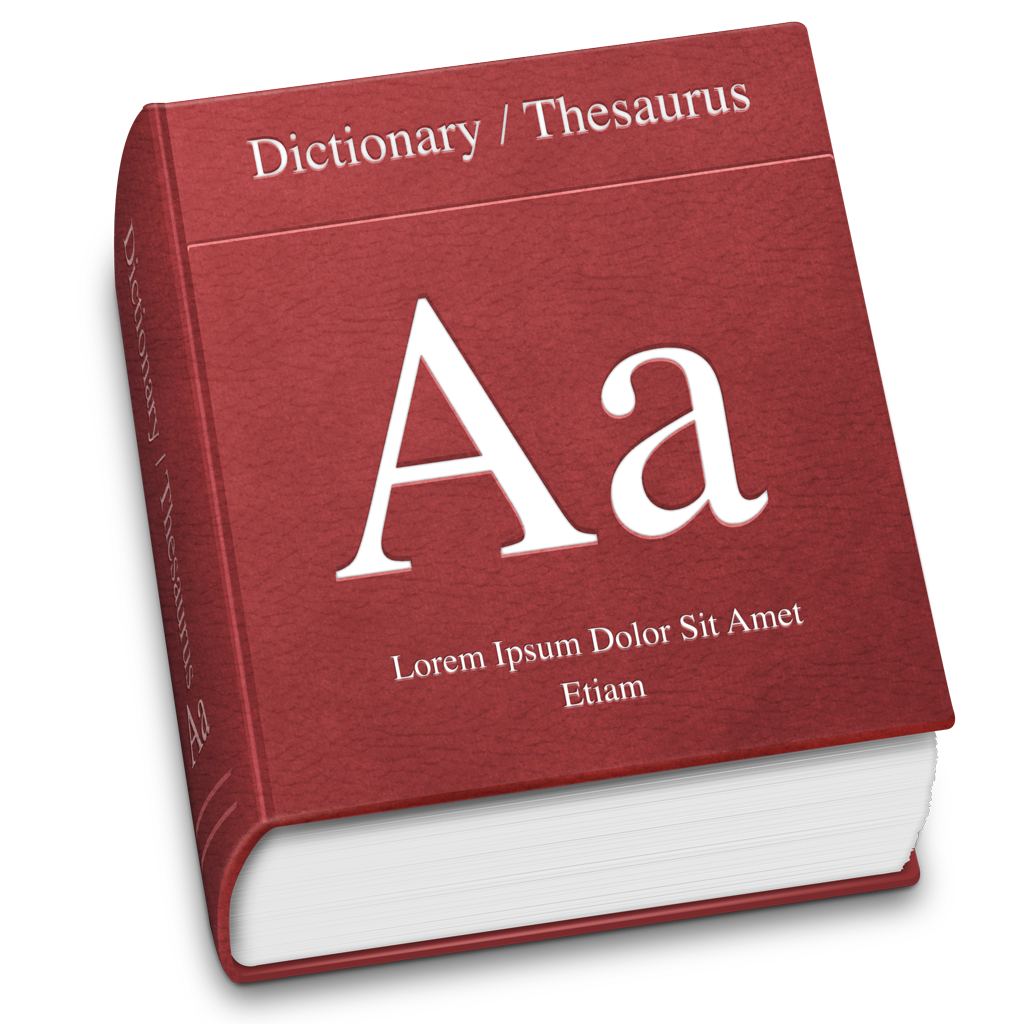
2013

Dictionary

****

Aman Oberoi

Bharti Chavan

Priyanka Pradhan

**DEPARMENT OF COMPUTER ENGINEERING**

**SARASWATI COLLEGE OF ENGINEERING**

**PLOT NO. 46/46A, SECTOR NO 5, BEHIND MSEB SUBSTATION, KHARGHAR,NAVI MUMBAI-410210**

**Tel. : 022-27743706 to 11 \* Fax : 022-27743712 \* Website: www.sce.edu.in**

**2012-2013**

**20220220**

****

**A**

**PROJECT REPORT ON**

**DICTIONARY**

SUBMITTED BY

NAME ROLLNO

**AMAN OBEROI 29**

**BHARTI CHAVAN 09**

**PRIYANKA PRADHAN 35**

**SARASWATI EDUCATION SOCIETY’S**

**SARASWATI COLLEGE OF ENGINEERING**

**(Approved by AICTE, recg. By Maharashtra Govt. DTE ,Affiliated to Mumbai University)**

**PLOT NO. 46/46A, SECTOR NO 5, BEHIND MSEB SUBSTATION, KHARGHAR,NAVI MUMBAI-410210**

**Tel. : 022-27743706 to 11 \* Fax : 022-27743712 \* Website: www.sce.edu.in**

**Certificate**

*This is to certify that the requirements for the project report entitled “DICTIONARY” has been successfully completed by the following students:*

Roll numbers Name

1)29 Aman Oberoi

2)09 Bharti Chavan

3)35 Priyanka Pradhan

In partial fulfillment of Sem –V, **Bachelor of Engineering of Mumbai University in Third Year of Computer Engineering** of Saraswati college of Engineering , Kharghar during the academic year 2012-13.

**Subject I/C**  **External Examiner**

Prof. Ruhi Ghormade

ACKNOWLEGDMENT

We wish to express our sincere gratitude to **Prof. Ruhi Ghormade** of Saraswati College of Engineering, Kharghar for providing us an opportunity to do OOSE project work on **“DICTIONARY”.**

This project bears on imprint of many people.

We sincerely thank to all those people for guidance and encouragement in carrying out this project work we also wish to express our gratitude to the officials and other staff members, who rendered their help during the period of our project work.

Our special thanks to our classmates for their kind co-operation to the completion of our project work. Last but not least we wish to avail ourselves of this opportunity, express a sense of gratitude and love to our friends and our beloved parents for their manual support, strength, help and for everything.

ABSTRACT

In earlier days, people used to search for words in the dictionary manually which was time consuming as well as a tiresome and tedious job. With the development in computers and their software technologies it now became very easy to use a dictionary or find synonyms for a particular word.

Our main aim is to develop a dictionary wherein it includes the functionalities which helps the user get the information all at one place. Also the interface is simple and self-explanatory. Following the requirements and needs we have developed this application.

**INDEX**

1. **INTRODUCTION**
2. **PROBLEM STATEMENT**
3. **PROPOSED SOLUTION**
4. **DESIGN ANALYSIS**
5. **ANALYSIS WORKFLOW**

* **Structural**
* **Behavioural**

1. **IMPLEMENTATION WORKFLOW**
2. **FUTURE SCOPE**
3. **CONCLUSION**
4. **REFERENCES**

**INTRODUCTION**

We, the students of third year engineering – Computer Science, have been entrusted with the Project of developing an application with proper data base, where in the application should be able to connect with the database.

We have desired to make a DICTIONARY now called as WordWideWeb where basic definition of dictionary means word reference, wordbook, lexicon or vocabulary i.e. a collection of words often listed alphabetically with usage information, definitions, phonetics, pronunciation and other information. We have provided the user with the functionalities so that the information desired can be obtained all at one place. The application consists of a wordlist containing all the words along with its synonyms, antonyms, part of speech of the word. Also there is provision to learn a new word everyday with word of the day option as also the user can bookmark the words for any future reference. Additional feature of browsing the internet to fetch word meanings is also added.

For the creation of this application we have made use of Microsoft Visual Studio and SQLite for the database. The entire application has been programmed using C#.

**PROBLEM STATEMENT**

**To design a DICTIONARY application.**

The problem given to us was to design a **dictionary** which is now named as WordWideWeb. It is very important to maintain efficient software to handle information of various words. The problem was to create an application that keeps all information about words such as synonyms, antonyms, part of speech etc. all in one place.

The specification of this problem was to build a **dictionary** displaying and accessing all the data that may be needed by the user, the means for the user to find meanings of words they are searching for.

**PROPOSED SOLUTION**

In our proposed solution we are going to provide solutions to all the above mentioned problems. The website provides quick search results from the database for the synonyms, antonyms or part of speech of the word. It also has the capability to display all the words from the database together for the user’s reference and can also access these them separately.

Our application also provides extra features like word of the day where each day a new word will be displayed that can help the user to improve vocabulary. Along with that it can also search for the nearest words to the searched word in the dictionary and also browse the internet for the synonyms of the word. The user can also bookmark the words according to his/her need.

**DESIGN ANALYSIS**

**EER Diagram**

The E-R diagram denotes the relation between the various entities stored in the database. This project stores information about words and their synonyms, antonyms. Each word has its own unique ID as a primary key.

The ER diagram to denote the system is as shown below:-

Parts of speech

bookmark

antonym

synonym

d

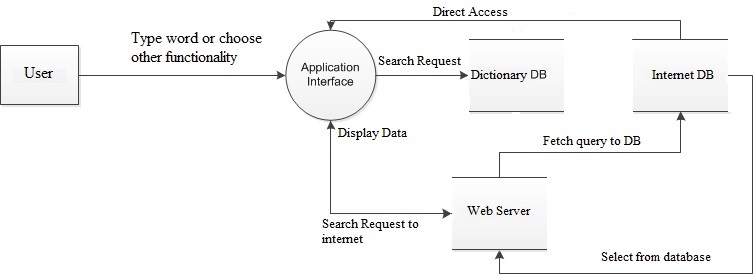
word

enter

user

**Data Flow Diagram:**

The data flow diagram of the system is as shown below:-



**Data Dictionary**

**Data Dictionary of Synonym**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Format | Size | Description | Example |
| ID | Integer |  | Size of integer | This is a unique ID of words, also acts as a primary key. | 1,2,3, etc. |
| Word | Varchar |  | 15 characters long | This is the word which can be searched for. | ‘abase’,’tenure’, etc. |
| Meaning | Varchar |  | 100 characters long | This contains the synonym of the word which is searched. | ‘lower,humiliate’,’term or period’,etc. |

**Data Dictionary of Antonym**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Format | Size | Description | Example |
| ID | Integer |  | Size of integer | This is a unique ID of words, also acts as a primary key. | 1,2,3, etc. |
| Word | Varchar |  | 15 characters long | This is the word which can be searched for. | ‘abase’,’concord’, etc. |
| Antonym | Varchar |  | 100 characters long | This contains the antonym of the word which is searched. | ‘raise,promote’,  ’disagreement’,etc. |

**Data Dictionary of PartOfSpeech**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Format | Size | Description | Example |
| ID | Integer |  | Size of integer | This is a unique ID of words, also acts as a primary key. | 1,2,3, etc. |
| Word | Varchar |  | 15 characters long | This is the word which can be searched for. | ‘abase’,’concord’, etc. |
| partofspeech | Varchar |  | 10 characters long | This contains the part of speech of the word which is searched. | ‘VERB’,’NOUN’,etc. |

**Data Dictionary for Bookmark:**

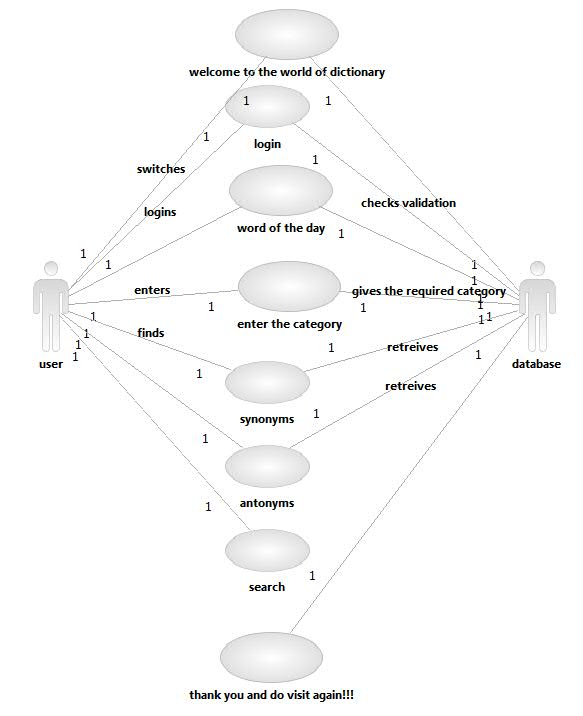
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Format | Size | Description | Example |
| ID | Integer |  | Size of integer | This is a unique ID of words, also acts as a primary key. | 1,2,3, etc. |
| Word | Varchar |  | 15 characters long | This is the word which can be searched for. | ‘abase’,’concord’, etc. |

**ANALYSIS WORKFLOW**

This section includes all the UML diagrams of the application DICTIONARY.

Structural and Behavioural UML diagrams

Use Case Diagram:



**Class Diagram:**

Bookmark

Id:int

Word:char

Parts of speech

Id:int

Word:char

Parts of speech:char

Antonym

Id:int

Word:char

Antonym:char

Synonym

Id:int

Word:char

Meaning:char

word

Type:int

Length:int

Language:char

Others

Id:int

Name:char

Student

Id:int

Name:char

User

Id:int

Name:char

Address:varchar

Phone no:int

Search()

**SEQUENCE DIAGRAM:**

Research word

database

word

Search a word

Search() enters()

[check==”true”]

Find()

Needs research()

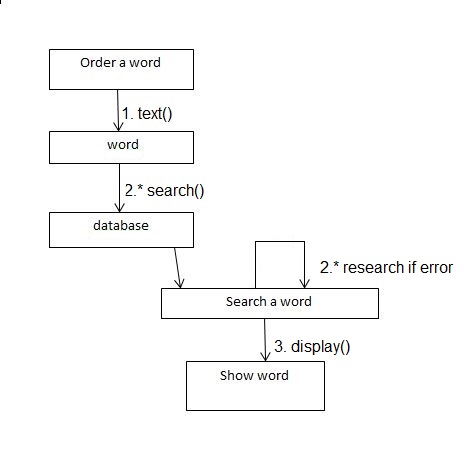
new

Show word

[chech ==”true”]

new

**COLLABORATION DIAGRAM:**

****

**ACTIVITY DIAGRAM:**

Ask for word

Refind word

Search word

Search word

Check word

Validate word

Received word

[rejected]

Cancel request

[succeeded]

**STATE CHART DIAGRAM:**

displaying

verifying

validatig

Searching word in database

Texting the word

unscheduled

search word

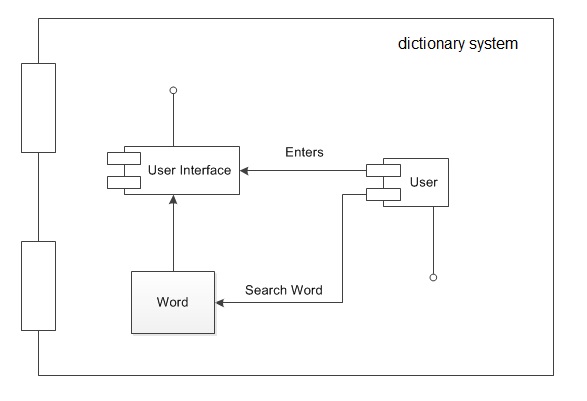
validate

verify

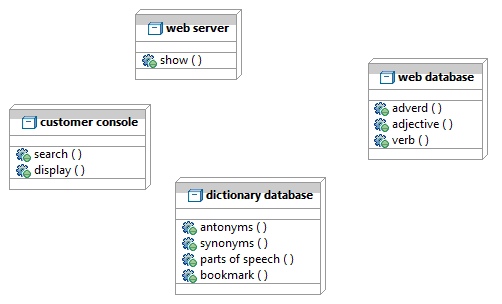
show

scheduled sign schedule

**COMPONENT DIAGRAM:**

****

**DEPLOYMENT DIAGRAM:**



**IMPLEMENTATION WORKFLOW**

Some part of the source code is given below.

**SOURCE CODE:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SQLite;

namespace WordWideWeb

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void searchbutton\_Click(object sender, EventArgs e)

{

SQLiteConnection conn = new SQLiteConnection(@"Data Source=e:/dictionary.s3db; Version=3;");

conn.Open();

SQLiteCommand cmd = new SQLiteCommand("Select \* from Synonym where word like @Pattern",conn);

cmd.Parameters.AddWithValue("@Pattern", searchbox.Text + "%");

SQLiteDataAdapter SDA = new SQLiteDataAdapter(cmd);

DataTable dt = new DataTable();

SDA.Fill(dt);

displaygrid.DataSource=dt;

this.webBrowser1.Navigate("http://google.com/search?q=" + searchbox.Text);

webBrowser1.ScriptErrorsSuppressed = true;

this.webBrowser2.Navigate("http://en.wikipedia.org/wiki/" + searchbox.Text);

webBrowser2.ScriptErrorsSuppressed = true;

}

private void button1\_Click(object sender, EventArgs e)

{

//NEAREST WORD BUTTON

SQLiteConnection conn = new SQLiteConnection(@"Data Source=e:/v.s3db; Version=3;");

conn.Open();

SQLiteCommand cmd = new SQLiteCommand("Select o2.word as PREVIOUS\_WORD,o1.word as CURRENT\_WORD,o3.word as NEXT\_WORD from Synonym o1,Synonym o2,Synonym o3 where o1.word like @Pattern AND o2.ID=(o1.ID-1) AND o3.ID=(o1.ID+1); ", conn);

cmd.Parameters.AddWithValue("@Pattern", searchbox.Text + "%");

SQLiteDataAdapter SDA = new SQLiteDataAdapter(cmd);

DataTable dt = new DataTable();

SDA.Fill(dt);

displaygrid.DataSource = dt;

}

private void button4\_Click(object sender, EventArgs e)

{

//DISPLAY ALL BUTTON

System.Data.SQLite.SQLiteConnection conn = new System.Data.SQLite.SQLiteConnection(@"Data Source=e:/dictionary.s3db; Version=3;");

System.Data.DataTable dt = new System.Data.DataTable();

SQLiteCommand cmd = new SQLiteCommand("Select s.id,s.word,s.meaning,a.antonym,p.partofspeech from Synonym s,Antonym a,PartOfSpeech p where s.id=a.id AND s.id=p.id;", conn);

System.Data.SQLite.SQLiteDataAdapter SDA = new System.Data.SQLite.SQLiteDataAdapter(cmd);

SDA.Fill(dt);

displaygrid.DataSource = dt;

}

private void button2\_Click(object sender, EventArgs e)

{

//Antonym button

SQLiteConnection conn = new SQLiteConnection(@"Data Source=e:/dictionary.s3db; Version=3;");

conn.Open();

SQLiteCommand cmd = new SQLiteCommand("Select \* from Antonym where word like @Pattern", conn);

cmd.Parameters.AddWithValue("@Pattern", searchbox.Text + "%");

SQLiteDataAdapter SDA = new SQLiteDataAdapter(cmd);

DataTable dt = new DataTable();

SDA.Fill(dt);

displaygrid.DataSource = dt;

}

private void button3\_Click(object sender, EventArgs e)

{

//part of speech button

SQLiteConnection conn = new SQLiteConnection(@"Data Source=e:/dictionary.s3db; Version=3;");

conn.Open();

SQLiteCommand cmd = new SQLiteCommand("Select \* from PartOfSpeech where word like @Pattern", conn);

cmd.Parameters.AddWithValue("@Pattern", searchbox.Text + "%");

SQLiteDataAdapter SDA = new SQLiteDataAdapter(cmd);

DataTable dt = new DataTable();

SDA.Fill(dt);

displaygrid.DataSource = dt;

}

private void button5\_Click(object sender, EventArgs e)

{

var myForm = new Form2();

myForm.Show();

}

private void randomWordToolStripMenuItem\_Click(object sender, EventArgs e)

{

Random r = new Random();

int i = (r.Next(4180));

SQLiteConnection conn2 = new SQLiteConnection(@"Data Source=e:/dictionary.s3db; Version=3;");

conn2.Open();

SQLiteCommand cmd2 = new SQLiteCommand("Select s.id,s.word,s.meaning,a.antonym,p.partofspeech from Synonym s,Antonym a,PartOfSpeech p where s.id=a.id AND s.id=p.id AND s.id = " + i, conn2);

SQLiteDataAdapter SDA2 = new SQLiteDataAdapter(cmd2);

DataTable dt2 = new DataTable();

SDA2.Fill(dt2);

displaygrid.DataSource = dt2;

}

private void displaygrid\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

//double click cell to bookmark

DataGridViewRow row = displaygrid.CurrentCell.OwningRow;

int i = int.Parse(row.Cells["ID"].Value.ToString());

SQLiteConnection conn = new SQLiteConnection(@"Data Source=e:/dictionary.s3db; Version=3;");

conn.Open();

SQLiteCommand cmd = new SQLiteCommand("Insert into Bookmark values ("+ i +")", conn);

SQLiteDataAdapter SDA = new SQLiteDataAdapter(cmd);

DataTable dt = new DataTable();

SDA.Fill(dt);

displaygrid.DataSource = dt;

MessageBox.Show("The word has been added to the bookmarks list.");

}

private void bookmarksToolStripMenuItem\_Click(object sender, EventArgs e)

{

var myForm = new Form3();

myForm.Show();

}

private void aboutToolStripMenuItem\_Click(object sender, EventArgs e)

{

var myForm = new Form4();

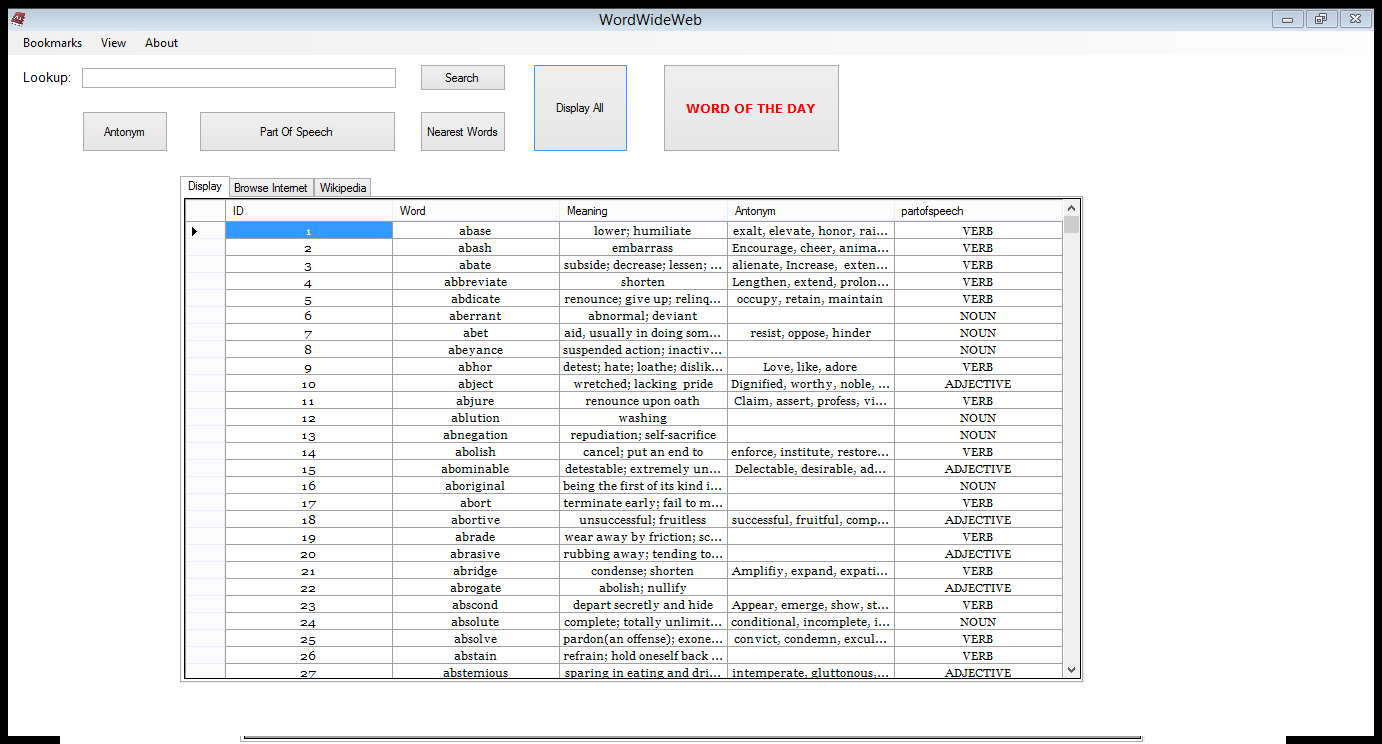
myForm.Show();

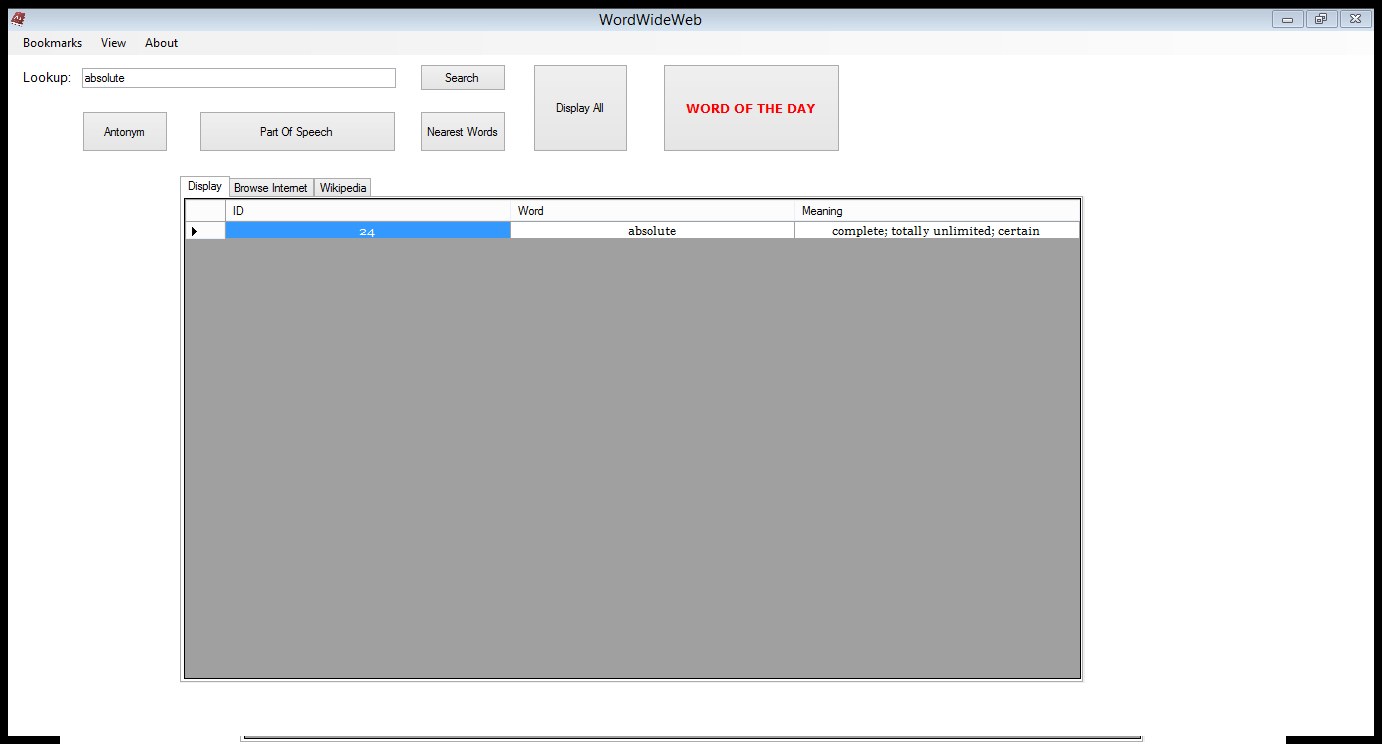
}

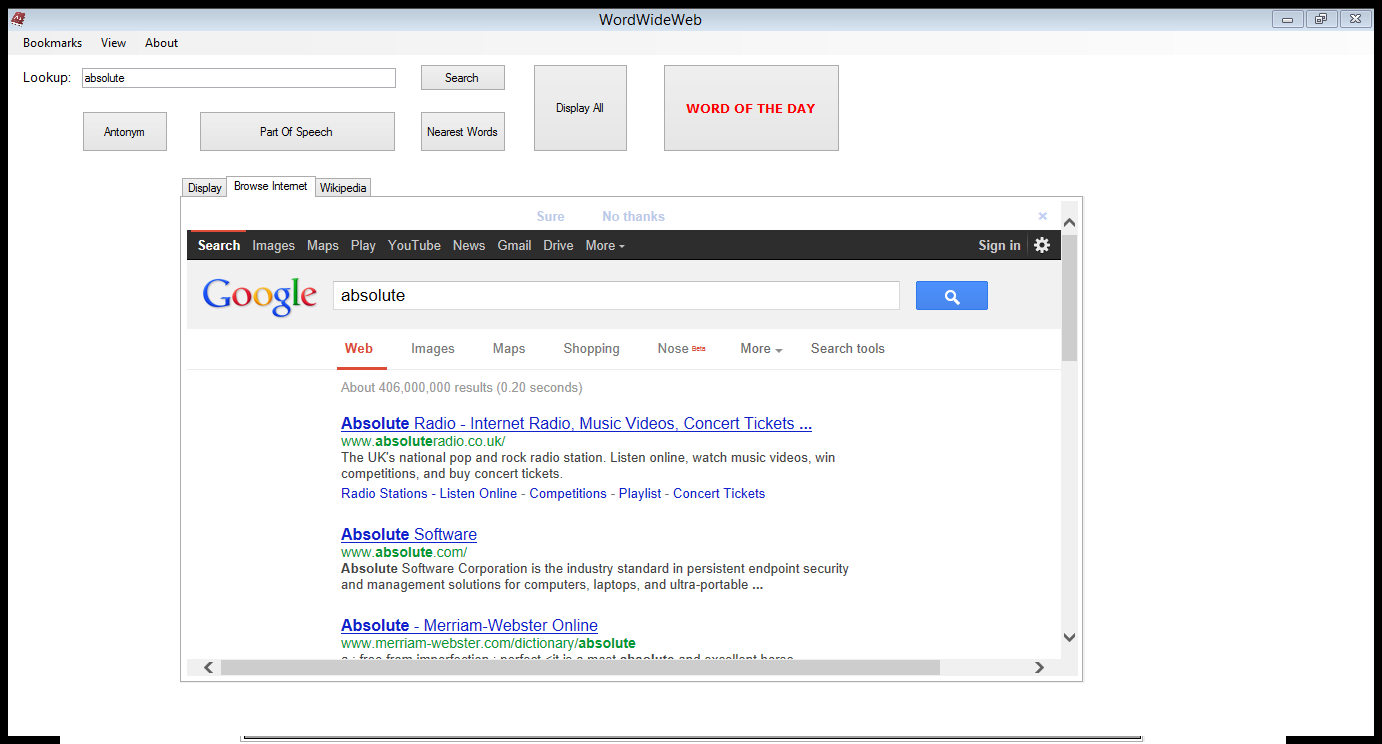
}

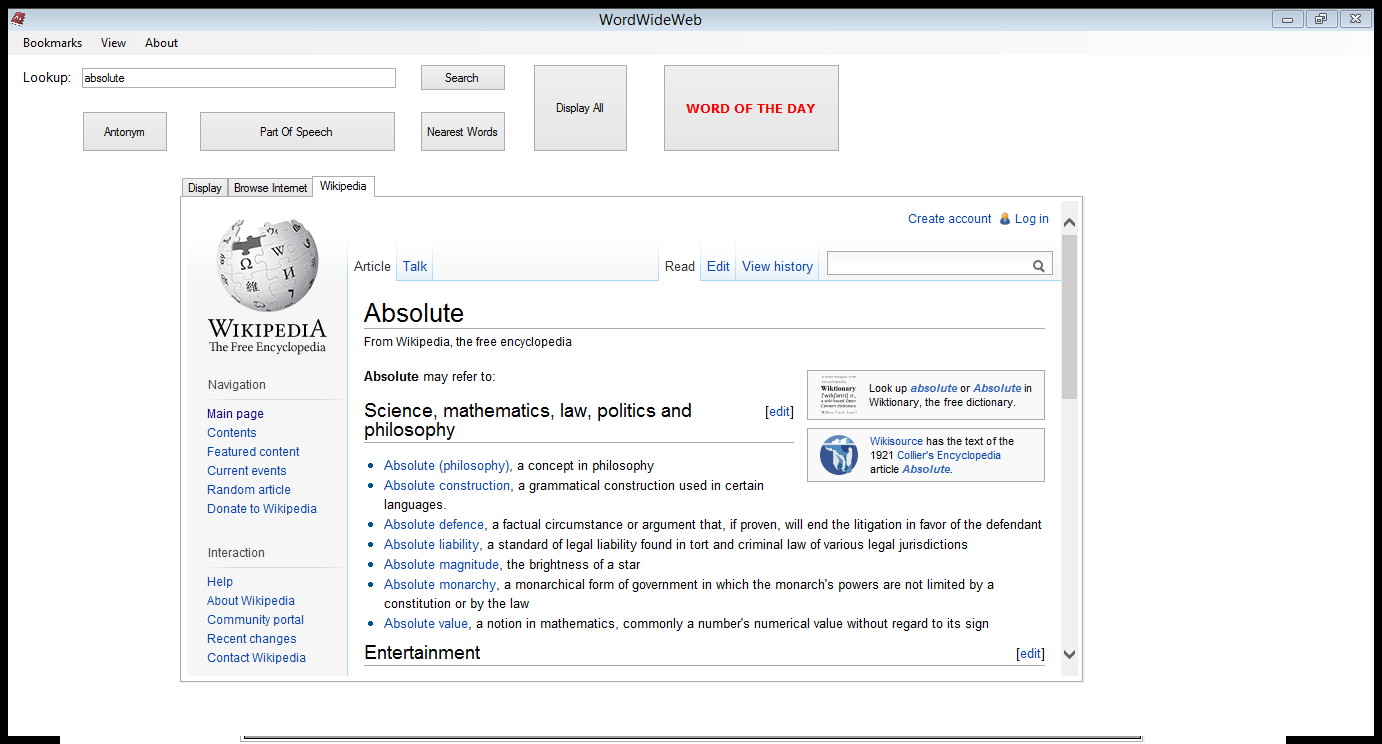
}

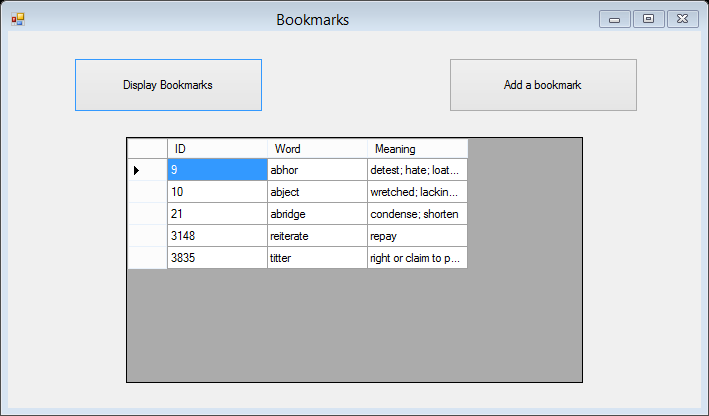
**APPLICATION SCREENSHOTS**

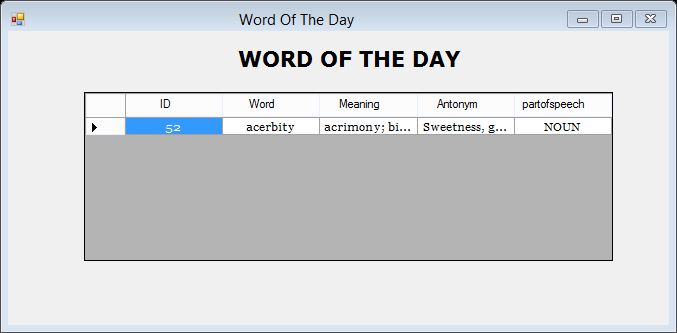














**FUTURE SCOPE**

There are several important features we would like to add to this application. We have created these lists to help us track these improvements, as well as share with the general public our goals for this project.

**User Interface Improvements:**

1. Provide much better browsing functionality to search words.
2. Introducing the feature to edit the dictionary by lexicographer.
3. Data improvements: More and more words as many possible should be included.
4. Create an improved Help system with popup instructions.
5. Create a search history.

**CONCLUSION**

We have been successful in implementing the DICTIONARY and have documented the required substantial differences. All the primary objectives –   
1) Create helpful DICTIONARY for users.  
2) Easy to use interface and all information in one place.

have been fulfilled with appropriate requirements.

Hence the ultimate goal has been fulfilled by this joint venture of team members.

**REFERENCES**

**Books:**

Rob Miles - C# yellow Book

<http://www.amazon.com/dp/1590594398/?tag=stackoverfl08-20>  
<http://shop.oreilly.com/product/9780596003388.do>

**Websites:**

<http://msdn.microsoft.com/en-US/>  
<http://msdn.microsoft.com/en-us/library/aa511258.aspx>  
[www.stackoverflow.com](http://www.stackoverflow.com/)