Forager : Project Analysis Report

Cycle 2
November 29, 2012

by

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

1. Introduction **6**
1.1 Executive Summary **6**
1.2 Project Goals **6**
1.3 Cycle Goals **7**
Sprint 1 7
Sprint 2 8

2. Requirements **9**
2.1 Project Environment, Technology, Hardware, Etc. **9**
2.2 User Stories **9**
2.3 USe Cases **11**

3. Design **26**
3.1 System Architecture **26**

4. Management Plan **29**
4.1 Planned Assignments and Schedule for First Cycle **29**
Week 1: 29
Week 2: 29
Week 3: 29
4.2 Actual Assignments and Schedule for FIRST Cycle **30**
Week 1: 30
Week 2: 30
Week 3: 30

5. Risk Mitigation **33**
5.1 Risk Mitigation **33**

6. Cycle Post-Mortem Analysis **35**
6.1 Successes **35**
6.2 Failures **35**
6.3 Lessons Learned/Risk Mitigation **35**

7. Test Plan and Procedures **37**
7.1 Test Plan **37**
7.1.1 Introduction 37
7.1.2 Test Items 37
7.1.3 Software Risk Issues 37
7.1.4 Approach 38
7.2 Test Results **38**

Appendix A: Slides **A-1**

Appendix B: Status Reports **B-1**

Appendix C: Minutes **C-1**

Appendix D: Email **D-1**

Appendix E: Source Code.....E-1
Appendix F: Commit Logs F-1

LIST OF ABBREVIATIONS

HTML	Hypertext Markup Language
JSON	JavaScript Object Notation
OS	Operating System
PID	Process ID
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, and offer an easy way to generate and compare reports.

Forager is user friendly and portable. Users are provided access to reports, scanning, and sorting 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

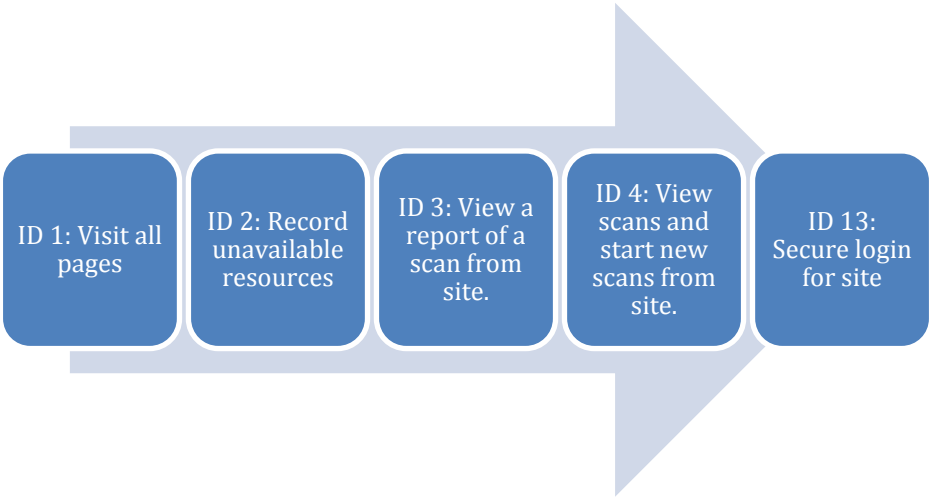


Figure 1 User stories to complete for Sprint 1

Sprint 2

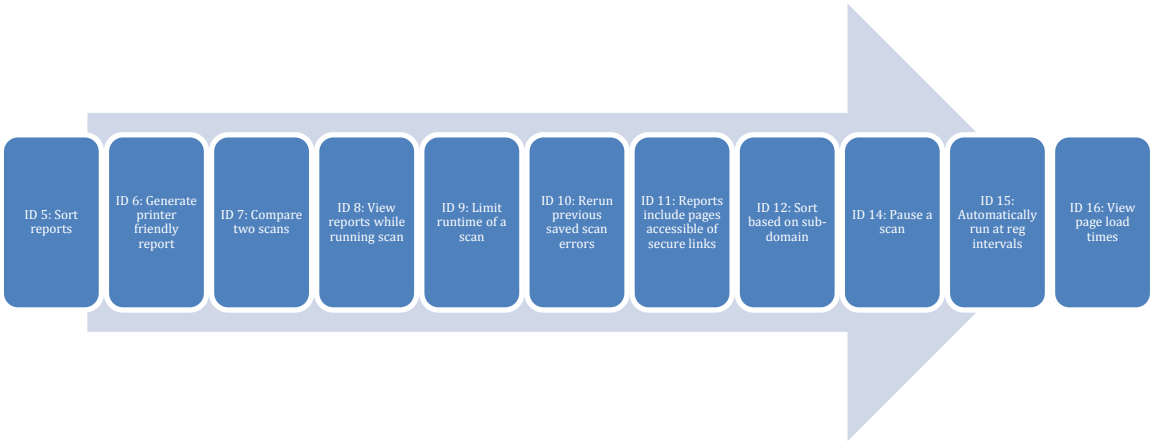


Figure 2 User stories to complete for Sprint 2

2. REQUIREMENTS

2.1 PROJECT ENVIRONMENT, TECHNOLOGY, HARDWARE, ETC.

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

These are the general requirements behind the project that were given to us by the product owner. These are non-technical and should be understood by all parties.

User Stories			
ID	Description	Priority	Cost
1	As a user, I would like to be able to visit and access all pages of my website.	10	20
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.	10	20
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.	10	30
4	As a user, I would like to view scans and start new scans from a website.	10	20
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.	7	10
6	As a user viewing a report, I should be able to generate that report in a printer friendly format.	6	10
7	As a user, I would like to be able to select two scans and show only the items that have changed.	6	6
8	As a user viewing a report, I should be able to view reports from scans that are in progress	6	4
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.	6	12
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.	5	10
11	As a user, I would like for reports to include pages that are accessible over secured links.	5	8
12	As a user, I would like to sort a report based on the subdomain.	3	14
13	As a user, I should have to login before initiating a scan or viewing a report.	3	2
14	As a user, I might like to pause a scan that was currently in progress.	2	4
15	As a user, I would like to have scans that were automatically run at regular intervals.	1	6
16	As a user, I would like to see page load times in my reports.	1	2

2.3 USE CASES

These technical use cases come from the above user stories and are used to determine what gets done in a project cycle.

ID: Crawler-1

Priority: 10

Cost: 20

Name: Basic Web Crawler

Description: The system should be able to visit all of the pages on the website.

Precondition: Report-1 (A scan has been initiated.)

Primary flow:

1. The web crawler will retrieve the root page of the domain.

2. The web crawler will create a page object with the relevant information and place it in a pending queue.

3. If there is a page object in the pending queue, it will be popped, and become the current page. If there is not, the scan is finished, and the crawler exits.

4. The web crawler will parse the retrieved page for resource URLs and insert new page objects into the pending queue if they have not yet been visited and are not already queued.

5. Return to 4.

Alternate flow:

1. The web crawler will initialize a visited list and a pending queue.

2. The web crawler will retrieve the root page of the domain.

3. If the root page cannot be retrieved, the scan exits.

Postconditions: Every URL referenced in any page reachable from the root page will be visited. This does NOT guarantee that it will be retrieved, as it may not exist.

ID: Crawler-2

Priority : 10

Cost: 20

Name: Record Crawler Results

Description: The results of crawling the website will be recorded in a database so that they can be used later.

Preconditions: The webcrawler must be running. The webcrawler must have generated a record in the "scans" table and retrieved a scan ID.

Primary flow:

1. When a page object is created, a matching entry in the database will be created, including the scan id, a link to the parent page (this can be null for the root of the scan), the current URL, the retrieval time, and the HTTP response code.

2. The resource_children table will be updated to create a link between the parent and the new child object.

3. The resource id of the newly created record will then be stored in the page object.

Alternate flow:

1. When a child resource is found to already have an associated page object in the pending or visited queue, a link will be created in the resource_children table between the current page object and the one that was found.

Postconditions: The resources table will contain a complete list of all reachable nodes on a graph of the website. The resource_children table will contain a complete list of the reachable edges of a graph of the website. The resources table will also contain results codes, retrieval times and a spanning tree of the graph of the website. As the response codes will be recorded and records will be created even for pages that are not accessible, this will be a record of broken links as well as working ones.

ID: Report-1

Priority : 10

Cost: 20

Name: Crawler Interaction

Description: The user must be able to start new scans and list the completed scans.

Preconditions: ID-13 (Login)

Primary flow:

1. The system will display a button to start a new scan.

2. The system will retrieve a list of scans that have been completed.

3. The system will display the list of completed scans.

4. The user may click on the "Start Scan" button.

5. When the button is clicked, a scan will be dispatched via a call to exec();

Alternate flow:

1. The system will display a button to start a new scan.

2. If there are no scans completed, none will be listed.

3. The user may click on the "Start Scan" button.

4. When the button is clicked, a scan will be dispatched via a call to exec();

Postconditions: The user will know what scans have been completed. If the button was depressed, a scan will have been started.

ID: Report-2

Priority : 10

Cost: 30

Name: Show Scan Results

Description: Reports of the scans will be visible to the user, showing all irretrievable resources under the specified domain.

Preconditions: Crawler-2 (A scan must have been recorded), **Report-1** (the user must have a list of scans.)

Primary flow:

1. The user will select a scan that has been run.

2. The scan container will open.

3. The scan container will request the scan data from the webserver.

4. The webserver will retrieve all page resources that match this scan id, format them as JSON, and return them to the scan container.

5. The scan container will then display the JSON results.

Alternate flow: None

Postconditions: The scan will be displayed.

ID: Report-3

Description: The page will offer buttons to sort the reports by the order in which the pages were visited, alphabetically, and alphabetically by the parent page.

Precondition: Crawler-2 (A scan must have been recorded.)**, Report-2** (A report must be displayed)

Primary Flow:

1) The system will retrieve data from the database.

2) The system will sort data based on the sorting order selected by the user.

3) The system will display the results of the sort in the website's UI.

Postcondition: The system will display reports sorted in the manner selected by the user.

ID: Report-4

Description: Reports will be displayed in a printer-friendly format.

Precondition: Crawler-1,Crawler-2,Report-1, Report-2, Report-3

Primary Flow:

1) The system will retrieve data from the database.

2) The system will sort retrieved data.

3) The system will display sorted or unsorted data in the website UI.

4) The system will parse the data generated in the web UI.

5) The system will create a printable text based format of the data displayed in the website's UI.

6) The system will display the printable version of the scan results in a new window.

Postcondition: The system will generate a new window displaying a printable text based format of the results of the scan.

ID: Report-5

Descripton: The system will be able to select two scans and show only the items that have changed.

Precondition: Crawler-2 (There must be a scan.), **Report -2** (The user must have a scan open.)

Primary Flow:

1) The system will be able to retrieve the results of two scans.

2) The system will be able to compare the results of the two scans.

3) The system will be able to display the results of the two selected scans that do not match.

Postcondition: The system will have shown the differences between two scans.

ID: Report -6

Description: The system will be able to display reports from scans that are in progress.

Preconditions: Crawler-1 (A scan is currently in progress.)

Primary Flow:

1) A scan will be started by the system.

2) The system will be able to retrieve data from the database while a scan is in progress.

3) The system will be able to display data from the database in the web UI while a scan is in progress.

Postcondition: The system will be able to retrieve data from the database while a scan is in progress.

ID: Crawler-3

Priority: 6

Cost: 12

Name: Runtime Limit

Description: The scan can take a long time to fully run. As a result, this use case is designed to allow a user to limit the run span of the scan. The user can decide to have the scan run as much as it can in a certain amount of time, or to limit the scan to only run until it gets to a certain distance away from the main page.

Pre-Condition: **Login** (User has logged in.)

Standard Flow:

1. The user decides how he/she wants to limit the search (either length or depth) and enters the appropriate value in the appropriate location for the choice. This starts the scan.

2. The scan will run as normal and go through all of the pages until it reaches the limit set by the user.

3. The finished report with all the errors for the pages searched will be displayed.

Alternate Flow 1:

1. The first step will be the same as the standard flow.

2. The scan will run until the user pauses the search. This will also pause the timer if the

user chooses to do a time based search. The scan will remained pause until the user resumes the search.

3. Same as step 4 in the standard flow.

Alternate Flow 2:

1. The first step will be the same as the standard flow.

2. The scan will run until the user stops the search. This will terminate the search.

3. The final report will be printed and displayed all of the errors up to the termination of the search, which will not be the same as what the user initially requested.

Alternate Flow 3:

1. The user enters invalid information for one of the two limiting factors trying to be used.

2. An error message will be displayed informing the user that the information entered is incorrect and prompt them to try again.

3. Return to step number 2 in the standard flow.

Post Conditions: A scan will be completed based on the limiting factor given by the user. A report will be completed and displayed for the user to examine. A record of the scan will be saved in the database so that it can be accessed in the future, used to be compared to other searches, or used to start another search to check for errors being fixed.

ID: Crawler-4

Priority: 5

Cost: 10

Name: Error Check

Description: This use case will select a scan that has been run previously. A second scan will begin and will only rescan what was scanned in the previous results that were initially selected. It will return a report and show a comparison of the two reports to show what errors are the same, what errors are new, and what errors are fixed (if anything has been).

Pre-Conditions: Report-1 (The user is looking at the list of scans.)

Standard Flow:

1. The user will select a previous report that he/she wants to see if any changes were made.

2. The user will start a comparative scan.

3.The scan will run and scan the same pages as the previous scan.

4. A report will be generated based on the scan that was just run.

5. The two scans will be compared, showing what errors were still present, any new errors that popped up, and showing which errors were fixed.

Alternate Flow 1:

1.The first 3 steps from the standard flow will be followed.

2. The user will pause the scan during the search.

3. A partial report will be shown, displaying what errors have been encountered thus far.

4. The scan remains paused until the user resumes the scan. The scan continues on normally.

5. Return to step 5 in the standard flow.

Alternate Flow 2:

1.The first 3 steps from the standard flow will be followed.

2. The user will stop the scan during the search. This terminated search.

3. A report will be created based on the partial scan, and the report will compare only the parts of the scan that were completed.

Post Condition: A repeat scan will have run. There will be two reports based off of the same scan. They will be compared and the user will see what. If anything, has changed for either better or worse.

ID: SSL

Priority: 5

Cost: 8

Name: Secure Check

Description: As the web crawler goes through the SPSU domain, it will come across links that are secure. These items should be listed as secure login links and not as errors (as they can not be accessed) and shown as such on the report that is displayed at the end.

Pre-Conditions: Crawler-1(A scan is running.)

Standard Flow:

1. User starts a scan.

2. Web crawler and error reporter are called and begin to work.

3. As the crawler tries to access something in the SPSU domain that requires a secure login, it will report that to the database as a link to a secure site.

4. When the scan has completed, the report will populate all of the items from the scan.

The secure links will appear on the report, and will be listed as secure links instead of errors.

Post Conditions: The user will have an updated report that no longer lists the secure pages as a link that can't be accessed, but as a secure page that requires a login.

ID: Report-7

Priority: 3

Cost: 14

Name: Subdomain Sort

Description: When looking at the report, we will be able to sort it based on the subdomain. This will allow us to break down the report and look at the errors only in certain subdomains, which would be useful to do something along the lines of finding all of the errors in a certain department's webpages.

Pre-Conditions: Crawler-2 (A scan must have been recorded.)**, Report-2** (A report must be displayed)

Standard Flow:

1. User starts a scan and receives a report, or opens a report that has been previous run and recorded.

2. User clicks on the sort feature that says subdomain.

3. A sorting algorithm (using pattern matching) is used to arrange the items in the report based off of their subdomain. The default case will be alphabetical. If the button is pressed twice, the report will be in reverse alphabetical order.

Post Conditions: The user has a viewable report that properly lists the items based on

their subdomain either from a-z or from z-a.

ID:Login

Name: User Login

Priority: 3

Cost: 2

Description: This is a security feature that prevents unauthorized users form making the product into a Denial of Service attack platform or find information about the target website that might be exploitable. This login will be set on creation of the system. The login will require a user name and simple password. The login information will be stored in the data base.

Preconditions: None

Standard flow: User will be prompted for user name and password upon arrival to the website. The user will then type in a user name and password into there appropriate blanks and press the login button or enter.

Alternative flow: IF the user does not enter a field such as password or user name the login will fail. If user enters a wrong user name or password the login will throw a invalid login error. Should the user try and bypass this step web pages beyond this will not be displayed.

Post Conditions: After a successful login the user will be able to go beyond the login page and view or start scans.

Considerations or issues: This is a low priory by the customer however for security reasons needs to be implemented. There also is no registration now and requires the system admin to add new login entries.

ID:Crawler-5

Name: Pause Scan

Priority: 2

Cost: 4

Description: This will temporally pause the web crawler aspect of the scan. This will not stop processes such as report builders. This option is only available after a scan has started. Once the scan has been stopped the user can continue scan at the press of a button.

Precondition: Crawler-1 (A scan is running.)

Standard Flow: User has started a scan and then presses the pause scan. The scan stops and can be restart by pressing continue scan.

Alternative Flow: Scan is unable to restart so error is thrown. User is able to access stop scan before scan has started do nothings.

Post Conditions: Scan continues.

Considerations/ Issues: This might include pausing threads.

ID: Crawler-6

Name: Timer

Priority: 1

Cost: 6

Description: This allows a user to run a scan at a user set time. This involves the server knowing what time it is and starting a process based on that. The scan will not display a live report unless a user has logged in during its run. This can be set up to run once, weekly or monthly.

Precondition: Login (User has logged in)

Standard Flow: A user has logged on and imputed a time and date in the required box and check the required boxes dealing with now often this reoccurs. Should no options be checked and no date and only a time the default is run once.

Alternative Flow: Should the user input nothing no scans will be scheduled. Should an impossible time be imputed no scans will run and a message will display to the user that this is an invalid time.

Post Conditions: A scan will run

Considerations/ Issues: This can be used to make a Denial of service attack if abused. However this scan could be slowed down should this become an issue.

ID: Crawler-7

Name: Stopwatch

Priority: 1

Cost: 2

Description: This will add into the report the time it took to load a page.

Precondition: Crawler-1 (A scan is running.)

Standard Flow: During the web crawler's execution times will be recoded of the load times of web pages.

Alternative Flow: None

Post Conditions: There is now a field in the data base that holds web load times

Considerations/ Issues: Adds a new field to the data base which could potential slow down the scan.

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 a menu allowing them to either interact with the crawler or existing scans. Crawler interaction is mediated with the control_json.php page, which will take several options, allowing it to start a new scan, as well as check the status, pause, or stop a currently running scan. When executed, this page will check the database for the PID of any scans without a stop time and attempt to send a liveness query (SIGUSR2) to them. If any of these signals succeed, this PID is assumed to be that of the currently running scan. Any PIDs where the signals fail are marked in the database as having exited. The script will then either send the signal requested to the running process (SIGUSR1 is used for pause, and SIGCONT for resume), or exec a new process. The PID of the currently running process is then returned in a JSON document, or -1 if there is not one. 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. If this is a new scan, this object initially contains only the URL and a null link to its parent. If this is a rescan of an existing report, the queue is initialized with the failed links from the base scan. 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, and the scan is a new one, 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. When the user opens the controller window, they will be able to stop, start, pause or resume scans. While a scan is running, the contents of that scan will be continually queried by a javascript function and added to the end of the table. The user will also be given the opportunity to compare scans, which will query the database for the contents of both scans and return any URLs that received different return codes.

4. MANAGEMENT PLAN

4.1 PLANNED ASSIGNMENTS AND SCHEDULE FOR FIRST CYCLE

Group 4's 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:

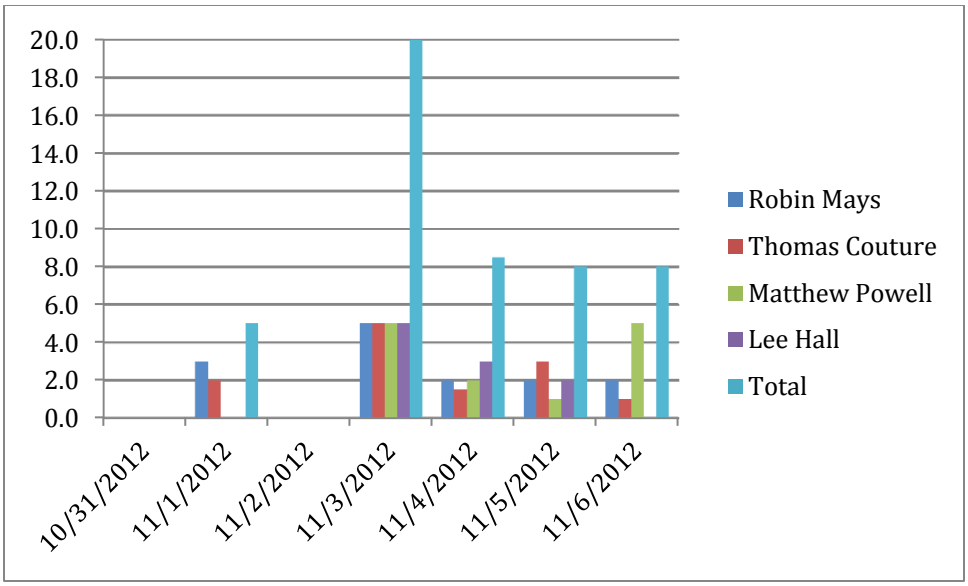
The Team Assignments
Revise requirements
Tune User Stories
Weekly Status Report

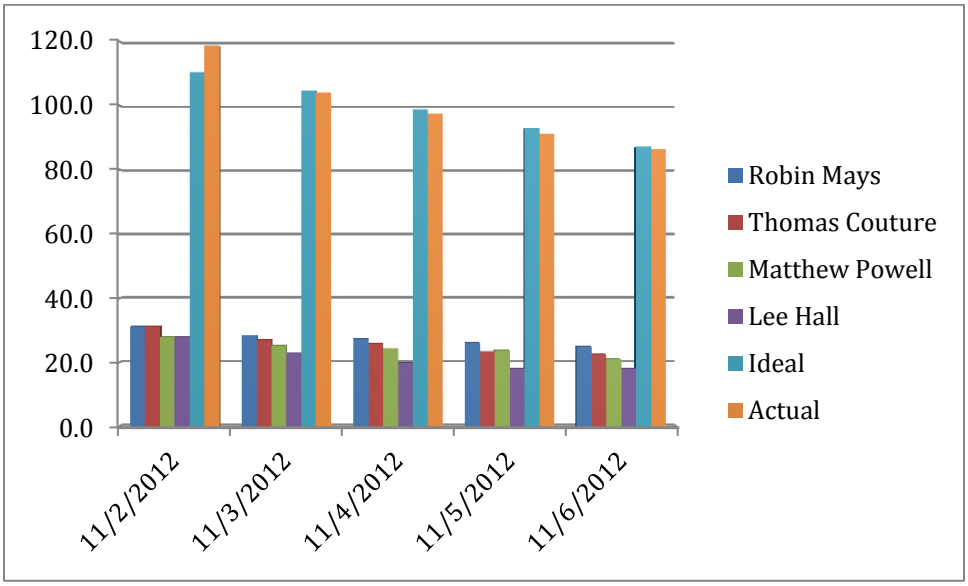
Individual Assignments
Matthew Powell – Coding, Documentation
Robin Mays – Coding: Web UI, Documentation
Thomas Couture – Coding: Web UI, Documentation
Samuel Hall – Coding: Backend, Documentation

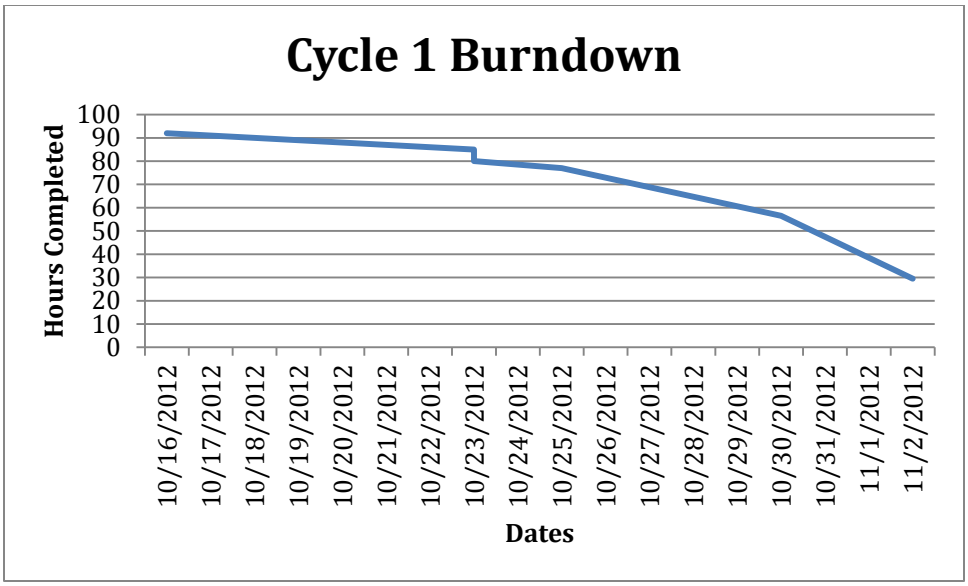
Week 3:

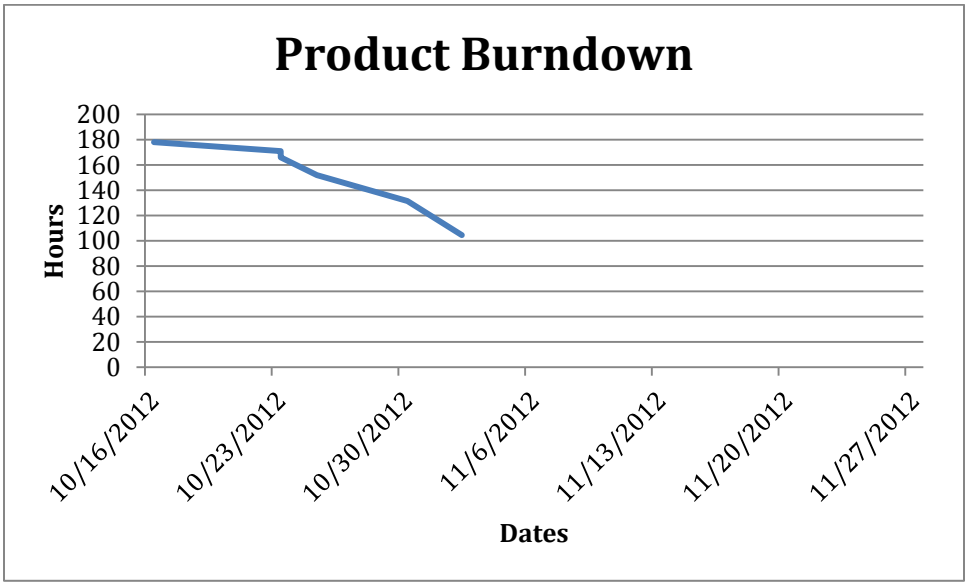
The Team Assignments
Weekly Status Report
Project Analysis Report
Project Demo

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


20.0
18.0
16.0
14.0
12.0
10.0
8.0
6.0
4.0
2.0
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


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


Cycle 1 Burndown
Hours Completed
100
90
80
70
60
50
40
30
20
10
0
10/16/2012
10/17/2012
10/18/2012
10/19/2012
10/20/2012
10/21/2012
10/22/2012
10/23/2012
10/24/2012
10/25/2012
10/26/2012
10/27/2012
10/28/2012
10/29/2012
10/30/2012
10/31/2012
11/1/2012
11/2/2012
Dates


Product Burndown
Hours
200
180
160
140
120
100
80
60
40
20
0
10/16/2012
10/23/2012
10/30/2012
11/6/2012
11/13/2012
11/20/2012
11/27/2012
Dates

5. RISK MITIGATION

5.1 RISK MITIGATION

This data in the below table and chart will be used to help reduce the damage and danger of risks that could potential delay or give unwanted functionality to this project.

ID	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.

	Impact								
Risk	1	2	3	4	5	6	7	8	9
9									
8			ID1						
7								ID3	
6									
5									
4									
3		ID3						ID2	
2	ID4								
1									

6. CYCLE POST-MORTEM ANALYSIS

6.1 SUCCESSES

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

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

6.2 FAILURES

Time management was significantly less than optimal due to communications failures during the creation of use cases. The use of email in this process was inefficient and cumbersome, causing communication to be limited and unproductive. In the future, GitHub will be used to better share documents through the internet and meetings will be held in-person, as that has proven to be the most productive use of the team's time.

6.3 LESSONS LEARNED/RISK MITIGATION

The failure to construct use cases in a timely fashion led to a late start of the project coding. However, when the use cases were rewritten, in most cases it was because they were too in-depth, providing algorithmic details that helped the actual implementation of the use cases. This is a situational problem and cannot be counted on in the future to work out as a net positive, so the plan outlined in 6.2 should be implemented so that this scenario does not happen again. After this process had finished, we also had a better understanding of the fundamentals of use cases, which will allow us to avoid this problem in the future.

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

7. TEST PLAN AND PROCEDURES

7.1 TEST PLAN

7.1.1 Introduction

The test plan for the Forager project will include verifying results of the web crawler and the correctness of the information in the reports. This will be done in such a way to detect errors in the web crawler's data collection as well as potential avenues for abuse of the web crawler or its user interface. This will also cover checking for misleading or inaccurate information and the usability of the web interface.

7.1.2 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 will be done on a fully functional project and not in parts. This will test both the integrity of the parts as well as the integration. Due to the nature of this project testing involved in finding ways to increase speed of the scan will not be undertaken because of potential risk to the SPSU webservers.

7.1.3 Software Risk Issues

As stated above, there is risk that this program could inadvertently cause 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. We also must take into account

that the server running this service has other functions and hosts services of its own, which limits most forms of aggressive penetration testing.

7.1.4 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 to pass 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 order that a normal user would interact with the system.

7.2 TEST RESULTS

7.2.1 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.

7.2.2 Main Page

The user is able to go to the “start a scan” and “view reports” page from 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.

7.2.3 Scan

When arriving at the scan page, the scan starts automatically. This however cannot be stopped via the web UI and needs to be changed. It is noted that 2 scans cannot be run at the same time.

7.2.4 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 send the user to the top of the page without reloading it. All other links from this page work as intended, and 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. 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.

7.2.5 Report Page

"Show # entries" works as intended however there is not 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 #s to search IDs and strings to search the URLs. In some cases, the sorting causes long URLs to display outside of 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 the user to the top of the page.

7.2.6 Crawler verification

After viewing a report some of the URLs 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.

Group 4: Product Demo

Matthew Powell
Robin Mays
Samuel Lee
Thomas Couture

Project Forger

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.

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

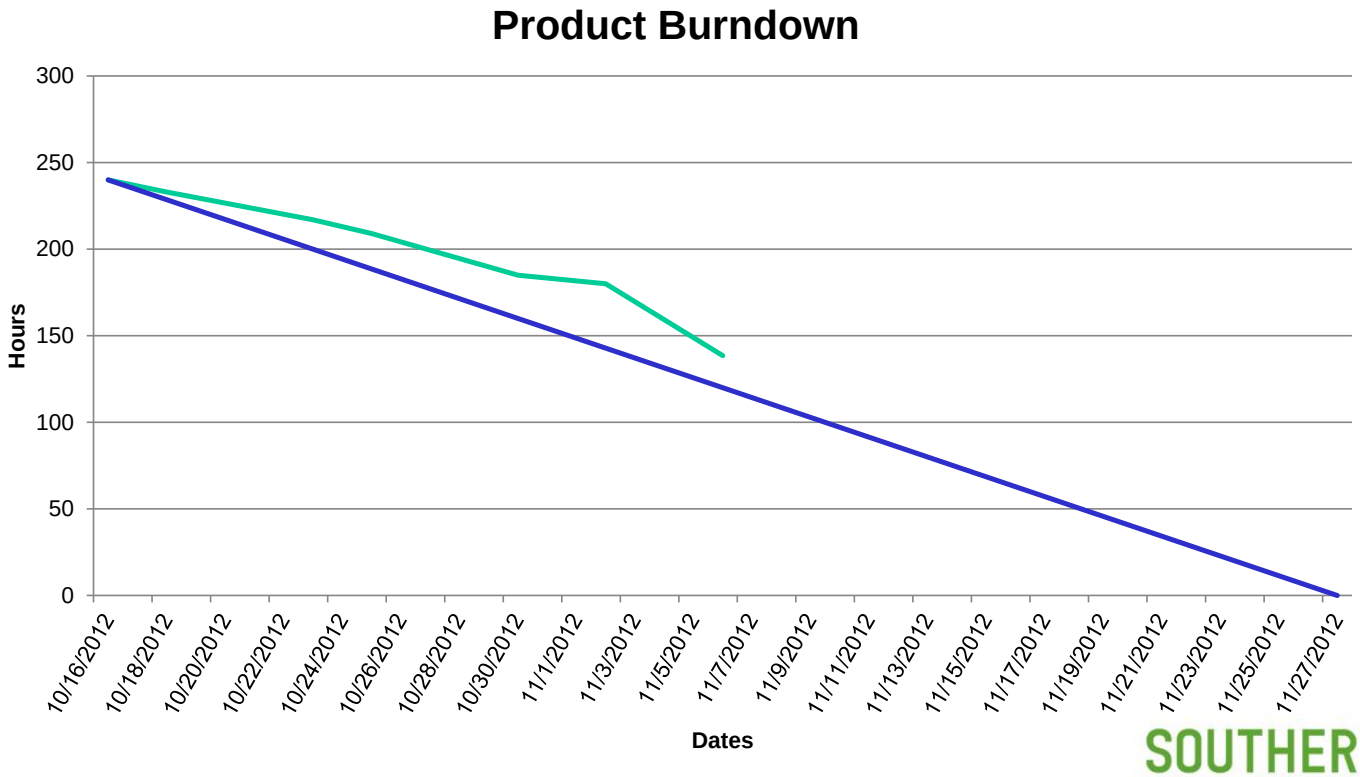
SOUTHERN POLYTECHNIC STATE UNIVERSITY

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

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.

SOUTHERN POLYTECHNIC STATE UNIVERSITY

Group 4: Product Demo

Matthew Powell
Robin Mays
Samuel Lee
Thomas Couture

Project Forger

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.

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

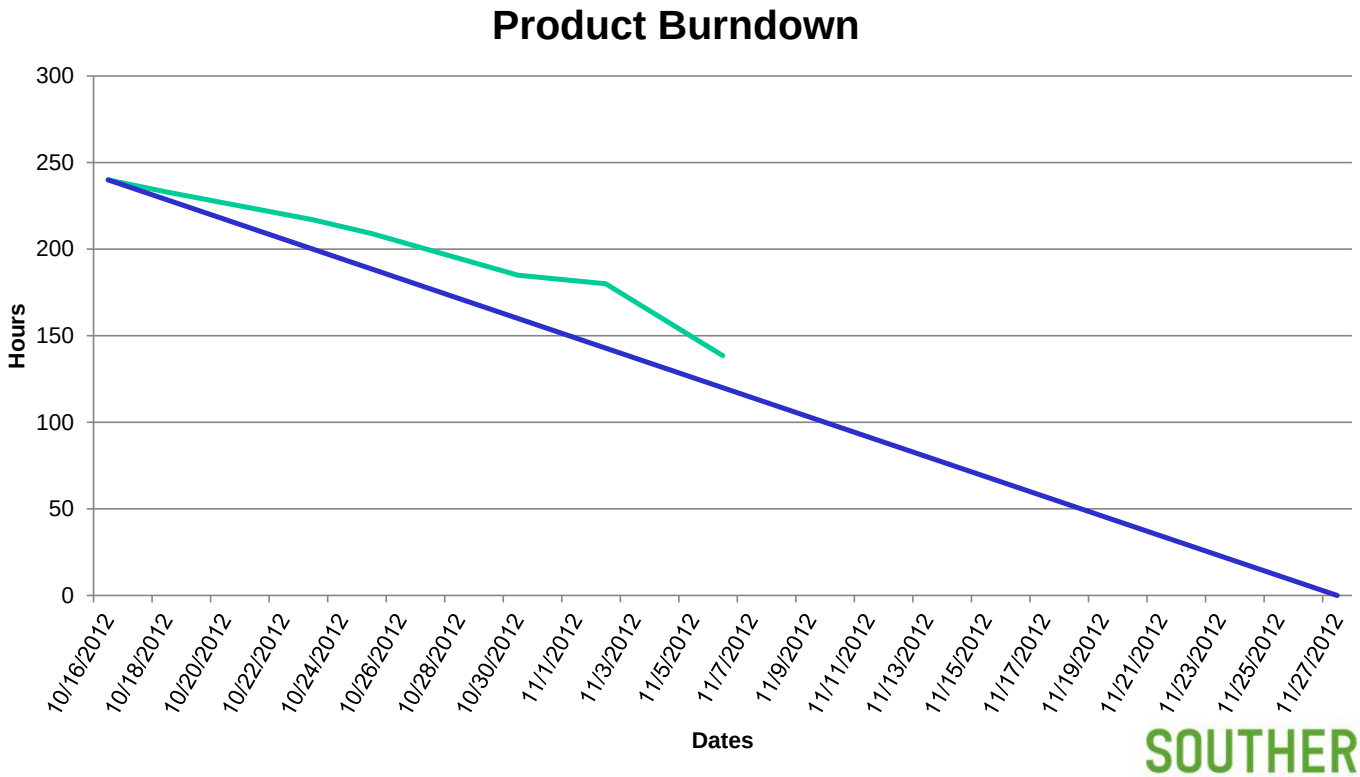
SOUTHERN POLYTECHNIC STATE UNIVERSITY

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

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.

SOUTHERN POLYTECHNIC STATE UNIVERSITY

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.

		Planned			Actual		
#	User Story	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

Name	Requirements			Design / Prototype			Development / Code			Integrate / Test			Documentation			Totals		
	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

Name	1		2		3		4		5		6		7		8		9	
	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

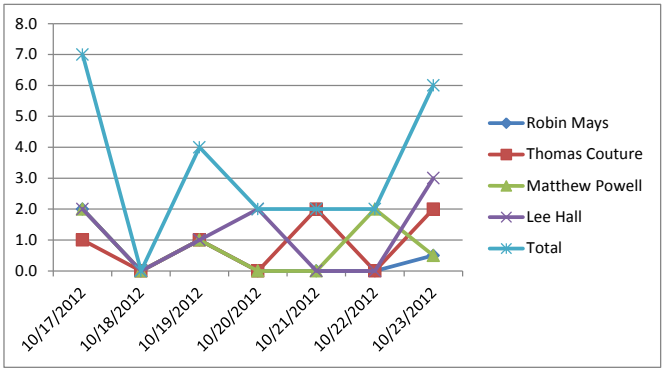
Project Name:	Forager - Group 4
Member:	Matthew Powell
Week Ending:	23-Oct-2012

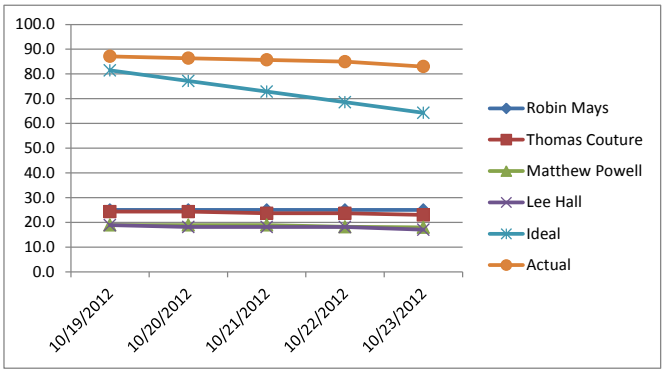
Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	Design for User Story 2
Date:	10/22/2012	Result:	
Hours Worked:	2	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Weekly Status Report
Date:	10/23/2012	Result:	Completed personal section of report/proof read.
Hours Worked:	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.

		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 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

Name	Requirements			Design / Prototype			Development /			Integrate / Test			Documentation			Totals		
	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

Name	Crawler 1		Crawler 2		Report 1		Report 2		Report 3		Report 4		Report 5		Report 6		UC 9	
	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

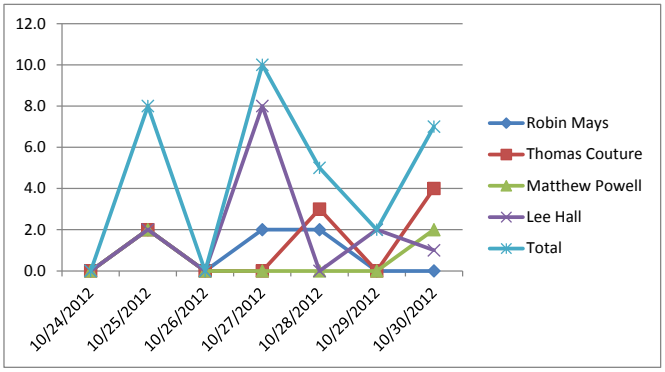
Project Name:	Forager - Group 4
Member:	Matthew Powell
Week Ending:	30-Oct-2012

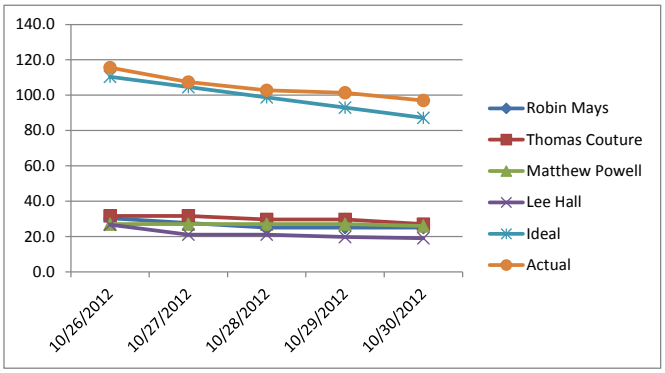
Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	
Date:	10/29/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Design Crawler 2
Date:	10/30/2012	Result:	
Hours Worked:	2	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	10/24/2012	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

Name	Requirements			Design / Prototype			Development /			Integrate / Test			Documentation			Totals		
	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

Name	Crawler 1		Crawler 2		Report 1		Report 2		Report 3		Report 4		Report 5		Report 6		Crawler 3	
	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

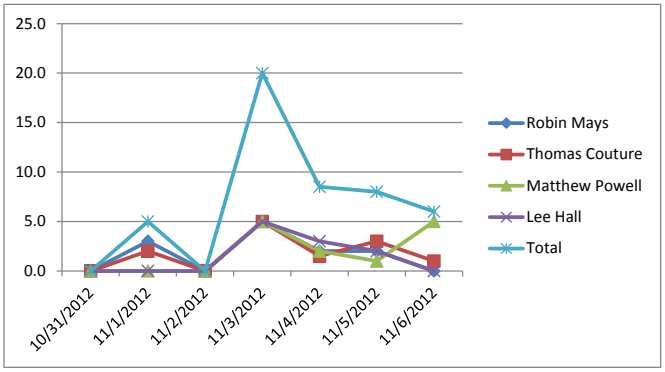
Project Name:	Forager - Group 4
Member:	Matthew Powell
Week Ending:	6-Nov-2012

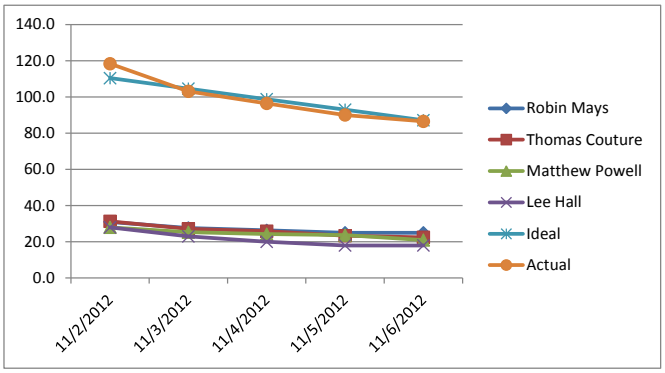
Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	Sprint Report
Date:	11/5/2012	Result:	Post Mortem Analysis
Hours Worked:	1	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Documentation
Date:	11/6/2012	Result:	Post Mortem, Test Plan, Screenshots, Power Point
Hours Worked:	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

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:	11/20/2012
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 added features of the product working. We plan to include sorting, printing, and comparing of the reports. We plan to be able to start a new scan of something previously scanned. Features to limit the scan based on certain criteria. Ability to pause and stop the scan.

		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	2	20	5	Development	7	15
Crawler 2	Record Crawler Results	1	20	0	Completed	0	15
Report 1	Show Scan Results	2	30	4	Development	3	32.5
Report 2	Crawler Interaction	1	20		Completed	0	9
Report 3	Sort Report	2	10	10	Development	9	9
Report 4	Print Report	2	10	10	Unstarted	0	0
Report 5	Multiple Report Changes	2	6	6	Development	2	2
Report 6	Live Report	2	4	4	Unstarted	0	0
Crawler 3	Runtime Limit	2	12	12	Development	0.5	0.5
Crawler 4	Error Check	2	10	10	Development	5.5	5.5
SSL	Secure Check	2	8	8	Development	0.5	0.5
Report 7	Subdomain Sort	2	14	14	Development	0.5	0.5
Login	User Login	2	2	1	Completed	1.5	2.5
Crawler 5	Pause Scan	2	4	4	Development	2.5	2.5
Crawler 6	Timer	2	6	2	Development	0.5	0.5
Crawler 7	Stopwatch	2	2	2	Development	0.5	0.5
		Planned Total	178	92	**Actual Total**	33	95.5

SWE 3613 Status Report

Date	11/20/2012
Members	Robin Mays, Thomas Couture, Matthew Powell, Lee Hall
Project	Forager - Group 4

HOURS BY DEVELOPMENT ACTIVITY

Name	Requirements			Design / Prototype			Development / Code			Integrate / Test			Documentation			Totals		
	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	1	0		4	0	1	14	1	5	5	0	0	4	1	1.5	28	2	7.5
Thomas Couture	1	0		4	0	2	14	1	6	5	0	0	4	1	2	28	2	10
Matthew Powell	1	0		4	0	3	14	0	2	5	4	4	4	0.5	0.5	28	4.5	9.5
Lee Hall	1	0		4	0	1	14	2	7	5	0	2	4	2.5	2.5	28	4.5	12.5
Totals:	4	0	0	16	0	7	56	4	20	20	4	6	16	5	6.5	112	13	39.5

HOURS BY USER STORY

Name	Crawler 1		Crawler 2		Report 1		Report 2		Report 3		Report 4		Report 5		Report 6		Crawler 3	
	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	1			1	5								
Thomas Couture					0	2			0	4			1	1				
Matthew Powell																	0.5	0.5
Lee Hall	0	7											0	1				
Totals:	0	7	0	0	0	3	0	0	1	9	0	0	1	2	0	0	0.5	0.5

	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																	1	6
Thomas Couture							0	1									1	8
Matthew Powell	0.5	5.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5			4	9
Lee Hall									2	2							2	10
Totals:	0.5	5.5	0.5	0.5	0.5	0.5	0.5	1.5	2.5	2.5	0.5	0.5	0.5	0.5	0	0	8	33

Accomplishments since last status report:

There has been progress made on interacting with the running webcrawler, and presenting the data in a visually pleasing way. The test plan is much more robust, and there has been significant copy editing on the final report.

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:

Make up for the missed meeting (that put us behind 20 hours). Finish up all of the Use Cases. Test functionality and prepare for final presentation.

Notes:

Thomas had family commitments which led to him being unavaliable for the weekend. Meeting planned for Friday, but was canceled because of sickness and unforseen conflicts.

SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Robin Mays
Week Ending:	20-Nov-2012

Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	UI Touch-up Work for Sorting
Date:	11/19/2012	Result:	
Hours Worked:	1	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	
Date:	11/20/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	Documentation
Date:	11/14/2012	Result:	
Hours Worked:	1	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	
Date:	11/15/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	11/16/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	
Date:	11/17/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	
Date:	11/18/2012	Result:	
Hours Worked:	0	Problems encountered:	

SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Thomas Couture
Week Ending:	20-Nov-2012

Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	CSS Styling for tables/UI
Date:	11/19/2012	Result:	Progres on Style
Hours Worked:	1	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Documentation
Date:	11/20/2012	Result:	Weekly Status Report
Hours Worked:	1	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	11/14/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	
Date:	11/15/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	11/16/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	
Date:	11/17/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	
Date:	11/18/2012	Result:	
Hours Worked:	0	Problems encountered:	

SWE 3613 Timesheet

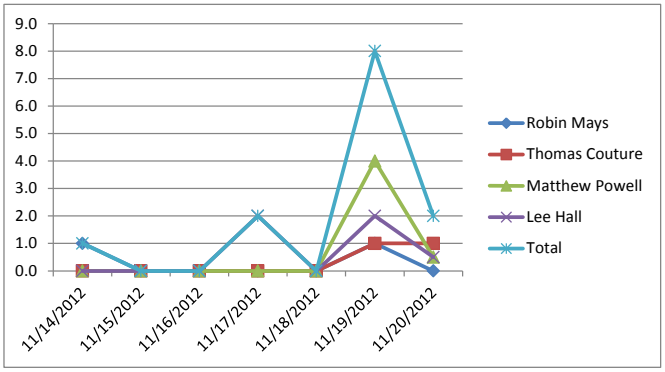
Project Name:	Forager - Group 4
Member:	Matthew Powell
Week Ending:	20-Nov-2012

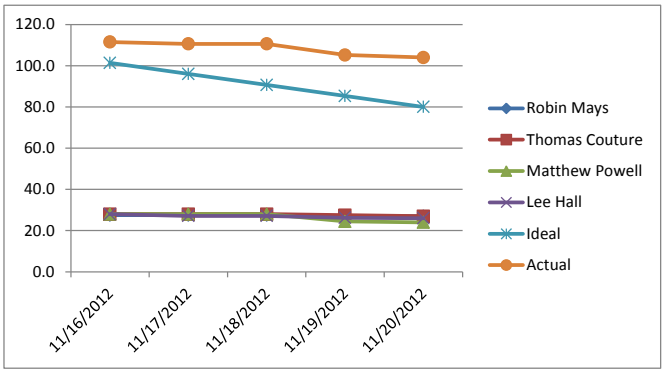
Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	Test Plan
Date:	11/19/2012	Result:	List of functionality that can be used to test the product
Hours Worked:	4	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Status Report
Date:	11/20/2012	Result:	Status Report/Design Implementation
Hours Worked:	0.5	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	11/14/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	
Date:	11/15/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	11/16/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	
Date:	11/17/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	
Date:	11/18/2012	Result:	
Hours Worked:	0	Problems encountered:	

SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Lee Hall
Week Ending:	20-Nov-2012

Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	Worked on signaling for the crawler
Date:	11/19/2012	Result:	The crawler accepts signals properly
Hours Worked:	2	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Status Report
Date:	11/20/2012	Result:	Worked on status report and disccused design
Hours Worked:	0.5	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	11/14/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	
Date:	11/15/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	11/16/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	Documentation
Date:	11/17/2012	Result:	Current version of the report has been copy edited
Hours Worked:	2	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	
Date:	11/18/2012	Result:	
Hours Worked:	0	Problems encountered:	


9.0
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
11/14/2012
11/15/2012
11/16/2012
11/17/2012
11/18/2012
11/19/2012
11/20/2012
Robin Mays
Thomas Couture
Matthew Powell
Lee Hall
Total


120.0
100.0
80.0
60.0
40.0
20.0
0.0
11/16/2012
11/17/2012
11/18/2012
11/19/2012
11/20/2012
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:	11/27/2012
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 added features of the product working. We plan to include sorting, printing, and comparing of the reports. We plan to be able to start a new scan of something previously scanned. Features to limit the scan based on certain criteria. Ability to pause and stop the scan.

		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	2	20	5	Completed	7	15
Crawler 2	Record Crawler Results	1	20	0	Completed	0	15
Report 1	Show Scan Results	2	30	4	Testing	4	33.5
Report 2	Crawler Interaction	1	20		Completed	0	9
Report 3	Sort Report	2	10	10	Completed	10	10
Report 4	Print Report	2	10	10	Development	7	7
Report 5	Multiple Report Changes	2	6	6	Testing	8	8
Report 6	Live Report	2	4	4	Development	7	7
Crawler 3	Runtime Limit	2	12	12	Testing	9.5	9.5
Crawler 4	Error Check	2	10	10	Testing	10.5	10.5
SSL	Secure Check	2	8	8	Testing	0.5	0.5
Report 7	Subdomain Sort	2	14	14	Development	9.5	9.5
Login	User Login	2	2	1	Completed	1.5	2.5
Crawler 5	Pause Scan	2	4	4	Testing	11.5	11.5
Crawler 6	Timer	2	6	2	Testing	1	1
Crawler 7	Stopwatch	2	2	2	Testing	1	1
		Planned Total	178	92	**Actual Total**	88	150.5

SWE 3613 Status Report

Date	11/27/2012
Members	Robin Mays, Thomas Couture, Matthew Powell, Lee Hall
Project	Forager - Group 4

HOURS BY DEVELOPMENT ACTIVITY

Name	Requirements			Design / Prototype			Development / Code			Integrate / Test			Documentation			Totals		
	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	1	0	0	4	1	2	14	9	14	5	2	2	4	2	1.5	28	14	19.5
Thomas Couture	1	0	0	4	1	3	14	10	16	5	2	2	4	2	4	28	15	25
Matthew Powell	1	0	0	4	1	3	14	11	13	5	1	5	4	2	2.5	28	15	23.5
Lee Hall	1	0	0	4	2	3	14	12	19	5	4	6	4	0	2.5	28	18	30.5
Totals:	4	0	0	16	5	11	56	42	62	20	9	15	16	6	10.5	112	62	98.5

HOURS BY USER STORY

Name	Crawler 1		Crawler 2		Report 1		Report 2		Report 3		Report 4		Report 5		Report 6		Crawler 3	
	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	2			1	6	4	4	1	1				
Thomas Couture					0	2			0	4	3	3	4	6				
Matthew Powell																	7	7.5
Lee Hall	0	7											0	1	7	7	2	2
Totals:	0	7	0	0	0	4	0	0	1	10	7	7	5	8	7	7	9	9.5

	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					4	4					0.5	0.5	0.5	0.5			11	18
Thomas Couture					5	5	0	1									12	21
Matthew Powell	5	10.5	0	0.5	0	0.5	0	0.5	0	0.5	0	0.5	0	0.5			12	21
Lee Hall									9	11							18	28
Totals:	5	10.5	0	0.5	9	9.5	0	1.5	9	11.5	0.5	1	0.5	1	0	0	53	88

Accomplishments since last status report:

Almost all functionality is in the testing phase. Scans can be run from the web UI as well as paused, resumed and stopped. You can compare multiple reports, start scans based on old reports, limit the runtime of new scans.

Obstacles encountered since last status report:

Printing problem in conjunction with the way we are displaying our table. Slow load time on the live scan has lead us to trying alternatives to get a feasable live runtime scan.

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:

Finish up any testing, and complete our Final Report and Power Point Presentation.

Notes:

SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Robin Mays
Week Ending:	27-Nov-2012

Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	Worked on completing UC Report 4,5,7
Date:	11/26/2012	Result:	Completed Reoprt 4 and 5, Progress towards 7
Hours Worked:	6	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Final Report Doc/Meeting Minutes
Date:	11/27/2012	Result:	
Hours Worked:	2	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	11/21/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	
Date:	11/22/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	11/23/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	
Date:	11/24/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	Worked on completing UC Report 3,4,5,7
Date:	11/25/2012	Result:	Completed Reoprt 3, Progress towards 4 and 5
Hours Worked:	6	Problems encountered:	

SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Thomas Couture
Week Ending:	27-Nov-2012

Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	Worked on completing UC Report 4,5,7
Date:	11/26/2012	Result:	Completed Reoprt 4 and 5, Progress towards 7
Hours Worked:	7	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Status report rework, new report, and burndowns
Date:	11/27/2012	Result:	Completed reports/charts
Hours Worked:	2	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	11/21/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	
Date:	11/22/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	11/23/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	
Date:	11/24/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	Worked on completing UC Report 3,4,5,7
Date:	11/25/2012	Result:	Completed Reoprt 3, Progress towards 4 and 5
Hours Worked:	6	Problems encountered:	

SWE 3613 Timesheet

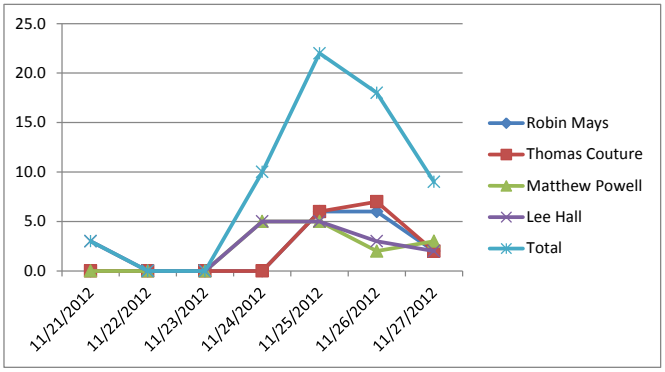
Project Name:	Forager - Group 4
Member:	Matthew Powell
Week Ending:	27-Nov-2012

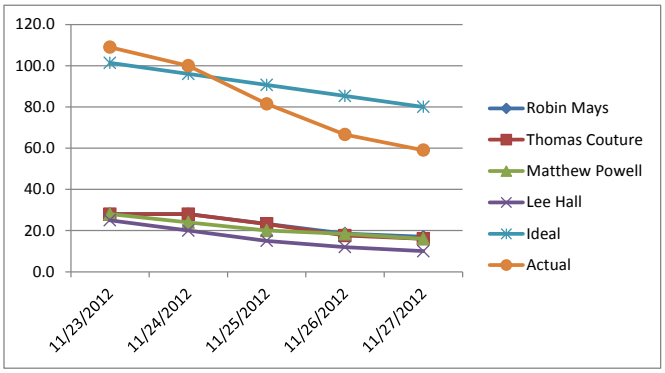
Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	Finished up UC Crawler 3
Date:	11/26/2012	Result:	Runtime finctionality works
Hours Worked:	2	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Final Doc/Testing
Date:	11/27/2012	Result:	
Hours Worked:	3	Problems encountered:	
Day:	***Wednesday***	Task(s) performed:	
Date:	11/21/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	
Date:	11/22/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	11/23/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	Worked on Crawler 4
Date:	11/24/2012	Result:	Completed Running scans on previous reports
Hours Worked:	5	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	Worked on Crawler 3
Date:	11/25/2012	Result:	Progress towards completing UC
Hours Worked:	5	Problems encountered:	

SWE 3613 Timesheet

Project Name:	Forager - Group 4
Member:	Lee Hall
Week Ending:	27-Nov-2012

Team Member Work Summary			
Day:	***Monday***	Task(s) performed:	Testing
Date:	11/26/2012	Result:	Fixed Web UI for Scan Controls
Hours Worked:	3	Problems encountered:	
Day:	***Tuesday***	Task(s) performed:	Report 6
Date:	11/27/2012	Result:	Changed logs to splint into chunks serverside
Hours Worked:	2	Problems encountered:	The DataTables module does not deal with this particularly well. This will probably be reverted.
Day:	***Wednesday***	Task(s) performed:	Crawler 5
Date:	11/21/2012	Result:	Pause functionality working (crawler side only)
Hours Worked:	3	Problems encountered:	
Day:	***Thursday***	Task(s) performed:	
Date:	11/22/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Friday***	Task(s) performed:	
Date:	11/23/2012	Result:	
Hours Worked:	0	Problems encountered:	
Day:	***Saturday***	Task(s) performed:	Worked on Crawler 4 and 5
Date:	11/24/2012	Result:	Php can send signals to pause and resume crawler
Hours Worked:	5	Problems encountered:	
Day:	***Sunday***	Task(s) performed:	Worked on Report 6 and Testing
Date:	11/25/2012	Result:	Bugs fixed in interface
Hours Worked:	5	Problems encountered:	Getting a constant stream of data is difficult, as loading large JSON takes a long time.


25.0
20.0
15.0
10.0
5.0
0.0
11/21/2012
11/22/2012
11/23/2012
11/24/2012
11/25/2012
11/26/2012
11/27/2012
Robin Mays
Thomas Couture
Matthew Powell
Lee Hall
Total


120.0
100.0
80.0
60.0
40.0
20.0
0.0
11/23/2012
11/24/2012
11/25/2012
11/26/2012
11/27/2012
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.

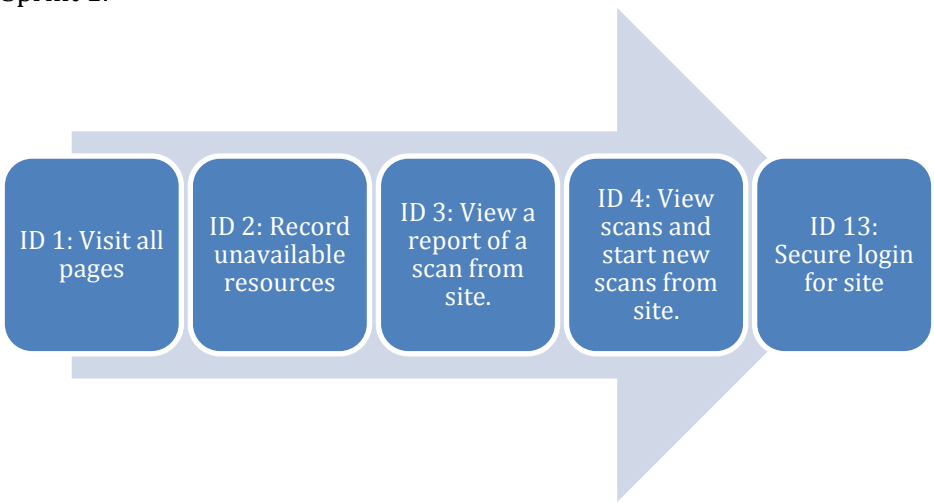
Dealing with javascript could make me so angry I stick forks in my eyes.
This might make typing difficult, and I should probably keep a bottle of medication uncorked and nearby to prevent such an occurrence.

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:



Sprint 2:

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

Group 4

Forager Meeting Minutes : Sprint Planning

November 7, 2012

Opening:
The meeting of ***Group4*** was called to order at 6:00 pm on November 7, 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**

We had two things carry over from the sprint 1, as they did not work as completely intended.

1. Rework on the Crawler as it is not picking up every link that it should. (Use Case Crawler-1)

2. Complete the table so you can view all reports and not just first 100, as well as some styling for the table. (Use Case Report-1)

Currently those two are at the top of the list, as the crawler and report viewer are of high priority.

D. **New Business**
Review Sprint 2 Backlog.

3. Sorting of the report fields (UC Report-3)

4. Printer Friendly Report (UC Report-4)

5. Report Comparison shows changes between two reports (UC Report-5)

6. Showing a report from a live scan. (UC Report-6)

7. Scan limitations by Runtime. (UC Crawler-3)

8. Rerun a previous scan and compare the two reports. (UC Crawler-4)

9. Distinguishing that a page is a secure page and not an error. (UC SSL)

10. Sort the report based off of sub domain. This specific sort was lower priority which is why it is not with the other sorting methods. (UC Report-7)

11. Login. The login is functioning but is lacking a UI at the moment.(UC Login)

12. Pause a running scan. (UC Crawler-5)

13. Running a predetermined scan set up by the user by time and date. (UC Crawler-6)

14. Loadtime determined and added to the report. (UC Crawler-7)

Questions for the customer:

Are these items satisfactory for the Product Owner/Customer?
Are there any items the customer does not want?
Are there any priority changes that the Product Owner wants?
Having seen our presentation, are there any things that you (in any role you play) did not like or want us to change/improve upon?
Was the UI more acceptable than the previous project?(You told us our UI was bad before)

E. Agenda for Next Meeting

Review answers to above question from the customer and update sprint 2 backlog if necessary.

Adjournment:
This meeting was adjourned at 7:30pm by The Team. The next general meeting will be at 6:00pm on Novemver 13, 2012 at SPSU.

Minutes submitted by: Robin Mays

Approved by: Matthew Powell

Group 4

Forager Meeting Minutes : Sprint Planning

November 19, 2012

Opening:
The meeting of ***Group4*** was called to order at 6:00 pm on November 7, 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**

We had two things carry over from the sprint 1, as they did not work as completely intended.

1. Review fixes to the web crawler including login functionality, UI design, and table sorting: Robin, Lee, Matthew, and Thomas

2. Reformat test plans and add test plans for new features: Matthew

Currently those two are at the top of the list, as the crawler and report viewer are of high priority.

D. **New Business**
Review Sprint 2 Backlog.

3. Sorting of the report fields (UC Report-3): Not yet complete/Robin and Thomas

4. Printer Friendly Report (UC Report-4): Not yet complete/Robin and Thomas

5. Report Comparison shows changes between two reports (UC Report-5): Not yet compelte: Robin and Thomas

6. Showing a report from a live scan. (UC Report-6): Not yet complete: Lee and Matthew

7. Scan limitations by Runtime. (UC Crawler-3): Complete/Lee and Matthew

8. Rerun a previous scan and compare the two reports. (UC Crawler-4): Not yet complete/Lee and Matthew(backend for #5)

9. Distinguishing that a page is a secure page and not an error. (UC SSL): Complete/Lee and Matthew

10. Sort the report based off of sub domain. This specific sort was lower priority which is why it is not with the other sorting methods. (UC Report-7): Not yet Complete/Robin and Thomas

11. Login. The login is functioning but is lacking a UI at the moment.(UC Login): Complete/Lee

12. Pause a running scan. (UC Crawler-5): Not yet complete/Lee and Matthew

13. Running a predetermined scan set up by the user by time and date. (UC Crawler-6): Not Yet Complete/Lee and Matthew

14. Loadtime determined and added to the report. (UC Crawler-7): Not yet complete/Lee and Matthew

E. Agenda for Next Meeting

Review backlog and discuss any impediment.

Adjournment:
This meeting was adjourned at 7:30pm by The Team. The next general meeting will be at 6:00pm on Novemver 22, 2012 at SPSU.

Minutes submitted by: Thomas Couture

Approved by: Matthew Powell

Group 4

Forager Meeting Minutes : Sprint Planning

November 22, 2012

Opening:
The meeting of ***Group4*** was called to order at 6:00 pm on November 22, 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**

Teams members update about the status of their tasks.

At this point we have the following features working:

Login: Lee

Table sorting: Robin and Thomas

Updated UI designed: Robin and Thomase

Show a report from a live scan: Lee and Matthew

Pages are secure: Lee and Matthew

Display Load time: Lee and Matthew

D. **New Business**
Review Sprint 2 Backlog.

3. Sorting of the report fields (UC Report-3): Complete/Robin and Thomas

4. Printer Friendly Report (UC Report-4): Not yet complete/Robin and Thomas

5. Report Comparison shows changes between two reports (UC Report-5): Not yet compelte: Robin and Thomas

6. Showing a report from a live scan. (UC Report-6): Complete: Lee and Matthew

7. Scan limitations by Runtime. (UC Crawler-3): Complete/Lee and Matthew

8. Rerun a previous scan and compare the two reports. (UC Crawler-4): Not yet complete/Lee and Matthew(backend for #5)

9. Distinguishing that a page is a secure page and not an error. (UC SSL): Complete/Lee and Matthew

10. Sort the report based off of sub domain. This specific sort was lower priority which is why it is not with the other sorting methods. (UC Report-7): Complete/Robin and Thomas

11. Login. The login is functioning but is lacking a UI at the moment.(UC Login): Complete/Lee

12. Pause a running scan. (UC Crawler-5): Not yet complete/Lee and Matthew

13. Running a predetermined scan set up by the user by time and date. (UC Crawler-6): Not Yet Complete/Lee and Matthew

14. Loadtime determined and added to the report. (UC Crawler-7): Complete/Lee and Matthew

E. Agenda for Next Meeting

Review spring backlog.

Adjournment:
This meeting was adjourned at 7:30pm by The Team. The next general meeting will be at 2:30pm on Novemver 25, 2012 at SPSU.

Minutes submitted by: Robin Mays

Approved by: Matthew Powell

Group 4

Forager Meeting Minutes : Sprint Planning

November 25, 2012

Opening:
The meeting of ***Group4*** was called to order at 2:30 pm on November 25, 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**

Coding session and code reviews.

During this coding session it is important that we implement most of the remaining features. They are:

4. Printer Friendly Report (UC Report-4): Not yet complete/Robin and Thomas

5. Report Comparison shows changes between two reports (UC Report-5): Not yet compelte: Robin and Thomas

8. Rerun a previous scan and compare the two reports. (UC Crawler-4): Not yet complete/Lee and Matthew(backend for #5)

12. Pause a running scan. (UC Crawler-5): Not yet complete/Lee and Matthew

13. Running a predetermined scan set up by the user by time and date. (UC Crawler-6): Not Yet Complete/Lee and Matthew

D. **New Business**

No new business. This is only a coding session to complete the assigned tasks.

E. Agenda for Next Meeting

Review the status of assigned tasks and prepare report and powerpoints for demo.

Adjournment:
This meeting was adjourned at 7:30pm by The Team. The next general meeting will be at 6:00pm on Novemver 27, 2012 at SPSU.

Minutes submitted by: Lee Hall

Approved by: Robin Mays

Group 4

Forager Meeting Minutes : Sprint Planning

November 27, 2012

Opening:
The meeting of ***Group4*** was called to order at 6:00 pm on November 27, 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**

Team members update of status of their tasks:

At this point, the following features have been completed since the last meeting:

Printer friendly report: Thomas and Robin

Report Comparison: Thomas and Robin

Rerun scans to compare reports: Lee and Matthew

Pause a running scan: Lee and Matthew

D. **New Business**
Review Sprint 2 Backlog.

3. Sorting of the report fields (UC Report-3): Complete/Robin and Thomas

4. Printer Friendly Report (UC Report-4): Complete/Robin and Thomas

5. Report Comparison shows changes between two reports (UC Report-5): Complete: Robin and Thomas

6. Showing a report from a live scan. (UC Report-6): Complete: Lee and Matthew

7. Scan limitations by Runtime. (UC Crawler-3): Complete/Lee and Matthew

8. Rerun a previous scan and compare the two reports. (UC Crawler-4): Complete/Lee and Matthew(backend for #5)

9. Distinguishing that a page is a secure page and not an error. (UC SSL): Complete/Lee and Matthew

10. Sort the report based off of sub domain. This specific sort was lower priority which is why it is not with the other sorting methods. (UC Report-7): Complete/Robin and Thomas

11. Login. The login is functioning but is lacking a UI at the moment.(UC Login): Complete/Lee

12. Pause a running scan. (UC Crawler-5): Complete/Lee and Matthew

13. Running a predetermined scan set up by the user by time and date. (UC Crawler-6): Not Yet Complete/Lee and Matthew

14. Loadtime determined and added to the report. (UC Crawler-7): Complete/Lee and Matthew

E. Agenda for Next Meeting

Review sprint backlog and practice demo.

Adjournment:
This meeting was adjourned at 7:30pm by The Team. The next general meeting will be at 6:00pm on November 29, 2012 at SPSU.

Minutes submitted by: Matthew Powell

Approved by: Lee Hall

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.';

DROP TABLE IF EXISTS http_responses CASCADE;
CREATE TABLE http_responses (
        http_response           INTEGER NOT NULL,
        response_name           varchar
);

COMMENT ON TABLE http_responses IS 'Pretty lables for HTTP Response codes. '
        'Responses >0 are defined in RFC 2616, 0< are error conditions '
        'reported by the browser. We do not want inserts into the resources '
        'table to fail if there is no label, so this is NOT treated as a formal '
        'key.';

INSERT INTO http_responses(http_response, response_name) VALUES
        (-1, 'Unfetched resource'),
        (-2, 'Timeout'),
        (-3, 'Unknown error retrieving page'),
        (-4, 'Library failure retrieving page'),
        (200, 'OK'),
        (301, 'Moved Permanently'),
        (302, 'Found'),
        (400, 'Bad Request'),
        (401, 'Unauthorized'),
```

```
        (403, 'Forbidden'),
        (404, 'Not Found'),
        (410, 'Gone'),
        (415, 'Unsupported Media Type'),
        (500, 'Internal Server Error'),
        (503, 'Service Unavailable'),
        (504, 'Gateway Timeout'),
        (300, 'Multiple Choices'),
        (999, 'Failure reading HTTP specification');


COMMIT;
```

2 Crawler

```
#! /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."""

    DOMAIN_RESTRICTION=""
    @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]

    @staticmethod
    def set_domain(url):
        resource.DOMAIN_RESTRICTION=url

    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)

        if (resource.DOMAIN_RESTRICTION is None):
            msg="Domain restriction is unset. " + \
                "You are scanning the entire internet."
            logging.critical(msg)

    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, timeout=5)
        except requests.Timeout:
            logging.info("Timed out fetching page {0}".format(self.url))
            self.visited=True
            self.response_code=-2
            #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
        except Exception as e:
            msg="Unknown, exception {0}. This is a bug in 'requests'."
            logging.info("".format(e))
```

```
            self.response_code=-4
            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

        # Don't parse contents of outside URLs
        if (not self.domain.endswith(resource.DOMAIN_RESTRICTION)):
            msg="Not parsing contents of URL \"{0}\", outside of {1}"
            logging.debug(msg.format(self.url, resource.DOMAIN_RESTRICTION))
            return

        # Catch null content-type header
        if (not r.headers.get('content-type')):
            msg="No content-type header on URL \"{0}\""
            logging.debug(msg.format(self.url))
            return

        # Only try to parse html content
        if (not r.headers.get('content-type').startswith('text/html')):
            msg="Not parsing contents of non-html URL \"{0}\""
            logging.debug(msg.format(self.url))
            return

        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
            if (attr[0:7] == "mailto:"):
                logging.info("Ignored mailto link \"{0}\" on {1}".format(
```

```
                attr, self.url))
            continue
        if (attr[0:11] == "javascript:"):
            logging.info("Ignored javascript \"{0}\" data on {1}".format(
                attr, self.url))
            continue
        if (attr[0:8] == "callto:"):
            logging.info("Ignored bogus URI \"{0}\" data on {1}".format(
                attr, self.url))
            continue
        if (attr[0:4] == "tel:"):
            logging.info("Ignored bogus URI \"{0}\" data on {1}".format(
                attr, self.url))
            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[0:5] == "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
    # We can't deal with the last two URI schemes, but we should
    # make sure we can record them anyway (especially file:// as this is
    # always wrong.
    if (url.startswith('http://') or
            url.startswith('https://') or
            url.startswith('mms://') or
            url.startswith('file://')):
        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/", "1")
    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
import argparse
import traceback


DEBUG=False
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, foreground=False):
        global LOGFILE
        self.LOGFILE=LOGFILE

        if (foreground):
            self.LOGFILE=None

        if (DEBUG):
            log_level=logging.DEBUG
        else:
            log_level=logging.INFO
        try:
            logging.basicConfig(level=log_level, filename=self.LOGFILE)
        except IOError as e:
            logging.basicConfig(level=log_level)
            logging.warn("Error {0} encountered opening {1}".format(
```

```
            e,LOGFILE))
        foreground=True

    if (DEBUG):
        logging.debug("Debugging enabled.")

    signal.signal(signal.SIGINT,  self.sig_close)
    signal.signal(signal.SIGTERM, self.sig_close)
    signal.signal(signal.SIGALRM, self.sig_close)
    signal.signal(signal.SIGCONT, self.sig_cont)
    signal.signal(signal.SIGUSR1, self.sig_pause)
    signal.signal(signal.SIGUSR2, self.sig_query)

    if (not foreground):
        logging.debug("Daemonizing process")
        self.daemonize()
    else:
        msg="Leaving process in foreground, output redirected to console."
        logging.debug(msg)

    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()
        logging.warn("Recording times and closing database connections.")
    else:
        logging.warn("Crawler exited before connecting to the database.")

def sig_close(self,sig, frame):
    if (sig == signal.SIGINT):
        logging.warn("Caught SIGINT. Exiting.")
    elif (sig == signal.SIGTERM):
        logging.warn("Caught SIGTERM. Exiting.")
    elif (sig == signal.SIGALRM):
        logging.warn("Allotted time has expired. Exiting.")
    self.dbclose()
    sys.exit(0)

def sig_query(self,sig, frame):
        logging.warn("Caught Liveness query.")
        return

def sig_pause(self,sig, frame):
    logging.warn("Caught pause signal.")
    print(signal.sigwait((signal.SIGCONT,signal.SIGINT,signal.SIGTERM)))

def sig_cont(self,sig, frame):
    logging.warn("Caught continue signal.")

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))
        resource.set_domain(DOMAIN)
        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):
            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
                pending.append(child_url)
                resource_list[child_url]=new_resource
        logging.info("All queued items have been scanned.");

    def time_handle(self, sig, frame):
        print ("time is up ending webcrall")
        sys.exit(1)#kill webcraller

    def recrawl(self,id):
```

```
        logging.info("Starting difference scan at {0}.".format(id))
        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,))#creates the new scan ID
        scan_row=self.cur.fetchone()
        self.scan_id=scan_row[0]

        create_list_sql="""SELECT url FROM resources
            WHERE scan_id = %s AND http_response != 200 ;"""
        self.cur.execute(create_list_sql,(id,))#retrieves old scan data

        scan_results=self.cur.fetchall()

        for x in scan_results:
            cur_resource=resource(x[0],self.scan_id)
            cur_resource.fetch()
            cur_resource.Sql_Call(self.cur)

        logging.info("All queued items have been scanned.");

def main():
    parser = argparse.ArgumentParser(description='Webcrawler')
    parser.add_argument('-v', action='store_true', dest="verbose",
        help="Enable debugging")
    parser.add_argument('-d', action='store', dest="diff", type=int,
        help="Recheck broken links from scan with the given id")
    parser.add_argument('-f', action='store_true', dest="foreground",
        help="Run in foreground and do not detatch")
    parser.add_argument('-r', action='append', dest="response", type=int,
        help="Specifies what kind of links to recheck (You may list as " +\
        "many you like, but you must specify -d)")
    parser.add_argument('-R', action='append', dest="not_response", type=int,
        help="Specifies what kind of links you DO NOT want " + \
        "to recheck (You may list as many you like, but you must specify -d)")
    parser.add_argument('-t', action='store', dest="time", type=int,
        help="Time scan funtion call")
    args=parser.parse_args(sys.argv[1:])

    if (args.verbose):
        DEBUG=True

    if (args.not_response and args.response):
        print("Specifying a set and a negation to check is undefined")
        sys.exit(1)
    if ((args.not_response or args.response) and not args.diff):
        msg="Specifying a set of results and no report makes no sense"
        print(msg)
        sys.exit(1)
    if (args.diff and not args.response and not args.not_response):
        args.response=(404,)

    try:
```

```
        c=crawler(args.foreground)
        if (args.time):
            msg="Halting scan in {0} seconds".format(args.time)
            logging.warn(msg)
            signal.ITIMER_REAL
            signal.alarm(args.time)#starts alarm that goes off in time seconds

        if (args.diff):
            msg="Rechecking scan {0}".format(args.diff)
            logging.warn(msg)
            c.recrawl(args.diff)
        else:
            c.crawl(START_PAGE)
        c.dbclose()
    except Exception as e:
        logging.critical("Something exploded: {0}".format(e))
        logging.critical("Traceback: {0}".format(traceback.format_exc()))

if __name__ == "__main__":
    main()
```

3 Web Interface

```
<?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");
?>

<!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/conf.php');
?>

<script src="/javascript/jquery/jquery.js"></script>
<script src="js/jquery.dataTables.min.js"></script>
<head>
          <meta http-equiv="Content-Type"
 content="text/html; charset=iso-8859-1">
  <title>Group 4</title>
  <link href="css/style.css" rel="stylesheet" type="text/css">
  <link href="css/buttons.css" rel="stylesheet" type="text/css">

  <script type="text/javascript">
//Holds the ID of the scan currently being displayed.
var current_scan;
var current_data;
```

```
var current_table;
var compare_one;
var compare;
var refresh;

function printpage()
{
window.print();
}

function fetch_table(url){
        $.getJSON(
            url,
            function(data) {
                if (! data['valid_session']){
                    window.location.href = "login.php";
                                        return;
                }
                                //This autorefresh has been preempted
                                if (url != current_table){
                                        alert(url + " != " + current_table);
                                        return;
                                }

                cur_table=$('#display_table').dataTable({
                    "bDestroy": true,
                    "sPaginationType": "full_numbers",
                    "aoColumns": data['aoColumns'],
                    "aaData": data['aaData']
                });
                cur_table.fnAdjustColumnSizing();
            }
        );
}

function show_home() {
        window.clearInterval(refresh);
    $('#welcome_div').show();
    $('#message_div').hide();
    $('#data_div').hide();
    $('#controller_div').hide();
}

function refresh_scan(){
        fetch_table('reports_json.php?scan_id=' + current_scan);
}

function show_control() {
    $('#welcome_div').hide();
    $('#message_div').hide();
    $('#data_div').hide();
    $('#controller_div').show();
    $.getJSON(
        'control_json.php?a=status',
```

```
        function(data) {
            if (! data['valid_session']){
                window.location.href = "login.php";
            }
            if( data['scan_id'] > 0){
                current_scan=data['scan_id'];
                $('#data_div').show();
                $('#start-button').hide();
                $('#stop-button').show();
                $('#pause-button').show();
                $('#resume-button').hide();
                $('#controller-params').hide();
                                current_table='reports_json.php?scan_id=' + current_scan;
                fetch_table(current_table);
                                refresh=window.setInterval(refresh_scan, 1000);
            } else {
                compare=3;
                $('#start-button').show();
                $('#stop-button').hide();
                $('#pause-button').hide();
                $('#resume-button').hide();
                $('#controller-params').show();
                                window.clearInterval(refresh);
            }
        }
    );
}

function show_comparison_list(){
        window.clearInterval(refresh);
    $('#welcome_div').hide();
    $('#data_div').show();
    $('#controller_div').show();
    $('#message_div').hide();
    compare = 3;// 3 = Starting comparison scan
                // 2 = no comparing, 1 = in compare state no links clicked,
                // 0 = In compare state one link clicked!!
        current_table='scans_json.php';
    fetch_table(current_table);
}

function show_list(){
        window.clearInterval(refresh);
    $('#welcome_div').hide();
    $('#data_div').show();
    $('#controller_div').hide();
    $('#message_div').hide();
    compare = 2;// 3 = Starting comparison scan
                // 2 = no comparing, 1 = in compare state no links clicked,
                // 0 = In compare state one link clicked!!
        current_table='scans_json.php';
    fetch_table(current_table);
}
function compare_list(){
```

```
        window.clearInterval(refresh);
    $('#welcome_div').hide();
        $('#data_div').show();
    $('#message_div').show();
    $('#controller_div').hide();

        current_table='scans_json.php';
    fetch_table(current_table);

        $('#message_div').text("Select the first scan to compare");
        compare = 1; // 1 = in compare state, 0 = Not in compare state!!

}

function select_scanId(scan_id)
{
if (compare == 0)
        {
        current_table='compare_json.php?firstId=' + compare_one  + '&secondId=' + scan_id
        fetch_table(current_table);
        }

if(compare == 1)
        {
        compare_one = scan_id;
        $('#message_div').text("Select the second scan to compare");
        compare = 0;
        }

if(compare == 2)
        {
        show_scan(scan_id);
        }

    if(compare==3){
        $('#base_scan').val(scan_id);
    }

}

function show_scan(scan_id){
    $('#welcome_div').hide();
    $('#data_div').show();
        current_table='reports_json.php?scan_id=' + scan_id;
    fetch_table(current_table);
}

function control_scan(action){
    url='control_json.php?a=' + action;
        $('#message_div').text("Select a basis scan.");
    if (action=='start'){
        $('#start-button').hide();
        $('#stop-button').show();
        $('#pause-button').show();
```

```
        $('#resume-button').hide();
        $('#controller-params').hide();
        $('#data_div').show();
        timeout=$('#timeout').val();
        if(timeout){
            if( (+timeout) == parseInt(timeout)){
                url+='&t=' + timeout;
            } else {
                alert("Non-integer timeouts will be ignored.");
                $('#timeout').val("");
            }
        }
        base_scan=$('#base_scan').val();
        if(base_scan){
            if( (+base_scan) == parseInt(base_scan)){
                url+='&d=' + base_scan;
            } else {
                $('#base_scan').val("");
                alert("Non-integer scan ids will be ignored.");
            }
        }
    } else if (action=='stop'){
        compare=3;
        $('#start-button').show();
        $('#stop-button').hide();
        $('#pause-button').hide();
        $('#resume-button').hide();
        $('#controller-params').show();
    } else if (action=='pause'){
        $('#pause-button').hide();
        $('#resume-button').show();
    } else if (action=='resume'){
        $('#pause-button').show();
        $('#resume-button').hide();
    }

    $.getJSON(
        url,
        function(data) {
            if (! data['valid_session']){
                window.location.href = "login.php";
            }
            if (action=='start'){
                current_scan=data['scan_id'];
                                current_table='reports_json.php?scan_id=' + current_scan;
                fetch_table(current_table);
            }
        }
    );
}

  </script>
</head>
<body>
```

```
<div id="header"></div>
<div id="nav">
<ul>
  <li><a href="javascript:show_home();">Home</a></li>
  <li><a id="scan_control" href="javascript:show_control()">
          Control a Scan</a></li>
  <li><a href="javascript:show_list()">View Reports</a></li>
  <li><a href="javascript:compare_list()">Compare Reports</a></li>
  <li><a href="javascript:printpage()">Print</a></li>
</ul>
</div>
<div id="message_div" class="site-content" style="display: none">
</div>
<div id="welcome_div" class="site-content" style="display: block">
        <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 cus
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 t
                <p class="border-1"> </p>
                <h2 class="h-text-2">About us</h2>
                <p class="text-1">Group 4 is made up of professionals with over 20 years o
                <p class="text-1"> </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, for
                </div>
                <div> </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 rep
                        <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>
                                   &
                        </p>
                </div>
        </div>
</div>
```

```
<div id="controller_div" class="site-content" style="display: none">
        <div id="main-controls">
                <a id="resume-button" href="javascript:control_scan('resume');"
                                class="regular">
                        <button class="control-button" name="resume">
                                <img src="images/bluedash.png" alt=""/>
                                Resume
                        </button>
                </a>

                <a id="pause-button" href="javascript:control_scan('pause');"
                                class="regular">
                        <button class="control-button" name="pause">
                                <img src="images/greencheck.jpg" alt=""/>
                                Pause
                        </button>
                </a>

                <a id="start-button" href="javascript:control_scan('start');"
                                class="regular">
                        <button class="control-button" name="pause">
                                <img src="images/redx.jpg" alt=""/>
                                Start
                        </button>
                </a>

                <a id="stop-button" href="javascript:control_scan('stop');"
                                class="negative">
                        <button class="control-button" name="pause">
                                <img src="images/redx.jpg" alt=""/>
                                Stop
                        </button>
                </a>
        </div>
    <div id="controller-params" class="site-content">
        <table>
            <tr>
                <td>Timeout (minutes):</td>
                <td><input type="text" id="timeout"></td>
            </tr>
            <tr>
                <td>Recheck Errors from Scan:</td>
                <td><input type="text" id="base_scan"></td>
                <td>
                    <a href="javascript:show_comparison_list()">
                        (List Scans)</a>
                </td>
            </tr>
        </table>
    </div>
</div>
<div id="data_div" class="site-content" style="display: none">
  <table cellpadding="0" cellspacing="0" border="0" class="display"
          id="display_table" width="100%">
```

```
        <thead></thead>
        <tbody></tbody>
  </table>
  <p class="text-1"> </p>
</div>
<div id="footer">
  <p>@ Copyright 2010. Designed by <a target="_blank"
    href="http://www.htmltemplates.net/">HTML Templates</a></p>
  <!--Yes we know this copyright is here. We left it in to show to you that we
  used a template online, and changed it to our needs. This was to make sure
  that you know that we are not trying to pass this off as 100% our work!!-->
  <ul class="footer-nav">
      <li><a href="javascript:show_home();">Home</a></li>
      <li><a href="javascript:show_list()">View Reports</a></li>
      <li><a href="compare">Compare Reports</a></li>
      <li><a href="extra">Extra</a></li>
  </ul>
</div>
</body>
</html>

<?php
require_once("include/conf.php");

/*
 * login.php
 * -Lee Hall Thu 06 Sep 2012 10:23:45 PM EDT
 * edits by Matthew Powell
 * Allow the user to login
 */
//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.");
}
?>
<!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>Group 4 - Forager</title>
  <link href="css/style.css" rel="stylesheet" type="text/css">
</head>
<body>

<div id="header">
    <h1 id="logo-text"></h1>
</div>

<div id="nav">
    <ul></ul>
</div>
```

```
<div class="site-content">
    <div id="col-left">
        <h1 class="h-text-1">LOGIN</h1>

        <ul class="list-1">
        <form action="login.php" 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>
    </ul>


    <div> </div>


    </div>
</div>

<div id="footer"> </div>

</body>
</html>
```

4 JSON - Report Generation

```
<?php
require_once("include/conf.php");
require_once('include/json_session.php');

$query = "SELECT scan_id,
    DATE_TRUNC('minutes', start_time) AS start_time,
    DATE_TRUNC('minutes', end_time) AS end_time,
    DATE_TRUNC('minutes', end_time - start_time) AS elapsed_time
    FROM scans";
$scans = pg_query($conn, $query);
```

```
$columns=array(
    array("sTitle" => "Scan ID",      "sClass" => "table_num"),
    array("sTitle" => "Start Time",   "sClass" => "table_time"),
    array("sTitle" => "End Time",     "sClass" => "table_time"),
    array("sTitle" => "Run Time",     "sClass" => "table_time")
);

$data = array();
while($row = pg_fetch_array($scans)) {
    $data[]=array(
    "<a href='javascript:select_scanId($row[scan_id]);'>$row[scan_id]</a>",
    $row['start_time'], $row['end_time'], $row['elapsed_time']);
}
pg_free_result($scans);
echo json_encode(
    array (
        "aoColumns" => $columns,
        "aaData" => $data,
        "valid_session" => True
    ));
?>

<?php
require_once("include/conf.php");
require_once("include/json_session.php");

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

$query = "SELECT resource_id, url,
        date_trunc('minutes', start_date) AS start_date, response_time,
        COALESCE(response_name, http_response || ' (Unknown Response)')
            AS http_response
    FROM resources
    LEFT JOIN http_responses USING(http_response)
    WHERE scan_id = $1";
$reports = pg_query_params($conn, $query,array($scan_id));

$columns=array(
    array("sTitle" => "Resource ID",      "sClass" => "table_num"),
    array("sTitle" => "URL",              "sClass" => "table_text"),
    array("sTitle" => "Response Time",    "sClass" => "table_time"),
    array("sTitle" => "HTTP Response",    "sClass" => "table_text")
);

$data = array();
while($row = pg_fetch_array($reports)) {
    $data[]=array($row['resource_id'], $row['url'],
        $row['response_time'], $row['http_response']);
}
```

```
pg_free_result($reports);
echo json_encode(
    array (
        "aoColumns" => $columns,
        "aaData" => $data,
        "valid_session" => true
    ));
?>

<?php
require_once("include/conf.php");
require_once("include/json_session.php");

if(array_key_exists('firstId',$_GET)) {
        $scan_id1 = $_GET['firstId'];
} else {
        die("Please specify a scan_id");
}

if(array_key_exists('secondId',$_GET)) {
        $scan_id2 = $_GET['secondId'];
} else {
        die("Please specify a scan_id");
}

$query = "SELECT url,
        COALESCE(r1.response_name, resource1.http_response ||
                ' (Unknown Response)')
            AS http_response1,
        COALESCE(r2.response_name, resource2.http_response ||
                ' (Unknown Response)')
                AS http_response2,
        resource1.response_time - resource2.response_time as runtime_delta
    FROM (SELECT * FROM resources WHERE scan_id=$1) AS resource1
    FULL JOIN (SELECT * FROM resources WHERE scan_id=$2) AS resource2
        USING(url)
    LEFT JOIN http_responses AS r1 ON(resource1.http_response=r1.http_response)
    LEFT JOIN http_responses AS r2 ON(resource2.http_response=r2.http_response)
    WHERE resource1.http_response != resource2.http_response OR
        resource1.http_response IS NULL OR resource2.http_response IS NULL;";
                //AND resource1.http_response != resource2.http_response";
$reports = pg_query_params($conn, $query,array($scan_id1, $scan_id2));

$columns=array(
    array("sTitle" => "URL",                    "sClass" => "table_text"),
    array("sTitle" => "First Report Response",  "sClass" => "table_text"),
        array("sTitle" => "Second Report Response", "sClass" => "table_text"),
        array("sTitle" => "Run Time Change",        "sClass" => "table_time")
);

$data = array();
while($row = pg_fetch_array($reports)) {
    $data[]=array($row['url'], $row['http_response1'],
        $row['http_response2'], $row['runtime_delta']);
```

```
}
pg_free_result($reports);
echo json_encode(
    array (
        "aoColumns" => $columns,
        "aaData" => $data,
        "valid_session" => true
    ));
?>
```

5 JSON - Crawler Interaction

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

define('SIGBREAK', 2);        //SIGINT
define('SIGPAUSE', 10);       //SIGUSR1
define('SIGALIVE', 12);       //SIGUSR2
define('SIGCONT', 18);

if (!array_key_exists('a', $_GET)){
    die("No action specified.");
}


switch ($_GET['a']){
    case 'start':
        $params=array();
        //Set the timeout
        if (array_key_exists('t', $_GET)){
            $timeout=intval($_GET['t']);
            if ($timeout==0){
                trigger_error("Ignoring illegal timeout value");
            } else {
                $params['t']=$timeout * 60;
            }
        }
        //Start a difference scan
        if (array_key_exists('d', $_GET)){
            $scan_id=intval($_GET['d']);
            if ($scan_id==0){
                trigger_error("Ignoring illegal scan_id value");
            } else {
                $params['d']=$scan_id;
            }
        }
        start($params);
        break;
    case 'stop':
```

```
        stop();
        break;
    case 'pause':
        pause();
        break;
    case 'resume':
        resume();
        break;
    case 'status':
        break;
    default:
        die("No legal action specified");
        break;
}

echo json_encode(get_status());
die();

function resume(){
    signal_running(SIGCONT);
}

function pause(){
    signal_running(SIGPAUSE);
}

function stop(){
    signal_running(SIGBREAK);
}

function start($params){
    $scan_id=get_running();
    if ($scan_id > 0){
        trigger_error("Scan already running with scan_id $scan_id.");
        return;
    }

    $cmd="/usr/local/src/forager/bin/crawler.py";
    foreach ($params as $key => $value){
        $cmd.=" -$key $value";
    }
    trigger_error("Starting scanning process with $cmd");
    exec($cmd);

    $pass=0;
    while ($scan_id < 0 and $pass < 5) {
        $scan_id=get_running();
        $pass++;
        if ($scan_id < 0){
            trigger_error("No scan running on pass $pass.");
        }
        sleep(1);
    }
```

```
    if ($scan_id < 0){
        trigger_error("Failed to start scan.");
    }
}

function signal_running($signal){
    $scan=get_status();
    if ($scan['scan_id'] < 0){
        trigger_error("No scan running.");
        return;
    }

    if (!posix_kill($scan['pid'], $signal)){
        trigger_error("Failed sending signal to scanning process");
    }
    return;
}


function get_status(){
    $scan_id=get_running();
    $scan_sql="SELECT scan_id,pid FROM scans WHERE scan_id=$1;";
    $scan_res=pg_query_params($scan_sql, array($scan_id,));

    if (pg_num_rows($scan_res)==0){
       $pid=0;
    } else {
       $scan_row=pg_fetch_array($scan_res);
       $pid=$scan_row['pid'];
    }

    return
        array(
            'valid_session' => True,
            'scan_id'   => $scan_id,
            'pid'       => $pid
        );
}

function get_running(){
    $procs_sql="SELECT scan_id,pid FROM scans WHERE end_time IS NULL;";
    $clear_sql="UPDATE scans SET end_time=NOW() WHERE scan_id=$1";

    $procs_res=pg_query($procs_sql);
    while($procs_row=pg_fetch_assoc($procs_res)){
        trigger_error("Checking if process $procs_row[pid] is still running.");
        if(runningp($procs_row['pid'])){
            $scan_id=$procs_row['scan_id'];
            pg_free_result($procs_res);
            return $scan_id;
        } else {
            //Cleanup unreaped processes.
            $clear_res=pg_query_params($clear_sql, array($procs_row['scan_id']));
            if (pg_affected_rows($clear_res) != 1){
```

```
                $msg="Error updating status for scan_id: $procs_row[scan_id].";
                trigger_error($msg);
            }
            pg_free_result($clear_res);
        }
    }
    pg_free_result($procs_res);
    return -1;
}

function runningp($pid){
    /* Send SIGUSR2 to a process.
     * False means there's no process with that id running. This can
     * fail weirdly with recycled pids, but as long as we can accurately
     * record when a process dies, there shouldn't be any user facing issues.
     */
    return posix_kill($pid, SIGALIVE);
}

?>
```

6 Library Functions

```
<?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'])){
        echo json_encode(array("valid_session" => False));
        die();
}

?>

<?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');
?>
```

7 Styling

```
/* CSS Document */

body {
        margin:0px;
        padding:0px;
        font-family:Arial, Helvetica, sans-serif;
        font-size:12px;
        color:#464a4f;
        background:url(../images/body-bg.jpg) repeat;

        }

p, h1, h2, h3, h4, h5, h6, ul, li {
        margin:0px;
        padding:0px;
        }

img {
        border:none;
        }
```

```
#header {
        width:929px;
        background:url(../images/heaser-bg.jpg) no-repeat;
        height:221px;
        }

#nav {
        width:929px;
        background:url(../images/nav-bg.jpg) repeat-x;
        height:38px;
        }

#nav ul {
        margin:0px;
        padding:0px;
        list-style:none;
        }

#nav ul li {
        float:left;
        font-family:Arial, Helvetica, sans-serif;
        font-size:14px;
        color:#FFFFFF;
        /*line-height:38px;*/
        padding:3px 50px 3px 50px;
        border-right:1px #00222a solid;
        margin:7px 0px 0px 0px;
        }

#nav ul li a {
        float:left;
        font-family:Arial, Helvetica, sans-serif;
        font-size:14px;
        color:#FFFFFF;
        text-decoration:none;
        }

#nav ul li a:hover {
        float:left;
        font-family:Arial, Helvetica, sans-serif;
        font-size:14px;
        color:#FFFFFF;
        text-decoration:underline;
        }

.site-content {
        width:929px;
        clear:both;
        float: left;
        background:url(../images/site-bg.jpg) repeat-y;
        }
```

```
#col-right {
        width:236px;
        float:left;
        }

#col-left {
        width:623px;
        float:left;
        padding:30px 35px 30px 35px;
        }


#footer {
        width:929px;
        float:left;
        border-top:8px #53666c solid;
        clear:both;
        height:70px;
        }

.h-text-1 {

        font-family:Arial, Helvetica, sans-serif;
        font-size:24px;
        color:#020202;
        font-weight:bold;
        padding:0px 0px 15px 0px;
        }

.h-text-2 {

        font-family:Arial, Helvetica, sans-serif;
        font-size:18px;
        color:#020202;
        font-weight:bold;
        padding:10px 0px 10px 0px;
        }

.h-text-3 {

        font-family:Arial, Helvetica, sans-serif;
        font-size:12px;
        color:#0000ff;
        font-weight:bold;
        padding:0px 0px 10px 15px;
        background:url(../images/arr-2.jpg) no-repeat 0px 3px;
        margin:10px 0px 0px 0px;
        }


.text-1 {
        font-family:Arial, Helvetica, sans-serif;
        font-size:12px;
        color:#464a4f;
```

```
        padding:5px 0px 5px 0px;
        }

.text-2 {
        font-family:Arial, Helvetica, sans-serif;
        font-size:12px;
        color:#020202;
        padding:0px 0px 5px 0px;
        }

ul.list-1 {
        margin:10px 0px 0px 15px;
        padding:0px;
        list-style:none;
        }

ul.list-1 li {
        margin:0px 0px 10px 0px;
        padding:0px 0px 0px 15px;
        background:url(../images/arr-1.jpg) no-repeat 2px 5px;


        }

.border-1 {
        background:url(../images/border-bg.jpg) bottom repeat-x;

        }

ul.footer-nav {
        width:430px;
        float:right;
        margin:20px 0px 0px 0px;
        padding:0px;
        list-style:none;



        }


ul.footer-nav li {
        display:inline;
        font-family:Arial, Helvetica, sans-serif;
        font-size:12px;
        color:#ffffff;
        margin:0px 10px 0px 10px;
        }

ul.footer-nav li a {
        display:inline;
        font-family:Arial, Helvetica, sans-serif;
        font-size:12px;
        color:#ffffff;
```

```
        text-decoration:none;
        }

ul.footer-nav li a:hover {
        text-decoration:underline;
        }

#footer p {
        font-family:Arial, Helvetica, sans-serif;
        font-size:12px;
        color:#ffffff;
        width:300px;
        float:left;
        margin:20px 0px 0px 10px;
        }

#footer a {
        display:inline;
        font-family:Arial, Helvetica, sans-serif;
        font-size:12px;
        color:#ffffff;
        text-decoration:none;
        }

table.display {
    width: "100%";
    clear: both;
        }

table.display td,
table.display th {
    padding: 5px;
    border: 1px solid #fff;
}

table.display.table_num {
    text-align: center;
}

table.display.table_text {
    text-align: left;
}

table.display.table_time {
    text-align: center;
}


        /* Header cells */
table.display thead th {
    text-align: center;
    background: #66a9bd;
}
```

```
table.display th.table_num {
    width: 20px;
}

table.display th.table_text {
    width: 200px;
}

table.display th.table_time {
        width: 100px;
}


        /* Body cells */

table.display tbody td {
    background: #d5eaf0;
}

table.display tbody tr.odd td {
    background: #bcd9e1;
}

table.display td.table_num {
        text-align: center;
}

table.display td.table_text {
        text-align: left;
}

table.display td.table_time {
        text-align: center;
}


/* Footer cells */
table.display tfoot td {
    background: #d7e1c5;
    text-align: center;
    font-weight: bold;
}

.buttons a, .buttons button{
    display:block;
    float:left;
    margin:0 7px 0 0;
    background-color:#f5f5f5;
    border:1px solid #dedede;
    border-top:1px solid #eee;
    border-left:1px solid #eee;

    font-family:"Lucida Grande", Tahoma, Arial, Verdana, sans-serif;
    font-size:12px;
```

```
    line-height:130%;
    text-decoration:none;
    font-weight:bold;
    color:#565656;
    cursor:pointer;
    padding:5px 10px 6px 7px; /* Links */
}
.buttons button{
    width:auto;
    overflow:visible;
    padding:4px 10px 3px 7px; /* IE6 */
}
.buttons button[type]{
    padding:5px 10px 5px 7px; /* Firefox */
    line-height:17px; /* Safari */
}
*:first-child+html button[type]{
    padding:4px 10px 3px 7px; /* IE7 */
}
.buttons button img, .buttons a img{
    margin:0 3px -3px 0 !important;
    padding:0;
    border:none;
    width:16px;
    height:16px;
}

/* STANDARD */

button:hover, .buttons a:hover{
    background-color:#dff4ff;
    border:1px solid #c2e1ef;
    color:#336699;
}
.buttons a:active{
    background-color:#6299c5;
    border:1px solid #6299c5;
    color:#fff;
}

/* POSITIVE */

button.positive, .buttons a.positive{
    color:#529214;
}
.buttons a.positive:hover, button.positive:hover{
    background-color:#E6EFC2;
    border:1px solid #C6D880;
    color:#529214;
}
.buttons a.positive:active{
    background-color:#529214;
    border:1px solid #529214;
    color:#fff;
```

```
}

/* NEGATIVE */

.buttons a.negative, button.negative{
    color:#d12f19;
}
.buttons a.negative:hover, button.negative:hover{
    background:#fbe3e4;
    border:1px solid #fbc2c4;
    color:#d12f19;
}
.buttons a.negative:active{
    background-color:#d12f19;
    border:1px solid #d12f19;
    color:#fff;
}

/* REGULAR */

button.regular, .buttons a.regular{
    color:#336699;
}
.buttons a.regular:hover, button.regular:hover{
    background-color:#dff4ff;
    border:1px solid #c2e1ef;
    color:#336699;
}
.buttons a.regular:active{
    background-color:#6299c5;
    border:1px solid #6299c5;
    color:#fff;
}

/* Pagination controls and images from
 * http://datatables.net/release-datatables/media/css/demo_table.css
 */

.dataTables_paginate {
        float: left;
        text-align: left;
}

.paginate_button_disabled {
        height: 19px;
        cursor: pointer;
        *cursor: hand;
        color: #111 !important;
}
.paginate_button_disabled:hover {
        text-decoration: none !important;
}
.paginate_button_disabled:active {
        outline: none;
```

```
}

.paginate_button_disabled.previous,
.paginate_button_disabled.next {
        color: #666 !important;
}
#display_table_previous {
        padding-left: 23px;
        padding-right: 10px;
}
#display_table_next {
        padding-left: 10px;
        padding-right: 23px;
}

.paginate_button_disabled.previous {
        background: url('../images/back_disabled.jpg') no-repeat top left;
}

.paginate_button.previous {
        background: url('../images/back_enabled.png') no-repeat top left;
}
.paginate_button.previous:hover {
        background: url('../images/back_enabled_hover.png') no-repeat top left;
}

.paginate_button_disabled.next {
        background: url('../images/forward_disabled.png') no-repeat top right;
}

.paginate_button.next {
        background: url('../images/forward_enabled.png') no-repeat top right;
}
.paginate_button.next:hover {
        background: url('../images/forward_enabled_hover.png') no-repeat top right;
}


.paging_full_numbers {
        width: 400px;
        height: 22px;
        line-height: 22px;
}

.paging_full_numbers a:active {
        outline: none
}

.paging_full_numbers a:hover {
        text-decoration: none;
}

.paging_full_numbers span.paginate_button,
        .paging_full_numbers span.paginate_active {
```

```
        border: 1px solid #aaa;
        -webkit-border-radius: 5px;
        -moz-border-radius: 5px;
        padding: 2px 5px;
        margin: 0 3px;
        cursor: pointer;
        *cursor: hand;
        color: #333 !important;
}

.paging_full_numbers span.paginate_button {
        background-color: #ddd;
}

.paging_full_numbers span.paginate_button:hover {
        background-color: #ccc;
        text-decoration: none !important;
}

.paging_full_numbers span.paginate_active {
        background-color: #99B3FF;
}
```

```
commit df2a10d1c6431b7e1babaa4ac36fbfa1882658d9
Author: Lee Hall <lhall@newfields.com>
Date:   Wed Nov 28 15:54:55 2012 -0500

    Documentation Updates

commit 6659e655b02a0385d68a94c717c76e271509c4e6
Author: Lee Hall <shall4@spsu.edu>
Date:   Wed Nov 28 11:33:09 2012 -0500

    Updated images

commit 4ff53c24359e39cd2bea0191069a79e0f2e3ab9a
Author: Lee Hall <lhall@newfields.com>
Date:   Wed Nov 28 10:27:44 2012 -0500

    Buttons for stop/start

commit 001b87fb34305a981456a208bbea59047f7daac9
Author: Lee Hall <shall4@spsu.edu>
Date:   Wed Nov 28 10:01:54 2012 -0500

    Live updates work

commit 703eb458cb5ba71e53a5040f29a3b9b2a9ec5eac
Author: Lee Hall <shall4@spsu.edu>
Date:   Tue Nov 27 21:20:18 2012 -0500

    Fixed table width

commit 4f4d566bf894cf3cfc530b8d02b8224660a6c7e0
Merge: 33cd7d7 e6e331c
Author: Lee Hall <lhall@newfields.com>
Date:   Tue Nov 27 19:57:19 2012 -0500

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

commit 33cd7d76238bc4f2655a457806f6520578727853
Author: Lee Hall <lhall@newfields.com>
Date:   Tue Nov 27 19:56:58 2012 -0500

    Bugfixes

commit e6e331c365f63d0b39bfb7642bdceb493396d8a8
Author: Lee Hall <shall4@spsu.edu>
Date:   Tue Nov 27 12:50:15 2012 -0500

    Bug cleanups in the CSS.

commit 26584cca0f9f8be3c1213c84799d5263dd1b8efe
Author: Lee Hall <lhall@newfields.com>
Date:   Tue Nov 27 04:05:44 2012 -0500

    Interfaces for timer and difference based crawling
```

```
commit e169e75361f70a529fc0cbf2f076c63ef3b204d3
Author: Lee Hall <lhall@newfields.com>
Date:   Tue Nov 27 01:11:36 2012 -0500

    Scanner UI, Pause/Resume

    Added the scanner UI from Robin, fixed pause functionality in crawler.

commit 5f244d2a2ad90ae446329788d5dce60bebae1152
Author: Lee Hall <shall4@spsu.edu>
Date:   Sun Nov 25 18:43:40 2012 -0500

    Fixed comparison query

commit 7f4eb623936eea3eb1807e3024d91c14073aae50
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sun Nov 25 17:49:19 2012 -0500

    compare

commit 3a968e18ce65b887004405e5437ff607f6e3649c
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sun Nov 25 17:24:29 2012 -0500

    compare update

commit cf5e1931a2e00d973caf37956fd7697ef9702007
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sun Nov 25 17:23:41 2012 -0500

    update

commit 2d9420f250c43a93e4f23c554f5ebf297e380403
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sun Nov 25 17:17:42 2012 -0500

    same update

commit fec8ed9608f782654fa625fe32b02e4061dc4e95
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sun Nov 25 17:14:55 2012 -0500

    compare reports update!!
    Print function minimally added!!

commit 68752d6276d2e078e560d9df44d73492c10cacaa
Author: tcouture127 <tcouture127@gmail.com>
Date:   Sun Nov 25 15:16:52 2012 -0500

    bad idea

commit 6347ca7132756e9c01b1f2f4e3e67b51d2354c6a
Author: tcouture127 <tcouture127@gmail.com>
```

```
Date:   Sun Nov 25 15:14:36 2012 -0500

    table update

commit 5eebaf80cddef030cedfd25b6fec9404a43c9b0e
Author: Lee Hall <lhall@newfields.com>
Date:   Sun Nov 25 14:49:32 2012 -0500

    Added json session management

    Apparently the file to do json session management didn't make it into
    the last commit

commit 05013ce72a75782289c5cc012039ab3a1442e318
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Nov 24 20:48:27 2012 -0500

    Dynamic AJAX based crawler control

    control_json.php can now start, stop, and query the status of the
    crawler. There's a lag on the latter that needs to be hammered out.
    Session timeouts are now correctly checked for in the javascript page
    loads, as well.

commit c64194bf37cca07b23766a06488199eb3d2be873
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Nov 24 17:03:26 2012 -0500

    More work on pause/resume functionality

commit 931983898d6fde6fa87e1d1f1f269e22ec9f65b7
Author: Lee Hall <lhall@newfields.com>
Date:   Sat Nov 24 16:15:49 2012 -0500

    Matthew's crawler updates

    Allow crawler to run based on a previous scan

commit 63b29575f887a7bda3a28e609a86770506ad2e16
Author: Lee Hall <lhall@newfields.com>
Date:   Wed Nov 21 05:38:46 2012 -0500

    JSON for progress.

    remaining_json.php reports number of records thus far in the current
    scan, as well as the start date.

commit c5b51b79174c7169b22be9226ed5895d232910c9
Author: Lee Hall <lhall@newfields.com>
Date:   Wed Nov 21 05:19:10 2012 -0500

    Signal handling for pause/resume works

    Pause: SIGUSR1 (10)
```

```
    Resume: SIGCONT (18)
    Pauses in the midst of a URL request return to the execution thread
    properly and nothing is lost.

commit cc31cb76f0aa2ac023c928cb92f881cf9c0d8927
Author: Lee Hall <lhall@newfields.com>
Date:   Wed Nov 21 04:58:14 2012 -0500

    Command Line Argument parsing

    Setup basic CLI argument parsing.

commit bbb8ffa3defe6674013155788139a3ca193490c0
Author: Lee Hall <shall4@spsu.edu>
Date:   Wed Nov 21 01:51:58 2012 -0500

    Cleanup stylesheets

    Fix some CSS problems I introduced shifting over to AJAX.

commit 4877af6f8b7467e56005fb5b8a78bd066cd553f6
Author: Lee Hall <lhall@newfields.com>
Date:   Wed Nov 21 00:19:29 2012 -0500

    Combined pages

    Moved all of the tables to independent json files that are loaded with
    javascript.

commit c6d0a60992e5ca1100e711d9c1e3207450efdb3b
Author: M <swordthane@gmail.com>
Date:   Tue Nov 20 15:02:26 2012 -0500

    Test Plan

commit 66af9d64822dbb79614d87d324baf01528acbd1d
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 17:18:02 2012 -0500

    update

commit 47918753ebfa7aa529044eb10760adc46a57f850
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 17:13:55 2012 -0500

    updates

commit 08fa2ae5cf43037ee39d0980a389194d3e73c432
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 17:06:50 2012 -0500

    updates

commit 29d905ab3517b42735017f511c33cebb0cf91e27
```

```
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 17:03:00 2012 -0500

    table changes

commit 5b32d3999ed61a70d1928e0697070c12fd6ae2d1
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 16:56:53 2012 -0500

    table

commit b3bff8f0ddb89b4f28a8accc53d29ae318a28660
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 16:48:30 2012 -0500

    updats

    Signed-off-by: tcouture127 <tcouture127@gmail.com>

commit d01d2a4aa11b1559c669a4fe17950978aaf45a42
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 16:44:03 2012 -0500

    update

commit d8f9f65552d5cfda1bfdfec4bf755d7cef2c9c93
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 16:41:24 2012 -0500

    table update

commit 5d1d5736d2313afb0216ef9fe86e93600de5f2e1
Merge: 2088d67 c3f263a
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 16:34:49 2012 -0500

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

commit 2088d670dddcf55143c5ca17bf0ef5b4c7cc847b
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 16:34:23 2012 -0500

    Table styling update

commit c3f263a024f8b61228cc600e1389a5f4a6862f32
Author: Lee Hall <lhall@newfields.com>
Date:   Tue Nov 13 14:47:42 2012 -0500

    Fixed undefined variable ref

commit c836bb21d65e709e821556ea5ca5e6d01c1b7c3b
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 14:29:20 2012 -0500
```

```
    Mass UI Minor Update

commit c722c0f27ec9f2a2219d94ce999bdb1c000a5440
Merge: b8c6ba1 82ba5f6
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 14:23:45 2012 -0500

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

commit b8c6ba146512e811d849afcbacb90985676cdfd9
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 14:21:14 2012 -0500

    login UI changes

commit 82ba5f6019f052b493f227124591fb7d831bb75d
Author: Lee Hall <shall4@spsu.edu>
Date:   Tue Nov 13 14:14:55 2012 -0500

    Readded form elements

commit b2ff4214a50b0e852587de773e5a2379b5a4f915
Author: Lee Hall <shall4@spsu.edu>
Date:   Tue Nov 13 14:05:11 2012 -0500

    Fixed redirect loop

commit 44cc29cd1c2d35f66e184bd1ee44b34a2ae4cdd4
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 14:02:47 2012 -0500

    Header Update/ login fix

commit 88d16543a7cf787d7fd0996b2a425f65b6411f41
Merge: 5b1cc86 f17acf7
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 13:42:58 2012 -0500

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

commit 5b1cc86c2227267e999003ec367f2e5bdfc1f8d3
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 13:42:31 2012 -0500

    update

commit f17acf753fd998cec428af2d6122397c861c53d3
Author: Lee Hall <shall4@spsu.edu>
Date:   Tue Nov 13 13:38:01 2012 -0500

    Updated the test corpus

commit ccfd447405c513a5666082a507de31bc66f4fedf
Author: tcouture127 <tcouture127@gmail.com>
```

```
Date:   Tue Nov 13 13:32:18 2012 -0500

    Login Updates

commit d3d022fc11482491d04acd34228006de58e6f65a
Merge: f465254 0905f22
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 13:24:04 2012 -0500

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

commit f465254306355bd4ab6fdd8babc6e8ddf01011b1
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 13 13:22:42 2012 -0500

    login updates

commit 0905f22ca386630412fc8fa45624c039d6708f5b
Author: Lee Hall <lhall@newfields.com>
Date:   Tue Nov 13 13:16:18 2012 -0500

    Crawler bugfixes

    Added NOTES to track crawler issues

    Moved check for whether or not a resource is local into the resource
    class where it belongs. Now we can test and record external resources
    without needing to parse them for their own children.

    Added some new catches to not fetch various URI schemes we don't
    understand. This should all go into resource.valid_uri() so that we can
    also record these references and not have our codebase look disgusting.

commit 6dc4a1fdab86399382c0cb6d0b2cade09ac08468
Author: Lee Hall <lhall@newfields.com>
Date:   Tue Nov 13 01:14:50 2012 -0500

    Use relative links

    Changed links to relative so that they work in non-identical
    environments.

commit ab35659b72062702466c1f54e5de42b099d6a66b
Author: Lee Hall <shall4@spsu.edu>
Date:   Tue Nov 13 01:09:22 2012 -0500

    Added a test case

    Added a testcase for inlined images

commit ea114e2fcb8c284686b4ec51b2c38ab8e0935ded
Author: Lee Hall <shall4@spsu.edu>
Date:   Tue Nov 13 01:09:07 2012 -0500
```

```
    Visual cleanup for presentation

    Forgot to check-in the visual cleanups for the presentation.
    Removed cruft from the login page and truncated date fields in the
    reports.

commit 7511d34031a3c9aa2f2dd66e3b7809e37817d568
Author: Lee Hall <lhall@newfields.com>
Date:   Mon Nov 12 23:55:11 2012 -0500

    Fixed bug when downloading inlined images

    Don't download inlined images or mailto links. Added more logging
    messages.

commit fff7c6b461888f2af121d63d10f12025bbcafc69
Author: Lee Hall <lhall@newfields.com>
Date:   Mon Nov 12 23:52:53 2012 -0500

    Documentation updates

commit 376b1591abc6b7b888f56d0ba31cb48a50ec04ec
Author: Lee Hall <lhall@newfields.com>
Date:   Tue Nov 6 18:32:16 2012 -0500

    Report Builder updates

commit 91b83b9d4b849bf2cdd145de26e0bd6e4d4e2c4a
Merge: 21bd9af 1d17921
Author: Lee Hall <shall4@spsu.edu>
Date:   Tue Nov 6 14:43:42 2012 -0500

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

commit 1d17921765450b05948e9d298a600951a59081cc
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 6 14:35:21 2012 -0500

    UI Minor Changes

commit 903b8f922187897233cd78453b5d45b9764fe888
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 6 13:41:13 2012 -0500

    Doc Updates

commit faaa7c5423a68108bb0c6a8d477aabae92c43d4c
Author: tcouture127 <tcouture127@gmail.com>
Date:   Tue Nov 6 13:13:50 2012 -0500

    Doc Updates!

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 21bd9af6d8eb3be9de514dfea41dea301b2c4a9c
Author: Lee Hall <shall4@spsu.edu>
Date:   Tue Nov 6 00:09:00 2012 -0500

    Made logging a bit more sane and readable

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
```