

Nichole N Kulobone

200995847

nkulobone@hotmail.com

ADS510s

Cassandra Database Project

Installation

Download Cassandra database version 0.6.5 from the following link

<http://cassandra.apache.org/> and then download thrift 0.4.0.tar.gz from the following link

<http://incubator.apache.org/thrift/download/> . Download jre from

<http://incubator.apache.org/thrift/download/> and then extract Cassandra on your local machine and add the CASSANDRA_HOME path in your environment, do the same for jre when done installing as JAVA_HOME. Do not forget to configure the storage-conf.xml file in Cassandra\conf folder, starting with the cluster name, the directories where data will be saved and adding your own keyspace.

My Environment

Storage-conf.xml file:

I setup my Cassandra in local C: drive and I set Cassandra path to c:\cassandra

And in the Cassandra\conf\storage-conf.xml file I change following;

<ClusterName>Test Cluster</ClusterName> remained the same and added my own keyspace which is ContentManagementSystem and two column families for users and cms which is the message center.

```
<Keyspace Name="ContentManagementSystem">
```

```
    <ColumnFamily Name="cms"
```

```
        CompareWith="UTF8Type"
```

```
        KeysCached="100%" />
```

```
    <ColumnFamily Name="users"
```

```
        CompareWith="UTF8Type"
```

```
        KeysCached="100%" />
```

I changed the directories to the following;

```
<CommitLogDirectory>C:\data\commitlog</CommitLogDirectory>
```

```
<DataFileDirectories>
```

```
    <DataFileDirectory>C:\data\data</DataFileDirectory>
```

```
</DataFileDirectories>
```

```
<CalloutLocation>C:\data\callouts</CalloutLocation>
```

```
<BootstrapFileDirectory>C:\data\bootstrap</BootstrapFileDirectory>
```

```
<StagingFileDirectory>C:\data\staging</StagingFileDirectory>
```

Templates created:

First off I have the class.php which is my Cassandra php wrapper for inserting, retrieving, deleting and updating content.

I created a content management system which lets you create an account first before you can use the application, create two users just for testing purposes of sending emails to each other to see the application`s functionality. The application requires an email, username and password for purposes of calling the column family row[username] and comparing the email and password stored in it. This is how the users column family stores data

```
Username:{  
    Email:"";  
    Password:"";  
}
```

For the cms column family it is as follows:

```
Number:{  
    to:"";  
    from:"";  
    subject:"";  
    content:"";  
    timestamp:"";  
    status:"";  
}
```

Number is a variable that changes, sort of a uuid but I created and test.txt file which stores numbers and retrieves it for the next email to be composed and adds 1 to it before storing data in the database.

For initializing Cassandra

```
$cassandra = new CassandraDB("ContentManagementSystem");
```

Setting the debug on for errors

```
$cassandra->SetDisplayErrors(true);
```

Login.php Template

In order to make sure the user is already in the database and the user input is correct[authenticating] , I passed the data to the same page and I used the following script to authenticate the user;

```
//get the username,email,password inputs
```

```
$username =  
(isset($_POST['username'])&&($_POST['username']!=""))?strip_tags($_POST['username']):";
```

```
$email = (isset($_POST['email'])&&($_POST['email']!=""))?strip_tags($_POST['email']):";
```

```
$password = (isset($_POST['pwd1'])&&($_POST['pwd1']!=""))?strip_tags($_POST['pwd1']):";
```

```
//get data from the database with username as the key
```

```
$records = $cassandra->GetRecordByKey("users",$username);
```

```
//see if there is such data stored in it
```

```
if(($records['email']=="$email")&&($records['password']=="$password"))
```

```
{
```

```
    //email,pass and name
```

```
    $_SESSION['ema'] = $records['email'];
```

```
//redirect after making sure that the user is registered and use sessions
```

```
echo"<script language=\"javascript\" type=\"text/javascript\">";
```

```
echo"window.location=\"default.php\"";
```

```

        echo"</script>";

    }

    else

    {

        echo"Please go back to the login page and register!<br />";

        echo"<a href='index.php'>Login</a>";

    }

```

It redirects to the content management system [**default.php**] where the user can compose email, search mail and see sent mail as well as delete [which is not working currently].

Compose.php Template

There is the default form with to input, subject input, content input and a submit button shown for the user to compose email and once the user submits the data inserted in the form, the data is sent to the same template where it is stored into the data by the following script:

```

// set the default timezone

date_default_timezone_set('Africa/Windhoek');

$curr_date = date("F j, Y, g:i a");


// Setup the path to the thrift library folder

$GLOBALS['THRIFT_ROOT'] = 'C:/xampp/htdocs/cassandra';

// Load up all the thrift stuff

require_once $GLOBALS['THRIFT_ROOT'].'/class.php';

require_once $GLOBALS['THRIFT_ROOT'].'/Thrift.php';

require_once $GLOBALS['THRIFT_ROOT'].'/protocol/TBinaryProtocol.php';

```

```

require_once $GLOBALS['THRIFT_ROOT'].'/transport/TSocket.php';
require_once $GLOBALS['THRIFT_ROOT'].'/transport/TBufferedTransport.php';
require_once $GLOBALS['THRIFT_ROOT'].'/tmp.php';
require_once $GLOBALS['THRIFT_ROOT'].'/key.php';

// Initialize Cassandra

$cassandra = new CassandraDB("ContentManagementSystem");

// Debug on

$cassandra->SetDisplayErrors(true);

//check the last uuid [number] of the last message inserted and add one in the
//key_cms() function

$keyC = key_cms();

//create an array of the data posted to the same template

$record =
array("to"=>"$to", "from"=>"$from", "subject"=>"$subject", "content"=>"$content",
      "timestamp"=>"$curr_date", "status"=>"unread");

//if insertion was successful, show message and information inserted in database
if ($cassandra->InsertRecordArray("cms", $keyC, $record))
{
    $records = $cassandra->GetRecordByKey("cms", $keyC);

    echo"<span style='font-family:arial; font-size:10pt;color:#045FB4;'>Your
message has been sent!</span><br />";

    echo"<span style='font-family:arial; font-
size:10pt;color:#045FB4;'>To:</span>

    <span style='font-family:arial; font-size:10pt;color:black;'>".
    $records['to'].<span><br />";

```

```
}
```

Search.php Template

Similar to compose as well, it has its own form with the keyword input and a search button where the user inputs a keyword to be searched in the messages composed or received by the certain user. And once the data is submitted the following code is processed:

```
//keyword posted is retrieved
```

```
$search = (isset($_POST['inner']))?strip_tags($_POST['inner']):'';
```

```
//call necessary libraries
```

```
$GLOBALS['THRIFT_ROOT'] = 'C:/xampp/htdocs/cassandra';
```

```
// Load up all the thrift stuff
```

```
require_once $GLOBALS['THRIFT_ROOT'].'/class.php';
```

```
require_once $GLOBALS['THRIFT_ROOT'].'/Thrift.php';
```

```
require_once $GLOBALS['THRIFT_ROOT'].'/protocol/TBinaryProtocol.php';
```

```
require_once $GLOBALS['THRIFT_ROOT'].'/transport/TSocket.php';
```

```
require_once $GLOBALS['THRIFT_ROOT'].'/transport/TBufferedTransport.php';
```

```
require_once $GLOBALS['THRIFT_ROOT'].'/tmp.php';
```

```
require_once $GLOBALS['THRIFT_ROOT'].'/key.php';
```

```
$emailD = $_SESSION['ema'] ;
```

```
$usernameD = $_SESSION['usrn'];
```

```
$passwordD = $_SESSION['pass'];
```

```
// Initialize Cassandra
```

```
$cassandra = new CassandraDB("ContentManagementSystem");
```

```
// Debug on
```

```

$cassandra->SetDisplayErrors(true);

for($i=0;$i<=100;$i++)
{
    //look through all the messages/emails in the database

    $records = $cassandra->GetRecordByKey("cms",$i);

    //search to or from by email, or subject, or content

    if($records['to']=="$search" || $records['from']=="$search" ||
    $records['subject']=="$search" || $records['content']=="$search")
    {
        echo"<tr>";

        echo"<td colspan='1' valign='top' width='30%' style='font-family:arial; font-
size:10pt;color:#045FB4;'>";

            if($records['to']=="")
            {
                echo"Empty]";
            }
            else
            {
                echo "[".$records['to'];
            }
        echo "|";

            if($records['from']=="")
            {
                echo"Empty]";
            }
            else

```



```

        {
            echo $records['from'].");
        }

    echo"</td>";

//output findings

    echo"<td colspan='1' valign='top' with='48%' style='font-family:arial; font-
size:10pt;color:#045FB4;'>";

    echo "<a href='default.php?do=message&key=$i&type=read' style='font-
family:arial; font-size:10pt;color:#045FB4;'>".

        $records['subject']. "</a>";

    echo"</td>";

    echo"<td colspan='1' valign='top' width='25%' style='font-family:arial; font-
size:10pt;color:#045FB4;'>";

    echo $records['timestamp'];

    echo"</td>";

        echo"<td colspan='1' valign='top' >";

        echo"<a href='default.php?do=delete&key=$i' style='text-decoration:none;font-
family:arial; font-size:8pt;color:red;'>[DELETE]</a>";

    echo"</td>";

    echo"</tr>";

        }

    }

```

Incoming.php Template

For messages received the following code is processed:

[Connection is established and necessary libraries are included]

```

for($i=0;$i<=100;$i++)
{
//go through all records in the database

    $records = $cassandra->GetRecordByKey("cms",$i);

//if the to value is similar to the logged in user then retrieve all messages for the user

    if($records['to']=="$emailD")
    {

        echo"<tr>";

        echo"<td colspan='1' valign='top' width='30%' style='font-family:arial; font-size:10pt;color:#045FB4;'>";

        echo $records['from'];

        echo"</td>";

        echo"<td colspan='1' valign='top' width='48%' style='font-family:arial; font-size:10pt;color:#045FB4;'>";

        echo "<a href='default.php?do=message&key=$i&type=read' style='font-family:arial; font-size:10pt;color:#045FB4;'>".

            $records['subject']."</a>";

        echo"</td>";

        echo"<td colspan='1' valign='top' width='25%' style='font-family:arial; font-size:10pt;color:#045FB4;'>";

        echo $records['timestamp'];

        echo"</td>";

        echo"<td colspan='1' valign='top' style='font-family:arial; font-size:10pt;color:#045FB4;'>";

        echo"</td>";

        echo"<td colspan='1' valign='top' >";

```

```

        echo"<a href='default.php?do=delete&key=$i' style='text-decoration:none;font-
family:arial; font-size:8pt;color:red;'>[DELETE]</a>";

        echo"</td>";

        echo"</tr>";

    }

}

```

Outgoing.php Template

It is similar to the outgoing.php template only that the following script

//if the to value is similar to the logged in user then retrieve all messages for the user

```
if($records['to']=="$emailD")
```

changed to the following:

//if the to value is similar to the logged in user then retrieve all messages sent by the
//user

```
if($records['from']=="$emailD")
```

Message.php Template

This template shows the full content of messages received or sent by a certain user by retrieving an id in the url and sent/received messages are differentiated by type which is also in the url.

//message id or row

```
$i =(isset($_GET['key']))?strip_tags($_GET['key']):";
```

//retrieve the message

```
$records = $cassandra->GetRecordByKey("cms",$i);
```

//type

```
$type = ($_GET['type'])?strip_tags($_GET['type']) :";
```

//if read

```
if($type=="read")
```

```
{
```

```

$top = "From";
$client = $records['from'];
}
else if($type=="sent") //if sent
{
    $top = "To";
    $client = $records['to'];
}

```

For differentiating email sent by/sent to in the header of the, to/from user content.

Delete.php Template

I still haven't wrote the delete function yet but it is suppose to just delete a message by id retrieved in the url since it is passed by url.

Default.php Template

Default template for all the links in the content management system and it retrieves the type of action it is suppose to act on from the url and passes it to the tmp.php template where it is processed by the following script.

First in the default.php template the we include the tmp.php

```
require_once $GLOBALS['THRIFT_ROOT'].'/tmp.php';
```

//then the action in the url is requested

```
$action=(isset($_REQUEST['do']))?strip_tags($_REQUEST['do']):'';
```

//if the action is set then we call the Cms function is called and data is spat out.

```
if(isset($action) &&($action!=""))
```

```
{
```

```
    $OutPut = Cms($action);
```

```
    echo $OutPut;
```

```
}
```

```
function Cms($x)
{

switch ($x)
{
case'incoming':
    require_once'incoming.php';
    break;

case'outgoing':
    require_once'outgoing.php';
    break;

case'search':
    require_once'search.php';
    break;

case'compose':
    require_once'compose.php';
    break;
```

case'insert':

 require_once'compose.php';

break;

case'delete':

 require_once'delete.php';

break;

case'message':

 require_once'message.php';

break;

case'logout':

 //session_start();

 unset(\$_SESSION['email']);

 unset(\$emailD);

 unset(\$_SESSION['pass']);

 unset(\$passwordD);

 unset(\$_SESSION['usrn']);

 unset(\$usernameD);

 // session_destroy();

 echo"<script language=\"javascript\" type=\"text/javascript\">";

 echo"window.location=\"index.php\"";

 echo"</script>";

```
        break;

default:

    require_once"login.php";

}

}
```

After processing the data is spat back to the default page.

Users I created are:

Email	Username	Password
1. nkulobone@hotmail.com	nicolas	kulobone
2. test@123.com	test	test
3. abc@123.com	abc	abc

Ones in the system already, you can create your own users for testing purposes.

Copy all the folders and files into the necessary place,run Cassandra server and you are up and running.