CIT 382
Web & Mobile Applications Development II
Winter 2016 — CRN 2167 — 04 Credits

UO Catalog Description

Prerequisites
CIT 281 Web Programming

Instructor
Michael Hennessy, CIS
Office: 158 Deschutes
Phone: 541-346-3487
Email: michaelh@uoregon.edu
Office hours: Available in our Canvas course.

GTF
Zach Schmidt
Office: 239 Deschutes
Email: zschmidt@cs.uoregon.edu
Office hours: Available in our Canvas course.

Meetings
Class: 1200-1320 MW, 101 Jacqua
Plus, one weekly lab in 026 Kla.

Required Textbook
Lingras et al, Building Cross-Platform Mobile and Web Apps, Menage Learning (2017)

Print format: Purchase at the UO DuckStore.
Digital eBook: Read the publisher’s step-by-step guide to purchasing the online eBook:
https://www.cs.uoregon.edu/Classes/16W/cit382/syll/purchase-eBook.pdf

Course Overview
CIT 382 is a second course in cross-platform Web App Development, and uses a mobile-friendly development strategy. 382 continues where CIT 281 left off.

In the 21c., every business— from fashion to finance, from large to small – is a tech company, and many have apps. Therefore, every business needs developers, and managers with a solid knowledge of development.
Even if you do not intend to be a developer, learning WebDev concepts and skills will qualify you to work in any Fortune 500 IT division, and/or make key contributions at a startup, including hiring decisions.

The following quote from the preface of our textbook provides an overview of the course:

Past app development process tended to be specific to a platform in the native application development environment such as Objective-C or Swift for Apple iOS, or Java for Google’s Android. While some apps still need to be developed in the native development environment, the emergence of HTML5 and JavaScript-based jQuery and Node.js makes it possible to develop apps that can run on most communication devices.

These technologies help businesses with limited resources to keep up with the recent shift to mobile technologies. Businesses still need to support the desktops/laptops/netbooks platforms. However, mobile devices such as tablets, smartphones, smart TVs, and phablets increasingly serve our personal and home information technology needs, and as such are equally and in some cases more important.

In order to address a fragmented computing, information, and communication device market, we need to adopt a mobile-first or mobile-friendly development strategy. The major web browsers for desktop/laptop/netbook computing such as Internet Explorer, Firefox, Chrome, Safari, and Opera that run on Windows, Mac, Linux, and Chrome OS now also include versions for smartphones, smart TVs, and other mobile devices that run on Android by Google, iOS by Apple, Windows RT by Microsoft, as well as several other competing mobile platforms.

382 uses the following technology stack:

- Client-side (HTML, CSS, JavaScript, JSON, jQuery, jQuery Mobile)
- Server-side (Node.js, Express)
- NoSQL databases (MongoDB)
- JavaScript Templating (Handlebars.js)
- Relational Databases and SQL (MySQL)
- Platform-As-A-Service (PaaS: Cloud Foundry, Heroku, AWS)
- Hybrid Mobile Development (Cordova/PhoneGap)
- Workflow Tools (Unix shell, Sublime Text, SSH/SFTP, Git, Github)

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and software in the data centers that provide those services. Key terms: Software as a Service (SaaS), Platform as a Service (PaaS).

Bring a Computer to Class

This is a CIT course, so you are encouraged to bring a computer to class. If you do not own a portable computer, UO notebook computers are available for two-hour checkout at the main desk of the Science Library. Also, your instructor can find a programming partner for you to team up with in class.

Student Learning Outcomes

Upon completion of CIT 382, students will be able to:

1. Define Information Architecture (IA) for Web Development
2. Structure information to support 3-tiered web applications
3. Use AJAX and JSON in applications as a data interchange format between servers and clients
4. Define the difference between NoSQL and SQL databases
5. Store content in a Redis NoSQL datastore
6. Store JSON objects in a MongoDB database
7. Model data with Mongoose (a Node.js module)
8. Deploy a Node.js web app to Cloud Foundry, a cloud-hosting service
**Expected Weekly Workload**

The *Reader’s Guide to the UO Catalog* states, "1 credit represents three hours of a student’s time each week in a lower division undergraduate course".

Therefore, a 4-credit course like 382 will require a total time commitment of 12 hours/week *each week* for the entire 10 weeks of the term.

This means 4 hours in class each week, and up to 8 hours outside of class spent doing projects, reviewing notes, reading and studying the textbook, etc.

**Computing your Course Grade**

1 Midterm .... 25%
1 Final .......... 25%
5 Projects ..... 50%

See *Course Schedule*, below, for exam dates.

The formula to compute your weighted percentage for the course is:

\[ \text{Wtd\%} = (0.5 \times \text{YourProjPts/TtlProjPts} + 0.5 \times \text{YourExamPts}/300) \]

382 uses a straight percentage (not a curve of any kind): 90% is an A, 80% a B, 70% C, 67% C-/P, 60% D, 0-59% F/N. A minimum of 97% is required for the grade of A+.

**Exams**

Both the midterm and the final will be the same length and format: 30 5-point questions, plus 2 extra credit questions.

Both exams are "closed-book" but you may bring one 8.5” x 11” page of notes as a reference during the exam (typed, double-sided, double-density is OK). See *Course Schedule*, below, for exam dates.

**Course Schedule & Significant Dates**

<table>
<thead>
<tr>
<th>Week</th>
<th>Reading</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ch. 1-5, pp. 1-122. Review Unix, Git</td>
<td>Project 1 Due</td>
</tr>
<tr>
<td>2</td>
<td>Ch. 1-5, pp. 1-122. jQuery Mobile, Arrays</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Ch. 6, pp. 123-160. Local Storage</td>
<td>1/18 is a UO Holiday (MLK).</td>
</tr>
<tr>
<td>4</td>
<td>Ch. 7, pp. 161-198. Canvas element</td>
<td>Project 2 Due</td>
</tr>
<tr>
<td>5</td>
<td>Ch. 8, pp. 199-218. Node.js</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Ch. 9, 219-244. NoSQL, MongoDB.</td>
<td>2/21 is the last day to change grade options.</td>
</tr>
<tr>
<td>8</td>
<td>Ch. 10, pp. 245-270. RDBMS, MySQL</td>
<td>Project 4 Due</td>
</tr>
<tr>
<td>9</td>
<td>Ch. 11-12, pp. 271-320. Handlebars.js, user location.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Wrap-up &amp; Review</td>
<td>Project 5 Due</td>
</tr>
<tr>
<td>Finals</td>
<td>382 Final Exam</td>
<td>Final 10:15 Fri 3/18. Optional Makeup Project</td>
</tr>
</tbody>
</table>
Universal Learning Environment
The University of Oregon, the CIS department and your instructional team all support inclusive learning environments. Please notify your instructor if there are aspects of instruction or design of this course that result in barriers to your participation.

Students with a UO disability notification letter should meet with your instructor early in the term. For further information, see the Accessible Education Center web page http://aec.uoregon.edu/.

Piazza vs. Email
Piazza is a next generation Question & Answer platform specifically designed to get you answers fast. You are encouraged to both ask and answer questions in Piazza.

1. Use Piazza for Q&A about course material and projects.
2. Use Email for questions about grading, advising questions, etc. You are welcome to contact me personally by telephone, email, office hours, or appointment.

Programming Teams
Working as part of a two-member team is an option.
Groups of three or more are not an option.

Late Projects & DEADLINES: The Marquess of Queensbury’s

"[Instructors] are a Superstitious Sect,
Great Keepers of Set Times and Places."
— Poor Richard’s Almanac

Programming Projects are due on the assigned due date, and are late thereafter. Late projects are not graded and receive a grade of zero. Therefore, if you anticipate missing a deadline, please contact your instructor or GTF as soon as possible to make arrangements in advance.

Missed Exams
A justifiable and verifiable reason is required to miss an exam. Contact your instructor in advance.

Academic Integrity: Play Hard but Play Fair.
Here is the 382 Academic Integrity Policy in Three Simple Points:

1. Do not share program code with other students.
2. Do not look at solution sets or program code from other terms.
3. Be prepared to explain any program code you submit.

In CIT 382, working together with one programming partner is OK, but plagiarism is not OK.

Cheating and dishonesty undermine the rules of fair play and degrade the class environment for everyone. If you are tempted to cheat, it’s time to elevate your game and think of your impact on others.

Dishonesty can result in severe sanctions that remain on your permanent record.

All UO students should be aware of the UO Student Conduct Code: http://uodos.uoregon.edu/StudentConductandCommunityStandards/AcademicMisconduct.aspx

The Code defines academic misconduct, and describes the formal procedures your instructor will follow in the event of individual student violations of the code.
Take-Away Points for Success

(1) Know the 382 Academic Policies.
(2) Submit projects on time. It is not possible to pass the course without completing the projects on time.
(3) Monitor your grades in the Grade Center. Notify your instructor of problems without delay.
(4) Have your UO email forwarded automatically to your primary account. This will enable you to keep up with the class news and Blackboard announcements, with no separate login required. Read, "How can I forward my e-mails to another account?" at this URL: https://it.uoregon.edu/faq/how-can-i-forward-my-e-mails-another-account
(5) If you are having difficulties in the course, for any reason whatsoever, see your instructor sooner rather than later. You are always welcome to make an appointment.