Job Description

● Build a high-performance full-stack team. Responsible for the introduction of department personnel and the performance evaluation and management of department personnel;

●Responsible for managing the core technology of software, undertaking the analysis and evaluation of software solutions, technical direction control and outline design, and ensuring the realization of relevant requirements and technical indicators

●Responsible for organizing and guiding the product technical team to carry out product design, system research and development, and pre-launch testing under the overall architecture, and coordinate project development

or all aspects of the implementation to grasp the overall progress of the project

● Deliver new features, fix bugs, and improve existing codebases

● Write high-quality, maintainable code, review other people's code, and give suggestions for changes

● Mentor team members and model best coding practices

● Lead the team to design the system architecture

● Maintain a keen perception of new technologies and industry trends

● Bachelor's degree in software engineering, computer engineering, or other related majors

● More than 8 years of experience in software development

● Proficiency in understanding the following:

○ Microservices Architecture

○ Modern programming languages, such as Golang+ node js

○ Modern databases and storage systems

○ Containerized deployment using docker and Kubernetes

○ Experience with cloud environments – AWS or GCP

○ Code versioning tools, such as Git

○ Knowledge of GraphQL and REST APIs

● Extensive knowledge of designing and developing scalable and reliable systems

● Excellent learning ability, efficient and effective execution, strong

Self-driven consciousness and rich experience in developing project management to the process

and results-oriented professional attitude.

● Fluent in reading and writing in English, listening and speaking are required to be able to complete basic communication in English

David’s Concerns

Firstly, several key questions or concerns

1. 【Question】What percentage of the current system development (including B2B and other related business systems) and team building (single choice)?
   1. 0-20% In the initial stage, the team is completely new and there are no available features online
   2. In the mid-20-70% stage, the minimum viable team (MVT) is formed, and the number of people will be increased in the later stage, but the team leader leaves or is transferred, there are no promotion candidates within the team, some features of the project are launched, and there are still important features to be developed and launched.
   3. In the long-term stage of 70%-100, the team size is stable, the basic functional team (product, development, testing) is formed, and the team leader leaves and is transferred
2. 【Reply to Matching Degree】About the compatibility between my experience and JD

Note: Matching degree (%) learning cost (\* indicates difficulty, up to 5\*).

1. Microservices Architecture
   1. Matching: 60-80%.
   2. Cost of Study: \*\*
   3. Explanation: I am familiar with the microservice framework of Java Spring Cloud, and I also have a certain self-development ability for self-built microservice architecture using the following conceptual design, which is more complex and inconvenient in configuration maintenance and server maintenance than Spring Cloud. However, maintenance can be done through a series of standardized CICD methods. The following concept design is:
      1. Separation of data and business (for the sake of code maintenance, frequency of business iteration, and data BI).
      2. Separation of business modules (for business isolation and system maintenance)
      3. Front-end separation (for the sake of program development workflow and user experience).
      4. Separate storage of dynamic and static data (for system read/write computing performance)
      5. Read/write splitting (for business system application scenarios)
      6. Separation of storage and computing between service and data reporting models (for system performance considerations).
      7. Task scheduling, message synchronization queues, and multi-level and multi-range caching scenarios (for system performance, timeliness of business scenarios, and performance of business services).
      8. Standardization of data and business services (for the sake of system maintenance management and document standardization management)
2. Modern programming languages such as Golang+ node js
   1. Match:
      1. Golang (C-like, Python-like, weak object-oriented, thread-oriented, fast response with high concurrency between servers): 70%.
      2. Nodejs (ECMAScript, front-end application, focus on interactive experience, lightweight application): 80%.
   2. Cost of study
      1. Golang : \*\*\*
      2. Nodejs: \*
   3. Explanation:
      1. Golang: I haven't done go before, but I'm familiar with Python, and C is even more basic. HOWEVER, C# AND JAVA ARE DEVELOPED A LOT. It may be necessary to pay attention to the execution of monolithic scripts and dilute the object-oriented programming thinking in program development. It's kind of similar to the JS execution logic on the browser side. Itwill take some time to figure out the best practices for the features and development logic of the Glolang language. It's better if your company's existing team has best practices.
      2. Nodejs: There is basically no learning cost, familiar with Vue, React development patterns, familiar with Ant-design, if there are other frameworks, you need to learn now.

1. Modern databases and storage systems
   1. Matching: 70%-90%.
   2. Examples of databases in possible scenarios
      1. The data database in the business operation scenario uses high-performance relational databases and database clusters
2. If you have the conditions to use a monolithic MS SQL server or Oracle , the query bottleneck is low.
3. The limited conditions of using MySQL and its clusters have high query bottlenecks, and architecture design is required to cope with complex business scenarios.
   * 1. Large file object storage
4. If you need to store and quickly link some large file entities for object databases, and it is not convenient to use cloud disks, you can use OSS and purchase OSS services
5. Server hard disk storage. Considering the self-built storage and access services of cloud disk VMs, you need to maintain them by yourself, which has medium maintenance costs
   * 1. Non-institutional data storage is mainly suitable for querying, aggregating, and batch transmission of a large amount of business data. The big data ecosystem Hadoop (Hbase HDFS Hive) \Spark can be used for storage, which is also convenient for the practice of data science.
     2. Cache storage of highly reusable Non-SQL data
6. Redis high-performance clusters are used to implement server-level caching. Facilitate business data caching with frequent access, high repetition, and minimal real-time changes.
7. The Localcached mechanism is used to cache cold data at the client level.
   * 1. Offline log data is transferred and stored locally by using Kafka offline data queues to Hadoop clusters or native application servers.
8. docker and KB8
   1. Match: 40-70%.
   2. Cost of Study:
   3. Explanation: Adopted in the past in a small number of customer projects, docker deploys multiple applications. K8s has understood, understood and practiced container multi-region dynamic scheduling management, but has not been operated in real projects. But learning to manage K8s is not a difficult task.
9. Cloud experience:
   1. Match: 70%.
   2. Cost of study:\*
   3. Explanation: I have been in contact with many cloud platforms and products in China, such as Azure, Aliyun, and HuaweiYun He has certain conceptual and practical ability to build Linux virtual machines, use SQL servers, server network formation, and server load balancing. In addition, I also have some understanding of Google's cloud function, and it is not difficult to learn.
10. Code versioning git
    1. Match: 100%.
    2. Cost of study: 0
    3. Explanation: Familiar with the git workflow, master develop feature test bug workflow, be able to design the CICD process of the git project according to the team's CICD model, and use Jenkins Automated deployment
11. GraphQL and Rest API
    1. Match: 100 %.
    2. Cost of study:\*
    3. Explanation: GraphQL is a structured JSON that facilitates data interaction and is easy to get started. The Rest API is an API design specification that is easy to use, and the whole team uses Postman for standardized development and documentation in production development engineering.
12. Scalable reliability system developed
    1. Through the above system design concepts, we have the ability to develop scalable systems
    2. The reliability system includes the following aspects
       1. Business level:
13. Business documents are standardized and iterative
14. The business logic is clear, minimized, simplified, and executable
    * 1. Code level:
15. Separation of query code and function code
16. The code of large modules is structured, named and standardized, and the code is readable.
    * 1. System integration
17. There is strong code decoupling between modules
18. Easy to split, systematic
19. API integration, standardized data flow, data storage
    * 1. Data level
20. The data structure is designed with business logic, system performance, and development simplicity in mind
21. Product design
    1. It has a certain design concept of interactive and easy to use, standardized and beautiful
    2. Need to find a professional product manager with business level and art design skills
22. Process-results double-oriented
    1. Process
       1. Clarify the development pipeline process
       2. Simplify the standardization logic and reduce personalized development
       3. The documentation is refined, which is convenient for long-term maintenance of the system and handover work when the team is updated.
       4. The process of system construction and software testing is automated and simplified, and the necessary work is not reduced while achieving automation and rapid response to development needs, and Ag ile is used for project development.
       5. The team management adopts a 3:7 model, with 7 points of experience, and when time permits, each team member carries out 3 points of new ability and new skills expansion.
    2. outcome
       1. Deliver products with high satisfaction within the effective time, and ensure that there are no frequent small bugs and major common scenario bugs when the product is launched
       2. Output high-quality documents and business code that are easy to iteratively maintain
23. English
    1. Listening, speaking, reading and writing are fluent, and telephone and email communication is almost barrier-free
    2. Technical questions can be communicated in English
    3. Important technical and business meetings can be held throughout the day
    4. Proactively aligns important issues and meeting brief
24. Major
    1. Bachelor with a mathematics major
    2. Candidates who graduate with mathematic majors have much more thinking advantages at the level of data (big data), algorithms, and system architecture than those who are computer science majors.
    3. When In college, having passed Cisco network technology experts, have a unique understanding of network communication solutions and strong trouble shoot capabilities, have a strong sense of the features of cloud products. I also have a github project

2. Career goals

I have been deeply engaged in the software and big data industry, accumulated experience in the development of software and data systems for various types of software and data application scenarios that are suitable and cutting-edge, and hope to focus on the field of big data and enterprise digital transformation in the future.

3. Hope salary (headhunters choose one after looking)

In the first question about the current situation of the team, ABC is given different salaries and salary expectations, salary appreciation, willingness to join, team quota, and the probationary period grace that the other party wants to give.

* 1. Option A
     1. Expecting Annual Pay :550k+ (100k floating above) RMB
     2. Salary structure: Base 450k + yearly bonus + KPI bonus
     3. Salary appreciation: Based on the basic salary once every 1-2 years, the salary increase is not less than 10%. If the frequency of salary adjustment is not met, the base amount of performance calculation or the additional increase of the year-end bonus for a half-year or one-year cycle can be accepted. Punitive calculations in performance are accepted, but must not be less than 30% of the performance base. The year-end bonus does not accept punitive adjustments and does not accept benefit adjustments.
     4. Willingness to join: 100% Reason: I'm good at building a team, not an old team revolution
     5. Team quota: MVT :
        1. A product manager with 5-8 years of experience (25k-35k)
           1. Considerable proficiency in prototyping software such as Axure, Sketch, Ink Knife, PS, AI
           2. Experience with at least 60% of the industry's products or platforms that match the company's software is required
           3. Have a medium or higher art foundation, and be able to lead artists and demand engineers in the future.
           4. Professional level and speed of rapid delivery of requirements documents and prototype design, and understand some data design theories
           5. Familiar with the Agile team model, familiar with Conflence Page, Jira Trello and other tool platforms
           6. From the company, team or brand level, the brand can be used to think, design, and dock the development of the design document output.
        2. A 3-5 year front-end development Lead (15k-30k).
           1. Able to develop with mature Nodejs frameworks such as react, vue, ant-design
           2. Able to control the UI, color matching, and operation permissions of the system from the data structure level, and design the underlying logic of the configuration file
           3. Able to transform the underlying logic of the excellent No de JS framework.
           4. Ability to participate in design discussions with product teams at the technical framework level.
           5. Able to accurately implement the product manager's design of interaction and color schemes.
           6. Ability to adapt to git version control and jienkins development and CI/CD processes
           7. Able to use webpack to package and publish applications, and deploy them using docker or k8s
           8. In the future, he will be able to lead a small team to carry out standardized process development
        3. A 4-7 years exp backend development Lead (25k-40k)
           1. Basic business data structure design and development capabilities
           2. Possess basic algorithm research and development capabilities
           3. Moderately difficult system understanding and tuning skills
           4. Able to understand data flows in SQL optimization, anti-injection, transaction management, and multi-threaded scenarios
           5. Able to develop modular structured code
           6. Able to tune system performance
           7. Able to package and publish code
        4. A 4-year+ exp test Lead (15k - 25k).
           1. Basic testing concepts (unit testing, white box testing, black box testing, bubbling testing, integration testing, stress testing).
           2. Possess the professional level of writing test cases and test reports
           3. It has certain automated test script development and process integrated application capabilities (Python).
           4. Scenario-based clustered test design and event capabilities (e.g., Jmeter cluster testing, asminefor test development).
           5. Ability to track defects (git and agile)
           6. Ability to lead teams to assign testing tasks
        5. Architect (myself, dedicated to the post-production team, for system operation and maintenance optimization, development supervision).
           1. Have the ability to design the system
           2. Have the ability to design program structure
           3. It has the ability to optimize servers and networks, and troubleshoot
           4. Possess standardized management capabilities
           5. It has the design and architecture capabilities of highly integrated and data business separation.
  2. Probationary period payment limit
     1. Contract period: 5 years
     2. Probationary period: 6 months
     3. Probationary salary: 90% discount is accepted for the part starting from 550k
     4. Work plan:

1. 2-3 months of systematic tutoring of language and framework, complete the construction of MVT, long-term design and construction of the system platform, and the development and launch of some simple functions. It would be better if you had more time
2. In 3-6 months, the team's basic development process is standardized, the delivery is standardized, and some medium-type functions are output in an orderly manner.
3. 6 months - long-term, enter the normal development, orderly arrangement of complex systems and functions of the research and development work. Ensure satisfaction and comfort of delivery.
   * 1. If the above conditions cannot be met, or if the above events are delayed or undeliverable due to personnel or resources, your company cannot tolerate it.
     2. In the first year, team building and tasks are the most important, and it is hoped that additional team rewards or project rewards can be allocated
   1. Option B
      1. Expecting Annual Pay:700k+ (50k floating above)
      2. Salary structure: Base 550k + yearly bonus + KPI bonus
      3. Salary appreciation: Based on the basic salary once every 2 years, the salary increase is not less than 5%. If the frequency of salary adjustment is not met, the base amount of performance calculation or the additional increase of the year-end bonus for a half-year or one-year cycle can be accepted. Punitive calculations in performance are accepted, but must not be less than 30% of the performance base. The year-end bonus does not accept punitive adjustments and does not accept benefit adjustments.
      4. Willingness to join: 65%, reason: the team has not yet formed a large amount of accumulation in the early development, documentation, code specification, and system design, and there is still room for easy transformation and re-specification. The amount of renovation work is medium.
      5. Team Allocation: Based on the MVT achieved in Option A, Junior players will be added to each functional team according to the severity of the task
      6. Probationary period payment limit
         1. Contract period: 5 years
         2. Probationary period: 6 months
         3. Probationary Pay：Based on Annual Pay 700k，Percentage down not accepted
         4. Work plan:
4. 3-5 months of systematic tutoring of language and framework, run-in and systematic improvement training with the original MVT, thoroughly understand the design, construction and functional code of the original system platform, and continue the development and launch of the current functions. However, it is necessary to slow down the development pace to 1.3 times - 1.5 times the original development rhythm, and it is recommended to standardize the basic development process, standardize the delivery, and output some medium-type functions in an orderly manner.
5. 5 months - The long-term team enters the regular development, and the R&D work of complex systems and functions is arranged in an orderly manner. Ensure delivery satisfaction, comfort, and orderly expansion of junior players.
   * + 1. If the above conditions cannot be met, or if the above events are delayed or undeliverable due to personnel or resources.if your company cannot tolerate it, then leave it.
   1. Option C
      1. Expecting Annual Pay:650k~1000k+ (200k floating above)
      2. Salary structure: Base 550k + yearly bonus + KPI bonus + options (regular dividends) + significant subsidies (for the sake of China's tax system.) I hope that your company can make a reasonable individual income tax plan for this, and reduce my individual income tax to less than 25% or even lower).
      3. Salary appreciation: Based on the basic salary of 3 years or more salary adjustment, the rest can be determined on a case-by-case basis
      4. Willingness to join: 35%-60%, reason: the team has developed in the regular period, and a large amount of documentation, code specifications, and system design have been formed, which has caused difficulties in transforming and re-specifying the team. The amount of work involved in the renovation is enormous. Of course, if your company's original team is very standardized. (My compensation part could be flexible).
      5. Team Allocation: Based on the MVT achieved in Option A, Junior players will be added to each functional team according to the severity of the task
      6. Probationary period payment limit
         1. Contract period: 5 years
         2. Probationary period: 6 months
         3. Probationary salary: The annual salary will be determined according to the negotiation between the two parties on the complexity of the work, and discounts will not be accepted
         4. Work plan:
            1. 8 months of systematic tutoring of language and framework, thoroughly understand the design, construction and functional code of the original system platform, deliver and develop according to the original rhythm, and optimize and reform not for or only for a small number of easy-to-fix problems. We will not carry out large-scale reforms at the bottom.
            2. The long-term team enters the regular development, and the research and development of complex systems and functions is arranged in an orderly manner. Ensure satisfaction and comfort in delivery. If there is a huge problem with the original underlying design that needs to be overturned, then I will give feedback upwards and the leadership will determine whether there is a need for major reforms, of course, there are some problems that I am very, very painful, but the leadership cannot give enough grace time and resources to make changes, I regret this time, it is clear that this is a career choice that is not very suitable for me. My advice is that I hope your company gives me enough information about your company's existing team during the interview stage. So as to make a comprehensive judgment.
         5. If the above conditions cannot be met, or if the above events are delayed or undeliverable due to personnel or resources. if your company cannot tolerate it, then leave it.

职位描述

●建立一支高性能的全栈团队。负责部门人员的引进及本部门人员的绩效考评管理工作;

●负责管理软件核心技术,承担软件方案分析评估、技术方向把控和概要设计,确保相关需求和技术指标实现

●负责组织并指导产品技术团队在整体架构下开展产品设计、系统研发、上线前的测试工作,协调项目开发

或实施的各个环节,把握项目的整体进度

● 交付新功能,修复缺陷,改进现有代码库

● 编写高质量的、可维护的代码, Review他人代码,并给出修改意⻅

● 指导团队成员并为最佳编码实践建模

● 带领团队设计系统架构

● 对新技术、行业动向保持敏锐的感知度

● 软件工程，计算机工程、或其他相关专业本科学历

● 8年以上软件开发工作经验

● 熟练理解以下内容:

○ Microservices架构

○ 现代编程语言,如Golang+ node js

○ 现代数据库和存储系统

○ 使用docker和Kubernetes的容器化部署

○ 有云环境经验—— AWS或GCP

○ 代码版本控制工具,如Git

○ 了解GraphQL和REST api

● 丰富的设计和开发可扩展和可靠系统的知识

● 具备优秀的学习能力，高效且有效的执行力，强烈

的自我驱动意识，丰富的开发项目管理经验，以过程

和结果双导向的职业态度。

● 英语读写流畅,听说方面要求能用英语完成基本沟通

 David Concerns

一、几个核心问题

1. 【问】目前的系统开发（包括B2B以及其他相关业务系统）、团队建设的完成度百分比（单选）？
2. 0-20%初期阶段，团队完全新建，没有可用功能上线
3. 20-70%中期阶段，最小可行性团队（MVT）形成，后期会增加人手，但团队 leader 离职或调任，团队内部不具备晋升候选人，项目部分功能上线，还有剩余重要功能待开发和上线。
4. 70%-100长期阶段，团队规模稳定，基本功能团队形成（产品，开发，测试）,团队Leader 离职调任
5. 【回复】关于我的经验和JD的匹配度

备注：匹配度（%）学习成本(\*表示难度,最高5个\*）

1. Microservices架构
   1. 匹配度：60-80%
   2. 学习成本：\*\*
   3. 解释：之前熟悉Java Spring Cloud 的微服务框架，对于使用以下概念设计进行自建微服务架构也有一定的自研能力，主要自研架构相对于Spring Cloud在配置维护和服务器维护上较为复杂和不方便。不过可以通过一系列标准 化CICD方法进行维护。以下概念设计为：
      1. 数据与业务分离 （出于代码维护和业务迭代的频率、数据BI考虑）
      2. 业务模块分离（出于业务隔离和系统维护考虑）
      3. 前后端分离 （出于程序开发工作流程、用户体验考虑）
      4. 动态数据与静态数据分离存储（出于系统读写计算性能考虑）
      5. 读写分离（出于业务系统应用场景考虑）
      6. 业务型与data reporting 型的存储计算分离（出于系统性能考虑）。
      7. 任务调度，消息同步队列，多级别多范围缓存场景（出于系统性能，和业务场景的时效性，以及业务服务的性能考虑）。
      8. 数据和业务服务标准化（出于系统维护管理，文档规范化管理考虑）
2. 现代编程语言,如Golang+ nodejs
   1. 匹配度 ：
      1. Golang （类C,类Python,弱面向对象、注重线程执行，服务器之间高并发快速响应) ：70%
      2. Nodejs (ECMAScript, 前端应用，注重交互体验，轻量级应用）：80%
   2. 学习成本
      1. Golang : \*\*\*
      2. Nodejs : \*
   3. 解释：
      1. Golang: 之前没有做过go,但对Python比较熟悉，C语言更是基础。不过开发C#和JAVA较多。可能在程序开发中需要注重单体脚本执行、淡化面向对象编程思维。有点类似与浏览器端的js执行逻辑。Golang 语言的特性和开发逻辑规划需要一段时间摸索出最佳实践。如果贵公司现有团队有最佳实践更好。
      2. Nodejs: 基本没有什么学习成本，熟悉Vue, React开发模式，熟悉Ant-design如果有其他的框架则需 要现学。

1. 现代数据库与存储系统
   1. 匹配度：70%-90%
   2. 可能的场景下对应的数据库案例
      1. 业务操作场景下的数据数据库，采用高性能的关系型数据库和数据库集群
         1. 有条件使用单体的MS SQL server 或者Oracle ，查询瓶颈较低。
         2. 条件有限采用Mysql及其集群，查询瓶颈较高，需要架构设计以应对 复杂的业务场景。
      2. 大文件对象存储
         1. 对象数据库，需要对某些 大文件实体进行存储和快速链接，不方便采用云硬盘的方式，则可以采用OSS,需要购买 OSS服务
         2. 服务器硬盘存储。考虑云盘虚拟机自建存储和访问服务，需要自行维护，有中等的维护成本
      3. 非机构化数据存储，主要适用于大量的业务数据查询、聚合、系统间批量传输需求。可以采用大数据生态Hadoop(Hbase HDFS Hive) \Spark 进行存储，也方便数据科学的实践。
      4. 高复用non-sql数据的缓存存储
         1. 采用Redis 高性能集群实现服务器层级的缓存。方便频繁访问、高重复、微量实时变更的业务数据缓存。
         2. 采用Localcache机制对用户冷数据进行客户端级别的缓存。
      5. 离线日志数据，采用Kafka离线数据队列向hadoop集群或者原生应用服务器本地进行数据log传输和存储。
2. docker和KB8
   1. 匹配度：40-70%
   2. 学习成本：\*\*\*
   3. 解释：过去在较少的客户项目中采用，docker 部署多例应用。K8s的容器多区域动态调度管理了解过，能够理解并且实践，但是并未在真正的项目中操作过。但是学习管理K8s并非难事。
3. 云环境经验：
   1. 匹配度：70%
   2. 学习成本：\*
   3. 解释：接触过国内很多的云平台和产品，比如Azure ,Aliyun, HuaweiYun. 对搭建linux虚拟机, 使用SQL服务器，服务器网络组建，服务器负载均衡具有一定的构思和实践能力。除此之外为google 的cloud function也有所了解，学习并不困难。
4. 代码版本控制 git
   1. 匹配度 ： 100%
   2. 学习成本：0
   3. 解释：熟悉git 工作流，master develop feature test bug的工作流，能够按照团队的 CICD模式设计git 项目的CICD流程，另外可以采用Jenkins的自动化部署
5. GraphQL和Rest API
   1. 匹配度：100 %
   2. 学习成本：\*
   3. 解释：GraphQL是一种方便数据交互的结构化JSON，很容易接受上手。Rest API 是API专业的设计规范很容易使用，在生产开发工程中将整个团队使用Postman 进行标准化开发和文档化。
6. 开发的可拓展可靠性系统
   1. 通过上述系统设计概念具备开发可拓展性的系统能力
   2. 可靠性系统包括以下几个方面
      1. 业务层面：
         1. 业务文档规范、可迭代
         2. 业务逻辑清晰最小化最简化可执行
      2. 代码层面：
         1. 查询代码、功能代码分离
         2. 大模块代码结构化、命名规范化、代码可读化。
      3. 系统集成
         1. 模块间代码解耦性强
         2. 易拆分系统性强
         3. API集成化，标准化数据流，数据存储
      4. 数据层面
         1. 数据结构设计具有业务逻辑性、系统性能和开发简易的考虑
7. 产品设计
   1. 具备一定的交互易用、规范、美观的设计理念
   2. 需要找到一位专业的具有业务底层和美工设计能力的产品经理
8. 过程结果双导向
   1. 过程
      1. 清晰化开发流水线过程
      2. 精简化标准化逻辑，减少个性化开发
      3. 文档精细化，便于系统的长期维护，和团队更新时的交接工作。
      4. 系统搭建，软件测试上线的流程自动化精简化，不减少必要工作的同时做到自动化快速响应开发需求，采用Agile 进行项目开发。
      5. 团队管理采用 3：7模式，用7分已用经验，在时间允许的情形下，每个团队成员进行3分的新能力新技能的拓展。
   2. 结果

1） 在有效时间内交付满意度较高的产品，保证产品上线没有频繁的小bug和重大常用的场景性bug

2） 输出高质量易于迭代维护的文档和业务代码

1. 英语
   1. 听说读写流畅，电话邮件交流几乎无障碍
   2. 技术问题英语可交流
   3. 可全天线上开重要技术和业务会议
   4. 会主动align 重要问题和会议纪要
2. 专业
   1. 数学专业出生
   2. 比软件专业出生的候选人在数据（大数据领域）层面，算法层面，和系统架构层面，更具备思考性优势。
   3. 在校即通过思科网络技术专家，对网络通信方案有全方位独到理解和强劲的trouble shoot能力。对云产品的特性有较强的感知力。自己有github项目，也会在家自己搭建系统网络。

二、职业目标

在软件和大数据行业深耕，积累适合的和前沿的各类型软件和数据应用场景的软件和数据系统开发经验，未来希望专注在大数据和企业数字化转型领域。

三、希望薪酬（猎头看后择一）

在上述第一个关于团队现状的问题中，针对ABC给出不同的薪酬以及薪酬期望、薪酬增值、加入意愿度、团队配额、希望对方给到的试用期宽限度。

1. 选项A
   1. 薪酬期望：55W+ (10W左右的浮动）
   2. 薪酬结构：Base 45W + 年终+ 平时绩效（平时绩效的平均值）总和
   3. 薪酬增值：基于底薪1-2年一次的调薪，调薪幅度不低于10%.

如果调薪频次达不到，则可以接受绩效计算的基数或者年终奖的半年或一年为周期的额外增幅。接受在绩效中的惩罚性计算，但不得低于绩效基数的30%。年终奖不接受惩罚性调整，不接受效益性调整。

1. 加入意愿度：100%，理由：我擅长搭建团队，而非老团队改造
2. 团队配额：MVT ：
   1. 一个5-8年经验的产品经理 (25k-35k)
      1. 需要对原型设计软件有相当熟练度，例如Axure,Sketch,墨刀, PS, AI.
      2. 需要对与公司软件有至少60%匹配的行业产品或平台经验
      3. 有中等以上的美工基础，未来能够带领美工和需求工程师。
      4. 对快速交付需求文档、原型设计有专业的水准和速度，懂得一些数据设计理论
      5. 熟悉Agile团队模式，熟悉Conflence Page, Jira Trello等工具平台
      6. 能够从公司或团队或品牌层面，对品牌进行色彩方案思考、设计、以及对接开发的设计文档性输出。
   2. 一个3-5年的前端开发Lead （15k-30k)
      1. 具有使用成熟的Nodejs 框架开发能力比如，react, vue,ant-design
      2. 能够从数据结构层面对系统的UI、配色、操作权限控制，进行配置文件的底层逻辑设计
      3. 能够改造优秀的node js框架的底层逻辑。
      4. 能够从技术框架层面参与产品团队的设计讨论。
      5. 能够精确的实现产品经理对交互和色彩方案的设计。
      6. 能够适应git版本控制和jienkins的开发和CICD流程
      7. 能够使用webpack打包发布应用，使用docker或者k8s部署
      8. 未来能够带领小团队进行标准化流程化开发
   3. 一个 4-7年的后端开发Lead (25k-40k)
      1. 具备基本的业务数据结构设计开发能力
      2. 具备基础的算法研究和开发能力
      3. 具备中等难度的系统理解和调优能力
      4. 具备SQL优化、防注入、事务管理、多线程场景的数据流理解能力
      5. 具备模块化结构化代码开发能力
      6. 具备系统性能调优能力
      7. 具备代码打包发布能力
   4. 一个4年以上的测试Lead( 15k - 25k）
      1. 具备基础的测试理念（单元测试，白盒测试，黑盒测试，冒泡测试、集成测试，压力测试）
      2. 具备撰写测试用例和测试报告的专业化水准
      3. 具备一定的自动化测试脚本开发和流程化集成化应用能力(Python)
      4. 具备场景化集群化测试设计和事件能力 （例如Jmeter 集群 测试、jasmine 面向测试开发）
      5. 具备缺陷跟踪管理能力(git 和Agile向)
      6. 具备带团队分配测试任务的能力

1. 架构师（我自己，后期团队中专人专做，针对系统运维优化，开发监督）
   1. 具备系统的设计能力
   2. 具备程序结构化设计能力
   3. 具备服务器，网络基本调优，troubleshoot能力
   4. 具备规范化管理能力
   5. 具备高度集成化，和数据业务分离的设计和架构能力。
2. 试用期款限度
   1. 合同期：5年
   2. 试用期：6个月
   3. 试用期薪资：55w起的部分，接受90%的打折
   4. 工作计划：
      1. 2-3个月时间语言和框架的系统性补习，完成MVT的建设，系统平台长期性设计、搭建，部分简单功能的开发上线。如果时间宽裕更好.
      2. 3-6个月团队基本的开发流程标准化，交付标准化，有序输出一些中等类型功能上线 。
      3. 6个月-长期，进入常态化开发，有序安排复杂系统和功能的研发工作。确保交付的满意度、舒适度。
   5. 如果不能满足上述条件，或者上述因为人员或资源造成延期和不可交付的事件，贵公司不能宽容则敬谢。
   6. 首年对于团队建设和任务在之重，希望能够额外分配团队奖励或者项目奖励
3. 选 项B
   1. 薪酬期望：70W+ (5W左右的浮动）
   2. 薪酬结构：Base 55W + 年终+ 平时绩效（平时绩效的平均值）总和
   3. 薪酬增值：基于底薪2年一次的调薪，调薪幅度不低于5%.

如果调薪频次达不到，则可以接受绩效计算的基数或者年终奖的半年或一年为周期的额外增幅。接受在绩效中的惩罚性计算，但不得低于绩效基数的30%。年终奖不接受惩罚性调整，不接受效益性调整。

1. 加入意愿度：65%，理由：团队在早期发展，文档，代码规范，系统设计，还没有形成大量累积，还有易于改造和重新规范的余地。改造工作量中等。
2. 团队配额：在选项A达到的MVT基础上根据任务的繁重程度给各个功能团队增加Junior 选手
3. 试用期款限度
   1. 合同期：5年
   2. 试用期：6个月
   3. 试用期薪资：Annual Pay70W折算，不接受打折
   4. 工作计划：
      1. 3-5个月时间语言和框架的系统性补习，与原有MVT的磨合和系统性改进培训，吃透原有系统平台设计、搭建和功能代码，继续当前的功能的开发上线。但必须放缓开发节奏至原有开发节奏的 1.3倍 - 1.5倍时间，推荐基本的开发流程标准化，交付标准化，有序输出一些中等类型功能上线 。
      2. 5个月-长期团队进入常态化开发，有序安排复杂系统和功能的研发工作。确保交付的满意度、舒适度，同时有序扩充junior选手。
   5. 如果不能满足上述条件，或者上述因为人员或资源造成延期和不可交付的事件，贵公司不能宽容则敬谢。
4. 选项C
   1. 薪酬期望：65W~100W+ (20W左右的向上浮动）
   2. 薪酬结构：Base 55W + 年终 + 团队期权（定期分红） + 重大补贴 + 平时绩效（平时绩效的平均值）总和 （出于对中国税制考虑，希望贵公司能够对此进行合理的个税规划，将本人的个税降至25%以内甚至更低）
   3. 薪酬增值：基于底薪3年或更久一次的调薪，其余可按具体情况定
   4. 加入意愿度：35%-60%，理由：团队在常规期发展，文档，代码规范，系统设计，已经形成大量累积，对团队造成改造和重新规范的困难。改造工作量巨大。当然如果贵公司的原先团队很规范化作业。（我的薪酬部分弹性很大）
   5. 团队配额：在选项A达到的MVT基础上根据任务的繁重程度给各个功能团队增加Junior 选手
   6. 试用期款限度
      1. 合同期：5年
      2. 试用期：6个月
      3. 试用期薪资：按照双方对工作复杂度的商议对年薪进行确定，不接受打折
      4. 工作计划：
         1. 8个月时间语言和框架的系统性补习，吃透原有系统平台设计、搭建和功能代码，按照原有的 节奏进行交付开发，不针对或者仅针对少量易修正问题进行优化改革。不进行底层大动作改革。
         2. 长期团队进入常态化开发，有序安排复杂系统和功能的研发工作。确保交付的满意度、舒适。如果出现原本底层设计的巨大问题，需要推翻，则我会向上反馈以及由领导层确定是否需要进行重大改革，当然存在某些问题是我非常很痛苦，但领导层无法给予足够 宽限时间和资源进行改革，我对次表示遗憾，很显然这是一次不是很适合我的职业选择。我的建议是，希望贵公司在面试阶段给到我足够的对贵公司现有团队的了解信息。从而进行综合判断。
      5. 如果不能满足上述条件，或者上述因为人员或资源造成延期和不可交付的事件，贵公司不能宽容则敬谢。