–

Aniket Arora ([anikaro](https://phonetool.amazon.com/users/anikaro" \t "_blank)) SDE L5 (Level 5)

Tenure: 2 years, 7 months | Training: MGHD & SDE FI | Interviews:

14

Coding (Data Structures and Algorithms)\*

Learn and Be Curious\*

Coding (Problem Solving)\*

inclined

Summary

I am inclined to hire the candidate. They demonstrated good problem-solving skills and coding abilities, with a strong understanding of trees and binary search. The candidate effectively explained both brute force and optimal approaches for both problems, and was able to implement the code and dry run it to produce correct output. They also demonstrated a good "Learn and Be Curious" scenario from their previous company, showing their ability to identify what to learn and solve challenging problems effectively.

Coding (Data Structures and Algorithms)\*

Strength

The candidate was asked two coding questions. The candidate understood the problems and asked the right questions about requirements. The candidate was asked about binary trees and binary search problems, and he demonstrated good knowledge of both data structures. For both problems, he was able to explain both the brute force and optimal approaches. The candidate demonstrated good knowledge of trees and algorithms required to solve the first problem and implemented the correct code without any bugs. They were able to dry run the code with the given input and produce the correct output. Candidate was able to tell time complexity and space complexity of both the problems.  
  
livecode link : https://livecode.corp.amazon.com/session/3a039456-c18b-4aa2-89ac-5f33b093d805

Learn and Be Curious\*

Strength

Questions Asked:  
Q1. Tell me about a time when you didn't know what to do next or how to solve a challenging problem.  
Q2. How do you identify what you need to learn?  
  
Candidate Response:  
  
The candidate shared a example from their previous experience working on display ads. They faced a significant challenge where they needed to scale their database records while simultaneously reducing query response time from 10 seconds to 1-2 seconds.  
  
To solve this problem, the candidate Identified multiple potential solutions and evaluated them   
- Implemented database sharding using domain-ID as the key  
- Improved data modeling by implementing a bucket pattern, which helped consolidate values based on keys  
- Optimized the one-to-many relationship in their data model, reducing data entries from 900 million to 200 million  
  
The final solution successfully:  
- Reduced query response time to 2-3 seconds  
- Effectively handled the increased database size  
  
As they were using MongoDB for their Database .To solve the problem he was not initially aware of MongoDB's data modeling capabilities, so he had to learn how to implement proper data modeling techniques. This learning helped him to restructure their database to better suit their use case.

Coding (Problem Solving)\*

Mild Strength

The candidate was asked two coding questions. The candidate's approach to solving the problems was good; he gathered the requirements and presented both brute force and optimal approaches. The candidate took a little bit extra time to solve the first question. Because of that time consumption, for the second problem, the candidate was able to come up with an optimized solution but was not able to complete the code covering all edge cases due to time constraints. However, he was able to identify and call out the missing edge cases.  
  
livecode link : https://livecode.corp.amazon.com/session/3a039456-c18b-4aa2-89ac-5f33b093d805

Submitted Feb 5, 2025 at 11:52PM

–

Divya Dharshini K ([divyadd](https://phonetool.amazon.com/users/divyadd" \t "_blank))S Software Development Engineer (Level 5)

Tenure: 2 years, 10 months | Training: MGHD & SDE FI | Interviews:

7

System Design\*

Invent and Simplify\*

inclined

Summary

The candidate was able to come up with the functional & non-functional requirements for the design question. He proposed the initial solution and was able to iteratively improve the solution based on the requirements. He was able to navigate through the sub-problems and identify proper solutions to them by discussing the trade-offs for each of them. He was also able to map his past projects to the given LPs and explained the complexities involved and how he was able to solve them. So, I am inclined to hire the candidate.

System Design\*

Strength

Design Google docs:  
  
Come up with your own requirements?  
-Candidate was able to come up with functional and non-functional requirements. With due guidance, he was able to narrow down few requirements.  
-Reference - https://client.ext.bluescape.ee-infra.aws.dev/UyXi7l74xwf0VvkF7tXY?targetRect=%5B-1049%2C-426%2C1125%2C667%5D  
  
Multiple users can edit at the same. How to achieve?  
- Use web socket - bi-directional communication.  
  
RT collaboration is the main feature - if making an edit, what are the workflows happens? How you will handle non-functional requirements?  
- Transferring the whole file will have high network call. So for every edit use indexing.  
- Naive way - use locking - from collab service (CS)  
  
How to ensure which is first request in CS?   
-Add some kind of queue to know which order the request comes first - lock based queue - kafka.  
  
Instead of queue , can we use timestamp?   
- Take server/client time. Can't take client time as it contains network delay.  
- So queue is better?  
- Use web socket- it will push to lock and gets the lock to DB.  
- Use operational transformation - this is to ensure proper indexing (martin kelper)  
  
How do you ensure scalability & latency?  
Latency  
- Use caching service for quick look ups  
- Use one web socket per document - cache  
  
Scalability  
- Web socket service can be scaled horizontally  
- Partition the queue by doc ids - so that the order is maintained  
- Use DB Sharding, to retrieve it based on doc/req it  
- Use timestamp based on indexing

Invent and Simplify\*

Mild Strength

1) Explain a situation where you solved a complex problem with a simple solution?  
  
Candidate worked on a project -> There are publishers who help publish the forms and buyers for buying. Their system acts as a middle man. Eg: GeeksForGeeks.  
Till last year its a manual process. Goal is to eliminate the middle man.   
  
Solution:  
Allow to generate HTML forms based on requirements. But generating html for non tech man is complex.  
  
To solve this there are 2 potential options :  
1) Use LLM to generate HTML based on the requirement like logo , UI design, etc.  
Potential complexities -> proper prompting is required, different for differnet users because no templates.  
  
2) Simple template rendering -> define logos , text asset , based on user input , we can render the template. This is the chosen option.  
Created pre-defined template -> master template and then fill in the gaps.  
  
Follow ups:  
How do you make sure that the solution fits the problem?  
- 50k$ to external vendor - unnecessary cost is eliminated  
- 4 to 5 hrs of time is wasted to manually build   
All these are solved.  
  
  
  
2) Describe a situation where you influenced your team to choose the right solution to a given problem?  
  
Project -> Display adds platform. Cookie master dataset.  
  
Problem: Cookies are short-lived. So need to refresh the dataset and it is huge (900million entries). Cookie started to overload. Had contract with external vendors, posting it to s3, 10k files per day.  
  
Solution:  
To consume all at a time zip it and store in DB. File has domain & cookie id (old -> update the existing, new -> add it). Already had batch solution -> cost memory issues -> CPU usage -> for not affecting the existing flow.  
So introduces Queue mechanism. Rabbitmq -> for every job.  
Initially proposed to use Lambda. But to make it cloud agnostic moved to queue.   
  
Follow ups:  
Batch was already there, so what is different?  
- 12 hr window to process all files -> batch processing. So queue mechanism worked well.  
- Daily latest data has been received upto date. 90% acceptance rate. So more impressions.

Hide interview notes (764 words)

Introduction:  
  
Project:  
  
BTB SA company:  
Display adds - specific use case.  
Display version -> implementation from scratch , end-to-end  
C# and .net code & AWS as cloud service.  
  
Load share company:  
LMD for various e-commerce  
Ledger for manager trip  
He owned the below:  
-Payment side of things & trips part  
-End-to-end shipment process   
-Warehouses & ERP tools for shipment flow  
  
Athena Health:  
HealthCare SAS company   
Arch tech solution to automate prescription and billing  
  
Design:  
come up with your own requirements:  
- ref quip  
  
- multiple users can   
- web socket - bi-directional communication  
  
RT collab is the main feature - if making an edit what are the workflows happens? how you will handle non-fucn requirements?  
expand any boxes wich you want. you can make assumptions as well but do call out?  
  
- transfering the whole file will have high neetwrk - for every edit  
- for every edit - indexing  
- naive way - lcoking - from collab service  
- how doendure which is first req in CS?  
- add some kind of queue whihc order req comes first  
- lock based queue - kafka  
- instead of q , use timestanp?   
- take server/client time? netwrk delay , cant take client time.   
- so queue is better?  
- websoct- it will push to lick - gets the lock to db  
  
operational transformation - martin lepler  
- to ensure proper indexing  
  
how do you ensure scalability & latency?  
latency  
- caching service for quick look ups  
- one web socket per doc - cache  
  
scalability  
- web soc service to horizontally scale  
- partition the queue by doc ids - the order is maintained  
- DB sharding 0 revireve it based on doc/req it  
- timstamp based on indexing  
  
Design:  
come up with your own requirements:  
- ref quip  
  
- multiple users can   
- web socket - bi-directional communication  
  
RT collab is the main feature - if making an edit what are the workflows happens? how you will handle non-fucn requirements?  
expand any boxes wich you want. you can make assumptions as well but do call out?  
  
- transfering the whole file will have high neetwrk - for every edit  
- for every edit - indexing  
- naive way - lcoking - from collab service  
- how doendure which is first req in CS?  
- add some kind of queue whihc order req comes first  
- lock based queue - kafka  
- instead of q , use timestanp?   
- take server/client time? netwrk delay , cant take client time.   
- so queue is better?  
- websoct- it will push to lick - gets the lock to db  
  
operational transformation - martin lepler  
- to ensure proper indexing  
  
how do you ensure scalability & latency?  
latency  
- caching service for quick look ups  
- one web socket per doc - cache  
  
scalability  
- web soc service to horizontally scale  
- partition the queue by doc ids - the order is maintained  
- DB sharding 0 revireve it based on doc/req it  
- timstamp based on indexing  
  
LP:  
1) Explain a situation where you solved a complex problem with a simple solution?  
Candidate worked on a project - publishers who help publish the forms   
buyers for buying   
he acts as a middle man  
step - mock proof of concept process  
requirements - confirm with biyer before sending it to the publisher - till last yr its a manual process  
iteration between vendor and buyer based on requirements  
- eliminate the middle man  
- generting html for non tech man is complex  
2 options   
- use llms to genearte html based on re of campign manager who then works with buyers - logo , ui  
- complexities - proper prompting , diff for diff users bcz no templates  
  
- simple template rendering define - define logos , text asset , based on user input , we can render the template  
- created pre-defined templaced - master template to fill in the gaps  
  
What is the actual requ to vendor?  
-   
  
How do you make sure that the solution fits the problem?  
- 50k$ to external vendor - unnecessary  
- 4 to 5 hrs of time is wasted to manually build  
  
What is the complexity?  
  
All these are solved.  
  
2) Influenced your team to choose   
Display adds platform. cookie master dataset  
cookie if short lived  
need to refresh the dataset and it si huge - 900million entries - master data - 4 months back - on step load - so cookie started to overload  
had contract with externla vendors, posting it to s3, 10k files per day  
soln to consume all at a time zip it and store in DB  
file has domain & cookie id - odd existing, new add it  
already had batch solution - cost memory issues - cpu usage - fo not affect esisting flow  
so introduces q mechanism. rapid mq - for every job - initially use lamdba proposal - but to make it cloud agnostic moved to q.   
  
batch already there, what os different?  
12 hr window to process all files. - batch processing  
q meachism worked well  
  
daily upto data is got. 90% accepatance rate. so more impressions.

Submitted Feb 16, 2025 at 5:35PM

–

Nishad Rahiman A ([nishadaz](https://phonetool.amazon.com/users/nishadaz" \t "_blank)) Software Development Engineer (Level 5)

Tenure: 3 years | Training: MGHD & SDE FI | Interviews:

18

System Design\*

Invent and Simplify\*

inclined

Summary

Rationale based on key dimensions:  
  
1. Technical Excellence (Strong Signal)  
System Design Performance:  
- Demonstrated strong technical acumen in designing collaborative document system  
- Proactively identified requirements and constraints  
- Made well-reasoned architectural decisions with clear tradeoffs  
- Showed depth in areas like:  
\* Real-time collaboration approaches  
\* Conflict resolution strategies  
\* Scaling considerations  
\* Performance optimization  
  
Professional Experience:  
- 5.5 years relevant experience across impactful domains  
- Strong technical foundation with modern stack:  
\* AWS cloud services  
\* .NET Core  
\* Large-scale data handling (900MM entries)  
- Diverse experience across different business domains:  
\* B2B SaaS platforms  
\* AdTech systems  
\* Healthcare solutions  
\* E-commerce applications  
  
2. Impact & Delivery (Meets Bar)  
Demonstrated Results:  
- Quantifiable business impact ($50K cost savings)  
- Efficiency improvements (4-5 hours time reduction)  
- Performance optimization (90% to 91% acceptance rate)  
- Successfully led technical improvements in ad platform refactoring  
  
3. Leadership & Ownership (Meets Requirements)  
Technical Leadership:  
- Led refactoring initiatives  
- Implemented innovative solutions (queue mechanism for cookie dataset)  
- Shows ownership in problem-solving approach  
- Demonstrates ability to drive technical decisions  
  
4. Scale & Complexity (Meets Bar)  
- Experience with large-scale data processing  
- Understanding of distributed systems  
- Ability to handle complex business requirements  
- Experience with mission-critical systems (payments, ads)  
  
Assessment Details:  
  
Strengths:  
1. System Design Capabilities  
- Strong architectural thinking  
- Clear understanding of scalability needs  
- Well-reasoned technical decisions  
- Comprehensive approach to requirements  
  
2. Technical Depth  
- Solid foundational knowledge  
- Practical experience with modern technologies  
- Understanding of distributed systems  
- Experience with scale  
  
3. Delivery Focus  
- Track record of shipping features  
- Measurable business impact  
- Focus on efficiency and optimization  
- Practical problem-solving approach  
  
Development Areas (Normal for L5 Level):  
- Continue developing broader influence skills  
- Expand impact measurement capabilities  
- Further strengthen innovation articulation  
- Build on leadership experiences  
  
Conclusion:  
The candidate meets the bar for L5 (SDEII) based on:  
1. Strong technical foundation and system design capabilities  
2. Proven track record of delivery and impact  
3. Appropriate level of technical leadership  
4. Relevant experience with scale and complexity  
  
Hiring Recommendation:  
HIRE - Candidate demonstrates the necessary technical strength, delivery capability, and leadership potential required for success at L5 level.  
  
Success Factors:  
1. Well-suited for teams requiring strong technical execution  
2. Can contribute immediately to system design and architecture  
3. Shows potential for growing into stronger technical leadership  
4. Has practical experience that aligns with Amazon's technical needs  
  
This assessment indicates the candidate has the right combination of technical strength, practical experience, and leadership potential to be successful as an L5 SDEII at Amazon.

System Design\*

Strength

Bluescape Link: https://client.ext.bluescape.ee-infra.aws.dev/UyXi7l74xwf0VvkF7tXY  
  
  
1. Initial Problem Statement: "Design a real-time collaborative document editing system similar to Google Docs"  
  
Candidate's Approach:  
- Proactively listed functional and non-functional requirements without being prompted  
Analysis: Shows strong initiative and systematic approach to problem-solving, aligning with SDEII expectations of independently identifying requirements.  
  
2. Question: "What consistency model would you use?"  
  
Candidate's Response:  
- Proposed eventual consistency  
- Provided justification for the choice  
Analysis: Strong technical decision-making with clear reasoning. Shows understanding of distributed systems concepts and their practical applications.  
  
3. Question: "Provide a basic High-Level Design"  
  
Candidate's Response:  
- Included core components:  
\* API Gateway  
\* Services layer  
\* Database layer  
\* Caching layer  
Analysis: Demonstrates solid understanding of distributed system architecture and key components needed for a scalable solution.  
  
4. Question: "How would you handle real-time collaboration?"  
  
Candidate's Response:  
- Proposed WebSocket implementation  
- Discussed alternatives:  
\* Long polling (mentioned latency issues)  
\* Server-Side Events  
- Provided tradeoff analysis  
Analysis: Shows depth of knowledge in real-time communication protocols and ability to evaluate technical choices based on their tradeoffs.  
  
5. Question: "How would you resolve conflicts?"  
  
Candidate's Response:  
Multiple approaches discussed:  
- Queue mechanism  
- Timestamp-based approach  
- Operational Transformation (OT)  
- Locking mechanism using queues  
Analysis: Demonstrates advanced technical knowledge and ability to provide multiple solutions with different tradeoffs.  
  
6. Question: "How to ensure latency and availability?"  
  
Candidate's Response:  
- Proposed caching solution  
Analysis: Shows consideration for performance optimization, though could have potentially expanded on other availability strategies.  
  
7. Question: "How would you handle scaling?"  
  
Candidate's Response:  
- Document ID-based partitioning  
- Database sharding by document ID  
Analysis: Demonstrates understanding of horizontal scaling concepts and practical implementation approaches.  
  
Overall Rating: STRENGTH  
  
Reasoning:  
1. Requirements Understanding (Strength)  
- Proactively identified requirements  
- Showed clear understanding of problem space  
  
2. Technical Design (Strength)  
- Comprehensive architecture  
- Multiple approaches to complex problems  
- Clear understanding of distributed systems concepts  
  
3. Scalability Considerations (Strength)  
- Clear partitioning strategy  
- Database sharding approach  
- Caching implementation  
  
4. Trade-off Analysis (Strength)  
- Well-reasoned choices between different technologies  
- Multiple approaches to conflict resolution  
- Clear justification for consistency model  
  
The candidate consistently demonstrated the behaviors listed in the "Strength" column of the rubric, showing both breadth and depth in their technical knowledge while maintaining practical considerations for implementation. Their responses aligned well with SDEII expectations of making appropriate technical decisions within broader business and technology strategy contexts.

Invent and Simplify\*

Mild Strength

Question 1: "Example of a complex problem solved with a simple solution"  
Response Analysis:  
- Problem: Lead management between buyers, publishers, and vendors  
- Original Complexity: Manual process, multiple handoffs, external vendor dependency  
- Solution: Template rendering system replacing LLM-generated HTML  
- Impact: $50K savings, 4-5 hours time reduction  
  
Evaluation:  
✓ Shows simplification (replacing complex process with templates)  
✓ Demonstrates cost and time savings  
△ Could have better explained exploration of alternatives  
△ Limited detail on implementation challenges  
  
Question 1.1: "What made the problem complex?"  
Response Analysis:  
- Answer was very brief: "Solution was complex"  
- Missing detailed explanation of complexity factors  
△ Weak response - doesn't effectively articulate problem complexity  
  
Question 1.2: "How do you know your solution addressed the problem?"  
Response Analysis:  
- Clear metrics: $50K savings  
- Time reduction: 4-5 hours  
- Eliminated external vendor dependency  
✓ Strong response with specific outcomes  
  
Question 2: "Time when you influenced new thinking and innovation"  
Response Analysis:  
- Problem: Cookie dataset management (900MM entries)  
- Innovation: Queue mechanism implementation  
- Technical solution: RabbitMQ, consumers, lambda  
△ Could have better explained innovation aspect  
△ Limited detail on team influence  
  
Question 2.1: "What was the impact?"  
Response Analysis:  
- Daily data updates achieved  
- Improvement from 90% to 91% acceptance rate  
△ Minimal metrics provided  
△ Could have included more comprehensive impact measures  
  
Overall Rating Justification (MILD STRENGTH):  
  
Strengths:  
1. Technical Problem Solving  
- Successfully simplified complex processes  
- Achieved measurable improvements  
- Demonstrated practical innovation  
  
2. Results Orientation  
- Clear metrics in first example  
- Tangible improvements in both cases  
- Focus on efficiency  
  
Areas for Development:  
1. Innovation Articulation  
- Could better explain innovative aspects  
- Limited detail on exploration process  
- Minimal information about team influence  
  
2. Impact Measurement  
- Second example lacks comprehensive metrics  
- Limited long-term impact discussion  
- Could provide broader business context  
  
3. External Awareness  
- Solutions primarily internally focused  
- Limited mention of external research/benchmarking  
- Could show more industry awareness  
  
  
The MILD STRENGTH rating reflects strong technical execution but room for growth in broader innovation thinking and influence. The candidate shows good problem-solving abilities but could develop in areas of external awareness and comprehensive solution development.

Hide interview notes (295 words)

Interview AShwath :   
  
  
Candidate Experience:   
5.5 years exp   
B2B SAS company   
Display ads , buy and sell leads via platform   
Help business target  
Rferactor version of display ads , deign , implementation.   
Csharp , dot net core , AWS cloud service   
Logistics Company   
Mobile app - ledger for truck owners   
payment handling   
E2E ecommerce.   
Athena health   
SAS company - tech solutions for hospitals - repittive works.   
  
  
1-Give me an example of a complex problem you solved with a simple solution.   
  
Problem : buyer need leads for the compnay to sell , publisher get leads -  
Step : Mock POC - before sending details to publisher we verify with the buyer. phone , checkbox etc. This was done as a manual process. done by external vendor. buyer to vendor to publisher loop caused manual work and time cost.   
  
  
Solution : eliminate the middle men. Generate the html   
  
using LLMS - gen html - send to campaign managers . lot of complexities. html which llm generates will not be the same.   
  
Simple template rendering. based on users input configure the template. defined template for diffrent use cases.   
  
1.1 What made the problem complex ?   
Solution was complex   
  
  
1.2 How do you know your solution addressed the problem?  
50k dollars was saved , inhouse template rendering.   
reduced the time 4-5 hrs.   
No dependency on external vendors.   
  
  
  
2-Describe a time when you influenced and drove new thinking and innovation out of your team.   
Cookie data set - to target company   
cookie is short lived   
refresh the data set.   
900MM entries.   
chat with external vednor 10k files /day   
build a solution to consume the files , read and update our db.  
batching solution already present. low memory - CPU.  
Queue mechanism - Rabbit MQ - consumers. every insert fetching , initial lambda.  
batch processing had time windows   
  
2.1 what was the impact ?   
Daily update to data . 90% acceptance rate - 91%

Submitted Feb 21, 2025 at 3:03PM

–

Anaiappan Govindan ([anaiapg](https://phonetool.amazon.com/users/anaiapg" \t "_blank)) Software Development Manager (Level 6)

Tenure: 13 years, 7 months | Training: MGHD, SDE FI & OA VERIFIER | Interviews:

285

Ownership\*

Deliver Results\*

Dive Deep\*

inclined

Summary

He was able to articulate his work right set of tech details but sometime we need probe him more to get the detailed explained that helps to understand the system/changes fully. He built and delivered the set of improvements based on the given required and helped the team members on the need basis as well. Also, he voluntarily proposed and built a POC for product improvements as based on the pain points observed in the current system

Ownership\*

Mild Strength

He explained about another situation where he built a POC for Operator Productivity improvement for a tool that was owned by a different team in his org and shown demo to leadership team about the benefits of this POC and later it got converted into a committed Projects.   
  
He explained about a situation where his peer struggling to rootcause an issue. He observed from the scrum update and guided him to narrow down the rootcause of that issue and to fix and release it .

Deliver Results\*

Strength

He explained about a project where his manager asked to build a POC for displaying ADs based on domian configuration. He mentioned that first he cleared all the ambiguities and defined the initial boundary for his POC based on the inputs from his manager/product and built a quick POC for planned MVP and demoed his leadership team within a quick timeframe. He also mentioned the list of roadblocks that he faced while Productionize it mainly w.r.t scaling and then he enhanced with synchronous to asynchronous and enabled batch processing for parallel execution to handle the load.

Dive Deep\*

Mild Strength

He was able to articulate the tech design details(with Producer and consumer model for parsing and persisting data.) for the new the POC that he build in Ads display domain and also how he mitigated the learnings he faced w.r.t handling scaling related issue. He mentioned that he didn't pick lambda/event notification based mechanism instead of building producer/consumer model as he wants to build a common solution that works in non AWS environment as well.

Submitted Feb 26, 2025 at 3:22AM

–

Pradeep Sivakumar ([psiva](https://phonetool.amazon.com/users/psiva" \t "_blank))BR Sr Manager, Software Dev (Level 7)

Tenure: 13 years, 5 months | Training: MGHD, SDE FI, TL FI & OA VERIFIER | Interviews:

853

Ownership\*

Deliver Results\*

Dive Deep\*

inclined

Summary

This loop is a part of the Bar Raiser In Training (BRIT) program. Shivam, who is currently training to become a Bar Raiser, conducted this interview independently. For detailed feedback about the candidate's performance, please refer to Shivam's interview notes.

Ownership\*

Mild Strength

This loop is a part of the Bar Raiser In Training (BRIT) program. Shivam, who is currently training to become a Bar Raiser, conducted this interview independently. For detailed feedback about the candidate's performance, please refer to Shivam's interview notes.

Deliver Results\*

Mild Strength

This loop is a part of the Bar Raiser In Training (BRIT) program. Shivam, who is currently training to become a Bar Raiser, conducted this interview independently. For detailed feedback about the candidate's performance, please refer to Shivam's interview notes.

Dive Deep\*

Mild Strength

This loop is a part of the Bar Raiser In Training (BRIT) program. Shivam, who is currently training to become a Bar Raiser, conducted this interview independently. For detailed feedback about the candidate's performance, please refer to Shivam's interview notes.

Submitted Feb 25, 2025 at 10:00AM

–

Shivam Kohli ([shikohli](https://phonetool.amazon.com/users/shikohli" \t "_blank))BRIT SDE 2 (Level 5)

Tenure: 5 years | Training: MGHD, SDE FI & OA VERIFIER | Interviews:

70

Ownership\*

Deliver Results\*

Dive Deep\*

inclined

Summary

Overall I'm inclined.

I will look for feedback from other design and coding round. Candidate was overall good in communication but went to much in details and the listener had to question a lot. When presented with an easy-to-medium complexity coding question to assess problem-solving skills and coding knowledge, the candidate showed the ability to think through the problem. However, upon deeper probing into low-level details, such as the rationale for sorting the array, the candidate's explanations were not entirely convincing.

Ownership\*

Mild Strength

Tell me about a time when you took on something significant outside your area of responsibility. Why was it important? What was the outcome?

The candidate demonstrated initiative by addressing a critical financial reporting issues. The metrics data for October through December, which directly impacted customer billing and campaign manager compensation. With the data team unavailable during the holiday period, the candidate voluntarily stepped in to investigate and resolve the problems. he took ownership of analyzing the codebase and successfully identified three major issues: count mismatches, currency mismatches, and failed currency conversions. The candidate was able to root cause. He fixed the initial problem and provided the root cause for currency problem to the data team to fix when they were back

Deliver Results\*

Mild Strength

Give me an example of a time when you were able to deliver an important project under a tight deadline. What sacrifices did you have to make to meet the deadline? How did they impact the final deliverable? What was the final outcome?

The candidate described managing a time-sensitive project involving targeted advertising based on cookie IDs for Google employees. The system initially showed promising results with high impression rates, but performance significantly declined due to expired cookies. When this critical issue was identified, the candidate was given a one-month deadline to implement a solution that would update cookie IDs and remove stale data from their database. Their solution involved processing customer-provided data dumps (approximately 10,000 files) stored in S3. The candidate developed a batch script to read these files and publish the data to RabbitMQ topics for processing. When questioned about why they didn't utilize S3 events for this purpose, the candidate explained that the team wanted to maintain cloud-agnostic architecture. The candidate successfully completed the first phase of the project within the deadline, implementing necessary logic to handle duplicate data and new entries.

Dive Deep\*

Mild Strength

Walk me through a big problem or issue in your organization that you helped to solve. How did you become aware of it? What information did you gather? What information was missing and how did you fill the gaps? Did you do a reflection at the conclusion of the project? If so, what did you learn?

The candidate described working on a display ads system redesign project where scalability was a major concern. he specifically focused on transitioning a web analytics tool that involved script injection in client environments to track user behavior and page interactions. The main challenge was to port existing functionality to a new system while ensuring zero customer intervention and preventing data loss during the live transition. The candidate's solution involved implementing URL-based logic to automatically determine which script version to use, eliminating the need for customers to manually update their implementations. For handling data during the transition, they implemented a dual database approach using PostgreSQL and MongoDB, with date range markers to track new data flow, and integrated both databases with Snowflake for analytics visualization. However, when questioned about the technical decision to use two databases, the candidate struggled to provide a strong justification, eventually admitting it was just a temporary migration solution with plans to deprecate one database after the transition. While the candidate demonstrated ability to handle backward compatibility and zero-downtime migration, their inability to defend key technical was not promising

Hide interview notes (165 words)

Koko loves to eat bananas. There are n piles of bananas, the ith pile has piles[i] bananas. The guards have gone and will come back in h hours. Koko can decide her bananas-per-hour eating speed of k. Each hour, she chooses some pile of bananas and eats k bananas from that pile. If the pile has less than k bananas, she eats all of them instead and will not eat any more bananas during this hour. Koko likes to eat slowly but still wants to finish eating all the bananas before the guards return. Return the minimum integer k such that she can eat all the bananas within h hours.

Input: piles = [11, 3,6,7], h = 8 Output: 4

def minEatingSpeed(piles, h):

def hours\_needed(speed):

return sum(ceil(pile/speed)) for pile in piles

low, high = 1, max(piles)

piles.sorted()

while low < high:

mid = (low + high) //2

if hours\_needed(mid) <= h:

high = mid

else:

low = mid + 1

return low

Submitted Feb 25, 2025 at 1:08AM

–

Kumaran Rajendiran ([rkumaran](https://phonetool.amazon.com/users/rkumaran" \t "_blank)) Software Development Engineer (Level 5)

Tenure: 4 years, 8 months | Training: MGHD & SDE FI | Interviews:

11

LLD\*

inclined

Summary

The candidate is technically capable. They're good at learning and fixing things when guided. When they got hints, they understood quickly and made good use of them. The candidate would be okay to hire but will need coaching intitally.

LLD\*

Mixed

Strengths:  
  
Good design decisions (interface vs abstract class)  
Created extensible and maintainable code  
Good at incorporating feedback  
Clear technical communication  
  
Areas for Development:  
Required a few hints for filter pattern implementation  
Needed guidance for combining filters  
Initial requirements analysis could be stronger

Hide interview notes (299 words)

/\*  
Implement Linux find command as an API. The API will support finding files that:  
  
Files that have a given size requirement.( ex: all files > 5 mb size)  
Files with a certain naming pattern. (ex: .pdf, .xml)  
  
Assume file class  
{  
getname()  
listFiles()  
isDirectory()  
}  
\*/  
  
'''  
Recap on requirments  
1. retrive based on naming pattern ( can be multiple patterns )  
2. recursive search within directories  
3. Filter based on size  
4. both saearch and filter can be applied together   
5. ex - Find - find xml and pdf greater than 5mb  
  
  
-- Entities  
File   
Directory  
Filters  
  
'''  
  
  
  
  
IFileFilter:  
def matches(File) -> bool  
  
ExtensionFilter(IFileFilter):  
def \_\_init\_\_(self, extension):  
self.extension = extension  
def matches(file):  
reutrn file.getExt() == self.extension  
  
SizeFilter(IFileFilter):  
def \_\_init\_\_(self, size, comparator):  
self.size = size  
self.comparator= comparator  
  
def matches(file):  
match comparator:  
case comparatorEnum.Equals:  
return file.getSize() == size  
case comparatorEnum.GreaterThan:  
return file.GetSize > size  
  
  
ANDFilter(IFileFilter):  
def \_\_init\_\_(self, ...filters)  
def matches(file):  
return all(f.matches(file) for f in self.filters) if self.filter else True  
  
OrFilter(IFileFilter):  
def \_\_init\_\_(self, ...filters)  
def matches(file):  
return any(f.matches(file) for f in self.filters) if self.filter else True  
  
class FindService:  
  
def find(self, directory, file\_filter):  
result = []  
  
for file in directory.listFile():  
  
  
  
- The candidate Asked questions on file patterns  
- The candidate spoke about querying across subdirectories  
- The Candidate spoke about recursive implementation on directories  
- Asked the candidate about the other requirements  
- The interviewer gave a hint on combination of filters  
- The candidate found that he should implement filter patterns   
- The interviewer asked why interface and not an abstract class - The candidate justified the usage of interface by differentiating between interfaces and abstract class  
- The candidate wrote code that was extensible. The candidate implemented filters based on combinations with some hints  
- Request interface defined by the candidate was extensible.