Feasibility	3
2.2.2 Technical Feasibility	3
2.2.3 Behavioral Feasibility	4
3. Design	5
3.1 Use case Diagram	6
3.2 Class Diagram	7
3.3 Database diagram	8
4. Functionality	9
5. Implementation	9
6. Testing	16
7. Project Evaluation	29
8. Results and Challenges	29
9. Conclusions	29
9.1 Limitations	30
9.2 Scope for Future Work	30
10. References	31

Acknowledgements

I would like to express my deepest gratitude to Dr Daniel Andresen for his encouragement, support, guidance and his valuable advices during the course of my work.

I would like to thank Dr Torben Amtoft and Dr Mitchell Neilsen for serving in my committee and for their valuable cooperation during this project.

Introduction:

1.1 Goal

The goal of this application is to create a web interface to link survivors, family members, and caregivers of individuals suffering from traumatic brain injuries (TBI) to potentially helpful agencies or service centers within their local communities. This project may be viewed as a strategy to promote successful living for survivors, family members, and caregivers in their communities at various points along the continuum of recovery. Research to date indicates that no such individualized program for survivors, family members, and caregivers of individuals with TBI is currently in existence. This project seeks to construct knowledge and improve quality of life for survivors, family members, and caregivers of TBI.

1.2 Scope

The application can be used to search for the TBI service centers from various part of “Kansas” state. The geographical regions within the state of Kansas that will appear on the map have already been identified through consultation with the two state agency partners and include 44 cities, many of them rural (for example, Ellis, population 1900; Hill City, population 1600; Marysville, population 3100). The feature of interactive map can be used by the survivors, family members, and caregivers of individuals with TBI or it can also be used for the purpose of research to find the number of centers located in the Kansas state and the kind of services they offer.

1.3 Need of the application

According to the Division of Injury Response, Centers for Disease Control and Prevention, approximately 1.4 million Americans sustain a traumatic brain injury each year. Of these individuals, those 75 or older have the highest rate of hospitalization and deaths, and, falls are the leading cause of TBI. One of three older adults (those 65+) will experience a fall in any given year (CDC, n.d.). In 2002, those aged 65+ comprised 7% of the world's population. By 2050, that number is expected to rise to nearly 17% (U.S. Census Bureau, 2004, March). Thus, addressing the needs of individuals with traumatic brain injury and their families is an emerging *older adult* issue. Additionally, the CDC in the report, *Traumatic Brain Injury in the United States: Emergency Department Visits, Hospitalizations, and Deaths,* states that "increased access to services for those who do not fully recover must continue to be national priorities". Thus this information gives an idea about the need for the current application.

1.4 Related Work and Problems

The problem if a single resource center serves more than one location in the Kansas state was a challenge to be implemented. Collection of information from different rural centers of Kansas and putting them on the website was also a difficult task.

1.5 Motivation

This application helped me to gain a lot of experience on various technologies like ASP.NET (C#), Oracle 10g, Microsoft Virtual Earth API, java script and Geo coder web services. If given a chance in future, I would add the google earth feature to the distance calculation module,

improve the performance of the application to handle more number of users, extend the application for other states of USA and also take care of the usability issues associated with the heavy loading of the pushpins.

2. System Requirement Analysis

2.1 Information Gathering

For the development of this project I collected requirements from the human ecology department of Kansas state university and took a regular feed from Dr Andresen, Associate Professor, CIS Department.

Other than this, I did a lot of research on various other methods of building this application which helped to incorporate a few stronger features into the application. The tools and controls used in the application are recommended ASP.NET controls by Microsoft which improves the navigation and map generation to a great extent.

2.2 System Feasibility

The system feasibility can be divided into the following sections:

2.2.1 Economic Feasibility

The project is economically feasible as the only cost involved is having a computer with the minimum requirements mentioned earlier. For the users to access the application, the only cost involved will be in getting access to the Internet.

2.2.2 Technical Feasibility

To deploy the application, the only technical aspects needed are mentioned below:

Operating Environment: Win XP/Vista

Platform .Net 3.5 Framework & IIS

Database Oracle 10g

For Users:

Internet Browser

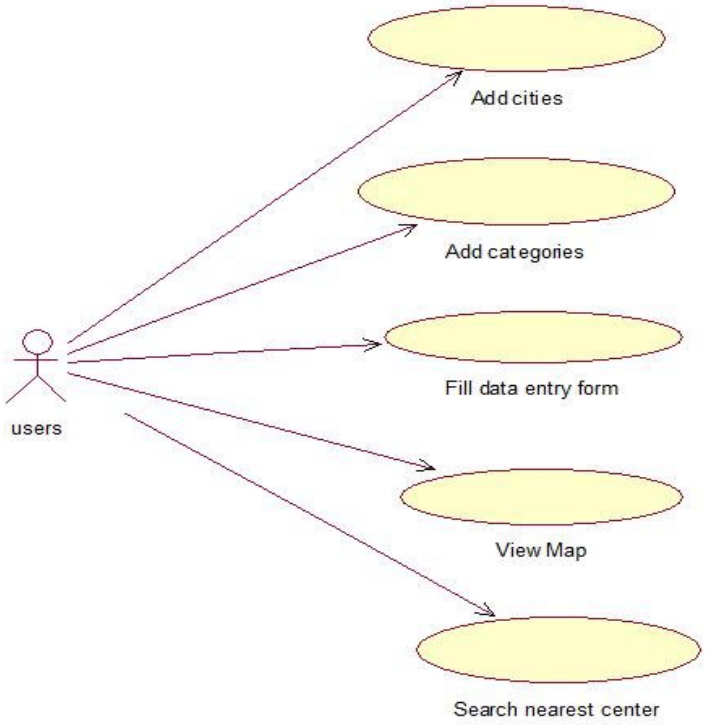
Internet Connection

2.2.3 Behavioral Feasibility

The application requires no special technical guidance and all the views available in the application are self explanatory.

3. Design

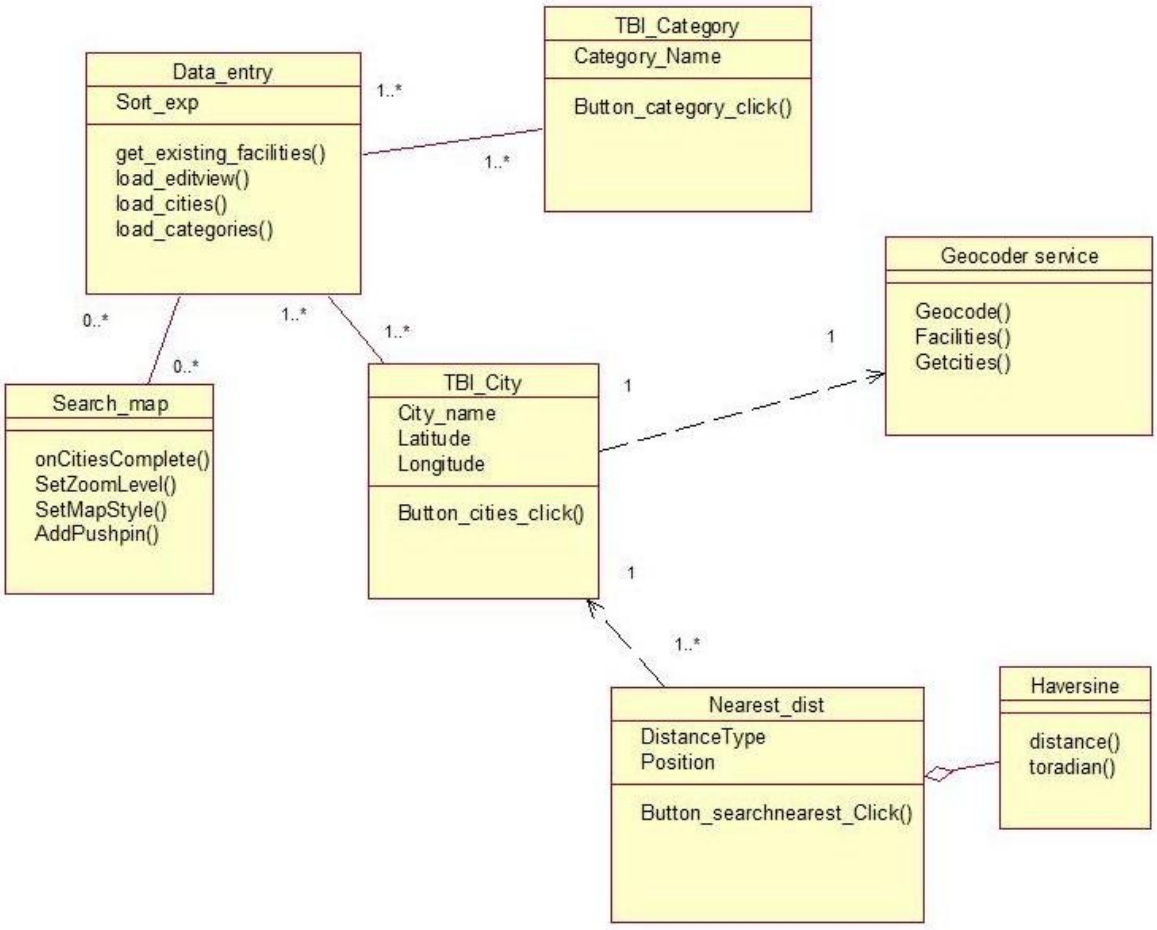
3.1 Use case diagram: The Use case Diagram describes the system functionality from an external observer's view. It shows who can do what. The following is the use case diagram for my application.



The above use case diagram indicates the interaction of user with different modules of the application.

3.2 Class Diagram:

Below is the class diagram that shows the various classes that I have used. It gives the structure of the system.



The above class diagram shows that

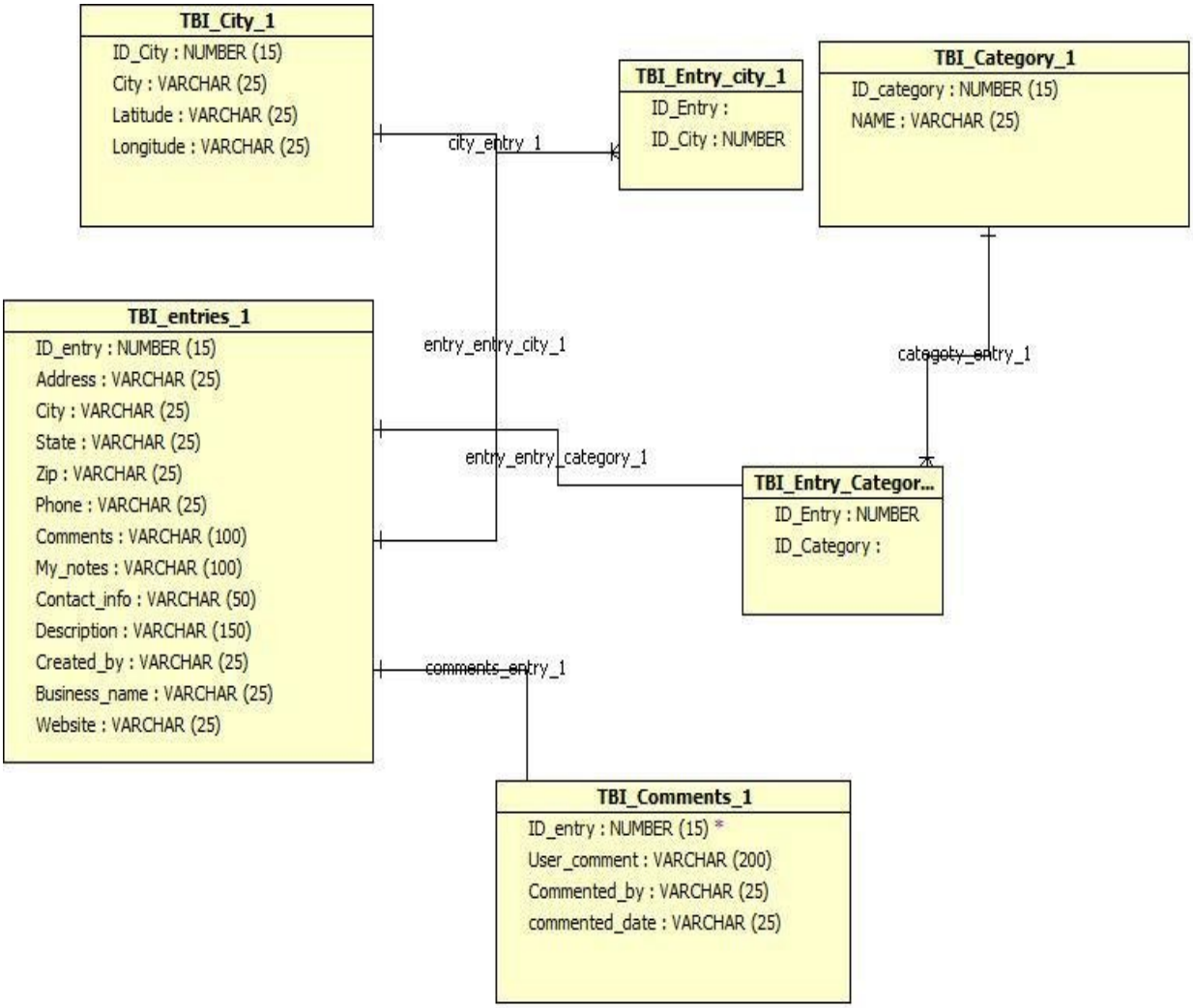
1) The classes TBI_City and TBI_Category have an association relationship with the Data_enty class. The cardinality mentioned is 1 to many because there can be multiple entries with the same city name, but there has to be a particular city name to enter at least 1 entry on that name.
2) There is association relationship with a cardinality of 0 to many between search_map and data_entry because even if there are no entries in the database, map can be viewed and if there are many entries then they popup using pushpins. There is also a possibility for multiple entries hence it is 0 to many.

3) There is a dependency relation between TBI_city and Geocoder service class because each cities attributes latitude and longitude values are calculated using the geocode() of Geocoder class.
4) There is a dependency relation between TBI_city and Nearest_dist class because each cities attributes latitude and longitude values are calculated using the geocode() of Geocoder class.
5) There is an aggregation relationship between Nearest_dist and Haversine class because haversine calculation is a part of Nearest_dist.

3.3 Database Design:

The database diagram below shows how the tables are related. The logic involved here is that I maintain 2 reference tables called TBI_Entry_City and TBI_Entry_Category. The reference table contains information about which city has which entry and similarly with categories. Multiple entries can have same city name because each resource center can serve people from multiple nearby cities.

So ID_city is a primary key in the “TBI_City” table, ID_entry is the primary in the “TBI_Entry_city” table, ID_entry is the primary key in the “TBI_Entries table”. Other tables relations and tables mentioned are understandable.



4. Functionality

Modular Approach

Following are all the modules designed for the TBI centers web application:

Cities / categories entry module: This module starts with the collection of information about different cities for which the TBI website serves. It also involves collecting information about the categories in to which the service centers can be divided. After collection, a web page is created to add / view cities and categories.

View Entries module: This module gives a display of the entries (information about different resource centers) that are currently in the database. It also has a search feature which allows the user to search for the entries based on the city name or category name.

Add Entry module: This module helps the resource center people to enter information about their business location, contact information and also about the services that they offer.

View Map: This module involves designing map using Java script and Microsoft virtual earth API. Using geo coder services and pushpins to locate a particular center and display all the categories for which particular center offer services.

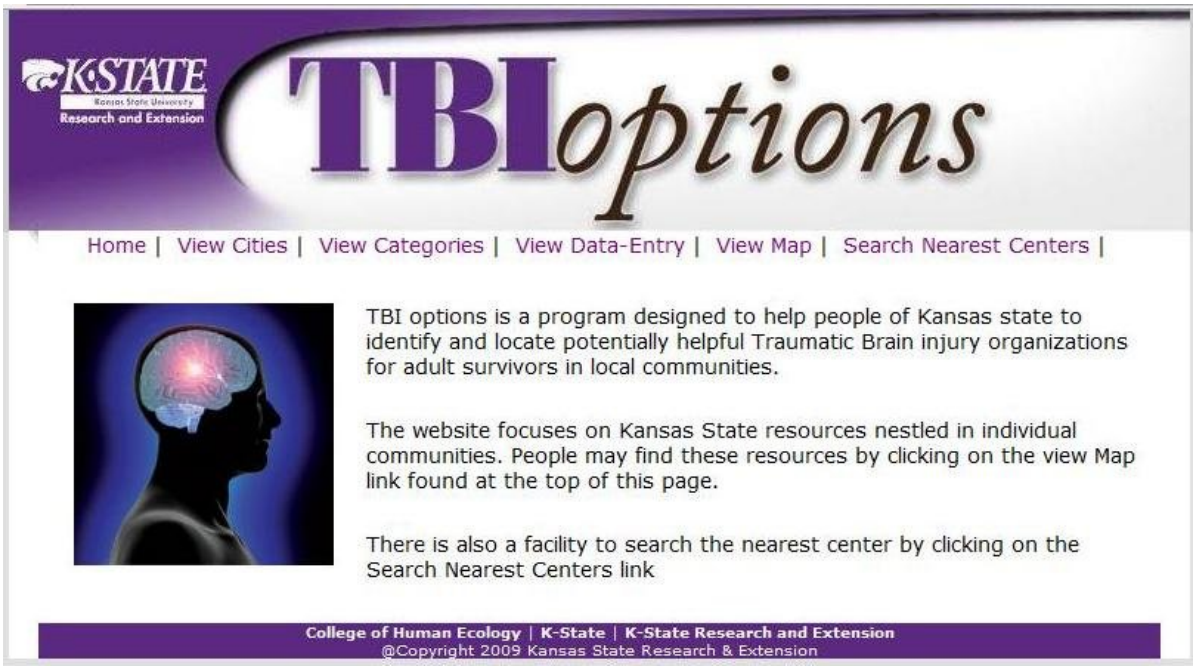
Distance calculator: This module helps the user to find out nearest resource center to his location. This is done by coding haversine formula using C#. This formula is used to calculate the shortest distance over the earth's surface by making use of the latitude and longitude values.

5. Implementation

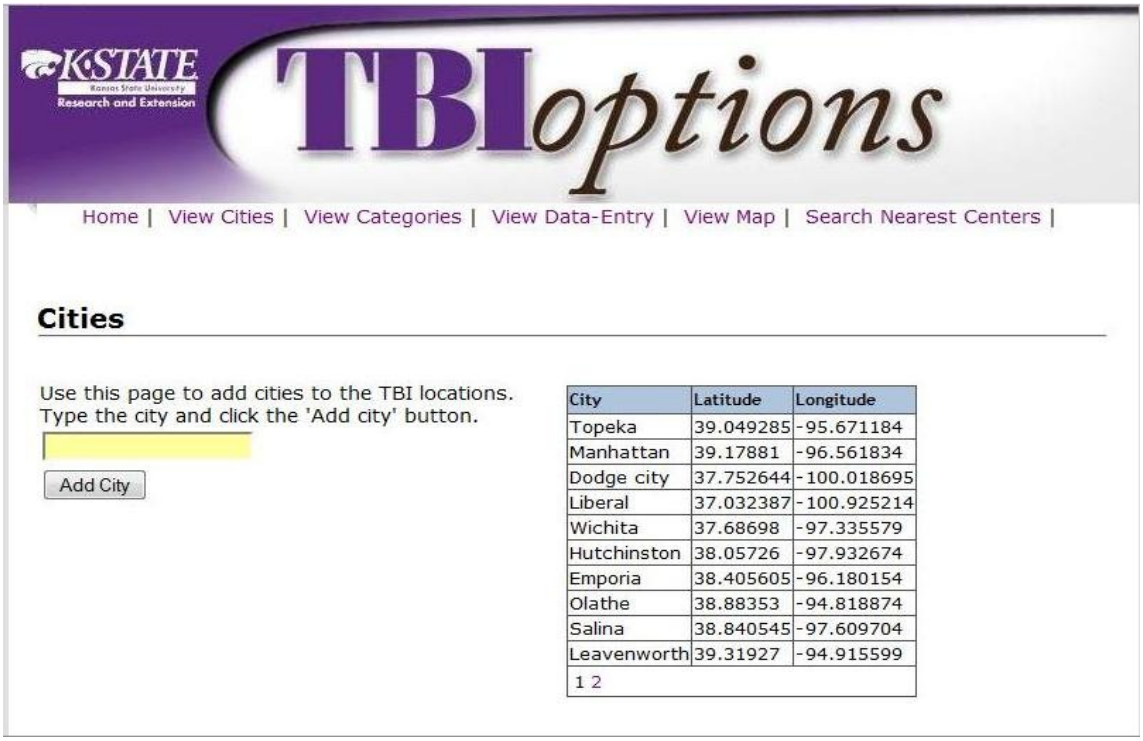
5.1 User Interface Design and Implementation

The user interface of the application has been designed using Microsoft Visual Studio 2008. The main controls used in the design are Grid View control, Menu Control and these controls are provided with ASP.NET 2008. Following are the screenshots of the user interface.

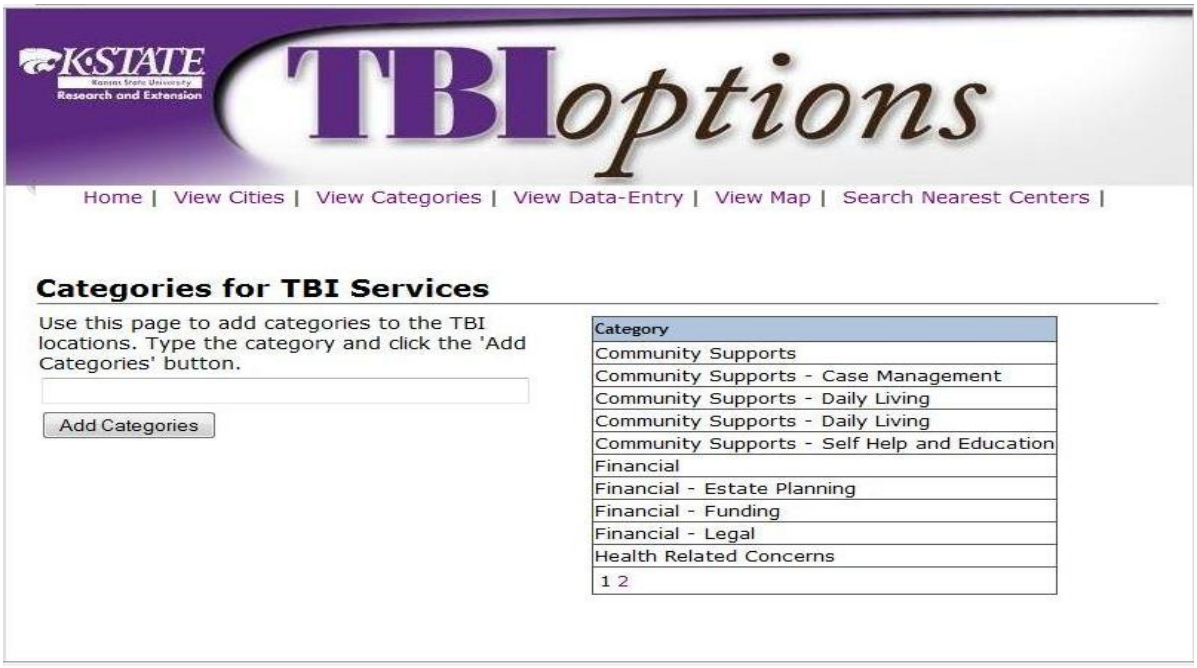
Following is the screen shot of home page of the website:



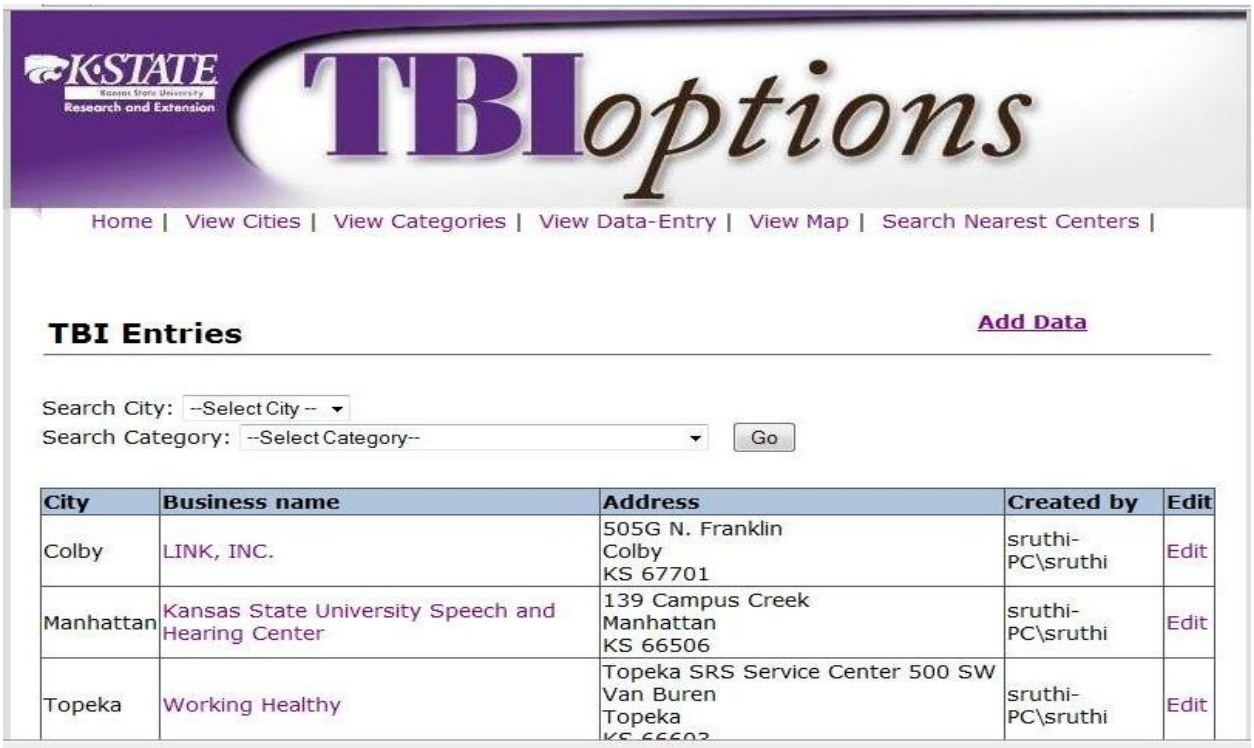
Following is the screen shot of view / add cities of the website:



Following is the screen shot of view / add categories of the website:

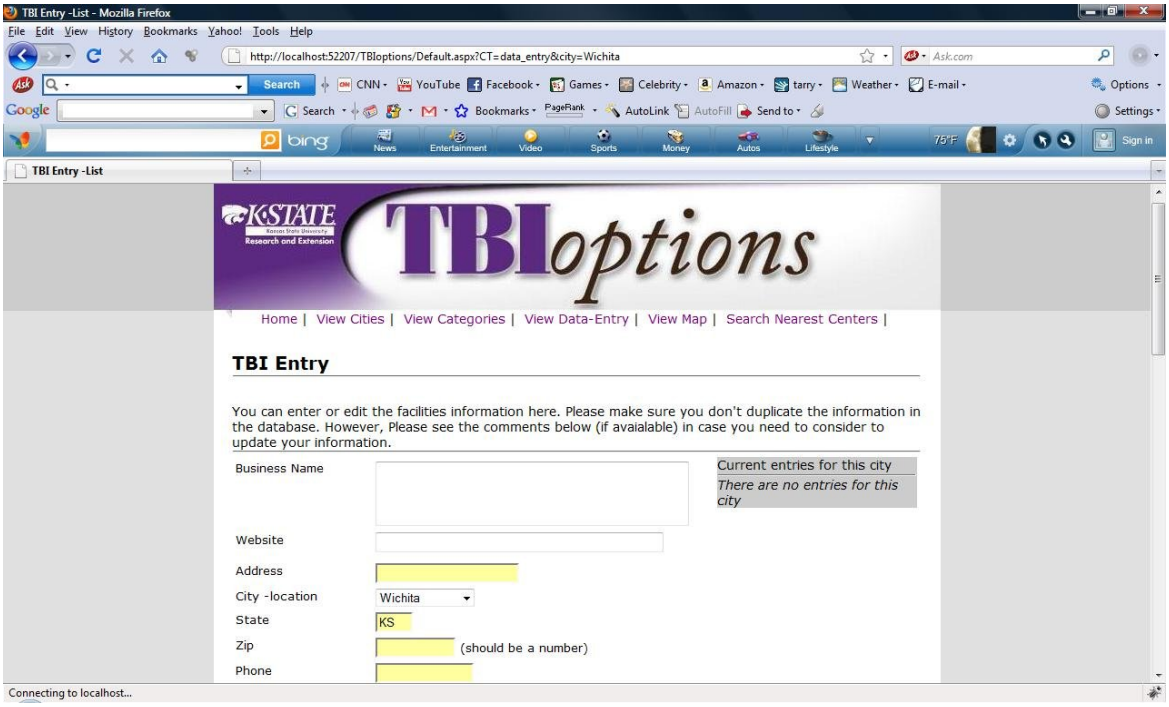


Following is the screen shot of view/ search TBI entries of the website:



The above page helps to view all the entries present in the database. Entries can be searched based on the city or category name.

Following is the screen shot of add TBI entries of the website:

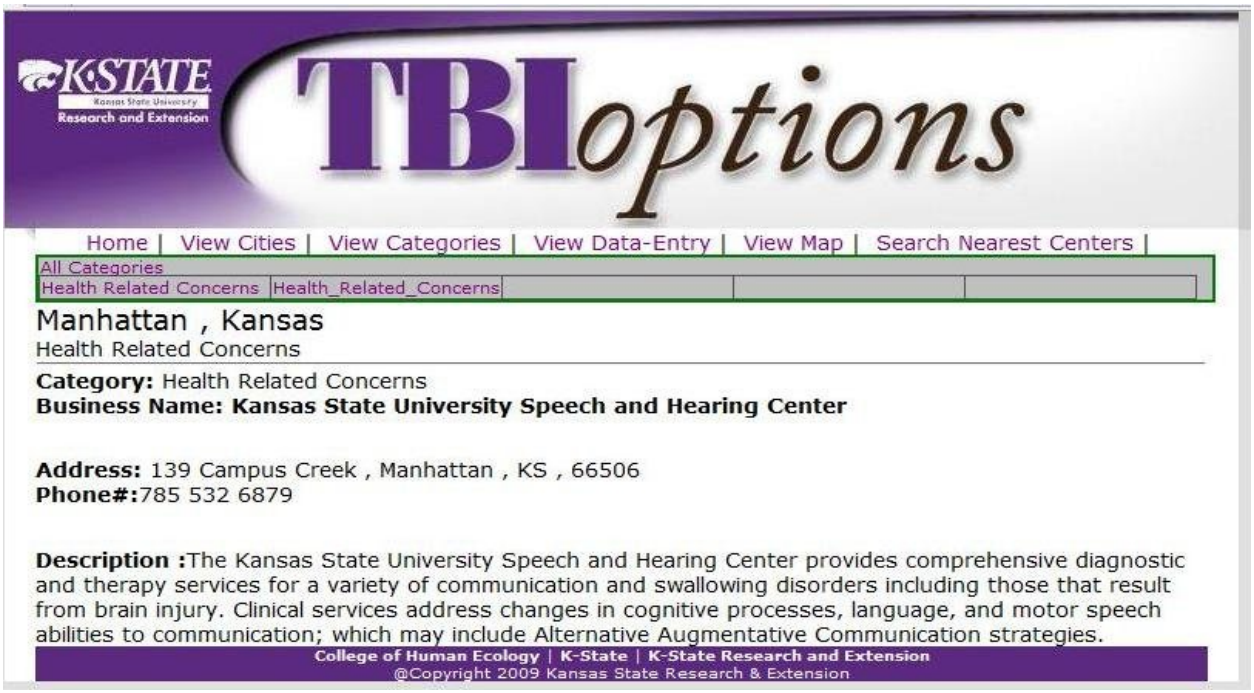


The above page helps TBI service center's to enter their details and let people of KS state know about their presence and the type of services they offer. I used HTML editors to design this page to help people to enter description, comments.

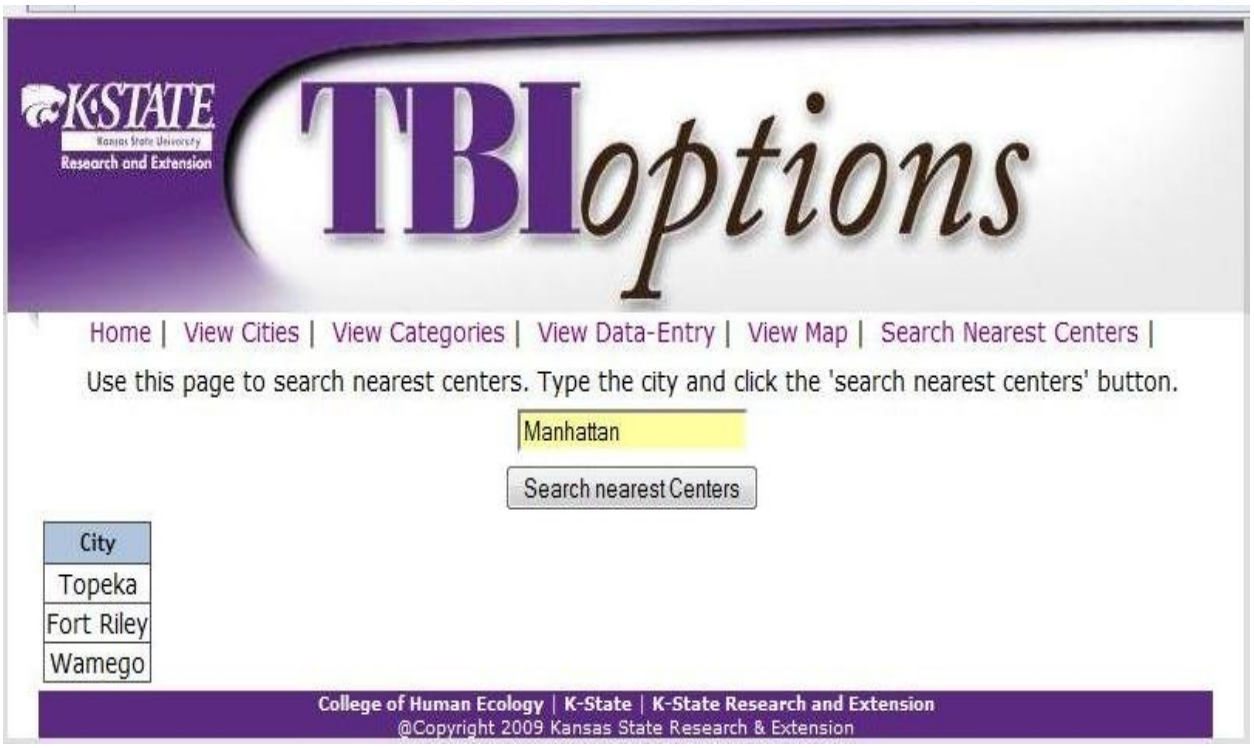
Following is the screen shot of view map of the website:

The above page is used to view map and the kind of services each city offers. When you click on the category in the pushpin then the information about the center which offers that service and brief information about that center is displayed as shown in the next screen shot.

Following is the screen shot of info / comments page of the website:



Following is the screen shot of search nearest TBI center page of the website:



The above page is used to calculate the nearest centers to peoples location. The example shown above is for Manhattan. So the nearest service centers to Manhattan are located in Topeka, Fort Riley and Wamego.

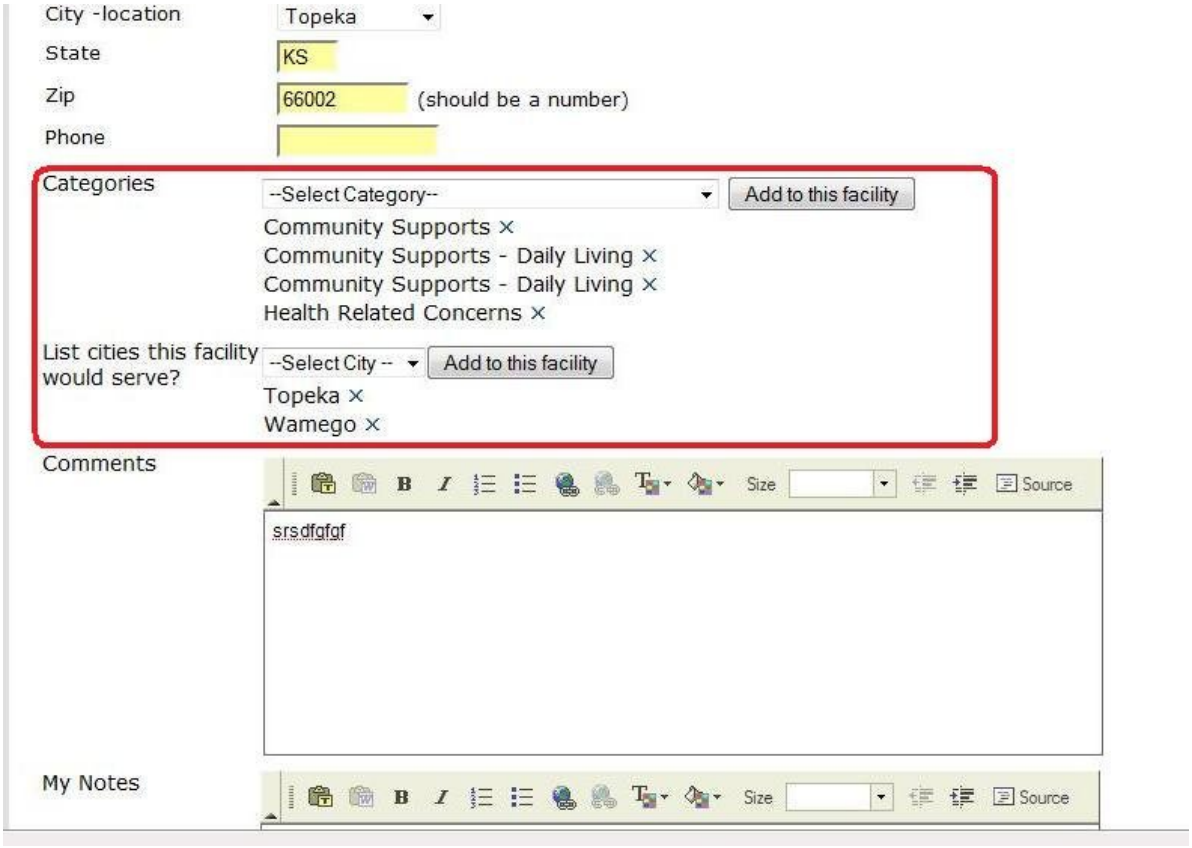
5.2 Technical Discussion

Interactive map in the web site is created using Java script, Microsoft Virtual earth API and geocoderservice.asmx which is a local ASP.NET web service. Some of the inbuilt methods like VEMap(), VELatLong(), VEShape(), VEClor(), AddPushpin() of Virtual earth are used.

There is always a possibility for the increase in the number of cities, categories to be added. Hence the dropdown lists are dynamically loaded from the database. Even the grid views are given Paging option so that when records are added then automatically grid view grows with the page numbers displayed below.

In the add data page there is a facility to add and delete multiple categories from the dropdown lists dynamically. This feature helps users to save time while creating an entry because user has to enter the details of the center only once and add all the categories and cities related to that center at the same time.

Screen shot of the Feature:



6. Testing

6.1 Software Testing : Software Testing is an empirical investigation conducted to provide Stake holders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software.

6.2 Unit Testing: In unit testing each module is tested alone in an attempt to discover any errors in the code, i.e. all executable source code will be subject to test. It verifies that certain

parts of the code are working properly. The tests will consist of all classes and Web forms as the base unit. The NUnit, a testing tool for C#, can be used as the testing tool for unit testing.

6.3 Performance Testing: This will be performed to test the entire system to see whether all driving requirements are satisfied. Allowing multiple users log into the system and perform the operations at the same time using the JMeter testing tool. This test verifies that the components of the systems meet the stated requirements for speed.

6.4 Manual Testing: Manual Testing of the system was done to ensure the correctness of various parts of the code.

6.5 Item Pass/Fail Criteria: Test cases executed on the TBI Options will pass if they meet the specific requirements as mentioned in the Vision Document. A test case will fail if any behavioral expectation is not met as described

6.6 Environmental Needs:

6.6.1 Hardware: All tests will be conducted on an Intel-based machine.

6.6.2 Software: Microsoft Visual Studio 2008, NUnit and JMeter testing software will be used as the testing environment.

6.6.3 Operating System: All tests to be performed will run under the Windows Vista platform.

6.7 Testing Results Summary :

	Test Case #	Description	Results

User	T-01	**Add city – to calculate latitude and longitude**	Passed
	T-02	**Add Category – to add the name of the category and paging of the grid view**	Passed
	T-03	**Search city and category functionality**	Passed
	T-04	**All the controls on Add entry page and the feature of multiple cities using same category**	Passed
	T-05	**View map with pushpins and redirecting information in the pushpin**	Passed

	T-06	**Search functionality which calculates the nearest center based on the distance.**	Passed

6.8 Detailed Test Results

Manual Testing

6.8.1 Test Case 01 - Add city – to calculate latitude and longitude

Test Unit	Test Case	Result
Add city – to calculate latitude and longitude	A city name is entered by the user and add city button is clicked	**Latitude and longitude values of that particular city are calculate using geo coder services code and they are displayed in the grid view.**

6.8.2 Test Case 02 - Add Category – to add the name of the category and paging of the grid view

Test Unit	Test Case	Result

Add category – to add a category name to the grid view	A category name is entered by the user and add category button is clicked	**Category is added to the grid view and display an error message if it already exists. Also tested paging feature of the grid view.**

6.8.3 Test Case 03 - Search city and category functionality

Test Unit	**Test Case**	**Result**
Search city and category functionality	From the drop down lists user can select a city name or category name and search the records based on that name.	**Based on the database query select records are displayed in the grid view.**

6.8.4 Test Case 04 - All the controls on Add entry page and the feature of multiple cities using Same category

Test Unit	**Test Case**	**Result**

All the controls on Add entry page and the feature of multiple cities using same category	Different controls like HTML editors, text boxes, buttons and so on	**All the controls and their functionality is tested**

6.8.5 Test Case 05 - View map with pushpins and redirecting information in the pushpin

Test Unit	Test Case	Result
View map with pushpins and redirecting information in the pushpin	User accessing the View map option in the website	**All the cities which are added have a pushpin and contains link to the centers**

6.8.6 Test Case 06 - Search functionality which calculates the nearest center based on the distance

Test Unit	Test Case	Result
Search functionality which calculates the nearest center based on the distance	User accessing the search option in the website	**Search functionality works with the logic of " haversine formula" to calculate nearest distance using latitudes and longitudes**

Manual Testing of the system was done to ensure the correctness of various part of the code.

6.9 Performance Testing

Performance testing is testing that is performed, from one perspective, to determine how fast some aspect of a system performs under a particular workload. It can also serve to validate and verify other quality attributes of the system, such as scalability, reliability and resource usage.

The tool, Jakarta JMeter, is used to simulate the virtual users (clients) and test the performance of the system. It may be used to test performance both on static and dynamic resources. JMeter can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types.

The peak number of users (threads) that I have used is 500

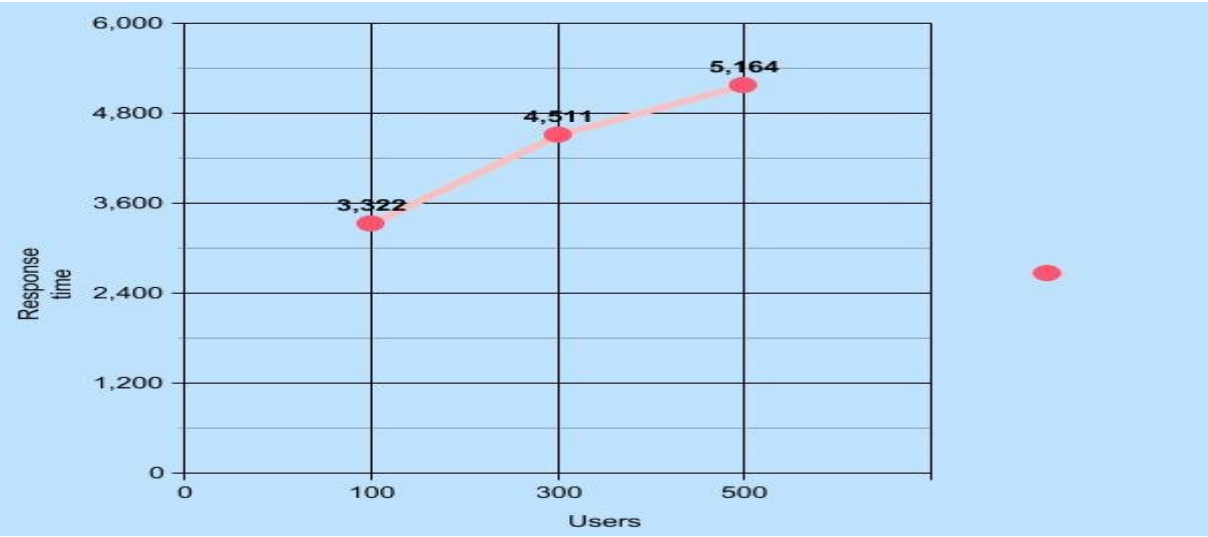
Home Page : [http://localhost:52207/TBIoptions/Default.aspx**]**

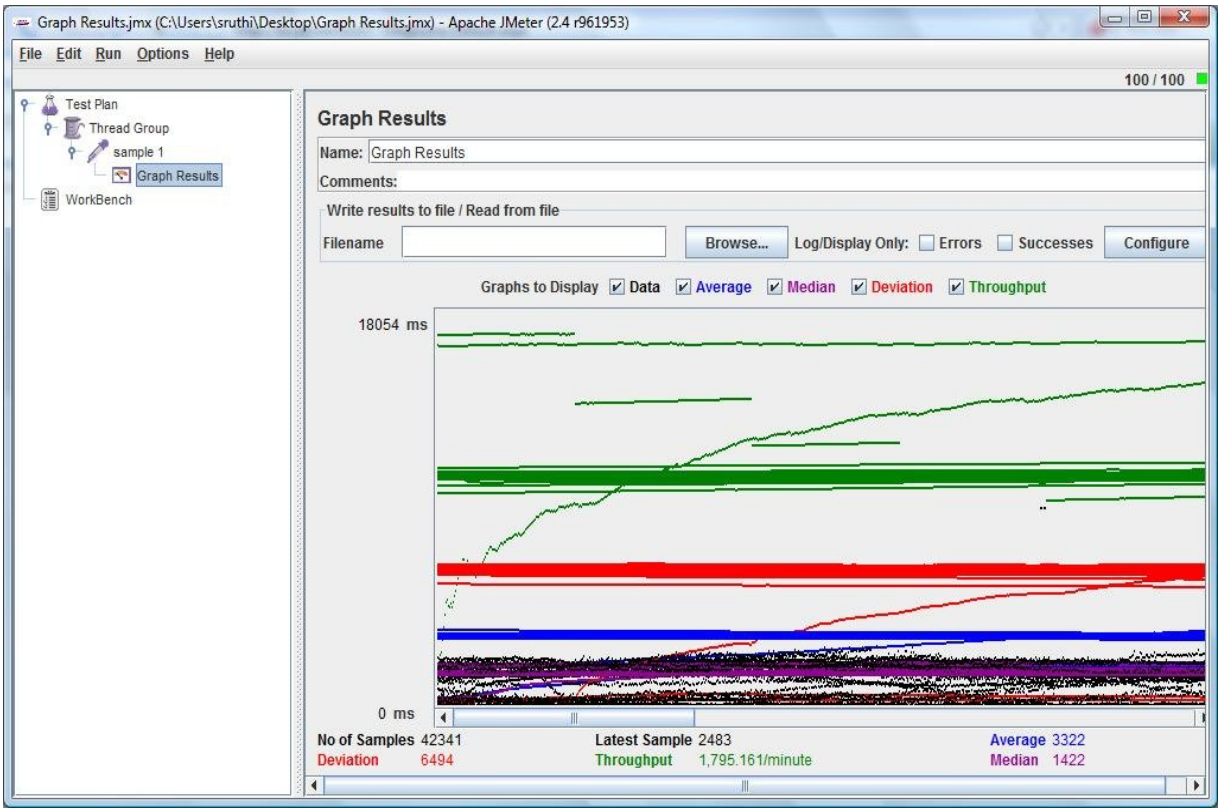
The response time for different combinations of users and requests is noted. In the following test, the number of loop-count is kept constant at 20,000 and the users are increased at intervals. The maximum number of users is limited to 500.

A graph is plot for number of requests (X-axis) versus the response time (Y-axis).

Users	Loop Count	Ramp-up period (sec)	Average Response Time (ms)
100	20000	5	3322
300	20000	5	4511
500	20000	5	5164

A graph is plot for number of users (X-axis) versus the response time (Y-axis)



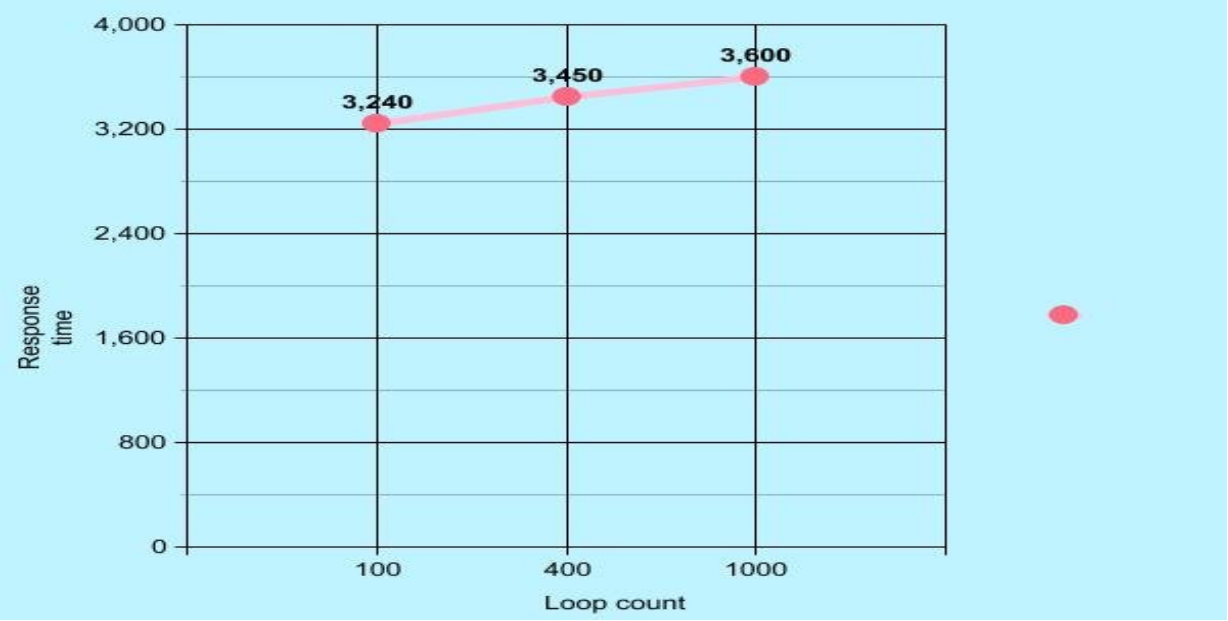


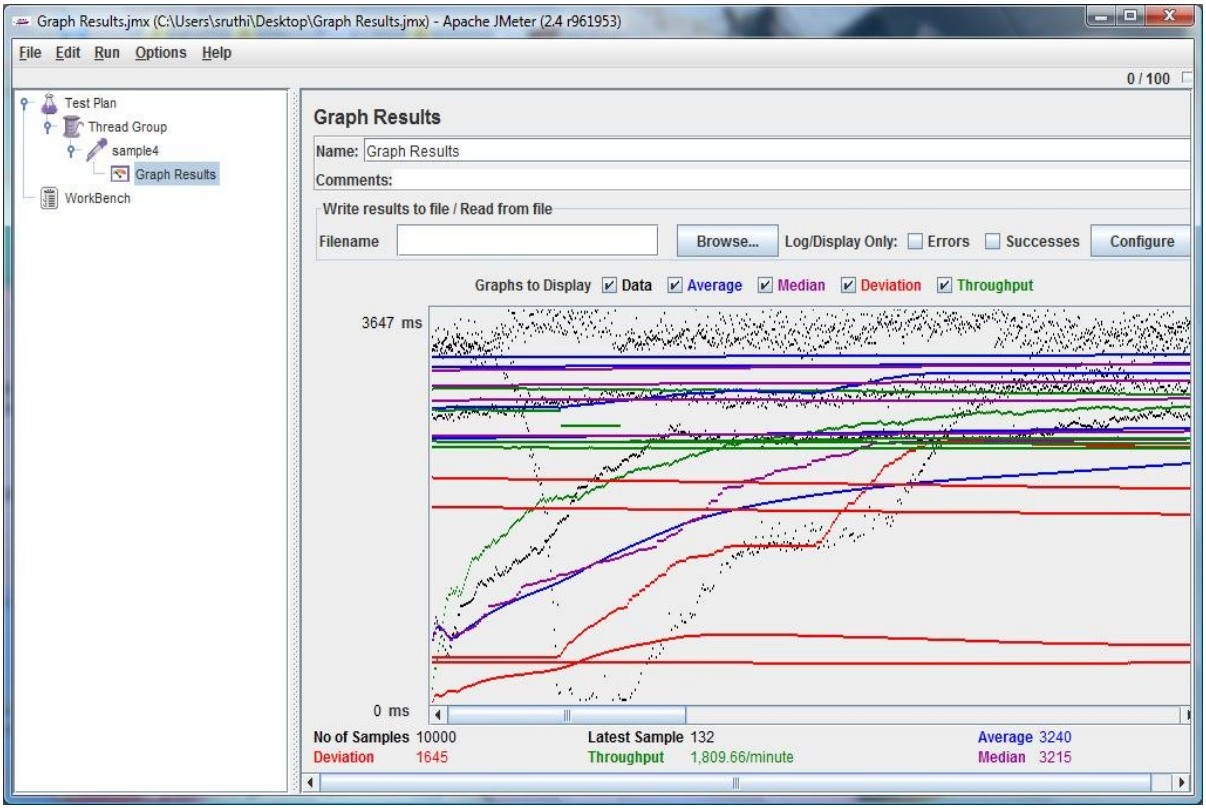
Home Page: [http://localhost:52207/TBIoptions/Default.aspx]

In the following test, the number of users is kept constant and the number of requests is increased and a graph is plot for loop count (X-axis) versus the response time (Y-axis).

Users	Loop Count	Ramp-up period (sec)	Number of Requests	Average Response Time (ms)
100	100	5	10,000	3240

100	400	5	40,000	3450
100	1000	5	1,00,000	3600



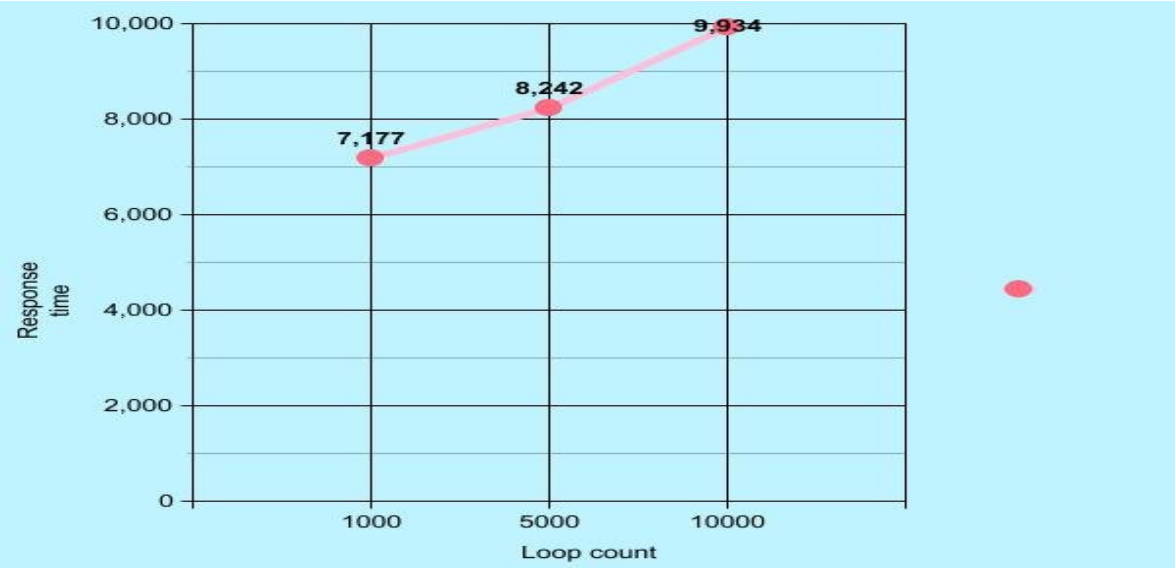


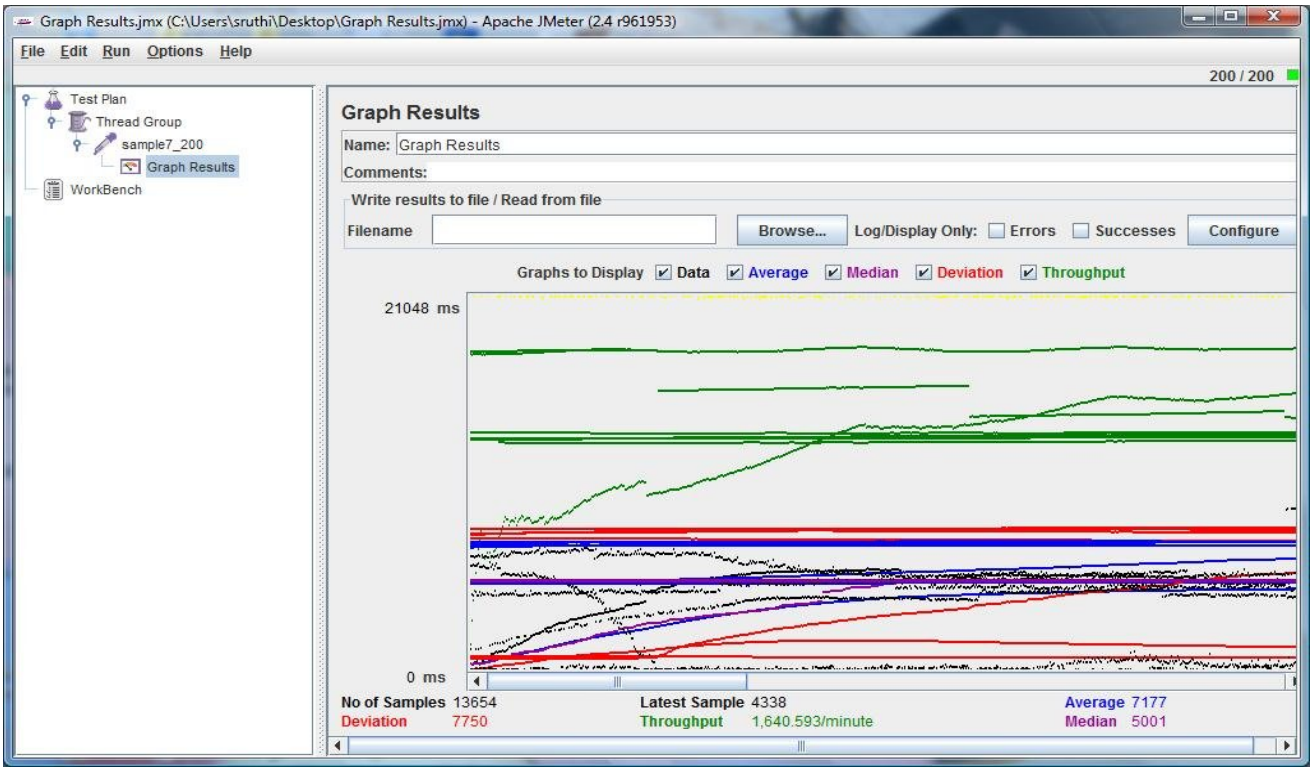
Home Page : [http://localhost:52207/TBIoptions/Default.aspx]

In the following test, the number of users is kept constant at 200 and the number of requests is increased up to 20,00,000 and a graph is plot for loop count (X-axis) versus the response time(Y-axis).

Users	Loop Count	Ramp-up period (sec)	Number of Requests	Average Response Time (ms)
200	1000	5	2,00,000	7177

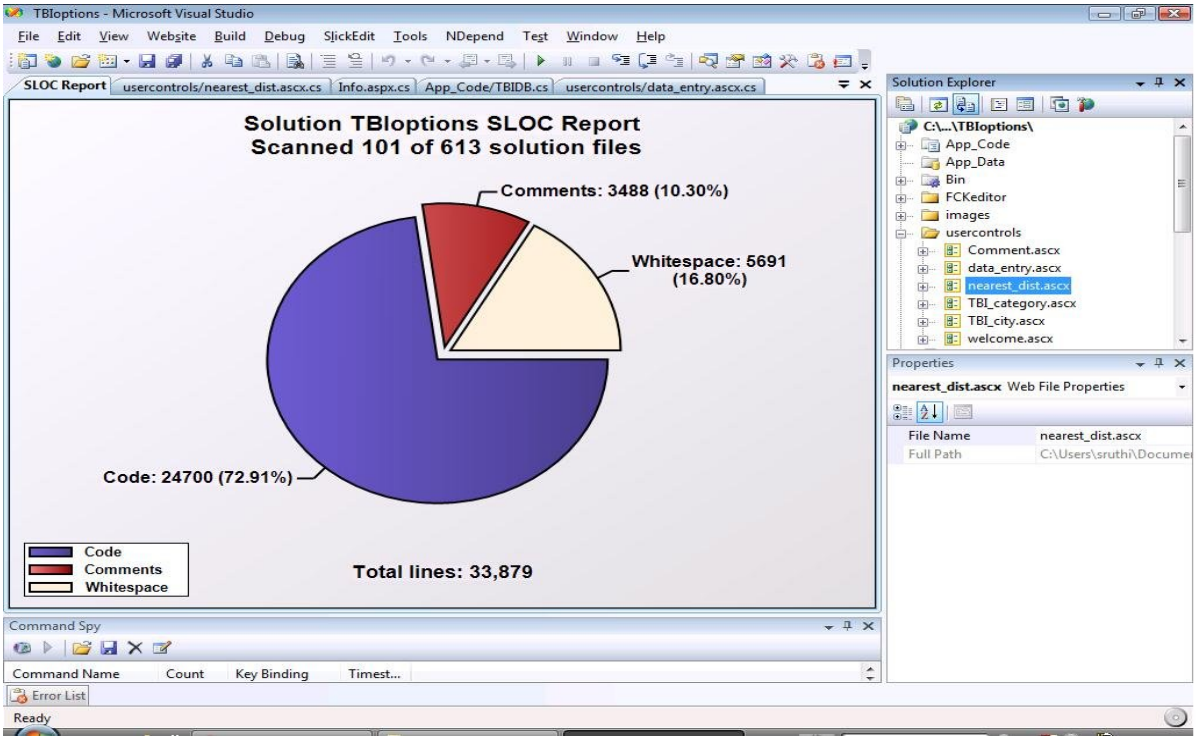
200	5000	5	10,00,000	8242
200	10000	5	20,00,000	9934




Graph Results.jmx (C:\Users\sruthi\Desktop\Graph Results.jmx) - Apache JMeter (2.4 r961953)
File Edit Run Options Help
200 / 200
Test Plan
Thread Group
sample7_200
Graph Results
WorkBench
Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Graphs to Display Data Average Median Deviation Throughput
21048 ms
0 ms
No of Samples 13654
Latest Sample 4338
Average 7177
Deviation 7750
Throughput 1,640.593/minute
Median 5001

7. Project Evaluation

Number of lines of code for the project is evaluated using a tool called SlickEdit which is a compatible tool for ASP.NET. It gives a graphical representation of the distribution of the actual code, white spaces and comments.



8. Results & Challenges

The application is being used by the human ecology department of Kansas state university to conduct research on number of participants who are being benefited by the use of the TBI resource location website.

8.1 Challenges

- Understanding the client requirements and creating a data entry web form for them to enter the details of the TBI resource centers.

- Next challenge is to learn to implement geo coder services to calculate latitude and longitude values and use it with C# for creating maps.
- The last but important challenge is to setup the application on the server and get it running. Because I was always involved in coding the application but was never into bringing up the application live.
- Learning technologies with little/no guidance.

The overall idea of doing this project is to get a real time experience. Learn new technologies

9. Conclusions

TBI options web application is designed to help the people of Kansas state to locate TBI service centers which are located in some remote areas. This application helped me to gain a lot of experience on various technologies like ASP.NET (C#), Oracle 10g, Microsoft Virtual Earth map, java script and Geo coder services.

9.1 Limitations:

Although the application is able to serve the people to find different TBI service locations, it is only limited to Kansas state. Hence the application can be extended to provide same kind of service to all the other states of United states.

9.2 Scope for Future Work:

The following things can be done in future.

- The application can be modified to resolve the above issues.
- The map module can be designed further to avoid the slow pushpin popping up.

10. References

- All about Microsoft controls in C#
  http://www.msdn.microsoft.com/

- All about Oracle 10g
  http://st-curriculum.oracle.com/tutorial/DBXETutorial/index.htm

- Wikipedia for various diagrams & testing methods
  http://www.wikipedia.org/

- Geo coding with virtual earth web service
  http://blogs.msdn.com/b/virtualearth/archive/2008/09/29/geocoding-with-the-virtual-earth-web-service.aspx

- Performance Testing with JMeter
  http://www.devx.com/webdev/Article/17950