Final Presentation

Spring 2017

Vocabulary in Reading

Team Members: **Camilo Rivera and Charles Benitez**
Product Owner: **Seyedjafar Ehsanzadehsorati**
Mentors: **Eric Dwyer and Mohsen Taheri**
Instructor: **Masoud Sadjadi**

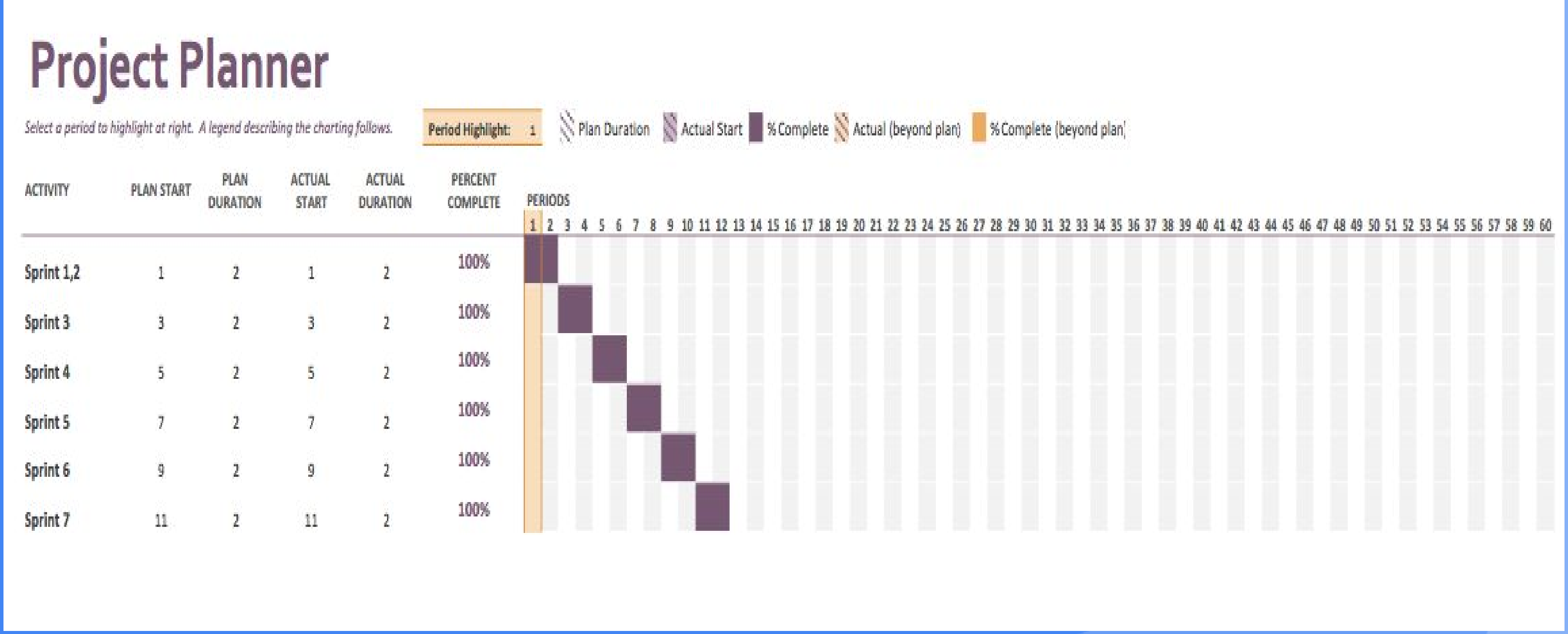
School of Computing and Information Sciences
Florida International University

2.Problem definition

- How can we help students learn a new language more efficiently?
- Can we use technology and linguistics to solve this?

3.Project management: Gantt Chart

Each period represents a week.



Project Planner

Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 1 ▨ Plan Duration ▨ Actual Start ■ % Complete ▨ Actual (beyond plan) ■ % Complete (beyond plan)

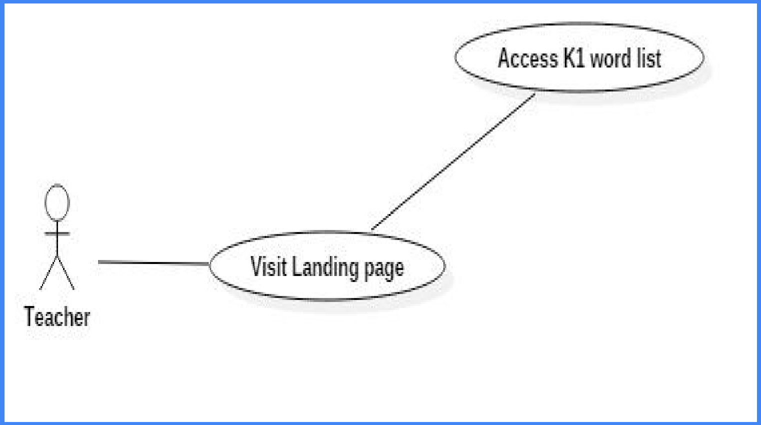
ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Sprint 1,2	1	2	1	2	100%
Sprint 3	3	2	3	2	100%
Sprint 4	5	2	5	2	100%
Sprint 5	7	2	7	2	100%
Sprint 6	9	2	9	2	100%
Sprint 7	11	2	11	2	100%

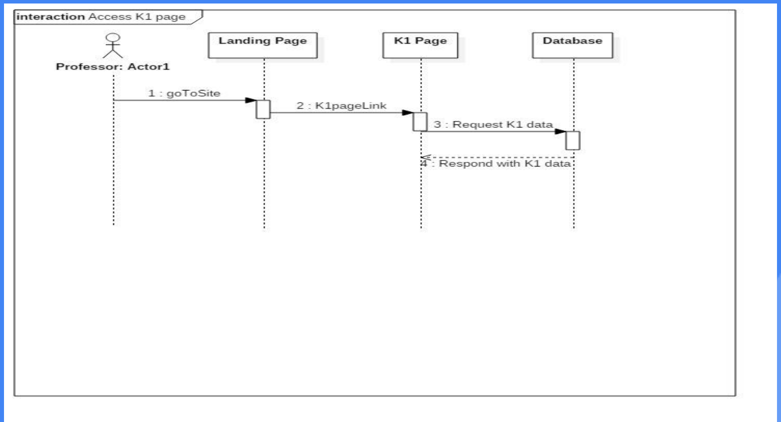
4. Requirements: User Stories

1. Landing Page
2. Set up MySQL Database
3. Interface MySQL to Express/Node
4. Words by category
5. Upload/Scan PDF

User Story #1 Landing Page

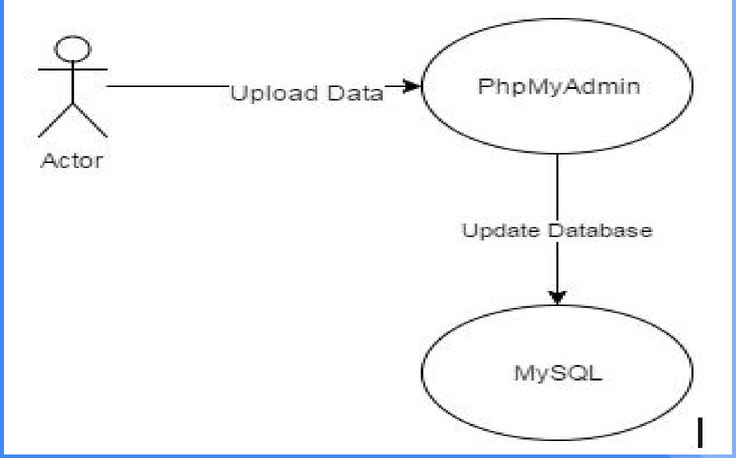
As **a** User **I would like** to choose which word list to see **so that** I can distinguish the word types.

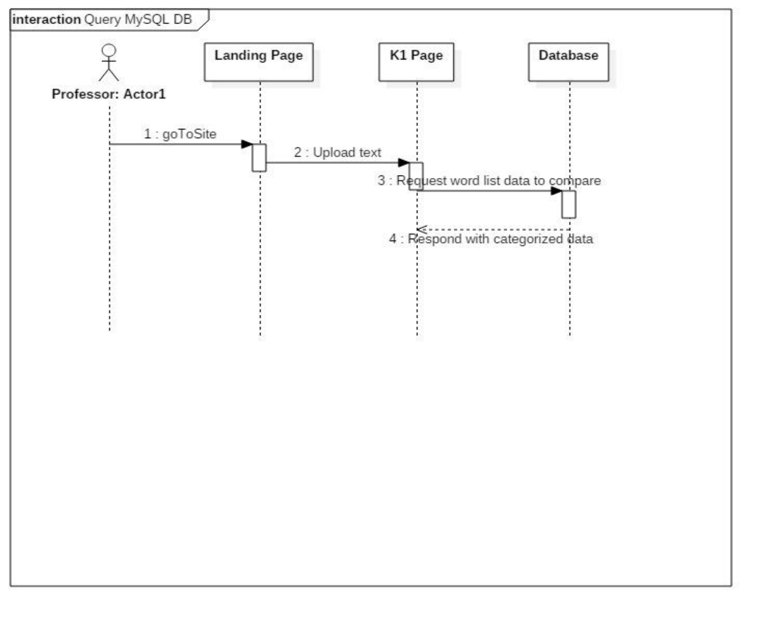




User Story #2 Set up MySQL Database

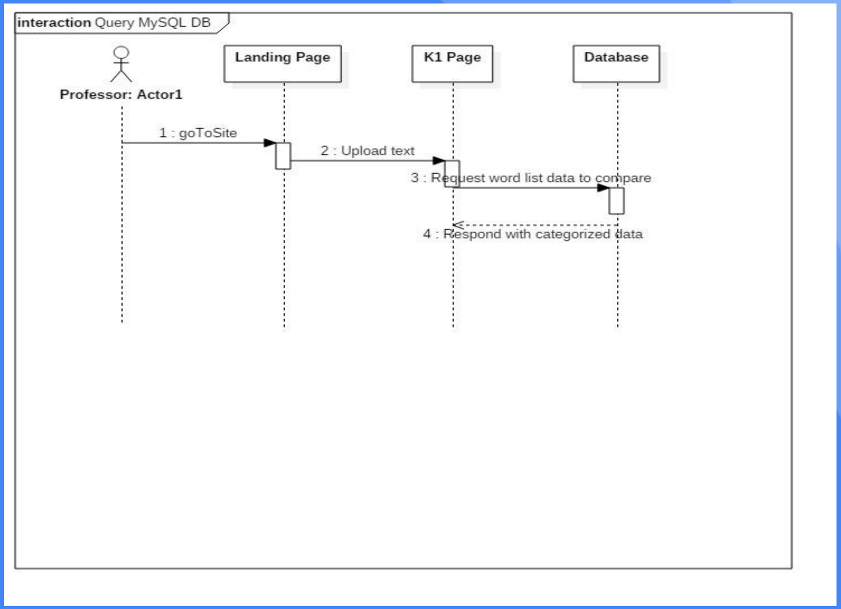
- As a user of the system, I need the system to be able to store data in and query data from an external database.
- As a project manager, I would like to not have to know all of the technicalities of the database, but to simply have a way to interface with the database securely and intuitively.

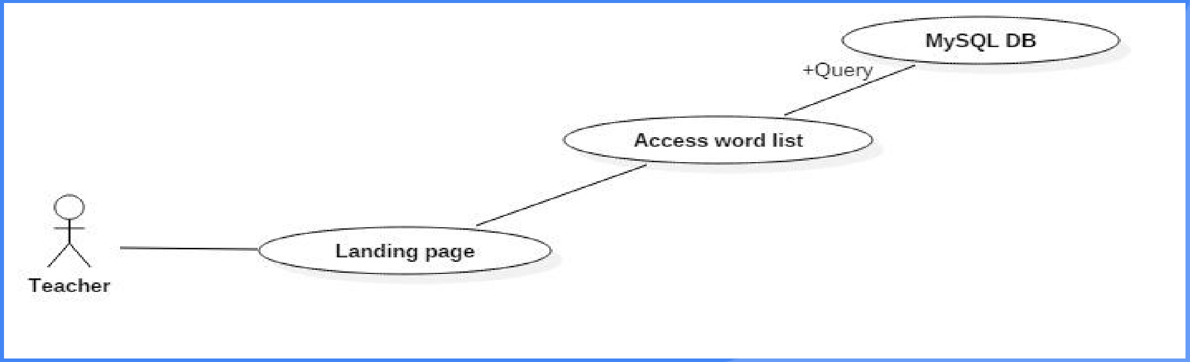



interaction Query MySQL DB
Professor: Actor1
Landing Page
K1 Page
Database
1 : goToSite
2 : Upload text
3 : Request word list data to compare
4 : Respond with categorized data

User Story #3 Interface MySQL to Express/Node

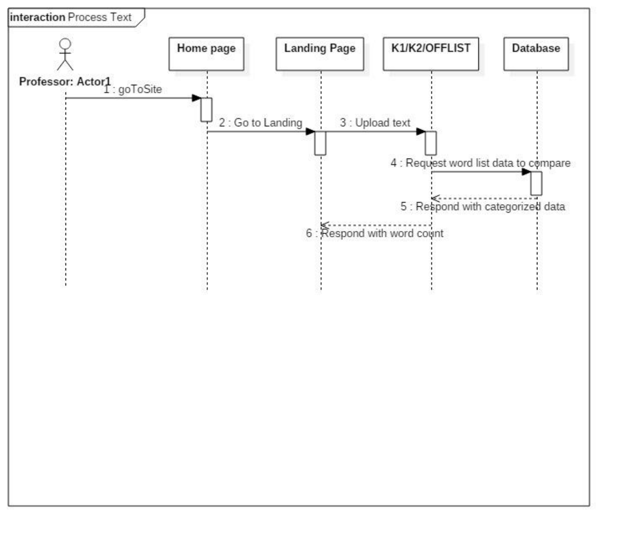
As an admin, I would like my app to know the categories of the words by communicating with the database so that it can compare them.

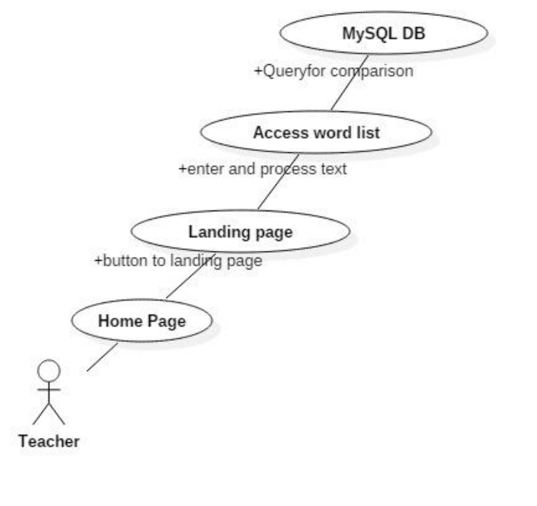

interaction Query MySQL DB
Professor: Actor1
Landing Page
K1 Page
Database
1 : goToSite
2 : Upload text
3 : Request word list data to compare
4 : Respond with categorized data



User Story #4 Words by category

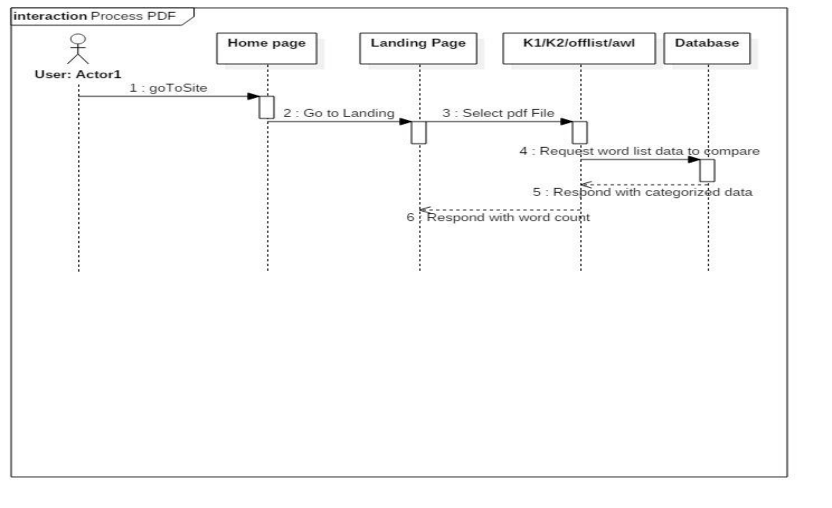
As a user I want to see the different categorize which the words in my submitted text belong to.





User Story #5 Upload/Scan PDF

As a user I want to see the words and their different category from a PDF text file so that I may know more about the words in the file.



interaction Process PDF

User: Actor1

Home page

Landing Page

K1/K2/offlist/awl

Database

1 : goToSite

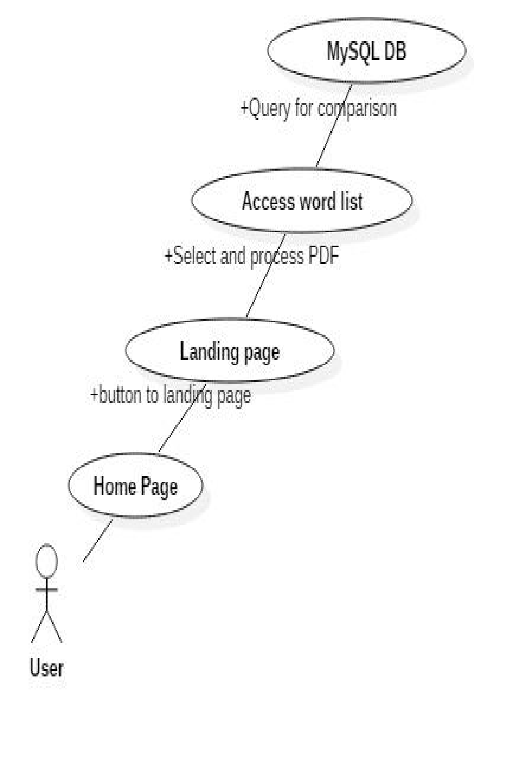
2 : Go to Landing

3 : Select pdf File

4 : Request word list data to compare

5 : Respond with categorized data

6 : Respond with word count



MySQL DB

+Query for comparison

Access word list

+Select and process PDF

Landing page

+button to landing page

Home Page

User

5. System Design

System decomposition:

- VIR Web app implements Model View Controller using MEAN stack.
- VIR Admin Panel implements Client/Server architecture

System deployment

Hardware requirements

- Local development environments (student laptops)
  - Used as XAMPP testing server
- Remote Linux host
  - Hosted by GoDaddy

Software requirements

- MEAN Stack
- PHP5 / MySQL

Persistent data design

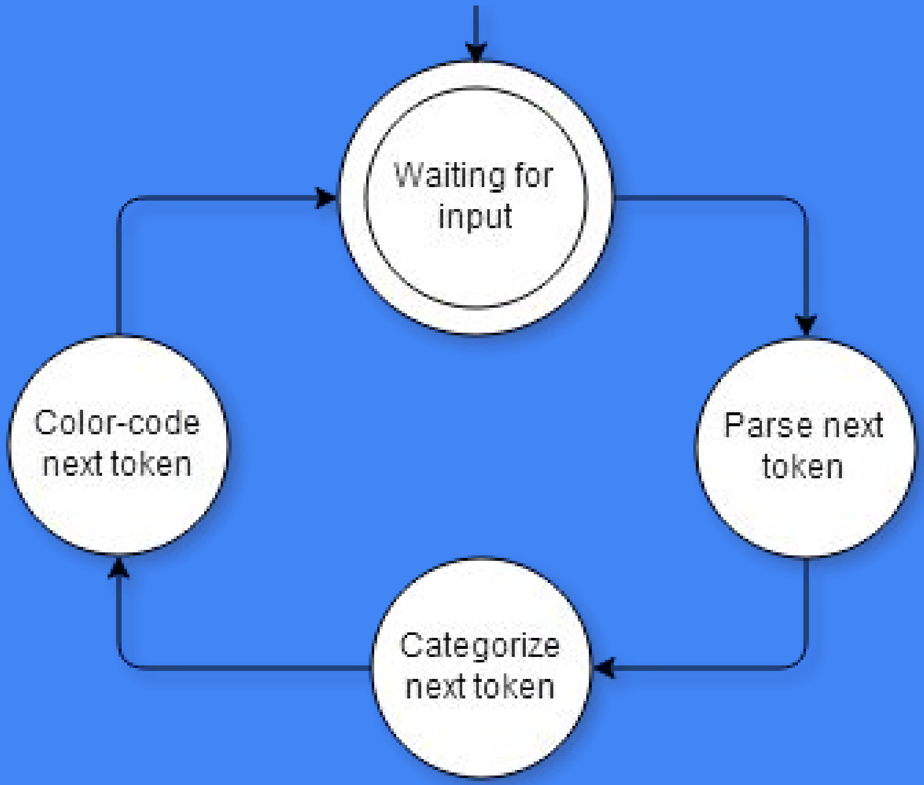
- Persistent data is maintained in our system through the use of MySQL databases. All data used by the app is stored in a single repository, and the two applications use MySQL queries to obtain and update data.

Security and Privacy

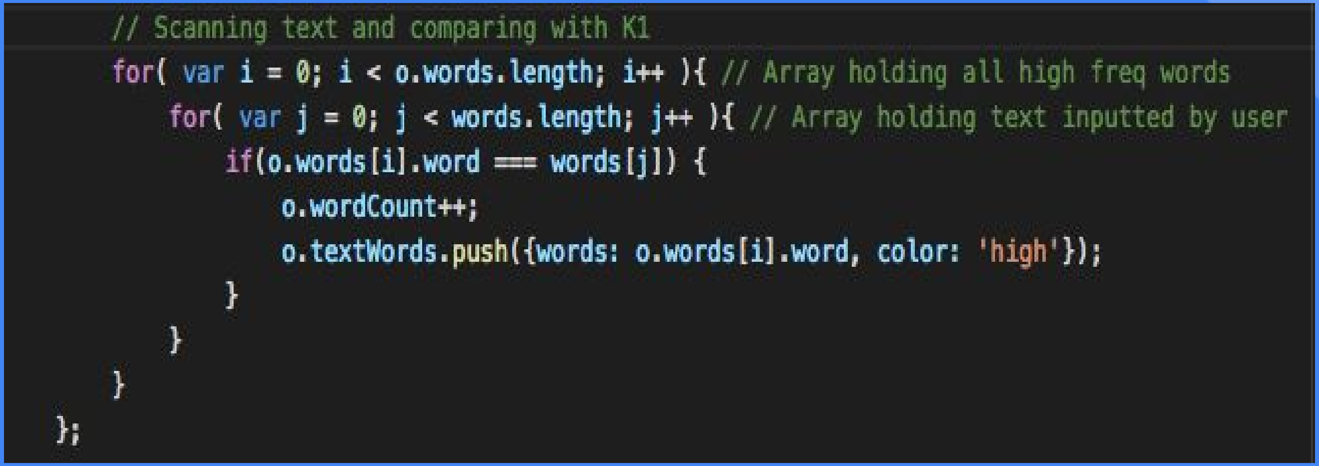
- Privacy is not as much of a concern as no sensitive data is collected from the actors
- Security of the system depends on the security of the hardware. GoDaddy's servers are encrypted and secure using HTTPS and SSH.

Class Diagram

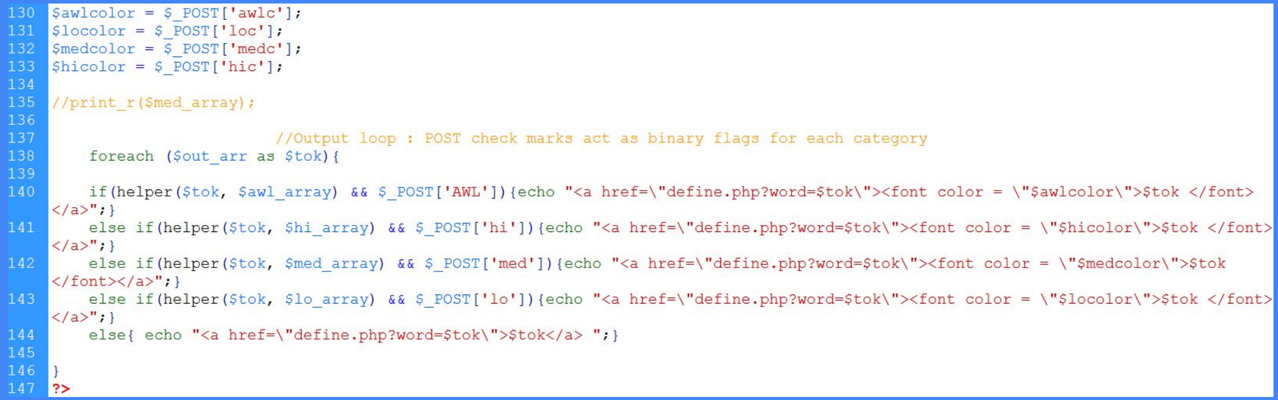
State Machine diagram



Main algorithm



```
    // Scanning text and comparing with K1
    for( var i = 0; i < o.words.length; i++ ){ // Array holding all high freq words
        for( var j = 0; j < words.length; j++ ){ // Array holding text inputted by user
            if(o.words[i].word === words[j]) {
                o.wordCount++;
                o.textWords.push({words: o.words[i].word, color: 'high'});
            }
        }
    }
};
```



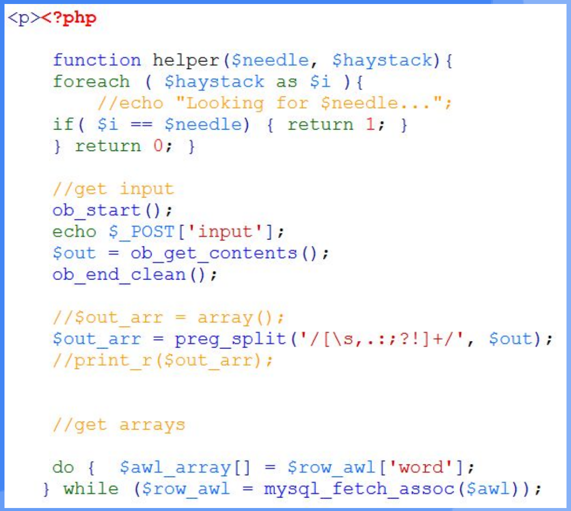
```
$awlcolor = $_POST['awlc'];
$locolor = $_POST['loc'];
$medcolor = $_POST['medc'];
$hicolor = $_POST['hic'];

//print_r($med_array);

                        //Output loop : POST check marks act as binary flags for each category
    foreach ($out_arr as $tok){

    if(helper($tok, $awl_array) && $_POST['AWL']){echo "<a href=\"define.php?word=$tok\"><font color = \"$awlcolor\">$tok </font></a>";}
    else if(helper($tok, $hi_array) && $_POST['hi']){echo "<a href=\"define.php?word=$tok\"><font color = \"$hicolor\">$tok </font></a>";}
    else if(helper($tok, $med_array) && $_POST['med']){echo "<a href=\"define.php?word=$tok\"><font color = \"$medcolor\">$tok </font></a>";}
    else if(helper($tok, $lo_array) && $_POST['lo']){echo "<a href=\"define.php?word=$tok\"><font color = \"$locolor\">$tok </font></a>";}
    else{ echo "<a href=\"define.php?word=$tok\">$tok</a> ";}

}
?>
```



```
<p><?php

    function helper($needle, $haystack){
    foreach ( $haystack as $i ){
        //echo "Looking for $needle...";
    if( $i == $needle) { return 1; }
    } return 0; }

    //get input
    ob_start();
    echo $_POST['input'];
    $out = ob_get_contents();
    ob_end_clean();

    //$out_arr = array();
    $out_arr = preg_split('/[\s,.:;?!]+/', $out);
    //print_r($out_arr);


    //get arrays

    do {  $awl_array[] = $row_awl['word'];
   } while ($row_awl = mysql_fetch_assoc($awl));
```

7. Test Suites and Test Cases

- Test case 1: Test database to make sure words are being stored correctly and reliably: PASS
- Test case 2: Categorizer algorithms correctly detect and prevent garbage input: PASS
- Test case 3: Categorizer algorithms correctly strip input of invalid symbols and punctuation: PASS
- Test case 4: All words are correctly categorized and color coded according to specifications: PASS