Comments on the Talks

• Enjoyable, wide range of topics, activities
• Need to do a better job of establishing need
  • Explaining what solution is
  • Why this is a better (the best) solution
• Still working on design
  • What not How
  • What technologies you plan to use can change
  • Choose technologies based on final design
• Design presentations (UI) in a week and half in Sayles
Final Exam

• The final exam in this course will be take-home
  • Available on 5/8
  • Due 5/15
  • Shouldn’t take more than 2-4 hours
• The final is not a collaborative project
• There will be separate questions for designers and concentrators
Node.js isn’t the only Back End

- PHP
- Servlets :: Using Java for the back end directly
- .NET :: Using C# for the back end directly
- GWT :: Using Java for the front and back end
- Content Management Systems (Monday)
- Django, Flask
- Ruby on Rails
- Embedded servers (nanohttpd, ...)
- ...
Which of the following is not a feature of PHP for web applications?

A. It includes built-in templating of web pages using <?php ... ?>
B. It has a large collection of libraries
C. It is a type-less scripting language
D. It can be used directly with the Apache web server
E. All of the above are features.
Simple PHP Web Site

  - Show the web page in action
    - Is there a prime consisting of all 1’s > 11?
  - Look at source
Primes.php

<html>
<br>
<?php
require("primetest.php");
$num = $_REQUEST['number'];
if ($num == '')
else if (testPrime($num)) echo "$num is PRIME"
else {
  echo "$num is COMPOSITE\n"
  while (!testPrime($num)) $num = bcadd($num,1);
  echo "<br>$num is the next prime\n"
}
?>
<!– Rest of the page goes here –>
</html>
Prime Numbers

• What are they?
• Why are they useful?
• How do you compute them?
  • Sieve of Eratosthenes
  • What about large numbers?
Rabin-Miller test

• Given an odd integer \( n \)
  • Let \( n = 2^s r + 1 \) with \( s \) odd
• Choose a random \( a \) \( 1 \leq a \leq n-1 \)
• If \( a^s \equiv 1 \pmod{n} \) or \( a^{2j} \equiv -1 \pmod{n} \)
  • For some \( 0 \leq j \leq r-1 \)
  • Then \( n \) passes the test
• A prime always passes the test
  • A composite has a 50% chance of passing
  • How can you tell a prime from a composite?
• Sounds easy, but you need to do infinite-precision math
  • There’s a library for that
function testPrime($n, $ct = 10) {
    $v = bccomp($n, 2);
    if ($v == 0) return true;
    if ($v < 0) return false;
    if (bccomp($n, 2) == 0) return false;
    for ($i = 0; $i < $ct; ++$i) {
        $a = bcmod(rand(), $n);
        while ($a == 0) $a = bcmod(rand(), $n);
        if (!testOnce($n, $a)) return false;
    }
    return true;
}

function testOnce($n, $a) {
    $n1 = bcsub($n, 1);
    $r = 0;
    while (bccomp($n1, bcpow(2, bcadd($r, 1))) == 0) $r = $r + 1;
    $s = bcdiv($n1, bcpow(2, $r));
    $t1 = powermod($a, $s, $n);
    if ($t1 == 1 || bcsub($t1, $n1) == 0) return true;
    for ($j = 1; $j < $r; ++$j) {
        $t2 = powermod($a, bcmul(bcpow(2, $j), $s), $n);
        if (bcsub($t2, $n1) == 0) return true;
    }
    return false;
}
Sample PHP Web Sites

- [http://eadotec.cs.brown.edu](http://eadotec.cs.brown.edu)
  - Purpose
  - Show the site
  - Show some of the code
- Code bubbles file upload (editor)
- Facebook
Frameworks

• Web applications are a lot of work
  • Much of it is busy work
• Routine chores and programming
  • Administration
  • URL -> command mapping
  • Providing HTML pages
  • Data structure mapping
• Frameworks try to simplify this
Data Manipulation

JavaScript Data ➔ ??? Data ➔ SQL/NOSQL Data

Web Browser       Web Server       Database

CS132 Lecture 18: Other Server Technologies
Model-View-Controller

View

Controller

Model

Data Display

Data Management

Data Store

Web Browser

Web Server

Database
Model-View-Controller

Data Display

Data Store

Data Management

View

Model

Controller
Model-View-Controller

• Basic idea is to separate the display, the data, and the logic
  • Each can be change independent of the others
• Exactly how this is done various from case to case
  • Some do it with a common data abstraction
  • Some do it with callbacks
  • All call themselves MVC
• Different people mean different things
DRY Principle

• Don’t Repeat Yourself
  • Every piece of knowledge must have a single unambiguous authoritative representation within a system

• Why have 3 different representations of the data
  • More code to maintain
  • More code to change when data changes
  • More chance for bugs
Django and Ruby on Rails

- Widely used
  - Django: Instagram, Pinterest, ...
  - Ruby/rails: GitHub, Basecamp, ...
- Similar frameworks exist (e.g. Flask)
- Mostly a back end technology
  - Can be paired with a templating engine
- Require knowing Python/Ruby
  - In addition to JavaScript, HTML, ...
Django/Ruby Frameworks

- Express-like dispatch
  - Based on static tables, not executed code
  - With functions to handle the results
- Logic to control deployment, server setup, etc.
- Libraries to handle common web app features
- Simple connection to database
- Simplified Data Management
DJANGO and Ruby/Rails

- Map from internal objects to SQL automatically
  - Changes in the object -> SQL updates
  - Objects created automatically from SQL database
  - SQL Tables created automatically from object definition
  - Changes to object definition change the database
- Map from internal objects to HTML automatically
  - Using templates
- Map from internal objects to JSON automatically
  - Changes in the object -> go to web site if needed
- **OBJECT-RELATIONAL MODELING**
Object-Relational Modeling

• Not limited to Django-Ruby
  • There are libraries to provide some of this functionality
• Not limited to SQL back ends
  • NoSQL databases can be used as
    • Direct mapping to object from json
  • Cache the current state in memory as objects
    • This allows fast query at times
  • Update updates memory and the database
• What is the problem with this?
RESTful Web Applications

- Client-Server model
  - Client handles presentation, server handles storage
- Stateless
  - Add data needed for request is passed

"The promise of REST being dealt with."
RESTful API HTTP Methods

• Collection API .../collection
  • GET : return list of elements in the collection
  • PUT : Replace the entire collection
  • POST : Add an entry to the collection
  • DELETE : Delete the entire collection

• Element API: .../collection/:item
  • GET : Retrieve the given item
  • PUT : Replace or create the given item
  • DELETE : delete the given item

• Action API: .../collection/:item/verb
DJANGO/RUBY with REST

- URL identifies the object in the server
  - What field to access or change
  - New value of the field (using PUT)
- Front end makes changes to long term objects
  - By sending POST requests
- Front end gets current state of objects
  - By sending GET requests
AngularJS

• Templates that are executed at run time
• Automatically update the page as values change
• MVC (Model-View-Controller)
  • Model = the data structures
  • View = the template
  • Control = commands that modify the data
• Combine this with Object-Relational Modeling
  • Make a simple, consistent web application
Next Time

• Monday: Guest lecture on CMS and the Brown CS web site
• Wednesday: HCI Introduction and Web Site Navigation
• Friday: Guest (TA) lecture on design
• Monday/Wednesday: Poster Sessions (Sayles)
• Friday: Design lab
JavaScript

- JavaScript was designed for writing simple things
  - Short routines to check inputs
  - Short routines to provide interactivity
- Not necessarily the best language for large systems
  - No name spaces or module structure
  - Prototype-based object model
  - Easy to write erroneous programs
    - No compile time
    - No type checking
  - Improvements in Node.JS, not in the front end
- jQuery, angular, and other libraries help
  - But really aren't enough
GWT

• Google wanted to write large JavaScript Programs
  • Java was the language their programmer’s knew
    • Why not write in Java
    • Compile into JavaScript rather than byte codes
  • Other similar approaches now available

• Is this a good idea?
  • Is Java a good match for JavaScript
  • Is Java more powerful than JavaScript?
  • What problems do you foresee?
Java for JavaScript Problems

- Java Object Model
  - JavaScript prototypes can be used to model classes
  - Both Java and JavaScript do garbage collection
- Thread and synchronization
  - JavaScript is single-threaded, so just don’t bother
  - JavaScript assumes everything is reactive
    - Everything is done via callbacks
      - Callback has to return for the browser to update
      - Java doesn’t make this assumption (or does it?)
  - Use a Swing-like model
    - Initialize, register for callbacks, process callbacks
Java for JavaScript Libraries

- Java programmers depend on standard libraries
  - `Java.util.*`: all the data types
    - JavaScript provides associative arrays via objects
      - Can be maps, lists, arrays, sets, ...
    - The compiler maps calls to these to JavaScript
  - Do this for most of the common libraries
    - Note that I/O is not valid in the browser
    - And other thing might not make sense
Java for JavaScript Graphics

• Java uses Swing and AWT libraries
  • These aren’t appropriate for the browser
  • Need to have something to create and manipulate pages

• Create a new library based on HTML
  • Labels are just HTML
  • Active widgets correspond to HTML input objects
  • Layout widgets of various types (based on tables)
  • Style internally or via style sheets
  • Can escape to pure HTML (not encouraged)
Calling the Server

- Assume we use servlets to handle the calls
  - Code should look like a function call to the server
  - What is wrong with this?
- Server calls are asynchronous
  - User can do things while waiting for response
  - JavaScript has to return control to the browser to process
- Server calls can fail
  - Server or network errors, etc.
  - Front end needs to handle this
Asynchronous Call Mechanism

• Specify the call, the arguments, and a callback object
  • Creates a json/xml message representing the call
  • Special servlet unbundles this and makes the call
  • Result of the call is encoded and sent back
    • Either as success or an exception

• Callback object is invoked when the call completes
  • Separate methods for success and failure
  • Can retrieve returned value and continue
S$^6$: A Sample GWT Application

- Show [http://conifer.cs.brown.edu/s6](http://conifer.cs.brown.edu/s6)
  - What has to be done and when
  - Why should this be done in the front end?

- Code Structure
  - Multiple directories
    - Public: CSS style sheets, default page
    - Server: Calls that can be invoked
    - Client: the actual web page code
    - Other: web.xml

- Client code
  - Layout out a test case
  - Calling the test case
  - Wait code for repeats
GWT Debugging

• You can compile and check code with Java compilers
  • Write in eclipse/code bubbles/netbeans/...
  • Just as you do for the backend, server, ...
• Can compile and test using a dummy server
  • Provided with GWT
  • Using a Java debugger
  • Integrated into Eclipse
GWT Pros and Cons

• When would you use this?
  • Want a highly-interactive front end
  • When the amount of front-end code is large
  • When providing a web front end for a Java application
  • When you are most comfortable coding in Java

• Why not use it
  • Overkill for most applications
  • Requires compiling and installation
  • Generated JavaScript code is hard to debug itself
  • Requires TOMCAT (servlet support) for the web server
  • Limited use of HTML
  • Limits access to things like D3, Jquery UI plugins, ...
Servlet Overview

• Web application logic can be complex
• Want to be able to write arbitrary Java code
• Let the web site call this code
  • In the normal way a web server is called
  • Send a HTTP request, get a HTTP response
    • Response can be HTML, XML, JSON
    • Can be page replacement or AJAX
• Then you don’t need a separate server
  • Or complex background process
  • At least not if your code is simple
Servlet Example

import java.io.*;
/**
* Servlet implementation class ServletDemo1
*/
@WebServlet(description = "course demo",
urlPatterns = { "/ServletDemo1" })
public class ServletDemo1 extends HttpServlet {
private static final long serialVersionUID = 1L;
public ServletDemo1() {
}
public void init() throws ServletException {
}
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    PrintWriter pw = response.getWriter();
    pw.println("<html><head><title>");
    pw.println("Page Title Goes Here");
    pw.println("</title></head><body>");
    pw.println("<h1>Display Title Goes Here</h1> ");
    pw.println("Page Body goes here");
    pw.println("</body></html>");
    pw.close();
}
// end of class ServletDemo1
J2EE

- HttpServlet is part of J2EE
  - Not in standard Java
- J2EE is an extended set of classes for web applications
  - Provides the framework-style functionality one needs
    - Not as easy as rails/django
    - No database mapping to objects
    - No automatic configuration
  - Environment (e.g. eclipse) can do the configuration
  - Sophisticated applications need sophisticated DB access
    - Transactions, views, logging for recovery
J2EE Capabilities

- Web application support
  - Sessions, cookies, XML, JSON, etc.
  - HTTP Response generation
- Mail
  - Sending/receiving messages, reading mailboxes, etc.
- Transactions
  - Separate from DB transactions, but including them
- Remove procedure calls using the web
  - SOAP
  - Web services: for implementing and using these
- Messaging
  - Sending complex messages to other processes
Java Server Pages

• JSP: Java server pages
  • Instead of having Java output everything
    • Embed the java calls in a web page
    • Much like PHP
  • <%% ... %%%> for embedding
  • Still access application java classes
  • Still managed by TOMCAT
TOMCAT: A Java Environment

• Uses class loaders to control access to a class
  • Can distinguish between same class in two apps
  • Can distinguish between same class in two versions of an app
  • This lets applications be loaded or unloaded
  • This lets multiple applications run at once

• Front end interacts using an HTTP interface
  • Runs on a separate port (8180)
  • Looks to the browser as another web server on the same host
TOMCAT Services

• Load web applications
  • The first time it sees a request for a servlet
• Handle mapping of URL requests to calls
  • Keep track of what servlets are running
  • Each is represented by a HttpServlet object
• Partition by applications
  • An application is a set of class files
  • Some of these files implement HttpServlet
  • These class files can access each other
    • And standard libraries
    • But not class files of other applications
Servlet Problems

• How long does the web server run?
  • How often is the web application changed
  • Can you change classes dynamically in Java
  • What if there are multiple applications on the same server

• How many web applications does the server manage
  • How much code is associated with each page
  • Can Java codes from one app interface interfere with another?
  • What about having two classes with the same name
TOMCAT: URL Mapping

• TOMCAT maps a URL to a Java Call
  • Need to know the class and method name
  • Need to find the appropriate object
  • Need to know what (typed) parameters to send in
  • Need to know how to get HTTP response out

• This is done using the HttpServlet abstract class
  • User class extends this class
  • Class defines initialization, termination
  • Class defines a method for GET, POST, PUT, ...
  • In J2EE (http://<DOC> page)
Defining TOMCAT Applications

• An application is defined using a WAR file
  • Web archive
  • Like a jar file

• Manifest file (jar manifest)
  • Provides security and startup information

• WEB-INF directory
  • Includes a web.xml file that defines the servlets
  • Includes the code for this application (/classes)
  • Includes other code /lib/*.jar