**协作：**

开发过程中，会检查需求文档上的配置说明，检测是否有出差的地方，如：埋点配置。

帮助市场同学与数据组的同学沟通，捞取一些过滤条件比较复杂的数据

埋点，事件追踪（Event Tracking），用于跟踪用户在页面上的行为和操作，并将相关数据发送到后端进行存储和分析。

**产品优化**：通过对用户行为数据的分析，可以了解用户的兴趣和需求，制定更加精准的运营策略，提高转化率和留存率，如：监控不同首批kv下，用户点击预约按钮次数，判断用户的喜好，提升预约转化率。

**页面性能监控**：收集页面性能指标数据，如：平均加载时间，及时对页面做性能优化，提升交互体验，减少用户流失

**其他优势**：

旧版ga也要停用，新版ga观察数据不直观

把埋点数据接入数据后台，保证数据安全性，方便查询具体数据，如查预约手机号等

**自研埋点工具：**

对比之前一直使用谷歌的第三方埋点分析工具，有以下优点：

1. 低数据延迟：ga数据收集需要经过24-48小时的数据处理和汇总，而数据后台则能够在较短的时间内（一两分钟）快速计算和呈现结果。
2. 精准投放：更快的改善页面效果，挖掘用户喜好，减少投放成本
3. 定制化程度高：根据业务需求，自定义上报数据内容，不受第三方工具影响限制，具有更高的灵活性和可扩展性，

如：

上包预约手机号，对不同渠道手机号预约数据分类汇总。

查询不同页面模块，用户的流失情况及预约情况

**研究GA埋点方案：**

一开始最大的问题是没有类似的项目经验，对埋点概念及其实现模糊不清，所以先从日常的活动埋点开始研究，了解活动的埋点方式及上报的埋点数据内容。之后阅读ga的官方文档，大致属性ga分析工具的使用及相关指标的概念和计算方式，并作出归纳总结

**开会讨论：**

开会讨论的目的是了解清楚市场运营所关注的数据指标，实现相关数据的收集上报

并向市场同学讲述ga分析工具的指标计算方式，便于他们更好理解。如：ga的许多指标都是基于会话实现的，讲解ga会话概念。

之后讨论并总结需要上报数据后台的数据指标。

**性能监控：**

参考ga文档，调用performance的相关api实现时间指标计算，由于performance.timing即将弃用，但getEntires api兼容性不是特别高，所以做了兼容出来，同时实现了两种方案去计算时间指标。

除此之外，还接入了web vitals库，计算并收集页面性能数据

**事件上报：**

主要解决将数据同时上报给ga和数据后台，并且要考虑旧版ga停用后，能将数据无缝切换上报给GA4，阅读了相关文档，发现可用gtag.js做旧版ga和GA4的数据上报兼容，于是将gtag.js接入埋点工具，在上报数据后台的同时，同时调用gtag.js的api上报ga。

**AB测试数据区分：**

活动的AB测试是用谷歌的优化工具实现，但是埋点工具无法识别优化工具对页面的更改，为了实现AB测试数据区分，了解了优化工具的运行逻辑，并通过优化工具的代码注入功能，注入一个区分不同页面效果的变量，通过识别该变量来实现数据区分

**前端埋点开发：**

研究前端的性能监控实现方案，如何计算加载、白屏、响应等事件指标

研究旧版ga和GA4的数据上报方式，并作两者的兼容，同时对接数据后台接口，最终将数据同时上报给三者

研究GA的会话计算方式、实现ga的转换率和跳出率计算

研究如果通过谷歌的优化工具达成对上报给数据后台的数据做AB测试区分

**为什么需要图形化界面：**  
查询的数据不够直观，数据后台总数据查询，会一次性查出不同维度下对应所有指标的数据，加上没有将数据图形化，导致数据量过大的同时，也无法很好的查看数据的变化趋势

缺少了对数据划分，比如：性能相关指标数据、事件指标数据和实时数据杂糅在一起，无法很好区分查询。

**图形化界面：**

性能监控页面：通过数据图形化，更好看出数据变化趋势，及时发现问题并改善页面问题

事件查询页面：可通过筛选特定的活动或事件，进行特定事件数据的搜索，并通过图形化展示数据趋势

实时数据页面：通过观察30分钟内用户的实时数据，可直观分析页面访问情况

**分析需求：**

参考ga，划分成性能、事件、实时页面。在埋点代码设计的时候就有考虑到区分不同的事件类型，通过上报event\_type来区分用户触发事件和非用户触发事件（如性能指标事件）

**设计原稿：**

一开始是考虑用mp设计原型的，但是了解现在比较火的在线设计软件figma的优势之后，决定使用figma来设计

**优化及交互：**

结合数据后台的设计风格，将原稿ui优化成初版ui，并增加新的数据筛选和分类功能。

后面在优化了ui颜色风格不统一问题，直接采用elementUI主题色。

完成页面开发后，总结了相应的操作文档，并且增加了数据后台总数据查询的操作文档。

之后收集用户的使用反馈，并且做交互和功能的优化。