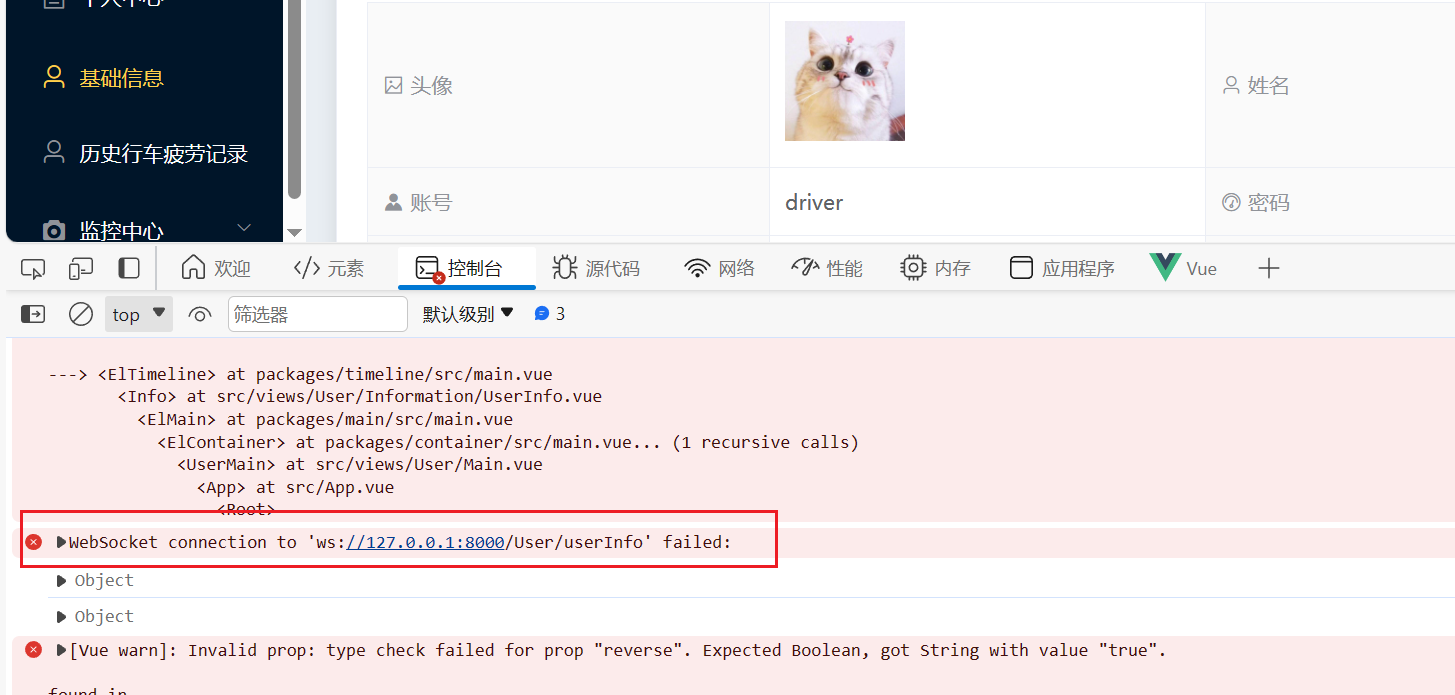
* 

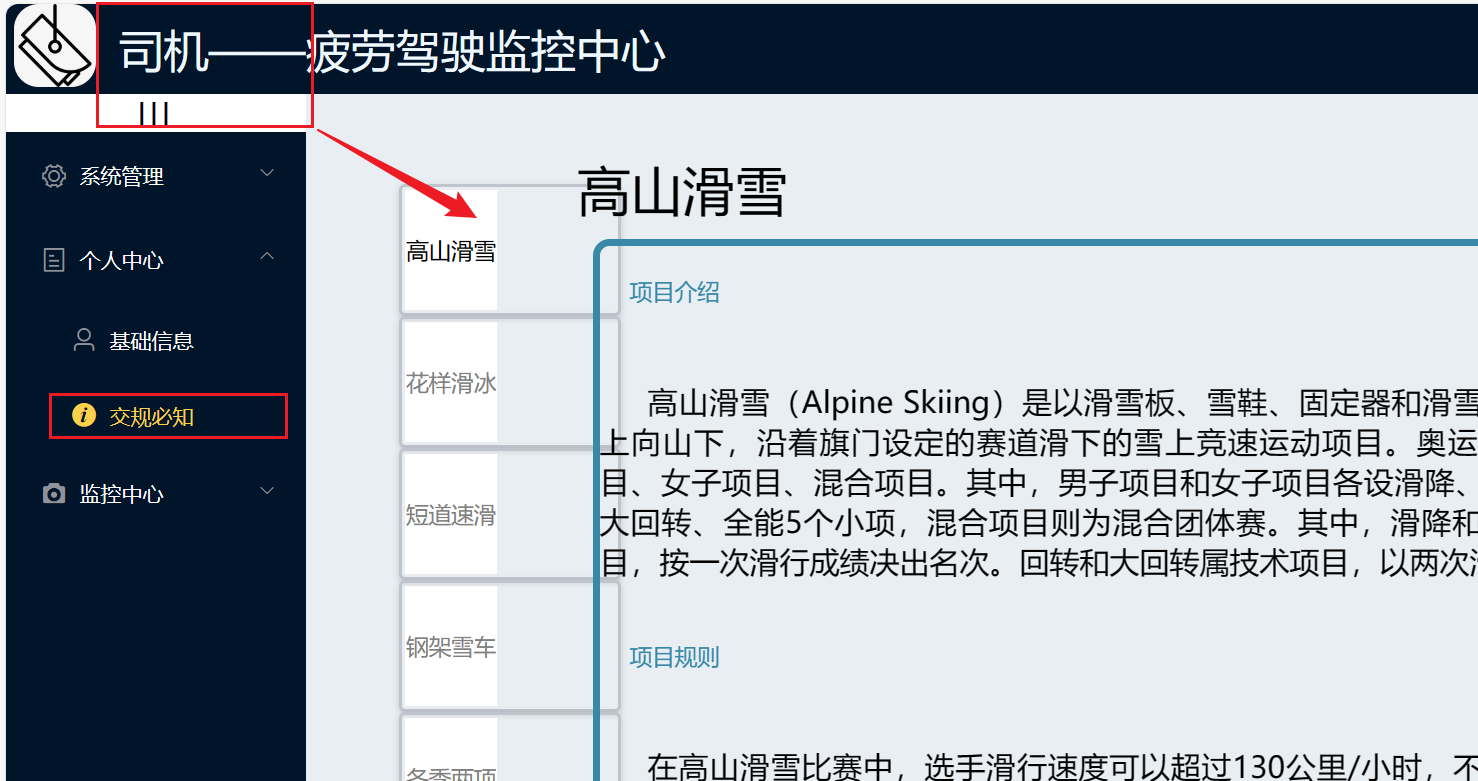
👆司机界面的地图或许显示最近的服务区并标注出来，并在前端页面上实时WebSocket连接显示距离，？？？，

* 

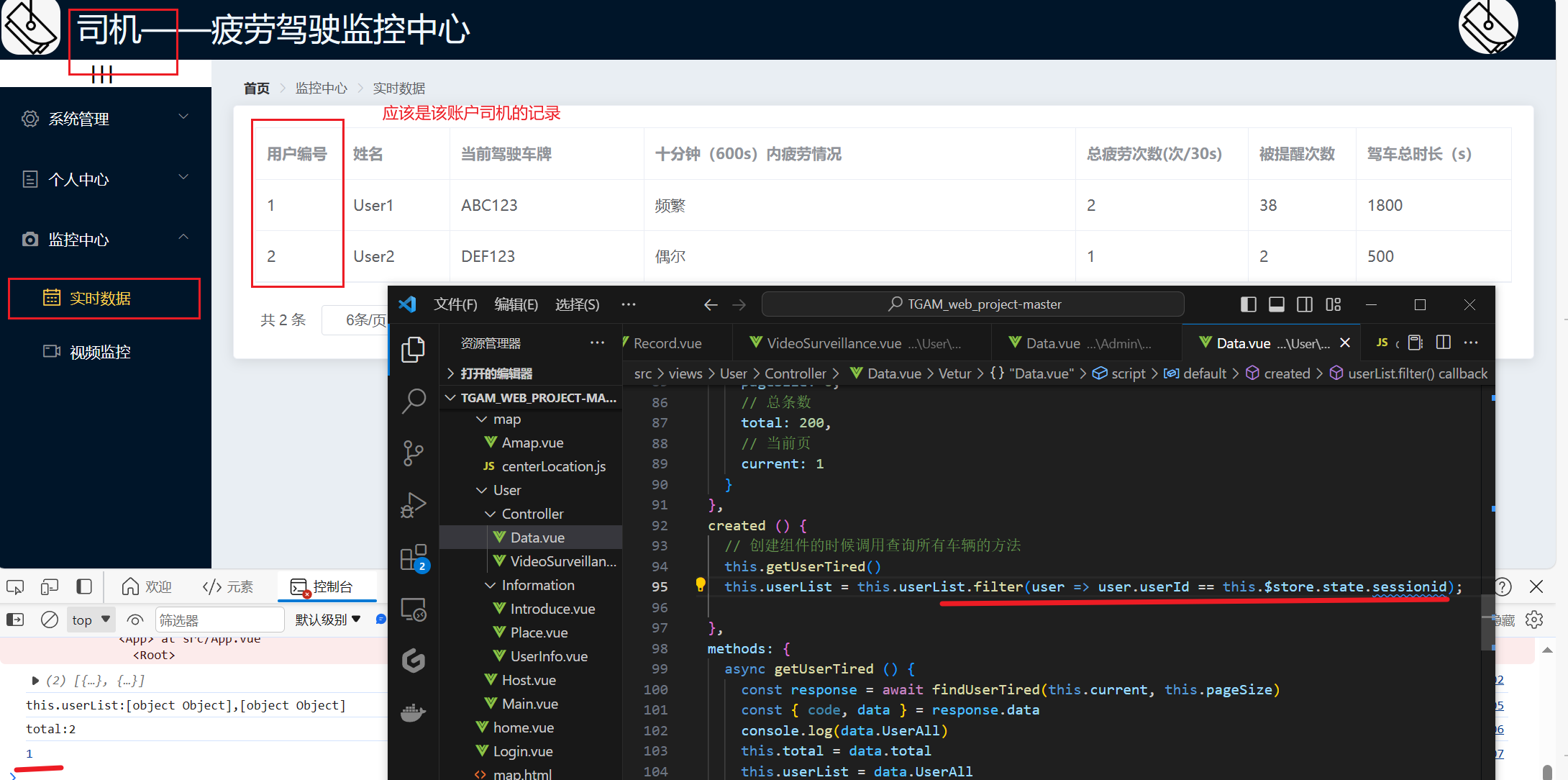
👆在该页面想要实现events的动态监听数据库：从管理员提交修改驾驶分处，后端检测到events数组发生变化，向前端[http://localhost:8080/#/userinfo发送消息，前端接收到消息，更新前端的events数组](http://localhost:8080/#/userinfo%E5%8F%91%E9%80%81%E6%B6%88%E6%81%AF%EF%BC%8C%E5%89%8D%E7%AB%AF%E6%8E%A5%E6%94%B6%E5%88%B0%E6%B6%88%E6%81%AF%EF%BC%8C%E6%9B%B4%E6%96%B0%E5%89%8D%E7%AB%AF%E7%9A%84events%E6%95%B0%E7%BB%84%EF%BC%8C%E8%AF%B7%E7%94%A8websocket%E5%AE%9E%E7%8E%B0)

但是我在前端WebSocket连接失败



* 

👆该界面打算做成静态界面，还没完成

* 

👆数据有点问题，不同于管理员界面是所有用户最近1h（6条）的记录，应该显示该账户司机的tired数据

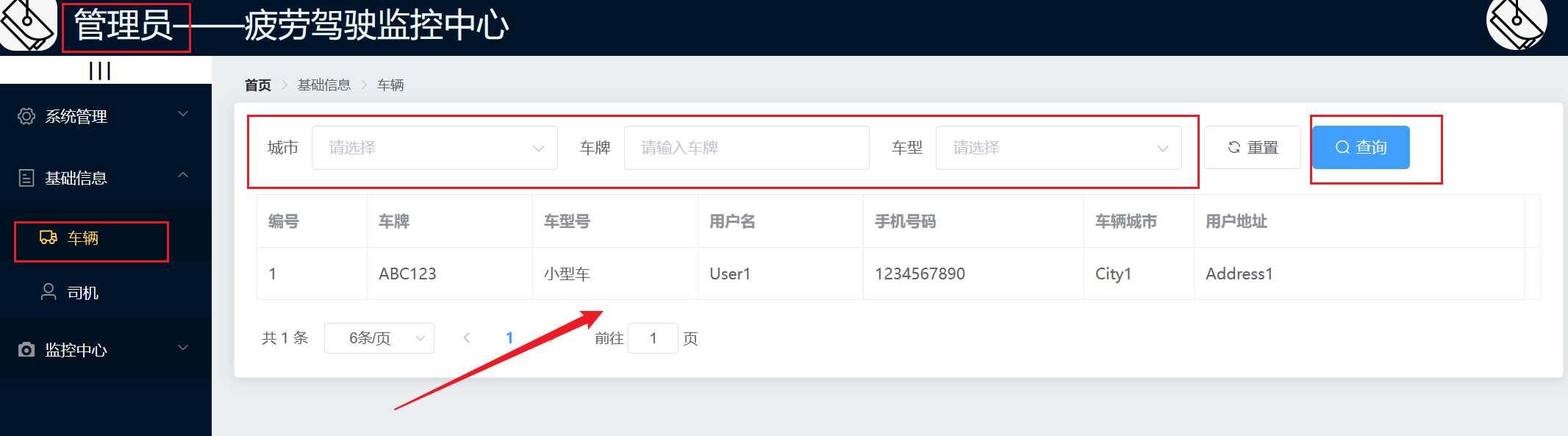
* 

👆对应措施还是没开始写

* 

👆点击“警告疲劳”按钮，所有启动车的司机界面（因为是车载系统所以默认司机在线-车子启动）有语音播报，WebSocket连接



* 

👆点击查询按钮，前端发送条件，后端实现分页带条件查询，返回数据，前端渲染

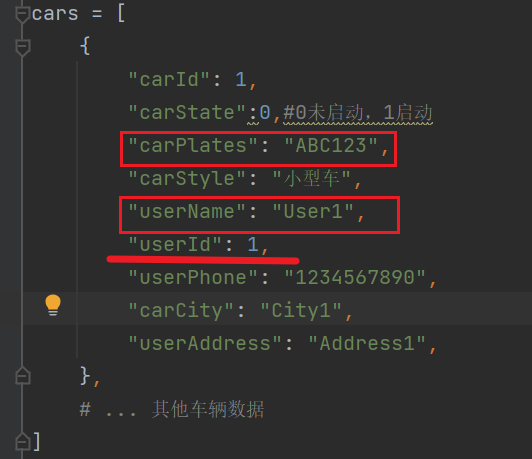
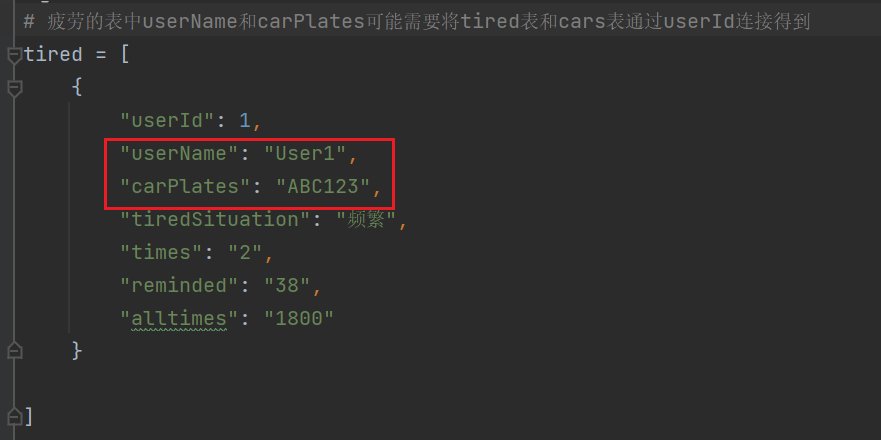
* 

👆点击编辑/删除按钮，前端发送响应数据，后端处理，返回处理结果是否成功，websocket连接渲染

* 

👆管理员界面的这个实时数据表中数据，后端在处理每个司机画面时，每十？分钟统计一下情况，形成一条记录，存储到tired表中（userId作为外键）。可以认为司机1h内的数据是有效的，即每位司机的tired记录最多6条，多余6条的删除掉最早的记录。

前端利用WebSocket动态监听数据库（手动刷新通过加载函数获取后端数据不现实），渲染表格。

* 

👆后端处理疲劳tired表中的userName和carPlates，可能需要和cars表通过userId连接获取



* 每警示一次，警示次数+1，当管理员界面自动检测到警告次数达到一定数量，触分“扣分”事件，将userId传给后端，后端处理：

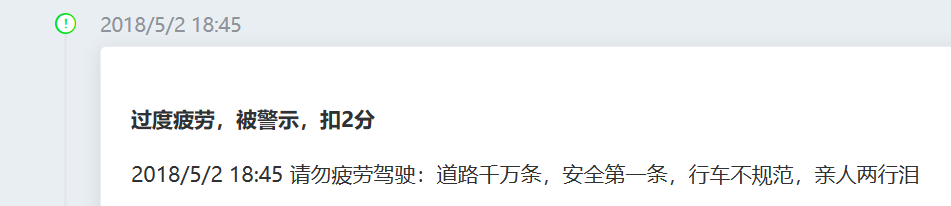
👆（后端数据）user实体类数组数据更新：该司机的驾驶分被扣2分（规定每次扣2分）

（后端数据）同时获取当前的年月日时分秒时间，写入events实体数组中

（前端界面）当该司机正好在线时，司机界面弹出警告窗口：分数变动

（前端界面）时间线的事件需要增加

上面两个需要前端user实体类、events实体类数组 跟后端的数据一直建立联系，前端才能在后端数据变动时及时处理。WebSocket动态监听数据库

* 

👆这个被扣分的理由是管理员警告次数达到一定的限定次数，自动触发事件，这一理由。如果想要丰富理由，改进：管理员界面除了有警示的按钮，还有扣分的按钮，点击弹出对话框，填写扣多少分及其理由，发后给后端，存储到events实体数组中。然后前端渲染出来

修改后参数变化：

* 

👆管理员获取重度疲劳司机的实时摄像，待完善