



Damien Russell

CSC493 Final Report

12/12/2011

TABLE OF CONTENTS

Application Development Section	3
Project Concept Proposal	3
Purpose:	3
Resources:	5
Reference Articles:	5
Challenges:	6
Measures:	6
Future Extensions:	6
Inspiration	7
Motivation:	7
Profession:	7
Vision and Scope	8
Software Requirements Specifications	9
System Design and Architecture	10
Main Functionalities:	10
Software Components:	11
Architectural Model	12
Entity-Relationship Diagram	13
Implementation	14
Test Plan	17
Test Cases	18
Known Bugs and Other Issues – in netStuff 0.0.1 beta	22
About the Author	23
Executive Section	24

Application Development Section

Project Concept Proposal

- **Purpose:**

The purpose of this website is to present users with a social platform to log their movie experiences. Also, the purpose is to find trends about user viewing habits, while providing a quick and easy way to incorporate this data into their social lives. Additionally, it seeks to fill the need for a bridge between the user's desire to see a film and the time it takes for the user to access that film on Netflix, and the gap between a user rating a movie on Netflix and their social experiences via the web.

- **Context:**

The context of this website is to fill the need for users to log their movie experiences, such as ratings for films or must-see's. The context is also to teach the user about his or her viewing habits. By incorporating Facebook platform/API and Netflix API with netStuff, netStuff will provide a richer platform for users to capture their viewing experience from multiple locations while allowing users to streamline their viewing experiences with their social lives via Facebook and Netflix, all while learning a little about themselves and the types of films they watch.

- **Goals:**

The goal of this project is to develop a website to log user's movie experiences and teach the user about his or her viewing habits. This project aims to provide the functionality for users to rate movies, review movies, log the amount of films they watch, examine their viewing habits in simple format for them to understand like pie charts or graphs. As well as provide a way for users to share this data on Facebook. Due to the time available for this project and the complexity already involved, netStuff goals for the number of film sources a user can capture his experiences from, such as Blockbuster, Netflix, Regal Cinemas, Google+, or Facebook, and limit this ability to capture experiences films viewed/rented/rated through Netflix and movies liked/shared on Facebook. NetStuff focuses on these two platforms because of their penetration into the social and entertainment aspect of so many lives.

- **Audience:**

The audience for this project is broad, as it targets anyone of the appropriate age to surf the web and engage in social networking platforms. It is for anyone who has a passion for movies, and would like to log their experiences to share with friends. The more specific audience would users who actively merge their daily lives with their virtual social experiences, like those found on Facebook, and seek to incorporate or share their viewing habits/experiences/data with their extended social networks.

- **Functionality:**

As a website, anyone with an internet connection and a web browser can track the movies they view on their Netflix account or share/like on Facebook. The site will present users with easy ways to view trends in their viewing history. Users can view all their ratings and review from Netflix or Facebook in one location. From this location users can also quickly get to films they haven't seen but would like to, as long as that film is on Netflix.

- **Milieu:**

Rotten Tomatoes, IMDB, and Flixster are all similar to netStuff, however none of them provide a way for the user to learn about their viewing habits, which would make the users experience on these sites more personal and encourage traffic by stimulating the users' interest in their own actions. Another advantage netStuff has over sites like Rotten Tomatoes, IMDB, and Flixster, is that it incorporates Netflix into its functionality. Netflix is a strong platform for users to cheaply watch TV shows and movies anytime from home or on the go. On top of being a platform with a very hefty user base, when users experience most of their viewing through Netflix the variety of films and shows available for users to log, as well as the amount of data that can be stored on those viewing sessions, is greatly increased.

- **Novelty:**

There are several databases/platforms designed to bring a social networking experience into our video viewing experiences, however netStuff, provides a way for user to learn about new things about themselves and their viewing habits, such as the viewing frequency of certain films, genres, or actors. As well as bridging the gap between the user's desire to see a film and the time it takes for the user to get to Netflix to view that film.

- **Resources:**

#Updated: 9/25/2011 added resources

Facebook API:

<https://developers.facebook.com/docs/>

Facebook Open Graph Protocol:

<http://ogp.me/>

JavaScript SDK:

<https://developers.facebook.com/docs/reference/javascript/>

Facebook Query Language (FQL):

<https://developers.facebook.com/docs/reference/fql/>

Notepad++:

<http://notepad-plus-plus.org/>

Adobe Photoshop CS5:

<http://www.adobe.com/products/photoshop.html>

PHP 5.3.0:

http://php.net/releases/5_3_0.php

HTML5:

<http://www.w3.org/TR/html5/>

CSS3:

<http://www.w3schools.com/css3/default.asp>

MySQL 5.1.36:

<http://www.w3schools.com/sql/default.asp>

- **Reference Articles:**

Grannell, Craig. "Plan your way to the perfect site." *.net* Summer. 2011: 44-50. Print.

Najmuddin, Iskandar, Folkes, Stephen. "The pro's guide to Facebook." *.net* Jun. 2011: 44-50. Print.

Smarr, Joseph. "Using Netflix's New API: A step-by-step guide." <http://josephsmarr.com>. WordPress. 1 Oct. 2008. Web. 2 Sept. 2011.

- **Challenges:**

Keeping it simple will be a challenge, my list of things to teach the user about his/her self will require more permissions in order for my website to access the users profile info on Netflix and Facebook. Facebook's data has shown an inverse correlation between the amount of users who accept the request and the amount of permissions being requested. I must keep this in mind.

Learning to use both Facebook and Netflix API in such a short amount of time will reduce the amount of goals I can strive for this semester.

Launching a fully functioning website in a little over 3 months is no easy task, especially when incorporating utilities from one of the webs biggest social networking platforms.

- **Measures:**

#Updated 12/10/11.

The website is up and running.

Users can see a profile page containing their data from Netflix and Facebook.

Users can see a facts page in a sense. It contains nifty facts about their viewing habits. [total movies, directors, and actor lists]

- **Future Extensions:**

Possible expansion to Google's social platform Google+.

Incorporate other entertainment platforms such as Hulu.

Expansion to other entertainment mediums such as music or video games.
(Via. Xbox Live Profiles and Spotify)

Inspiration

- **Motivation:**

This project peaks my interest because the lack of tools on the web to share my Netflix experiences with my social networks. For a while I have been disappointed that Netflix does not have a like button, which would be a simple bridge between the movies and shows I view on Netflix and my Facebook social network.

- **Profession:**

This project will help my skills as a programmer by challenging my ability to pick up new programming tools like APIs and engage the web as the new open social graph/web it is becoming today. The experience and credibility associated with completion of such a project will certainly help my professional growth. Working with the APIs for platforms like Facebook will help me to further meld my interests in mass collaboration as described in Wikinomics and MacroWikinomics, and the latest in business and marketing strategies. Learning to utilize APIs for social platforms like Facebook puts me one step ahead when launching new ventures suited for our society as it is today.

Vision and Scope

NetStuff seeks to become an entertainment social networking platform, that leverages current social networks and services to engage all forms of entertainment the user may like. NetStuff envisions a site that incorporates all your favorite games, movies, and music scattered across the web, all in one easy to access location. NetStuff seeks to not only collect and organize this data for the user, but to also interpret and present this data in such a way that it teaches the user something new about their viewing, listening, or gaming habits.

Considering the amount of work required for such a platform and the limited time available to implement this project, the scope must be scaled back significantly. For this project, I will focus on implementation for one browser, Google Chrome, and will reduce the social networks being leveraged to just Facebook. For this project the forms of entertainment that will be pulled from Facebook will be limited to movies. All within the scope of this project for the fall semester, netStuff will provide a functional site for users, with a page that breaks down the data from Facebook, organizing facts about the user's movie viewing habits in a simple table. If time permits users will be able move data back to Facebook from netStuff. Also depending on time, users will be able to pull their viewing data from Netflix, to be incorporated with their viewing data from Facebook.

Software Requirements Specifications

#Updated: 10/22/2011 added Authentication & Authorization and edited requirement 3.

1. A live website
 - **Evaluation Method:** Can be accessed via Chrome
 - **Dependencies:** None
 - **Priority:** High
2. Users login/create a profile
 - **Evaluation Method:** Users can view their profile through gonetstuff.com
 - **Dependencies:** 1
 - **Priority:** High
3. Obtain a Facebook access token for a user.
 - **Evaluation Method:** Gain access to the users Open Graph data.
 - **Dependencies:** 8
 - **Priority:** High
4. Access a movie, shared/like by a user, from the users Open Graph Data
 - **Evaluation Method:** Movie data in XML format is obtained.
 - **Dependencies:** 1, 2, 3
 - **Priority:** Medium
5. Store a movie [title, actors, description] from users open graph data, in a database.
 - **Evaluation Method:** A record is made for the movie connected to the user.
 - **Dependencies:** 1, 2, 3, 4
 - **Priority:** Low
6. Interpret users data and represent it on their viewing habits breakdown.
 - **Evaluation Method:** User can see a breakdown of viewing habits on netStuff.
 - **Dependencies:** 5
 - **Priority:** High
7. Move data from netStuff to Facebook.
 - **Evaluation Method:** Data can be seen in users Facebook Profile.
 - **Dependencies:** 1, 2

- **Priority:** If time permits

8. Authenticate User and Authorize App

For the Sever Side flow User authentication and app authorization are handled at the same time by redirecting the user to our OAuth Dialog.

- **Evaluation Method:** User is asked to login/approve site.
- **Dependencies:** 1, 2
- **Priority:** High

System Design and Architecture

Main Functionalities:

1. Log Users: Allows users to have profiles with an open graph connection to the user's Facebook account. This account can then be connected to sets of viewing data that will be accumulated and then presented, in a friendly manner, the data and trends discovered in the data.

2. Log Movies: Allows for movies like/shared on a user's Facebook profile to be gathered and analyzed, then connected with the user's netStuff profile. A movie will be found under the user's Facebook profile metadata in the open graph. netStuff will then find the movie's metadata in the open graph to collect pictures, directors names, actors names, and other data. This data can be filtered and presented in tables, pie charts, or bar graphs to allow users to discover things about their viewing habits

3. Analyze User Data: Once logged, user viewing data can be tallied and filtered to produce tables, pie charts, and bar graphs for the user to discover new trends or facts about their viewing experiences.

Software Components:

#Updated 12/10/11 grammar correction.

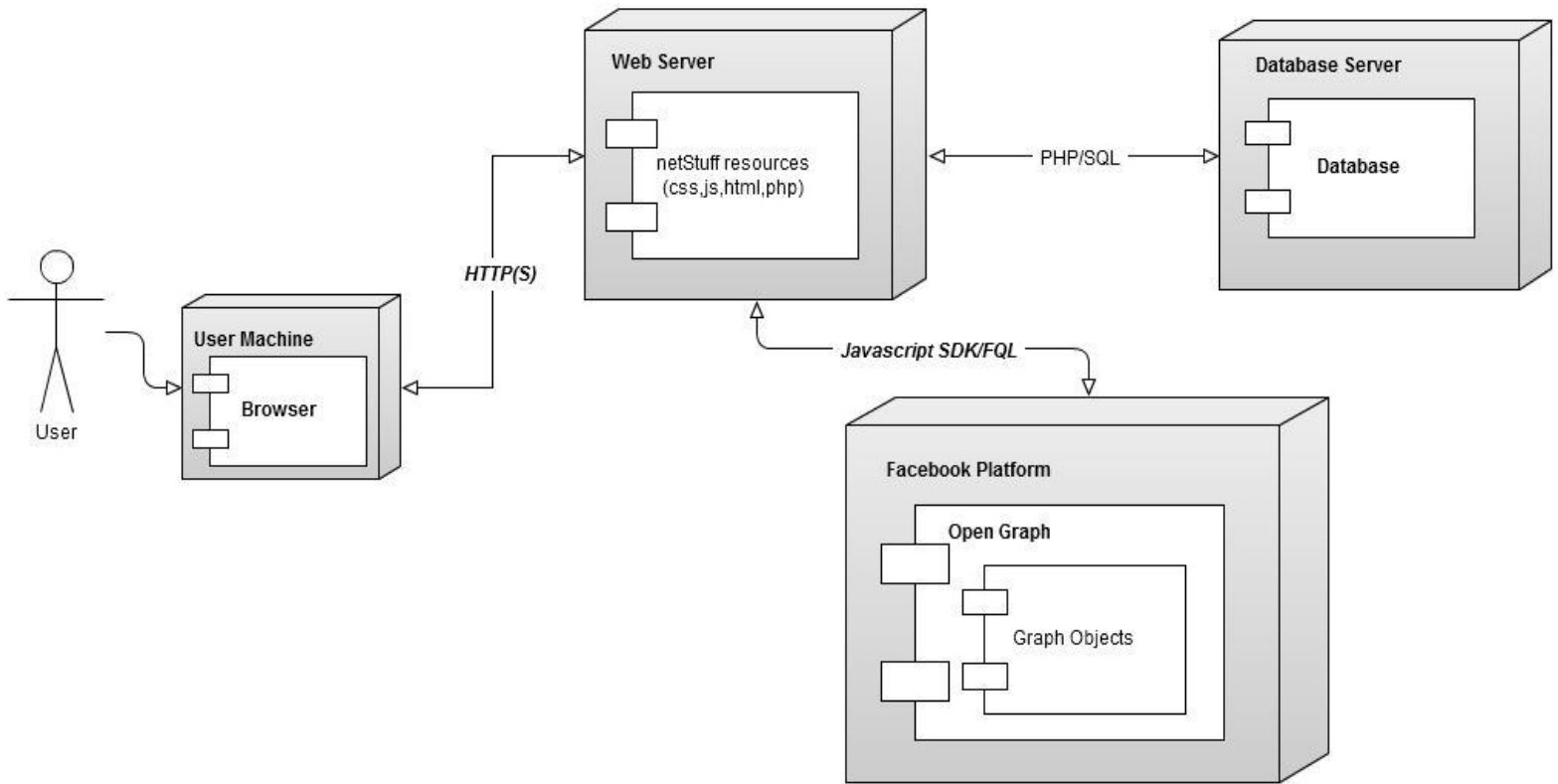
1. Database: MySQL database contains all data related to users. As well as, all data from films viewed by a user. This includes an open graph id, a name, a director name, actors, a picture, genre, the number of likes, and a plot outline.

2. Web server: Allows for communication between the users web browser and the netStuff website and the underlying database.

3. Web Site: The PHP, HTML, CSS, and JavaScript files that make up the interface for the user to interact with the netStuff website.

4. Facebook Platform: An external platform independent of netStuff's functionality; though it will play a large supporting role for much of netStuff's functionality. Facebook's open graph will provide a majority of the data for users along with movies and serve as a base to build netStuff on.

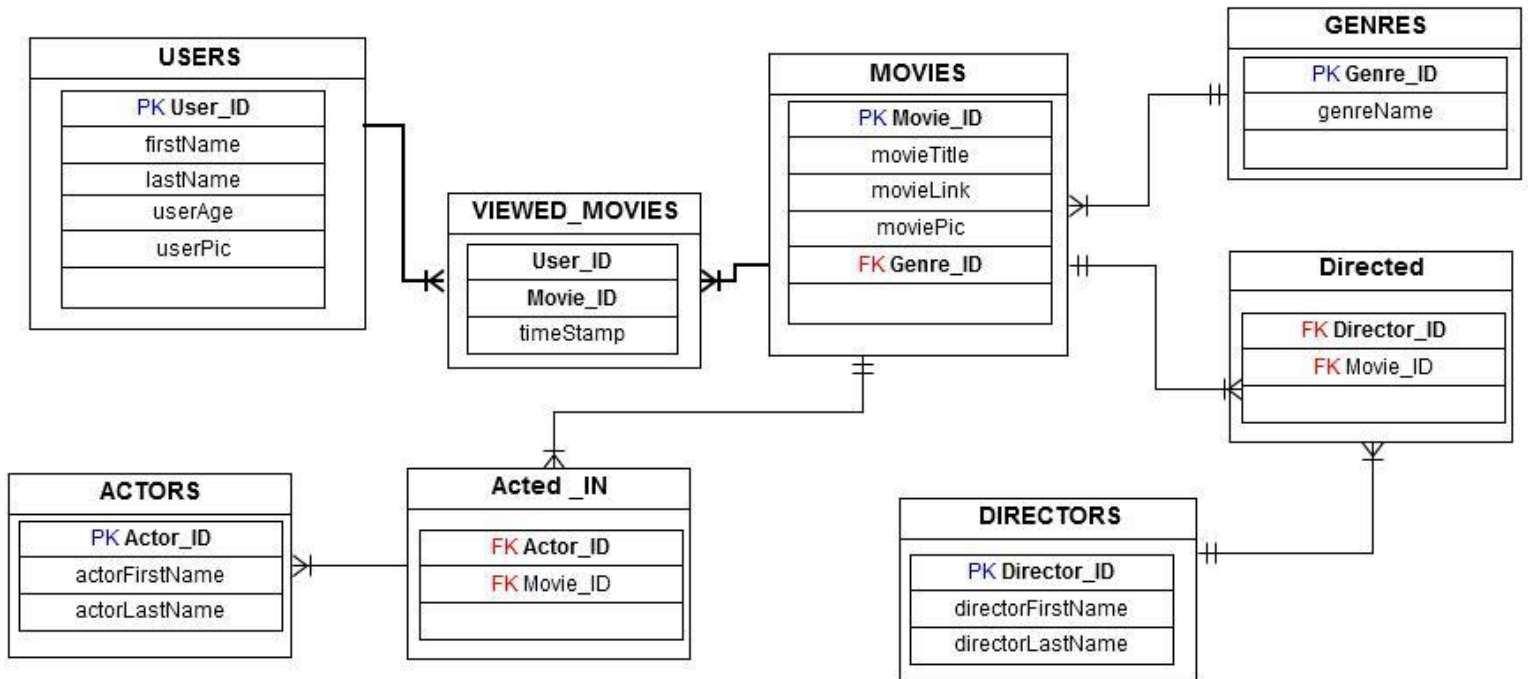
Architectural Model



The above diagram illustrates the flow of data between the user and the netStuff website; As well as, the flow of data between the netStuff site and the database and the flow between the Facebook platform and the netStuff site.

Entity-Relationship Diagram

#Updated: 10/14/2011 normalized tables



create and share your own diagrams at gliffy.com



The above diagram illustrates the entities, their attributes, and their relationships as they will appear in the database.

Implementation

#Updated: 12/11/2011 semantics corrections and additional files added and some renamed.

- ReadMe.Txt
 - Software Requirements and Instillation.
- index.php
 - Page seen by users when not logged in or on initial visit.
- home.php
 - In this php script we check to see if the user's has like/or shared any movies not in the database. This script also checks to see if we have made note of the user's viewing of the given film. Moreover, the script checks to see if there are any actors, directors, or genres, from a particular movie, that are not in the database. If so it adds them.
- Profile.php
 - This page displays the user's totals and a list of films the user has seen. Most importantly the User's Profile picture is simply the picture the User is currently using as their Facebook profile picture.
- discoverUser.php
 - discoverUser.php checks each user that logs in to see if the user is in the database. If the user is not found then he or she is added using the information provided by Facebook.
- Contact.php
 - Contact.php is the contact page for the netStuff website. This page features information about me the developer; as well as, an email address to reach me and a link to the netStuff Facebook page.
- About.php
 - About.php is a static page that informs netStuff users about the vision of the site. It is an about page for the site not me the developer this can be found on the Contact.php page.

- genreCount.php
 - This script checks for the number of genres, horror, comedy, action and drama in particular, that a user has viewed. It searches the database for movies the user has viewed and returns the movies with genres 'like' Action, Comedy, or Drama; then the number of rows found is returned.
- footer.html
 - The master footer for the NetStuff website. The login page(index.php) is the only page that does not use the master header and footer.
- header.html
 - The master header for the NetStuff website. The login page (index.php) is the only page that does not use the master header and footer.
- Channel.php
 - It is important for the channel file to be cached for as long as possible. When serving this file, you must send valid Expires headers with a long expiration period. This will ensure the channel file is cached by the browser which is important for a smooth user experience. Without proper caching, cross domain communication will become very slow and users will suffer a severely degraded experience.
- Img folder:
 - This is a folder of images that are essential for certain pages to display correctly. This includes the logo in the header across pages.
- css folder:
 - index.css
 - Style for the index.php page.
 - Rest.css
 - Based upon 'reset.css' in the Yahoo user interface library: <http://developer.yahoo.com/yui> (Borrowed From Adi Purdila Tuts.com). This file clears the standard styling set by user's web browser.
 - masterStyle.css
 - This is the style sheet for the header and footer on each page of the netStuff website; as well as, the general styles such as header html tags.
 - Home.css

- This is the style sheet for the home.php page. This style sheet is necessary to style the tables that are created from User Facebook data.
- Contact.css
 - This style sheet is used for the content on the Contact.php page.
- Profile.css
 - This style sheet styles the data found on Profile.php.
- Facebook PHP SDK folder:
 - This folder is the PHP SDK downloaded from source forge. This file allows the sever to work with the Facebook platform.

Test Plan

#Updated 12/10/11 grammer corrections and approach rewritten.

Purpose

The purpose of this website is to present users with a social platform to log their movie experiences. Also, the purpose is to find trends about user viewing habits, while providing a quick and easy way to incorporate this data into their social lives.

Features to be tested

- Site's live status.
- User authentication.
- App authentication.
- App authorization.
- Data flow from user's openGraph data to netStuff databases.
- Display of user movie data in charts and tables.

Features not to be tested

Sending users data from netStuff to Facebook.

Approach

Test accounts will be created on Facebook. A few movies will be liked before going to goNetStuff.com. Dummy accounts will login to Facebook through netStuff and navigate the site insuring data and tables are functioning as expected.

Suspension criteria and resumption requirements

Test breaks in one of the steps.

Problem is recorded, identified, and corrected.

Environmental Needs

Windows 7. Latest version of Google Chrome. Facebook profile with movies like or shared through Facebook account. netStuff Database is installed on eleven2 sever.

Schedule

- Nov 3: TC-1; TC-2; TC-8;
- Nov 5: TC-3; TC-4;
- Nov 6: TC-5; TC-6;
- Nov 7: TC-7;

Acceptance criteria

Site must meet all test cases with security risks at a minimum.

Roles and responsibilities

Sole responsibility for testing each case will be left up to me, the lead developer.

Test Cases

#Updated 12/10/11

Name	TC-1: Verify that the page is live
Requirment	None
Preconditions	User's browser is google chrome.
Steps	<ol style="list-style-type: none">1. Connect to the internet2. enter gonetstuff.com into the address bar.
Expected Results	<ol style="list-style-type: none">1. The users should see be directed to index.html.

Name	TC-2: Login/Registration
Requirment	TC-1: Veryfy that the page is live;
Preconditions	User has a Facebook account. User is not logged into Facebook.
Steps	<ol style="list-style-type: none">1. User sees a Facebook login button.2. Click Login with Facebook3. Give Facebook your credentials.4. Click login.
Expected Results	<ol style="list-style-type: none">1. Facebook verifies user credentials.
Results	<ol style="list-style-type: none">2. Facebook logs user into Facebook.2. User should be directed to the index.php.3. An email address field with a submit button is displayed where the Facebook login was.

Name	TC-3: Access token obtainment.
Requirement	TC-2; TC-8;
Preconditions	User is logged into Facebook.
Steps	1. Direct to the netStuff page.
Expected	1. User will be directed to index.php.
Results	2. Facebooks servers will return a signed_request with an access token to access the user's data on behalf of the user.

Name	TC-4: User's movie(openGraph) access.
Requirement	TC-2; TC-3;
Preconditions	User has a Facebook. User is logged into Facebook.
Steps	1. Like or share a movie on Facebook. 2. Direct to the netStuff page. 3. Verify your email. 4. Direct to user's profile page.
Expected	1. User should see the movie on their Facebook page.
Results	2. User should see the movie on their netStuff movies table.

Name	TC-5: Store movie viewed by user.
Requirement	TC-4;
Preconditions	User has like/shared atleast one new movie on Facebook. User is logged into Facebook.
Steps	<ol style="list-style-type: none"> 1. Direct to the netStuff page. 2. netStuff request all movies you have shared. 3. If a movie was is not already in the movies table it will be added. 4. If a movie is not already associated with the userID via the watched table it will be added to the watched table with the userID. 5. netStuff updates movies table and watched table
Expected Results	<ol style="list-style-type: none"> 1. NetStuff catches movies not already in the database. 2. NetStuff updates watched tables when user's share/like new movies.

Name	TC-6: Display user movie data in charts and tables.
Requirement	TC-5;
Preconditions	User has several movies liked/shared on Facebook. User is logged into Facebook.
Steps	<ol style="list-style-type: none"> 1. Direct browser to netStuff.com 2. Enter email address on the login page then click submit. 2. Visit home page via navigation bar. 3. Three tables should be displayed Directors, Genres, and Actors.
Expected Results	<ol style="list-style-type: none"> 1. User see's three tables.
Results	

Name	TC-8: Authenticate User/ Authorize App
Requirment	TC-2;
Preconditions	User is logged into Facebook.
Steps	<ol style="list-style-type: none"> 1. User directs browser to gonetstuff.com. 2. User clicks' the Facebook registration button. 3. User completes TC-2; 4. User clicks allow in the authorize app window.
Expected Results	<ol style="list-style-type: none"> 1. The users should be directed to index.php. 2. User is will see a email address field to be completed with the users email used for Facebook.

Name	TC-9: Logout Button
Requirment	TC-2; TC-3
Preconditions	User is logged into Facebook and is viewing the login page gonetstuff.com/index.php.
Steps	<ol style="list-style-type: none"> 1. Click logout of Facebook button displayed bellow the email address filed and submit button. 2. In a new tab direct to Facebook.com
Expected Results	<ol style="list-style-type: none"> 1. The gonestuff.com page should perform a page load and display the login button where the email address field was located. 2. The tab directed to Facebook.com should display the Facebook login page, note the user should no longer be logged into Facebook.

Known Bugs and Other Issues – in netStuff 0.1

#Updated 12/11/11

-Currently when logging into netStuff after giving Facebook your credentials in the OAuth dialogue window; the dialogue window itself hangs. You have to close it and refresh the page in order to proceed and have the email address field displayed.

-The email address field on the home page does not currently check the email you enter; though it still directs you to the home.php page.

-The subscribe to our email form does not currently have a script to add users to the email list.

-The search box does not currently have an algorithm to search the netStuff site.

Software Demo(s)

- A demo for goNetStuff.com. The demo shows a dummy Facebook user Ruben who shares a few films on Facebook. Ruben directs his browser to goNetStuff.com to show how those movies are represented on NetStuff.
- Video at: <http://www.youtube.com/watch?v=9xVfE-5pkXM>

About the Author

Damien Russell is a senior Computer and Information Science major at Berea College, in Berea, Kentucky. Damien is a family man, an artist, and a programmer. When he is not spending time with his two year old daughter and his lovely wife, he loves to play in Adobe Illustrator, to paint, to draw, and to program in Python.



Executive Section



To: Jan Pearce

From: Damien Russell

Subject: netStuff

Date: 09/05/2011

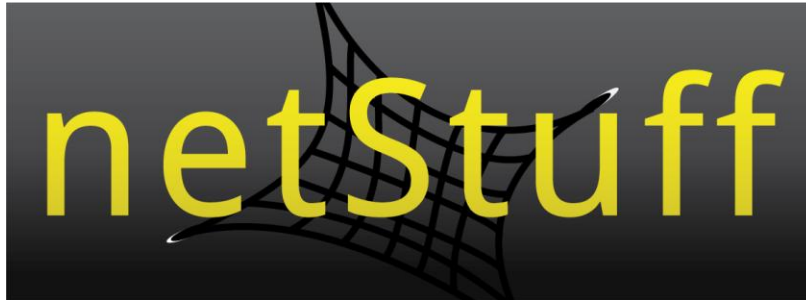
Accomplishments: This week I explored and evaluated the potential of several of my ideas. After determining a project, I determined a company name, netStuff. I also, designed a logo for netStuff. Then I registered for and API key for Netflix and Facebook. Lastly, I completed the preliminary project proposal.

Challenges: A big challenge for me in this phase of the project was determining the company name and logo for the project. Another large challenge was actually defining the traits that made my idea different from all the other ideas similar to netStuff.

Time Spent: 1.5 hours studying the Facebook API, evaluating the complexity of the project. 1 hour studying Netflix API. 1.5 hours critiquing sites and apps similar to netStuff, [Rotten Tomatoes, IMDB, and Flixster]. 3 hours designing a logo. 1 hour brainstorming company names.

Total Time Spent to Date: 8hours.

Goals: Discuss further steps with professor. Begin planning site layout. Get some hands on experience with the APIs.



To: Jan Pearce

From: Damien Russell

Subject: netStuff

Date: 09/11/2011

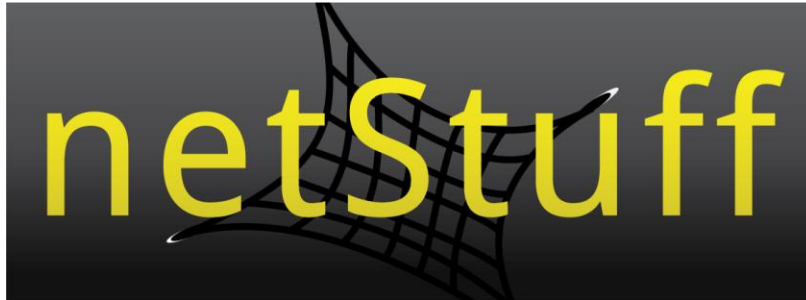
Accomplishments: This week I registered a Domain name for netStuff, gonetstuff.com, as in take your net and go net something, not go to the netStuff website. I spent a majority of my time working this week, studying the Facebook Open Graph API and Facebook Social Plugins. This allowed me to successfully create a test page with a working like button, a recent activity feed, and a Facebook Log In button. Lastly I developed the software requirement specifications for this project.

Challenges: The first challenge this week was remembering how to take an html page and make it a live page. I have done this before; however, it has been several semesters and relearning these concepts on my own, rather than with a class or group of friends, proved difficult. Also this week I had planned to begin working with some of the social plugins from Facebook, to get some hands on experience with the API, but difficulty arose when I began testing anything from the Facebook API. From the API documentation I understood that you can only test things from a live page, which must have an App Key associated with it, obtained by registering the app/website with Facebook. However, one forum I read stated it was possible to test locally, though I have not been able to accomplish this yet. If you visit gonetstuff.com currently, you will find some of the Facebook plugins on an almost blank page. Though I hate to say I have an almost naked page live on the web I can at least be proud in saying my plugins are working.

Time Spent: 5 hours playing with Facebook open graph and social plugins. 1 hour playing with Netflix API. 0.5 hours refining netStuff logo. 0.5 hours registering a domain name and becoming familiar with my domain host.

Total Time Spent to Date: 15 hours

Goals: Begin planning site layout. Refresh my knowledge of SQL. Develop a stronger understanding of the Facebook Open Graph and FQL.



To: Jan Pearce

From: Damien Russell

Subject: netStuff

Date: 09/18/2011

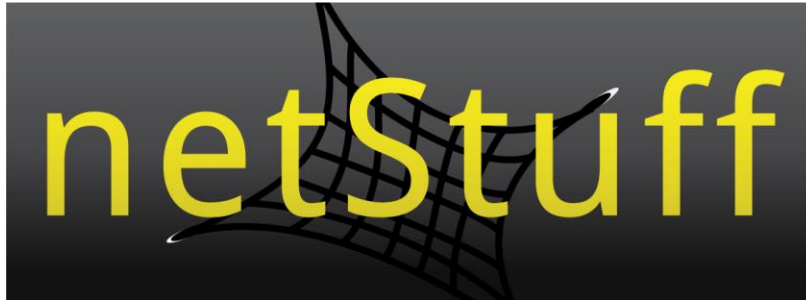
Accomplishments: This week I spent most of my time working on a resume. Though it mostly felt like play I did get to work with the Facebook open graph this week. I can comfortably say I am wrapping my head around the concepts. I also spent some time this week brushing up my SQL skills. Next, I began brainstorming site layouts and developed several thumbnail sketches. Finally, over the past week I have developed a resume, though it is still in need of work.

Challenges: I had two obstacles I had to work through this week. The first was related to my resume. I found it very difficult to come up with my objective. I have had several objectives over the last few months regarding my future. My issue has been I cannot figure out which objective I am more passionate about. I had to spend a significant amount of time thinking about this part of the resume over the last week. The other challenge occurred when I began brain storming site layouts. I do not usually find it difficult to be creative; However, I just could not feel satisfied with the thumbnails of the site I sketched out.

Time Spent: 1 hour refreshing my SQL knowledge. 1.5 hours working/playing with the Facebook Open Graph. 1.5 hours brainstorming site layouts and drawing out thumbnail sketches. 0.5 hours reviewing software requirement specifications though no changes were made as of yet.

Total Time Spent to Date: 19.5 hours

Goals: Obtain feedback from peers on thumbnail sketches. Get a little more hands on with the open graph. Develop and understanding of FQL.



To: Jan Pearce

From: Damien Russell

Subject: netStuff

Date: 09/25/2011

Accomplishments: This week I spent several hours reviewing client-server relationships and Entity Relationship Diagrams; as well as, the fundamentals of http(s) protocol. Next, I reviewed PHP documentation and became familiar with the syntax as well as common application of PHP. Furthermore, I designed the architectural model of the software, found in this document. Lastly, I developed an E-R diagram of the necessary entities and attributes, also found in this document.

Challenges: This week was the first time in the course of this project so far, that I felt intimidated by the amount of work required for this project. My biggest challenge this week was not letting the feeling of intimidation overwhelm me. Another challenge this week for me, was understanding client-server relationships, even though I began this project thinking I had a firm understanding of such relationships.

Time Spent: 2.5 hours reviewing client-server relationships. 1 hour reading documentation on http(s) protocol. 1 hour studying PHP. 0.5 Hours installing WAMP. 1 hour developing software architectural model. 2 hours developing E-R Diagrams.

Total Time Spent to Date: 27.5 hours

Goals: Practice more with php, the open graph, and FQL by writing a few php scripts to test common expressions and queries of the Facebook Query Language.



To: Jan Pearce
From: Damien Russell
Subject: netStuff
Date: 10/2/2011

Accomplishments: This week I accomplished several tasks. I revised my logo after several modifications I felt satisfied. I also planned out a color scheme the site. On top of this I designed the header and footer in Photoshop along with the 'first time visitor page'. With these documents I began to write the markup for the site, as well as style it. I only got too the 'fist time visitor page this week. However, with PSD templates laid out my work flow should help me move quickly during the actual implementation.

Challenges: Time was a challenge for me this week. I wanted to jump right into coding the site. After an hour of thinking and not much more I realized taking the wire frames I sketched up a step further would make writing the markup so much easier. The problem here was I am not being graded for design. I wanted to spend more time coding but felt I could not code a decent amount by Monday without a solid idea of how the site will look when it's finished.

Time Spent: 2 hours tweaking logo. 1 hour reading Facebook API documentation. 3 hours laying out the site in Photoshop. 3 hours coding up the 'fist time visitor page'.

Total Time Spent to Date: 36.5 hours

Goals: Implement the master header and footer. Address issues with the systems design and architecture of the database.



To: Jan Pearce

From: Damien Russell

Subject: netStuff

Date: 10/16/2011

Accomplishments: The first thing I was able to accomplish this week was normalizing my Entity Relationship Diagram for my database. Next, I laid out the design for my body/main content using Adobe Photoshop CS5. Then, I was able to implement my master header and footer. After, finishing the header and footer I scripted the database. Lastly, I spent some time working with php, just tinkering.

Challenges: A big challenge for me this week was floats. Everything I know about html and css I taught myself over the summer. Though I felt I had a firm grasp of cascading style sheets I soon realized my knowledge of floats was not as firm as I thought. Another challenge I faced this week was scripting the database. It had been a while since I scripted a database from the SQL command line. After several beeps from my SQL command line I decided to use PHP my admin to script the database.

Time Spent: 0.75 hours tweaking ER-Diagram. 0.5 hours writing my about the author section. 5 hours laying out the master header and footer. 1.5 hours scripting the netStuff database. 1 hour

Total Time Spent to Date: 44.25 hours

Goals: Layout the markup for the main page and user profile page. Write the php script to store user names and Open Graph Ids when they login with Facebook. Work on scripts to dynamically display data on the netStuff site.



To: Jan Pearce

From: Damien Russell

Subject: netStuff

Date: 10/23/2011

Accomplishments: Though I have put more than 12 hours this week into the project my accomplishments mainly include reading. I first read Book V of VIII from HTML, XHTML, & CSS ALL-IN-ONE FOR DUMMIES by Andy Harris. This help me wrap my head around sever side scripting with PHP. Next, I read book VI from the same book. This was more of a refresher of SQL but I believe it was necessary for me to really understand where I am in this project. Next I got my database moved to the eleven2 sever from my wamp server. Then, I began my class (netStuffDBM) to handle all of my standard data transfer needs when working with the netStuff Database. Then I began to work on the login/registration. I re-explored the 3 various control flows for Authentication of Users and Authorization of the netStuff app/site finally boiling down to the best method, this is also the latest method Facebook is encouraging.

Challenges: I got confused several times this week. My major confusion of the week was the name of my web host when it came to connecting to my database on eleven2.com. The correct host name is localhost on that sever as well, though I really wanted it to be november.unisonplatform.com. It took me almost an entire 24 hours to catch that problem. My next confusion involved the flow of authentication and authorization involving the Facebook severs. When utilizing the latest updates for Facebook“ s JavaScript and php api“ s you can easily manage session states for all you users and greatly simplify the login process. However, there is an issue at line #37 of the Facebook.php document under the php-sdk folder. This php document starts a session. When utilizing the JavaScript/php flow the JavaScript allows Facebook to also start a session for the user. I dealt with „cannot send session cache limiter – headers already sent...“ for almost an hour before I found the solution at <http://gitHub.com/Facebook/php-sdk/issues/366>. Due to time I had to submit what I had at 3 a.m. and call it a night. One syntax error is holding everything up at the moment. –Line 92 index.php-

Time Spent: 6 hours reading Server Side Scripting with PHP by Andy Harris. 4 hours reading Managing Data with MySQL, also by, Andy Harris. 2 hours re-reading Facebook documentation for User Authentication, App authorization, and App Authentication. 1 hour working on netStuffDBM Class. 3 hours re-exploring control flows for authorization and authentication. 1 hour correcting several things in the project’s application development section.

Total Time Spent to Date: 61.25 hours

Goals: Fix bug with query built from openGraph data returned upon login/authentication. Build scripts to gather user movies and data on movies.



To: Jan Pearce

From: Damien Russell

Subject: netStuff

Date: 10/31/2011

Accomplishments: I was able to read most of the chapter on JavaScript in HTML, XHTML, & CSS ALL-IN-ONE FOR DUMMIES by Andy Harris. Next I was able to develop a test plan. After, I developed each test case mostly based on the software requirements written earlier in the semester.

Challenges: This week working on anything was a challenge for me. I spent most of my week stressing about everything in my life. Honestly I felt like I was not going to make it through the week or weekend. Just trying to be productive was difficult for me. I could not get my mind into anything I was doing. I reached a wall that I have been trying to get over for two weeks. I think I finally feel distressed and back on my feet. I had to get away from it all and it put me behind. However, I think it was necessary if I am going to get my act together to graduate in the spring, and have a functioning site to be proud of. When I did work this week a more specific challenge for me was fql and the openGraph. Working with the graph on the developer site is really straight forward. It gets a little more difficult when querying from my server. I have to have access tokens for the user and for the app.

Time Spent: 2 hours reading the chapter on javascript. 2 hours working on a query to get into the open graph. 3 hours working on test cases and a test plan.

Total Time Spent to Date: 68.25

Goals: Continue building FQL quires for movies on the openGraph.



To: Jan Pearce

From: Damien Russell

Subject: netStuff

Date: 11/14/2011

Accomplishments: This week I was able to implement a function to get all user movies from the opengraph. This function also parses through each returned film, searching the opengraph with the film ID. If it find the film it begins to build a query string with the appropriate fields if available if the field is not available the field will be set to null. I have one error holding everything up. I was able to move the algorithm for building a string of movie's not in the database, to its own file, registration.php. With and include statement this script is executing but I have an error preventing all of them from being submitted to the database.

Challenges: I was having trouble breaking up the multidimensional array of user films. After some discussion with Andrew, I was able to solve the problem really easily with a foreach loop. Another challenge for me this week was submitting a dynamically built string of the appropriate fields for each movie. I could build the string with every film and the appropriate fields. However, when the query executed I received an error saying I had a duplicate primary key. Which is the ID for the movie. The confusing part is the id doesn't match any of the films.

Time Spent: 3 hours reading a book on Ajax connections. 2 hours reading FB documentation. 3 hours designing and implementing an algorithm to build a dynamic SQL query of Movies to send to the database.

Total Time Spent to Date: 76.25

Goals: Get the movies in the database. Get the movieID's and userID's in the watched table. Complete my about.php page.



To: Jan Pearce

From: Damien Russell

Subject: netStuff

Date: 11/20/2011

Accomplishments: The first accomplishment of the week was submitting the movie's to the movies table with the correct movieID's.

Challenges: Tuesday before class I was still dealing with the error of submitting the wrong ID for a movie into my database. The issue happened to be the size of the movieID. Though I had the movieID field set to 255 characters. This was not enough to represent an integer with a length of 10 or more. After Renaldo suggested I use a string, using PHP myAdmin, we realized Big Int would solve our problems.

Time Spent: 4 hours debugging trying to solve the mystery of the wrong movieID making it to the database.

Total Time Spent to Date: 76.25

Goals:



To: Jan Pearce

From: Damien Russell

Subject: netStuff

Date: 12/12/2011

Accomplishments: Though this is a much over due memo. I just finished the site. I was able to get the movie data for each film like/shared on Facebook by a user, into my database. After accomplishing this I began working on implementing graphs to display the data for each user. There were several issues at this point, see challenges for more detail. In the end I gave up on the Kendo UI graphs, though my heart was set on these graphs. I needed to represent the data in some way to move this project into the completion stage and to actually make a demo, in the end I switched to tables because of a time crunch. I do have a profile page which pulls the user photo from the open graph on a page load, as well as totals for the actors, directors, and films for a user.

Challenges: Some major challenges have prevented me from completing anything in the last week. First, I had corrected the size of movie and IDs to accommodate for IDs bigger than a normal INT by changing the field to a BIG INT. Later when testing and creating new Facebook dummy accounts for testing, the UserID of those accounts also required a BIG INT. It took a while to catch that issue. The next major challenge for me has been trying to implement the Kendo UI graphs Andrew Smith had shown me earlier in the semester. Finally the final challenge for me was correcting the style sheets so my FB login button was clickable. At one point the only place to click the login button was approximately a 2px by 2px section of the button. In the end time has gotten the best of me and some of the features I hoped to put in the site do not currently have functionality; however, the main functionality of the site and the overall goal of the site, retrieving user's movie data and analyzing it, is working.

Time Spent: 4 hours working on MySQL error before realizing userID's needed to be BIG INT's. 5 hours polishing up css and site navigation (footer links and such). 1.5 hours creating Contact page. 1 hour creating About page. 10 hours trying to implement kendo ui graphs. 2 hours creating scripts to retrieve user genre, actor, and director data from netStuff database. 2 hours creating tables to display data from database. 2 hours creating Facebook logout button. 1.5 hours working on a demo. (31 hours)

Total Time Spent to Date: 107.25 hours

Goals: Have a great poster session/presentation!