Summer 2023 Syllabus

Course Information	**CS 321: Software Engineering**
Instructor	**Prof. Wes Masri** Email: wmasri@gmu.edu Office Hours: MTWR after class Location: 10:30 am - 12:35 pm - MTWR - Nguyen Engineering Building 1107 GTA: Sajed Jalil – sjalil@gmu.edu – Office Hours: TBD UTA: TBD – Office Hours: TBD Classroom rules: 1) No electronic devices during lectures 2) Project teams are encouraged to meet during the last 30 minutes of each class meeting 3) Teams will vary in size between 5 and 7 members 4) Teams will be picked randomly then possibly adjusted based on skill sets 5) Teams will likely have a mix of inexperienced and highly experienced members. It is the duty of the experienced team members to help and even train the other teammates Notes: 1) The tests are worth 30%. You will need to prepare well for them in order to end up with an A. We will have 4 tests of which one will be dropped. 2) You can opt out of one of the following deliverables: a) P2-API & Testing; and b) P3-Software Architecture. However, doing both might slightly help when computing your final grade. 3) There will be no Discussion Board deliverables To help you with your writing deliverables, we will be using a writing assistant tool called Packback (www.packback.co). Please note that you will need to purchase a license for it (at the cost of $29).
Course Description	An introduction to concepts, methods, and tools for the creation of large-scale software systems. Methods, tools, notations, and validation techniques to analyze, specify, prototype, and maintain software requirements. Introduction to object-oriented requirements modeling, including use of case modeling, static modeling, and dynamic modeling using the Unified Modeling Language (UML) notation. Concepts and methods for the design of large-scale software systems. Fundamental design concepts and design notations are introduced. A study of object-oriented analysis and design modeling using the UML notation. Students participate in a group project on software requirements, specification, and object-oriented software design.

	Specialized Designation: Writing Intensive in Major. The links below list the related outcomes: https://wac.gmu.edu/writing-intensive-courses/wi-learning-outcomes/ https://masoncore.gmu.edu/writing-intensive-2/
Course Objectives	Upon completion of the course, students should have: 1. An understanding of all phases of the software engineering lifecycle (requirements, design, implementation, testing, deployment, maintenance). 2. An understanding of several lifecycle models including both prescriptive and agile methodologies and knowledge of tradeoffs among the methodologies. 3. An ability to document software requirements and design artifacts. 4. An ability to analytically evaluate software usability. 5. An understanding of fundamental techniques used to lead a software team. 6. An ability to apply software engineering techniques to create a minimum viable product. CS 321 includes Writing Intensive (WI) activities that, together with those of CS 306 (Ethics course), meet the GMU WI Requirements in the BS CS Program (http://wac.gmu.edu). This means you will write 1750+ graded words (or about 7 standard pages). For this course, part of the writing will include an individual essay on your experiences working with your team.
Course Methodology	The class format will combine reading, lectures, presentations, and other learning tools. The class will be interactive and require every student to be engaged in the classroom discussion and assignments.
Required textbook(s) and/or materials	Lecture notes are essential. A Packback Deep Dives license – cost $29 (https://www.packback.co/product/deep-dives/) - details will be discussed during the first week of the semester Optional Reading Resources (do not purchase): • Engineering Software Products: An Introduction to Modern Software Engineering, Ian Sommerville, 1st edition, Pearson • Software Engineering, Ian Sommerville, 9th/10th Edition, Addison-Wesley / Pearson Education Limited. • https://innolution.com/essential-scrum/table-of-contents
Computer Requirements	**Hardware:** You will need access to a Windows or Macintosh computer with at least 2 GB of RAM and access to a fast and reliable broadband internet connection (e.g., cable, DSL). A larger screen is recommended for better visibility of course material. You will need speakers or headphones to hear recorded content and a headset with a microphone is recommended for the best experience. For the amount of Hard Disk Space required taking a distance education course, consider and allow for: 1. The storage amount needed to install any additional software and 2. Space to store work that you will do for the course. If you consider the purchase of a new computer, please go to Patriot Tech to see recommendations. **Software:** You will need a browser and operating system that are listed compatible or certified with the Blackboard version available on the myMason Portal. See supported browsers and operating systems. Log in to myMason to access your registered courses. Some courses may use other learning management systems. Check the syllabus or contact the instructor for

	details. Online courses typically use Acrobat Reader, Flash, Java, and Windows Media Player, QuickTime and/or Real Media Player. Your computer should be capable of running current versions of those applications. Also, make sure your computer is protected from viruses by downloading the latest version of Symantec Endpoint Protection/Anti-Virus software for free here. Students owning Macs or Linux should be aware that some courses may use software that only runs on Windows. You can set up a Mac computer with Boot Camp or virtualization software so Windows will also run on it. Watch this video about using Windows on a Mac. Computers running Linux can also be configured with virtualization software or configured to dual boot with Windows. Note: If you are using an employer-provided computer or corporate office for class attendance, please verify with your systems administrators that you will be able to install the necessary applications and that system or corporate firewalls do not block access to any sites or media types. **CS Department Recommended Computer Systems for Students** can be found here.
Course Website	Blackboard will be used for this course. You can access the site at http://mymasonportal.gmu.edu. Login and click on the “Courses” tab. You will see CS 321. NOTE: Username and passwords are the same as your Mason email account. You must have consistent access to an internet connection in order to complete the assignments in this course through Blackboard (http://mymason.gmu.edu). MS Teams (Link will be shared on Blackboard)
Participation	Learning can only happen when you are playing an active role. It is important to place more emphasis on developing your insights and skills, rather than transmitting information. Knowledge is more important than facts and definitions. It is a way of looking at the world, an ability to interpret and organize future information. An active learning approach will more likely result in long-term retention and better understanding because you make the content of what you are learning concrete and real in your mind. Although an active role can look differently for various individuals, it is expected in this class that you will work to explore issues and ideas under the guidance of the professor and your peers. You can do this by reflecting on the content and activities of this course, asking questions, striving for answers, interpreting observations, and discussing issues with your peers.
Rules and Expectations	In correspondence/communication students will be expected to: a) Be professional and respectful in correspondence b) Make reasonable requests of the instructor. We will be happy to clarify course material and answer legitimate questions; however, please exhaust other information sources (e.g., syllabus, Blackboard, Discussion board) for answering your question before contacting me and remember, “Poor planning on your part does not constitute an emergency on my part” In regard to honesty in work students will be expected to: a) Review the University integrity and honesty policies in the student handbook for guidelines regarding plagiarism and cheating (summarized below). I will gladly clarify my stance on any questionable or “grey area” issues you may have. b) Refrain from dishonest work as it will receive a minimum penalty of zero on the assignment and a maximum penalty of a zero for the course with a report to the Honor committee. The GMU Honor Code requires that faculty submit any suspected

	Honor Code violations to the Honor Committee. Therefore, any suspected offense will be submitted for adjudication. Contesting of grades on any/all submissions must be requested within one week of the item's return. No grade changes will be considered subsequent to that deadline, or after the final exam meeting.
Mason Honor Code	**The complete Honor Code is as follows:** *To promote a stronger sense of mutual responsibility, respect, trust, and fairness among all members of the George Mason University community and with the desire for greater academic and personal achievement, we, the student members of the university community, have set forth this honor code:* **Student members of the George Mason University community pledge not to cheat, plagiarize, steal, or lie in matters related to academic work.** *(From the Catalog – catalog.gmu.edu)* All students are expected to abide by the GMU Honor Code and the CS Department Honor Code. This policy is rigorously enforced. All class-related assignments are considered individual efforts unless explicitly expressed otherwise (in writing). Review the university honor code and present any questions regarding the policies to instructor. Cheating on any assignment will be prosecuted and result in a notification of the Honor Committee as outlined in the GMU Honor Code.
Cheating Policy	Any form of cheating on an activity, project, or exam will result in zero points earned. “Cheating” includes, but is not limited to, the following: reviewing others’ exam papers, having ANY resources utilized when not allowed, collaborating with another student during an individual assignment. If you have questions about when the contributions of others to your work must be acknowledged and appropriate ways to cite those contributions, please talk with the professor or utilize the GMU writing center.
Plagiarism and the Internet	Copyright rules also apply to users of the Internet who cite from Internet sources. Information and graphics accessed electronically must also be cited, giving credit to the sources. This material includes but is not limited to e-mail (don't cite or forward someone else's e-mail without permission), newsgroup material, information from Web sites, including graphics. Even if you give credit, you must get permission from the original source to put any graphic that you did not create on your web page. Shareware graphics are not free. Freeware clipart is available for you to freely use. If the material does not say "free", assume it is not. Putting someone else's Internet material on your web page is stealing intellectual property. Making links to a site is, at this time, okay, but getting permission is strongly advised, since many Web sites have their own requirements for linking to their material. Review the Honor Code here.
Individuals with Disabilities	Students with documented disabilities should contact the Office of Disability Services (703) 993-2474) to learn more about accommodations that may be available to them. *(From the 2019-2020 Catalog – catalog.gmu.edu)* Students with documented disabilities should speak with the instructors ASAP to discuss accommodations.
Academic Integrity and Inclusivity	This course embodies the perspective that we all have differing perspectives and ideas and we each deserve the opportunity to share our thoughts. Therefore, we will conduct our discussions

<table>
<tr><td></td><td>with respect for those differences. That means, we each have the freedom to express our ideas, but we should also do so keeping in mind that our colleagues deserve to hear differing thoughts in a respectful manner, i.e. we may disagree without being disagreeable. http://oai.gmu.edu/</td></tr>
<tr><td>Student Privacy Policy</td><td>George Mason University strives to fully comply with FERPA by protecting the privacy of student records and judiciously evaluating requests for release of information from those records.

Please see George Mason University’s student privacy policy https://registrar.gmu.edu/students/privacy/</td></tr>
<tr><td>E-Mail Policy</td><td>Web: masonlive.gmu.edu
Mason uses electronic mail to provide official information to students. Examples include notices from the library, notices about academic standing, financial aid information, class materials, assignments, questions, and instructor feedback.

Students are responsible for the content of university communication sent to their Mason e-mail account and are required to activate that account and check it regularly.

Students are also expected to maintain an active and accurate mailing address in order to receive communications sent through the United States Postal Service.

<i>(From the 2017-18 Catalog – catalog.gmu.edu)</i></td></tr>
<tr><td>Course Grading & Evaluation</td><td>Course Grading policy:
<table>
<tr><td>Written Tests</td><td>30%</td></tr>
<tr><td>Technology Presentation/Tutorial (recorded)</td><td>4%</td></tr>
<tr><td>Writing Assignment 1 (WI)</td><td>6%</td></tr>
<tr><td>Writing Assignment 2 - SWE Topic Presentation/Essay (WI)</td><td>6%</td></tr>
<tr><td>~~Discussion Board (WI)~~</td><td>~~3%~~</td></tr>
<tr><td>Instructor Evaluation (of each team member)</td><td>8%</td></tr>
<tr><td></td><td></td></tr>
<tr><td><b><u>Project Deliverables</u></b>: (43%)</td><td></td></tr>
<tr><td>P0: Team Formation – Product Vision</td><td>1%</td></tr>
<tr><td>P1: Requirements</td><td>6%</td></tr>
<tr><td>P2: API & Testing</td><td>4%</td></tr>
<tr><td>P3: Software Architecture</td><td>5%</td></tr>
<tr><td>P4: Application Testing</td><td>5%</td></tr>
<tr><td>P5: Presentation/Video</td><td>7%</td></tr>
<tr><td>P6: Writing Assignment 3 - Project report - (WI)</td><td>7%</td></tr>
<tr><td>Peer Evaluation - Midterm</td><td>2%</td></tr>
<tr><td>Peer Evaluation – Final</td><td>6%</td></tr>
</table>
The following scale will be applied to compute final course letter grades:
A+ (>= 98.0%) A (>= 92.0%) A- (>= 90.0%)
B+ (>= 88.0%) B (>= 82.0%) B- (>= 80.0%)
C+ (>= 78.0%) C (>= 72.0%) C- (>= 70.0%)
D (>= 60.0%) F (< 60.0%)</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Written Test (30%)</td><td>~4 tests</td></tr>
<tr><td>Writing Assignments (WI)</td><td>1) Writing Assignment 1 (6%)
2) Writing Assignment 2 - SWE Topic Essay (6%)</td></tr>
</table>

(22%)	3) Writing Assignment 3 - Project Report (7%) ~~4) Discussion Board (3%): ~4 online discussions~~
Technology Presentation/Tutorial (4%)	Each team (5 to 7 students) will select a combination of technologies to study and present to the class (recorded). A hands-on demonstration is expected. A list of potential technologies is provided at the end of this document. Special care should be given to the ability of testing pure business logic and GUI-heavy applications. This is a very crucial preparation for the project (15 to 30 minutes, recorded).
SWE Topic Pres/Essay (WI) (6%)	One or more students will pick and present a SWE topic not covered in this course. The topic should be based on research paper(s) and/or industry article(s). The deliverable will also include an essay that will satisfy part of the WI requirements. If several students are involved, the presentation length and essay size should be scaled accordingly.
Team Project (43%)	CS 321 will have a software engineering project that requires students to participate in working teams where students organize, manage, and practice the software engineering lifecycle. The team project will cover software requirements, architecture, design, coding, and testing. Your final grade for this component will include peer evaluation grades provided by your teammates. Also, each team member will be assessed by the instructor.

Below are some of the technologies that previous CS321 teams have used:

1) Backend: Nodejs, Expressjs, EJS, and Mongodb
2) REACT
3) Frontend: HTML,CSS/SCSS, Javascript, and Bootstrap 5
4) Backend: Nodejs, Expressjs, EJS, Mongodb, Multer, Gridfs, Jest, and Supertest
5) Spring Boot, JAVA, My SQL and CSS/BootStrap/HTML
6) QUnit/Jest/Javascript
7) JUnit, Mockito, AssertJ
8) Flutter, Java Spring, HTTP, MongoDB (NoSQL)
9) Firebase Authentication
10) Firestore
11) Statement coverage tools (Jacoco, Flutter’s testing framework)
12) Selenium, Jest
13) JavaScript/TypeScript
14) RESTful API
15) Angular/React
16) MEAN stack: JavaScript-based framework (MongoDB, Express, Angular, and Node)
17) Relational vs. NoSQL databases
18) Cloud computing: AWS, Azure, Google Cloud…Heroku
19) Dependency Injection frameworks
20) Spring, SpringBoot
21) Issue Tracking systems, e.g. JIRA
22) DevOps CI/CD
23) Mobile Development: Android, IOS, Flutter
24) Testing tools, Coverage Tools, GUI Testing Tools
25) Agile supporting tools (e.g., estimation, prioritization)
26) Chrome Extensions, Discord Bots, Twitter Bots
27) Other

Some useful links:

- Spring Boot Annotations: https://www.baeldung.com/spring-boot-annotations
- Thymeleaf with Spring: https://www.baeldung.com/thymeleaf-in-spring-mvc
- Mockito tutorial series: https://www.baeldung.com/mockito-series
- **Jest and Nodejs Sources**
  - https://www.youtube.com/watch?v=FgnxcUQ5vho&t=197s
  - https://www.youtube.com/watch?v=ENrzD9HAZK4

- https://www.udemy.com/course/the-full-stack-web-development/
- https://www.youtube.com/Spring Security Tutorial https://www.w3schools.com/html/
- https://getbootstrap.com/
- Jest Testing Online Resources
  - https://vhudyma-blog.eu/3-ways-to-mock-axios-in-jest/
  - https://medium.com/wesionary-team/mocking-axios-in-jest-c68933a1a4fb

Other:
https://www.leighhalliday.com/mocking-axios-in-jest-testing-async-functions
https://www.javaniceday.com/post/mock-axios-with-jest
https://jestjs.io/docs/getting-started
https://testing-library.com/docs/react-testing-library/intro/
https://www.npmjs.com/package/supertest
https://stackoverflow.com/questions/64098340/mocking-with-jest-supertest-in-typescript
https://stackoverflow.com/questions/68353437/supertest-jest-how-to-force-or-mock-a-400-status-code

https://blog.logrocket.com/user-authentication-firebase-react-apps/
https://youtu.be/1HWVUkCTiM8
https://www.guru99.com/download-install-node-js.html
https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript
https://www.youtube.com/watch?v=w7ejDZ8SWv8&ab_channel=TraversyMedia