武汉大学计算机学院

本科生实验报告

**公交安行——司机检测和云端交互**

**一体化管理系统的设计与实现**

专 业 名 称 ：计算机科学与技术

课 程 名 称 ：软件构造基础

指 导 教 师 ：贾向阳

学 生 学 号 ：2020302111148

学 生 姓 名 ：曹雨欣

二○二二年五月

**郑 重 声 明**

本人呈交的实验报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本实验报告不包含他人享有著作权的内容。对本实验报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本实验报告的知识产权归属于培养单位。

本人签名： 日期：

摘 要

本项目旨在完成一个保障公交驾驶安全、关注公交司机健康的智能安全驾驶系统。本项目包括传感器硬件、Android端司机APP（java）、web端管理云平台（javaScript、html、css），以及服务器部分（c#）。

本项目中，前端使用Vue框架，实现数据绑定，并在实时更新后自动渲染；使用bootstrap框架进行了css样式的编写；使用Ajax方法与服务器交换数据并更新部分网页内容；使用Leaflet库构建web交互式地图。

后端使用ASP.NET Core框架，利用EF Core，结合LINQ操作MySQL数据库，利用HTTP协议进行数据发送与接收，通过异步编程来实现高并发的请求，并保证了页面交互的流畅度。

此外，Android端使用java开发。

我在本项目中主要担任了前端开发工作，参与了后端c#控制层的部分编写与Android APP的部分编写。

**关键词：**公交车管理; .NET; Vue; Android

目 录

[武汉大学计算机学院 1](#_Toc107178108)

[本科生实验报告 1](#_Toc107178109)

[目 录 3](#_Toc107178110)

[1 选题动机 4](#_Toc107178111)

[1.1 选题背景 4](#_Toc107178112)

[1.2 选题意义 4](#_Toc107178113)

[1.3 我承担的工作 5](#_Toc107178114)

[2 软件需求与设计 6](#_Toc107178115)

[2.1 软件需求 6](#_Toc107178116)

[2.2 功能组成 7](#_Toc107178117)

[3 关键技术 8](#_Toc107178118)

[3.1 ASP.NET Core与Web API 8](#_Toc107178119)

[3.2 EF Core 9](#_Toc107178120)

[3.3 Jwt Bearer 9](#_Toc107178121)

[3.4 前端框架 Vue, Bootstrap, Jquery Ajax 与 Leaflet 10](#_Toc107178122)

[3.5 Android开发技术 10](#_Toc107178123)

[4 案例 10](#_Toc107178124)

[5 结论 12](#_Toc107178125)

# 选题动机

## 选题背景

我国公共交通行业蓬勃发展，拥有世界上规模最大的城市公共交通网络，公交市场规模庞大。

然而，近年来公交车突发安全事故屡见不鲜，如10.28 重庆公交坠江事故、7.7 安顺公交车坠湖事故等，造成人员伤亡和重大财产损失。公交车突发安全事故主要集中于以下几种情况：

（1）司机突发心梗、脑梗、脑出血等，失去意识及控制能力，甚至猝死。

（2）司机违规驾驶、疲劳驾驶、酒驾、毒驾、开车打电话、开小差、单手开车等。

（3）乘客打骂甚至劫持司机，严重影响乘客与司机的安全。

此外，公交车司机的健康问题长期受到忽视。由于工作压力和劳动强度大，公交司机易患上包括颈椎病、高血压、肠胃疾病等多项职业病。驾驶中精神长期处于高度紧张状态，也极易诱发焦虑抑郁等心理问题。不仅持续危害着公交司机的身心健康，也对全车乘客安全造成潜在威胁。

本项目提出一款保障公交车驾驶安全、关注公交司机身体健康的安全驾驶监测系统——公交安行，包括传感器硬件、Android端司机APP（java）、web端管理云平台（javaScript、html、css），以及服务器后端（c#）。

## 选题意义

基于公交交通的广泛性与普及性，公交司机状态检测对于公共安全举足轻重。本项目提出一款保障公交驾驶安全、关注公交司机健康的智能安全驾驶系统。

从社会角度，有助于维护公共交通安全，减少意外交通事故发生频率，促进社会治理，守护居民生命及财产安全。从公交车司机角度，有助于改善公交车司机身心健康情况，保障其劳动者权益。此外，增设公交车安全系统管理人员一职，还将促进社会就业，助力建设智能化的城市公交体系。

目前，在现有的检测解决方案中，公交车广泛使用离线本地存储数据来进行事后追责；油罐车等危险车辆使用GPS定位并计算，来判断是否超速或者疲劳驾驶；中高端轿车使用基于图像识别的DMS驾驶疲劳监测系统。

已有的解决方案往往受限于网络带宽，只能将数据保存在公交车黑匣子中，无法实现及时的预警功能；此外，过去的解决方案局限于视觉或者GPS定位等检测方法，手段较为单一。

受益于当今移动互联网与物联网的飞速发展，本项目为公交车管理提供了全新的解决方案：通过高速移动网，传输传感器检测的实时体征数据与实时图像到服务器云端，并在服务器端储存、分析，将智能分析结果与建议及时地反馈给系统管理人员。我们的项目为提供了更优秀的实时性监管与更多元化的检测指标，为事故的避免提供更多的先机，也为事故之后追责提供足够的依据。

## 我承担的工作

我们三人共同完成本项目的功能与总体架构的设计。此外，我个人在本项目中主要担任了前端web界面的开发工作，参与了后端c# Service层的编写与Android APP的数据显示编写。

前端部分，我编写html完成页面的基本主干部分，使用Vue框架进行数据绑定，参考bootstrap框架完成css编写，与javascript脚本语言完成界面的动态交互，调用DOM等web API来将 web 页面和js语言连接起来。

后端部分，我参与Service层对于数据异常状况判断的函数编写。

Android移动端部分，我设计并编写了健康报表，以及一版没有包括在最终版本的SQLite数据库架构。

# 软件需求与设计

## 软件需求

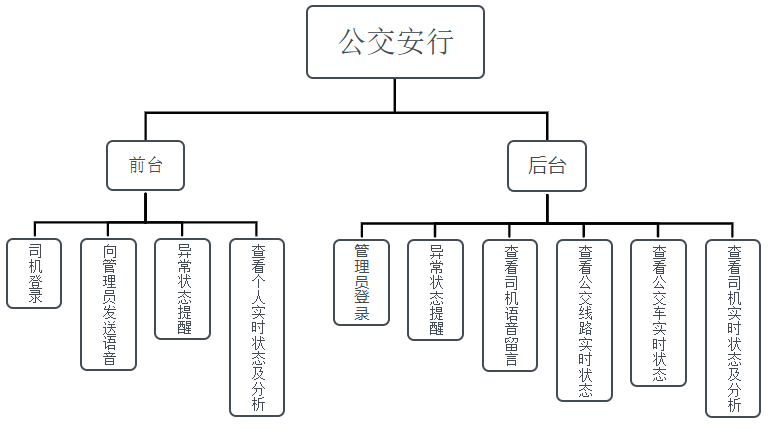


图 2‑1 系统用例图

“公交安行”系统的软件需求可以大体分为前台和后台两部分，下面主要介绍从用户角度看到的系统功能。

前台具体包括司机的操作行为。司机可以使用安卓平板，在APP上登录账号、查看个人实时体征数据（心率、血压、血氧、体温等）。APP会对司机一段时间内的状态做出分析，当判定司机出现异常行为时，会发出语音提醒。

后台具体包括管理员的操作行为。管理员可以在web端登录账号，查看公交线路以及相应司机的实时信息。服务器端储存线路和司机数据，并对司机的实时状态（心率、血压、血氧、体温等指标，以及是否打哈欠、是否系好安全带、是否与乘客发生肢体冲突等）进行分析。当司机出现异常状况时，管理员可以在各个页面看到相关的提示（弹窗、红色未读红点标识等），并及时地回复处理。

## 功能组成

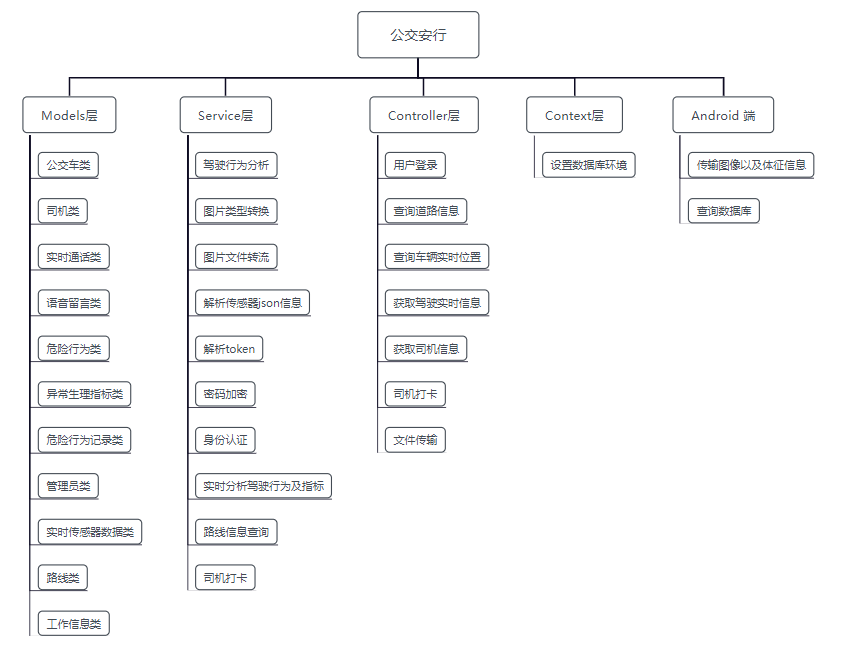


图 2‑2功能组成图

本项目功能组成图见图 2‑2功能组成图，前后端+服务器系统结构关系见图 2‑3。下面分别介绍各个功能组成部分。

Models层用于定义实体，用于存放实体类，与数据库中的属性值基本保持一致。Models层定义了公交车类、司机类、实时通话类、语音留言类、危险行为类、异常生理指标类、危险行为记录类、管理员类、实时传感数据类、路线类、工作信息类，并定义这些类属性的get ()和set ()方法。

Service类用于实现操作的具体方法，并与数据库交互。Service用于存放业务逻辑处理，不直接和数据库打交道，为Controller层提供方法接口。Service层包括驾驶行为分析、司机实时照片类型转换、解析json信息、解析token、密码加密、身份认证、文件转流、实时指标和行为分析、查询路线信息、司机打开等功能。

Controller层控制业务逻辑，并向外提供接口。Controller层负责具体的业务模块流程的控制，主要调用Service层里面的接口控制具体的业务流程。Controller层包括用户登录、查询道路信息、获取车辆及司机的实时信息、数据库操作等功能。

Context层主要用于设置数据库环境。

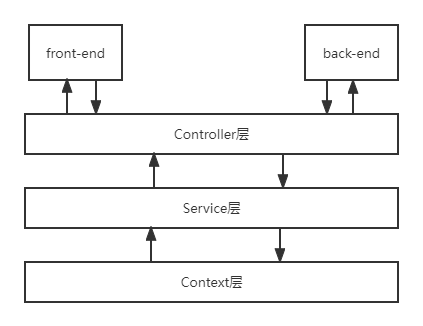


图 2‑3

Android端部分目前可以传输照片以及传感器监测的司机体征信息，以及向数据库请求，查询司机实时数据。APP通过解析传感器发来的UDP包来获取数据并在本地建立SQLite数据库，利用preference生成xml文件保存在本地来保存设置的联系人信息和需要检测的指标。

# 关键技术

## ASP.NET Core与Web API

ASP.NET Core 是一个跨平台的高性能开源框架，用于在 Windows、macOS 或 Linux 上生成启用云且连接 Internet 的新式应用。ASP.NET Core 支持使用控制器或使用最小 API 创建 Web API。

本项目使用 ASP.NET Core搭建Web API服务端程序，并将其部署在Linux PC上，在校园网内提供服务。

本项目服务端框架可以分为Controller层、Service层与Models层。

Controller层负责服务端程序与后台前端页面及车载终端通信，完成数据的传输工作；Service层负责将Controller层获得的数据进行相关业务功能的处理；Models层负责生成相应的数据库表结构。

## EF Core

Entity Framework Core 是轻量化、可扩展、开源和跨平台版的常用 Entity Framework 数据访问技术。

EF Core 可用作对象关系映射程序，可以实现使 .NET 开发人员能够使用 .NET 对象处理数据库。无需再像通常那样编写大部分数据访问代码。

我们使用 EF Core 将实体类之间的关系转化为数据库内表之间的关系，并通过操作对象的方式来对数据库进行操作。本项目数据库的结构如图 3‑1 数据库ER图所示。

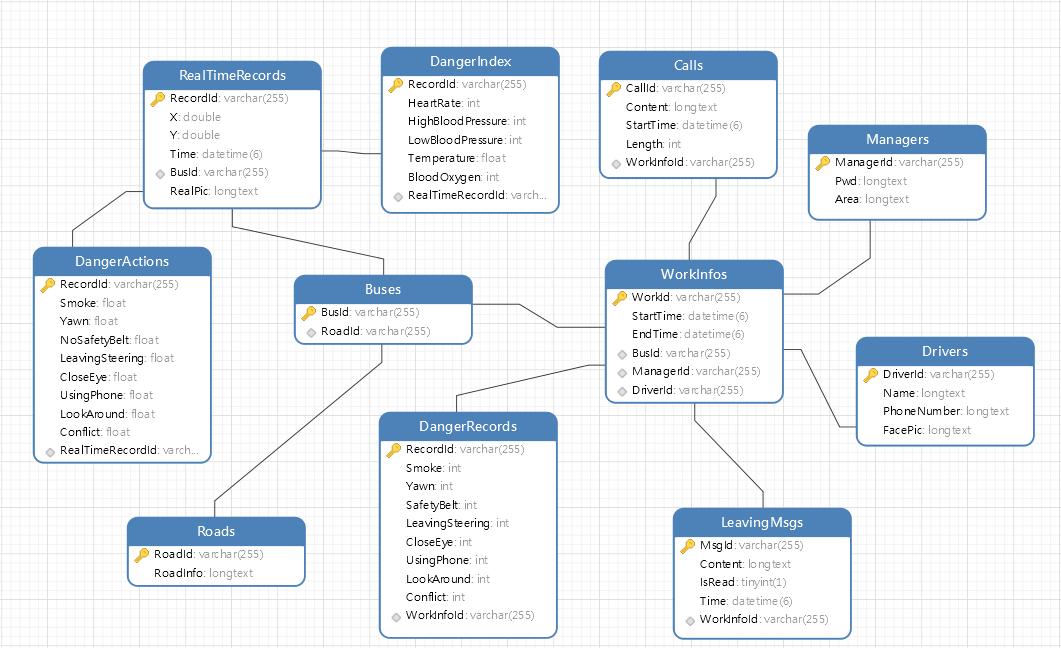


图 3‑1 数据库ER图

## 前端应用的框架与库（Vue、Bootstrap、Ajax 与 Leaflet）

我们主要使用 BootStrap 框架进行页面的搭建；使用 Vue 完成前端页面的数据绑定；使用 Ajax 请求在不重载全部页面的情况下，实现了对部分网页的更新；使用Leaflet库来进行前端地图的渲染。

Bootstrap 是一个 HTML、CSS 和 JS 框架，用于开发响应式布局、移动设备优先的 WEB 项目。

Vue是一套用于构建用户界面的渐进式框架，被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，便于与第三方库或既有项目整合。

AJAX 方法可以在不重新加载整个页面的情况下，可以与服务器交换数据并更新部分网页内容。

Leaflet是一个开源并且对移动端友好的交互式地图 JavaScript 库。Leaflet 简单、高效并且易用。它可以高效的运行在桌面和移动平台, 拥有着大量的扩展插件、 优秀的文档、简单易用的 API 和完善的案例, 以及可读性较好的源码 。我们使用调用高德地图的API作为Leaflet的底图，进行公交线路与实时位置的绘制。

DOM是 HTML 和 XML 文档的编程接口。它提供了对文档的结构化的表述，并定义了一种方式可以使从程序中对该结构进行访问，从而改变文档的结构，样式和内容。DOM 将文档解析为一个由节点和对象（包含属性和方法的对象）组成的结构集合。简言之，它会将 web 页面和脚本或程序语言连接起来。

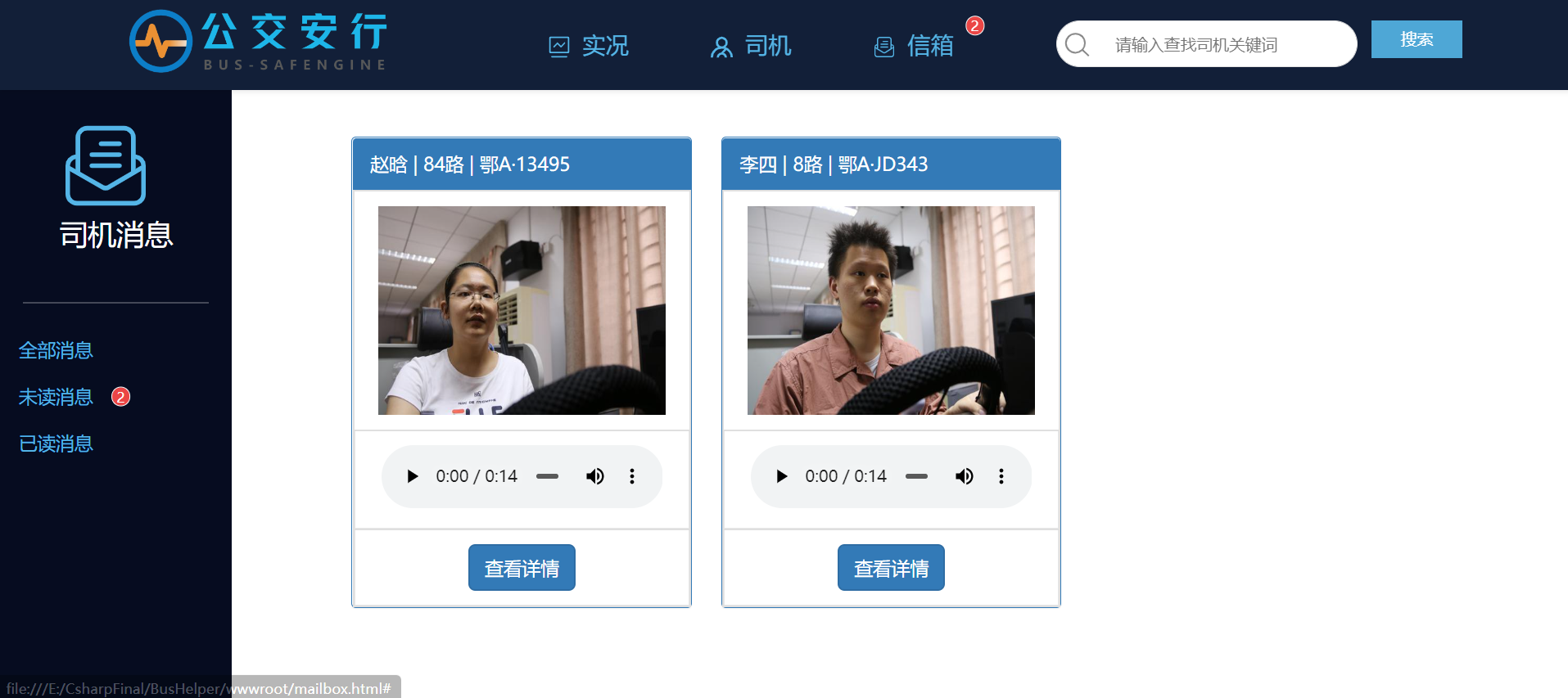


图 3‑2 mailbox页面

我主要负责这个部分的邮箱界面、信息留言弹窗、警告弹窗的开发。我参考了bootstrap库的代码，并在其基础上，根据本项目的需求来修改其源码，写出符合我们需求的css以及javascript代码。

为了实现在空白页面中，根据后端数据来动态地显示消息，我学习使用了DOM（文档对象模型）。通过在mailbox.html文件中增加一个XML类型模板，再在mailbox.js中调用showMsg()，读取后端传输的json，来生成DocumentFragment片段，并插入html中，从而能动态的生成见图 3‑2 mailbox页面的语音留言卡片。

// mailbox.js

1. function showMsg(msg\_json) {
2. var keys = Object.keys(msg\_json);
3. var docFrag = document.createDocumentFragment();
4. for (var i = 0; i < keys.length; i++) {
5. var tempNode = document.getElementsByTagName("template")[0]
6. .content.cloneNode(true);
7. tempNode.getElementById("title").innerHTML =
8. msg\_json[keys[i]].name + "&nbsp;|&nbsp;"
9. + msg\_json[keys[i]].busNo + "路&nbsp;|&nbsp;"
10. + msg\_json[keys[i]].plateNum;
11. tempNode.querySelector("img").src = msg\_json[keys[i]].imgUrl;
12. tempNode.querySelector("audio").src = msg\_json[keys[i]].audioUrl;
13. tempNode.querySelector("audio").setAttribute(
14. 'id', msg\_json[keys[i]].name);
15. docFrag.appendChild(tempNode);
16. }
17. var rowRoot = document.getElementById("card-group");
18. rowRoot.appendChild(docFrag);
19. delete docFrag;
20. }

javascript按照<template>内的模板信息，将json数据插入到html的div标签中。

1. // mailbox.html
2. <template>
3. <div class="driver-col-md-4" >
4. <div class="panel panel-primary">
5. <div class="panel-heading">
6. <a href="info.html" target="\_blank"
7. class="panel-title"
8. id="title">司机名字 | 8路 | 车牌号</a>
9. </div>
11. <ul class="list-group">
12. <li class="driver-list-group-item ">
13. <a href="info.html" target="\_blank">
14. <img src="./media/img/driver1.jpg"
15. alt="张三的照片"height="180"
16. width="100%" >
17. </a>
18. </li>
19. <li class="driver-list-group-item">
20. <audio controls style="width: 98%;" id="0">
21. <source src="./media/audio/music.mp3"
22. type = "audio/mpeg" >
23. 您的浏览器不支持 audio 元素
24. </audio>
25. </li>
26. <li class="driver-list-group-item">
27. <a href="info.html" target="\_blank"
28. class="btn btn-primary" role="button">查看详情</a>
29. </li>
30. </ul>
32. </div>
33. </div>
34. </template>
35. <!-- 司机状态卡片，在下面的div显示-->
36. <div class="row" id="card-group"></div>

## Android开发与SQLite技术

我们选择Android Studio平台完成安卓应用开发，使用Android studio虚拟机和华为实体机综合完成APP的测试。

Android是一个开源的，基于Linux的移动设备操作系统，主要使用于移动设备，如智能手机和平板电脑。Android是由谷歌及其他公司带领的开放手机联盟开发的。

SQLite是遵守ACID的关联式资料库管理系统，它包含在一个相对小的C程式库中。与许多其它资料库管理系统不同，SQLite不是一个客户端结构的资料库引擎，而是被整合在使用者程式中。

我主要负责健康报表和SQLite数据库的开发。

健康报表部分的绘制，我主要使用了java的Graphic类，编写两种图表：条形图与折线图，具体效果见图 3‑3、图 3‑4。开发时，我根据数据的值来定位绘图的具体位置，更改Paint画笔来实现图表的多元样式。

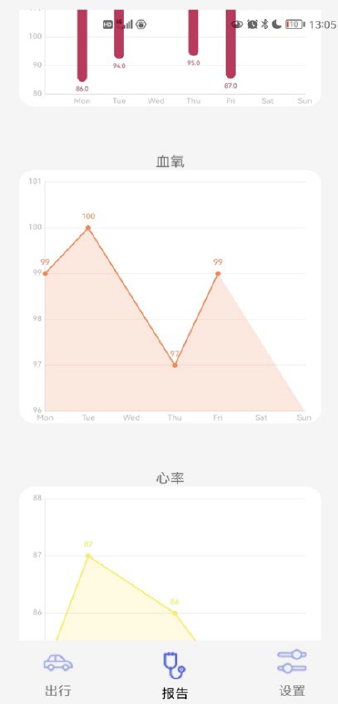
 

图 3‑3  图 3‑4

SQLite建立数据库与表的语句可以直接使用SQL。目前，我们在APP本地数据库保存司机的姓名、id、线路、联系方式等信息。

# 案例

这里从司机与管理员两个方面举例说明本项目的应用效果，由于篇幅有限，只介绍主要功能。

## 司机

司机上车后，穿戴好传感器设备（见图 4‑1），在Android平板登录APP。传感器持续采样司机体征数据，并拍摄司机照片，通过UPD发送给APP，APP再定时将司机的心率、体温、血压、血氧等指标以及图片传送到服务端。

当司机抽烟时，服务器会判定此为异常情况，APP向司机发出语音提示。



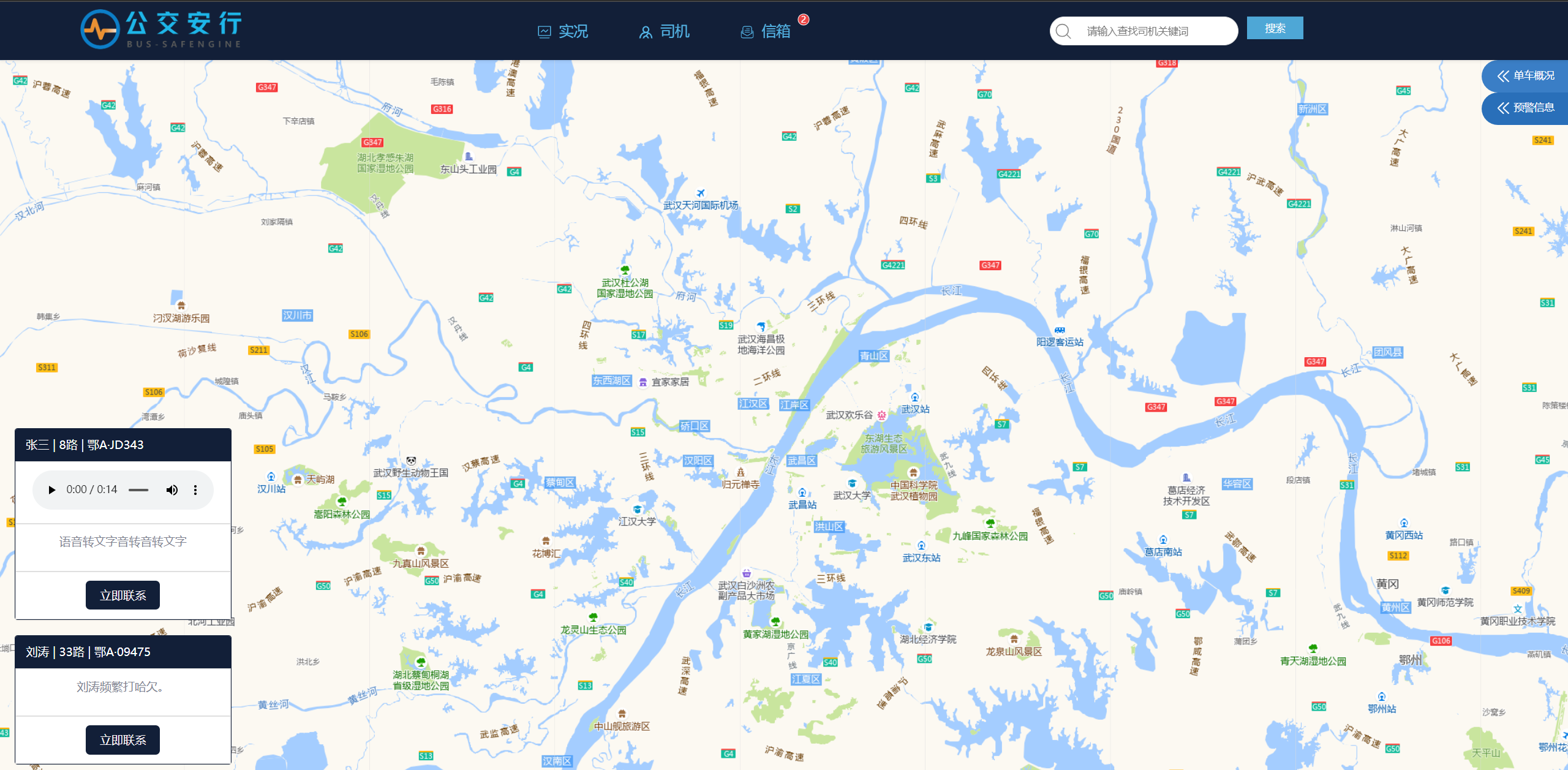
图 4‑1 装有传感器的方向盘

## 管理

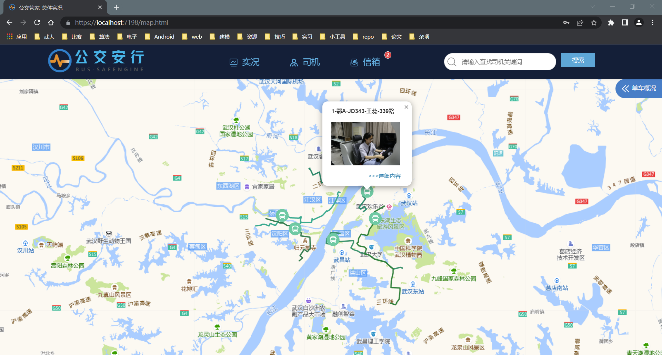
管理员可以在web端登录，浏览本市所有公交线路的运行情况。当服务器判定司机体征低危异常（比如心率略微偏高），管理员可以在主页面中左下角看到异常提示，并可以与司机联系；当服务器判定出现高维异常（比如与乘客出现肢体冲突），管理员可以在主页面中心到橘红色的警报信息，并可以与司机联系。

管理员可以在“司机”页面，看到单独司机的实时图像、生理指标，以及对司机一段时间依赖的状态分析建议。

管理员可以在“邮箱”页面，查看所有已读、未读的司机留言录音，以及相应司机的照片。







# 结论

这次的c#大作业是我首次参与开发一个结构完整、规模足够庞大的项目。我在其中主要担任了前端web界面的开发工作，参与了后端c# Service层的编写与Android APP的图表和数据库的开发。

在此次开发中，我确实遇到了诸多的不顺。我们小组曾因为git使用频频出现confilct而有过激烈的争吵，开发进度一度停滞；常常在深夜，当室友都进入梦乡，我们三人却坐在电脑前，绝望而疲劳地排查bug；也曾因为交流功能需求时，我的理解出现偏差，导致我浪费了2天去写一个不需要的功能；很难忘记在畅享出美好的功能全景后，我面对互联网上庞杂的资料，却不知如何入门的茫然挫败。

但是这次完整项目的开发，让我学会了很多新东西。从技术上来说，我新学习了前端JavaScript的编写、很多框架和web API的使用。出了这次经历也我终于了解了前后端之间的耦合。以前我只会调用后台提供的API接口，并不知道服务器接口内部是如何实现的，但是这次我实际参与了后端开发，这次课程设计让我对web开发有了更深的理解。

此外，我更加熟悉了git的使用。我们在项目开发之初就确立了全面使用Github进行代码托管的策略。git真好用！我个人认为它对团队开发最有用的功能是可以回溯之前版本的代码。这样，有时候合并出了bug时，可以方便地回退到之前的版本，便于细致排查错误。我们三个人都开了自己的分支，只在自己的分支上面commit，这样可以区分出不同的人的修改，便于排查bug。

最后，我想感谢贾向阳老师。贾向阳老师上课时会对针对每个同学的代码进行逐行的点评，让我获益匪浅。我非常难忘的是，第一次上课（当时是这学期的第二次课）时，贾老师站在我身旁让我把逻辑代码与控制台打印代码分离，当时我第一次有了“软件结构设计”的粗浅观念。这门课“翻转课堂”的形式也非常新颖，经过课前与课上两次的学习，让我对知识掌握更扎实。其实我觉得这个形式的最大意义是老师可以在课上现场指点我们的代码。从前几次课的时候，老师一站在我身旁，我就会紧张地无法思考，到后来逐渐适应，可以比较自如地与老师交谈，我有了很大的改变。非常感谢老师对我的各种问题的及时回复，让我感受到非常的温暖，那次连接MySQL始终连接不上，我都要放弃了，但老师却还在群里为我们支援，我真的特别感动。

**教师评语评分**

评语：

评分：

评阅人： 年 月 日

（备注：对该实验报告给予优点和不足的评价，并给出百分制评分。）