

COMPSCI 497S Scalable Web Systems

Syllabus / Spring 2021

Course Description

The web has become a large and complex area for application development. Access to an abundance of open source languages, libraries, and frameworks has led to the quick and easy construction of a variety of applications with several moving parts working in coordination to present to the user the illusion of a single program. In reality, web applications are extremely difficult to get right. They involve a large collection of coordinated services, multiple databases, complicated user interfaces, security and performance issues, and ever changing 3rd party services, spread across physical and virtual machines. These complications are further stressed by the large number of concurrent users that access these applications every second. This course will investigate several well known web-based applications and the technology and software architecture used to scale these applications. We will also study a specific topic related to scalability in software design in the context of web application architecture.

Prerequisites

- COMPSCI 326 Web Programming is required to take this course.

Notes

- It is recommended to have (COMPSCI 220 *Programming Methodology* and COMPSCI 230 *Computer Systems Principles*) or 377 *Operating Systems*, however, this is not required but it will make your experience more rich. (e.g., background in JavaScript, basic security, network protocols, processes, client/server architecture, and general operating systems concepts)

Major Applicability

- This course counts as a 400-level Computer Science Major elective.
- This course counts as a 400-level Informatics elective.

Required Materials

- Textbook: [Building Microservices - Designing Fine-Grained Systems](#), Sam Newman. This book is referenced in many places where scalability is the discussed topic. It is well written for an O'Reilly book and relatively easy to read. It will be used as the main thrust of this course as an important building block for scalability. This book is also [available for free](#) from the UMass Safari subscription. Follow this link, click on the "SIGN IN" link at the top right, and type in your UMass email address in the "Email Address" box and click "Sign In" without a password. You should be prompted with the UMass login page where you can add your credentials.

- A computer is required for this course with a reliable internet connection.
- Other material will be distributed for free such as articles on the web, videos, etc.

Objectives

A student who has successfully completed this course with a C or above will be able to:

- Define what a web system is and what scalability means.
- Explain issues that relate to scalability of modern web systems and possible solutions.
- Describe scalability issues in modern web systems.
- Identify potential scalability issues in modern web systems.
- Identify libraries and tools that help solve scalability problems.
- Apply languages, libraries, and tools that are important for scalability.
- Analyze an existing web system and identify problems related to and requirements for scalability.
- Define and explain microservices and their importance in scalability.
- Create a research project that explores technology that impacts or influences scalability.
- Create a software artifact that demonstrates areas of scalability.
- Give a presentation that effectively communicates the results of a research project.

How to Succeed

Resources

Your success in this class is important to us. We all learn differently and bring different strengths and needs to the class. If there are aspects of the course that prevent you from learning or make you feel excluded, please let us know as soon as possible. Together we'll develop strategies to meet both your needs and the requirements of the course. There are also a range of resources on campus, including:

- [Learning Resource Center](#)
- [Center for Counseling and Psychological Health \(CCPH\)](#)
- [English as a Second Language \(ESL\) Program](#)

Self Motivation

The structure of this course is different from standard courses. It relies on the self motivation of each student. The more you put into this course the more you will get out of it. Although several topics will be presented, it is expected that students in this course will contribute a substantial amount of research into various topics and technologies in order to be successful. This is not an introductory course and attempts to scaffold and prepare you to investigate the material to be successful in industry and/or graduate school. You will be evaluated on your contributions and dedication to your team.

Course Structure

The structure of this course will be online and mostly asynchronous. There will be a weekly release of course material along with short videos from the instructor discussing the course content. Weekly material includes readings from the text, online articles, and possibly code examples. We will communicate asynchronously online in several different forms and synchronously on zoom for team meetings..

Access to Course Material

All the material for this course will be available on Moodle in a weekly format. The course material will be released weekly and you are expected to complete the readings and assignments that are assigned that week by the due date. In Moodle you will find text and links to various resources including readings from the book, recorded videos, as well as other online resources.

Assignments and Grading

The final course grade involves assignments that relate to discussion and notes of the material and a semester group project. The breakdown of assignments (some graded, some not) are:

- **(30%) Assignments:** individual activities that range from setting up various software components and/or platforms (e.g., git, github, coding environment, docker, AWS/Azure) to programming assignments in a variety of languages.
- **(40%) Project Assignments:** team-based activities that involve the construction of a scalable web system that your team will design and implement. This will range from a team github repository setup to the construction of various microservices and front-end UI components.
- **(15%) Team Meetings:** you will be required to meet as a team with the instructor each week to communicate the progress of your team's scalable web system. These meetings will be 30 minutes in length and your team will be prepared to present individual contributions as well as team efforts. Teams will be assembled and meeting times will be scheduled during the first couple of weeks of the course starting date. You must attend at least 90% of the scheduled meetings in order to pass the course. Failure to attend will result in a failure for the course.
- **(15%) Project Presentation:** a recorded presentation of your team's project and demonstration of a working prototype.

The weights for each of the above assessment areas may be updated slightly as the course progresses depending on the effectiveness of our remote plan.

All grades will be accessible through the Moodle gradebook and updated at the completion of grading each assessment.

Final Grades

To evaluate your understanding of the course content we will use scores achieved on each of the above assessment components. Your final grade will convey what you demonstrated that you know from the course and how well you demonstrate that knowledge. Missing and late assignments can have a dramatic impact on your final grade so it is important that you are attentive to submission deadlines and avoid any missing work. The typical breakdown of percentages and final grades for this course are A (93-100), A- (90-92), B+ (87-89), B (83-86), B- (80-82), C+ (77-79), C (73-76), C- (70-72), D+ (67-69), D (60-66), F (0-59). This grading scheme may be adjusted based on the overall performance of students in the course.

Course Schedule

The schedule is as follows (subject to change):

- Week 1 - What is a microservice?
- Week 2 - What is an evolutionary architect?
- Week 3 - How do you model services?
- Week 4 - Integration 1
- Week 5 - Integration 2
- Week 6 - Splitting the monolith 1
- Week 7 - Splitting the monolith 2
- Week 8 - Deployment
- Week 9 - Testing
- Week 10 - Monitoring
- Week 11 - Project Work
- Week 12 - Project Work
- Week 13 - Project Work and Presentation

Communication

We will be using the Piazza discussion forum system for communication. The discussion forum should be your first choice for asking questions as others most certainly have the same question. You should check the discussion forum before asking your question to see if the same question has already been posted. We will tend not to answer questions directed towards us that have already been answered in the discussion forum. Think before you post. We expect you to do a reasonable amount of thinking to try to solve your problems before posting for help. Make sure you understand the rules and try to be articulate and clear with your post (again, think before you post). You should post questions related to assignments early rather than waiting until the last minute. If you post a question too close to an assignment deadline, you may not receive an answer before that deadline. Course staff are expected to answer questions Monday through Friday. Do not expect prompt answers on Saturday, Sunday, and scheduled holidays and breaks.

Statement of Inclusivity

The staff for this course support the UMass commitment to diversity, and welcome individuals regardless of age, background, citizenship, disability, sex, education, ethnicity, family status, gender, gender identity, geographical origin, language, military experience, political views, race, religion, sexual orientation, socioeconomic status, and work experience. In this course, each voice in the classroom has something of value to contribute. Please take care to respect the different experiences, beliefs and values expressed by students and staff involved in this course.

Accommodations

Accommodations are collaborative efforts between students, faculty, and Disability Services (DS). Students with accommodations approved through DS are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through DS should contact DS immediately at (413) 545-0892. If you are a student with a documented disability and are registered with Disability Services, please contact me immediately to facilitate arranging academic accommodations. Reasonable arrangements will be made in accordance with your accommodations provided by DS in the context of this course.

Course Incompletes

Students who are unable to complete course requirements within the allotted time because of severe medical or personal problems may request a grade of Incomplete from the instructor of the course. Incomplete grades are warranted only if a student is passing the course at the time of the request and if the course requirements can be completed by the end of the following semester. Furthermore, an incomplete will be granted if at least 75% of the work has been completed for the course. Otherwise, the recommended course of action is to withdraw and retake the course in the future. Please see the Academic Regulations Section IV Grading System and Credit Guidelines for further details.

Note: an incomplete means you are on your own to complete the material agreed upon by the instructor of this course. Do not expect additional help or one-on-one teaching of the material past the course completion date. It is your responsibility to complete the remaining material.

Academic Honesty

It is very important in all courses that you be honest in all the work that you complete. In this course you must complete all assignments, quizzes, exams, etc. on your own unless otherwise specified. If you do not, you are doing a disservice to yourself, the instructors for the course, the School of Computer Science, the University of Massachusetts, and your future. We design our courses to provide you the necessary understanding and skill that will make you an excellent computer scientist. Assignments and exams are designed to test your knowledge and

understanding of the material. Plagiarism and academic honesty of any sort may seem like an easy way to solve an immediate problem (which it is not), however, it can have a substantial negative impact on your career as a computer science student. There are many computing jobs out there and many more people working hard to get those positions. If you do not know your stuff you will have a very difficult time finding a job. Please take this seriously.

We will carefully review your submissions automatically and manually to verify that "cheating" has not taken place. If you are suspected of plagiarism we will follow an informal path to determine if academic dishonesty has taken place. If you are found guilty you will receive an F for the course and it will go on your permanent record at UMass. This will disrupt your schedule for completing courses and may lead to you not completing your degree in a timely fashion. You should carefully review the [Academic Honesty Policy](#), [Avoiding Plagiarism](#), and the [Academic Honesty Flowchart](#) to understand what academic honesty is, how you can avoid it, and the procedure we will follow if you are under suspicion. In general, you should review all documentation described by UMass' [Academic Honesty Policy and Procedures](#).