Welcome to CPEN 321!

W1 L1: Introductions, Goals, and Logistics

Agenda

- Essential information about the course
- Introductions (about me, about you)
- Course goals and format
- Your next steps

What This Course is About

→ This is **not** a course in programming

- It assumes you already took CPEN221 and CPEN322
- You can write code
- You can self-learn to develop Android client and Node.js backend applications
  - Introductory tutorials and reference materials are in Canvas
  - TAs are available for Q/A during the first two lab sessions

Delivery Mode: Both Summer Terms, Online

1. Lectures: Mon. and Wed., 3:30-5pm, MacMillan 360
   - Attendance is part of the grade
     … and is strongly encouraged: in-class discussions are essential
     to your success in the course (and are fun)
   - Please see announcements on Canvas about temporary classroom change and make-up-Monday date

2. Labs: Wed., 1-3pm, MacLeod 3018, 4002, 4006
   - Attendance is part of the grade
     …and you need to be there to work on your project
     in collaboration with your team members
   - Room will be assigned per project, not per registration
   - (more on this later)

Grading

- **Project** – 60%
  1. M1 (App Frontend and Backend): individual, 12%
  2. M2 (Requirements): group, 5%
  3. M3 (Design), group, 7%
  4. M4 (MVP): group, 12%
  5. M5 (Code Review): group, 5%
  6. M6 (Testing): group, 7%
  7. M7 (final release): group 12%
- **Mid-term 1** – 12%
- **Mid-term 2** – 14%
- **Participation** – 14%
  1. Peer-evaluation by team members – 6%
  2. Active participation in class and labs – 4%
  3. Helpful answers to piazza questions – 4%
     - Questions that were answered earlier or questions for which answers can be found in the course syllabus will not be answered again.
     - Repeated questions of this kind will cause mark deduction

What's
NEW?

AI Technologies

Will be allowed and will be part of the course (more on this later)

Questions?

Agenda

- Essential information about the course
- Introductions (about me, about you)
- Course schedule and logistics
- Your next steps

Julia Rubin

Associate Professor, University of British Columbia, Vancouver, Canada
Canada Research Chair in Trustworthy Software
Lead of UBC Research Excellence Cluster on Trustworthy ML

Earlier:

- Senior Software Engineer, Actona Technologies
  (acquired by Cisco for $82 million USD)
- Research Staff Member and Research Group Manager in IBM Research
- Postdoctoral researcher at EECS, MIT, USA
- PhD in Computer Science – University of Toronto, Canada

Research Focus: Quality, security, and reliability of software and AI systems

Prof. Rubin Research: Reliable, Secure and Sustainable Software Lab

Mobile Software

malware detection, efficient testing and analysis

Trustworthy ML

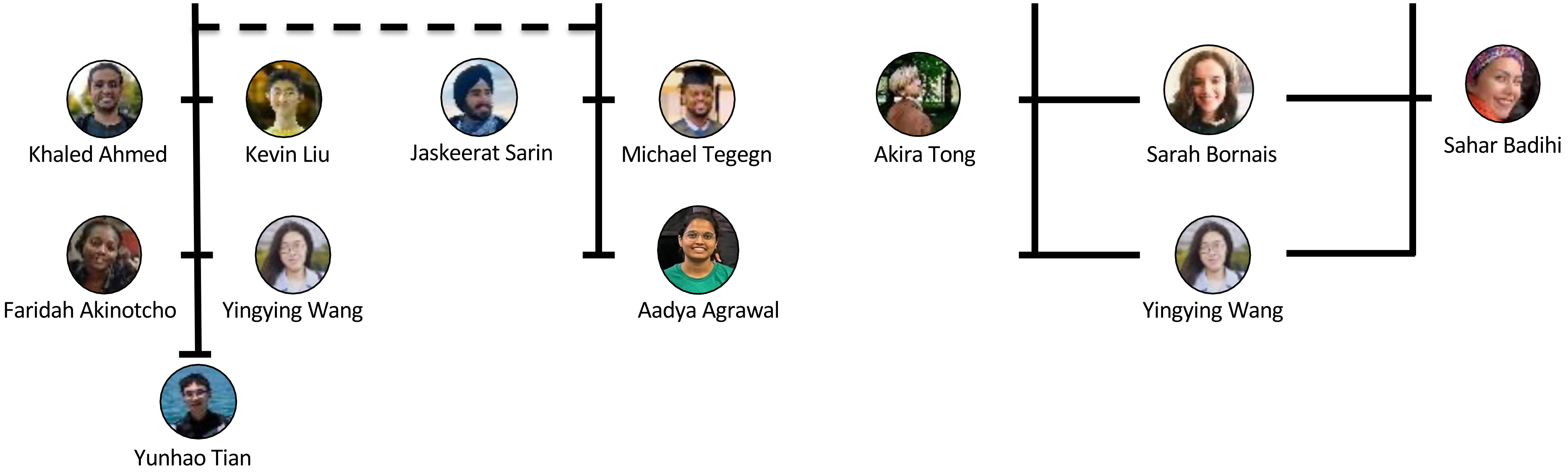
reliability, data quality, explainability

Cloud-based Software

security, performance, architecture, quality

Collaborative SE

reuse, integration and upgrade management



Khaled Ahmed

Kevin Liu

Jaskeerat Sarin

Michael Tegegn

Akira Tong

Sarah Bornais

Sahar Badihi

Faridah Akinotcho

Yingying Wang

Aadya Agrawal

Yingying Wang

Yunhao Tian

Prof. Rubin Research:
Reliable, Secure and Sustainable Software Lab

Mobile Software

Trustworthy ML

Cloud-based Software

Collaborative SE

- Develop novel approaches for making
  - Software and AI systems more reliable
  - Software deployment less costly
  - Software developers more productive

- Relying on
  - automated code analysis
  - formal methods
  - ML
  - exploratory research

Course Staff

Instructor: Prof. Julia Rubin

Office hours: Mon. 5-6pm, KAIS 4053 (or by appointment)

TAs:

- Sahar Badihi
- Faridah Akinotcho,
- Michael Tegegn,
- Yunhao Tian,
- Yingying Wang

What about you?

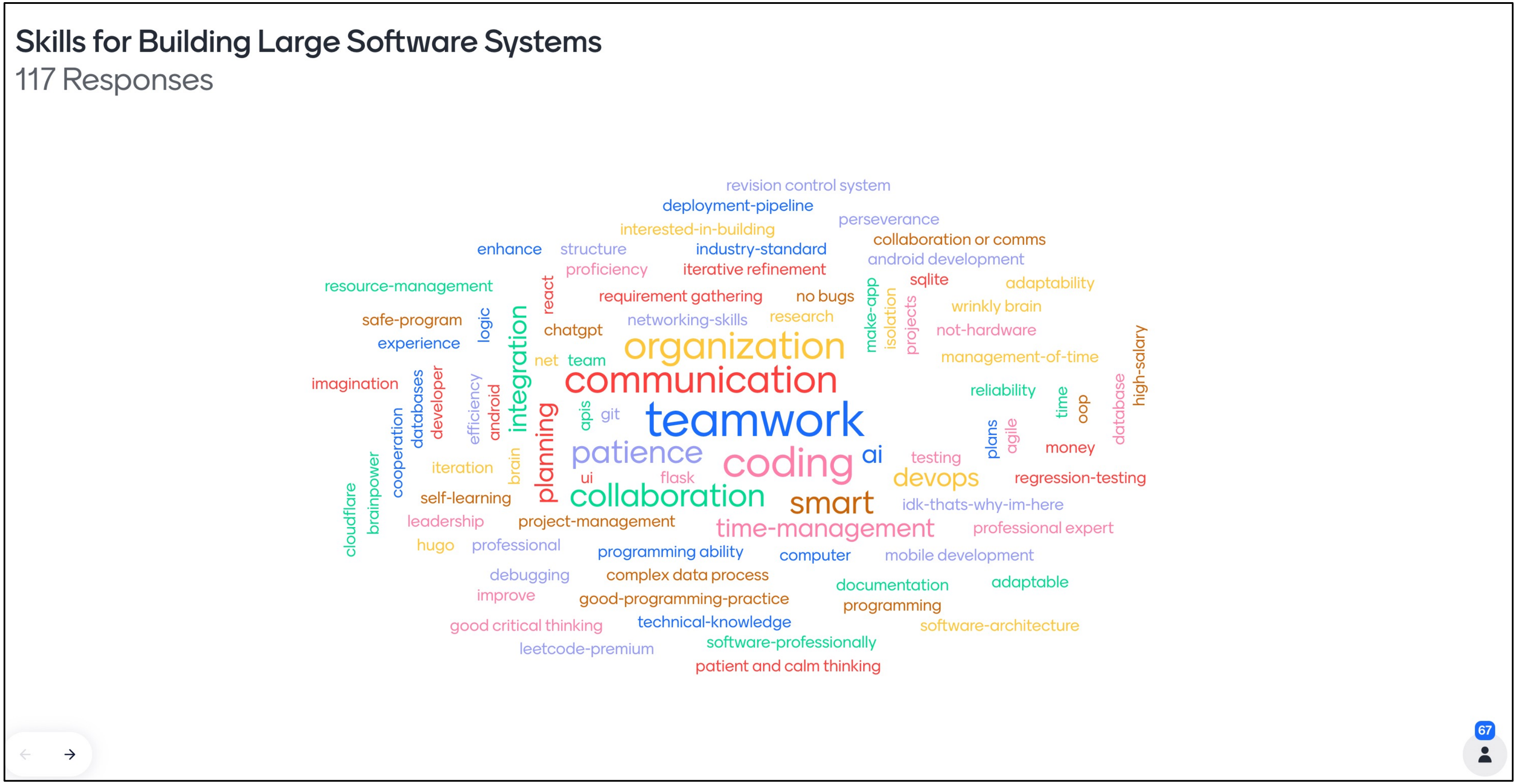
www.menti.com

Code: XXX

What do you plan to learn in this course?

www.menti.com

Code: XXX



Skills for Building Large Software Systems

117 Responses

What You Learn in This Course

- How to build large software-intensive systems
- *Software Engineering ≠ Programming*

What You Learn in This Course

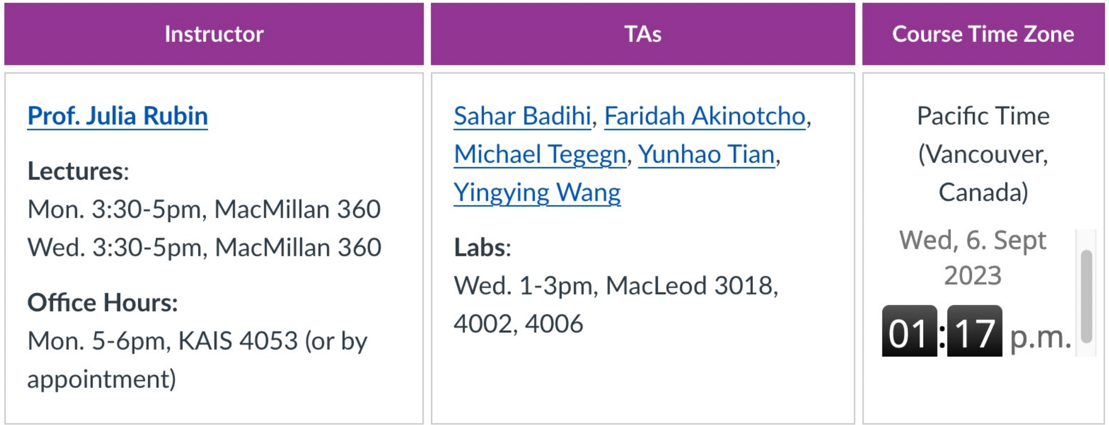
- How to build large software-intensive systems
  - Define **requirement**
  - **Design** the system (thorough planning)
  - Write **high-quality code** (and be able to maintain it)
  - Use version control systems efficiently
  - **Test** (and other forms of validation and verification)
  - Deal with **uncertainty**
  - **Plan** and work in a **team**
    - … under time pressure and tight deadlines.

- Strengths and weaknesses of modern generative AI technologies
- Efficient technical communication and presentation skills

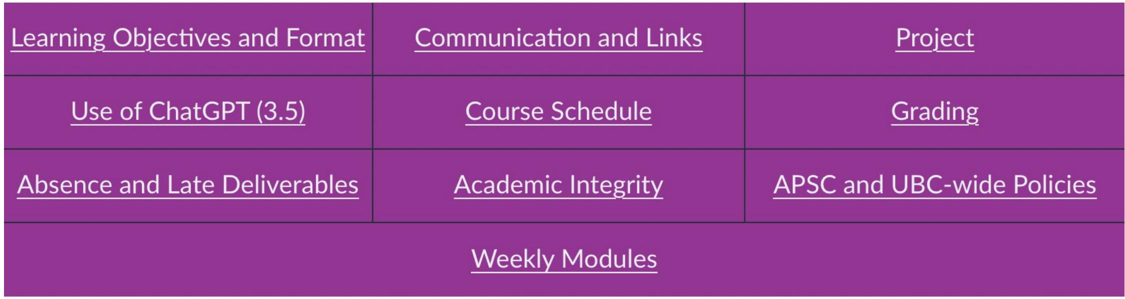
How?

- “In theory” (lectures) and “in practice” (projects)
  - Trying it for real

- Full course syllabus is on Canvas (please read carefully!)



Instructor	TAs	Course Time Zone
Prof. Julia Rubin **Lectures:** Mon. 3:30-5pm, MacMillan 360 Wed. 3:30-5pm, MacMillan 360 **Office Hours:** Mon. 5-6pm, KAIS 4053 (or by appointment)	Sahar Badihi, Faridah Akinotcho, Michael Tegegn, Yunhao Tian, Yingying Wang **Labs:** Wed. 1-3pm, MacLeod 3018, 4002, 4006	Pacific Time (Vancouver, Canada) Wed, 6. Sept 2023 01:17 p.m.



Learning Objectives and Format	Communication and Links	Project
Use of ChatGPT (3.5)	Course Schedule	Grading
Absence and Late Deliverables	Academic Integrity	APSC and UBC-wide Policies
Weekly Modules		

Course Prerequisites

- CPEN 221, CPSC 210, or equivalent
- CPEN 322 (co-requisite but is not given this term)

Communication

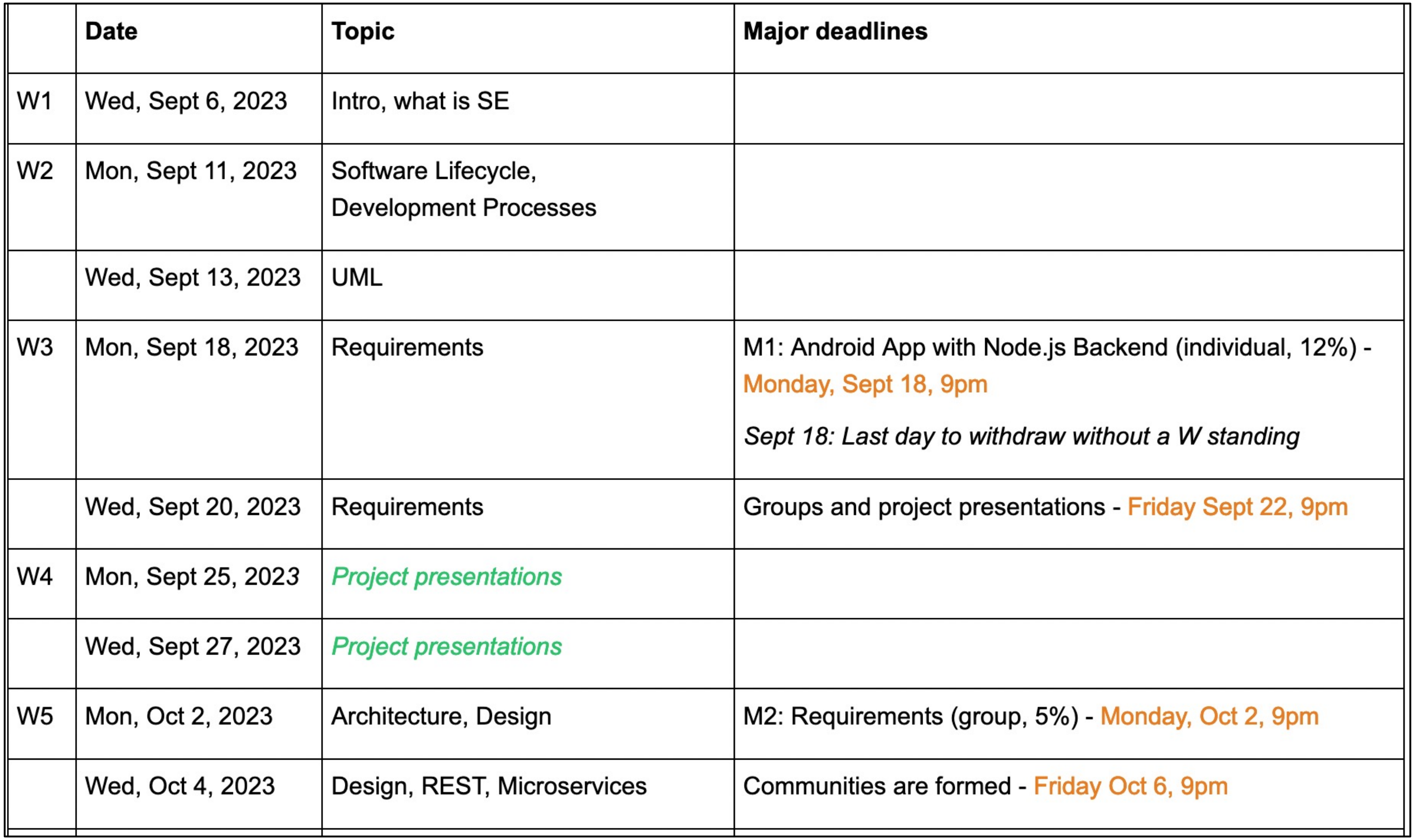
- https://canvas.ubc.ca/ ← main info hub
  - Full syllabus, assignments, lecture slides, tutorials, grades, Piazza link, etc.

- https://people.ece.ubc.ca/mjulia/teaching/CPEN321-W23T1/index.html
  - Another copy of the syllabus

- https://piazza.com/ubc.ca/winterterm12023/cpen321
  - Discussions, Q/A
  - Private messages to course staff (**I will reply to Piazza posts but not emails**)

Piazza servers might be hosted outside of Canada.
If you have issues using these services, please email me by the end of this week!

Tentative Schedule (full version is on Canvas)



	Date	Topic	Major deadlines
W1	Wed, Sept 6, 2023	Intro, what is SE	
W2	Mon, Sept 11, 2023	Software Lifecycle, Development Processes	
	Wed, Sept 13, 2023	UML	
W3	Mon, Sept 18, 2023	Requirements	M1: Android App with Node.js Backend (individual, 12%) - Monday, Sept 18, 9pm *Sept 18: Last day to withdraw without a W standing*
	Wed, Sept 20, 2023	Requirements	Groups and project presentations - Friday Sept 22, 9pm
W4	Mon, Sept 25, 2023	*Project presentations*	
	Wed, Sept 27, 2023	*Project presentations*	
W5	Mon, Oct 2, 2023	Architecture, Design	M2: Requirements (group, 5%) - Monday, Oct 2, 9pm
	Wed, Oct 4, 2023	Design, REST, Microservices	Communities are formed - Friday Oct 6, 9pm

Main Lecture Topics

- Software Lifecycle, Development Processes
- UML
- Requirements
- Architecture, Design
- REST, Microservices
- Teamwork, Advanced Version Control
- Continuous Integration, DevOps
- Code Reviews, Anti-patterns
- Verification and Validation, Analysis
- Testing

Project

- A client-server software system

- **Client side**: a mobile application
  - Must run natively on Android (not in a browser)
  - Must be written in Android/Java
    - Using other frameworks, e.g., Expo, ReactNative, etc. is not permitted.
  - Must run on at least one real device and on a Google Pixel 3 emulator running Android Q (API 29).

- **Server side**: node.js backend with either MySQL or MongoDB database
  - Must run in a cloud (AWS, Azure, Google, IBM, etc.)
  - Microsoft Azure credit is available for interested teams

Minimal Project Requirements

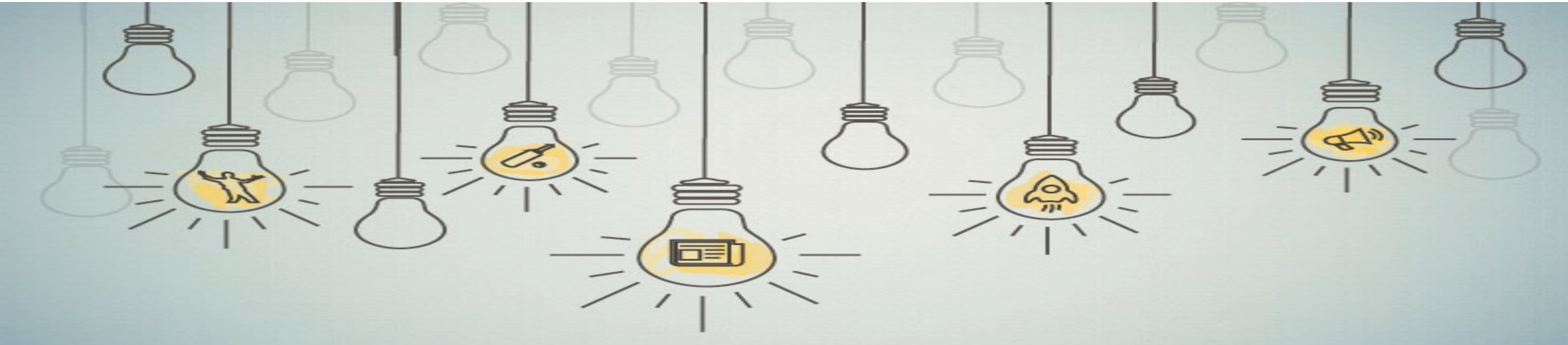
- Use external authentication service
  - E.g., Google or Facebook authentication
- Use at least one additional external service
  - Google calendar, Google maps, etc.
- Real-time updates
  - E.g., Multi-users chats, push notification, etc.

Projects that satisfy these requirements will get up to 85 points.

Beyond Minimal Requirements

- Implement “interesting” / non-trivial logic in either front-end or back-end component (beyond simple database read/update/delete)
- Non-trivial application
  - We saw a lot of scheduling and event management systems

… use your creativity… there are also awards for top projects at the end



Past Winners – 1/3

JobToGo streamlines the job search process and makes it more interactive and convenient by using a Tinder-like interface.

Main Features:

- Parses PDF/photocopy of resume for past skills/experiences
- Uses a friendly user interface for “liking” and “disliking” jobs (by swiping right/left)
- Recommends jobs to users based on TF-IDF weighting of keywords in job descriptions and users' previous "liked" and "disliked" jobs
- Sends email of the list of “liked” jobs to user
- Builds a community by having a friend system where a user can send jobs to their friends

Past Project Examples – 2/3

UBCSafe is an app that provides a peer location-monitoring and alert system, and easy access to local safety resources

Main Features:

- A user can subscribe as a traveller or as a watcher
- Travellers can share their location and select a destination
- Travellers can select a watcher for the duration of their trip, based on their preferences (e.g., age, gender, rating) and location proximity
- Companion sessions are created and scheduled in the calendar
- In session notifications (e.g., “Near Destination”, “Reached Destination”, etc.) and chat features
- Travellers and watchers are rated (like in Uber)
- Integrated Campus Safety Resources (SafeWalk call button, Blue Phone locator, Security Alert feed, etc.)
- Integrated Google/Facebook login, Google maps, Google calendar, push notifications

Past Project Examples – 3/3

An app that automatically selects and navigates the user to the closest Car2Go parking location with free spots

Main features:

- Integration with Car2Go REST APIs
- Integration with Google Maps
- Real-time processing of changes in availability and re-calculation of optimal target location

More examples:
https://docs.google.com/document/d/1l_DX9GbduIRSMQxV9nPnOfwWWd--W55895v5ASaWqS0

Your Project

- M1: Android App with Node.js Backend (individual)
- *(Make groups of 4: 2 front-end and 2 back-end, project ideas presented in class)*
- M2: Requirements (group)
- M3: Design
- **M4: MVP (group)**
- M5: Code Review (group)
- M6: Testing (group)
- M7: Final Release (group)

Timeline for Building Teams

- By Friday, September 22, 9pm
- Start early and pick your team members wisely!
- Contact each other (Piazza, etc.)
- No team of 4 by the deadline?
  - We will assign you to one randomly
  - We will “fill up” all teams to have 4 members, potentially “breaking” teams of 3 or less to multiple teams

Lab Sessions

- Time for each team to meet and work on the next project milestone
- Facilitated by TAs
  - I.e., you have the opportunity to ask clarification questions/help with your particular project
- Attendance is mandatory
- Active participation is part of the grade
- The structure of the lab sessions will be explained by TAs starting W3

Mini-Tutorials

- Android and node.js mini-tutorials are on Canvas
- TAs will answer questions about the tutorials and provide help with your milestones during the first two labs

→ Watch the tutorial in advance

- take around 3 hours in total and you might need to also check other material online
- Android versions change several times per year; tutorials might be slightly outdated and you need to complete the missing info by searching online

Important Clarification: You are not expected to finish working on a milestone during the lab time and will need to complete the work outside of the class hours.

Late Deliverables

- Each **group** → **5 grace days** per term in total
  - can be used for any group milestones (M2-M6).
  - Can use the grace days as you wish
    - i.e., all three days on one deliverable, distribute across deliverables, etc.
    - grace days are counted as integers, i.e., if you are one hour late, that counts as one grace day.
  - Once all grace days are used → late submissions receive **0**
    - In exceptional circumstances, apply for AppSci's academic concession

- There are **no grace days for M1 and M7**. Late submissions will be deducted **30% for each late day** (which also count as integers).

Team Work

- Part of your grade
- Meet often
  - Lack of communication and synchronization between team members is one of the main reasons projects did not succeed in the past
- Each team member reports on individual work accomplished for each milestone
- If there are problems, discuss them with course staff early

Tools and Code Management

- All code of your project (front-end and back-end) should be in one Git repository
  - Well-structured internally
  - (https://www.wired.com/2015/09/google-2-billion-lines-codeand-one-place/)
- Create GitHub or Bitbucket account
  - feel free to create a throw-away/anonymous account for this course
  - Must give all TAs access to this account
- Getting started with Git
  - Atlassian Git Introduction
  - Shorter Atlassian Guide
- Specification for the test and code review frameworks will be in M6 and M7 documents (released within the following weeks)
  - Do not start earlier, even if you know how

ChatGPT and Assistive AI

- AI tools introduced a dramatic shift in the software industry
- The use of ChatGPT in this course, as an assistant in completing the assignments, is allowed!
  - Focus on documenting and critically analyzing the strengths and weaknesses of AI technologies → **one of the educational objectives of the course**
- More instruction on how to use ChatGPT will follow
- **All usages should be explicitly declared and documented.**
  - Undocumented use will be considered academic misconduct and will be treated accordingly.
- Exciting innovation that might deserve research report
  - More to follow…

Grading (again)

- **Project** – 60%
  1. M1 (App Frontend and Backend): individual, 12%
  2. M2 (Requirements): group, 5%
  3. M3 (Design), group, 7%
  4. M4 (MVP): group, 12%
  5. M5 (Code Review): group, 5%
  6. M6 (Testing): group, 7%
  7. M7 (final release): group 12%
- **Mid-term 1** – 12%
- **Mid-term 2** – 14%
- **Participation** – 14%
  1. Peer-evaluation by team members – 6%
  2. Active participation in class and labs – 4%
  3. Helpful answers to piazza questions – 4%
     - Questions that were answered earlier or questions for which answers can be found in the course syllabus will not be answered again.
     - Repeated questions of this kind will cause mark deduction

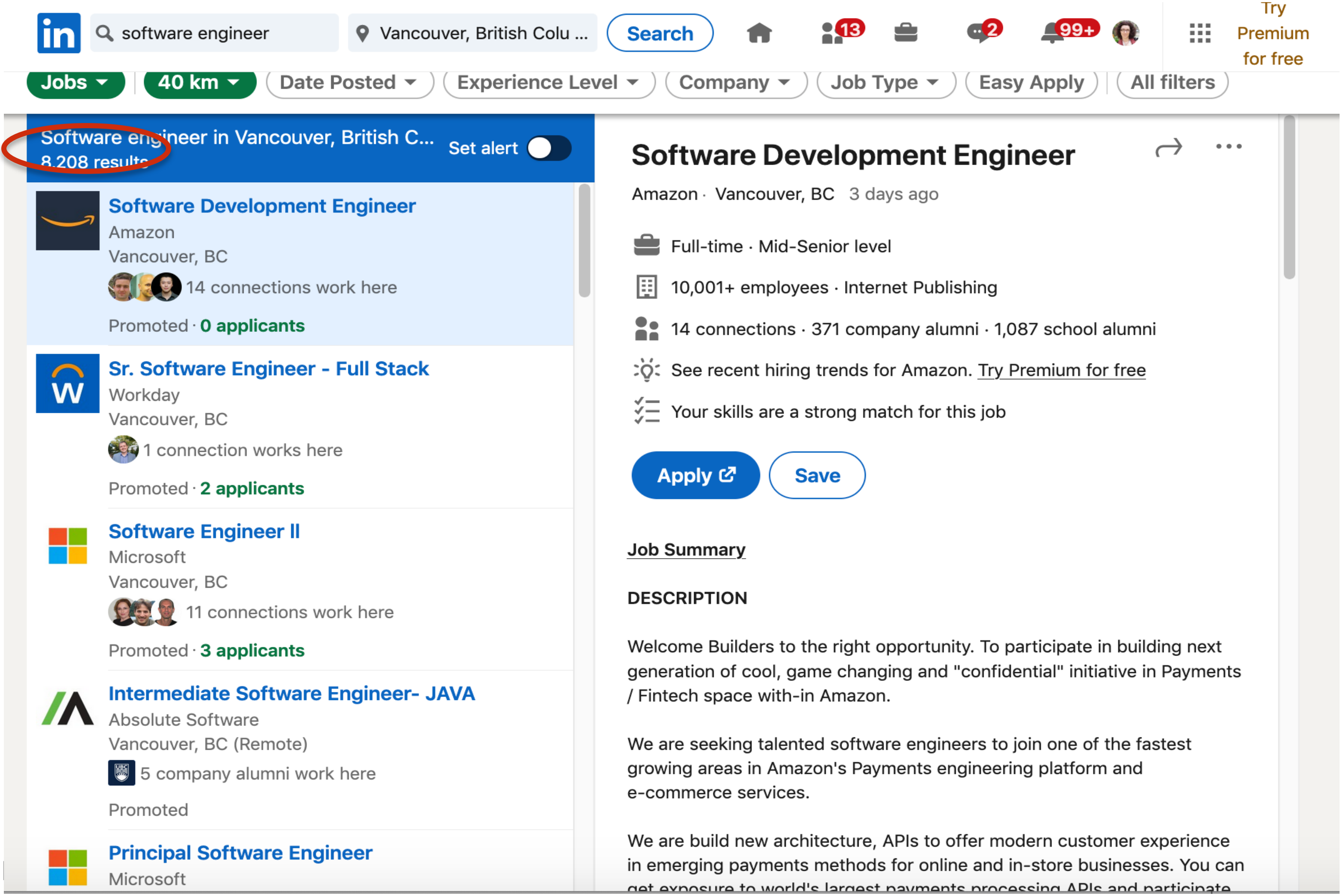
Why?

Because it's fun!

To simulate real development settings (as much as we can):

- Learn to work individually (when needed) and also collaborate with other team members (when needed)
- Practice following software development processes
- Develop technical communication skills
- Learn state-of-the-art technologies
  - that are constantly changing, so independent learning is an essential skill for SE

Why?



Software engineer in Vancouver, British C...
8,208 results

Software Development Engineer
Amazon
Vancouver, BC
14 connections work here
Promoted · 0 applicants

Sr. Software Engineer - Full Stack
Workday
Vancouver, BC
1 connection works here
Promoted · 2 applicants

Software Engineer II
Microsoft
Vancouver, BC
11 connections work here
Promoted · 3 applicants

Intermediate Software Engineer- JAVA
Absolute Software
Vancouver, BC (Remote)
5 company alumni work here
Promoted

Principal Software Engineer
Microsoft

Software Development Engineer

Amazon · Vancouver, BC 3 days ago

Full-time · Mid-Senior level

10,001+ employees · Internet Publishing

14 connections · 371 company alumni · 1,087 school alumni

See recent hiring trends for Amazon. Try Premium for free

Your skills are a strong match for this job

Job Summary

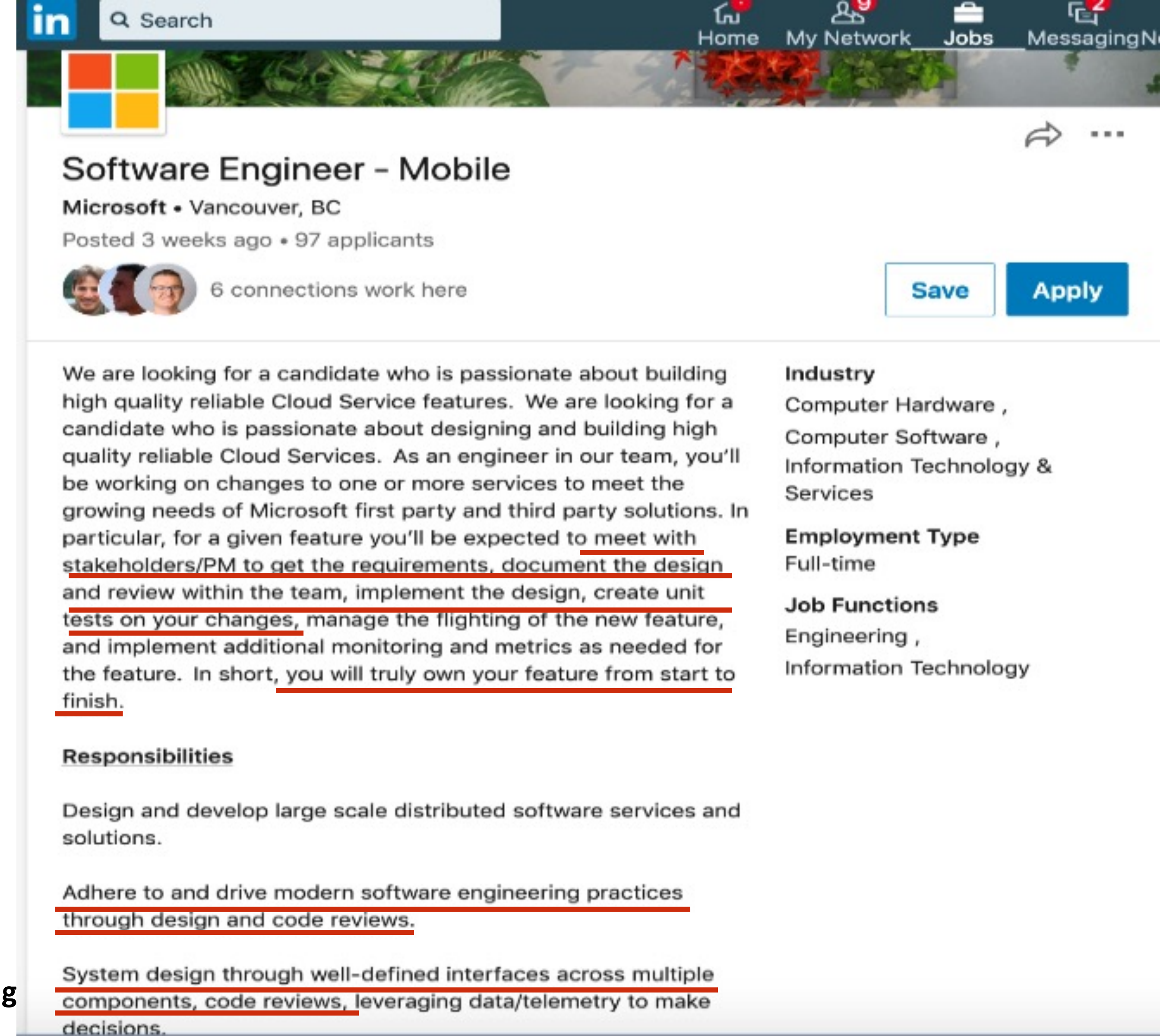
DESCRIPTION

Welcome Builders to the right opportunity. To participate in building next generation of cool, game changing and "confidential" initiative in Payments / Fintech space with-in Amazon.

We are seeking talented software engineers to join one of the fastest growing areas in Amazon's Payments engineering platform and e-commerce services.

We are build new architecture, APIs to offer modern customer experience in emerging payments methods for online and in-store businesses. You can get exposure to world's largest payments processing APIs and participate

Why?



Software Engineer - Mobile

Microsoft • Vancouver, BC

Posted 3 weeks ago • 97 applicants

6 connections work here

We are looking for a candidate who is passionate about building high quality reliable Cloud Service features. We are looking for a candidate who is passionate about designing and building high quality reliable Cloud Services. As an engineer in our team, you'll be working on changes to one or more services to meet the growing needs of Microsoft first party and third party solutions. In particular, for a given feature you'll be expected to meet with stakeholders/PM to get the requirements, document the design and review within the team, implement the design, create unit tests on your changes, manage the flighting of the new feature, and implement additional monitoring and metrics as needed for the feature. In short, you will truly own your feature from start to finish.

Responsibilities

Design and develop large scale distributed software services and solutions.

Adhere to and drive modern software engineering practices through design and code reviews.

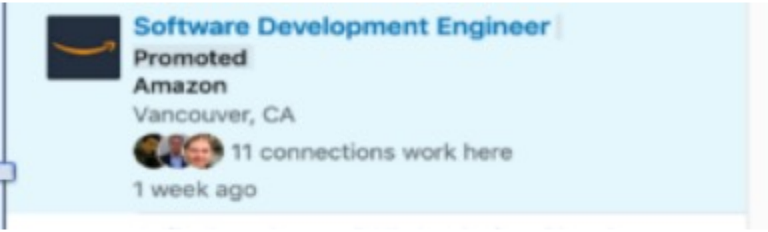
System design through well-defined interfaces across multiple components, code reviews, leveraging data/telemetry to make decisions.

Industry
Computer Hardware ,
Computer Software ,
Information Technology & Services

Employment Type
Full-time

Job Functions
Engineering ,
Information Technology

Why?



Software Development Engineer

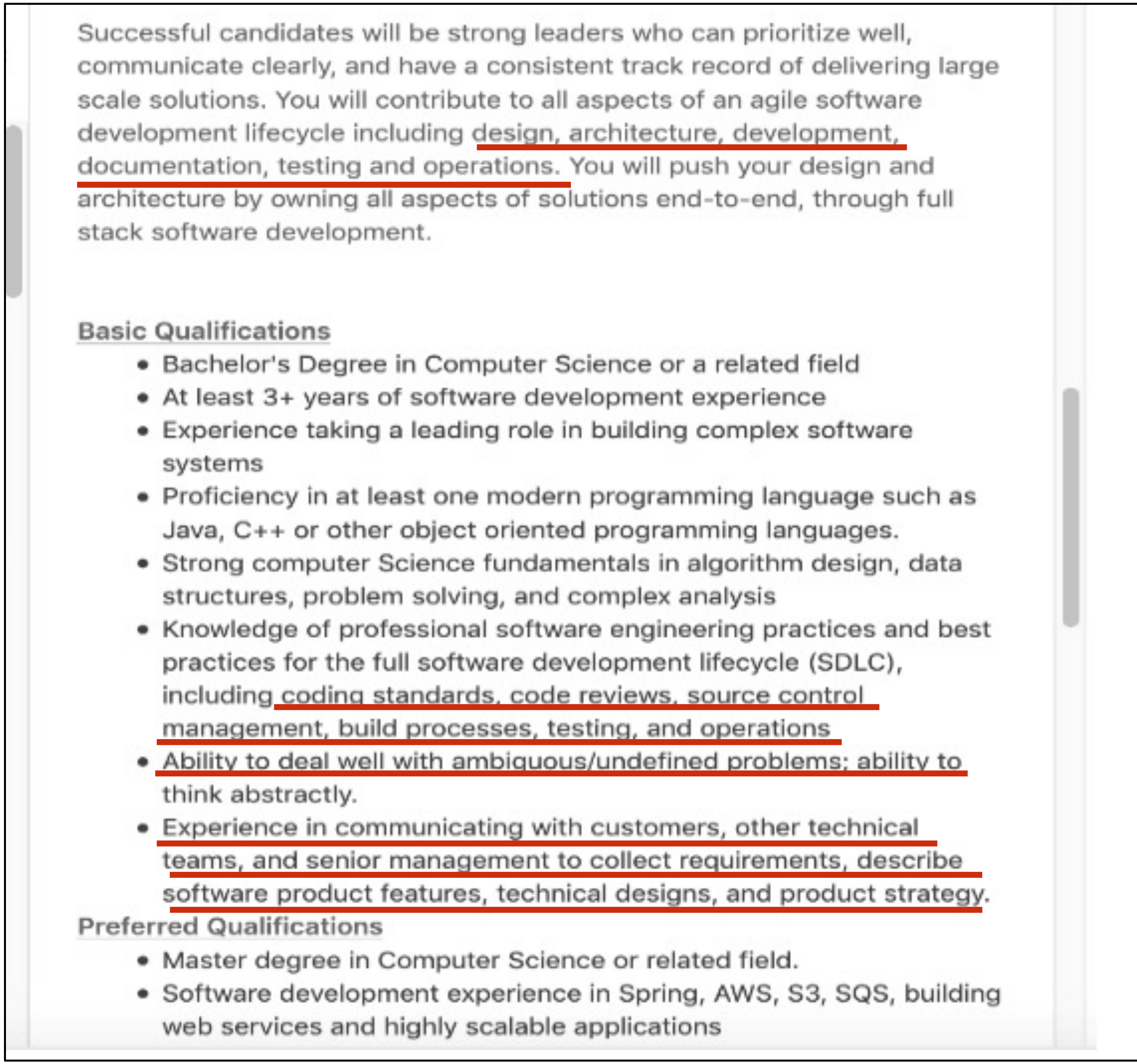
Promoted

Amazon

Vancouver, CA

11 connections work here

1 week ago



Successful candidates will be strong leaders who can prioritize well, communicate clearly, and have a consistent track record of delivering large scale solutions. You will contribute to all aspects of an agile software development lifecycle including design, architecture, development, documentation, testing and operations. You will push your design and architecture by owning all aspects of solutions end-to-end, through full stack software development.

Basic Qualifications

- Bachelor's Degree in Computer Science or a related field
- At least 3+ years of software development experience
- Experience taking a leading role in building complex software systems
- Proficiency in at least one modern programming language such as Java, C++ or other object oriented programming languages.
- Strong computer Science fundamentals in algorithm design, data structures, problem solving, and complex analysis
- Knowledge of professional software engineering practices and best practices for the full software development lifecycle (SDLC), including coding standards, code reviews, source control management, build processes, testing, and operations
- Ability to deal well with ambiguous/undefined problems; ability to think abstractly.
- Experience in communicating with customers, other technical teams, and senior management to collect requirements, describe software product features, technical designs, and product strategy.

Preferred Qualifications

- Master degree in Computer Science or related field.
- Software development experience in Spring, AWS, S3, SQS, building web services and highly scalable applications

It’s a Long Way…

What You Learn in This Course

- How to build large software-intensive systems
  - Define **requirements**
  - **Design** the system (thorough planning)
  - Write **high-quality code** (and be able to maintain it)
  - Use version control systems efficiently
  - **Test** (and other forms of validation and verification)
  - Deal with **uncertainty**
  - **Plan** and work in a **team**
    - … under time pressure and tight deadlines.

Your Immediate Next Steps

1. Read the course syllabus in detail!
2. Subscribe to Piazza and Discord
3. Watch Android mini-tutorial towards the lab tomorrow
4. Attend the tutorial for intro to Discord use, start working on M1, and ask questions about M1
   - Due in 10 days, May 26
5. Remember that M2 is due June 2
6. Find group members, due June 6
   - Use Piazza
7. Start thinking about your project scope and requirements
   - M3 is due June 9
8. *All milestones are in Canvas. Plan wisely for the rest of the term*

PLANNING
ANALYZE
DESIGN
SOFTWARE ENGINEERING
VALIDATION AND VERIFICATION
LOW-LEVEL-TEST
HIGH-LEVEL-TEST
PROGRAMMING