**CHAPTER 1**

***INTRODUCTION***

**LANGUAGE USED**

The language used in the making of PADC Dictionaryis PYTHON.

**DICTIONARY:**

Dictionary is designed to help people in order to identify the meaning of a particular word. It is a reference book on a particular subject or in general, the items of which are typically arranged in alphabetical order. a book or electronic resource that lists the words of a language (typically in alphabetical order) and gives their meaning, or gives the equivalent words in a different language, often also providing information about pronunciation, origin, and usage.

A well built dictionary can benefit the following people:

1. Students
2. Teachers
3. Researchers
4. Lexicographers

The dictionary we’ve built is called PADC Dictionary.

* 1. **PURPOSE**

The purpose of designing PADC Dictionary is that to give the best results for a word searched by reducing the time taken by a person to take the manual dictionary out and search for a particular word.

It also provides a means of finding a word in such a manner that all the relevant information about the word reflected on the same screen.

This dictionary is prepared to provide all categories of words rather than focusing only a particular category.

Moreover the manual dictionary does not have all the words in them so with the Power of Internet all the words are under one roof in PADC Dictionary.

* 1. **SCOPE**

The scope for the PADC Dictionary system is that it requires internet. But we’re looking for such systems which can contain all the possible words under one roof so that the requirement of internet can be negotiated.

* 1. **OBJECTIVE**

The objective of PADC Dictionary System is that to reduce the usage of manual search of words in dictionary and to provide all the possible words in one application. Though the words can be searched on google.com as well but to avoid ambiguity the best site is tracked for best word meaning. The moment any lexicographer put any new word with its meaning on the site our dictionary is able to cope with those new words as well.

This system not only provides the word meanings but also every other information which might be required with the usage of words like the sentence built up, pronunciation in phoenix and pronunciation in voice as well.

There are certain users who are focusing on improving their vocabulary by introducing one word everyday. So for them this system contains a feature where they can find Word Of The Day.

**1.4 SDLC METHODOLOGIES**

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality softwares. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

* SDLC is the acronym of Software Development Life Cycle.
* It is also called as Software Development Process.
* SDLC is a framework defining tasks performed at each step in the software development process.
* ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

* The following figure is a graphical representation of the various stages of a typical SDLC.

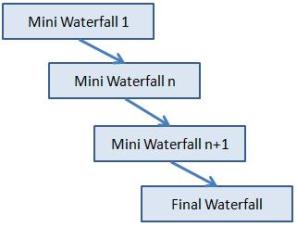


Types of Software developing life cycles (SDLC) Methodologies

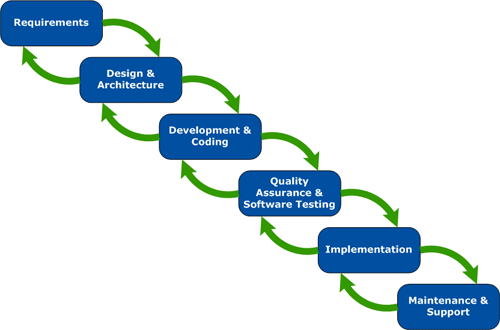
* [Waterfall Model](http://melsatar.blog/2018/02/16/the-waterfall-model-a-different-perspective/)
* [Evolutionary Prototyping](http://en.wikipedia.org/wiki/Software_prototyping) Model
* [Spiral](http://en.wikipedia.org/wiki/Spiral_model) Method ([SDM](http://en.wikipedia.org/wiki/Software_development_methodology))
* [Iterative and Incremental](http://en.wikipedia.org/wiki/Iterative_and_incremental_development) Method

The development of PADC Dictionary has been done with [Iterative and Incremental](http://en.wikipedia.org/wiki/Iterative_and_incremental_development) Method.

## *Iterative and Incremental Model*

It is developed to overcome the weaknesses of the waterfall model. It starts with an initial planning and ends with deployment with the cyclic interactions in between. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during the development of earlier parts or versions of the system. It can consist of mini waterfalls or mini V-Shaped model

**The usage:** It is used in shrink-wrap application and large system which built-in small phases or segments. Also, can be used in a system has separated components, for example, ERP system. Which we can start with the budget module as a first iteration and then we can start with the inventory module and so forth.



Fulfilling all the requirements in one go was very tedious task to do. At the very initial stages the Basic dictionary was created as a console based application. In which the user puts the word and gets the meaning. In further enhancements features like pronunciation, sentence, examples were added.

**CHAPTER 2**

***SYSTEM ANALYSIS STUDY***

**2.1 INTRODUCTION**

System analysis is the act, process, or profession of studying an activity (such as a procedure, a business, or a physiological function) typically by mathematical means in order to define its goals or purposes and to discover operations and procedures for accomplishing them most efficiently.

**2.1.1 Project Overview**

System analysis is one of the most critical error prone and most communication intensive aspect of software development. It is an activity that helps to understand the problem to be solved after the problem statement and next step is to gather all possible requirements according to the user.

For our project we had some basic ideas what the user expects out of a dictionary. At the initial stages we built such a system that could search all the words that a user inputs. The rest requirements were investigated and then be acted upon.

**2.1.2 Existing System**

The existing system was the user was expected to take out the manual dictionaries and then search for a word which was a bit tedious. Even the possibility of all the words under one roof was also an issue. There were many sites which could provide the dictionary meaning of the particular word but in this case the ambiguity was the major issue because the users used to get a bit confused that which meaning is appropriate for the given word.

To solve both these issues we researched a lot and we came up with the site called dictionary.com. So the data for dictionary was fetched from the same site. The site was showing too much detail about the world so we created a very simple user interface in which the user can see all the relevant information. Besides that some other features were also included in the dictionary which will be further discussed.

**2.2 SYSTEM REQUIREMENTS GATHERING**

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and **detailed**. In software engineering, such requirements are often called functional specifications.

**2.2.1 Requirement Gathering Techniques**

1. [Brainstorming](https://www.brighthubpm.com/risk-management/48947-effective-brainstorming-methods/)
2. Interview
3. Observation
4. Prototyping
5. Requirements Workshop
6. Reverse Engineering
7. Survey
8. Group Interviews

In PADC Dictionary Requirement Analysis is done by Interviews with students and lexicographers.

Interviewing is the technique in which a potential user of the software is interviwed about his possible requirements.

**2.2.2 Requirement Analysis**

By interviewing a certain group of students we came to a conclusion that a user wants the following features in a dictionary:

* Word Meaning
* Sentence
* Pronounciation by pheonix
* Pronounciation by voice
* Simple GUI

Any specific requirement by lexicographers was encountered.

**2.3 FEASIBILITY STUDY**

A project feasibility study is a comprehensive report that examines in detail the five frames of analysis of a given project. It also takes into consideration its four Ps, its risks and POVs, and its constraints (calendar, costs, and norms of quality). The goal is to determine whether the project should go ahead, be redesigned, or else abandoned altogether.

**2.3.1 Technical Feasibility**

A study of resource availability that may affect ability to achieve an acceptable system. This evaluation determines whether technology needed for the proposed system is available or not. The following points are taken into consideration:

* Can the work for the project be done with current equipment existing technology and available personnel?
* Can the system be upgraded if need be?
* If new technology is needed then what can be developed?

**Technical Requirements**

* Microsoft Windows 10/8/7/Vista/2003/XP (incl.64-bit)
* 1 GB RAM minimum
* 2 GB RAM recommended
* 1024x768 minimum screen resolution
* Python 3 or higher
* Internet Connection

**2.3.2 Economic Feasibility**

The purpose of an [**economic feasibility study (EFS)**](https://www.ofm.wa.gov/sites/default/files/public/legacy/policy/glossary.asp#economicFeasibilityStudy) is to demonstrate the net benefit of a proposed project for accepting or disbursing electronic funds/benefits, taking into consideration the benefits and costs to the agency, other state agencies, and the general public as a whole.

The EFS is composed of **two required forms**:

* Business Case
* Cost Benefit Analysis

This dictionary is economically feasible as this application is open source.

**2.3.3 Operational Feasibility**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

Operationally feasible this dictionary solves many issues like:

* No manual dictionaries required
* Solve the issues of ambiguity as only one site is scrapped.
* It has everything that an E-Dictionary should have. This includes:
  + 1. Meaning
    2. Sentence
    3. Pronunciation by phoenix
    4. Pronunciation by voice

**CHAPTER 3**

***SYSTEM DESIGN***

**3.1 INTRODUCTION**

The purpose of the design phase is to plan a solution of the problem specified by the requirement Documents. This phase is the first step in moving from the problem domain to the solution domain. The output of this phase is the design document, which will act as a blue print for the solution and is used later during implementation, testing and maintenance. The design activity is divided into three separate phases: Conceptual Data Modeling, Logical Data Base Design, and Physical Data Base Design.

Conceptual Data Modeling analyzes the overall data requirement of the proposed information system.

 Logical Data Base Design transform the conceptual data model into a standard relation called relation based on relational database theory and a process called Normalization.

In Physical Data Base Design and Definition, one decides on the organization of the database in computer storage (usually disk) and defines the physical structure of data base management system.

**3.2 PHYSICAL DESIGN:**

**3.2.1 Activity Diagram**

An activity diagram visually presents a series of actions or flow of control in a system similar to a [flowchart](https://www.smartdraw.com/flowchart/) or a [data flow diagram](https://www.smartdraw.com/data-flow-diagram/). Activity diagrams are often used in business process modeling. They can also describe the steps in a [use case diagram](https://www.smartdraw.com/use-case-diagram/). Activities modeled can be sequential and concurrent. In both cases an activity diagram will have a beginning (an initial state) and an end (a final state). In between there are ways to depict activities, flows, decisions, guards, merge and and time events and more.

**Basic Activity Diagram Notations and Symbols**

**Initial State or Start Point**

A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram. For activity diagram using swimlanes, make sure the start point is placed in the top left corner of the first column.

Start point symbol - Activity diagram

**Activity or Action State**

An action state represents the non-interruptible action of objects. You can draw an action state in SmartDraw using a rectangle with rounded corners.



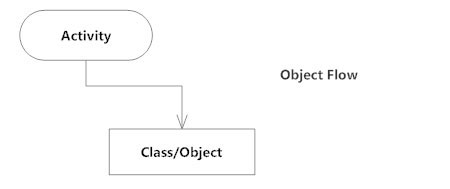
**Action Flow**

Action flows, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line.

Action flow - Activity diagram

**Object Flow**

Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. An object flow arrow from an object to an action indicates that the action state uses the object.



**Decisions and Branching**

A diamond represents a decision with alternate paths. When an activity requires a decision prior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."

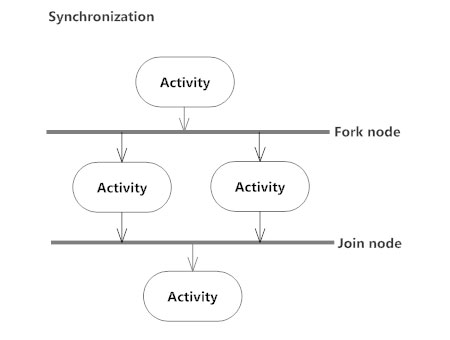


**Synchronization**

A fork node is used to split a single incoming flow into multiple concurrent flows. It is represented as a straight, slightly thicker line in an activity diagram.

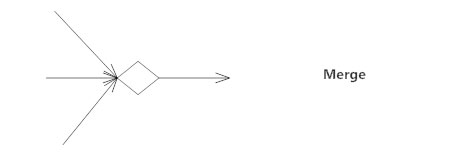
A join node joins multiple concurrent flows back into a single outgoing flow.

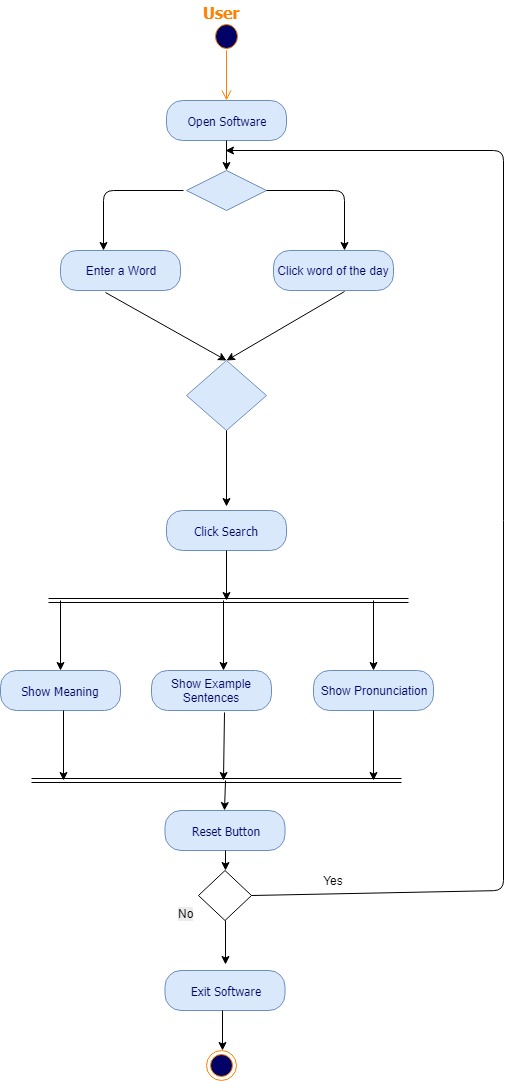
A fork and join mode used together are often referred to as synchronization.



**Merge Event**

A merge event brings together multiple flows that are not concurrent.





**3.2.2 Use Case Diagram**

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It consists of a group of elements (for example, classes and interfaces) that can be used together in a way that will have an effect larger than the sum of the separate elements combined. The use case should contain all system activities that have significance to the users. A use case can be thought of as a collection of possible scenarios related to a particular goal, indeed, the use case and goal are sometimes considered to be synonymous.

**Basic Use Case Diagram Symbols and Notations**

**System**   
Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.

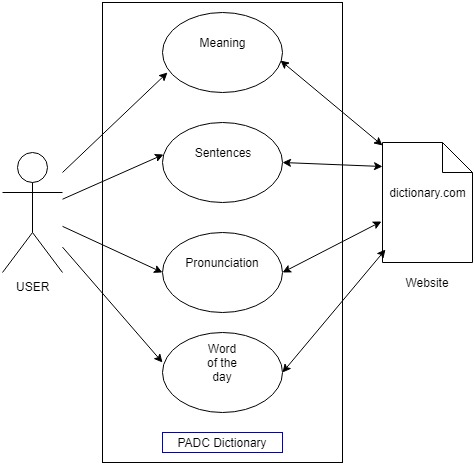


**Use Case**   
Draw use cases using ovals. Label the ovals with verbs that represent the system's functions.



**Actors**   
Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.





**3.2.3Data Flow Diagram**

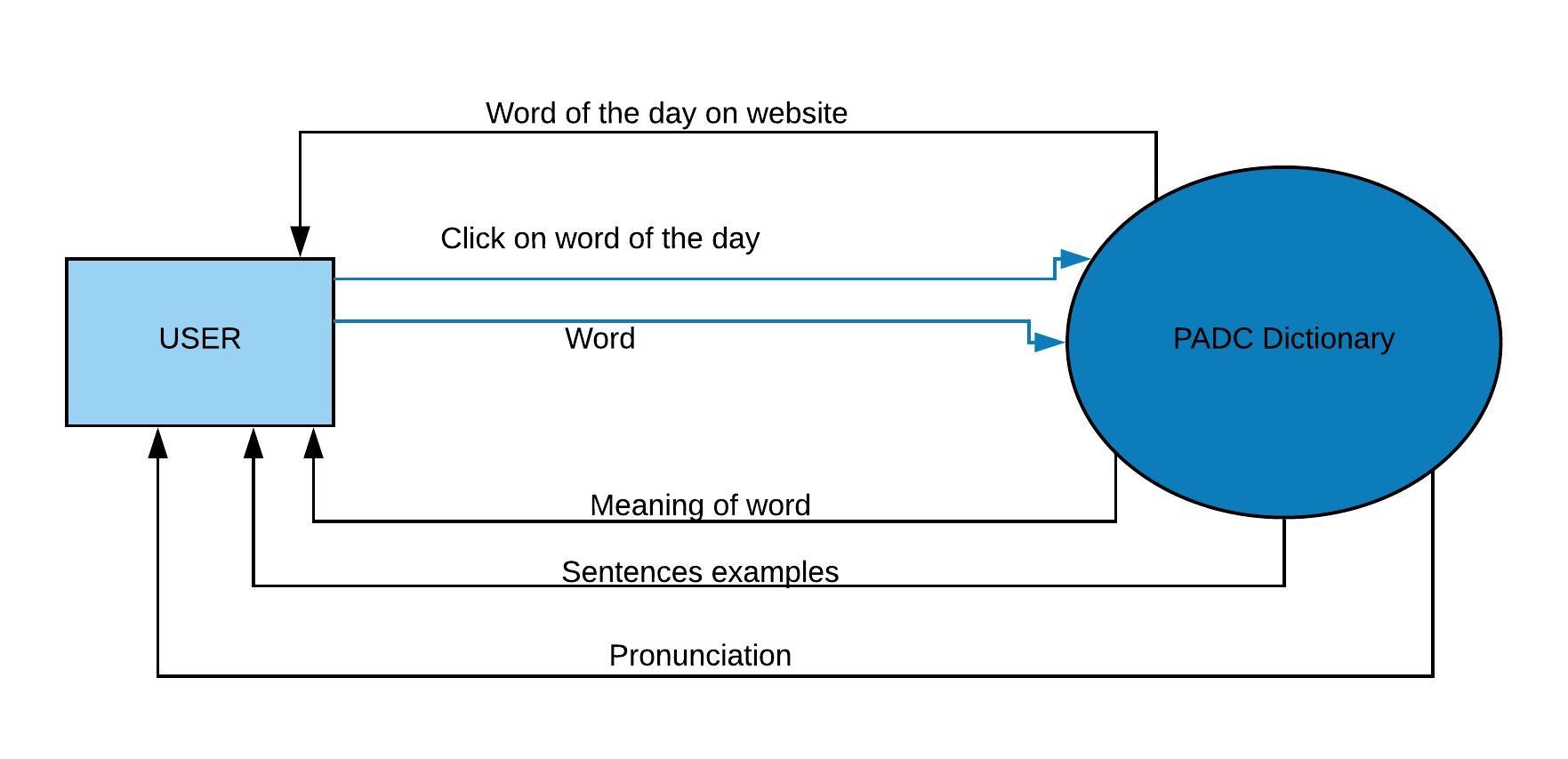
A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

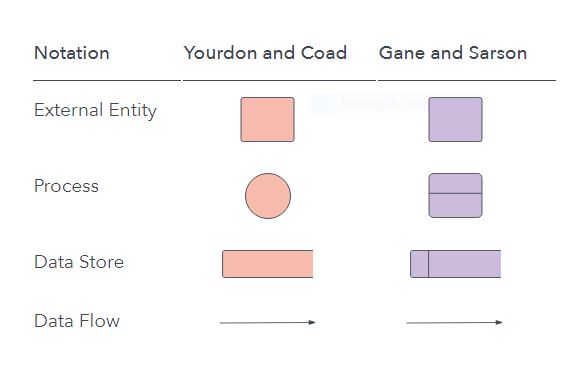
**TWO COMMON SYSTEMS OF SYMBOLS ARE NAMED AFTER THEIR CREATORS:**

* Yourdon and Coad
* Yourdon and DeMarco
* Gane and Sarson

One main difference in their symbols is that Yourdon-Coad and Yourdon-DeMarco use circles for processes, while Gane and Sarson use rectangles with rounded corners, sometimes called lozenges. There are other symbol variations in use as well, so the important thing to keep in mind is to be clear and consistent in the shapes and notations you use to communicate and collaborate with others.

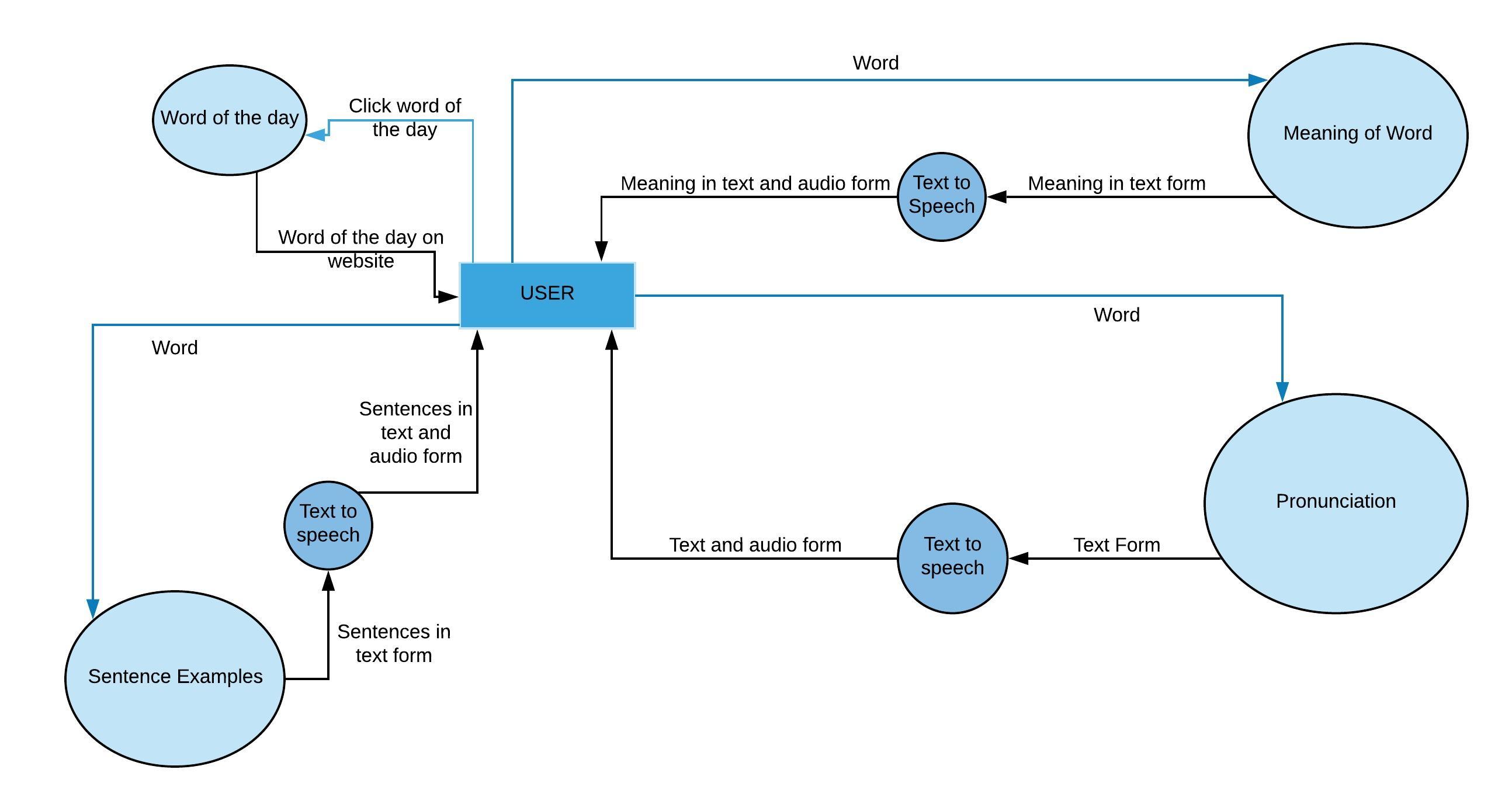
Using any convention’s DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

1. **EXTERNAL ENTITY:** an outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.
2. **PROCESS:**any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules. A short label is used to describe the process, such as “Submit payment.”
3. **DATA STORE:** files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label, such as “Orders.”
4. **DATA FLOW:** the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name, like “Billing details.”



Level-0

Level-1

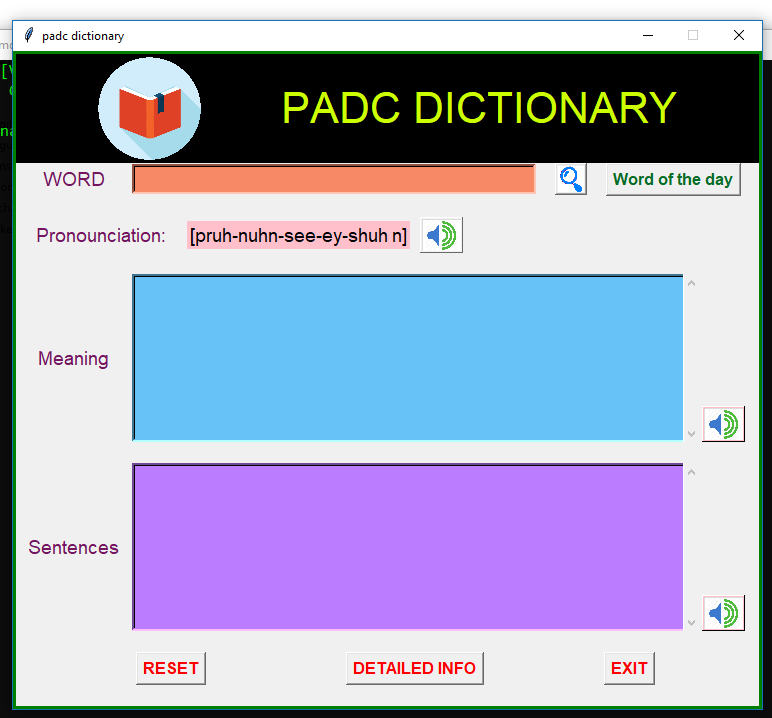


**3.2.4 Site Maps /APP Maps/Network Flow Charts etc.**

A site map is a visual or textually organized model of a Applications content that allows the users to navigate through the site to find the information they are looking for, just as a traditional geographical map helps people find places they are looking for in the real world. A site map is a kind of interactive table of contents, in which each listed item links directly to its counterpart sections of the Application. Site maps perform the same service that the layout maps in large shopping malls perform: without them, it is possible to explore a complex site by trial and error, but if you want to be sure to find what you're looking for, the most efficient way to do that is to consult a model of the resources available. If an application is small and uncomplicated, a site map may be unnecessary, just as a layout map may not be required for shoppers to find their way through small shopping malls.

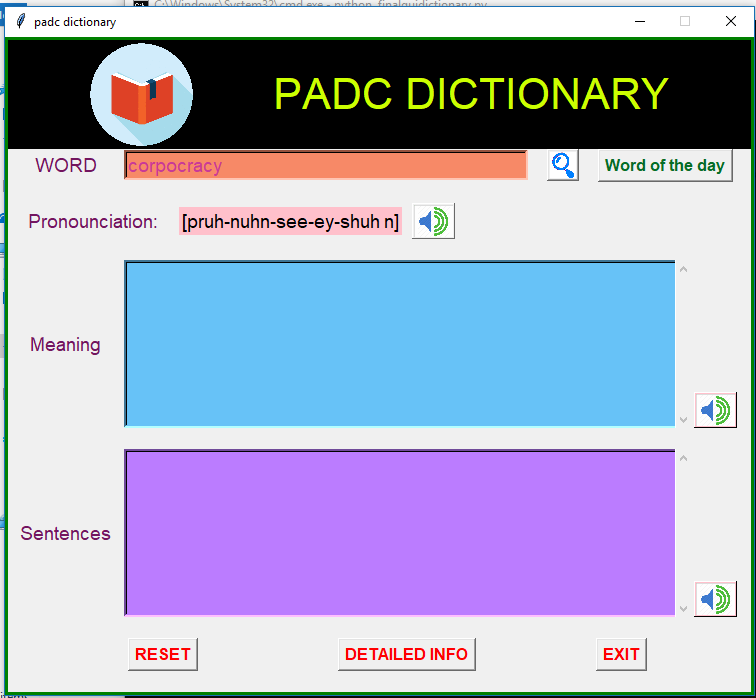
**3.3 USER INTERFACE DESIGN**

In this Application, the user sees only one interface. The Main Page loads when user double clicks on the software. User is available with various options their:-

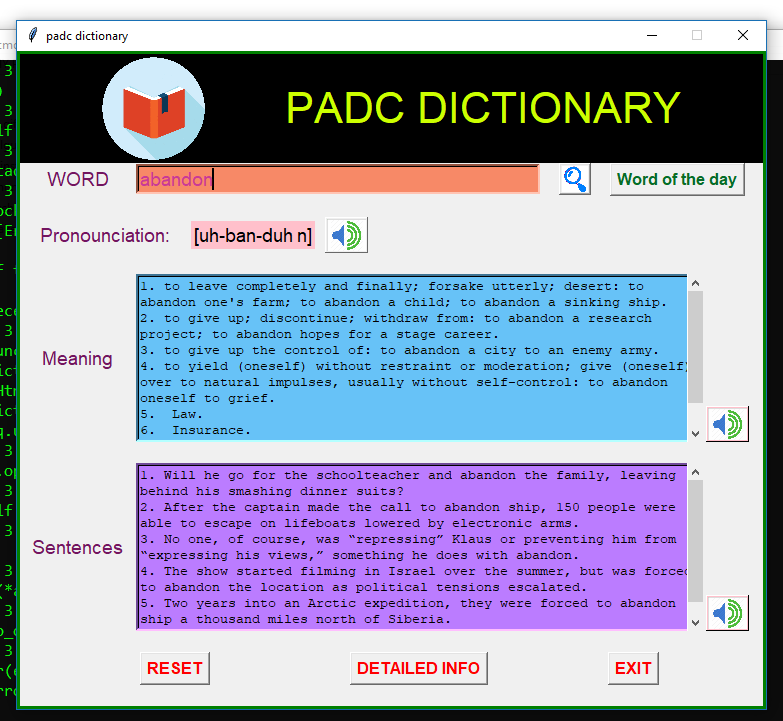


**1.Word of the day:**

User can click on the button which is located at the upper left corner of our page. It will fetch the word of the day from the website and display that word in the text field of “Word”. After that user can click on the “Search” button which will provide him/her with the meaning, sentence examples, pronunciation of that word.

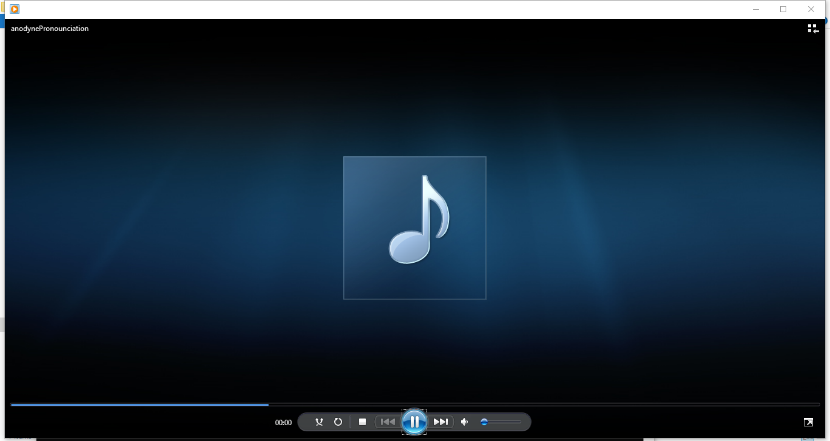


**2.Enter word:** Instead of clicking on “Word of the day” user can enter his/her own word to search for the meaning in the text field of “word”. ”. After that user can click on the “Search” button which will provide him/her with the meaning, sentence examples, pronunciation of that word in various other text fields.



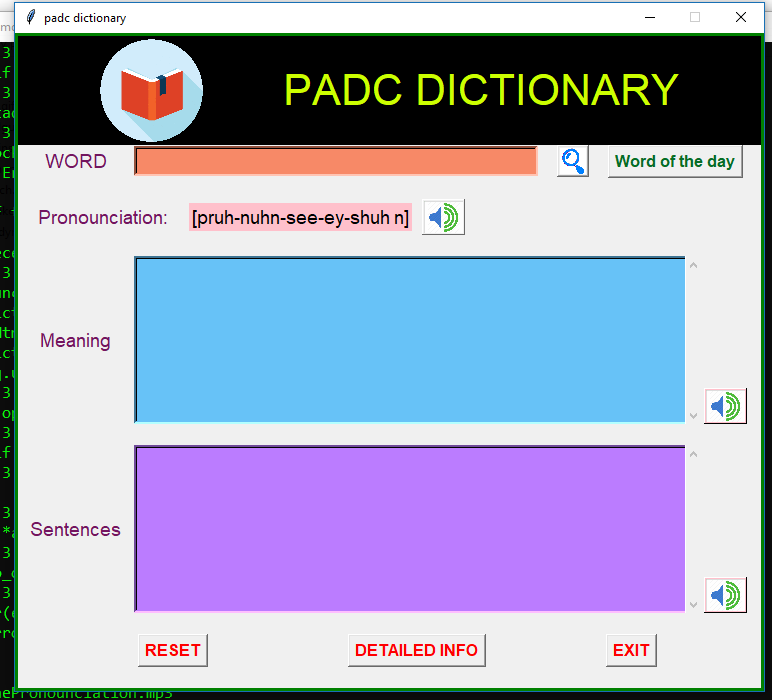
**3.Text to Audio:**

When user search for any word, he/she will be available with meaning, sentence examples, pronunciation in the form of text in text fields. Furthermore user can click on the audio icon on the right side of every text field to convert the text to audio form. It will open up the media player whichever is available in the users’ computer. In our case it is “windows media player” and that audio file is saved offline in the system.



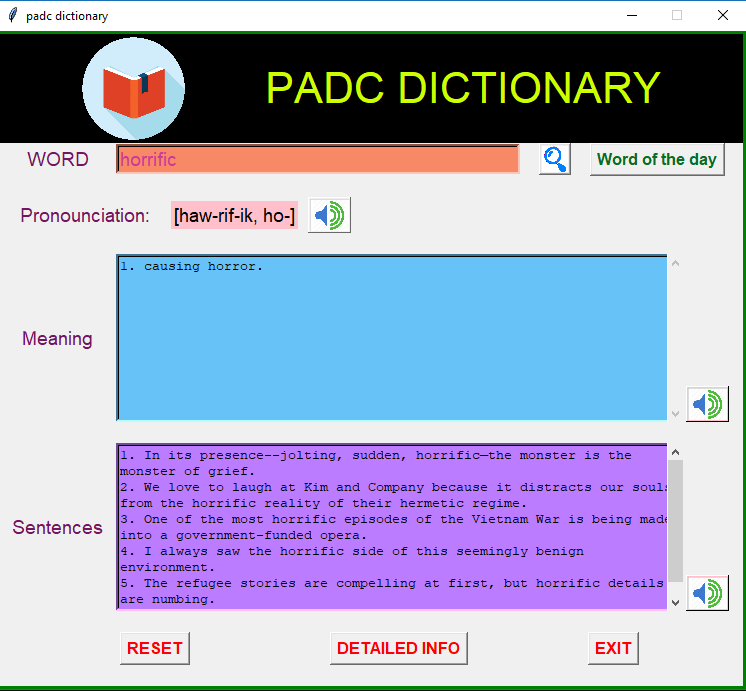
**4.Reset Button:**

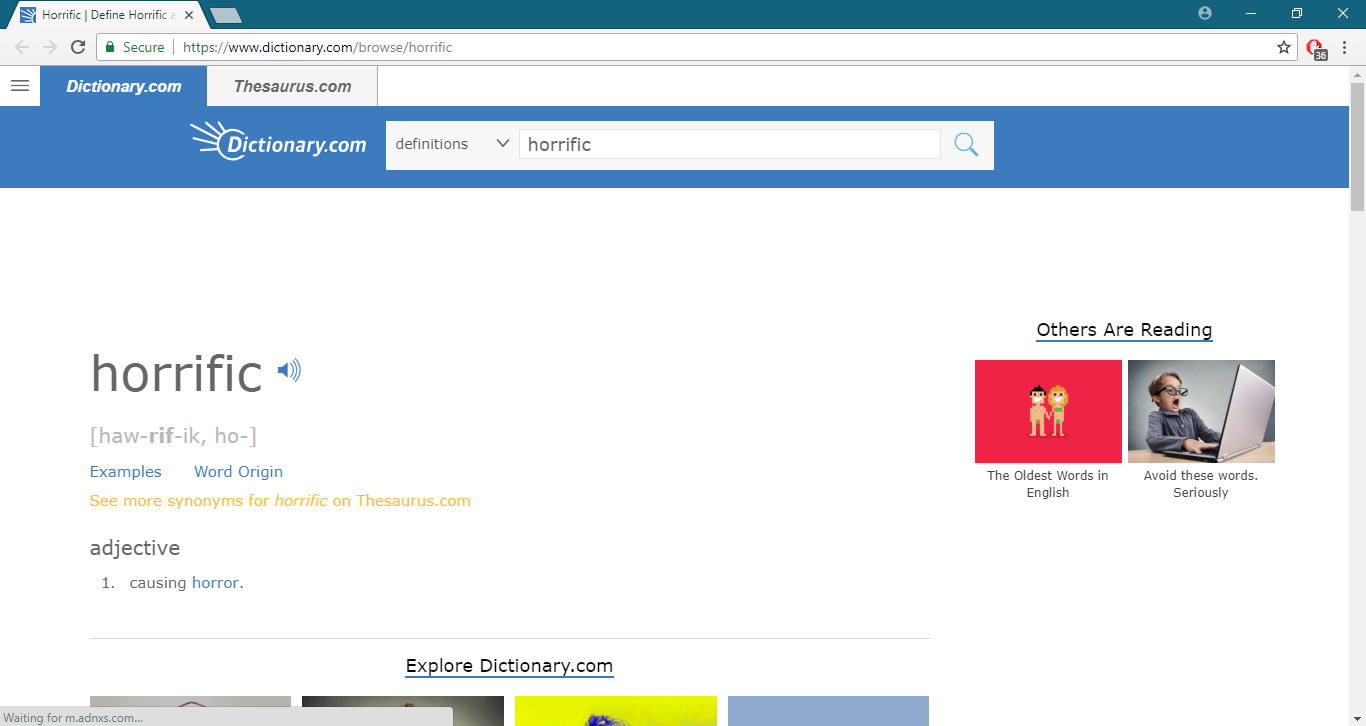
If the user wants to clear each and every text field, he/she can click on the “Reset” button which will clear every text field on the page.



**5.Detailed Info:**

If the user is not satisfied with something provided in our software he/she can click on “Detailed Info” button which will open up the browser with the website tab providing detailed information about the word asked by the user.





**CHAPTER 4**

***SOFTWARE CODING, TESTING & IMPLEMENTATION***

* 1. **SOFTWARE SOURCE CODE**

from tkinter import \*

import urllib.request as req

from bs4 import BeautifulSoup

import pyperclip    #copy to clipboard

from gtts import gTTS   #text to speech

import os   # to run speech file

import webbrowser   #to open web browser

ab =Tk()

ab.title('padc dictionary')

ab.configure(*highlightbackground*="green", *highlightcolor*="green", *highlightthickness*=3)

*def* reset():

entry.delete('0','end')

t1.delete('1.0','end')

t2.delete('1.0','end')

prolabel.set('[pruh-nuhn-see-ey-shuh n]')

*def* getHtmlData(*URL*):

    response = req.urlopen(URL)

    data = response.read()

    soup = BeautifulSoup(data, "html.parser")

    return soup

*def* GetWordOfTheDay():

    url="http://www.dictionary.com/wordoftheday/"

    HtmlData = getHtmlData(url)

    word\_section = HtmlData.find\_all("div", {"class" : "definition-header"})

    wordOfTheDay = word\_section[0].strong.string

    wordenter.set(wordOfTheDay)

*def* getWordMeanings(*HtmlData*):

    # url="http://www.dictionary.com/browse/"+word

    # HtmlData = getHtmlData(url)

    allListOfDetails = HtmlData.find\_all("ol")

    listOfDetails = allListOfDetails[0].find\_all('li')

    index = 1

    result=""

    # print("Meanings of the word " + word +" is :")

    print()

    # result +="Meanings of the word " + word +" is :"+"\r\n"

    for ele in listOfDetails:

        result+= *str*(index) + ". " + ele.span.text +"\r\n"

        # print(str(index) + ". " + ele.span.text)

        index = index + 1

    return result

*def* getWordExamples(*HtmlData*):

    # url="http://www.dictionary.com/browse/"+word

    # HtmlData = getHtmlData(url)

    allListOfDetails = HtmlData.find\_all("ul", {'class', 'css-o9b79t e15kc6du4'})

    if allListOfDetails==[]:

        result="NO EXAMPLES FOUND"

    #print(allListOfDetails)

    else:

        listOfDetails = allListOfDetails[0].find\_all('h4')

        # print(listOfDetails)

        index = 1

        result=""

        for ele in listOfDetails:

            result+= *str*(index) + ". " + ele.text +"\r\n"

            # print(str(index) + ". " + ele.text)

            index = index + 1

    return result

*def* textToSpeech(*convertText*,*word*,*use*):

    #print("Generating text-to-speech for " + word + "\n ")

    if (convertText!=""):

        language = 'en'

        myobj = gTTS(*text* = convertText , *lang* = language)

        Filename = word +use

        print("Filename is "+ Filename +".mp3")# """instead of printing make a pop up of this"""

        myobj.save(Filename + ".mp3")

        os.system(Filename + ".mp3")

    else:

        pass

*def* getWordPronounce(*HtmlData*):

    # url="http://www.dictionary.com/browse/"+word

    # HtmlData = getHtmlData(url)

    allListOfDetails = HtmlData.find\_all("span", {'class', 'css-1khtv86 e1rg2mtf2'})

    pronounce=allListOfDetails[0].text

    return pronounce

*def* detailedinfo():

    word=wordenter.get()

    if (word==""):

        url ="http://www.dictionary.com/"

    else:

        url="http://www.dictionary.com/browse/"+word

    pyperclip.copy(url)

    webbrowser.open(url,*new*=2)

*def* OnSearchButtonClick():

word1=wordenter.get()

url = "http://www.dictionary.com/browse/"+word1

HtmlData =getHtmlData(url)

    # print(HtmlData)

wordPronounce = getWordPronounce(HtmlData)

# prolabel.set("")

prolabel.set(wordPronounce)

#print(wordPronounce)

    #print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

t1.delete('1.0','end')

t2.delete('1.0','end')

wordMeaning=getWordMeanings(HtmlData)

t1.insert('1.0',wordMeaning)

    #print(wordMeaning) #'''change msgBox in front of meaning'''

WordExamples=getWordExamples(HtmlData)

t2.insert('1.0',WordExamples)

    #print(WordExamples) #'''change msgBox in front of example'''

    #print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

*def* onSpeech1ButonClick():

    textToSpeech(wordenter.get(), entry.get(), 'Pronounciation')

*def* onSpeech2ButonClick():

    textToSpeech(t1.get('1.0','end'),entry.get(),'MEANING')

*def* onSpeech3ButonClick():

    textToSpeech(t2.get('1.0','end'),entry.get(),'SENTENCES')

guiframe=Frame(ab)

guiframe.grid(*row*=0,*column*=0,*sticky*=W)

logo=PhotoImage(*file*='dict.gif')

label=Label(guiframe,*text*='PADC DICTIONARY',*font*=("jokerman",32),*padx*=80,*background*='black',*foreground*='#ccff00')

sub\_logo=logo.subsample(*x*=5,*y*=5)

label.img=sub\_logo

label.config(*image*=label.img)

label.config(*compound*='left')

label.grid(*row*=0,*column*=0,*sticky*='w')

guiframe2=Frame(ab)

guiframe2.grid(*row*=1,*column*=0,*sticky*='w')

wordenter=StringVar()

label2=Label(guiframe2,*text*='WORD',*font*=("arial",14),*foreground*='#73155F',*width*=10).grid(*row*=1,*column*=0)

entry=Entry(guiframe2,*font*=("arial",14),*width*=36,*textvariable*=wordenter,*bd*=3,*foreground*='#C93A97',*bg*='#f78967')

entry.grid(*row*=1,*column*=1)

logo1=PhotoImage(*file*='search.gif')

search\_logo=logo1.subsample(*x*=10,*y*=10)

searchb=Button(guiframe2,*image*=search\_logo,*bd*=2,*command*=OnSearchButtonClick).grid(*row*=1,*column*=5,*padx*=19)

wod=Button(guiframe2,*text*='Word of the day',*bd*=2,*font*=('arial',12,'bold'),*fg*='#076d27',*command*=GetWordOfTheDay).grid(*row*=1,*column*=6)

empr1=Label(ab).grid(*row*=2)

prolabel=StringVar()

prolabel.set('[pruh-nuhn-see-ey-shuh n]')

guiframep=Frame(ab)

guiframep.grid(*row*=3,*column*=0,*sticky*='w')

labelp=Label(guiframep,*text*='Pronounciation:',*font*=("arial",14),*foreground*='#73155F',*width*=15).grid(*row*=3,*column*=0)

labelp1=Label(guiframep,*textvariable*=prolabel,*font*=("arial",14),*foreground*='black',*bg*='pink').grid(*row*=3,*column*=1)

speaker=PhotoImage(*file*='speaker.gif')

smallsp=speaker.subsample(*x*=16,*y*=16)

s=Button(guiframep,*image*=smallsp,*bd*=1,*command*=onSpeech1ButonClick).grid(*row*=3,*column*=2,*padx*=10)

empr2=Label(ab).grid(*row*=4)

guiframe3=Frame(ab)

guiframe3.grid(*row*=5,*column*=0,*sticky*='w')

t1=Text(guiframe3,*width*=70,*height*=10,*wrap*='word',*bd*=3,*bg*='#67c2f7')

s1=Scrollbar(guiframe3,*orient*='vertical',*bg*='#137e84')

t1.configure(*yscrollcommand*=s1.set)

s1.configure(*command*=t1.yview)

t1.grid(*row*=5,*column*=1,*sticky*=N+S)

s1.grid(*row*=5,*column*=1,*sticky*=N+S+E)

label3=Label(guiframe3,*text*='Meaning',*width*=10,*font*=("arial",14),*foreground*='#73155F').grid(*row*=5,*column*=0)

searchb=Button(guiframe3,*image*=smallsp,*bd*=1,*bg*='pink',*command*=onSpeech2ButonClick).grid(*row*=5,*column*=2,*sticky*='s',*padx*=2)

empr3=Label(ab).grid(*row*=6)

guiframe4=Frame(ab)

guiframe4.grid(*row*=7,*column*=0,*sticky*='w')

t2=Text(guiframe4,*width*=70,*height*=10,*wrap*=WORD,*bd*=3,*bg*='#bb7cff')

s2=Scrollbar(guiframe4,*orient*='vertical',*bg*='#137e84')

t2.configure(*yscrollcommand*=s2.set)

s2.configure(*command*=t2.yview)

t2.grid(*row*=7,*column*=1,*sticky*=N+S)

s2.grid(*row*=7,*column*=1,*sticky*=N+S+E)

label3=Label(guiframe4,*text*='Sentences',*width*=10,*font*=("arial",14),*foreground*='#73155F').grid(*row*=7,*column*=0)

searchb1=Button(guiframe4,*image*=smallsp,*bd*=1,*bg*='pink',*command*=onSpeech3ButonClick).grid(*row*=7,*column*=2,*sticky*='s',*padx*=2)

empr4=Label(ab).grid(*row*=8)

guiframe5=Frame(ab)

guiframe5.grid(*row*=9,*column*=0,*sticky*='w')

resetb=Button(guiframe5,*text*='RESET',*font*=("arial",12,'bold'),*fg*='red',*command*=reset)

resetb.grid(*row*=9,*column*=0,*padx*=120)

detail=Button(guiframe5,*text*='DETAILED INFO',*font*=("arial",12,'bold'),*fg*='red',*command*=detailedinfo)

detail.grid(*row*=9,*column*=1,*padx*=20)

exitb=Button(guiframe5,*text*='EXIT',*font*=("arial",12,'bold'),*fg*='red',*command*=ab.destroy)

exitb.grid(*row*=9,*column*=2,*padx*=100)

empr5=Label(ab).grid(*row*=10)

ab.resizable(False, False)

ab.mainloop()

**4.2. T****ESTING**

**Testing Strategy**

A testing strategy is a general approach to the testing process rather than a method of devising particular system or component tests. Different testing strategies may be adopted depending on the type of system to be tested and the development process used. There are two different strategies available:

1. Top-Down Testing
2. Bottom-Up Testing.

### Types of Testing

#### Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

#### Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

#### Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

#### System Testing

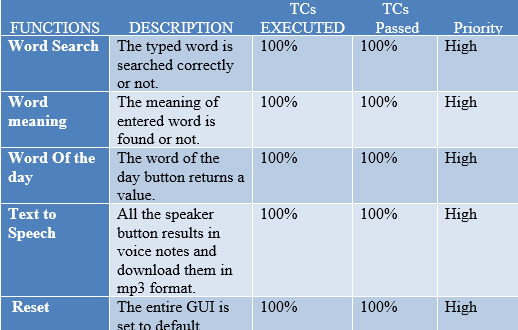
System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

#### Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

#### Performance Testing

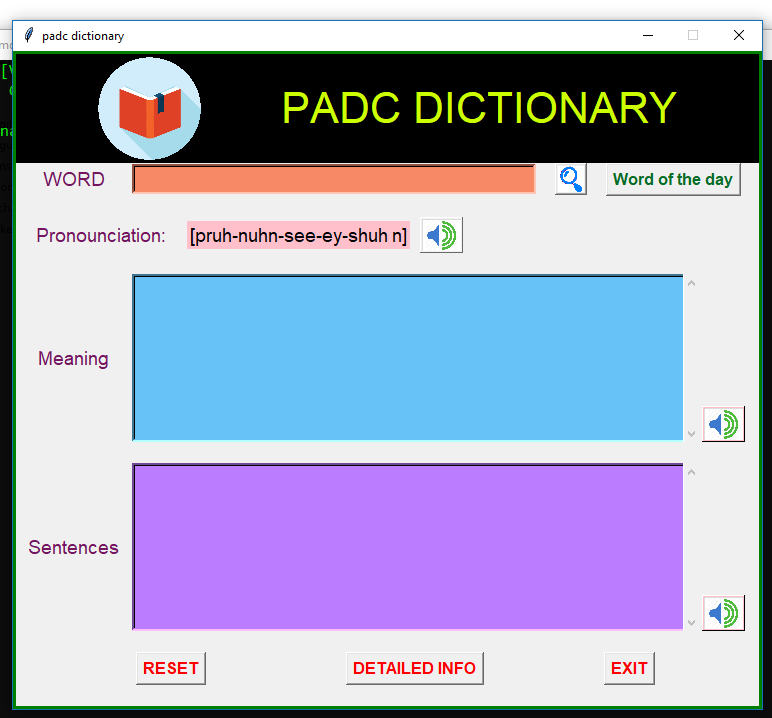
Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

**4.3. TEST CASES**

**CHAPTER 5**

***OUTPUT FORMS & REPORTS***

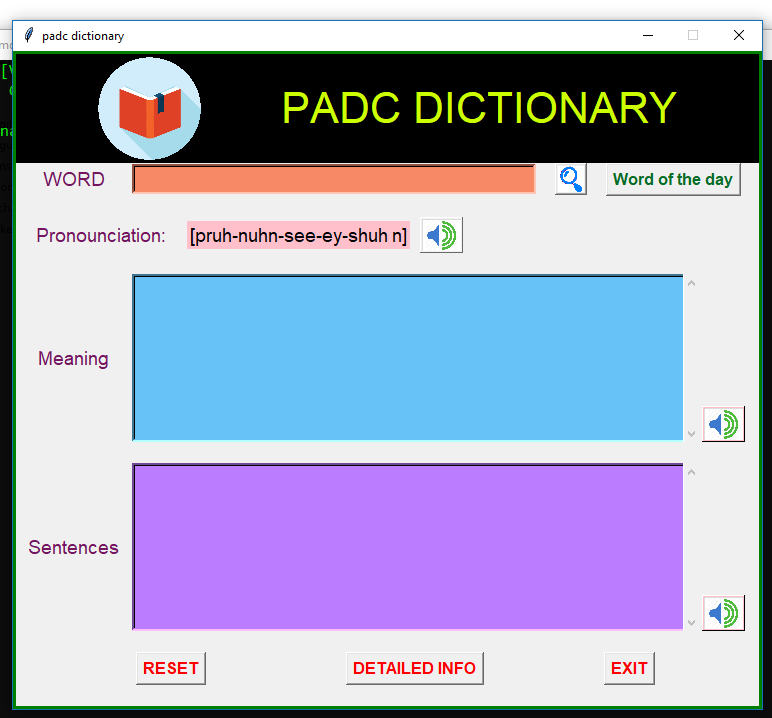
**5.1 INPUT / OUTPUT FORMS SCREEN**

******

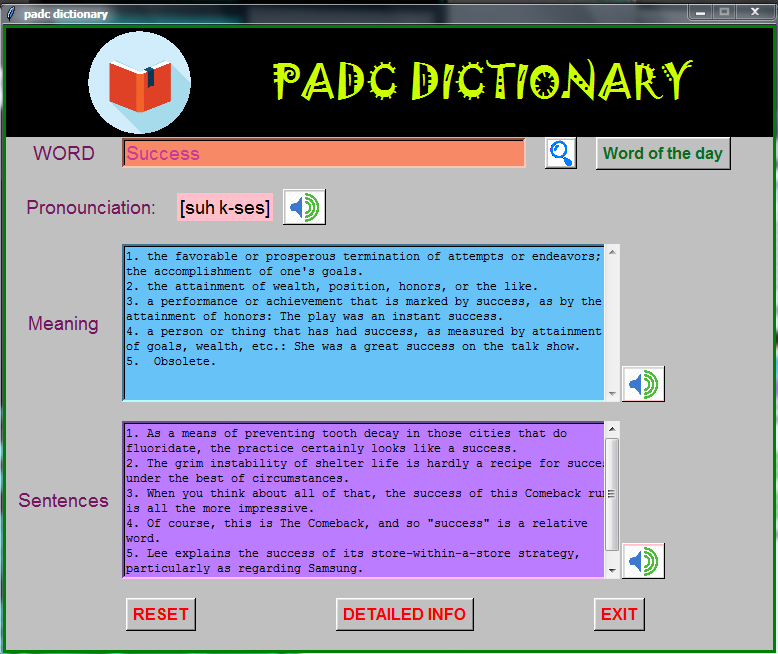
This is the Graphic User Interface of PADC Dictionary. Unique idea behind this UI is that usually all the dictionaries have a very boring interface which psychologically makes the user lethargic.

This GUI is completely functional and have a very user friendly interface.

**5.2 REPORTS WITH ACTUAL DATA**

****

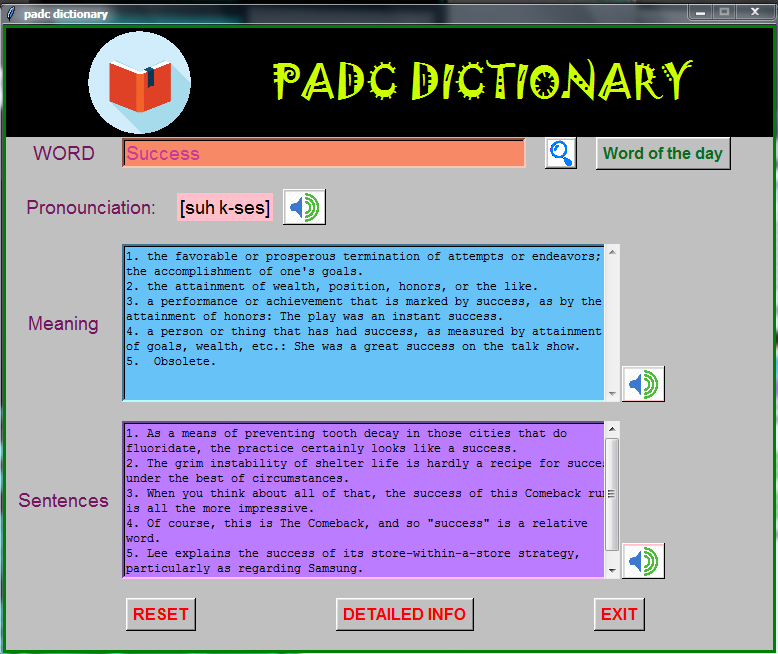
**The user interface is quite simple which has a textbox to get the word to searched from the user and a button to search that word in the dictionary.**

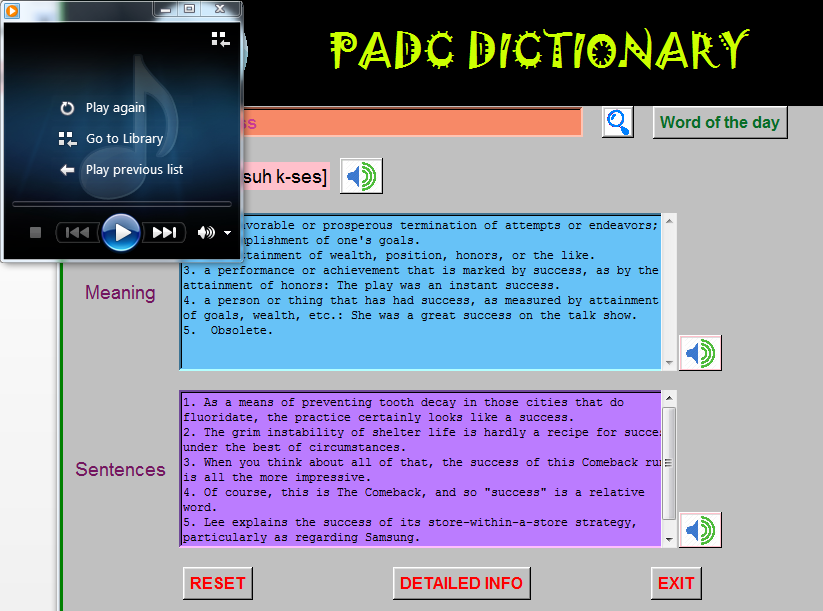


The textbox where the meaning of word is shown.

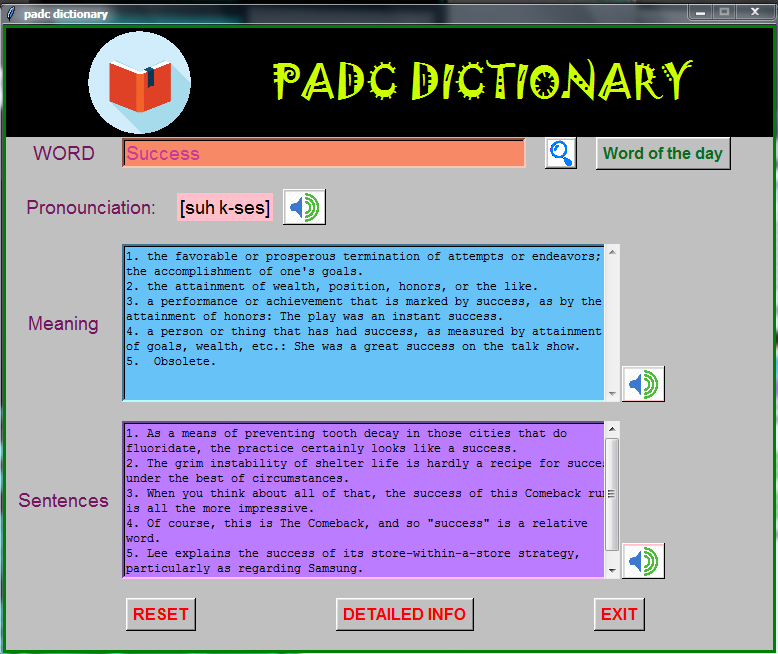
The sentences related to the searched word is displayed in this textbox.

**Here, the outlook of the meaning and sentence of the searched word is displayed.**



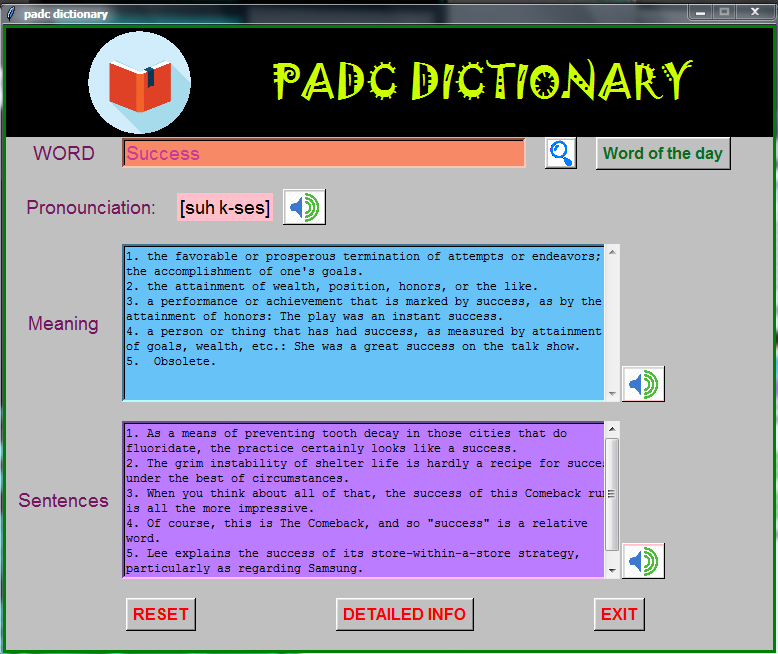


**The moment pronunciation button is clicked an mp3 file gets downloaded**



These are the two text to speech buttons.

**These two buttons will read out all the text written in the textbox. The moment any of these speakers button is clicked it will download the mp3 file which will automatically played the moment it gets downloaded.**



This will exit the software.

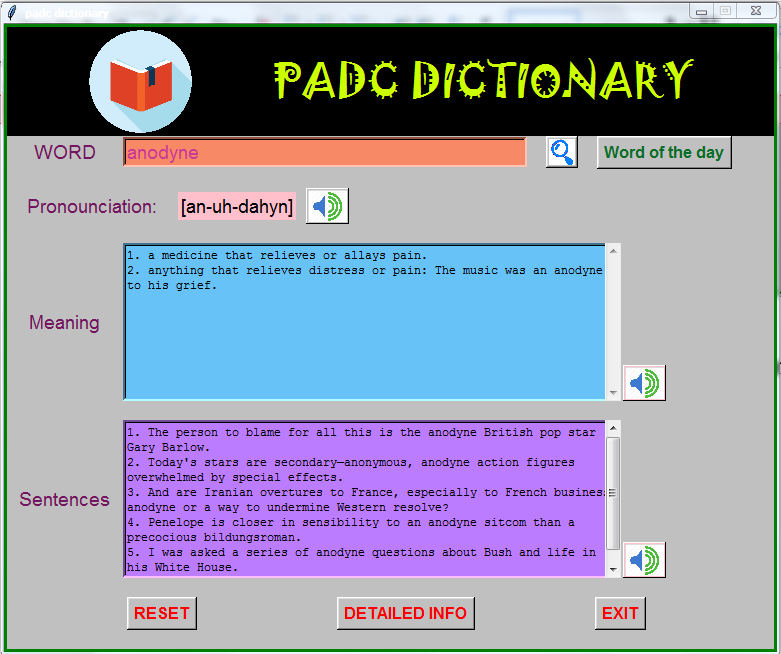
This will redirect the user to the original data source of dictionary.

If new word is to be searched the all entries must be reset.

**Reset button will reset all the data and make the dictionary blank to search another word.**

**Detailed information button will take the user to the source of this dictionary for complete understanding of the word.**

**Exit button will terminate the dictionary.**



This button will give the word of the day

**Word of the day button as the name suggests will give the word of the day. All the same operations can be performed on the word of day the as well.**

**CHAPTER 6**

**CONCLUSIONS AND FUTURE ENHANCEMENT**

**6.1. CONCLUSION**

PADC dictionary has been designed to perform all the tasks which a dictionary is expected to perform. PADC Dictionary has easy and simple terms with interesting features, which makes this model best of the best.

PADC Dictionary provides the following features:

* The most appropriate word meaning.
* Sentences based on that particular word.
* Pronunciation of that particular word by phoenix.
* Pronunciation in the form of voice note as well.
* Text to Speech conversion of word meaning and its sentence.
* To learn a new word every day it provides you the Word Of The Day.

Along with all these features, the most important functional advantage of PADC Dictionary is, the moment any lexicographer puts a new word on the website we don’t need to make any changes. As the webscrapping done will automatically reflect the changes.

**6.2. FUTURE SCOPE**

The current module is fully functional and requires internet connection for its working. The following are the future enhancement of PADC Dictionary:

1. Currently the dictionary is working on internet, so the next aim is to make it an offline application so that if user is out of internet he can still use this dictionary.

A history of all the words searched is to be maintained so that in case a user wants to search any previous word he doesn’t require to type that word again.

**BIBLIOGRAPHY**

* [www.python.org](http://www.python.org)
* [www.tutorialspoint.com](http://www.tutorialspoint.vom)
* [www.github.com](http://www.github.com)
* [www.stackoverflow.com](http://www.stackoverflow.com)
* [www.youtube.com](http://www.youtube.com)
* [www.google.com](http://www.google.com)