**SYLLABUS**

# Applied and Cloud Computing for Electrical Engineers

EE 599: Spring 2020 (2 units)

This course introduces tools and concepts to deploy and maintain full stack software solutions in modern computing environments. This project-driven course guides students through the process of taking ideas from concept to product. The course is intended for graduate electrical engineering students with prior programming experience. It exposes students to technologies and practices relevant to modern application deployment. The course consists of three main parts: (1) introduction to cloud technologies and integration methods, (2) frontend and backend programming, and (3) deployment within the global computing infrastructure. The course will introduce students to cloud environments and teach cross-service concepts applicable across platforms and within the medium-term future (4-6 years).


| | | |
|---|---|---|
| Instructor: | Brandon Franzke | |
| Email: | franzke@usc.edu | |
| Office: | EEB 504B (remote) | |
| Zoom: | meet: | 998 5176 5591 |
| | code: | 574987 |
| Hours: | Mo | 14:00 – 15:00 |
| | Tu | 10:00 – 11:30 |
| | We | 14:00 – 15:30 |


## Lecture / Discussion

*Monday and Wednesday (section: 30644)*

12:00 – 13:50

*Tuesday and Thursday (section: 30641)*

15:00 – 16:50

Discussion section is not optional. Homework assignments are discussed primarily during the discussion section. Teaching assistants will cover tools and demonstrate tools during the discussion. They may also cover important additional material.

## Piazza

Piazza gets you help fast and efficiently from classmates, the TAs, and me. I encourage you to post questions on Piazza rather than emailing questions to the teaching staff.

https://piazza.com/usc/spring2021/ee599/home

## Canvas *(replaces Blackboard)*

Use Canvas to electronically submit your homework and view course grades. You will receive an email to register during the first week of classes. Contact the instructor with any issues.

https://canvas.usc-ece.com

### TAs and grader

| | | | | | |
|---|---|---|---|---|---|
| TA: | Xiou Ge | | Grader: | Karkala (Shashank) Hegde | |
| Zoom | meet: | 259 648 9787 | Office hours: | by appointment | |
| | code: | 564788 | E-mail: | khegde@usc.edu | |
| Office hours: | Fr | 10:00 – 12:00 | | | |
| Email: | xiouge@usc.edu | | | | |

| | | |
|---|---|---|
| TA: | Bin Wang | |
| Zoom | meet: | 954 3630 5730 |
| | code: | 311853 |
| Office hours: | Th | 09:30 – 12:00 |
| Email: | wang699@usc.edu | |

### Course materials *(most titles are **not** available through the campus bookstore)*

*Designing Data-Intensive Applications*, Martin Kleppmann, O'Reilly Media, 2017. *(required)*

*Release It!: Design and Deploy Production-Ready Software*, 2nd edition, Michael Nygard, Pragmatic Bookshelf, 2018. *(optional)*

*Mastering Node.js*, 2nd edition, Sandro Pasquali and Kevin Faaborg, Packt Publishing, 2017. *(required)*

*Speaking JavaScript*, 1st edition, Axel Rauschmayer, O'Reilly Media, 2014. *(required)*

*The Road to GraphQL*, Robin Wieruch, 2018. *(online: https://github.com/the-road-to-graphql/the-road-to-graphql)*

### Technical Proficiency and Hardware/Software required

You need access to a full stack for both Node.js and Python development. Some tools may require a Linux installation. You may also consider installing a Linux virtual machine to ensure maximum interoperability and access to any tools. The instructor and teaching assistants will give guidance during the first weeks of class.

## Learning Objectives

A student that successfully completes this course will:

- Know how to manage the lifecycle of a software application from concept, deployment, maintenance, and end of life.
- Understand the role of cloud services in a modern application stack.
- Develop client-side programming skills to deploy purpose-built applications that require complex or asynchronous user input.
- Develop interactive applications that expose backend state and data storage to distributed clients.
- Distinguish standard databases and apply option(s) that best represent a given data model, cost requirement, or compatibility.

- Ability to work within common cloud platforms, understand their limitations, and how choices affect the scope or reach of their software solution.
- Be comfortable working within virtual or containerized environments and have knowledge to access host level devices.
- Understand how applications exist within and interact with the global computing infrastructure.

## Course Outline (tentative)

| | *week of* | | |
|---|---|---|---|
| 1 | 18 / 19 | Jan | ***No class:*** *Martin Luther King Day, University holiday.* |
| | 20 / 21 | Jan | Architecture (distributed vs local). Tasks: synchronous vs. asynchronous. |
| 2 | 25 / 26 | Jan | Creating a backend: language overview. |
| 3 | 01 / 02 | Feb | Databases I: relational. |
| 4 | 08 / 09 | Feb | Databases II: NoSQL. |
| 5 | 17 / 18 | Feb | ***No class:*** *Presidents' Day, University holiday.* |
| | 19 / 20 | Feb | Databases III: graph and in-memory. |
| 6 | 22 / 23 | Feb | Virtualization (OS vs hardware, containers). Cloud services (AWS, Azure, GCS). |
| 7 | 01 / 02 | Mar | **Exam #1.** |
| 8 | 08 / 09 | Mar | Frontend application development: HTML, CSS, JavaScript. |
| 9 | 15 / 16 | Mar | Backend application development: Node.JS. |
| 10 | 22 / 23 | Mar | ***No class:*** *Wellness Day.* |
| | 24 / 25 | Mar | Backend and API development. JSON. REST vs GraphQL. |
| 11 | 29 / 30 | Mar | Authentication. |
| 12 | 05 / 06 | Apr | Lifecycle: testing, continuous deployment, maintenance. |
| | 07 / 08 | Apr | ***No class:*** *Wellness Day.* |
| 13 | 12 / 13 | Apr | Networking and infrastructure: DNS, routing, HTTP, SSL + certificates. |
| 14 | 19 / 20 | Apr | System design and scalability: instrumentation and anomaly detection. |
| | 21 / 22 | Apr | ***No class:*** *Wellness Day.* |
| 15 | 26 / 27 | Apr | Prospects: Digital ledger databases and blockchain. |
| final | | | Project Presentations |

## Grading Procedure

### Homework

Homework will be assigned every 1-2 weeks. Problems will be a mix of applied, analytical, and computational problems. Your total homework score sums your best homework scores (as a percentage) after removing the lowest one score. Homeworks are due by the posted due date. Late homework will be accepted with a 10% deduction per 24-hours for up to 48-hours.

You may discuss homework problems with classmates but each student must do his or her own work. Cheating warrants an F in the course. Turning in identical homework establishes a rebuttable presumption of cheating.

### pExam

Ensure that you can stream live video (mute or audio off) during the entirety of the work time for proctoring. You must make prior arrangements with me if that is not possible. You may use a single 8.5"x11" reference sheet (front and back OK). You may not use any additional resources. You are expected to bring a non-graphing scientific calculator. You must show how you arrived at your answers to receive full credit. Any cheating may result in an "F" in the course and will be referred to Student Affairs for other penalties. Make up exams will only be given for valid medical or family emergency excuses (proof required).

### Attendance and Participation

Attendance is mandatory to all lectures and discussions. You are responsible for missed announcements and changes to the course schedule and assignments. Your attendance may be synchronous or asynchronous. Make synchronous attendance a priority. Per university guidance: you should plan to attend every synchronous session of this class regardless of when it occurs in your time zone. Some unreasonable hours (sessions outside 07:00 – 22:00) may preclude this general rule.

Synchronous class dynamics are improved substantially with visible participants in the class. Arrange to have your cameras on (default: "camera on, audio off") during synchronous online sessions. You must make prior arrangements with me if that is not possible.

USC policy requires that all classes conducted online be recorded for asynchronous viewing with transcriptions made available. These recordings are considered "educational records" subject to federal privacy laws (FERPA) as students may be personally identifiable in class recordings by voice, name, or image. Students are not permitted to create their own class recordings without prior written permission. Violations of these policies will be met with the appropriate disciplinary sanction.

### Cheating

Cheating is not tolerated on homework or exams. Penalty ranges from F on exam to F in course to recommended expulsion.

### Course Grade

| | | | |
|---|---|---|---|
| Homework | 35% (lowest 1 thrown out) | A | if 90 – 100 points |
| Exam | 25% | B | if 80 – 89 points |
| Final Project | 40% | C | if 70 – 79 points |
| | | D | if 60 – 69 points |
| | | F | if 0 – 59 points |

("+" and "-" within approx. 2% of grade boundary)

## Final project

This course has a team-based final project In lieu of a final exam.

Teams of three students (teams of two with instructor approval) design and implement a complete software product that connects *two or more* independent asynchronous components (often "frontend" and "backend"). The instructor will guide teams with difficulty identifying a suitable application. Teams may build an application similar to existing services or tools but their must efforts demonstrate understanding of the entire development stack and the product lifecycle from idea to deployment to maintenance. Though teams are encouraged to devise problems of particular interest to their backgrounds, interest, or research. All projects must obtain the instructor's

written approval. Teams will prepare and present their *approved* project and show how it applies course material, concepts, and best-practices. Attendance *and participation* during the project presentation session(s) are mandatory.

## Requirements

Project topics must include sufficient scope and apply course knowledge to a useful end. The project must compose *at least two* distinct units that operate and act independently but provide greater function when acting together. The project must demonstrate comprehensive understanding of the entire development stack and the product lifecycle from idea to deployment to maintenance. You may use any computer language you like but deviations from Python, C++, Node.js, GoLang, or other frameworks used in class require *prior* instructor approval.

## Grading and Milestones

| | | |
|---|---|---|
| Topic proposal | week 9 | 10% |
| Phase 1 – Design, components, classes, and tests | week 13 | 15% |
| Phase 2 – Integration and deployment | week 15 | 20% |
| Demo and presentation | final | 20% |
| Project report and video | | 35% |

## Deliverables and Demo

- ***Written project report:*** the project report should summarize the topic, provide relevant background (theoretical or applied), timeline and contributions, and document challenges and extensions. It should provide discussion sufficient that an uninformed expert could understand the logic, algorithmic decisions, and implementations. Teams should provide quantifiable metrics to justify engineering tradeoffs.
- ***Presentation:*** Approximately 10-minute (depends on class size) presentation to describe to the class their topic problem and their solution. It should provide only what is necessary to understand the "what" and "why" and include minimal theoretical background.
- ***Video:*** 3-4-minute video that describes the problem, your design, and implementation. All team members must participate equally. You may choose to upload this to a video sharing site such as YouTube but that is not required. We may also post videos in a playlist on the CloudComputing USC YouTube channel.
- ***Source code:*** submitted to instructor by providing link to pull from GitHub.

## Example projects

- ***Neural network platform UI***: Design and implement a frontend and backend to interface an existing 3rd party Machine Learning API such as Amazon SageMaker or Microsoft Azure ML. It should be a responsive design and deliver a premium user experience. The application may allow or enforce different policies based on authentication, provide visualization of models or results in addition to the platform tools. It may also integrate information from multiple services such billing and quota.
- ***BigData insight tool***: Design a suite of tools (frontend) that integrate with a potentially large backend dataset. The frontend should expose a UI that lets a user validate multiple hypotheses. The backend should cache and manage the datastore in a way to provide an optimal user experience. The tools might include a rudimentary statistics engine to perform regressions or estimation and may also report abnormal cases or data that violates common statistical assumptions such as zero-correlation, homoscedasticity, and non-normality.

# Academic Conduct

## Plagiarism

Presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in SCampus in Section 11, Behavior Violating University Standards https://scampus.usc.edu/1100-behavior-violating-university-standards-andappropriate-sanctions. Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct.

Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the Office of Equity and Diversity http://equity.usc.edu or to the Department of Public Safety http://capsnet.usc.edu/department/department-public-safety/online-forms/contactus. This is important for the safety of the whole USC community. Another member of the university community – such as a friend, classmate, advisor, or faculty member – can help initiate the report, or can initiate the report on behalf of another person. The Center for Women and Men http://www.usc.edu/studentaffairs/cwm/ provides 24/7 confidential support, and the sexual assault resource center webpage http://sarc.usc.edu describes reporting options and other resources.

## Academic Integrity

Academic integrity is critical the assessment and evaluation we perform which leads to your grade. In general, all work should be your own and any sources used should be cited. Gray-areas occur when working in groups. Telling someone how to do the problem or showing your solution is a VIOLATION. Reviewing examples from class or other sources to help a fellow classmate understand a principle is fine and encouraged. All students are expected to understand and abide by these principles. SCampus, the Student Guidebook, contains the University Student Conduct Code in Section 10, while the recommended sanctions are located in Appendix A. Students will be referred to the Office of Student Judicial Affairs and Community Standards for further review, should there be any suspicion of academic dishonesty.

# Support Systems

A number of USC's schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the American Language Institute http://dornsife.usc.edu/ali, which sponsors courses and workshops specifically for international graduate students. The Office of Disability Services and Programs http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, USC Emergency Information http://emergency.usc.edu will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.

## Academic Accommodations

Any student requiring academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to me as early in the semester as possible. DSP is located in GFS 120 and is open 08:30 – 17:00, Monday through Friday. The phone number for DSP is (213) 740-0776.