

SKYGRAPH

Software Test Report

Version 1.0

2014/11/25

**Software Engineering CS440
GROUP 13**

**Hardik Prajapati
Junior Luna
Keith Thomas
Gowdhaman Sadhasivam**

**University of Illinois Chicago
Fall 2014**

Table of Contents

I. Introduction	3
1. Document Overview	3
II. Overview of Test Results	3
2. Unit Testing - Release I.....	3
3. Unit Testing - Release II.....	3
4. Integration Testing.....	3
5. Functional Testing.....	3
6. Black-Box Testing	3
7. White-Box Testing.....	4
III. Detailed Test Results	4
8. Unit Testing - Release I.....	4
9. Unit Testing - Release II.....	5
10. Integration Testing.....	6
11. Functional Testing.....	8
12. Black-Box Testing	11
13. White-Box Testing.....	14
IV. Inspections.....	22

I. Introduction

1. Document Overview

This document is the software test report of SkyGraph application. It contains results of tests , which were executed during the testing phase of SkyGraph application.

II. Overview of Test Results

The SkyGraph application was tested on Windows test platform from 2014/09/25 to 2014/11/25. The tests were executed and ,

Testers are:

- Junior Luna
- Hardik Prajapati
- Keith Thomas
- Gowdhaman

2. Unit Testing - Release I

Test Date - 2014/09/25 - 2014/10/05

Test Results - 90% of test cases passed with one medium defect.

Comments - Defects was raised and the fix for respective defects were implemented and has been deployed

3. Unit Testing - Release II

Test Date - 2014/10/20 - 2014/11/09

Test Results - 85 % of test cases passed with one high and medium severity defects.

Comments - Defects was raised and the fix for respective defects were implemented and has been deployed

4. Integration Testing

Test Date - 2014/11/10 - 2014/11/13

Test Results - 92% of test cases passed with one high severity defect.

Comments - Defects was raised and the fix for respective defects were implemented and has been deployed

5. Functional Testing

Test Date - 2014/09/25 - 2014/11/25

Test Results - 93% of test cases passed with one medium severity defect.

Comments - Defects was raised and the fix for respective defects were implemented and has been deployed

6. Black-Box Testing

Test Date - 2014/09/25 - 2014/11/15

Test Results - 100% of test cases passed with no defect.

Comments - No defects have been raised

7. White-Box Testing

Test Date - 2014/09/25 - 2014/11/25

Test Results - 100% of test cases passed with no defect.

Comments - No defects have been raised

III. Detailed Test Results

8. Unit Testing - Release I

Unit testing was done during development of the application. The following table summarizes the result of unit testing done for release I.

Test Case #	Test Date	Test Case	Tester	Test Status	Severity	Summary of Defects	Comments
UT-1	2014/09/25	Home page is displayed with 'Current Location' weather details	Gowdhaman	Pass			
UT-2	2014/09/28	Clicking of different cities option will display respective weather details	Junior Luna	Pass			
UT-3	2014/09/30	Clicking on 'Help' menu will display SkyGraph information with detailed screen shot images	Hardik Prajapati	Fail	Medium	Defect - 1: Images were not loaded	Defect fixed and deployed in Release II
UT-4	2014/10/04	Click on 'Create account' menu, user can register with application	Keith Thomas	Pass			
UT-5	2014/10/05	Click on 'Sign In' menu, user can successfully login into SkyGraph by providing valid user credentials	Gowdhaman	Pass			
UT-6	2014/10/05	Click on 'Logout' menu user will be logged out from SkyGraph	Junior Luna	Pass			

9. Unit Testing - Release II

The following table summarizes the results of unit testing done for release II.

Test Case #	Test Date	Test Case	Tester	Test Status	Severity	Summary of Defects	Comments
UT-1	2014/10/20	Click on 'Widgets' menu, it will display widgets home page	Gowdhaman	Pass			
UT-2	2014/10/20	Click on 'Import', will show file upload page	Junior Luna	Pass			
UT-3	2014/10/20	Click on 'Choose File' button will open file explorer window that allows user to select business file from their system	Hardik Prajapati	Pass			
UT-4	2014/10/25	Click on 'Upload' button will upload the selected business file into the SkyGraph	Keith Thomas	Pass			
UT-5	2014/10/25	Click on 'Compare' menu in Widgets home page will show Compare page that allows user to compare their business data with weather data	Gowdhaman	Pass			
UT-6	2014/10/25	In Compare window, from 'Select your business data' and 'City' drop down list, user select necessary options and choose fiscal quarter.	Junior Luna	Pass			
UT-7	2014/10/25	Click on 'Compare' button, a graph will be shown that compares the business data and	Hardik Prajapati	Pass			

		weather data of selected city					
UT-8	2014/11/01	Click on 'Share email' option will send current city weather details and future forecast will be sent to user registered email address	Keith Thomas	Fail	High	Defect-2: Connection to email server failed	Defect was fixed during Release II
UT-9	2014/11/01	Click on 'Export' option will download weather data to user system	Gowdhaman	Pass			
UT-10	2014/11/01	In Widgets page, by clicking on 'Account Info' menu will display user account details in account page	Junior Luna	Pass			
UT-11	2014/11/05	In account page, by click on 'Edit Account' button will display 'Enter New Details' page	Hardik Prajapati	Fail	Medium	Defect-3 : Page URL is not mapped	Defect was fixed in Release II
UT-12	2014/11/05	In 'Enter New Details' page, it allows user to enter new user information. By click on 'Submit' button, new details will be saved	Keith Thomas	Pass			
UT-13	2014/11/09	In Widgets page, by clicking on 'Account Info' menu will display new user account details in account page	Keith Thomas	Pass			

10. Integration Testing

The bottom-up testing strategy is used in this integration testing. Each unit is developed and tested individually first and they were integrated with its parent unit. Below table summarizes the result of integration testing done during development of this application.

Test Case #	Test Date	Test Case	Tester	Test Status	Severity	Summary of Defects	Comments
IT-1	2014/09/25	Integration of different cities home page with SkyGraph home page	Gowdhaman	Pass			
IT-2	2014/09/28	Integration of 'Current Location' menu with home page	Junior Luna	Pass			
IT-3	2014/10/05	Integration of 'Create account' function with SkyGraph Home page	Hardik Prajapati	Pass			
IT-4	2014/10/10	Integration of 'Sign In' function with SkyGraph home page	Keith Thomas	Pass			
IT-5	2014/10/15	Integration of 'Help' page with SkyGraph home page	Gowdhaman	Pass			
IT-6	2014/11/01	Integration of 'Import' function with Widgets page	Junior Luna	Fail	High	Defect -3: Imported file location is not found	Defect was fixed and deployed in Release II
IT-7	2014/11/12	Integration of 'Compare' function with Widgets page	Hardik Prajapati	Pass			
IT-8	2014/11/12	Integration of 'Share email' function with Widgets page	Keith Thomas	Pass			
IT-9	2014/11/12	Integration of 'Export' function with	Gowdhaman	Pass			

Widgets page				
IT-10	2014/11/12	Integration of Widgets page with SkyGraph home page	Junior Luna	Pass
IT-11	2014/11/13	Integration of 'Logout' menu with SkyGraph home page	Hardik Prajapati	Pass
IT-12	2014/11/13	Integration of 'Enter New Details' with Edit Account function	Keith Thomas	Pass
IT-13	2014/11/13	Integration of 'Edit Account' function with Account Info page	Gowdhaman	Pass
IT-14	2014/11/13	Integration of 'Account Info' page with SkyGraph home page	Junior Luna	Pass

11. Functional Testing

The following table summarizes the test cases for functional testing and test results of each test cases.

Test Case #	Test Date	Test Case	Tester	Test Status	Severity	Summary of Defects	Comments
FT-1	2014/09/25	SkyGraph displays current weather details for given all cities in home page, by clicking respective cities	Gowdhaman	Pass			

FT-2	2014/09/28	SkyGraph displays weather details of current location by clicking 'Current Location' option in home page	Junior Luna	Pass
FT-3	2014/10/10	By clicking on 'Create account' option, SkyGraph will allow user to create new SkyGraph account	Hardik Prajapati	Pass
FT-4	2014/10/15	SkyGraph will allow user to login by providing registered login details in 'Sign In' page	Keith Thomas	Pass
FT-5	2014/11/10	By Clicking on 'Widgets' menu, SkyGraph will show widgets page	Gowdhaman	Pass
FT-6	2014/11/15	From Widgets page, click on 'Import' menu, SkyGraph will allow user to input business data to SkyGraph	Junior Luna	Pass
FT-7	2014/11/20	Click on 'Compare' menu, SkyGraph will allow user to compare uploaded business data with weather data of user selected city	Hardik Prajapati	Pass

FT-8	2014/11/23	Click on 'Share email' option SkyGraph will send email of current weather and future forecast to registered email address	Gowdhaman	Pass				
FT-9	2014/11/24	Click on 'Export' option SkyGraph will allow user to download weather report to user's local system	Keith Thomas	Pass				
FT-10	2014/11/24	Click on 'Account Info' menu SkyGraph will display user's account details	Junior Luna	Pass				
FT-11	2014/11/25	SkyGraph will allow user to edit account details and new user information will be updated when it is saved	Hardik Prajapati	Fail	Medium	Defect - 4: Inconsistency User details in master database	Defect was fixed on 2014/11/25	
FT-12	2014/11/25	SkyGraph will allow user to logout from it by clicking 'Logout' option from any of its page	Keith Thomas	Pass				
FT-13	2014/11/25	SkyGraph will display details help information with images when user clicks 'Help' menu from any of its page	Gowdhaman	Pass				

12. Black-Box Testing

The following table summarizes the results of black box testing employed for each test case.

Test #	Function:Class	Tester	Description	Expected Result	Test status
BB-01	home() : Views	Gowdhaman	1. Get URL from browser 2. Check if user is logged in 3. Display SkyGraph home page with login and account creation option	The application should display SkyGraph home page with menus to create new account and Sign In	Pass
BB-02	area() :Views	Junior Luna	1. Get requested function from URL 2. Check if user is logged in 3. Display weather details of requested city with login and account creation option	The application should display weather details of requested city	Pass
BB-03	newy() :Views	Hardik Prajapati	1. Get requested function from URL 2. Check if user is logged in 3. Display weather details of requested city with login and account creation option	The application should display weather details of requested city	Pass
BB-04	losa() :Views	Keith Thomas	1. Get requested function from URL 2. Check if user is logged in 3. Display weather details of requested city with login and account creation option	The application should display weather details of requested city	Pass
BB-05	hous() :Views	Gowdhaman	1. Get requested function from URL 2. Check if user is logged in 3. Display weather details of requested city with login and account creation option	The application should display weather details of requested city	Pass
BB-06	sanf() :Views	Junior Luna	1. Get requested function from URL 2. Check if user is logged in 3. Display weather details of requested city with login and account creation option	The application should display weather details of requested city	Pass
BB-07	login() :Views	Hardik Prajapati	1. Check if user is already loggedin 2. If not, allow user to enter	The application should display login page	Pass

SkyGraph login details					
BB-08	login() :Views	Keith Thomas	1. Check if user is already loggedin 2. Display the SkyGraph home page with 'Logout' menu	The application should display SkyGraph home page with logout menu	Pass
BB-09	invalid_login() : Views	Gowdhaman	1. Check user entered login details 2. If user login in details are not valid, display invalid login message with 'Create account' and 'Sign In' option	The application should display invalid login error page with 'Create account' and 'Sign In' option	Pass
BB-10	auth_view() : Views	Junior Luna	1. Get user entered login details 2. Check if user is registered with SkyGraph 3. Check user login details are valid 4. Allow valid user to login into SkyGraph	The application should allow valid user to successfull sign in into SkyGraph	Pass
BB-11	register_user() : Views	Hardik Prajapati	1. Validate the new account details entered by user 2. Check if user already registered with SkyGraph 3. Display error message if user is already registered	The application should display error message if user is already registered	Pass
BB-12	register_user() : Views	Keith Thomas	1. Validate the new account details entered by user 2. if user is not already registered with SkyGraph, save user account details 3. Display account created confirmation page	The application should display account created confirmation page with option to 'Sign In'	Pass
BB-13	help() : Views	Gowdhaman	1. Get the requested URL 2. Check is user is loggedin 3. If user is loggedin, display help page with 'Logout' and 'Widgets' menu	The application should display help page with 'Widgets' and 'Logout' menu	Pass
BB-14	help() : Views	Junior Luna	1. Get the requested URL 2. If user is not loggedin, display help page with 'SignIn' and 'Create account' option	The application should display help page with 'SignIn' and 'Create account' option	Pass

BB-15	widgets() : Views	Hardik Prajapati	1. Get the requested URL 2. Check if user is loggedin 3. If user is not loggedin, display login page	The application should display login page	Pass
BB-16	widgets() : Views	Keith Thomas	1. Get the requested URL 2. Check if user is loggedin 3. If user is loggedin, display widgets home page	The application should display widgets home page	Pass
BB-17	list() : Views	Gowdhaman	1. If user is loggedin, get the file choosen by user 2. Upload the file to SkyGraph when user click upload button 3. Display uploaded file name in the file list	The application should display uploaded file name in file list	Pass
BB-18	compare() : Views	Junior Luna	1. Check if user is loggedin 2. If user is not loggedin, display login page	The application should display login page	Pass
BB-19	compare() : Views	Hardik Prajapati	1. If user is loggedin, display drop down list for weather data and uploded business data 2. Display fiscal quarter drop down list as data parameters 3. Display graph that compares weather and business data when 'Compare' button is clicked	The application should display a graph with weather data and business data	Pass
BB-20	share() : Views	Keith Thomas	1. Check if user is loggedin 2. If user is not loggedin, display login page	The application should display login page	Pass
BB-21	share() : Views	Gowdhaman	1.If user is loggedin, send current weather and future forecast to user's registered email address 2. Display email sent confirmation page	The application should send email of current and future weather to user's registered email address	Pass
BB-22	export(): Views	Keith Thomas	1. Check if user is loggedin 2. If user is not loggedin, display login page	The application should display login page	Pass
BB-23	export(): Views	Keith Thomas	1.If user is loggedin, download the weather details to user local system 2. Display file download confirmation page	The application should download weather file to user local system folder	Pass

BB-24	account() : Views	Gowdhaman	1. Check if user is loggedin 2. If user is not loggedin, display login page	The application should display login page	Pass
BB-25	account() : Views	Junior Luna	1. Check if user is loggedin 2.If user is loggedin, display user's SkyGraph account details	The application should display user's account details	Pass
BB-26	editacc() : Views	Hardik Prajapati	1. Check if user is loggedin 2. If user is not loggedin, display login page	The application should display login page	Pass
BB-27	editacc() : Views	Keith Thomas	1.If user is loggedin, display edit account details page when 'Edit Account' button is clicked 2. Save updated user's account details when it is submitted 3. Display user's updated account details when 'Account Info' page is clicked	The application should display update user's account details in 'Account Info' page	Pass
BB-28	logout() : Views	Gowdhaman	1. Display logout option when user is loggedin already 2. Display login page when user is loggedout from SkyGraph	The application should display login page	Pass

13. White-Box Testing

The following table summarizes the results of white box testing employed for each test case.

Test #	Function:Class	Tester	Description	Expected Result	Test status
WB-01	home() : Views	Gowdhaman	1. Get URL from browser 2. Create instance of 'Views' class in 'urls' class 3. Invoke home() method and redirect user to home page	Home page is displayed either with 'Logout' menu if user is loggedin or 'Sign In' menu is user is not loggedin into SkyGraph	Pass

			4. Check client side validating if user is loggedin or not		
WB-02	area() :Views	Junior Luna	1. Get URL from browser 2. Create instance of 'Views' class in 'urls' class 3. Invoke area() method 4. Create instance of 'getweather' class and invoke yahoow(), getww() and getweather() method 5. Redirect user to requested page with returned values from above method 5. Check client side validating if user is loggedin or not	Requested URL page should be displayed with returned values from invoked methods. The page should be shown only when user is loggedin into SkyGraph	Pass
WB-03	newy() :Views	Hardik Prajapati	1. Get URL from browser 2. Create instance of 'Views' class in 'urls' class 3. Invoke newy() method 4. Create instance of 'getW4ny' class and invoke yahony(), noaany() and weatny() method 5. Redirect user to requested page with returned values from above method 5. Do client side validating if user is loggedin or not	Requested URL page should be displayed with returned values from invoked methods. The page should be shown only when user is loggedin into SkyGraph	Pass

WB-04	losa() :Views	Keith Thomas	<ol style="list-style-type: none"> 1. Get URL from browser 2. Create instance of 'Views' class in 'urls' class 3. Invoke losa() method 4. Create instance of 'getW4la' class and invoke yahola(), noaala() and weatnla() method 5. Redirect user to requested page with returned values from above method 5. Do client side validating if user is loggedin or not 	Requested URL page should be displayed with returned values from invoked methods. The page should be shown only when user is loggedin into SkyGraph	Pass
WB-05	hous() :Views	Gowdhama n	<ol style="list-style-type: none"> 1. Get URL from browser 2. Create instance of 'Views' class in 'urls' class 3. Invoke hous() method 4. Create instance of 'getW4ho' class and invoke yahoho(), noaaho() and weath() method 5. Redirect user to requested page with returned values from above method 5. Do client side validating if user is loggedin or not 	Requested URL page should be displayed with returned values from invoked methods. The page should be shown only when user is loggedin into SkyGraph	Pass
WB-06	sanf() :Views	Junior Luna	<ol style="list-style-type: none"> 1. Get URL from browser 2. Create instance of 'Views' class in 'urls' class 3. Invoke sanf() method 4. Create instance of 'getW4ho' class and invoke yahosa(), noaasa() and weatsa() 	Requested URL page should be displayed with returned values from invoked methods. The page should be shown only when user is loggedin into SkyGraph	Pass

			method 5. Redirect user to requested page with returned values from above method 5. Do client side validating if user is loggedin or not		
WB-07	login() :Views	Hardik Prajapati	1. When 'Sign In' menu is clicked, create instance of 'Views' class 2. Invoke login() method using created instance from 'urls' class 3. Create 'csrf()' method and redirect the user to login page	csrf() request should be updated and login page should be displayed	Pass
WB-08	auth_view() : Views	Keith Thomas	1. When user submitted login details, create instance of 'Views' class 2. Invoke the auth_view() method from 'urls' class 3. auth_view() method should get the user name and encrypted password from POST HttpRequest 4. Create instance of 'auth' class and invoke login() method to validate the login credentials 5. If user account is valid redirect the user to loggedin page	Instance of 'Views' class should be created and auth_view() method should be invoked. Username and password should be get from session and login() method of 'auth' class should be invoked	Pass
WB-09	auth_view() : Views	Gowdhama n	1. When user submitted login details, create instance of 'Views' class 2. Invoke the auth_view() method from 'urls' class 3. auth_view() method should get the user name and encrypted password from POST	Instance of 'Views' class should be created and auth_view() method should be invoked. Username and password should be get from session and login() method of 'auth' class should be	Pass

			<p>HttpRequest</p> <p>4. Create instance of 'auth' class and invoke login() method to validate the login credentials</p> <p>5. If user account is not valid redirect the user to invalid login page</p>	invoked	
WB-10	register_user() : Views	Junior Luna	<p>1. When Sign In menu is clicked and register form is submitted, create instance of 'Views' class</p> <p>2. Invoke register_user() method from 'urls' class</p> <p>3. Get user account details form and invoke MyRegistrationForm() method of 'Myform' class</p> <p>4. Validate the form and return the result</p> <p>5. If form is valid redirect user to register success page</p>	<p>Register page should be displayed with user account form. User account details should be get from HttpRequest and send it to validate the submitted form.</p> <p>Register success page should be displayed when from is valid</p>	Pass
WB-11	register_user() : Views	Hardik Prajapati	<p>1. When Sign In menu is clicked, create instance of 'Views' class</p> <p>2. Invoke register_user() method from 'urls' class</p> <p>3. Get user account details form and invoke MyRegistrationForm() method of 'Myform' class</p> <p>4. Validate the form and return the result</p> <p>5. If form is not valid redirect user to register page</p>	<p>Register page should be displayed with user account form. User account details should be get from HttpRequest and send it to validate the submitted form.</p> <p>Register again page should be displayed when from is invalid</p>	Pass

WB-12	help() : Views	Keith Thomas	<ol style="list-style-type: none"> 1. Get requested URL when help menu is clicked after user loggedin 2. Create instance of 'Views' class from 'urls' class 3. Invoke help() method 4. help() method should redirect user to help page with user loggedin menu options 	<p>Instance of 'Views' class should be created and help() method should be invoked.</p> <p>The user loggedin menu options should be displayed in help page</p>	Pass
WB-13	help() : Views	Gowdhaman	<ol style="list-style-type: none"> 1. Get requested URL when help menu is clicked from SkyGraph home page 2. Create instance of 'Views' class from 'urls' class 3. Invoke help() method 4. help() method should redirect user to help page with menu options 'Sign In' and 'Create account' 	<p>Instance of 'Views' class should be created and help() method should be invoked.</p> <p>The menu options should be displayed to allow user to either 'Sign In' or 'Create account'</p>	Pass
WB-14	widgets() : Views	Junior Luna	<ol style="list-style-type: none"> 1. Function have flag to check if user is loggedin or not 2. Get user name from Http session 3. Redirect user to widgets page 	<p>If user is loggedin, flag set 'true' and display loggedin menu options. The widgets page should be displayed only when flag is true</p>	Pass
WB-15	list() : Views	Hardik Prajapati	<ol style="list-style-type: none"> 1. The function checks if HttpRequest method is 'POST' 2. If POST is true, send uploaded file for validation 3. Save the file in database 4. Redirect the user 	<p>Valid uploaded file should be saved in database and its name should be displayed in file list</p>	Pass

to Import page					
WB-16	compare() : Views	Keith Thomas	<ol style="list-style-type: none"> 1. Function create instance to 'Documents' model class 2. Invoke object() method to retrieve all weather data from database 3. Redirect user to compare page 	Weather data from database should be retrieved	Pass
WB-17	compare_with_weather() : Script	Gowdhaman	<ol style="list-style-type: none"> 1. Function selects weather data file, business data file and graph report generate parameters 2. Read business and weather data 3. Invoke parse() method of 'd3' class to parse the data 4. Get parsed data, set the graph scale 5. Generate the graph 	Weather data and business data should be retrieved from database. Graph should be generated with accurate scale given	Pass
WB-18	share() : Views	Junior Luna	<ol style="list-style-type: none"> 1. Function access the database and retrieve user email address 2. Establish connection to e-mail server 3. Compose the message with weather details 4. Send message to registered e-mail address 	E-mail should be sent to registered email address with weather forecast details	Pass

WB-19	export(): Views	Keith Thomas	<ol style="list-style-type: none"> 1. Function maintains flag to check if user is authenticated 2. Access the database and retrieve requested weather details 3. Download file to user system 	Weather file should be downloaded to user local system	Pass
WB-20	account() : Views	Keith Thomas	<ol style="list-style-type: none"> 1. Function maintains flag to check if user is loggedin 2. Function establish connection to database 3. Retrieve user details from database 4. Display user account details 	User account details should be displayed when flag is set to 'true'	Pass
WB-21	editacc() : Views	Gowdhaman	<ol style="list-style-type: none"> 1. Function maintains flag to check if user is authenticated 2. Display new user account form when flag is set to true 3. Invoke edituser() method 	User flag should be 'true' and pass update user account details to edituser() method	Pass
WB-22	edituser() : Views	Junior Luna	<ol style="list-style-type: none"> 1. Function maintains entry condition to check if HttpRequest method is POST 2. Retrieve edit user form from Http session and validate 3. Update the user details in database and commit 	User account details should be displayed with update information	Pass
WB-23	logout() : Views	Hardik Prajapati	<ol style="list-style-type: none"> 1. Function maintains flag and function should be called only when flag is set to 'true' 	Function flag should be true when function is called All session values should be flushed and	Pass

2. Invoke logout() method from parent class 'auth'

3. Redirect user to logout page

logout page should be displayed

IV. Inspections

SI No.	Function : Class	Inspector	Analysis
I-1	patterns() : urls	Hardik Prajapati	The function code completely implements its functionalities. It captures requested URL from the browser and matches the URL with regular expressions mentioned in url() method. When an invalid URL is requested from browser, the function throw error and displays page not found message. Upon valid URL request, the function instantiate Views class and invoked respective methods.
I-2	MyRegistrationForm() : Myform	Junior Luna	This code executes when a user request to create new SkyGraph account. After user submitted the registration form, the form fields are retrieved from Http session and this method is invoked with form fields. From this method, User() method of auth class is invoked to with form fields in order to add custom fields like city, email to parent auth class. In addition to that, form fields value are stored in User back up table and user session is returned. The registered account is activated and it can be used to login into the application
I-3	authenticate() : auth	Gowdhaman	The piece of code implements its functionality completely. When user login into the system, it pass the login credentials from session parameters. At the back end, database connection is established and login details are validated with stored values from database. The function implements additional security by encrypting user password. After user credentials are validated, user session is created and user is redirected to home page. With invalid user credentials, system is not authenticated any login attempts
I-4	share() : Views	Keith Thomas	The function achieves its basic functionality of establishing secure connection with email server using SMTP protocol. From user session, this function retrieves user account name without causing any conflicts with user

			<p>interaction with system. By using the session parameter user's registered email address is retrieved from database and message is composed. One change that can be recommended is, instead of storing user account information in Http session, user details can be stored in temporary database tables and flag values can be used to maintain user's session.</p>
I-5	compare() : Views	Gowdhaman	<p>This piece of code is being executed when comparing weather data and business data. This function performs its basic functionality by retrieving user's business data and weather data from system repository. In order to distribute the server load to client side graphs are generated in client side instead of executing graph generator in server. One change that can be recommended is content delivery network is used for java scripts libraries, these script libraries can be stored in system instead of using online libraries. This will result in no use of compare() function when content delivery network of libraries are down.</p>