Forager : Project Analysis Report

Cycle 1
November 6, 2012

by

Group4

Matthew Powell
Robin Mays
Thomas Couture
Samuel Hall

A project report submitted for
SWE3613 Software Engineering Systems
Fall 2012

Department of Computer Science and Software Engineering
Southern Polytechnic State University
Marietta, Georgia

Table of Contents

LIST OF ABBREVIATIONS

HTML	Hypertext Markup Language
JSON	JavaScript Object Notation
OS	Operating System
SQL	Structured Query Language
UI	User Interface

1. INTRODUCTION

1.1 EXECUTIVE SUMMARY

Group4 is the producer of the website analysis tool Forager. Forager will allow a systems administrator or webmaster to easily scan their site for broken links and missing resources, then offer an easy way to generate and compare reports.

Forager is user friendly and portable. Users are provided access to reports and scanning tools through a website produced using common web standards. This means that Forager is accessible from any PC, laptop, tablet, or mobile device regardless of the client OS.

1.2 PROJECT GOALS

Forager is a web based website analysis tool. A user will login to the service and be able to start a variety of scans of their website. When the scans have completed, they will then be able to examine the results of the scans and compare them in various ways.

Forager will allow scans to be generated starting from the front page, limited to specific subdomains, or limited by time and distance from the front page. It will also allow the user to use a list of broken links from a previous scan instead of revisiting the entire website.

Forager will then allow users to sort their scans based on page load time, response time, errors, and the individual subdomains on which the page was found.

Group 4 plans to complete Forager's features in two sprints. The project is expected to be complete by November 27, 2012.

1.3 CYCLE GOALS

Group4's goal is to complete Forager in two sprint cycles as outlined below.

Sprint 1

Sprint 2

2. REQUIREMENTS

2.1 PROJECT ENVIRONMENT, TECHNOLOGY, HARDWARE, ETC.

The Forager is a web application written in a combination of PHP5 and Python. PHP is a widely-used, general-purpose scripting language that is especially suited for Web development and can be embedded into HTML, the primary markup language for displaying web pages in a web browser. Python is a common scripting language that has a rich set of features for interacting with webservers and processing HTML data. Both the Python and PHP components communicate with a PostgreSQL 8.4 database back end and the user interface is served by the Apache 2.2 webserver. This selection of mature, multi-platform technologies will allow Forager to run with little modification on most modern operating systems. It is currently being developed and tested on a virtual machine running Debian Linux 6.0 (Squeeze) on VMWare ESXi 4.1.

Users of this application are not limited to Linux or any major web browser. Any computer capable of running a web browser that supports basic HTML and SSL, which includes all of the most popular mobile and PC browsers, will be able to access the application.

2.2 USER STORIES

3. DESIGN

3.1 SYSTEM ARCHITECTURE

The Forager system consists of two parts. The webcrawler is written in Python3, using the requests library (a wrapper around the native urllib3 HTTP client). It uses PsycoPg2, a PostgreSQL connector that supports the DB-API interface defined in PEP 249. The report viewer and user interface is written in PHP 5, which is served by an Apache 2.2 Web server. Scan results are stored in a PostgreSQL 8.4 database, and all components are deployed on a machine running Linux.

A user wishing to interact with the system will go to the login page and enter their authentication credentials. They will then be presented with an option to start a new scan or view the results of existing scans. Starting a scan will check the database for any scans not marked as completed and attempt to send a Continue (SIGCONT) signal to them. If any of these signals succeed, the user is alerted that a scan is already in progress, otherwise the crawler process is spawned. On startup, the crawler will initialize its database connections and logs, and then use a simplification of Richard Stevens' daemon initialization algorithm to cleanly detach from the console. It will then create a scan record in the database, storing its process ID and the time that it was started.

The crawler uses a resource object to represent the URLs that it is given. When the crawler is initialized, it creates an object for the initial URL and stores a reference to it in a hashtable of existing resources and in a pending queue for objects that have not yet been retrieved. This object initially contains only the URL and a null link to its parent. The crawler then processes the pending queue until it is empty by retrieving the first element, using the resource's fetch() method to visit the page and store relevant information, like the HTTP response code and the load time. If the resource is an HTML page, the HTML is parsed and a list of children is stored in the resource. The resource's SQL_Call() method is then used to store the resource in the database, whose row structure matches the object definition. If any children were found in the resource, the crawler will iterate over them, and any that do not already exist in the resource list have resources created for them and are placed in the pending queue and the resource list. When the pending queue is exhausted, all resources that meet the restrictions of the crawler and that are reachable from the first page will be stored in the database, and the parent records will describe a spanning tree of the site map. When the queue is exhausted, the crawler will store its exit time in the database and shut down.

Once a scan has been registered in the database, it will be visible from the scans page on the website. The data is retrieved from the database and converted into JSON (JavaScript Object Notation). This JSON data is loaded by jQuery and processed into a sortable table with the DataTables jQuery plugin. Each of the scans can then be clicked to retrieve a list of the URLs visited in a similar manner. The list will show and allow sorting based on the response time, response code and URL.

4. MANAGEMENT PLAN

4.1 PLANNED ASSIGNMENTS AND SCHEDULE FOR FIRST CYCLE

Group 4 planned assignments for cycle 1 will breakdown as follows:

Week 1:

The Team Assignments
Gather requirements
Tune User Stories
Assignment of tasks

Individual Assignments
Matthew Powell - Requirements
Robin Mays - Requirements
Thomas Couture - Requirements
Samuel Hall – Requirements, Set-up server

Week 2:

The Team Assignments
Revise requirements
Tune User Stories
Weekly Status Report

Individual Assignments
Matthew Powell – Coding, Documentation
Robin Mays – Coding: Web UI, Documentation
Thomas Couture – Coding: Web UI, Documentation
Samuel Hall – Coding: Backend, Documentation

Week 3:

The Team Assignments
Weekly Status Report
Project Analysis Report
Project Demo

Individual Assignments
Matthew Powell – Coding, Testing, Documentation
Robin Mays – Coding: Web UI, Documentation
Thomas Couture – Coding: Web UI, Documentation
Samuel Hall – Coding: Backend, Documentation

4.2 ACTUAL ASSIGNMENTS AND SCHEDULE FOR FIRST CYCLE

Week 1:

The Team Assignments
Gather requirements
Tune User Stories
Assignment of tasks

Individual Assignments
Matthew Powell - Requirements
Robin Mays - Requirements
Thomas Couture - Requirements
Samuel Hall – Requirements, Set-up server

Week 2:

The Team Assignments
Revise requirements
Tune User Stories
Weekly Status Report

Individual Assignments
Matthew Powell – Coding, Documentation
Robin Mays – Coding: Web UI, Documentation
Thomas Couture – Coding: Web UI, Documentation
Samuel Hall – Coding: Backend, Documentation

Week 3:

The Team Assignments
Weekly Status Report
Project Analysis Report
Project Demo

Individual Assignments
Matthew Powell – Coding,Testing, Documentation
Robin Mays – Coding: Web UI, Documentation
Thomas Couture – Coding: Web UI, Documentation
Samuel Hall – Coding: Backend, Documentation

5. CYCLE POST-MORTEM ANALYSIS

5.1 SUCCESSES

Group 4 welcomed many successes during the course of Cycle 1. They were:

The creation of usable user stories and their conversion to use cases that better defined the project.

5.2 FAILURES

Time management was significantly less than optimal due to methods of communication during the use case creation. The use of email in this process was inefficient and cumbersome causing communication to be limited and unproductive. In the future of document writing Github will be used to better share documents through the internet and in-person meetings will be implemented when possible as they have proven to be the most productive use of the team's time.

5.3 LESSONS LEARNED/RISK MITIGATION

The failure to construct use cases in a timely fashion led to the late start of the actual coding of the project. However, when constructing the use cases, the cases that were redeveloped were, in most cases, revisited because they were too technical and described the algorithms to be used during the implementation. This is a situational problem and cannot be counted on to work in such a way again. The plan outlined in 5.2 should be implemented to prevent this scenario from recurring. However after this process had finished we had a better understanding of use cases. That experience will help mitigate future time wasting.

Another risk that was averted was lack of code comments in some sections of the code. There were no negative consequences to this in the current cycle, as paired programing and good communication mitigated the risk. However this might not always be the case, so in future cycles more comments in code are strongly advised.

6. TEST PLAN AND PROCEDURES

6.1 TEST PLAN

6.1.1 Introduction

The test plan for the Forager project will include verifying the results of running the web crawler, and the correctness of the information in the reports. This will be done in such a way as to detect issues with data collection done by the web crawler, and potential avenues of abuse that the web crawler can inflict. This will also cover checking for the display of misleading or inaccurate information and the usability of the web interface.

6.1.2 Test Items

The test will cover all use cases started and/or completed in this sprint, as well as potential avenues for abuse and non-intended functionality. However, the comprehensive testing is being done on a fully functional project, not in parts. This will test both the integrity of the parts, as well as their integration. Due to the nature of this project testing involving finding ways to increase speed of the scan will not be undertaken because of potential risk to the SPSU domain.

6.1.3 Software Risk Issues

As stated above there is risk that his program can inadvertently create a denial of service attack on the SPSU domain. This risk has been mitigated by a policy of not running the web crawler during normal operating hours, both on the weekdays and weekends. We also must take into account the server running this service has other functions and hosts services of its own which limits many forms of aggressive penetration testing.

6.1.4 Approach

Testing of User Interface (UI) elements will be done with the 3 most common web browsers: Google’s Chrome, Microsoft’s Internet Explorer, and the Mozilla Foundation’s Firefox. Malicious non-UI input will be tested with the TamperData extension to Firefox and direct access via telnet. Items will be considered passing if the application behaves as expected. In response to legitimate user input, the application should either take the action requested, provide the user clear instructions on how to proceed, or notify the user and the system administrator of an uncorrectable failure of the application. In response to abusive input that cannot be accomplished directly from the user interface, the application should refuse to leak information. Useful information may be provided when it does not leak information, but generic failure messages are also acceptable. The test results will come in the order that a normal user would interact with the system.

6.2 TEST RESULTS

6.2.1 Login

Login credentials can be guessed however page data does not release any data that it should not. Once logged in the user cannot log out until browser session is closed.

6.2.2 Main Page

The user is able to go to the “Start a Scan” and “View Reports” page form here. However “Compare Reports” and “Extra” give 404 errors. These sections have not been implemented, so the error is to be expected at this time. It would also appear that there are links at the bottom of the page that do not do anything. These will be most likely removed and at this time pose no risk. After viewing this page information no data was found giving the user information that they should not have.

6.2.3 Scan

When arriving to the scan page the scan starts automatically. This however cannot be stopped via the web UI (This functionality is scheduled for Cycle 2). It is noted that 2 scans cannot be run at the same time.

6.2.4 Reports Main Page

When arriving to this page the first 10 scans are displayed. The show # of entries bar works as intended and cannot be changed with the program tamper data. Tamper data, the add-on for firefox, is unable to interact with any of the fields on this page. The search bar filters results as opposed to refreshing the page and works as intended. Clicking on a scan ID # will direct the browser to a new page. The same links at the bottom of the page as seen in the main page will sent the user to the top of the page without reloading it. All other links from this page work as intended, as previously seen on the main page. It should be noted that I can sort by ID, Start Time, End Time and Run Time. These sorts appear to work as intended and it should be noted that some scans have been deleted and there numbers have not been reused and this may have to change in the future. The end time and run time for the scans do not show up, the exception is some user created data that is not from a real scan. The start time has unnecessary noise in it as well.

6.2.5 Report Page

Show # entries works as intended however there is no “next page” option available and the maximum number of entries per page is 100. A full scan has 3,467 entries. The search bar works as intended and can take both numbers (to search IDs) and strings (to search the URSs). In some cases of the sorting it has been found that long URL’s can cause the data to try and display the errors off the intended area. This effect can be found when the user sorts by “Http Response” having code 999 on the top and setting entries to 100. It should be noted that the user cannot go from this page directly back to the view reports using the links bar in the banner. I am still able to go to the main page or use the back page function to navigate. The links at the bottom return user to the top of the page.

6.2.6 Crawler verification

After viewing a report some of the URL’s were checked in the browser. All negative Http responses were reachable but required login credentials. Responses with 200 were reachable and all 404 and 999 were unreachable. This was using a random sampling form the results found.

7. CODE

***PDF ONLY**

8. CORRESPONDENCE

PDF ONLY

9. COMPLETE POWERPOINTS

1 Schema

```
/*
 * forager.sql
 * -Lee Hall Sat 20 Oct 2012 09:07:15 PM EDT
 */

/*
 *      If we need to drop a table, we can use a conditional drop like this:
 *  DROP TABLE IF EXISTS table_name;
 *  And, better yet, we can put it in the transaction, so it rolls back
 *      if things go belly up.
 */
BEGIN;

SET ROLE forager;

DROP TABLE IF EXISTS users CASCADE;
CREATE TABLE users (
        user_id         SERIAL PRIMARY KEY,
        user_name       varchar UNIQUE NOT NULL,
        password        varchar
);

COMMENT ON TABLE users IS 'Rudimentary user login table.';

INSERT INTO users(user_name,password) VALUES
        ('test', md5('test'));


DROP TABLE IF EXISTS scans CASCADE;
CREATE TABLE scans (
        scan_id         SERIAL PRIMARY KEY,
        pid                     INTEGER,
        start_time      timestamp,
        end_time        timestamp
);
COMMENT ON TABLE scans IS 'List of scans, referenced by resources';

INSERT INTO scans(scan_id,pid,start_time,end_time) VALUES
        (1, -1, '10/31/2012 4:00', '10/31/2012 4:30'),
        (2, -1, '10/30/2012 16:00', '10/31/2012 0:01');

SELECT setval('scans_scan_id_seq', max(scan_id)) FROM scans;


DROP TABLE IF EXISTS resources CASCADE;
CREATE TABLE resources (
        resource_id             SERIAL PRIMARY KEY,
        scan_id                 integer REFERENCES scans(scan_id)
                ON DELETE CASCADE,
        url                             varchar,
        parent_id               integer REFERENCES resources(resource_id)
                ON DELETE CASCADE,
        start_date              timestamp,
        response_time   interval,
        http_response   integer,
        UNIQUE (scan_id,url)
);

INSERT INTO resources(resource_id,scan_id,url,parent_id,
                start_date,response_time,http_response) VALUES
        (1, 1, 'http://minerva.gtf.org/test/', NULL,
                '10/31/2012 4:00', '.1s', 200),
        (2, 1, 'http://minerva.gtf.org/test/index.html', 1,
                '10/31/2012 4:01', '.1s', 200),
        (3, 1, 'http://minerva.gtf.org/test/bork.html', 1,
                '10/31/2012 4:01', '.1s', 404);

SELECT setval('resources_resource_id_seq', max(resource_id)) FROM resources;

COMMENT ON TABLE resources IS 'List of pages retrieved. This forms a tree'
        ' for each scan, rooted at the node with a null parent_id. This is a'
        ' spanning tree of the graph in resource_children.';

DROP TABLE IF EXISTS resource_children CASCADE;
CREATE TABLE resource_children (
        resource_id             integer REFERENCES resources(resource_id)
                ON DELETE CASCADE,
        child_id                integer REFERENCES resources(resource_id)
                ON DELETE CASCADE
);

COMMENT ON TABLE resource_children IS 'Edge set of the graph of the website.'
        ' Edges in the tree specified by parent_id also exist here.';

COMMIT;

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1
<html xmlns="http://www.w3.org/1999/xhtml">

<?php
require_once('include/secure.php');
require_once('include/conf.php');
?>

<script src="/javascript/jquery/jquery.js">
</script>
<script src="/js/jquery.dataTables.js">
</script>
<head>
           <meta http-equiv="Content-Type"
  content="text/html; charset=iso-8859-1">
   <title>Your Company</title>
   <link href="css/style.css" rel="stylesheet" type="text/css">
</head>
<body>
<div id="container">
```

```
<div id="header"> <img src="images/logo.jpg" alt="" id="logo">
<h1 id="logo-text">Reports</h1>
</div>
<div id="nav">
<ul>
  <li><a href="main">Home</a></li>
  <li><a href="scan">Start a Scan</a></li>
  <li><a href="<?php echo "Reports.php"; ?>">View Reports</a></li>
  <li><a href="compare">Compare Reports</a></li>
  <li><a href="extra">Extra</a></li>
  <li style="border-right: medium none;"><a href="#">Links</a></li>
</ul>
</div>
<div id="site-content">
<div id="demo">



<script type="text/javascript">



<?php

if(array_key_exists('scan_id',$_GET))
{
        $scan_id = $_GET['scan_id'];
}
else
{
        die;
}

$query = "SELECT resource_id, url, start_date, response_time, http_response FROM resources
$scans = pg_query_params($conn, $query,array($scan_id));

$js_array = "[";

while($results = pg_fetch_array($scans))
{
$js_array .= "[";
$js_array .= $results['resource_id'];
$js_array .= ",";
$js_array .= "\"";
$js_array .= $results['url'];
$js_array .= "\"";
$js_array .= ",";
$js_array .= "\"";
$js_array .= $results['response_time'];
$js_array .= "\"";
$js_array .= ",";
$js_array .= $results['http_response'];
$js_array .= "]";
$js_array .= ",";
}


$js_array .= "]";

?>


$(document).ready(function() {
    $('#demo').html( '<table cellpadding="0" cellspacing="0" border="0" class="display" id=
    $('#example').dataTable( {
        "aaData": <?php echo $js_array; ?> ,
        "aoColumns": [
            { "sTitle": "Resource ID" , "sClass": "center" },
            { "sTitle": "URL" , "sClass": "center" },
            { "sTitle": "Response Time", "sClass": "center" },
                        { "sTitle": "HTTP Response", "sClass": "center" },
        ]
    } );
} );


</script>









<p class="text-1">&nbsp;</p>
</div>
<div id="col-right">
<div style="padding: 30px 10px 10px;">
<h2 class="h-text-2">Latest News</h2>
<h3 class="h-text-3">Forager Version 1.0</h3>
<p class="text-2">Version 1.0 has been released. At the moment, forager is capable of searc

</div>
<div>&nbsp;</div>
<div style="padding: 5px 10px;">
<h2 class="h-text-2">Contact Info</h2>
</div>
<div
 style="padding: 5px 10px 15px; background: rgb(216, 214, 215) none repeat scroll 0%; -moz-
<p class="text-2"> 00/00 Lorem Ipsum is simply dummy text of the
printing and typesetting.<br>
<br>
E.mail: abc@Lorem Ipsum<br>
<br>
Fax: 000.000.0000<br>
<br>
Phone: 000.000.0000/<br>
```

```
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
000.000.0000 </p>
</div>
</div>
</div>
<div id="footer">
<p>@ Copyright 2010. Designed by <a target="_blank"
 href="http://www.htmltemplates.net/">HTML Templates</a></p>
<ul class="footer-nav">
  <li><a href="#">Home</a></li>
  <li><a href="#">About us</a></li>
  <li><a href="#">Recent articles</a></li>
  <li><a href="#">Email</a></li>
  <li><a href="#">Resources</a></li>
  <li><a href="#">Links</a></li>
</ul>
</div>
</div>
</body>
</html>

<?php
/*
 * include/secure.php
 * -Lee Hall Tue 09 Oct 2012 01:23:17 AM EDT
 * This should be included in the header of any page that sends a password
 */
if (!array_key_exists('HTTPS', $_SERVER) || $_SERVER['HTTPS'] != "on") {
    $url = "https://". $_SERVER['SERVER_NAME'] . $_SERVER['REQUEST_URI'];
    header("Location: $url");
    die("Forwarding to a secure page");
}

?>

<?php
/*
 * session.php
 * -Lee Hall Thu 06 Sep 2012 10:13:49 PM EDT
 *
 * Check that a session exists.
 * If not, bounce them to the login page and die.
 */

session_start();
if (! array_key_exists('user_id', $_SESSION) ||
        !isset($_SESSION['user_id'])){
    header( "location: login.php");
    die("User not logged in");
}

?>

<?php
```

```
/*
 * config.php
 * -Lee Hall Thu 06 Sep 2012 10:10:03 PM EDT
 *
 * This file opens the database connection and provides some useful global
 * variables to the project
 */

$conn_str="user=apache dbname=forager";
$conn= pg_connect($conn_str);
if (!$conn)
    die("Unable to connect to database.");

function db_cleanup($conn){
    pg_close($conn);
}
register_shutdown_function('db_cleanup', $conn);

$URL_BASE="https://$_SERVER[SERVER_NAME]";

// Set the default timezone for calls to date().
// Everythign generated by PHP should go in the DB, and it can be queried in
// the proper TZ at that point.
date_default_timezone_set('UTC');
?>

<?php
/*
 * index.php
 * -Lee Hall Tue 23 Oct 2012 11:43:10 AM EDT
 */
require_once("include/conf.php");
require_once("include/session.php");
header("Location: main.php");
die("Forwarding to index");
?>

<?php
/*
 * start.php
 * -Lee Hall Sat 03 Nov 2012 06:50:57 PM EDT
 */
require_once('include/conf.php');
require_once('include/session.php');

$procs_sql="SELECT pid FROM scans WHERE end_time IS NULL;";
$procs_res=pg_query($procs_sql);
while($procs_row=pg_fetch_assoc($procs_res)){
        /* Send SIGCONT to every crawler process that doesn't have an end-time in
         * the db. False means there's no process with that id running. This can
         * fail weirdly with recycled pids, but as long as we can accurately
     * record when a process dies, there shouldn't be any user facing issues.
         */
        trigger_error("Checking if process $procs_row[pid] is still running.");
        if(posix_kill($procs_row['pid'], 18)){
                trigger_error("Scan already running with pid $procs_row[pid].");
                die("Scan with pid $procs_row[pid] is still running.");
        }
}
trigger_error("Starting scanning process.");
exec("/usr/local/src/forager/bin/crawler.py");
die("Started webcrawler.");
header("Location: main.php");
?>

<?php
require_once("include/conf.php");
require_once("include/session.php");
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type"
 content="text/html; charset=iso-8859-1">
  <title>Your Company</title>
  <link href="css/style.css" rel="stylesheet" type="text/css">
</head>
<body>
<div id="container">
<div id="header"> <img src="images/logo.jpg" alt="" id="logo">
<h1 id="logo-text">Forager</h1>
</div>
<div id="nav">
<ul>
  <li><a href="main">Home</a></li>
  <li><a href="start.php">Start a Scan</a></li>
  <li><a href="scans.php">View Reports</a></li>
  <li><a href="compare">Compare Reports</a></li>
  <li><a href="extra">Extra</a></li>
  <li style="border-right: medium none;"><a href="#">Links</a></li>
</ul>
</div>
<div id="site-content">
<div id="col-left">
<h1 class="h-text-1">WELCOME</h1>
<p class="text-1"><strong>Group 4 is an entity that strives to give its customer the best
software agent technology that is available. Our product is called Forager and it provides
<ul class="list-1">
  <li>Scan any web site</li>
  <li>Generate reports</li>
  <li>Sort reports</li>
  <li>Print reports</li>
  <li>Run timed scans</li>
</ul>
<p class="text-1">Forager is a web crawler that scan, sorts and generates the reports that
<p class="border-1">&nbsp;</p>
<h2 class="h-text-2">About us</h2>
<p class="text-1">Group 4 is made up of professionals with over 20 years of joint experienc
```

```
<p class="text-1">&nbsp;</p>
</div>
<div id="col-right">
<div style="padding: 30px 10px 10px;">
<h2 class="h-text-2">Latest News</h2>
<h3 class="h-text-3">Forager Version 1.0</h3>
<p class="text-2">Version 1.0 has been released. At the moment, forager is capable of searc

</div>
<div>&nbsp;</div>
<div style="padding: 5px 10px;">
<h2 class="h-text-2">Contact Info</h2>
</div>
<div
 style="padding: 5px 10px 15px; background: rgb(216, 214, 215) none repeat scroll 0%; -moz-
<p class="text-2"> Southern Polytechnic State University.<br>
<br>
E.mail: Spsu@Spsu.edu<br>
<br>
Fax: 678-915-7778<br>
<br>
Phone: 678-915-7778/<br>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
000.000.0000 </p>
</div>
</div>
</div>
<div id="footer">
<p>@ Copyright 2010. Designed by <a target="_blank"
 href="http://www.htmltemplates.net/">HTML Templates</a></p>
<ul class="footer-nav">
  <li><a href="#">Home</a></li>
  <li><a href="#">About us</a></li>
  <li><a href="#">Recent articles</a></li>
  <li><a href="#">Email</a></li>
  <li><a href="#">Resources</a></li>
  <li><a href="#">Links</a></li>
</ul>
</div>
</div>
</body>
</html>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1
<html xmlns="http://www.w3.org/1999/xhtml">

<?php
require_once('include/secure.php');
require_once('include/conf.php');
?>

<script src="/javascript/jquery/jquery.js">
</script>
<script src="/js/jquery.dataTables.js">
```

```
</script>
<head>
          <meta http-equiv="Content-Type"
 content="text/html; charset=iso-8859-1">
  <title>Your Company</title>
  <link href="css/style.css" rel="stylesheet" type="text/css">
</head>
<body>
<div id="container">
<div id="header"> <img src="images/logo.jpg" alt="" id="logo">
<h1 id="logo-text">Reports</h1>
</div>
<div id="nav">
<ul>
  <li><a href="main">Home</a></li>
  <li><a href="scan">Start a Scan</a></li>
  <li><a href="<?php echo "scans.php"; ?>">View Reports</a></li>
  <li><a href="compare">Compare Reports</a></li>
  <li><a href="extra">Extra</a></li>
  <li style="border-right: medium none;"><a href="#">Links</a></li>
</ul>
</div>
<div id="site-content">
<div id="demo">



<script type="text/javascript">



<?php

$query = "SELECT scan_id, start_time, end_time, end_time - start_time as elapsed_time FROM
$scans = pg_query($conn, $query);

$js_array = "[";

while($results = pg_fetch_array($scans))
{
$js_array .= "[";
$js_array .= "\"<a href='Reports.php?scan_id=$results[scan_id]'> $results[scan_id]</a>\"";
$js_array .= ",";
$js_array .= "\"";
$js_array .= $results['start_time'];
$js_array .= "\"";
$js_array .= ",";
$js_array .= "\"";
$js_array .= $results['end_time'];
$js_array .= "\"";
$js_array .= ",";
$js_array .= "\"";
$js_array .= $results['elapsed_time'];
$js_array .= "\"";
$js_array .= "]";
$js_array .= ",";
}


$js_array .= "]";

?>







$(document).ready(function() {
    $('#demo').html( '<table cellpadding="0" cellspacing="0" border="0" class="display" id=
    $('#example').dataTable( {
        "aaData": <?php echo $js_array; ?> ,
        "aoColumns": [
            { "sTitle": "Scan ID" , "sClass": "center" },
            { "sTitle": "Start Time" , "sClass": "center" },
            { "sTitle": "End TIme" , "sClass": "center" },
            { "sTitle": "Run Time", "sClass": "center" },
        ]
    } );
} );


</script>









<p class="text-1">&nbsp;</p>
</div>
<div id="col-right">
<div style="padding: 30px 10px 10px;">
<h2 class="h-text-2">Latest News</h2>
<h3 class="h-text-3">Forager Version 1.0</h3>
<p class="text-2">Version 1.0 has been released. At the moment, forager is capable of searc

</div>
<div>&nbsp;</div>
<div style="padding: 5px 10px;">
<h2 class="h-text-2">Contact Info</h2>
</div>
<div
 style="padding: 5px 10px 15px; background: rgb(216, 214, 215) none repeat scroll 0%; -moz-
<p class="text-2"> Southern Polytechnic State University.<br>
<br>
```

```
E.mail: Spsu@Spsu.edu<br>
<br>
Fax: 678-915-7778<br>
<br>
Phone: 678-915-7778/<br>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
000.000.0000 </p>
</div>
</div>
</div>
<div id="footer">
<p>@ Copyright 2010. Designed by <a target="_blank"
 href="http://www.htmltemplates.net/">HTML Templates</a></p>
<ul class="footer-nav">
  <li><a href="#">Home</a></li>
  <li><a href="#">About us</a></li>
  <li><a href="#">Recent articles</a></li>
  <li><a href="#">Email</a></li>
  <li><a href="#">Resources</a></li>
  <li><a href="#">Links</a></li>
</ul>
</div>
</div>
</body>
</html>

<?php
/*
 * login.php
 * -Lee Hall Thu 06 Sep 2012 10:23:45 PM EDT
 * edits by Matthew Powell
 * Allow the user to login
 */
require_once('include/secure.php');
require_once('include/conf.php');

//Is there a user trying to log in?
if (array_key_exists('login', $_POST)){

    if (!array_key_exists('user_name', $_POST) ||
            !array_key_exists('password', $_POST) ){
        die("User or password not set. How did you get here?");
    }
        $UserName=strtolower($_POST['user_name']);
    // Get user info from database. Only retrieve users who have authenticated
    // their accounts.
    // If this gets slow, we can pull the quota after getting user_id so we
    // don't have to scan the whole files table, but this works for now
    $sql="SELECT user_id,password
            FROM users
            WHERE user_name=$1;";
    $params=array($UserName);
    $results=pg_query_params($conn, $sql, $params);
    if (!$results || pg_num_rows($results) > 1){
        $msg="Unrecoverable database error.";
        trigger_error($msg);
        die($msg);
    }

    //Bail and reload the page if we didn't find a user
    $row=pg_fetch_array($results);
    if (! $row){
        header("Location: $_SERVER[PHP_SELF]?msg=Unknown User");
        die("User not found.");
    }

    //Does the password match?
    if (md5($_POST['password']) == $row['password']){
        session_start();
        $_SESSION['user_name']=$UserName;
        $_SESSION['user_id']=$row['user_id'];

                header("Location: main.php");

                die("Done loading user.");
    } else {
        header("Location: $_SERVER[PHP_SELF]?msg=Bad Password");
        // This leaks information about whether or not a user exists on the
        // system. The ease of use is a net positive, however.
        // This problem can be alleviated with rate limiting on the login.
        die("Bad password.");
    }
}
if (array_key_exists('logout', $_GET)){

    // Make sure the session's started so we have access to the variables we
    // want to clear
    session_start();
    $_SESSION=array();
    session_destroy();

    header("Location: $_SERVER[PHP_SELF]");
    die("Reloading login page.");
}
?>
<HTML>
<HEAD>
  <TITLE>Forager Login</TITLE>
</HEAD>
<BODY>
<table cellspacing="1" cellpadding="0" border="0"
    id="shell" height="471" width="1168">
   <tr height="50">
      <td height="83" colspan="2" bgcolor="white">
         <table title="Banner" id="banner" border="0">
            <tr><td width="1195"></a></td></tr>
         </table>
      <img src="images/Honeycomb Logo 2.jpg"
```

```
          width="1221" height="137" alt="Honeycomb Logo 2">       </td>
  </tr>
   <tr height="200">
      <td width="260" bgcolor="white">
         <table id="navigation" title="Navigation" border="0">

         <tr><td>
 <table border="0" cellspacing="0" cellpadding="0">
      </td>
   </tr>
    <tr height="200">
     <td width="260" bgcolor="white">


       <table id="navigation" title="Navigation" border="0">


        </table>



      </td><td width="397" bgcolor="white">

         <table title="Content" id="content" border="0">
            <tr><td>

            </td></tr>
         </table>
      </td>
   </tr>
</table>

            <tr><td><form action="<?php echo $_SERVER['PHP_SELF']; ?>"
            method="post" id="login">
        <table>
            <tr>
                <td>User Name:</td>
                <td><input name="user_name" type="text"></td>
            </tr>
            <tr>
                <td>Password:</td>
                <td><input name="password" type="password"></td>
            </tr>
            <tr>
                <td><input name="login" type="hidden"</td>
                <td><input value="Login" type="submit"></td>
            </tr>
            <tr>
                <td></td>
                <td>
<?php
    if (array_key_exists('msg', $_GET)){
        echo "$_GET[msg]";
    }
```

```
?>
                </td>
            </tr>
        </table>
    </form></td></tr>
         </table>
      <img src="images/bigbox.jpg" width="432" height="432">      </td>
      </td>
   </tr>
</table>
</table>
</BODY>
</HTML>

#! /usr/bin/env python3
# resource.py
# -Lee Hall Sat 27 Oct 2012 12:21:55 PM EDT

import requests
import logging
import time
from bs4 import BeautifulSoup

DEBUG=False

class resource:
    """Represent a URL/resource."""

    @staticmethod
    def get_domain(url):
        method_end=url.find('://') + 3
        domain_end=url.find('/', method_end)
        return url[method_end:domain_end]

    @staticmethod
    def get_method(url):
        return url[:url.find('://') + 3]

    def __init__(self, url, scan_id):
        # Directory names MUST end in a trailing space in the URL
        # URLs should start with 'http://'
        self.url=url
        self.domain=resource.get_domain(url)
        self.method=resource.get_method(url)
        self.scan_id=scan_id
        self.visited=False
        self.parent=None
        self.children=[]
        self.response_code=-1
        self.resource_id=None
        self.time_started=-1
        self.time_elapsed=-1
        self.time_start=-1
        if (DEBUG):
            logging.basicConfig(level=logging.DEBUG)

    def __str__(self):
        representation = "<resource( url: {0}, domain: {1}".format(
            self.url,self.domain)
        if (self.visited):
            representation+=", response: {0}, children[".format(
                self.response_code)
            for child in self.children:
                representation += " {0} ".format(child)
            representation+="]"
        representation+=")>"
        return representation

    def __repr__(self):
        return "<resource: {0}>".format(self.url)

    def __eq__(self, other):
        if (type(other) is str):
            return  self.url==other
        return self.url == other.url

    def fetch(self):
        start=time.time()
        try:
            # Don't verify SSL connections
            r=requests.get(self.url, verify=False)
        except requests.Timeout:
            logging.info("Timed out fetching page {0}".format(self.url))
            self.visited=True
            self.response_code=-1
            #page timed out not 404
            return
        except requests.RequestException as e:
            logging.info("Unknown exception {0}".format(e))
            self.visited=True
            self.response_code=-3
            #dead or unreachable page not 404
            return
        finally:
            elapsed=time.time()-start

        self.time_started=start
        self.time_elapsed=elapsed

        if(r is None):
            logging.warn("Request failed for {0}".format(e))
            #dead or unreachable page not 404 (should not happen)
            return
            #3 above should not happen on day to day bassis
        self.visited=True
        self.response_code=r.status_code
        self.response_time=r.headers
        # Only try to parse html content
        if (r.headers.get('content-type').startswith('text/html')):
            self.parse_children(r)

    def parse_children(self, request):
        if (not self.visited):
            assert self.visited==True, "Cannot parse an unvisited page"

        try:
            parsed=BeautifulSoup(request.text)
        except Exception as e:
            logging.warn("Exception {0} while parsing {1}".format(
                e, self.url))
            return False

        for link in parsed.find_all(['a', 'link']):
            attr=link.get('href')
            if (attr is None):
                continue
            self.children.append(self.canonicalize(attr))
        for link in parsed.find_all(['script', 'img']):
            attr=link.get('src')
            if (attr is None):
                continue
            if (attr[1:5] == "data:"):
                logging.info("Ignored inline image data on {0}".format(
                    self.url))
                continue
            self.children.append(self.canonicalize(attr))

    def canonicalize(self, url):
        # Absolute URL
        if (url.startswith('http://') or url.startswith('https://')):
            return url
        elif (url.startswith('/')):
            can_link=self.method + self.domain + url
        else:
            can_link=self.url[:self.url.rfind('/') + 1] + url

        logging.debug("Canonicalized link {0}".format(can_link))
        return can_link

    def Sql_Call(self, connection):
        self.cur=connection
        if (self.parent is None):
            parent_id=None
        else:
            parent_id=self.parent.resource_id
        insert_sql="""
            INSERT INTO resources(scan_id,url,
                parent_id,response_time,http_response)
            VALUES (%s,%s,%s,%s,%s)
            RETURNING resource_id"""
        self.cur.execute(insert_sql, (self.scan_id,self.url,
            parent_id,"'{0} seconds'".format(self.time_elapsed),
            self.response_code))
        result=self.cur.fetchone()
        self.resource_id=result[0]


# This only to be run when testing hte module independently
def main():
    r=resource("http://minerva.gtf.org/test/")
    r.fetch()

#! /usr/bin/env python3
# crawler.py
# -Lee Hall Sat 27 Oct 2012 02:13:07 PM EDT

from resource import resource
from collections import deque
import psycopg2
import logging
import signal
import sys
import os

DEBUG=True
CONN_STRING="dbname=forager user=apache"
DOMAIN="spsu.edu"
START_PAGE="http://spsu.edu/"
LOGFILE="/var/log/forager.log"
# DOMAIN="gtf.org"
# START_PAGE="http://minerva.gtf.org/test/"

class crawler:

    def __init__(self):
        if (DEBUG):
            logging.basicConfig(level=logging.DEBUG, filename=LOGFILE)
            logging.debug("Debugging enabled.")
        else:
            logging.basicConfig(level=logging.INFO)
        signal.signal(signal.SIGINT, self.sig_handler)
        signal.signal(signal.SIGTERM, self.sig_handler)

        self.daemonize()
        self.dbinit()

    def dbclose(self):
        set_term_sql="UPDATE scans SET end_time=NOW() WHERE scan_id=%s";
        if(hasattr(self, 'cur') and self.cur is not None):
            self.cur.execute(set_term_sql,(self.scan_id,))
            self.cur.close()
            self.DB_Connection.close()
        else:
            logging.warn("Crawler exited before connecting to the database.")
    def sig_handler(self,sig, frame):
        if (sig == signal.SIGINT):
            logging.warn("Caught SIGINT. Exiting.")
            self.dbclose()
            sys.exit(0)
        elif (sig == signal.SIGTERM):
            logging.warn("Caught SIGTERM. Exiting.")
            self.dbclose()
            sys.exit(0)

    def dbinit(self):

        try:
            self.DB_Connection = psycopg2.connect(CONN_STRING)
        except psycopg2.Error as e:
            msg="Target Database configuration error: \"{0}{1}\".".format(
                type(e),e)
            logging.critical(msg)
            exit(1)

        self.cur=self.DB_Connection.cursor()

        #Autocommit database queries. We don't need transactions.
        self.DB_Connection.set_session(autocommit=True)

    # Daemonize crawler process. This is adapted from Stevens's Advanced
    # Programming in a Unix Environment, and ported to python3 by an anonymous
    # user. Source is available here: http://www.jejik.com/files/examples/daemon3x.py
    # Stevens's original code starts on page 426 in the second edition, (c) 1995.
    def daemonize(self):
        #FOrk
        try:
            pid= os.fork()
            if (pid > 0):
                sys.exit(0)
        except OSError as e:
            logging.warn("Fork failed: {0}.".format(e))
            sys.exit(1)

        logging.info("Forked as.".format(pid))
        # Reset env
        os.chdir('/')
        os.setsid()
        os.umask(0)

        #Fork again.
        try:
            pid= os.fork()
            if (pid > 0):
                sys.exit(0)
        except OSError as e:
            logging.warn("Fork failed: {0}.".format(e))
            sys.exit(1)
        logging.info("Forked again.".format(pid))

        sys.stdout.flush()
        sys.stderr.flush()


        # Open devnull and move input/output over there.
        si=open(os.devnull, 'r')
        so=open(os.devnull, 'a+')
        se=open(os.devnull, 'a+')

        os.dup2(si.fileno(), sys.stdin.fileno())
        os.dup2(so.fileno(), sys.stdout.fileno())
        os.dup2(se.fileno(), sys.stderr.fileno())

        logging.info("Detatched from terminal.".format(pid))



    def crawl(self,url):
        logging.info("Starting crawl at {0}.".format(url))
        pid=os.getpid()
        create_scan_sql="""INSERT INTO scans(start_time,pid)
            VALUES (NOW(), %s) RETURNING scan_id;"""
        self.cur.execute(create_scan_sql, (pid,))
        scan_row=self.cur.fetchone()
        self.scan_id=scan_row[0]
        resource_list={}
        pending=deque()
        pending.append(url)
        resource_list[url]=resource(url,self.scan_id)

        while (len(pending) > 0):
            logging.debug(pending)
            cur_url=pending.popleft()

            assert cur_url in resource_list, "{0} ".format(cur_url) + \
                 "was placed in the pending queue, but no resource was created"
            cur_resource=resource_list[cur_url]

            assert not resource_list[cur_url].visited, \
                "Already visited resource {0} was requeued".format(cur_url)

            logging.info("Processing \"{0}\"".format(cur_url))

            cur_resource.fetch()
            #makes all data on creation
            cur_resource.Sql_Call(self.cur)

            for child_url in cur_resource.children:
                if (child_url in resource_list):
                    logging.debug(
                        "Skipping existing URL \"{0}\"".format(child_url))
                    continue
                logging.debug("Queueing \"{0}\"".format(child_url))
                new_resource=resource(child_url,self.scan_id)
                new_resource.parent=cur_resource
                if (not new_resource.domain.endswith(DOMAIN)):
                    logging.debug(
                        "Skipping URL \"{0}\", outside of {1}".format(
                            child_url, DOMAIN))
                    continue
                pending.append(child_url)
                resource_list[child_url]=new_resource

try:
    c=crawler()
    c.crawl(START_PAGE)
    c.dbclose()
except Exception as e:
    logging.critical("Something exploded: {0}".format(e))
```

SWE 3613 Status Report

IMPORTANT: File naming instructions

Name this file in the following manner: YYYYMMDD_TEAM_NAME_HERE.pdf

Example: 20120911_Group1.pdf

Project Name	Forager - Group 4
Team Members	Robin Mays, Thomas Couture, Matthew Powell, Lee Hall
Week Ending:	10/23/2012
Cycle	Cycle 1
System Metaphor	The system is designed to check through the entire SPSU domain and return a detailed report. This report should include all errors including dead links, missing images, scripts, and css files. All reports created should be stored and accessable through an easy to use user interface. These reports should also be sortable to make it easier to find certain reports, as well as comparable to each other to check and see the differences between them. It was also include features such as pausing, stoping, or restricting a search.
Cycle Intent	The intent of this cycle is to get the web crawler working. That includes getting the web crawler to not just search through the different pages, but to also return the errors that it encounters. We also plan on taking those results and putting them into a user friendly report to be examined through our user interface. We plan on having the web crawler functional through our user interface. We also plan on completing the login for security purposes.

#	User Story	Planned			Actual		
		Cycle planned for completion	Total planned hours	Planned hours this cycle	Status	Actual hours this cycle	Total hours
1	As a user, I would like to be able to visit and access all pages of my website.	1	20	20	Unstarted	0	
2	As a user, I would like this program to record any resources that are unavailable, including dead links, missing images, scripts or css files.	1	20	20	Design	3	3
3	As a user I must be able to view a report of a given scan. This report should show all broken links and missing images that fall under my domain. These scans should be	1	30	30	Design	3	3
4	As a user, I would like to view scans and start new scans from a website.	1	20	20	Unstarted	0	
5	As a user viewing a report, I should be able to generate that report in a printer friendly format.	2	10	0	Unstarted	0	
6	As a user viewing a report, I should be able to generate that report in a printer friendly format.	2	10	0	Unstarted	0	
7	As a user, I would like to be able to select two scans and show only the items that have changed.	2	6	0	Unstarted	0	
8	As a user viewing a report, I should be able to view reports from scans that are in progress.	2	4	0	Unstarted	0	
9	As a user, I would like to be able to limit the run time of a scan when I start it, either by time, or by distance from the start page.	2	12	0	Unstarted	0	
10	As a user I would like to select a scan, and run a new scan that will check if the previous errors have been corrected.	2	10	0	Unstarted	0	
11	As a user, I would like for reports to include pages that are accessible over secured links.	2	8	0	Unstarted	0	
12	As a user, I would like to sort a report based on the subdomain.	2	14	0	Unstarted	0	
13	As a user, I should have to login before initiating a scan or viewing a report.	1	2	2	Completed	1	1
14	As a user, I might like to pause a scan that was currently in progress.	2	4	0	Unstarted	0	
15	As a user, having Scans that were automatically run at regular intervals.	2	6	0	Unstarted	0	
16	As a user I would like to see page load times in my reports.	2	2	0	Unstarted	0	
		Planned Total	178	92	**Actual Total**	7	7

SWE 3613 Status Report

Date	10/23/2012
Members	Robin Mays, Thomas Couture, Matthew Powell, Lee Hall
Project	Forager - Group 4

	HOURS BY DEVELOPMENT ACTIVITY																	
	Requirements			Design / Prototype			Development / Code			Integrate / Test			Documentation			Totals		
Name	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total
Robin Mays		3	3				25							0.5	0.5	25	3.5	3.5
Thomas Couture		2	2		2	2	25							2	2	25	6	6
Matthew Powell		3	3		2	2	20							0.5	0.5	20	5.5	5.5
Lee Hall		3	3		2	2	20							3	3	20	8	8
Totals:	0	11	11	0	6	6	90	0	0	0	0	0	0	6	6	90	23	23

	HOURS BY USER STORY																	
	1		2		3		4		5		6		7		8		9	
Name	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle
Robin Mays																		
Thomas Couture					2	2												
Matthew Powell			2	2														
Lee Hall			1	1	1	1												
Totals:	0	0	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0
	10		11		12		13		14		15		16		0		**Totals**	
	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle
Robin Mays																	0	0
Thomas Couture																	2	2
Matthew Powell																	2	2
Lee Hall							1	1									3	3
Totals:	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	7	7

Accomplishments since last status report:

Completed the User Stories. Divided up the work load for the project. Database and backend set up. Github up and ready.

Obstacles encountered since last status report:

Risks facing the project:

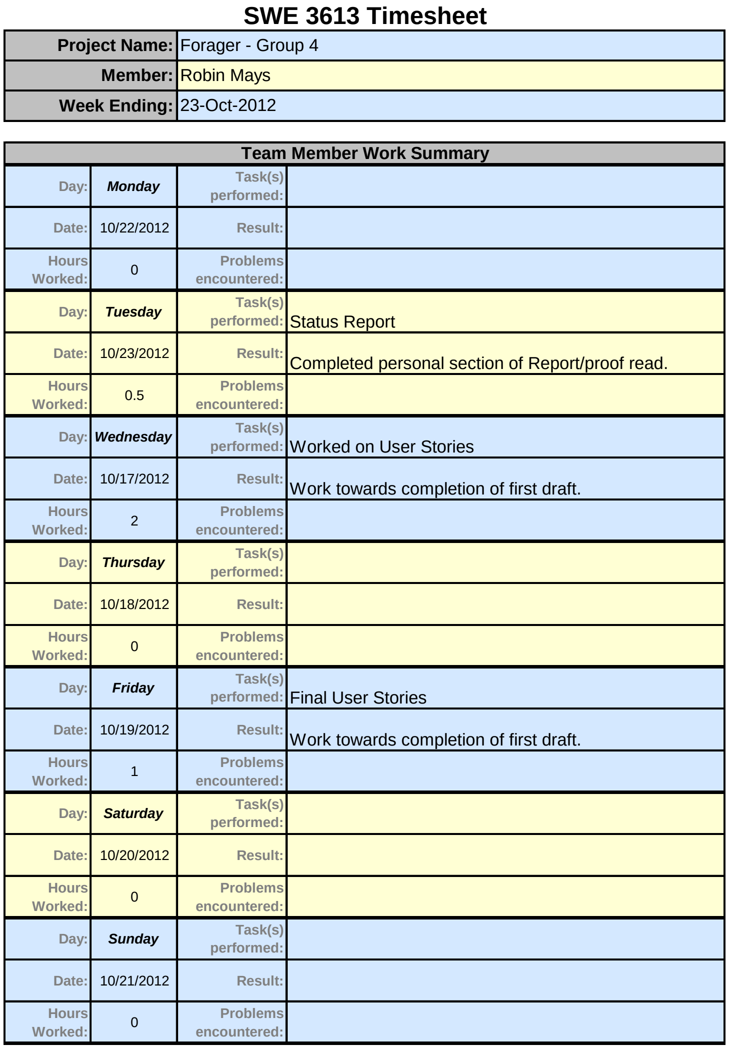
We could have library problems dealing with SSL pages. This can be mitigated by research, and is not a huge issue as it's a relatively low priority story. We could have communications problems between the report viewer and the web crawler.This would be a huge issue. A backup plan might be to use python to generate JSON directly from pickled object from the webcrawler, or potentially generate oneshot reports directly in the webcrawler, those these options are both less flexible.

Objectives for the next week:

Have the webcrawler complete and ready to be integrated into the UI. Have the UI complete and ready for the integration. Login page completed. Begin work on report viewing.

Notes:

Database and Github was reused from last project. Work to shift over from last project to this was minimal (10 minutes) and thus not added in to the work.



SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Robin Mays
Week Ending:	23-Oct-2012

Team Member Work Summary

Day:	***Monday***	Task(s) performed:	
Date:	10/22/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Status Report
Date:	10/23/2012	Result:	Completed personal section of Report/proof read.
Hours Worked:	0.5	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	Worked on User Stories
Date:	10/17/2012	Result:	Work towards completion of first draft.
Hours Worked:	2	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	
Date:	10/18/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Friday***	Task(s) performed:	Final User Stories
Date:	10/19/2012	Result:	Work towards completion of first draft.
Hours Worked:	1	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	
Date:	10/20/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	
Date:	10/21/2012	Result:	
Hours Worked:	0	Problems encountered:	

SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Thomas Couture
Week Ending:	23-Oct-2012

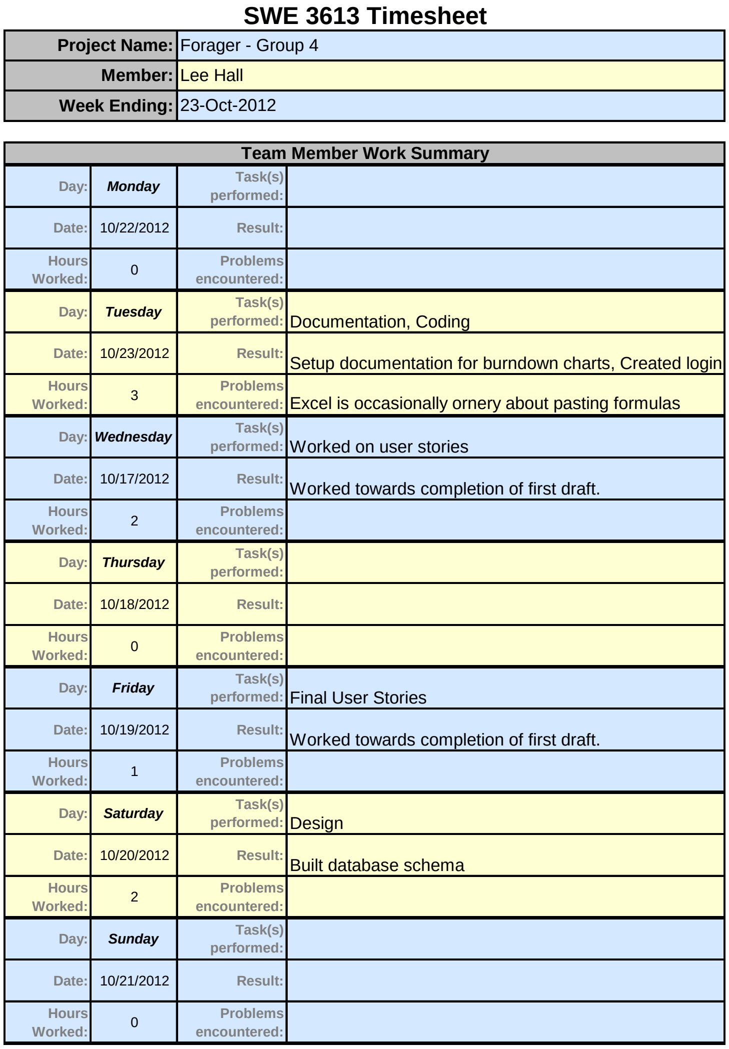
Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	
Date:	10/22/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Weekly Status Report
Date:	10/23/2012	Result:	Completed Weekly Status report
Hours Worked:	2	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	Worked on User Stories
Date:	10/17/2012	Result:	Work towards completion of first draft.
Hours Worked:	1	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	
Date:	10/18/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Friday***	Task(s) performed:	Final Copy of User Stories
Date:	10/19/2012	Result:	Work towards completion of first draft.
Hours Worked:	1	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	
Date:	10/20/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	Swaign work for UI
Date:	10/21/2012	Result:	Made progress on design of user story 3 and overall UI
Hours Worked:	2	Problems encountered:	slight language learning curve

SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Matthew Powell
Week Ending:	23-Oct-2012

Team Member Work Summary

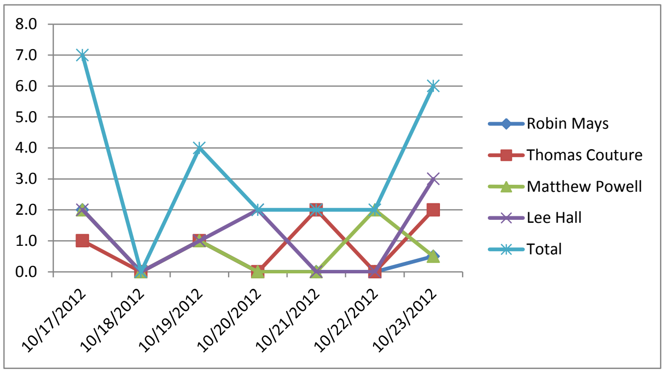
Day:	***Monday***	Task(s) performed:	Design for User Story 2
Date:	10/22/2012	Result:	
Hours Worked:	2	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Weekly Status Report
Date:	10/23/2012	Result:	Completed personal section of report/proof read.
Hours Worked:	0.5	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	First Draft of User Stories
Date:	10/17/2012	Result:	Completed first draft
Hours Worked:	2	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	
Date:	10/18/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Friday***	Task(s) performed:	Final draft of user stories
Date:	10/19/2012	Result:	work towards the completion of the final draft.
Hours Worked:	1	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	
Date:	10/20/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	
Date:	10/21/2012	Result:	
Hours Worked:	0	Problems encountered:	

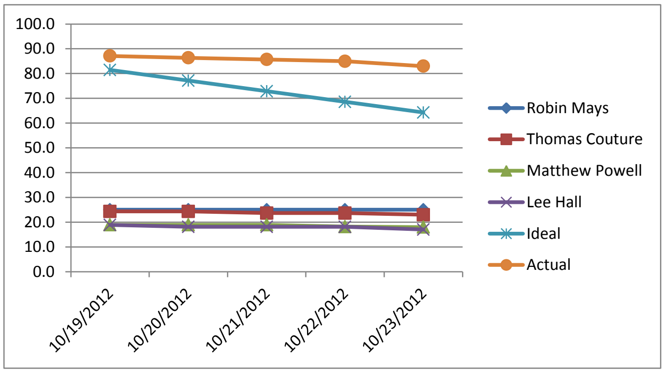


SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Lee Hall
Week Ending:	23-Oct-2012

Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	
Date:	10/22/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Documentation, Coding
Date:	10/23/2012	Result:	Setup documentation for burndown charts, Created login
Hours Worked:	3	Problems encountered:	Excel is occasionally ornery about pasting formulas
Day:	***Wednesday***	Task(s) performed:	Worked on user stories
Date:	10/17/2012	Result:	Worked towards completion of first draft.
Hours Worked:	2	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	
Date:	10/18/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Friday***	Task(s) performed:	Final User Stories
Date:	10/19/2012	Result:	Worked towards completion of first draft.
Hours Worked:	1	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	Design
Date:	10/20/2012	Result:	Built database schema
Hours Worked:	2	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	
Date:	10/21/2012	Result:	
Hours Worked:	0	Problems encountered:	


8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
10/17/2012
10/18/2012
10/19/2012
10/20/2012
10/21/2012
10/22/2012
10/23/2012
Robin Mays
Thomas Couture
Matthew Powell
Lee Hall
Total


100.0
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
10/19/2012
10/20/2012
10/21/2012
10/22/2012
10/23/2012
Robin Mays
Thomas Couture
Matthew Powell
Lee Hall
Ideal
Actual

SWE 3613 Status Report

IMPORTANT: File naming instructions

Name this file in the following manner: YYYYMMDD_TEAM_NAME_HERE.pdf

Example: 20120911_Group1.pdf

Project Name	Forager - Group 4
Team Members	Robin Mays, Thomas Couture, Matthew Powell, Lee Hall
Week Ending:	10/30/2012
Cycle	Cycle 1
System Metaphor	The system is designed to check through the entire SPSU domain and return a detailed report. This report should include all errors including dead links, missing images, scripts, and css files. All reports created should be stored and accessable through an easy to use user interface. These reports should also be sortable to make it easier to find certain reports, as well as comparable to each other to check and see the differences between them. It was also include features such as pausing, stoping, or restricting a search.
Cycle Intent	The intent of this cycle is to get the web crawler working. That includes getting the web crawler to not just search through the different pages, but to also return the errors that it encounters. We also plan on taking those results and putting them into a user friendly report to be examined through our user interface. We plan on having the web crawler functional through our user interface. We also plan on completing the login for security purposes.

ID	**Use Case Name**	**Planned**			**Actual**		
		Cycle planned for completion	Total planned hours	Planned hours this cycle	Status	Actual hours this cycle	Total hours
Crawler 1	Basic Web User	1	20	20	Development	8	8
Crawler 2	Record Crawler Results	1	20	20	Design	5	5
Report 1	Show Scan Results	1	30	30	Design	10	13
Report 2	Crawler Interaction	1	20	20	Unstarted	0	
Report 3	Print Report	2	10	0	Unstarted	0	
Report 4	Sort Report	2	10	0	Unstarted	0	
Report 5	Report Changes	2	6	0	Unstarted	0	
Report 6	Live Report	2	4	0	Unstarted	0	
UC 9	Runtime Limit	2	12	0	Unstarted	0	
UC 10	Error Check	2	10	0	Unstarted	0	
UC 11	Secure Check	2	8	0	Unstarted	0	
UC 12	Subdomain Sort	2	14	0	Unstarted	0	
13	Login	1	2	2	Completed	1	1
14	Pause Scan	2	4	0	Unstarted	0	
15	Timer	2	6	0	Unstarted	0	
16	Stopwatch	2	2	0	Unstarted	0	
		Planned Total	178	92	**Actual Total**	24	27

SWE 3613 Status Report

Date	10/30/2012
Members	Robin Mays, Thomas Couture, Matthew Powell, Lee Hall
Project	Forager - Group 4

	HOURS BY DEVELOPMENT ACTIVITY																	
	Requirements			**Design / Prototype**			**Development /**			**Integrate / Test**			**Documentation**			**Totals**		
Name	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total
Robin Mays	4	2	5	5	2	2	17	2	2	3	0		4	0	0.5	33	6	9.5
Thomas Couture	4	2	4	5	2	4	17	4	4	3	0		4	1	3	33	9	15
Matthew Powell	4	2	5	4	2	4	13	0		3	0		4	0	0.5	28	4	9.5
Lee Hall	4	2	5	2	0	2	13	8	8	5	2	2	4	1	4	28	13	21
Totals:	16	8	19	16	6	12	60	14	14	14	2	2	16	2	8	122	32	55

	HOURS BY USER STORY																	
	Crawler 1		Crawler 2		Report 1		Report 2		Report 3		Report 4		Report 5		Report 6		UC 9	
Name	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle
Robin Mays					4	4												
Thomas Couture					6	8												
Matthew Powell			2	4														
Lee Hall	8	8		1		1												
Totals:	8	8	2	5	10	13	0	0	0	0	0	0	0	0	0	0	0	0

	UC 10		UC 11		UC 12		13		14		15		16		0		**Totals**	
	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle
Robin Mays																	8	4
Thomas Couture																	6	8
Matthew Powell																	2	4
Lee Hall							1	1									9	11
Totals:	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	25	27

Accomplishments since last status report:

Completed first draft of Use Cases. Completed web crawler. Major work completed on UI (Enough to integrate working components). Design progress on Crawler 2.

Obstacles encountered since last status report:

Risks facing the project:

We could have library problems dealing with SSL pages. This can be mitigated by research, and is not a huge issue as it's a relatively low priority story. We could have communications problems between the report viewer and the web crawler.This would be a huge issue. A backup plan might be to use python to generate JSON directly from pickled object from the webcrawler, or potentially generate oneshot reports directly in the webcrawler, those these options are both less flexible.

Objectives for the next week:

Integrate webcrawler with the website. Finish creating the reports. Finish up and test Use Cases 1-4.

Notes:

SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Robin Mays
Week Ending:	30-Oct-2012

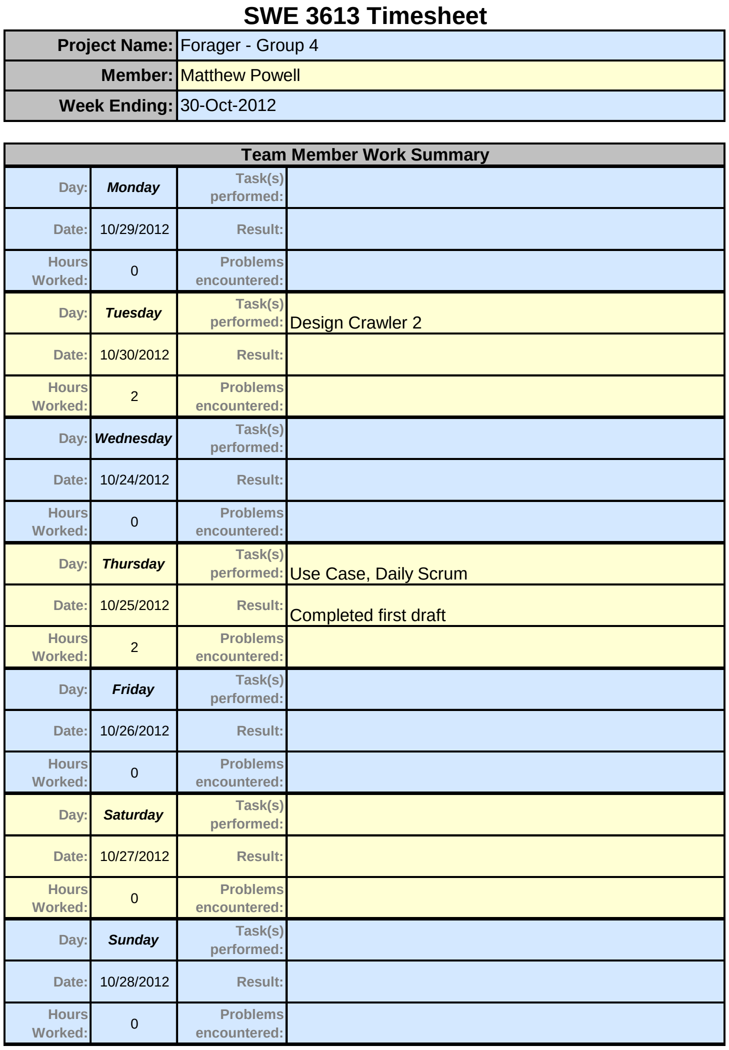
Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	
Date:	10/29/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	
Date:	10/30/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	10/24/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	Scrum Meeting, Use Cases
Date:	10/25/2012	Result:	Completed first draft of Use Cases
Hours Worked:	2	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	10/26/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	Design for UI
Date:	10/27/2012	Result:	Plans for main UI Flow
Hours Worked:	2	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	UI coding
Date:	10/28/2012	Result:	Progress on Main Page
Hours Worked:	2	Problems encountered:	

SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Thomas Couture
Week Ending:	30-Oct-2012

Team Member Work Summary

Day:	***Monday***	Task(s) performed:	
Date:	10/29/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	UI/Coding for Report 1/ Status report
Date:	10/30/2012	Result:	More UI progress, work towards report displaying
Hours Worked:	4	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	10/24/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	Scrum Meeting, Use Cases
Date:	10/25/2012	Result:	Completed first draft of Use Cases
Hours Worked:	2	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	10/26/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	
Date:	10/27/2012	Result:	
Hours Worked:		Problems encountered:	
Day:	***Sunday***	Task(s) performed:	UI Coding
Date:	10/28/2012	Result:	Progress towards main page, Reports page
Hours Worked:	3	Problems encountered:	



SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Matthew Powell
Week Ending:	30-Oct-2012

Team Member Work Summary

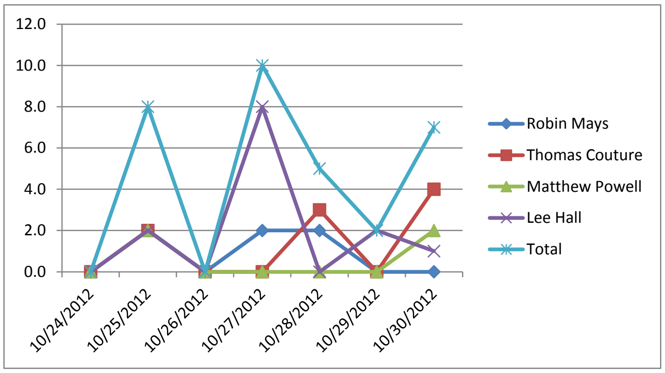
Day:	***Monday***	Task(s) performed:	
Date:	10/29/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Design Crawler 2
Date:	10/30/2012	Result:	
Hours Worked:	2	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	10/24/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	Use Case, Daily Scrum
Date:	10/25/2012	Result:	Completed first draft
Hours Worked:	2	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	10/26/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	
Date:	10/27/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	
Date:	10/28/2012	Result:	
Hours Worked:	0	Problems encountered:	

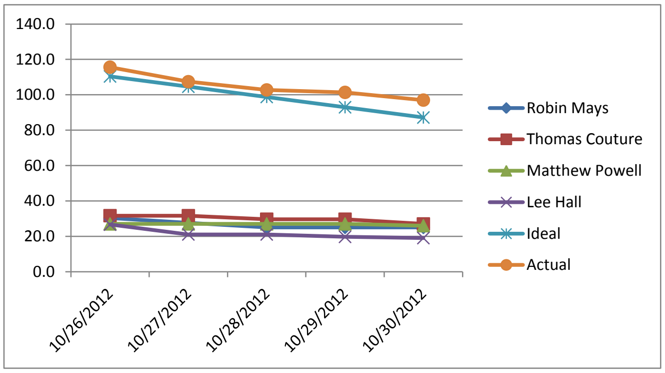
SWE 3613 Timesheet

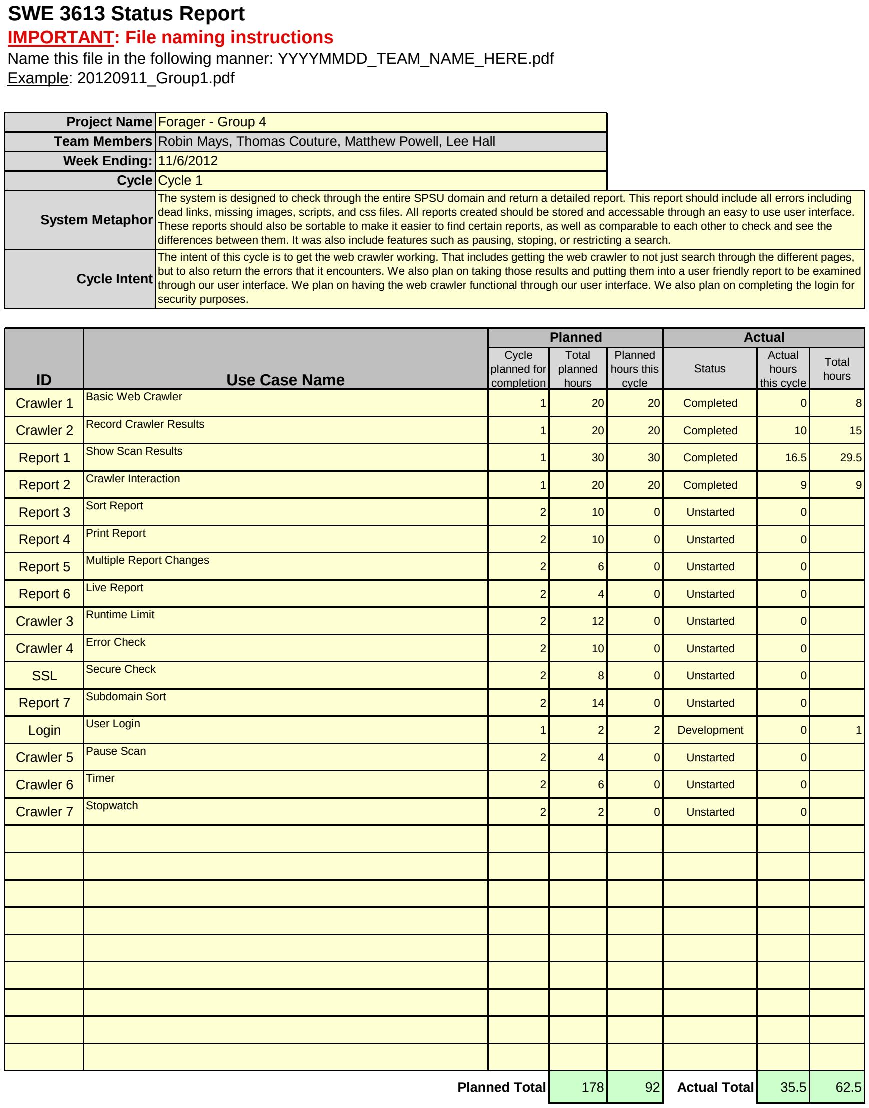
Project Name:	Forager - Group 4
Member:	Lee Hall
Week Ending:	30-Oct-2012

Team Member Work Summary

Day:	***Monday***	Task(s) performed:	Testing
Date:	10/29/2012	Result:	Removed some bugs in the web crawler
Hours Worked:	2	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Documentation
Date:	10/30/2012	Result:	Weekly Status reports and meeting
Hours Worked:	1	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	10/24/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	Documentation
Date:	10/25/2012	Result:	Daily Scrum, Use cases
Hours Worked:	2	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	10/26/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	Coding
Date:	10/27/2012	Result:	Built Web Crawler, Use Case 1
Hours Worked:	8	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	
Date:	10/28/2012	Result:	
Hours Worked:	0	Problems encountered:	


12.0
10.0
8.0
6.0
4.0
2.0
0.0
10/24/2012
10/25/2012
10/26/2012
10/27/2012
10/28/2012
10/29/2012
10/30/2012
Robin Mays
Thomas Couture
Matthew Powell
Lee Hall
Total


140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
10/26/2012
10/27/2012
10/28/2012
10/29/2012
10/30/2012
Robin Mays
Thomas Couture
Matthew Powell
Lee Hall
Ideal
Actual



SWE 3613 Status Report

IMPORTANT: File naming instructions

Name this file in the following manner: YYYYMMDD_TEAM_NAME_HERE.pdf

Example: 20120911_Group1.pdf

Project Name	Forager - Group 4
Team Members	Robin Mays, Thomas Couture, Matthew Powell, Lee Hall
Week Ending:	11/6/2012
Cycle	Cycle 1
System Metaphor	The system is designed to check through the entire SPSU domain and return a detailed report. This report should include all errors including dead links, missing images, scripts, and css files. All reports created should be stored and accessable through an easy to use user interface. These reports should also be sortable to make it easier to find certain reports, as well as comparable to each other to check and see the differences between them. It was also include features such as pausing, stoping, or restricting a search.
Cycle Intent	The intent of this cycle is to get the web crawler working. That includes getting the web crawler to not just search through the different pages, but to also return the errors that it encounters. We also plan on taking those results and putting them into a user friendly report to be examined through our user interface. We plan on having the web crawler functional through our user interface. We also plan on completing the login for security purposes.

		Planned			**Actual**		
ID	**Use Case Name**	Cycle planned for completion	Total planned hours	Planned hours this cycle	Status	Actual hours this cycle	Total hours
Crawler 1	Basic Web Crawler	1	20	20	Completed	0	8
Crawler 2	Record Crawler Results	1	20	20	Completed	10	15
Report 1	Show Scan Results	1	30	30	Completed	16.5	29.5
Report 2	Crawler Interaction	1	20	20	Completed	9	9
Report 3	Sort Report	2	10	0	Unstarted	0	
Report 4	Print Report	2	10	0	Unstarted	0	
Report 5	Multiple Report Changes	2	6	0	Unstarted	0	
Report 6	Live Report	2	4	0	Unstarted	0	
Crawler 3	Runtime Limit	2	12	0	Unstarted	0	
Crawler 4	Error Check	2	10	0	Unstarted	0	
SSL	Secure Check	2	8	0	Unstarted	0	
Report 7	Subdomain Sort	2	14	0	Unstarted	0	
Login	User Login	1	2	2	Development	0	1
Crawler 5	Pause Scan	2	4	0	Unstarted	0	
Crawler 6	Timer	2	6	0	Unstarted	0	
Crawler 7	Stopwatch	2	2	0	Unstarted	0	
		Planned Total	178	92	**Actual Total**	35.5	62.5

SWE 3613 Status Report

Date	11/6/2012
Members	Robin Mays, Thomas Couture, Matthew Powell, Lee Hall
Project	Forager - Group 4

	HOURS BY DEVELOPMENT ACTIVITY																	
	Requirements			**Design / Prototype**			**Development /**			**Integrate / Test**			**Documentation**			**Totals**		
Name	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total	Cycle Plan	Week Actual	Cycle Total
Robin Mays	4	0	5	5	0	2	17	7	9	3	1	1	4	4	4.5	33	12	21.5
Thomas Couture	4	0	4	5	0	4	17	9.5	12.5	3	1	1	4	2	5	33	12.5	26.5
Matthew Powell	4	0	5	4	0	4	13	5	5	3	2	2	4	6	6.5	28	13	22.5
Lee Hall	4	0	5	2	0	2	13	8	16	5	2	4	4	0	4	28	10	31
Totals:	16	0	19	16	0	12	60	29.5	42.5	14	6	8	16	12	20	122	47.5	101.5

	HOURS BY USER STORY																	
	Crawler 1		Crawler 2		Report 1		Report 2		Report 3		Report 4		Report 5		Report 6		Crawler 3	
Name	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle
Robin Mays					7	11	1	1										
Thomas Couture					8.5	16.5	2	2										
Matthew Powell			5	9	1	1	1	1										
Lee Hall		8	5	6		1	5	5										
Totals:	0	8	10	15	16.5	29.5	9	9	0	0	0	0	0	0	0	0	0	0

	Crawler 4		SSL		Report 7		Login		Crawler 5		Crawler 6		Crawler 7		0		**Totals**	
	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle	This Week	This Cycle
Robin Mays																	8	12
Thomas Couture																	10.5	18.5
Matthew Powell																	7	11
Lee Hall								1									10	21
Totals:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	35.5	62.5

Accomplishments since last status report:

Updates to the UI. Robin and Thomas completed Use Case Report 1. Matthew and Lee completed Use Case Crawler 2. Lee finished up Use Case Report 2.

Obstacles encountered since last status report:

Risks facing the project:

We could have library problems dealing with SSL pages. This can be mitigated by research, and is not a huge issue as it's a relatively low priority story. We could have communications problems between the report viewer and the web crawler.This would be a huge issue. A backup plan might be to use python to generate JSON directly from pickled object from the webcrawler, or potentially generate oneshot reports directly in the webcrawler, those these options are both less flexible.

Objectives for the next week:

Spring Planning meeting and starting to make progress on Sprint 2.

Notes:

SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Robin Mays
Week Ending:	6-Nov-2012

Team Member Work Summary

Day:	***Monday***	Task(s) performed:	Documentation
Date:	11/5/2012	Result:	Sprint 1 report/Power Point
Hours Worked:	2	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	
Date:	11/6/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	10/31/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	UI Coding
Date:	11/1/2012	Result:	Work completed on Reports/ Scans
Hours Worked:	3	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	11/2/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	Coding for Report 1
Date:	11/3/2012	Result:	Results are displayed on Website
Hours Worked:	5	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	Documentation
Date:	11/4/2012	Result:	Sprint 1 report
Hours Worked:	2	Problems encountered:	

SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Thomas Couture
Week Ending:	6-Nov-2012

Team Member Work Summary

Day:	***Monday***	Task(s) performed:	UI updates/fixes and Final Doc
Date:	11/5/2012	Result:	UI touchup work
Hours Worked:	3	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Final Report/Power Point
Date:	11/6/2012	Result:	Last touch up work and submission
Hours Worked:	1	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	10/31/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	UI Coding
Date:	11/1/2012	Result:	Work completed on Report and Scans page
Hours Worked:	2	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	11/2/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	Coding for Report 1
Date:	11/3/2012	Result:	Results are displayed on Website
Hours Worked:	5	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	Status Report
Date:	11/4/2012	Result:	Final status report and final changes
Hours Worked:	1.5	Problems encountered:	

SWE 3613 Timesheet

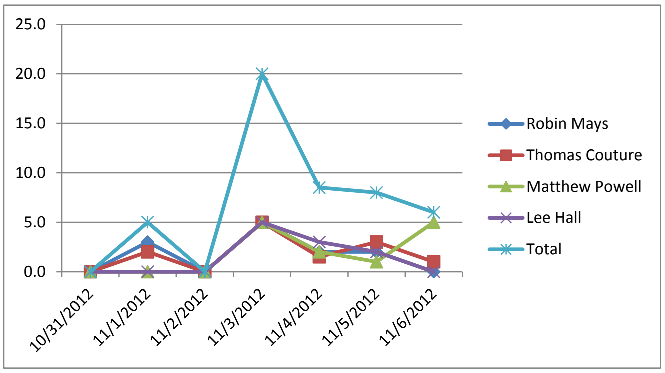
Project Name:	Forager - Group 4
Member:	Matthew Powell
Week Ending:	6-Nov-2012

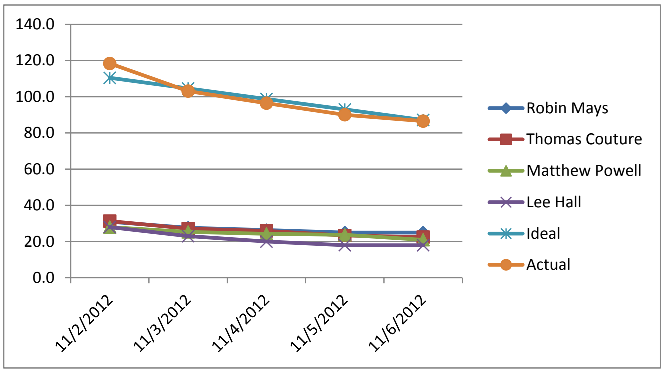
Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	Sprint Report
Date:	11/5/2012	Result:	Post Mortem Analysis
Hours Worked:	1	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Documentation
Date:	11/6/2012	Result:	Post Mortem, Test Plan, Screenshots, Power Point
Hours Worked:	5	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	10/31/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	
Date:	11/1/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	11/2/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	Coding on Crawler 2
Date:	11/3/2012	Result:	Completed Crawler 2 Use Case
Hours Worked:	5	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	Test Plan
Date:	11/4/2012	Result:	Testing Use Case Crawler 2, Report 1, Report 2
Hours Worked:	2	Problems encountered:	

SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Lee Hall
Week Ending:	6-Nov-2012

Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	Testing of Report 2
Date:	11/5/2012	Result:	Worked on Signal Handling between Crawler and Web
Hours Worked:	2	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	
Date:	11/6/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	10/31/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	
Date:	11/1/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	11/2/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	Coding Crawler 2
Date:	11/3/2012	Result:	Completed Crawler 2 use case
Hours Worked:	5	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	Worked on Report 2
Date:	11/4/2012	Result:	Integration of Web Crawler with Web UI
Hours Worked:	3	Problems encountered:	


25.0
20.0
15.0
10.0
5.0
0.0
10/31/2012
11/1/2012
11/2/2012
11/3/2012
11/4/2012
11/5/2012
11/6/2012
Robin Mays
Thomas Couture
Matthew Powell
Lee Hall
Total


140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
11/2/2012
11/3/2012
11/4/2012
11/5/2012
11/6/2012
Robin Mays
Thomas Couture
Matthew Powell
Lee Hall
Ideal
Actual

Group 4

Forager Meeting Minutes

October 16, 2012

Opening:
The first meeting of ***Group4*** was called to order at 6:00 pm on October 16, 2012 at SPSU by The Team.

Present:
Matthew Powell
Robin Mays
Thomas Couture
Samuel Hall

Present:

A. **Approval of Agenda**

The agenda was unanimously approved as discussed.

B. **Approval of Minutes**

The minutes of the previous meeting are available for viewing.

C. **Opening Issues**

Prepare for post-mortem and sprint retrospective for the Honeycomb project.

D. **New Business**
Review user stories/requirements for Forager's two sprints. Discuss Risk Mitigation for the upcoming sprints.

E. **Agenda for Next Meeting**

Design architecture for Forager and review and update risk mitigation plan.

Adjournment:
This meeting was adjourned at 7:30pm by The Team. The next general meeting will be at 6:00pm on October 23, 2012 at SPSU.

Minutes submitted by: Robin Mays

Approved by: Matthew Powell

Group 4

Forager Meeting Minutes : Sprint Planning

October 18, 2012

Opening:
The first meeting of ***Group4*** was called to order at 6:00 pm on October 18, 2012 at SPSU by The Team.

Present:
Matthew Powell
Robin Mays
Thomas Couture
Samuel Hall

Present:

A. **Approval of Agenda**

The agenda was unanimously approved as discussed.

B. **Approval of Minutes**

The minutes of the previous meeting are available for viewing.

C. **Opening Issues**

Complete risk mitigation and sprint planning.

D. **New Business**
Review user stories/requirements for Forager's two sprints. Discuss and update risk mitigation for the upcoming sprints.

User Stories/Requirements:
ID: 1
As a user, I would like to be able to visit and access all pages of my website.
Priority: 10, Cost: 20

ID: 2
As a user, I would like this program to record any resources that are unavailable, including dead links,
missing images, scripts or css files.
Priority: 10, Cost: 20

ID: 3
As a user I must be able to view a report of a given scan. This report should show all broken links and
missing images that fall under my domain. These scans should be listed and accessible from a website.

Priority: 10, Cost: 30

ID: 4
As a user I would like to view scans and start new scans from a website.
Priority:10, Cost 20

ID: 5
As a user, I would like to be able to sort the reports that are presented to me. Useful sorting functions
would be: by the order in which the pages were visited, alphabetically, and alphabetically by the parent
page.
Priority: 7, Cost: 10

ID: 6
As a user viewing a report, I should be able to generate that report in a printer friendly format.
Priority: 6, Cost: 10

ID:7
As a user, I would like to be able to select two scans and show only the items that have changed.
Priority: 6, Cost: 6

ID: 8
As a user viewing a report, I should be able to view reports from scans that are in progress.
Priority: 6, Cost: 4

ID:9
As a user, I would like to be able to limit the run time of a scan when
I start it, either by time, or by distance from the start page.
Priority: 6, Cost: 12

ID: 10
As a user I would like to select a scan, and run a new scan that will check if the previous errors have
been corrected.
Priority: 5, Cost: 10

ID: 11
As a user, I would like for reports to include pages that are accessible
over secured links.
Priority: 5, Cost: 8

ID:12
As a user, I would like to sort a report based on the subdomain.

Priority: 3, Cost: 14

ID: 13
As a user, I should have to login before initiating a scan or viewing a report.
Priority: 3, Cost: 2

ID:14
As a user, I might like to pause a scan that was currently in progress.
Priority: 2, Cost: 4

ID: 15
As a user, having Scans that were automatically run at regular intervals.
Priority: 1, Cost: 6

ID:16
As a user I would like to see page load times in my reports.
Priority:1 Cost: 2

Risk mitigation plan:

We could have library problems dealing with SSL pages.
This can be mitigated by research, and is not a huge issue as it's a relatively low priority story.

We could have communications problems between the report viewer and the web crawler.
This would be a huge issue. A backup plan might be to use python to generate JSON directly from pickled object from the web-crawler, or potentially generate one shot reports directly in the web-crawler, those these options are both less flexible.

Dealing with javascript could make me so angry I stick forks in my eyes.
This might make typing difficult, and I should probably keep a bottle of medication uncorked and nearby to prevent such an occurrence.

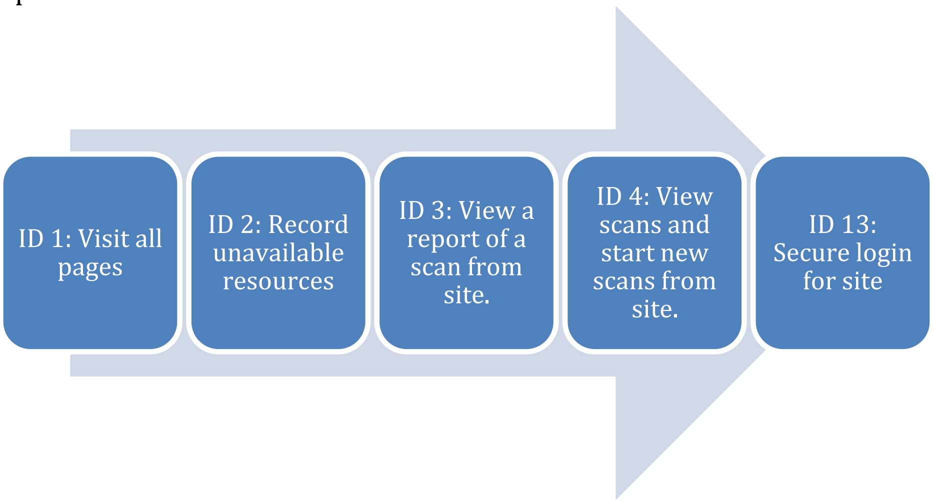
System Architecture:

We will be writing the web-crawler in python utilizing libcurl and communicating with a postgres datastore using PsychoPG2.

The front end will be in PHP, but to provide user interaction (Sorting lists in sane ways, etc), we're going to have to use javascript, which means using jQuery.

Sprint Planning : Cycle Plans

Sprint 1:



ID 1: Visit all pages

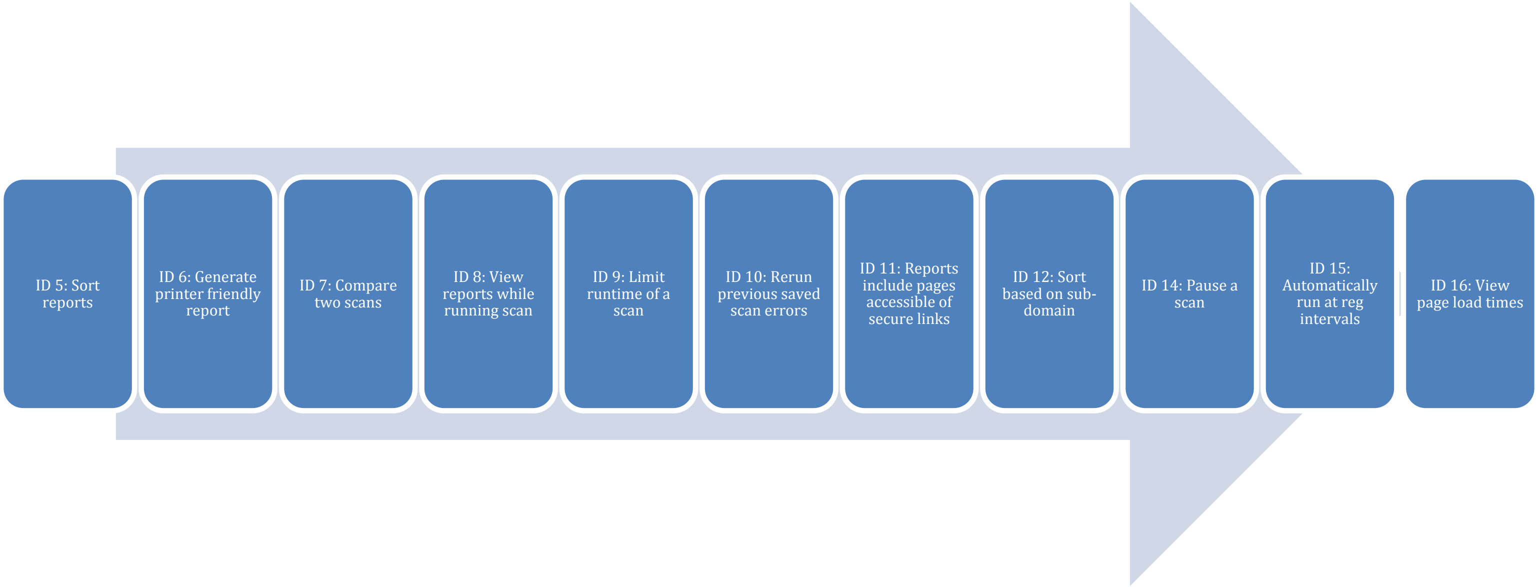
ID 2: Record unavailable resources

ID 3: View a report of a scan from site.

ID 4: View scans and start new scans from site.

ID 13: Secure login for site

Sprint 2:



ID 5: Sort reports

ID 6: Generate printer friendly report

ID 7: Compare two scans

ID 8: View reports while running scan

ID 9: Limit runtime of a scan

ID 10: Rerun previous saved scan errors

ID 11: Reports include pages accessible of secure links

ID 12: Sort based on sub-domain

ID 14: Pause a scan

ID 15: Automatically run at reg intervals

ID 16: View page load times

E. Agenda for Next Meeting

Design architecture for Forager and review and update risk mitigation plan.

Adjournment:
This meeting was adjourned at 7:30pm by The Team. The next general meeting will be at 6:00pm on October 25, 2012 at SPSU.

Minutes submitted by: Robin Mays

Approved by: Matthew Powell

```
commit 7bb62998ca4fbc2eccf9e1b5c2428487608df1cc
Author: Lee Hall <lhall@newfields.com>
Date:   Tue Nov 6 12:25:18 2012 -0500

    Updated Risk_Mitigation

commit 2a449ad6f1e0fb4465152ced58e901c665722df2
Author: Lee Hall <lhall@newfields.com>
Date:   Mon Nov 5 16:22:54 2012 -0500

    Documentation updates

commit c2583868eeee39743b9393f3fe314db7f14a8e7b
Author: Lee Hall <lhall@newfields.com>
Date:   Mon Nov 5 15:38:29 2012 -0500

    Add timing

commit 120a91715cb3ba3dbcbbff928b2e0c398a793e4e
Author: Lee Hall <shall4@spsu.edu>
Date:   Mon Nov 5 16:06:59 2012 -0500

    Store timing data

commit 8f13d7bae1d417c44d7833b075d12192ecf46b8c
Author: Lee Hall <lhall@newfields.com>
Date:   Mon Nov 5 03:31:01 2012 -0500

    Use cases

commit 9b8b4a99777c48e71366c2e37ca4ba238828996e
Author: Lee Hall <shall4@spsu.edu>
Date:   Mon Nov 5 03:02:08 2012 -0500

    Daemonize crawler.

    Crawler now backgrounds properly.

commit 88e21809fee654327684935a8ebe171f887fb4d8
Author: Lee Hall <shall4@spsu.edu>
Date:   Mon Nov 5 02:02:02 2012 -0500

    Readded pid saving

    This got lost in a merge. Fixed.

commit 3030a4d534799adbee2bfc1574897f76b0cfb8c8
Merge: 9ffc5d0 1be39df
Author: Lee Hall <shall4@spsu.edu>
Date:   Mon Nov 5 01:49:33 2012 -0500

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit 1be39dfa3ec78d8a3568dfa5080259d514a11317
```

```
Merge: bcc7c48 6f90920
Author: Lee Hall <lhall@newfields.com>
Date:   Mon Nov 5 01:48:59 2012 -0500

    Merge branch 'master' of github.com:lhall23/forager-t4

    Conflicts:
        bin/crawler.py
        html/start.php

commit 9ffc5d0ba7be5d2925260e78594e093a5f129bba
Author: Lee Hall <shall4@spsu.edu>
Date:   Mon Nov 5 01:48:58 2012 -0500

    Fixups

commit bcc7c48c7d14cba8ba63840fc07d741e559c2b2a
Author: Lee Hall <lhall@newfields.com>
Date:   Mon Nov 5 01:44:39 2012 -0500

    Classed crawler so it can be daemonized.

commit beeaa5734ac2bed4c266876765b75201dca284c3
Author: Lee Hall <lhall@newfields.com>
Date:   Mon Nov 5 01:44:16 2012 -0500

    Detatch from crawler.

commit 6f9092074cc69f923a24931d010f5cb63247b2c8
Merge: 36e8987 ffad08f
Author: Lee Hall <shall4@spsu.edu>
Date:   Sun Nov 4 19:41:59 2012 -0500

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit 36e89876ef54aafea99107e244b4ee7397d9c185
Author: Lee Hall <shall4@spsu.edu>
Date:   Sun Nov 4 19:30:47 2012 -0500

    Check current status

    Check to see if a crawler is already running before starting

commit a5a871b3b43d076935c554c21871e1fbd09a975f
Author: Lee Hall <shall4@spsu.edu>
Date:   Sun Nov 4 19:30:18 2012 -0500

    Track PID for signaling.

commit 9da1f7b59ee9683a67881108b91820c2629b9147
Author: Lee Hall <shall4@spsu.edu>
Date:   Sun Nov 4 19:28:15 2012 -0500

    Added catch for inlined image data.
```

```
commit ffad08f701fc079acaebbd9ff5c4f6eee99823d9
Author: M <swordthane@gmail.com>
Date:   Sun Nov 4 15:14:39 2012 -0500

    Change of contact info main page

commit e62d23baec7711a699cf3abb483ceece59bac971
Author: M <swordthane@gmail.com>
Date:   Sun Nov 4 14:53:04 2012 -0500

    Reports page Contact info

commit 65a6f617580134de2520838f7befda08b7a4ef37
Merge: 731b78b 92d40cb
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Nov 3 18:57:34 2012 -0400

    Merge branch 'master' of github.com:lhall23/forager-t4

commit 731b78b21a9763609ff053c0f6cbe8af996df761
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Nov 3 18:57:10 2012 -0400

    Added exec

commit 92d40cb1cbe23fc6ec3b654e5cb7b41922537415
Merge: 908103e fc8413d
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 18:50:56 2012 -0400

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit 908103ea0c4079725ea46efa334edada67267419
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 18:50:53 2012 -0400

    report

commit fc8413d28b0b40f51f1850e9df8b1fcbc1ffe141
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Nov 3 18:47:30 2012 -0400

    Need another import

commit fef73812dd89a5d132950d596e5dfebfbb1021dc
Merge: 8849fdd 12614d1
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 18:46:13 2012 -0400

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit 8849fdd120f42f38f848be5eac03b5c768fe7415
Author: tcouture127 <tcouture127@gmail.com>
```

```
Date:   Sat Nov 3 18:46:00 2012 -0400

    RAGE!!!!

commit 12614d1a7431af82b8dd3daa54ee19a4df0707e4
Merge: 602c726 8c0c43f
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Nov 3 18:43:47 2012 -0400

    Merge branch 'master' of github.com:lhall23/forager-t4

commit 602c726406be8b7aba83bf905894e9a97ac267e2
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Nov 3 18:43:31 2012 -0400

    Typo

commit 8c0c43f22fc184998218e9770ab01ea28ff6950f
Merge: f8ce8ef e9843c0
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 18:43:01 2012 -0400

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit f8ce8efb85c4a7b2af1e71ed68f57d0d7132e651
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 18:42:28 2012 -0400

    2 =(

commit e9843c06c33f9feea65bb0ccacc662069d8b7494
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Nov 3 18:40:02 2012 -0400

    Fixed Typo

commit 7e18ffd8f3a92578b925db8be0b0b9fb7bd92434
Merge: cdcf0a1 7dc75c2
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 18:39:24 2012 -0400

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit cdcf0a1404ba4c132224c453505204eca9b55b22
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 18:38:58 2012 -0400

    first try??

commit 7dc75c2c579251ca903548563fc91a81578ddb4c
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Nov 3 18:38:26 2012 -0400

    Added signal handling
```

```
commit 2db0e42e231b0ebb0292b7632563b3b27df1ecb7
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 18:24:38 2012 -0400

    yay

commit a3b5d5ca77f3d525a58fb5d8e3efc927f2c2ec07
Author: Lee Hall <shall4@spsu.edu>
Date:   Sat Nov 3 18:24:15 2012 -0400

    Added DataTables library

commit 3f65c44076ef8bbb7a829618d754bc62e066ac7e
Merge: a3f7ba5 2db0e42
Author: Lee Hall <shall4@spsu.edu>
Date:   Sat Nov 3 18:23:23 2012 -0400

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit ff55cd5671c0687cf25f97d4d1b2a896bcb8d131
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 18:21:26 2012 -0400

    update

commit a3f7ba5e3a543c6c21eda790c0fa194f01a56ea3
Merge: 039febf ff55cd5
Author: Thomas Couture <tcoutur2@minerva.gtf.org>
Date:   Sat Nov 3 18:20:02 2012 -0400

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit 31eae1554e251107c1c332e4ae3dabc48bb4cccc
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 17:56:24 2012 -0400

    final ???

commit 039febfb61c07c2319a5b756f882e8de9d2788ef
Merge: 944bcc0 31eae15
Author: Thomas Couture <tcoutur2@minerva.gtf.org>
Date:   Sat Nov 3 17:54:32 2012 -0400

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit d2e1f7867a1ac977b7b8cd3a14b6d657dded0ecf
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 17:53:33 2012 -0400

    show me maybe??

commit 944bcc0ea935b39f9e2dfbe95125c97849911dd2
Merge: b738ae6 d2e1f78
```

```
Author: Thomas Couture <tcoutur2@minerva.gtf.org>
Date:   Sat Nov 3 17:52:28 2012 -0400

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit d2396396789bfbaf7d81ec10ee6505c16676e2e8
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 17:41:49 2012 -0400

    Preview Scans

commit b738ae6c576b7db8e24a63e4fdcc3d8dcec07b91
Merge: 09bebd3 d239639
Author: Thomas Couture <tcoutur2@minerva.gtf.org>
Date:   Sat Nov 3 17:40:07 2012 -0400

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit 4d9d18dcd49a592a615fcd3818d5a6694b2ea5ce
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 17:28:27 2012 -0400

    asda

commit 40c434de6d26c205a9f88468e50a773b06cb58a4
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 17:28:03 2012 -0400

    adsa

commit 09bebd368a222ec6b06a269e72ffae7cba6f0db2
Merge: 3ee44ee 4d9d18d
Author: Thomas Couture <tcoutur2@minerva.gtf.org>
Date:   Sat Nov 3 17:26:32 2012 -0400

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit d83b6ddda83be81e30df8e45bf3dfeacb8d98d11
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 17:26:28 2012 -0400

    update

commit 3ee44ee0fb9510596d3c321f0cc787e46e232d1f
Author: Lee Hall <shall4@spsu.edu>
Date:   Sat Nov 3 17:26:19 2012 -0400

    Added parent ids

commit c88323d15331cde6b57cf0bacf67ff99cf2e90cc
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 17:25:10 2012 -0400

    update
```

```
commit 9233defe730ce9109b433f2965964f91994dd639
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 17:22:15 2012 -0400

    tedious

commit d361d3f5b0dbaa1e6a53d8968d5bf6ec8ef841e2
Merge: 1ef070d b654263
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 17:18:08 2012 -0400

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit 1ef070d5ed023eae209c5f5666dc50c5a9f0f1b3
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 17:17:26 2012 -0400

    Now please work!!

commit 614ff6666aebed61f6074ab55b9e73abe987c19b
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 17:14:26 2012 -0400

    pleas ework!!

commit b65426346339cfe4d53b17e11916ba648e3780fb
Merge: 4fbd5f7 614ff66
Author: Lee Hall <shall4@spsu.edu>
Date:   Sat Nov 3 17:13:35 2012 -0400

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit 4fbd5f74a694c96bf8a83ec769907c6ad17d6332
Author: Lee Hall <shall4@spsu.edu>
Date:   Sat Nov 3 17:13:08 2012 -0400

    Bug fixes for crawler

commit 2e6b871e5d460712ad8c211f3a49c1da999240f3
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 16:57:50 2012 -0400

    Scan View Update!!

commit 7dae8e506dcfda0d1b619db79e4c225a73299e2e
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 16:41:40 2012 -0400

    fad

commit 8867cbc9b784f36dc56b82edd9f358852146d563
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 16:39:51 2012 -0400
```

```
    updates!!

commit 9ce46fe1b8270cd624f306f91c6d644f5b1bad7b
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 16:35:17 2012 -0400

    update

commit fba1d7ea688ff62bacac56df0e750d3701a7cabc
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 16:31:21 2012 -0400

    Scan viewing

commit 105670444710a74fcd348040bfb2198c949be404
Merge: f6deba2 d3321bc
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Nov 3 16:16:20 2012 -0400

    Merge branch 'master' of http://github.com/lhall23/forager-t4

commit f6deba2d2de1f743952998f01eabe1a0d3f09498
Author: M <swordthane@gmail.com>
Date:   Sat Nov 3 15:16:54 2012 -0400

    DBcalling

commit d3321bc7606162f8303b3f94075f9c94d57be439
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 14:33:28 2012 -0400

    reports loading

commit ab8c74de0f1213a9682467415558f1a50f365dd4
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sat Nov 3 14:26:41 2012 -0400

    test

commit 3910dbc451f6d3d836c6b44863436169d033a4a1
Author: Lee Hall <shall4@spsu.edu>
Date:   Thu Nov 1 23:50:03 2012 -0400

    Tied login together with main page

commit 34bdceee1cb1b91d64d38e9ccd72bdf6e4df2556
Author: Lee Hall <shall4@spsu.edu>
Date:   Thu Nov 1 23:49:17 2012 -0400

    Added some more tests to the html

commit 9ef7762460ec2f535cd26454fed5c995eb66f673
Author: tcouture127 <tcouture127@gmail.com>
```

```
Date:   Thu Nov 1 16:02:35 2012 -0400

    Week 2 Status Reports

commit 64505daaab63d647d0b096f5a80b202ac6295d21
Author: Robin Mays <rmays36@gmail.com>
Date:   Thu Nov 1 14:49:34 2012 -0400

    Updated main

    Updated main

commit 72390dd555a4a5d3fee1ac7c38e267b34c6d6fbb
Author: Robin Mays <rmays36@gmail.com>
Date:   Thu Nov 1 14:46:45 2012 -0400

    Reports.php push

    Adding reports to the repository

commit 8dcb6b8171f493c3c90959d42556869973d3b8ad
Merge: 7e0c733 30738f3
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Oct 30 21:19:39 2012 -0400

    Merge branch 'master' of https://github.com/lhall23/forager-t4

commit 7e0c7332d09ff986613d1b718473fee1e512a3b2
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Oct 30 21:19:01 2012 -0400

    Main Page Push!

commit 30738f35eafcb08422563f714939ad772d31c4b4
Author: M <swordthane@gmail.com>
Date:   Tue Oct 30 19:01:25 2012 -0400

    Commnets

commit c91a36d78f86482b939e6fe3a6933b5558cc9d63
Author: Lee Hall <lhall@newfields.com>
Date:   Tue Oct 30 17:45:20 2012 -0400

    Added sample data to schema

commit 67ee3c3ecfc0bd564a7c4a6eddfc9f78471f642e
Author: Lee Hall <lhall@newfields.com>
Date:   Mon Oct 29 10:44:59 2012 -0400

    Bug fixes

    Made sure that https:// references were canonicalized properly. As a
    result, we seem to get SSL for free.
    Fixed some bugs in the parsing code -- make sure that exceptions in the
    parser are handled, and try to do a better job not feeding it non-html
    documents.

commit 988ff73a13acc913379dcd65bd854e269c415724
Merge: 901312e 537bcb2
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Oct 27 17:15:18 2012 -0400

    Merge branch 'master' of github.com:lhall23/forager-t4

commit 901312e8344c2deb103635728776a855c6269854
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Oct 27 17:14:06 2012 -0400

    Crawler v2

    Added support for fetching resources fetched in script, image and link
    tags. Don't parse images.

commit 537bcb22a8ace5d7d347339088b64d4608d008aa
Author: Lee Hall <shall4@spsu.edu>
Date:   Sat Oct 27 16:35:58 2012 -0400

    Added some test data for the webcrawler

commit bccdff342c88bb9b8f646ab805dff4f8b2eada34
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Oct 27 16:31:23 2012 -0400

    new .gitignore

commit 48df4e172dbf686da587cbc8ea1f8d7875e3ef49
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Oct 27 16:30:36 2012 -0400

    Crawler v1

    The crawler works and is restricted to a single domain. It currently
    only retrieves URLS from anchor tags, and does not record where it has
    been.

commit c73cb8352f8f463745c5de0bba034ca997ee8d03
Author: Lee Hall <lhall@newfields.com>
Date:   Thu Oct 25 17:59:49 2012 -0400

    Status reports and other documentation added.

commit 2ae633e9d42030a8977e550f4a7054756514093b
Author: Lee Hall <lhall@newfields.com>
Date:   Wed Oct 24 16:40:56 2012 -0400

    Documentation additions
    Added Documents and cleaned up a bit

commit 2abe9879ef073258916d5b7fd3b3f640444fdb92
Author: Lee Hall <lhall@newfields.com>
Date:   Tue Oct 23 11:46:39 2012 -0400

    Copied login from honeycomb project

    Removed unused functionality (password requests, registration, etc).

commit d86ad221554d66c6a647fb1930fd935fd2759e68
Author: Lee Hall <shall4@spsu.edu>
Date:   Sat Oct 20 22:25:19 2012 -0400

    Fixed some dependencies

commit 29005c3b116c534b9fa44889b1223ca8130f5411
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Oct 20 21:58:50 2012 -0400

    Filed Documentation

commit b2a2771aaeb2e77ce6a4c479f96f24ea582c1abc
Merge: b1f45f0 f88c442
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Oct 20 21:49:38 2012 -0400

    Merge branch 'master' of github.com:lhall23/forager-t4

commit b1f45f0a59adc63268cdc5abf8908931ce1fcd11
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Oct 20 21:48:31 2012 -0400

    Database schema

commit 83619d1d4a4e5c6ac862df9d0c29f9068c2337b7
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Oct 20 21:47:54 2012 -0400

    Documentation system

    Makefiles for the documentation.

commit f88c442063b3d8e4b69f71c48f5ca0c14574a41d
Author: M <swordthane@gmail.com>
Date:   Fri Oct 19 16:17:26 2012 -0400

    UserStorys

commit 9e623ebe69844fe0b0628506c487fb8c959765bf
Author: lhall23 <twitch@gtf.org>
Date:   Thu Oct 18 20:28:59 2012 -0700

    Initial commit
```

Group 4: Product Demo

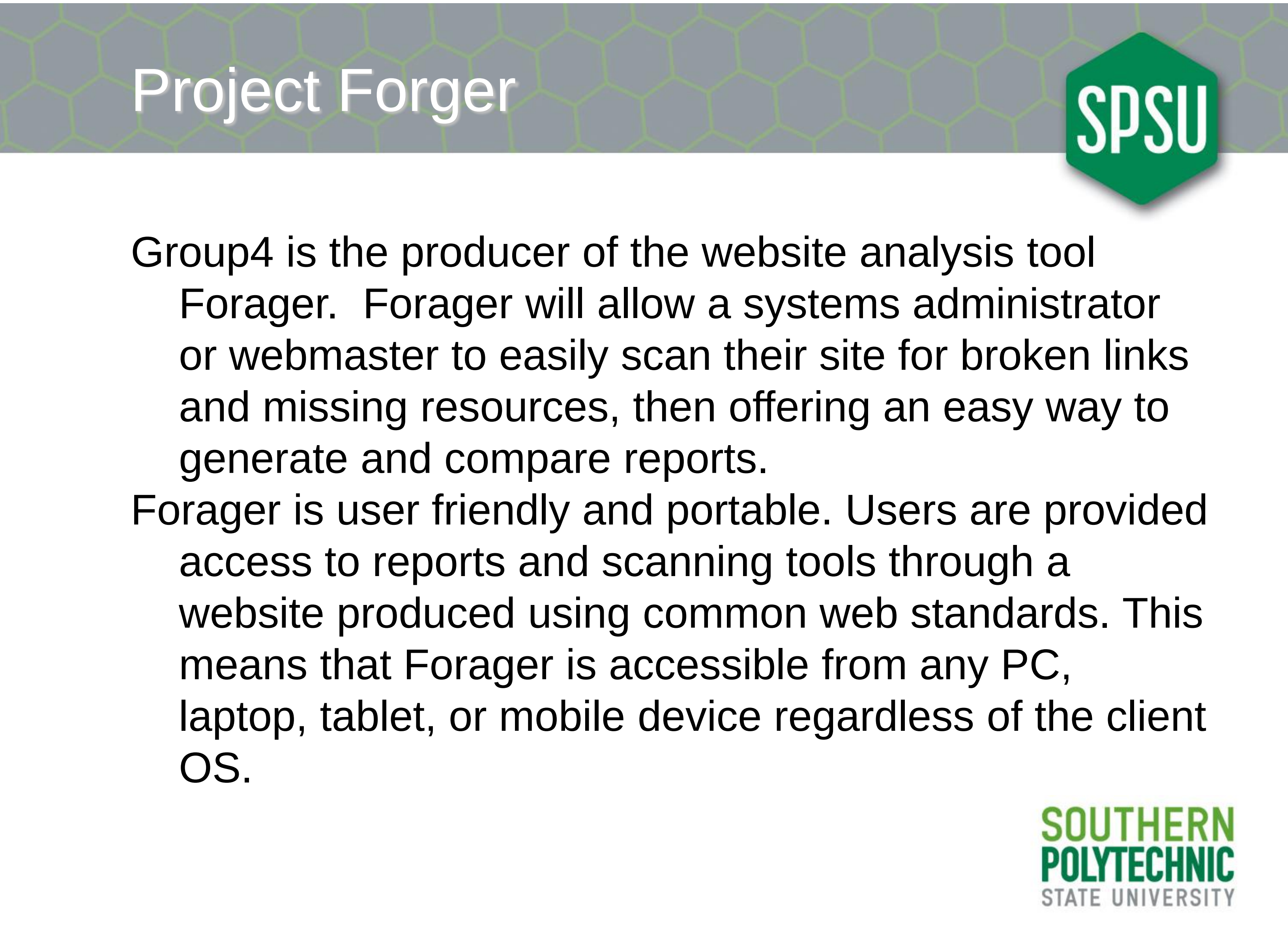
Matthew Powell

Robin Mays

Samuel Lee

Thomas Couture

SOUTHERN
POLYTECHNIC
STATE UNIVERSITY



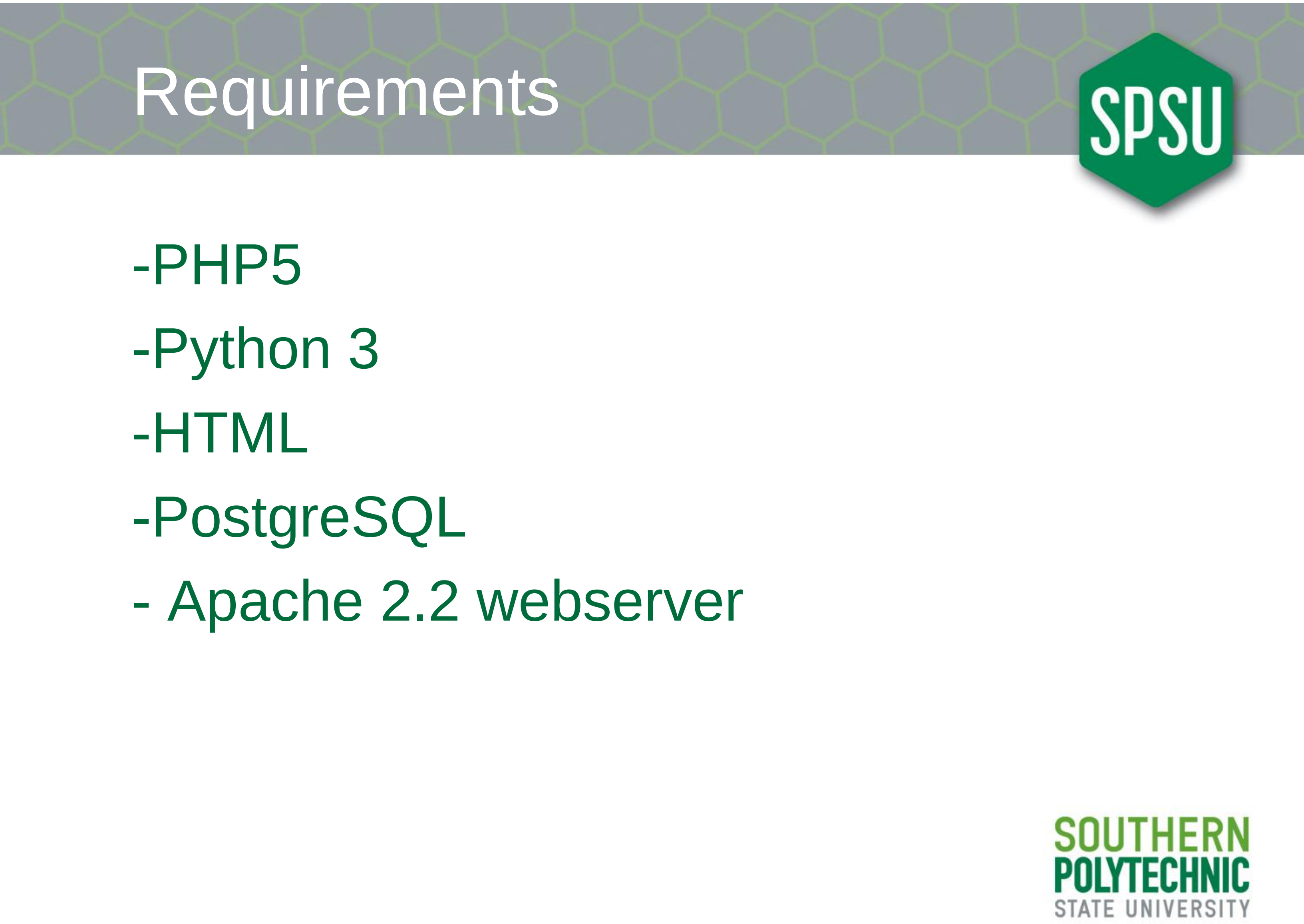
Project Forger

SPSU

Group4 is the producer of the website analysis tool Forager. Forager will allow a systems administrator or webmaster to easily scan their site for broken links and missing resources, then offering an easy way to generate and compare reports.

Forager is user friendly and portable. Users are provided access to reports and scanning tools through a website produced using common web standards. This means that Forager is accessible from any PC, laptop, tablet, or mobile device regardless of the client OS.

SOUTHERN
POLYTECHNIC
STATE UNIVERSITY



Requirements

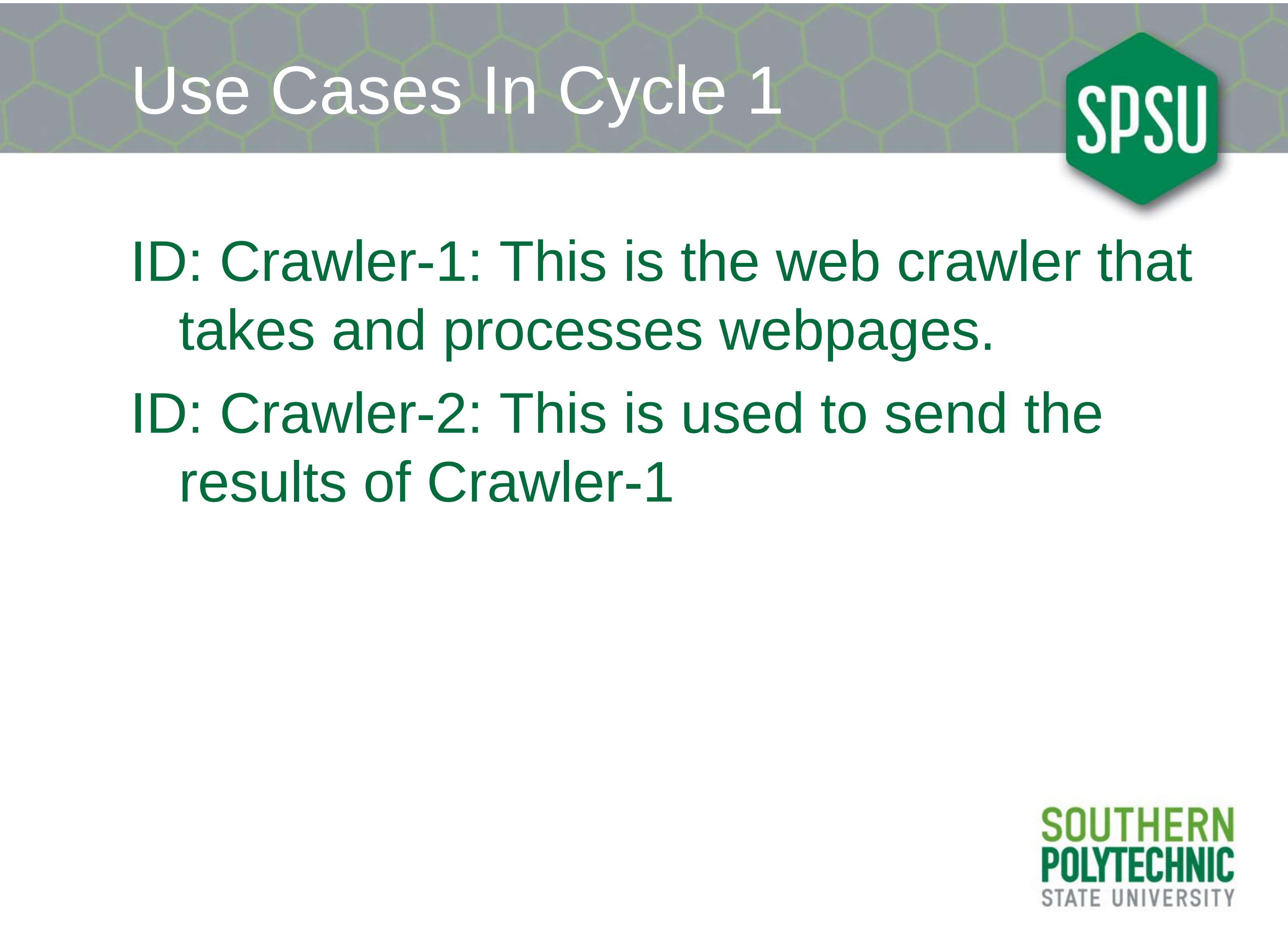
-PHP5

-Python 3

-HTML

-PostgreSQL

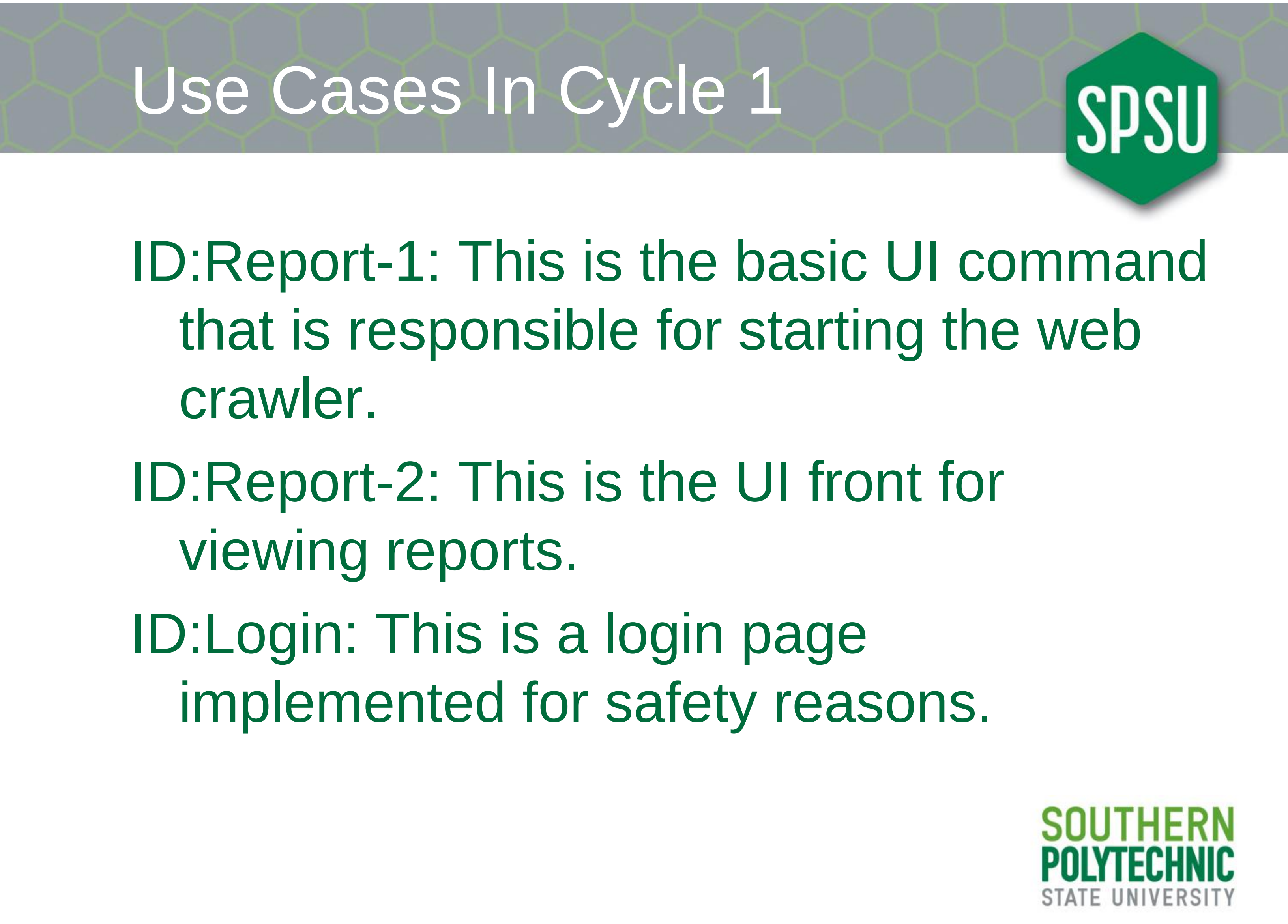
- Apache 2.2 webserver



Use Cases In Cycle 1

ID: Crawler-1: This is the web crawler that takes and processes webpages.

ID: Crawler-2: This is used to send the results of Crawler-1

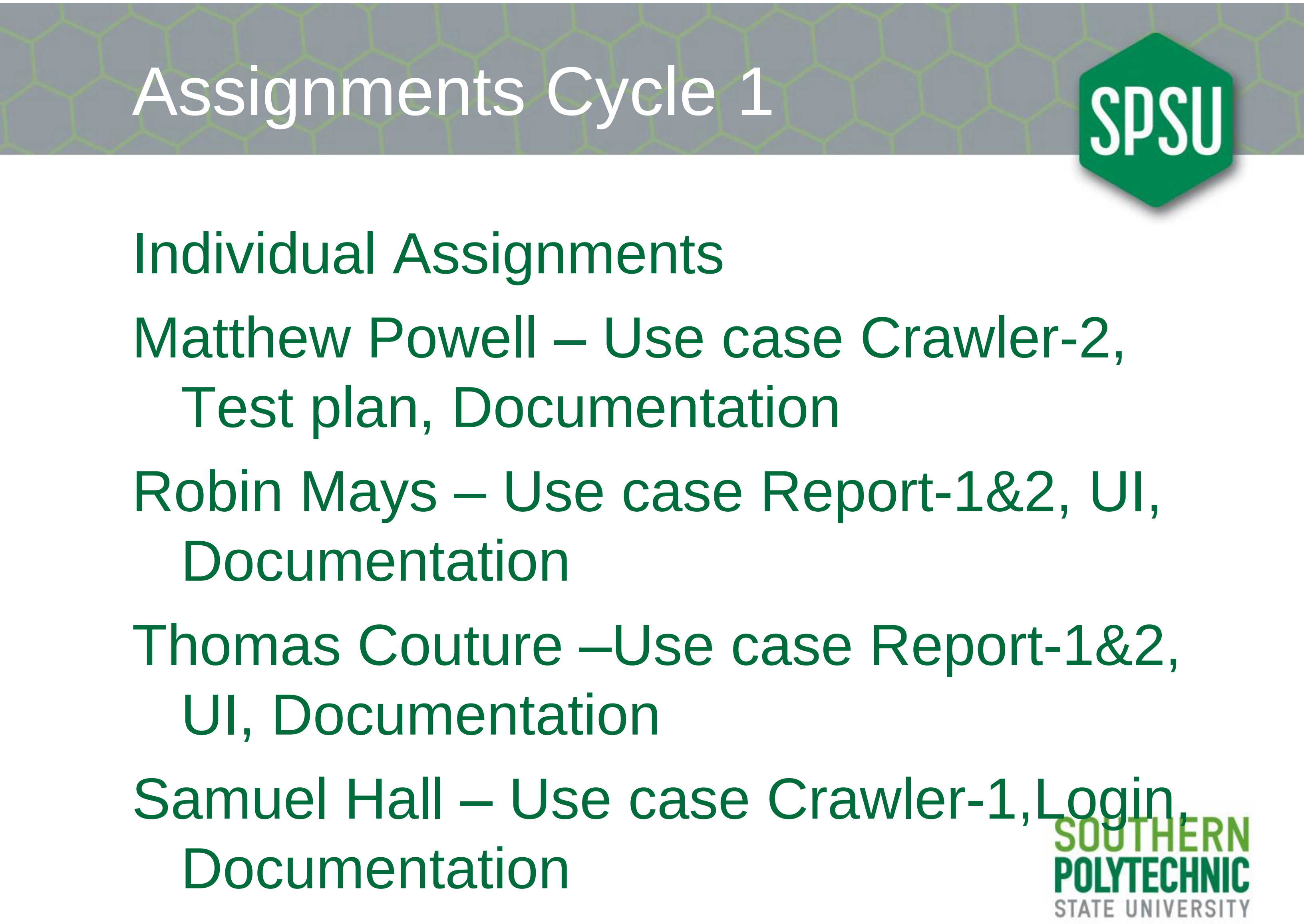


Use Cases In Cycle 1

ID:Report-1: This is the basic UI command that is responsible for starting the web crawler.

ID:Report-2: This is the UI front for viewing reports.

ID:Login: This is a login page implemented for safety reasons.



Assignments Cycle 1

Individual Assignments

Matthew Powell – Use case Crawler-2, Test plan, Documentation

Robin Mays – Use case Report-1&2, UI, Documentation

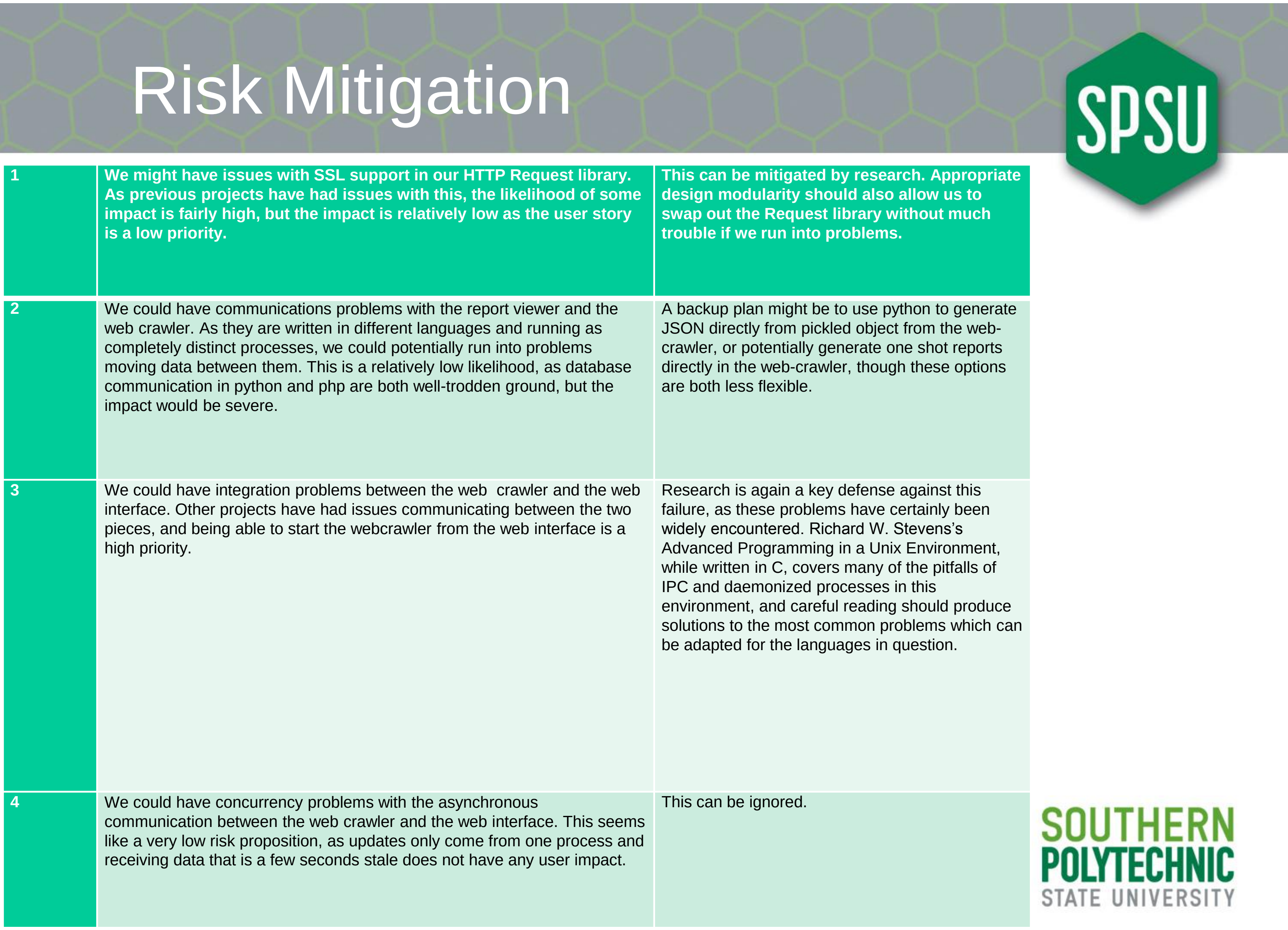
Thomas Couture –Use case Report-1&2, UI, Documentation

Samuel Hall – Use case Crawler-1,Login, Documentation

Planned Use Cases for Cycle 2

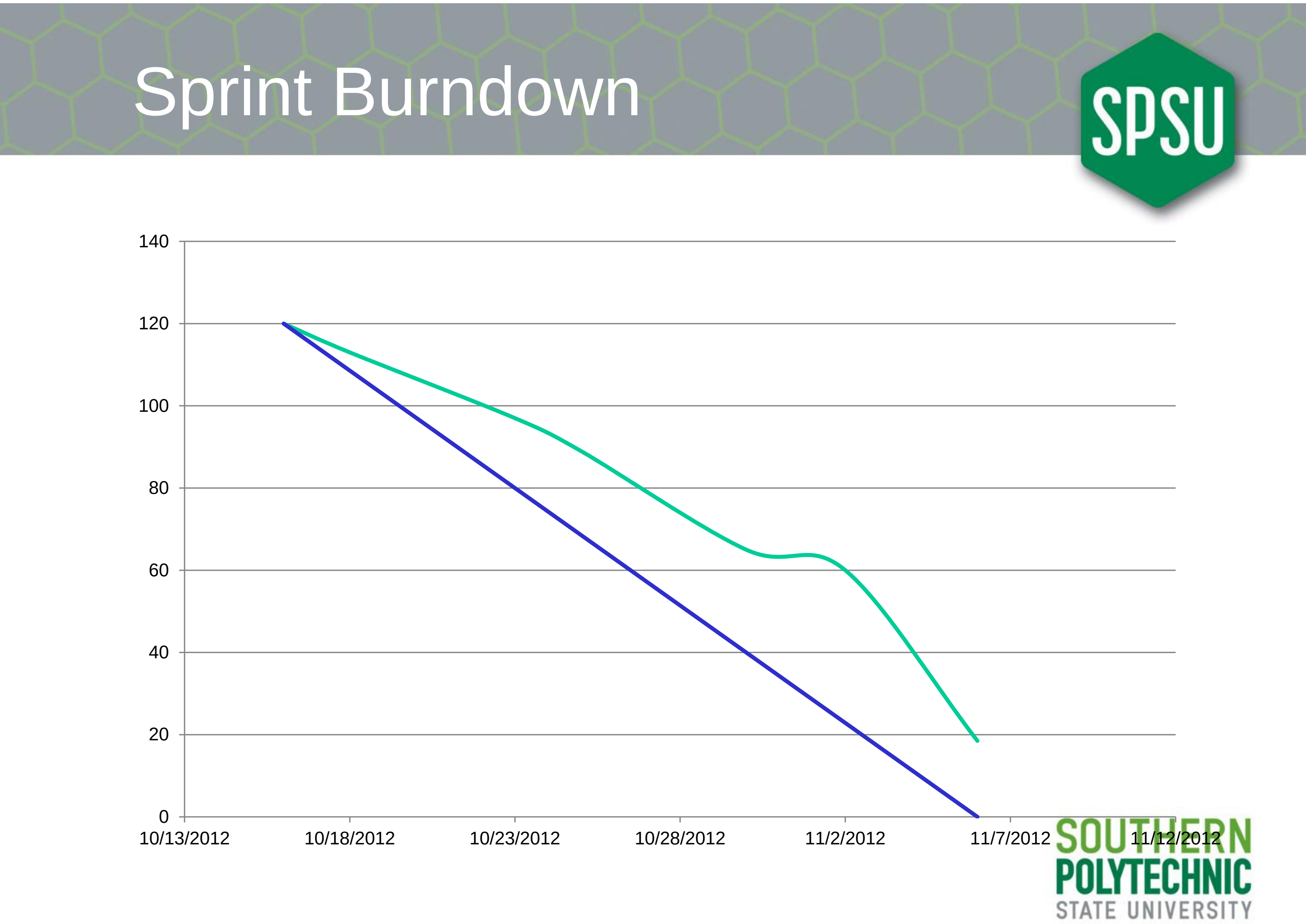
Report-3,4,5,6,7

Crawler-3,4,5,6,7



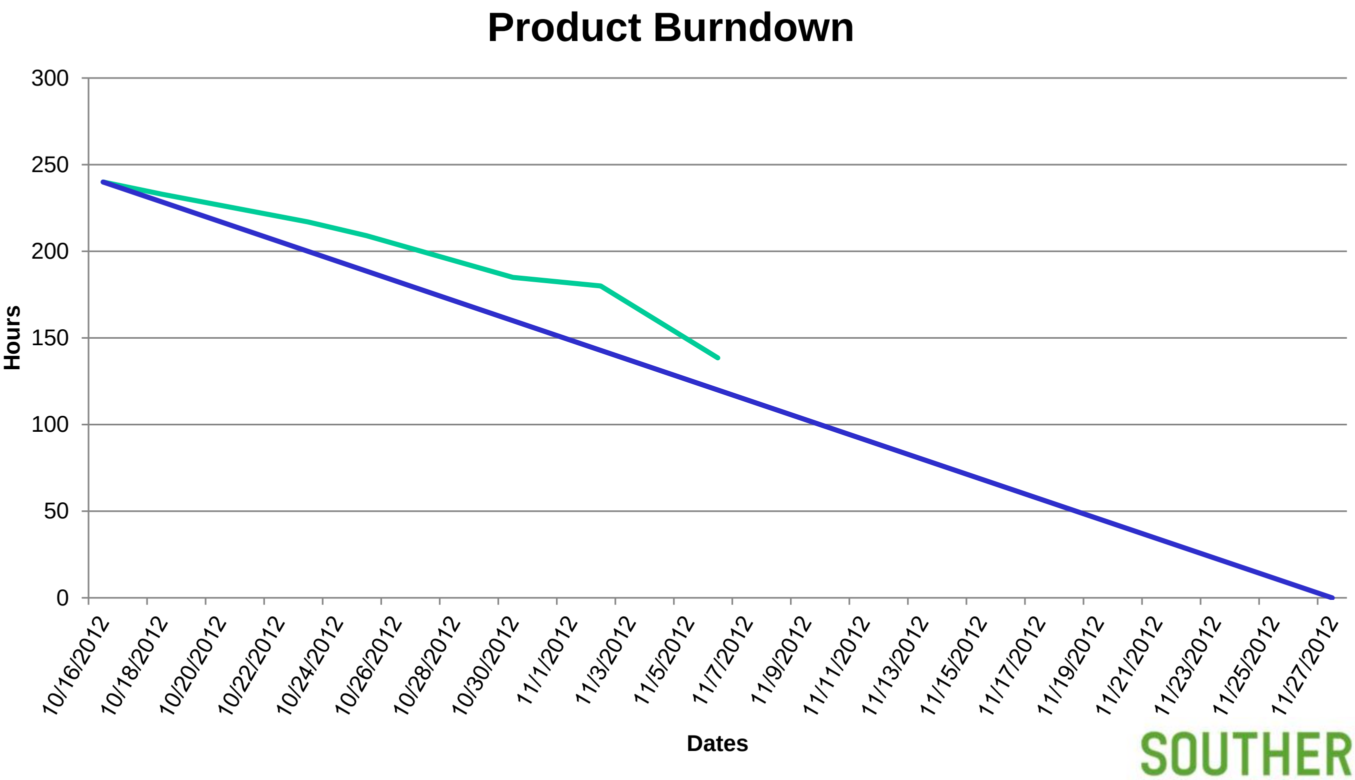
Risk Mitigation

1	**We might have issues with SSL support in our HTTP Request library. As previous projects have had issues with this, the likelihood of some impact is fairly high, but the impact is relatively low as the user story is a low priority.**	**This can be mitigated by research. Appropriate design modularity should also allow us to swap out the Request library without much trouble if we run into problems.**
2	We could have communications problems with the report viewer and the web crawler. As they are written in different languages and running as completely distinct processes, we could potentially run into problems moving data between them. This is a relatively low likelihood, as database communication in python and php are both well-trodden ground, but the impact would be severe.	A backup plan might be to use python to generate JSON directly from pickled object from the web-crawler, or potentially generate one shot reports directly in the web-crawler, though these options are both less flexible.
3	We could have integration problems between the web crawler and the web interface. Other projects have had issues communicating between the two pieces, and being able to start the webcrawler from the web interface is a high priority.	Research is again a key defense against this failure, as these problems have certainly been widely encountered. Richard W. Stevens's Advanced Programming in a Unix Environment, while written in C, covers many of the pitfalls of IPC and daemonized processes in this environment, and careful reading should produce solutions to the most common problems which can be adapted for the languages in question.
4	We could have concurrency problems with the asynchronous communication between the web crawler and the web interface. This seems like a very low risk proposition, as updates only come from one process and receiving data that is a few seconds stale does not have any user impact.	This can be ignored.



Sprint Burndown

Product Burndown



Post Mortem Successes

-Creation of a functional web crawler

-Implementation of basic UI functions and report viewing

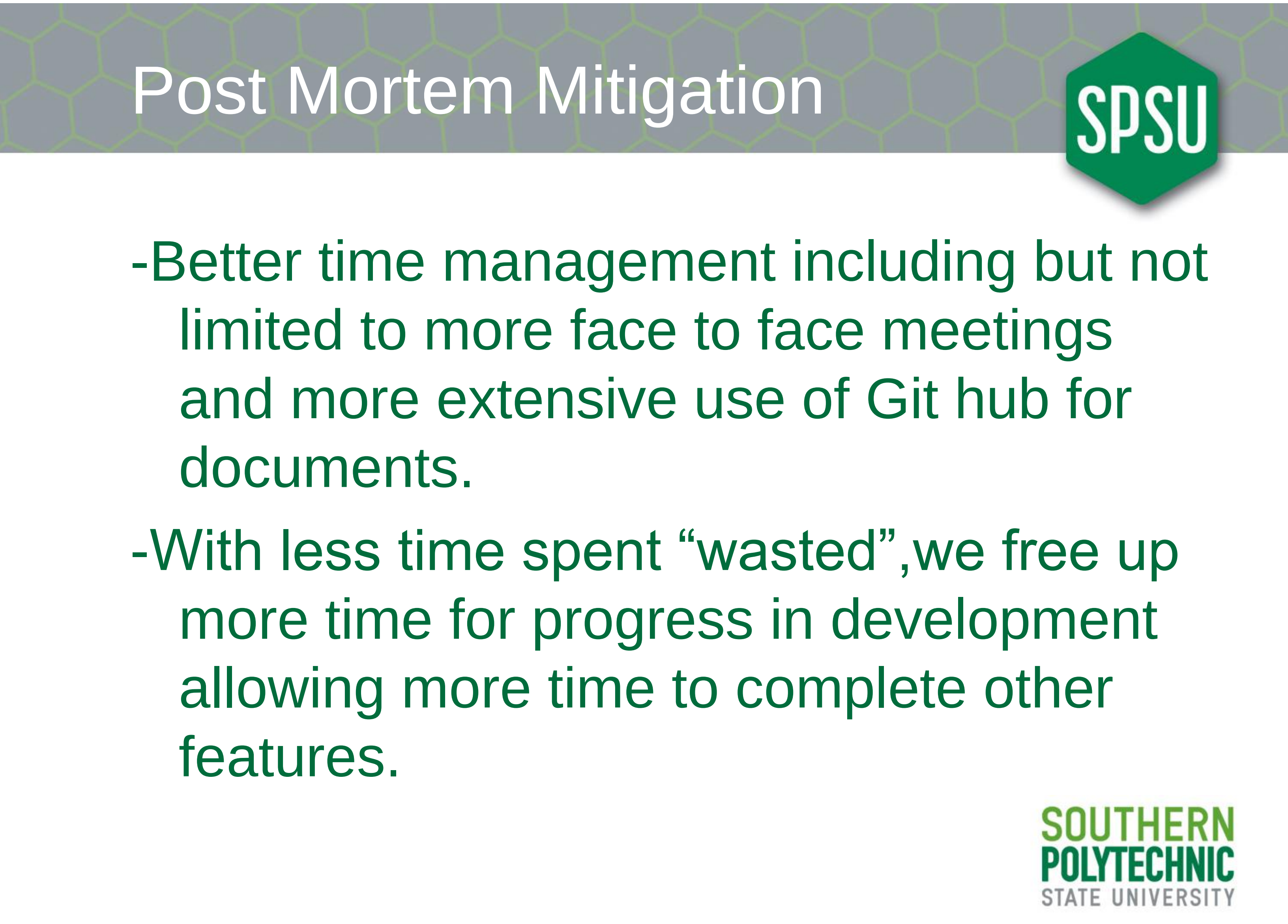
-As a bonus, some sorting features were implemented

- Reports are searchable

SOUTHERN POLYTECHNIC STATE UNIVERSITY

Post Mortem Failures

- Time management for Requirements
- Lack of a stop function on web crawler
- Problems viewing report data



Post Mortem Mitigation

-Better time management including but not limited to more face to face meetings and more extensive use of Git hub for documents.

-With less time spent “wasted”,we free up more time for progress in development allowing more time to complete other features.