

大规模信息系统构建技术导论

分布式MySQL系统模块实现与测试报告

|  |  |
| --- | --- |
| 姓名： | 黄政 |
| 学院： | 计算机科学与技术学院 |
| 系： | 软件工程 |
| 专业： | 软件工程 |
| 学号： | 3200100836 |

2023 年 6 月 1 日

目录

[一．系统模块简介 3](#_Toc136082328)

[二．Client (Web)模块实现说明 3](#_Toc136082329)

[2.1 模块组件设计 3](#_Toc136082330)

[2.2 主要数据结构 4](#_Toc136082331)

[2.3 流程图设计 5](#_Toc136082332)

[三．测试结果 6](#_Toc136082333)

[3.1 网页版Client功能测试 6](#_Toc136082334)

[3.1.1 测试用例 6](#_Toc136082335)

[3.1.2 测试结果 7](#_Toc136082336)

[四．开发体会 10](#_Toc136082337)

[参考文献 10](#_Toc136082338)

# 一．系统模块简介

本人在分布式MiniSQL系统中负责研发实现Client（Web）部分，通过BS架构，使用React+flask搭建Client Web服务平台，并且进行公网部署，实现多端访问和查询的基本功能。

在网页版的客户端中，需要实现以下功能：

（1）能够进行表的建立、删除

（2）能够进行记录的插入、删除

（3）能够进行记录的查找

（4）能够保证数据的一致性

（5）能够显示数据表和有关查询结果以及报错信息

# 二．Client (Web)模块实现说明

## 2.1 模块组件设计

本模块包括以下几个部分：

（1）前端UI界面

（2）前端网页的生存周期检测

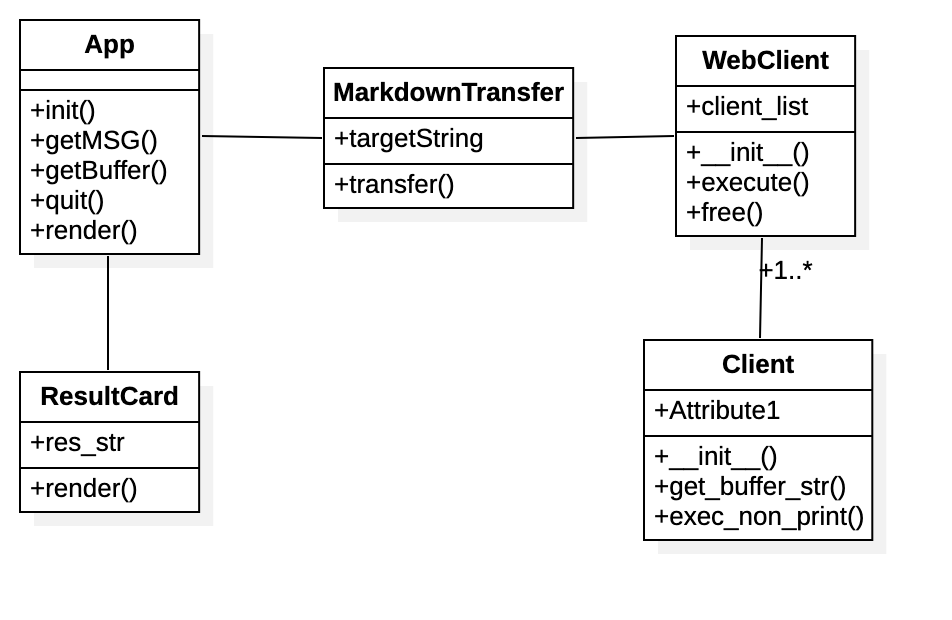
（3）前端查询和连接请求

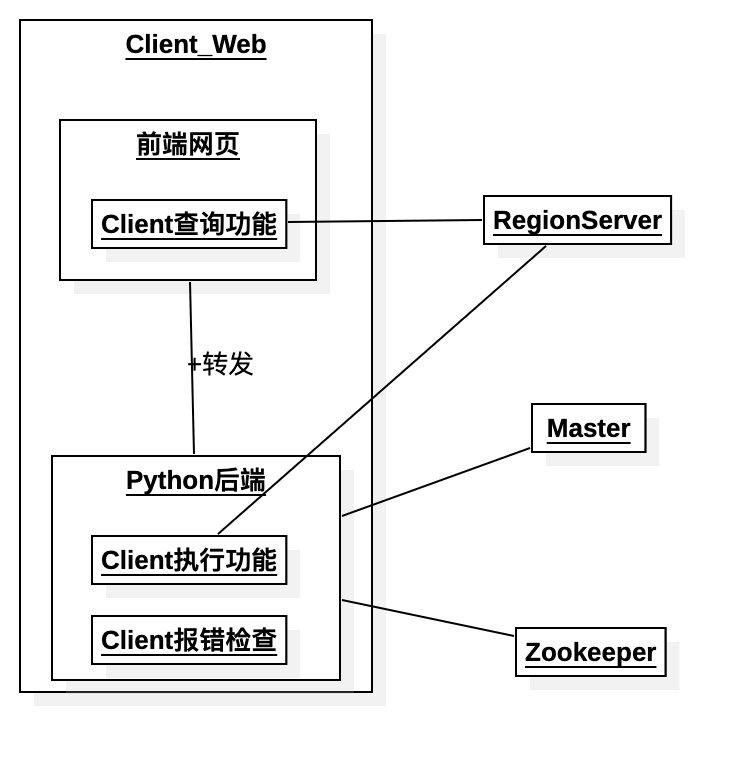
（4）后端对接入请求的响应

（5）后端转发查询结果、报错信息

相比于原有的单机客户端，本客户端在其基础上做了网页版的移植和重写，使得其适应于BS架构的分布式查询。

## 2.2 主要数据结构

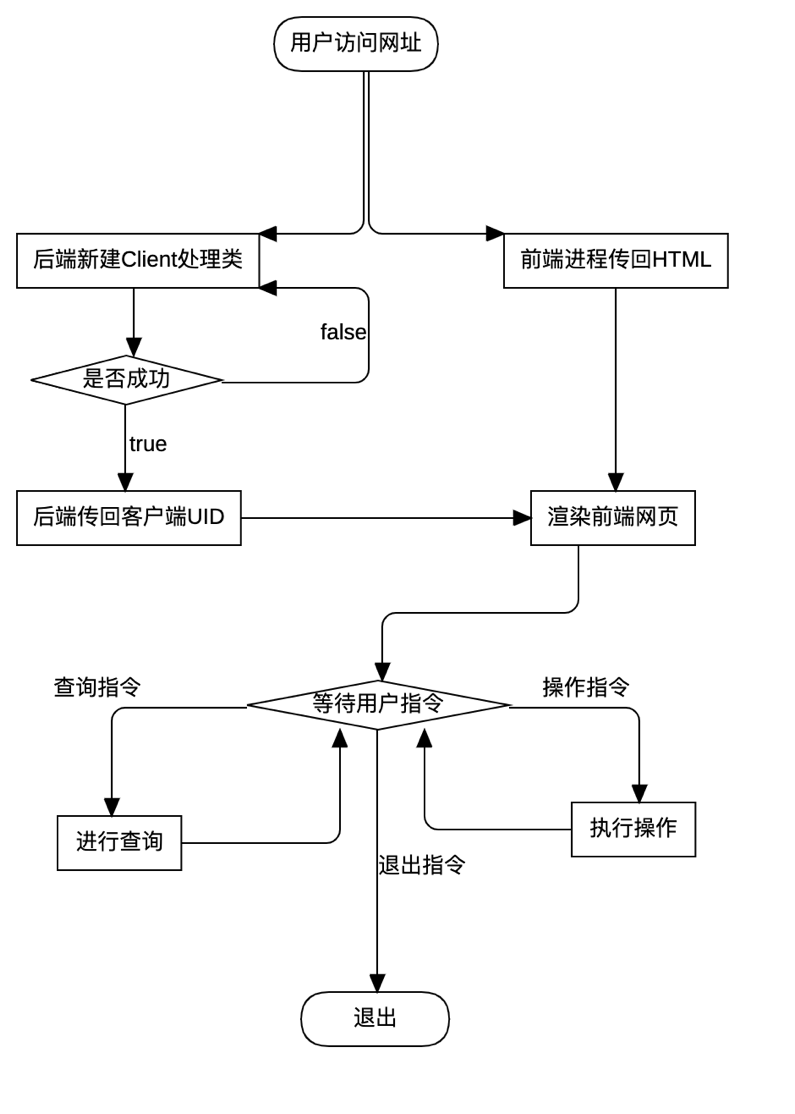




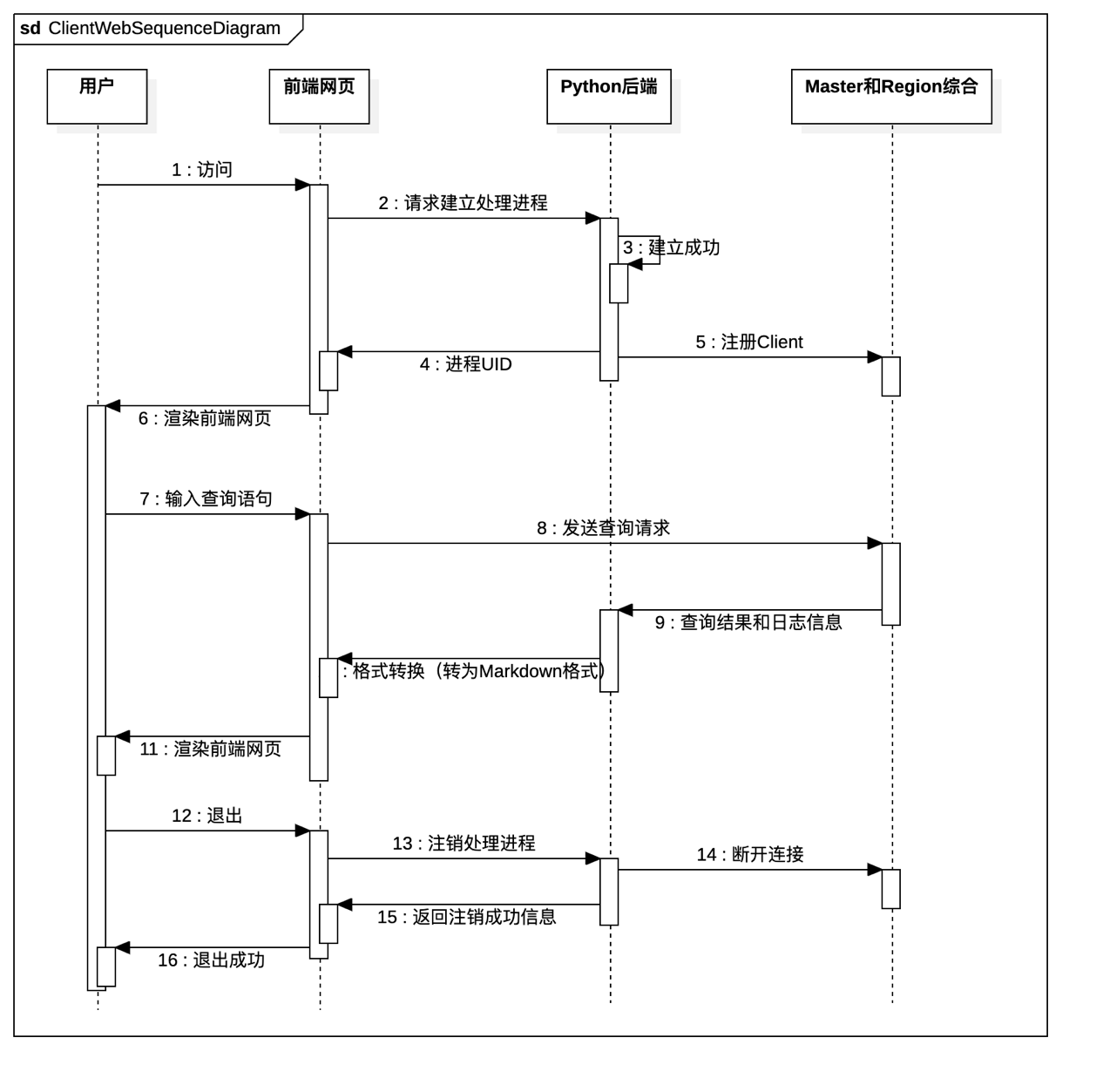
在网页版的客户端中，在用户输入网址请求时，网页APP会想Python后端进行一个申请和注册，Python后端会分配相应的处理线程给前端网页，并且代为向Master和zookeeper进行注册。

对于前端网页获取用户查询和执行的信息，通过后端客户端进行语法检查，客户端将结果返回，随后网页端向对应的Region发送执行和请求信息，执行和操作结果返回至前端网页，随后再请求后端进行字符处理，将表格数据转换为Markdown格式，使用网页端现成组件渲染为表格。

## **2.3 流程图设计**



时序图：



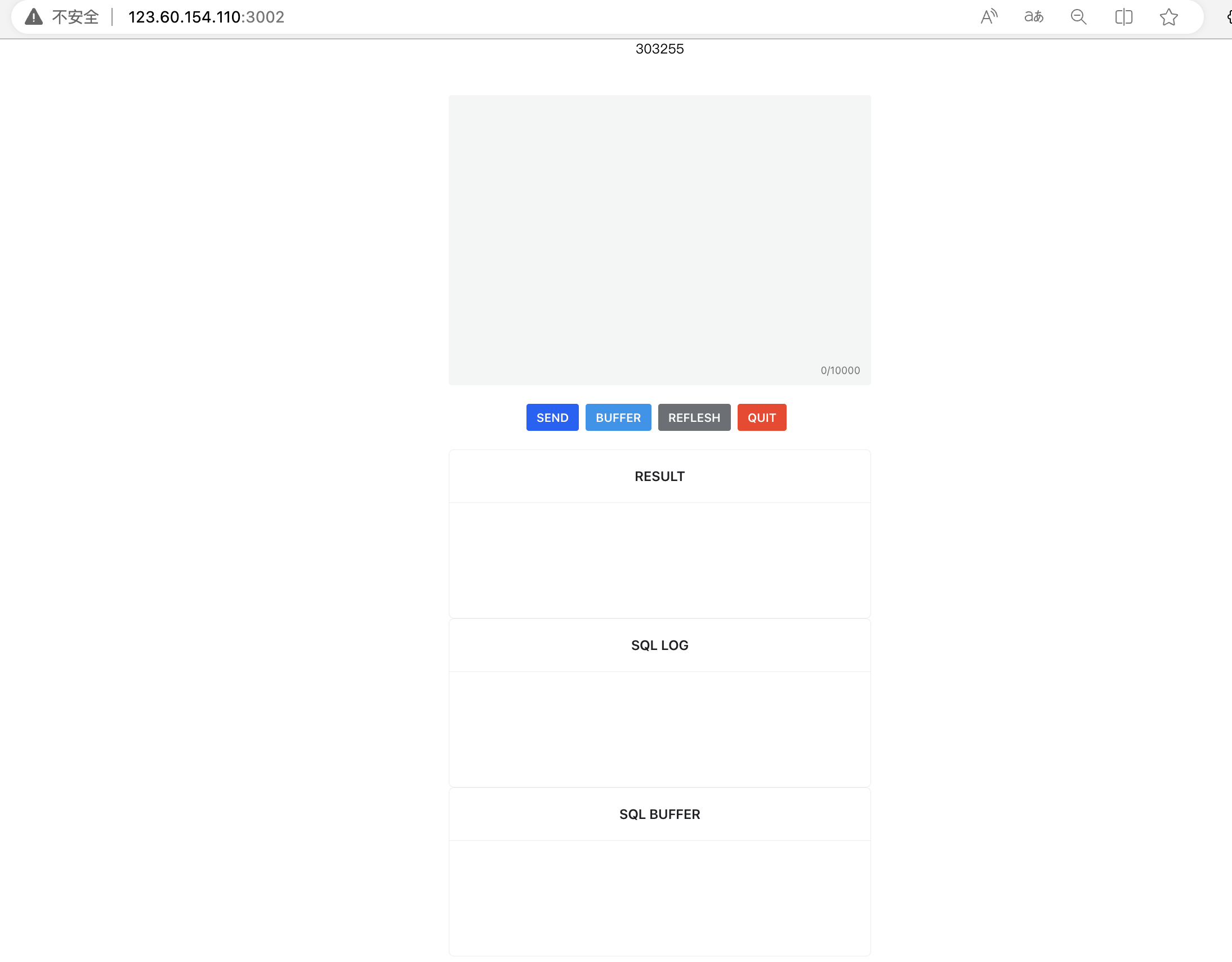
# 三．测试结果

## 3.1 网页版Client功能测试

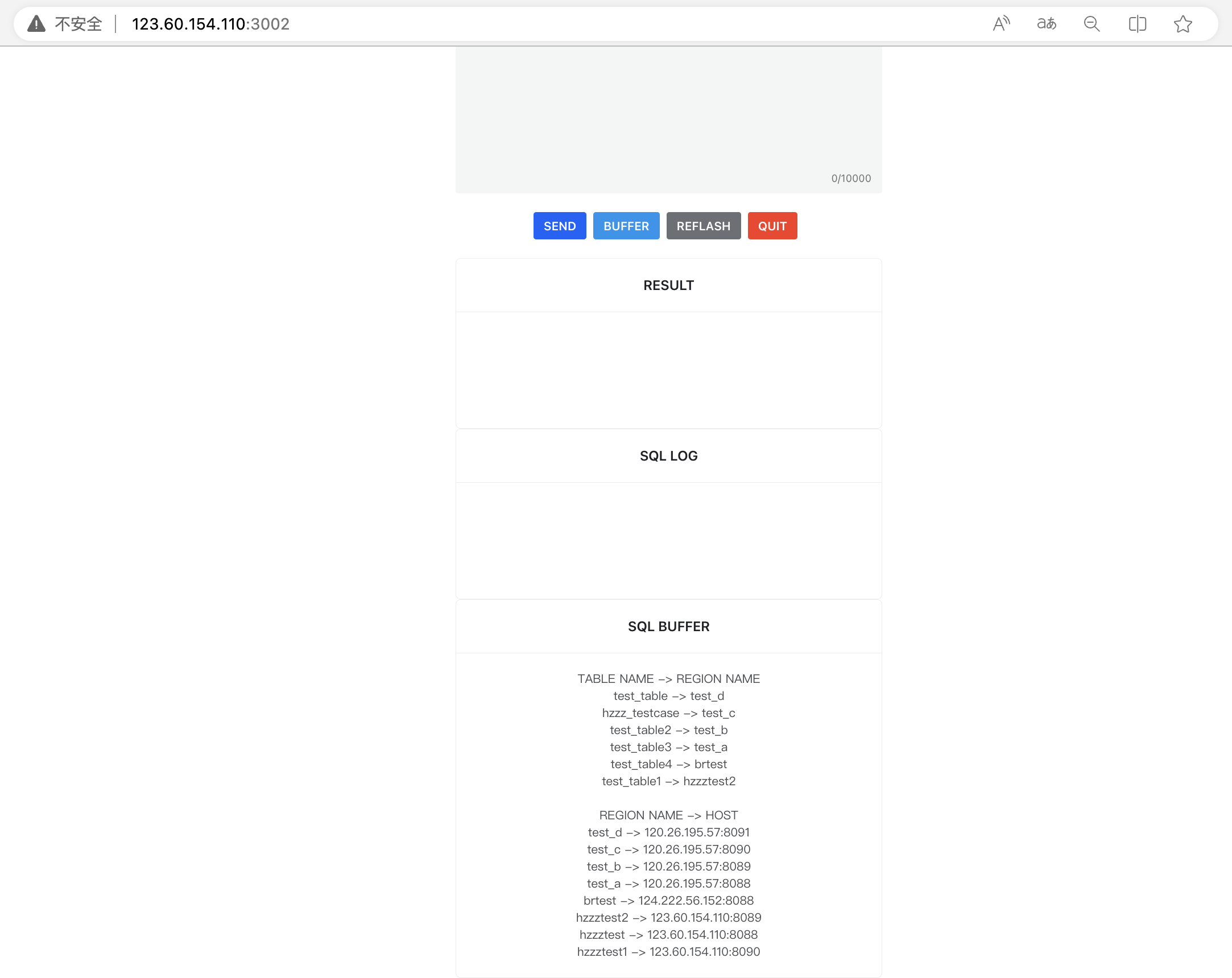
### 3.1.1 测试用例

### 3.1.2 测试结果

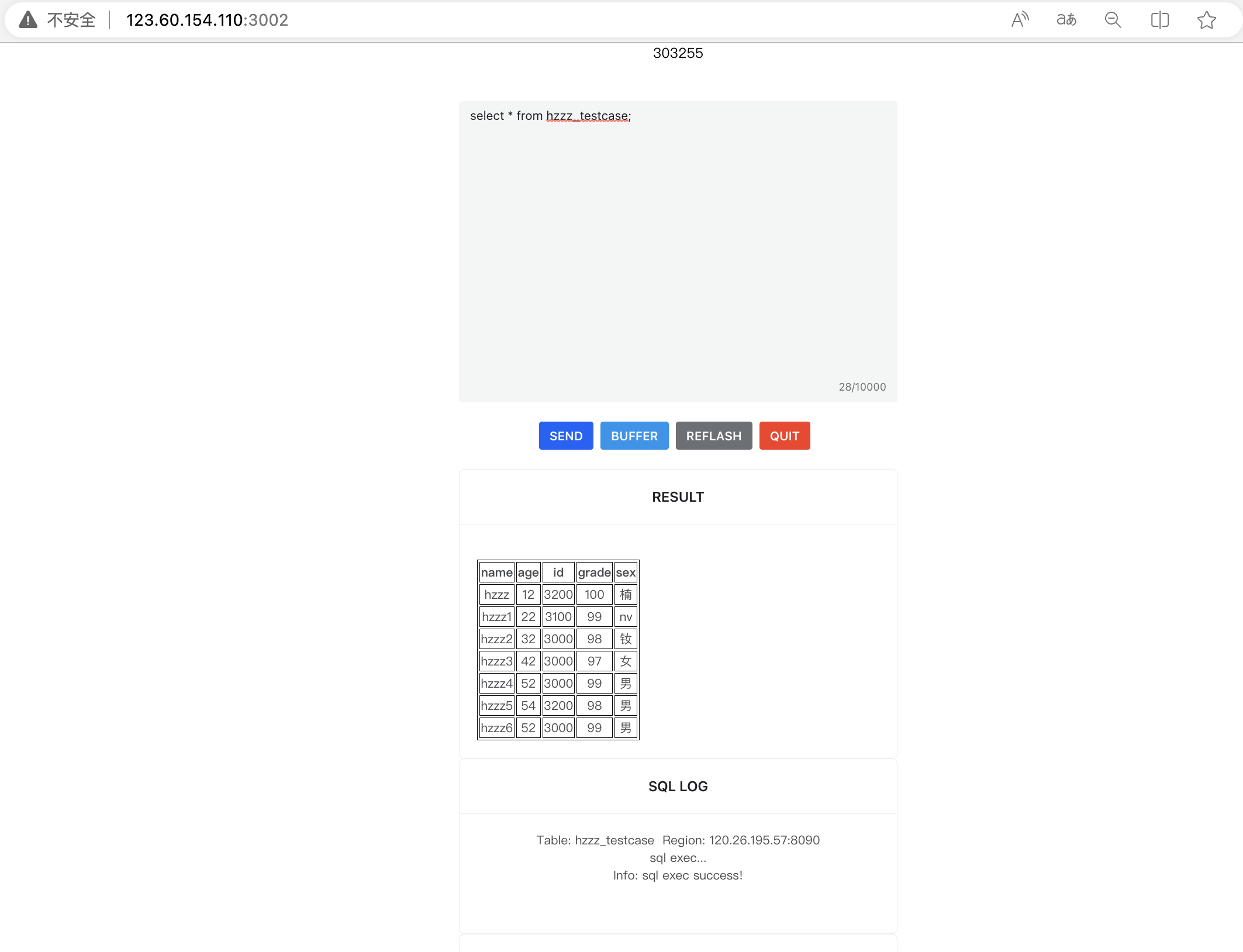
自上而下依次是客户端uid，指令输入框，指令操控按钮，查询结果、SQL的日志输出、客户端缓存



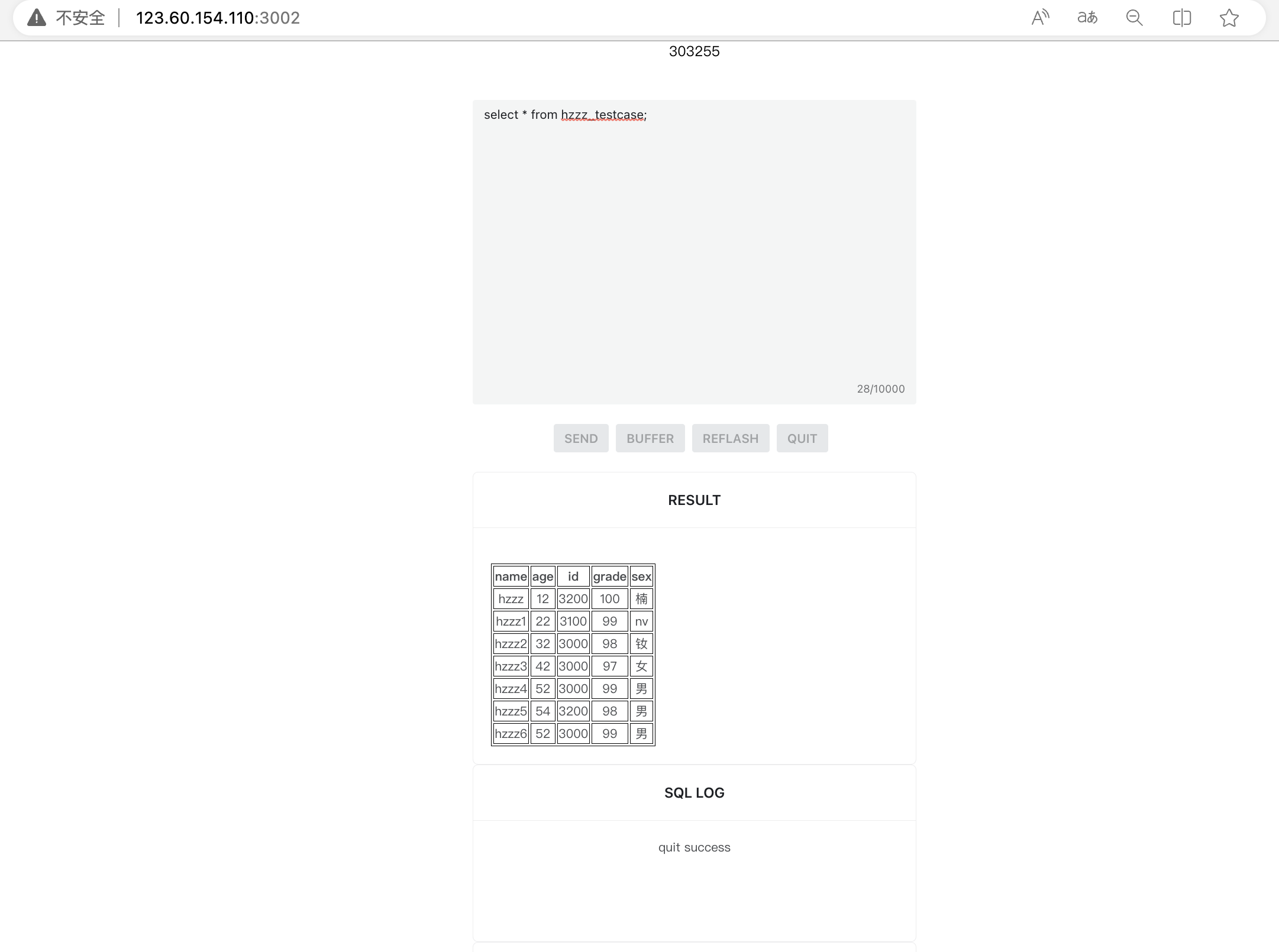
点击buffer按钮可以得到当前客户端的缓存，点击reflash可以刷新当前的客户端缓存



在输入框中输入要查询的SQL语句或者要执行的操作，点击SEND即可发送相应的执行请求，并且会显示相应的日志输入和查询结果：



点击quit按钮即可退出，此时如果成功退出，控制按钮都会处于不可用状态。



# 四．开发体会

在本次分布式MySQL系统模块实现中，为了体现我们小组的创新性，我进行了网页版Client客户端的开发工作，使得我们的整个分布式系统在客户端的平台可移植性上有了非常大的跨越。

我的工作能够顺利进行离不开我们小组成员良好的代码习惯和严格的开发规范，为我的工作提供了清晰明确的文档和API。此次合作让我深刻明白了软件开发中规范性的重要。

同时，此次开发工作让我更加深刻地理解了BS架构下的软件开发，对于分布式的服务功能有了更加深刻的认识。

不过此次开发工作也有许多不足，比如说客户端的许多功能并没有被完全的使用JavaScript实现，部分功能比如Markdown文本的转换，还依赖于客户端服务器的处理，希望以后如果有机会我能够继续完善我们的工作。

# 参考文献

[1] React官方文档https://react.docschina.org

[2] Flask官方中文文档https://flask.net.cn