USC Viterbi
School of Engineering

Course ID and Title: EE 599 Distributed Systems for Medical Applications

Units: 4
Term—Day—Time: Spring 25—TBD
Location: TBD

Instructor:

Office: EEB 342
Office Hours: TR, 9:30-10:30
Contact Info: coreybak@usc.edu

Teaching Assistant:

Office:
Office Hours:
Contact Info:

Catalogue Description

The course content will be selected each semester to reflect current trends and developments in the field of electrical engineering.

Course Description

This course addresses topics related to designing medical based distributed systems for patients, physicians, and engineers. There are many types of data produced in such systems, patient objective sensor data, subjective information, etc. The course will cover the technical aspects and best practices for building medical based distributed systems; specifically, synchronization, data-consistency, front-end mobile applications and operating systems, virtualized backend containers (Docker), databases, fast healthcare interoperability resources (FHIR), interaction with privacy-sensitive electronic health records, and HIPAA compliance. Other possible topics: cyber-security, cloud computing, AI/ML for medical-based distributed systems.

Design and implementations will be conducted in Swift, Python, and NodeJS programming languages, with additional discussions that span UI development, object-oriented programming (OOP), protocol-oriented programming (POP), functional programming, model view view-model (MVVM), memory allocation, synchronous vs asynchronous tasks, databases, networking, version control, agile development and other aspects of software engineering. The course will also discuss how to evaluate and optimize system performance and discuss design decisions based on system resources.

Learning Objectives

Students will learn how to:

- Build real-world distributed systems using front-end and back-end tools
- Design end-to-end systems that are intrinsically private and HIPAA compliant
- Distinguish between serial programs and loosely coupled distributed systems
- Apply classic distributed algorithms for synchronization and consistency
- Aggregate sensor-based data from smart devices
- Evaluate system and algorithm performance in time and space complexity
- Assessment of algorithm decisions and cost criteria
- Understand protocol-oriented programing (POP)
- Leverage functional programming principles

- Implement model view view-model (MVVM) design pattern
- Design robust unit-tests to minimize errors and regressions

Prerequisite(s): N/A
Co-Requisite(s): N/A
Concurrent Enrollment: N/A
Recommended Preparation: Completed a course such as

Course Notes
N/A

Technological Proficiency and Hardware/Software Required
Students are required to be comfortable in programming languages such as C++, Java, Python, or similar. It is recommended to have access to an Apple computer that can run the latest macOS. Linux and Windows based systems are allowed, but assistance with using these systems will be limited.

Required Readings and Supplementary Materials
- **M. van Steen and A.S. Tanenbaum, Distributed Systems, 4th ed., distributed-systems.net, 2023** - ***DS** - Available for Free
- The Swift Programming Language. By Apple Inc. - ***SPL** - Available for free: iBooks - here, online - here
- Apple Developer Documentation, By Apple Inc. - ***ADD** – Available for free here

Optional Readings and Supplementary Materials
- Digital Health Technologies for Remote Data Acquisition in Clinical Investigations
- A Systematic Literature Review of the use of Formal Methods in Medical Software Systems
- Technical Guidance for Clinicians Interested in Partnering With Engineers in Mobile Health Development and Evaluation
- Leveraging mobile health applications for biomedical research and citizen science: a scoping review
- Solutions for Mitigating Cybersecurity Risks Caused by Legacy Software in Medical Devices: A Scoping Review
- Potential of Internet of Medical Things (IoMT) applications in building a smart healthcare system: A systematic review
- Internet of Medical Things (IoMT): Overview, Emerging Technologies, and Case Studies
- Medical 4.0 technologies for healthcare: Features, capabilities, and applications
- A user-centered design framework for mHealth
- Designing personalized mHealth solutions

Description and Assessment of Assignments
Assignments, Submissions, and Additional Material

- Brightspace
- GitHub Organization
- GitHub Classroom

Homeworks

All homework assignments are designed to assist you with understanding wireless communications. Required readings will be provided to students via Brightspace. Homework will consist of the following type of assignments:

- Literature reviews - requires students to read published research papers or blogs online about specific topics in networking and mobile app development. These assignments will require each student to write a unique 1-2 page review of the research paper or blog

    - Review should consist of:
        - 1 paragraph summary of the paper
        - 3 strengths in the paper
        - 3 weaknesses of the paper
        - Based on your understanding of the paper, state at least 1 potential problem you would like to explore
    - Identify 3-5 reference papers included in this paper that look to be interesting reads for the future
- Coding assignments - requires students to write code and may follow a written tutorial from the web. The instructor may require students to write a 0.5 - 1 page write-up describing the work completed in the coding assignment

Projects

The objective of the course is for each person/team (depending on class size) to create a functioning medical/health distributed system with supporting documentation (write-up) of contributions. Ongoing work for the project will last the duration of the semester with updates due in the form of presentations, write-up, and demos at the 9 and 16 week dates of the semester.

Committing to Github Open-source Projects

Open-source projects are critical enabling a number of technologies and applications. Throughout the semester, students will use a number of open-source libraries for their homework and projects. While using open-source projects, it is common to find bugs or require additional functionality from a framework. In these situations, students can earn extra credit by submitting "pull-requests" to public Swift frameworks. Additional credit can be given if the "pull-request" is accepted by the owner. Some notable open-source projects that students can contribute to during the semester are here: https://github.com/stars/cbaker6/lists/health-care

Participation

Please post all questions in the Brightspace/Slack discussion board first as other students may have a similar question. Posting in Brightspace/Slack allows all questions/answers to be centralized and easily viewed by all classmates. Other students enrolled in the course are encouraged to answer questions and provide suggestions.

With regard to questions directly related to code, provide inline comments and post the repo on the course GitHub. The question should still be posted in the Brightspace/Slack with links to your GitHub repository. These type of questions should point out the problem, the line number where the problem occurs, and different steps taken to attempt to remedy the problem.

If your question is not answered in Brightspace/Slack, please contact the instructor via email or office hours. Note that private questions to the TA's or the professor can be posted in Brightspace/Slack if needed.

Note: Most questions should be asked in Brightspace/Slack discussions (can ask direct questions to the professor or TAs within Brightspace/Slack). If for some reason you need to email the professor or TAs **always include "EE 599: " in the subject line**.

Grading Breakdown

Table 1 Grading Breakdown

Assessment Tool (assignments)	Points	% of Grade
Homework		30%
Leading in-class paper discussions		10%

Assessment Tool (assignments)	Points	% of Grade
Midterm Project Presentation		10%
Midterm Project Write-Up (2-3 pg IEEE/ACM conference format)		10%
Final Project Presentation		10%
Final Project Write-Up (6-7 pg IEEE/ACM conference format)		20%
Final demo		10%
TOTAL		100%

Grading Scale

Course final grades will be determined using the following scale:

Table 2 Course Grading Scale

Letter grade	Corresponding numerical point range
A	95-100
A-	90-94
B+	87-89
B	83-86
B-	80-82
C+	77-79
C	73-76
C-	70-72
D+	67-69
D	63-66
D-	60-62
F	59 and below

Assignment Submission Policy

All assignments and project submissions will be submitted by 1:30 pm (or a designated time by the Professor) of the deadline through BrightSpace or Github Classroom. **Late homework will not be accepted under any conditions.**

Grading Timeline

Assignment grades are expected to be given to students within 1.5 weeks of the assignment due date unless otherwise specified by the professor.

Course Specific Policies
N/A

Attendance
Students are responsible for knowing the material provided in the lectures as well as being able to apply the concepts to develop the project.

Classroom norms
Active class participation will count towards a student's participation grade. Discussion sessions will be used to assist students with understanding key concepts. Attending discussions is mandatory. Students will work independently on literature reviews and work in groups on programming assignments. Students are expected to assist one another in the learning process by sharing experiences with overcoming design and implementation hurdles as well as best practices. Though students will work in 1-2 person teams, cross interaction of teams is mandatory. For example, each team will become an expert in a particular system and will be responsible for leading a discussion board on Brightspace/Slack on the subject. By the end of the semester, students will not only understand the ongoing problems in their field but will have some idea on how they will provide contributions to addressing the identified problems.

The overarching goal of this course is for students to accomplish the ambitious tasks of learning about state-of-the-art medical based systems as well as developing them within one semester. A final project (discussed below) will require students to leverage concepts from the open-source community. Classroom lectures are intended to teach basic concepts and encourage innovative ways to accomplish project goals. Cross collaborations are encouraged.

Zoom étiquette
N/A

Academic Integrity
The University of Southern California is foremost a learning community committed to fostering successful scholars and researchers dedicated to the pursuit of knowledge and the transmission of ideas. Academic misconduct contrasts with the university's mission to educate students through a broad array of first-rank academic, professional, and extracurricular programs and includes any act of dishonesty in the submission of academic work (either in draft or final form).

This course will follow the expectations for academic integrity as stated in the USC Student Handbook. All students are expected to submit assignments that are original work and prepared specifically for the course/section in this academic term. You may not submit work written by others or "recycle" work prepared for other courses without obtaining written permission from the instructor(s). Students suspected of engaging in academic misconduct will be reported to the Office of Academic Integrity.

Other violations of academic misconduct include, but are not limited to, cheating, plagiarism, fabrication (e.g., falsifying data), knowingly assisting others in acts of academic dishonesty, and any act that gains or is intended to gain an unfair academic advantage.

Academic dishonesty has a far-reaching impact and is considered a serious offense against the university. Violations will result in a grade penalty, such as a failing grade on the assignment or in the course, and disciplinary action from the university itself, such as suspension or even expulsion.

For more information about academic integrity see the student handbook or the Office of Academic Integrity's website, and university policies on Research and Scholarship Misconduct.

Please ask your instructor if you are unsure what constitutes unauthorized assistance on an exam or assignment or what information requires citation and/or attribution.

Using Generative AI and Large Language Models
This course aims to develop creative, analytical, and critical thinking skills. Therefore, all assignments should be prepared by the student working individually or in groups (if allowed on respective assignment). Students may not have another person or entity complete any substantive portion of the assignment. Developing strong competencies in these areas will prepare you for a competitive workplace. Therefore, using AI-generated text, code, or other content is prohibited in this course, will be identified as plagiarism, and will be reported to the Office of Academic Integrity.

Using Stack Overflow, Opensource, or Code Found Online
All assignments should be prepared by the student working individually or in groups (if allowed on respective assignment). Students may not have another person or entity complete any substantive portion of the assignment. Small snippets of code may be used to assist with solving small parts of the assignments, but this code **MUST** be documented with the original link and author in the comments directly above the respective code. The respective code will be evaluated by the TAs and the Professor to determine if the documented code is feasible to not impact the student's grade.

Course Content Distribution and Synchronous Session Recordings Policies

USC has policies that prohibit recording and distribution of any synchronous and asynchronous course content outside of the learning environment.

Recording a university class without the express permission of the instructor and announcement to the class, or unless conducted pursuant to an Office of Student Accessibility Services (OSAS) accommodation. Recording can inhibit free discussion in the future, and thus infringe on the academic freedom of other students as well as the instructor. (Living our Unifying Values: The USC Student Handbook, page 13).

Distribution or use of notes, recordings, exams, or other intellectual property, based on university classes or lectures without the express permission of the instructor for purposes other than individual or group study. This includes but is not limited to providing materials for distribution by services publishing course materials. This restriction on unauthorized use also applies to all information which has been distributed to students or in any way had been displayed for use in relation to the class, whether obtained in class, via email, on the internet, or via any other media. Distributing course material without the instructor's permission will be presumed to be an intentional act to facilitate or enable academic dishonesty and is strictly prohibited. (Living our Unifying Values: The USC Student Handbook, page 13).

Course Evaluations

Course evaluation occurs at the end of the semester university wide. It is an important review of students' experience in the class. The process and intent of the end-of-semester evaluation should be provided. In addition, a mid-semester evaluation is recommended practice for early course correction. Contact CET for support in creating a mid-semester evaluation.

Course Schedule

Table 3 Course schedule (Tentative)

Week	Topic	Readings	Assignments
1	Intro to Programming via Swift	**SPL**: Strings & Characters, Collection Types	Homework 1: Intro to Github

2	Advanced Programming (Protocol Oriented Programming and OOP)	**SPL**: Functions, Structures & Classes, Closures	Homework 2
3	Swift UI and MVVM	**SPL**: Extensions, Protocols, Generics, Access Control	Homework 3
4	Designing and deploying your first distributed system	**SPL**: Initialization, Deinitialization, Optional Chaining	Homework 4
5	Intro to Distributed Computing	**DS:** Selected sections	Homework 5
6	Virtualization: Containers & Virtual Machines	**DS:** Chap 3.0 - 3.4	Homework 6
7	Coordination & Synchronization: Logical Clocks	**DS:** Chap 5.0-5.3	Homework 7
8	Coordination & Synchronization: Gossip-based Coordination & Event Matching	**DS:** Chap 5.5-5.6	Homework 8
9	Midterm Presentations		Submit midterm write-up and presentation
10	Distributed Data Storage: Consistency & Replication	**DS:** Chap 7.0 - 7.3	Homework 9
11	Distributed Data Storage: Consistency & Replication	**DS:** Selected sections	Homework 10
12	Selected Paper Review	See Optional Readings section	Homework 11
13	Selected Paper Review	See Optional Readings section	Homework 12
14	Selected Paper Review	See Optional Readings section	Homework 13

15	Selected Paper Review	See Optional Readings section	Homework 14
Final	Final Presentation and Demo		Date: For the date and time of the final for this class, consult the USC Schedule of Classes at www.usc.edu/soc.

Statement on University Academic and Support Systems

Students and Disability Accommodations:
USC welcomes students with disabilities into all of the University's educational programs. The Office of Student Accessibility Services (OSAS) is responsible for the determination of appropriate accommodations for students who encounter disability-related barriers. Once a student has completed the OSAS process (registration, initial appointment, and submitted documentation) and accommodations are determined to be reasonable and appropriate, a Letter of Accommodation (LOA) will be available to generate for each course. The LOA must be given to each course instructor by the student and followed up with a discussion. This should be done as early in the semester as possible as accommodations are not retroactive. More information can be found at osas.usc.edu. You may contact OSAS at (213) 740-0776 or via email at osasfrontdesk@usc.edu.

Student Financial Aid and Satisfactory Academic Progress:
To be eligible for certain kinds of financial aid, students are required to maintain Satisfactory Academic Progress (SAP) toward their degree objectives. Visit the Financial Aid Office webpage for undergraduate- and graduate-level SAP eligibility requirements and the appeals process.

Support Systems:
Counseling and Mental Health - (213) 740-9355 – 24/7 on call
Free and confidential mental health treatment for students, including short-term psychotherapy, group counseling, stress fitness workshops, and crisis intervention.

988 Suicide and Crisis Lifeline - 988 for both calls and text messages – 24/7 on call
The 988 Suicide and Crisis Lifeline (formerly known as the National Suicide Prevention Lifeline) provides free and confidential emotional support to people in suicidal crisis or emotional distress 24 hours a day, 7 days a week, across the United States. The Lifeline consists of a national network of over 200 local crisis centers, combining custom local care and resources with national standards and best practices. The new, shorter phone number makes it easier for people to remember and access mental health crisis services (though the previous 1 (800) 273-8255 number will continue to function indefinitely) and represents a continued commitment to those in crisis.

Relationship and Sexual Violence Prevention Services (RSVP) - (213) 740-9355(WELL) – 24/7 on call
Free and confidential therapy services, workshops, and training for situations related to gender- and power-based harm (including sexual assault, intimate partner violence, and stalking).

Office for Equity, Equal Opportunity, and Title IX (EEO-TIX) - (213) 740-5086
Information about how to get help or help someone affected by harassment or discrimination, rights of protected classes, reporting options, and additional resources for students, faculty, staff, visitors, and applicants.

Reporting Incidents of Bias or Harassment - (213) 740-2500

Avenue to report incidents of bias, hate crimes, and microaggressions to the Office for Equity, Equal Opportunity, and Title for appropriate investigation, supportive measures, and response.

The Office of Student Accessibility Services (OSAS) - (213) 740-0776
OSAS ensures equal access for students with disabilities through providing academic accommodations and auxiliary aids in accordance with federal laws and university policy.

USC Campus Support and Intervention - (213) 740-0411
Assists students and families in resolving complex personal, financial, and academic issues adversely affecting their success as a student.

Diversity, Equity and Inclusion - (213) 740-2101
Information on events, programs and training, the Provost's Diversity and Inclusion Council, Diversity Liaisons for each academic school, chronology, participation, and various resources for students.

USC Emergency - UPC: (213) 740-4321, HSC: (323) 442-1000 – 24/7 on call
Emergency assistance and avenue to report a crime. Latest updates regarding safety, including ways in which instruction will be continued if an officially declared emergency makes travel to campus infeasible.

USC Department of Public Safety - UPC: (213) 740-6000, HSC: (323) 442-1200 – 24/7 on call
Non-emergency assistance or information.

Office of the Ombuds - (213) 821-9556 (UPC) / (323-442-0382 (HSC)
A safe and confidential place to share your USC-related issues with a University Ombuds who will work with you to explore options or paths to manage your concern.

Occupational Therapy Faculty Practice - (323) 442-2850 or otfp@med.usc.edu
Confidential Lifestyle Redesign services for USC students to support health promoting habits and routines that enhance quality of life and academic performance.