CS1320 Creating Modern Web and Mobile Applications Lecture 1:



Course Introduction

1/26/20

Welcome To CSCI1320 (CS132)

- Professor (<u>spr@cs.brown.edu</u>)
- HEAD TAs (<u>csl32headtas@cs.brown.edu</u>)
- TAs and ETAs (<u>csl32tas@cs.brown.edu</u>)





Everyone today uses the Internet

- On-line shopping from
- Browse social media
- Read email
- Do web searches
- How many do on-line research (in place of using the library)?

• Do you use an app or a web browser?

Who Has Created a Web Site?

- Static or dynamic?
- What technology did you use?
- How often do you update it?



If you have problems installing or compiling code bubbles, contact Email: me.

The slides for the SoftVis keynote on The Visualizations of Code Bubbles are available (warning: this is a large file).

- There are now ancient (25-30 year old) videos linked from this site for various of the systems (Pecan, Field, Garden, Desert, Cacti, and Bloom). See what the state of the art was back then and compare it to today's tools.
- The Code Bubbles environment now includes automatic error correction (spelling, syntax, imports, ...). The environment has been upgraded to work with the latest
 versions of Eclipse (which was problematic on the Mac). The environment is also available for editing JavaScript files for Node.JS development.
- My current status (displayed outside my office and updated automatically)
- The Code Bubbles programming environment is now available in both source and binary forms. See the web page at http://www.es.brown.edu/people/spr/codebubbles for more information.
- + S^6 code search tool is now available. Try the <u> S^6 web interface</u>.
- DYMEM memory visualization tool is available as part of DYPER.
- DYPER dynamic performance analysis tools are available.

Research Projects

The overall theme of our research is making programming easier. We are undertaking a number of projects directed at this goal. These include:

- Programming Environment (PECAN CARDIN FIELD DESERTCLINE RUBBLES) We have a long history of developing programming environments and the accompanying tools and techniques. Our
 recent efforts center avoud the Code in Bubbles environment. Code Bubbles as in efforts to provide better, working e-sect entric, user interface for programming. If lets the user easily view the variety of software a total techniques. Our
 artificits that are relevant to their current task the same time on the same screen. I provides powerful and efficient navigation facilities and a variety of software visualizations. Recent work within the
 environment has looked at automatically correcting problems as the user edit.¹
- Automatic Bug Repair Our efforts in this area concentrate on using large open source repositories such as GitHub to assist in fixing semantic errors in programs. We are also looking at how to make automatic
 bug repair practical by integrating the user into the process.
- Performance Measurement This work is summarized as part of the <u>DYVISE</u> framework which includes low-overhead performance monitoring tools with appropriate visualizations and a tool for automatically constructing automatic performance models of complex Java systems in order to do predictive what-if analysis.

Creating a Modern Web Application

- Think of simple shopping site
- Would you feel confident creating it

 What do you see as the difficulties
 How much work would it be
- You should be able to do this • By the end of the course



What is a Web or Mobile Application

- A program that the user interacts with through the Internet.
 - o Interact via a browser
 - o Or a mobile front end
 - o Using standard protocols (HTTP)
 - Where part of the program runs on a server
 - o (Where the program uses a database)



Web Pages & Graphic Files

Template Pages, Code & Data

Sample Web and Mobile Applications

Name some web and mobile applications

 Which you like to use
 Which you don't like to use







Lecture 1: Course Introduction Web Application Architecture





Elements of Creating Web/Mobile Applications

- The importance of Human-Centric Computing
- Distributed Programming
- Security and Privacy
- Scalability
- Evolution
- Software Engineering (specs, design, testing, ...)









SECURITY

What's Involved in Web/Mobile Apps

honesty integrity responsible professional trust customer-led competitive collaboration quality teamwork innovation

- Requires **understanding**
 - The application, the users, and the needs of the users
- Requires design expertise
 - o User interface design, usability, scalability, maintainability
- Requires sophisticated programming skill
 - Handling 10,000 users; 3-5 9's of up time; updatable
- Requires **programming expertise** in several areas
 - o Interactive, Large-scale server technology, distributed programming
- Typically requires a **development team**
 - o Designers, programmers, testers, **users**

In This Course

- You are going to build a real web or mobile application
 o For real users
 - o In teams with mixed skills
- You are going to learn the basics of web and mobile applications
 - Won't become an expert in all of them
 - Will learn the alternatives, terminologies, etc.
 - o Will learn enough to build your own application if desired
 - Will become an expert in some aspect for your project
- You are going to learn to work in teams



Harbinger Systems

teal-time Collaborative

CSCI1320 has Two+ Tracks

Concentrator's Track

- For CS students with programming background (CS32/CS33)
 - How necessary is CS32/CS33
- Emphasis on programming skills
- Responsible for programming aspects of projects
- o Different levels of programming in the projects

• Designer's Track

- o For students with design skills
- o Limited or no programming experience required
- o Emphasis on web and mobile design and learning how it can be used
 - Assignments are design-oriented not programming-oriented
- o Responsible for human-centric aspects of projects
- o Please email cs132headtas@cs.brown.edu



CS132 Has Two+ Tracks

Others Track
For students without significant programming experience
Who don't know HTML/SaveScript
Who have little design experience



CS132 Has Two+ Tracks

- Capstone Track
 - o Students taking the course as a capstone
 - o Expected to either
 - Propose and supervise a project
 - Serve as the team leader on their project
 - Both
- CS Design Track
 - o Mix concentrator and design assignments to maximize knowledge gained
 - o For CS students with a strong design bent
 - o Work on design aspects of final project



CS132 Has Two+ Tracks

- Entrepreneurship Track
 - Student(s) propose a project that is basis for a startup
 - o Initial proposal is for MVP
 - o Elevator talk, poster, presentations
 - o Build MVP (prototype) in the course, develop product over summer



Course Mechanics

• Laptops / Phones

o Used in lab classes o Shouldn't be used in lectures

• SEAS, extensions, late days





Course Contents

- There are three parts to the course
 - 1. Learning the fundamentals of web & mobile applications
 - 2. Learning the **basics of building web & mobile applications**
 - 3. Creating a web or mobile application for a client
- Reflected in time commitment and grading



Course Contents



Fundamentals of Web Apps

- There are lots of different web and mobile technologies
 - o More than we can cover in one course in any depth
 - o But a web app or mobile expert should know **of** them all
 - What they are, what they are good for, how they work, ...
 - You need to know what to use
 - You need to be able to talk to clients and others
- There are lots of things to consider in designing and building a web or mobile application

 Security, human factors, universal access, testing, design, ...
 You need to understand and deal with these

Expertise

Fundamentals of Web and Mobile Apps

- Covered in lectures, homeworks, tutorials & labs

 I'll try to make this accessible to both tracks
 Questions and comments are encouraged
 PLEASE !!!!
- Checked w/ homeworks, labs, participation (23%)
 All should be relatively easy if you come to class
- Tested in the Final exam (10%) o Take-home



Basics of Building Web and Mobile Apps

- Understand a specific set of technologies
 - o HTML5/CSS
 - o JavaScript
 - o Vue
 - o NativeScript
 - o Node.JS / SQL and NoSQL / AJAX
- These will be covered by 5 programming/design assignments
 - o Each one to two weeks
 - o Separate assignments for the two tracks
 - o Count for 35% of your final grade



Web or Mobile Project



- We have gathered a suite of projects from real clients
 Mix of commercial, non-profit, local
- Based on your preferences we will assign teams
 - Four people, mixed backgrounds, apt for project
- Teams should meet weekly with sponsor & mentor TA
 - Keep them happy
 - o Teams should have a leader and a sponsor contact person
- You will have opportunities to present your project
- Counts as 32% of your grade
 - Grading based on project itself, presentations, milestones, sponsor feedback

Student Projects

- We are going to allow a limited number of student projects
 - o If you have a web or mobile application you really want to create
 - o Proposer will act as project mentor
- These need to be well-defined and scoped
 - Should be something different (not another scheduling application)
- If anyone is interested in doing one of these
 - o We need a detailed proposal for it soonest
 - Today preferred, Saturday at the latest
 - o Talk to the TAs for advice and suggestions



Software Engineering in the Project

- Different programming languages and models
- How to work in teams
- How to work with clients
- How to work with deadlines
- How to organize a larger project
- How to plan for evolution
- How to plan for problems





Lecture 1: Course Introduction

Project Schedule I

- 1/28: Initial project preferences out; due 1/31
- 2/05: Final project teams announced
- 2/14: Initial client report
- 2/24: Project specifications hand in
- 3/02: Project Elevator Talks
- 3/06: Potential user feedback reports hand in
- 3/09: Initial project design presentation to TA
- 3/16: Project front end design hand in
- 3/16, 3/18: Project Poster Fair



Project Schedule II

- 4/10: Project implementation design hand in
- 4/20: Project prototype up and running (target)
- 4/24: Prototype feedback from client
- 4/29: Project testing reports due
- 5/04: Final project presentations (whole day)
- 5/08: Final project hand-in
- 5/08-5/12: TA meeting with project team



Collaboration Policy (Homework 0)

- Please download from the web site, read and sign, hand in
 - You won't be assigned to a project unless you do
- We expect you to do your **own work** on the 5 programming/design assignments
 - o Not copy from others
 - o Not copy from the web
- Much of the rest of the class is collaborative
 - Except for the final exam
- We will detect cheating
 - When in doubt about using something, ASK.
 - Always cite any external code, references, ideas, etc.
 - Always include external copyrights, etc.
 - o Several students got directed NCs for the course in the past



Intellectual Property (IP)

- You own your code (Brown's policy)
 - In a group project, this is generally shared ownership
- When you are working with others (sponsors), they have rights too
 - o Non-exclusive perpetual right to the code and its use
 - o Complete rights to any images, etc. they provide
- You should negotiate/agree with sponsor on final rights
 - They might want code open sourced (must be in project definition)
 - o They might want documentation on maintenance and use
 - o They might want code non-programmers can easily modify
 - Do this early in the process
- Some projects are constrained
 - o Take this into account in choosing projects



Hours

• TA Hours will be announced

- o Based on homeworks, assignments, etc.
- o Each project will have an assigned mentor TA
 - You are responsible for setting up meetings

• My office hours

- o Monday, Thursday 1-3 (tentative)
- o Open office policy (8:30-3:30)
- o I'm hoping to see everyone at office hours at least once
- o I'm usually free even if the TA hours are very busy



Lecture 1: Course Introduction

Course Web Site

- http://www.cs.brown.edu/courses/csci1320
- Reference Materials
 - o Links to that other material, cherry-picked
- Calendar
- Keeping up to Date
- Piazza
 - o Information for opting out

Home (CSCI 1120 x +	- 🗆 ×
→ C △ = csbrown.edu/courses/csci1320/	ŵ 🖸 🧕 E
Apps ★ Bookmarks 🔞 Debian.org 🔞 Latest News 🔞 Help 🚯 localhost/800/Hie/ 🥥 steven_reiss@brow 🍈 Tools Help Androi 🕲 SmarterThings Rule 📒 Imported from 1	IE 🛛 » 📔 Other bookmarks
	ĺ
CSC 1320	
NOME HOME LECTURES ASSIGNMENTS FINAL LABS NOVES STAFF	Pocs
WELCOME TO CREATING MODERN WEB & MOBILE APPLICATIONS! TIME & LOCATION: MWF IOAM-IO-ISOAN IN 83 WATERMAN, ROOM ISO	4
If get relations receives in solutions of proposal for a final project, plass visit in range, for more holmatic CSCI 1320, Creating Mademi Web and Mobile Applications, is a spring semester course within the Brown department. The course has two tracks, one intended for CS concentrators, and one intended for non- concentrators with previous design experience. It takes a holistic look at the process of developing web and mobile applications and alims to bring the students to a point of mastery of many of the most used technologies and development proceices. The course includes a semester long group find project in which the students will be working with external companies, non-profits, and other organizations. Plazze: <u>Link</u>	on CS Ind

Questions regarding the Course





Next Time

- The Web Front End: The Browser, HTML, CSS
- Homework:
 - o Assignment 0
 - Available on web page, due Friday 1/31
 - Collaboration policy
 - Account setup, etc.
 - o Preliminary work for Lab 1 (due Wednesday)
 - Available on web site
 - Get started now!!!

Problem

- You have been hired to work on creating a system for web-based course registration, lets call it Banter.
 - What are the problems you would anticipate?
 - How would you proceed?
- What do you see as the potential problems
- What would you work on first
- How would you sell it



Human-Centric Computing

- User interface design
- Ease of use
- Looking good
- Accessibility and internationalization
- These make or break a web application



Distributed Computing

- Web applications are inherently distributed
- They use facilities outside of programmer's control
- They are written in a multitude of languages
- Communication is asynchronous
- Frameworks try to make this simpler
- Nothing is standard



Security and Privacy

- Are major concerns
 - o In the press daily
 - Your application is exposed to the world
 - o All types of attacks are possible
- Same interface used by multiple users at once
- Multiple applications might run on same server
- Private data needs to be secure
 - o Especially sensitive data (e.g. credit cards, health data (HIPAA))
- Applications often have real-world implications (\$\$\$)
- Liability issues arise



Scalability

- How many users do you expect to have
 After you've been slash-dotted
 On Cyber Monday
- Handling 1000 users at once is hard
 - o Handling 10,000 requires a different approach
 - o Handling 1,000,000 requires rethinking the application



Evolution

• Web apps need to change

- o The look gets stale after a year (more or less)
- o New functionality desired
- o Users expect new features, new look and feel
- o Need to keep up with competition

• Different form factors and capabilities

- o Different browsers
- o Tablets, phones, watches and other devices



