尊敬的面试官：

您好，接下来由我开始我的自我介绍。

我是中国海洋大学25届硕士研究生靳豪杰，来自河南焦作，很高兴参加今天的面试。我本科就读于河南科技大学，顺利保研至中国海洋大学。在校期间曾担任院学生会主席团成员、学生会学习部副部长，帮助老师组织各类活动，如组织学院参加校运动会等，得到老师和同学们的一致认可。在校期间连续三年荣获国家励志奖学金，蓝桥杯省赛二等奖，CCPC省级银奖，河南科技大学学习标兵、优秀毕业生等诸多荣誉称号，同时，以获得软件设计师资格证书。此外我也有一些实习经历，曾在河南睿软科技有限公司做java开发实习生，组内主要负责河南省公共资源监管平台的开发工作，我主要负责交易与监管信息管理模块、数据统计模块、为各地市提供对外接口等。曾在连连银通电子支付有限公司任职技术支持，主要帮助各个商户接入连连的支付业务，并在领导安排下写了一套支付业务模板，供商户参考。在实习期间，得到领导和同事一致认可。在研究生期间，我主要负责了实验室的横向课题，该横向课题是走航式智能岸基监控系统，该系统主要是实现海上无人船只的信息管理、监控报警、数据收集、卫星通讯。我主要负责整体系统的设计，困难点的突破其中包括数据接收与解析，报警等模块。目前该系统已持续运行一年多，并在不断的进行功能升级与维护。熟悉java、mysql、redis、springboot等，同时了解springcloud、docker等技术。平时喜欢写博客，CSDN访问量10W+,粉丝数1k+，也搭建了自己的个人博客网站。以上是我的自我介绍。

Java项目都有用到微服务。

主要是用nacos 来做注册中心，用openfegin进行远程调用，用elk做日志收集，sleuth zipkin做链路追踪，查看哪个服务时间比较长去优化，mongodb存储解析后的数据，redis存储热点数据，kafka做消息队列，mysql存储关系型数据，minio存储文件。

介绍一下你的项目

我最熟悉的项目是走航式智能岸基监控系统，这个项目主要是负责海上无人船信息的管理。我们想要看到这个无人船在什么位置，以及无人船可以探测到的信息。我们在无人船上装上这个传感器，比如声学、速度、摄像头等，然后这个数据被采集到之后，由于海上是没有网络的，所以我们只能通过这个卫星去传输数据。

功能模块拆分、利用云效管理

功能拆分基本上是根据业务低内聚高耦合程度进行拆分的，比如用户模块与无人船模块无关联，拆分为两个模块，模块与模块之间的信息交流利用openfegin来实现，openfegin其实就是一次http的调用。利用云效进行需求的管理，需求的拆分，二次确认，缺陷管理等。

数据接收与解析模块

北斗服务是已经提供了一个北斗的服务端，我们利用netty做了一个客户端连接北斗服务端接收北斗服务端数据的推送，这一块我们可以动态的配置北斗服务端的ip 端口 账号 密码等信息。

天通服务，是我们自己做了一个netty服务端，接收客户端传过来的消息。

铱星服务，是有消息发送到邮件，我们通过定时任务获取邮件进行附件内容的获取。

在netty服务端和客户端中我们做了自定义的编码解码器。

我们通过这三种方式进行数据的接收后，把接收后的数据文件进行minio文件备份，然后发送的kafka消息队列进行解析。

解析是我们根据报文，获取到协议id,我们的协议信息是存储到我们mysql中，从mysql中拿到协议项的具体内容进行解析，协议项中存放的有主键、协议id、协议项的含义、所占的字节、起始位置。由于我们经常按照协议id来查询，所以增加了协议id的普通索引。我们拿到协议项后就像我们查字典一样，根据起始位置，所占字节进行协议解析，解析完成后将数据存储到mongodb数据库中。因为我们的协议有多种多样的所以字段个数和名称并不相同。协议id是在我们的固定位置的。

报警模块

我们是将解析后的数据一方面进行存储外，另一方面发送到kafka传送到报警服务，因为我们解析后的数据里有无人船的唯一标识id，去判断该无人船需不需要报警如果需要就根据我们的报警规则，我们报警规则大多数是判断某个值是不是不在正常范围内，所以我们使用spel表达式，里面存储其实就是一个逻辑表达式，比如a==1，spel 可以把a给我们替换为具体的值，我们根据解析到的a进行替换从而实现报警。此外我们还有超时报警，比如10分钟内没有消息回传，我们是根据redis监听过期时间来实现的，在给设备添加完这样的规则后，开始计时，如果在10分钟内有就续期，没有就报警并重新开始计时。我们将报警的信息分三个等级，紧急的异常我们会通过电话通知，不紧急的通过短信和邮件通知。

前后端遇到的问题

地图显示页面，需要查设备的位置，需要查mongo数据库，往往是查最后一条的就可以知道位置了，但是由于mongo数据大概有100w左右吧，导致查询比较慢大概10s左右，因此我给mongo加了索引，有效果但是还达不到理想的状况。因此我们将这部分数据放到了缓存中，因为它经常被查询，并且数据回传是差不多10分钟回传一次的。在我们每次调接口时，我们先查缓存，如果redis中存在且失效时间大于1分钟，则直接返回，如果小于1分钟也返回，在返回之前开一个异步任务来更新redis中的数据，如果不存在就查mongo放入缓存。

部署部分我们使用流水线部署，编写dockerfile利用k8s进行编排，组了一个三节点的服务。

接口的校验 vaild Validated 校验非法参数

数据统计 就是去查数据库 该月交易多少，该年交易多少，把历史的交易做了缓存处理。

Elasticsearch 存储了公告信息，利用自带的分词器，直接进行查询。