公开论文

涉密论文



http://www.nit.net.cn/__local/C/EE/E5/AEF522EFB83938A8BD165923DEC_001BEC5E_9976.png

**毕业论文（设计）**



|  |  |
| --- | --- |
| 题 目 | 基于高德地图的道路病害导航APP |
| 姓 名 | 李英豪 |
| 学 号 | 3190439021 |
| 专业班级 | 数据科学与大数据技术191 |
| 指导老师 | 舒振宇 |
| 学 院 | 计算机与数据工程学院 |
| 日 期 |  |

浙大宁波理工学院本科毕业论文（设计）承诺书

1.本人郑重承诺所呈交的毕业论文（设计），是在指导教师的指导下严格按照学校和学院有关规定完成的。

2.本人在毕业论文(设计)中引用他人的观点和参考资料均加以注释和说明。

3.与我一同工作过的同学对本研究所做的任何贡献均已在论文中做了明确的说明并表示谢意。

4.本人承诺在毕业论文（设计）工作过程中没有抄袭他人研究成果和伪造数据等行为。

5.若本人在毕业论文（设计）中有任何侵犯知识产权的行为，由本人承担相应的法律责任。

6.本人完全了解浙大宁波理工学院有权保留并向有关部门或机构送交本论文（设计）复印件和电子文档，允许本论文（设计）被查阅和借阅。本人授权浙大宁波理工学院可以将本论文（设计）的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或者扫描等复制手段保存和汇编本论文（设计）。

作者签名： 日期： 年 月 日

致谢

光阴荏苒，岁月如梭。四年的时光转瞬即逝，弹指一挥间落笔致谢。回首大学生涯，有喜有悲，转眼已不再是懵懵懂懂的新生，转眼已到了分别的时刻，转眼就要踏上新的征程。

此刻，思绪万千。首先我要特别感谢我的论文指导老师舒振宇老师，舒老师有着严谨的教学态度和丰富的学科知识，大学四年中对我有着无微不至的照顾。在我毕业论文的撰写中给予了悉心的指导，给我指明了前行的方向。善之本在教，教之本在师，感谢舒老师，也感谢我的所有任课老师！

其次，我要感谢我的同学们、我的两位室友，四年时光有你们陪伴真好！也要感谢一位十年之交的周姓男子，感谢所有给予我帮助的朋友们！世界很大，人海茫茫，愿友谊长存。

浅喜似苍狗，深爱如长风。感谢陈诗韵同学1623天的陪伴，人生的喜悦不多，但就像草木有阳光和水，风筝有清风和线，我有你的陪伴。

感谢我的父母二十多年来对我无微不至的照顾，感谢你们为我的成长付出的一切，愿椿萱并茂。同样的祝福也送给从小带我的奶奶，希望你寿比南山。

不积跬步无以至千里，漫漫长路，感谢自己的坚持，定前程似锦，不负韶华。

摘要（居中，黑体二号）

摘要是论文内容的简要陈述，应尽量反映论文的主要信息，包括研究目的、方法、成果和结论，不包含图表，不加注释，具有独立性和完整性。中文摘要一般不超过300字。

本文介绍了一款基于Android的道路病害管理系统，该系统旨在帮助路政管理部门更高效地收集、处理和管理道路病害信息。系统涵盖了病害数据的录入、检索、分析和可视化等功能，并通过高德地图组件来呈现病害数据分布情况进行数据可视化。数据管理方面，系统采用了SQLite数据库与Room库进行交互，并通过封装查询方法实现数据的增删改查操作，并提供事务支持保证数据的完整性和一致性。在系统测试方面，我们进行了功能测试和用户体验测试，并对测试结果进行了分析和总结。结果表明，该系统具有较好的稳定性和易用性，可以有效提升道路病害信息的收集和管理效率。

关键词：移动应用；安卓；道路病害管理；数据可视化

**Abstract**

This paper introduces an Android-based road disease management system, which aims to help road management departments to collect, process and manage road disease information more efficiently. The system covers the functions of disease data entry, retrieval, analysis, and visualization, and uses the Gaode map component to present the distribution of disease data for data visualization. In terms of data management, the system uses the SQLite database to interact with the Room library, and implements data addition, deletion, modification and query operations through encapsulation and query methods, and provides transaction support to ensure data integrity and consistency. In terms of system testing, we conducted functional testing and user experience testing, and analyzed and summarized the test results. The results show that the system has good stability and ease of use, and can effectively improve the collection and management efficiency of road disease information.

**Keyword：**Mobile application；Android；Road disease management；Data visualization

目录（居中，黑体二号）

[1 绪论**（左顶格，章节序号与标题之间空1字，宋体小四号，1.5倍行间距）** 1](#_Toc24017166)

[1.1 背景 1](#_Toc24017167)

[1.1.1 小节标题 1](#_Toc24017168)

[2 正文 3](#_Toc24017169)

[2.1 节标题 3](#_Toc24017170)

[2.1.1 小节标题 3](#_Toc24017171)

[3 论文总结 3](#_Toc24017172)

[4 参考文献 3](#_Toc24017173)

[附录 6](#_Toc24017174)

# 绪论

## 背景介绍

### 研究背景和意义

当今社会，道路建设和维护一直是一个重要的议题。随着经济的发展和城市化的进程，公路的数量和长度不断增加，得到了蓬勃发展。然而，由于自然和人为因素的影响，道路病害成为公路运输的一个重要问题，随之而来的是我们如何进行建后保养[1]。道路病害通常包括裂缝、坑洼、冲沟、路面不平整等，它们不仅会损坏道路，而且还会对车辆和行人的安全构成威胁。

目前来看，传统上通过人工的方式对道路病害进行检测和修复通常需要大量的人力、物力和财力投入，越来越难以适应公路的快速发展，尤其是对于一些较为隐蔽的病害，可能需要长时间的观察和检测才能发现。计算机技术的快速发展，尤其是图像处理的发展与应用，为我们对道路病害进行检测开辟了全新的途经，不再那么依赖人工的肉眼观察，研究者对此也有着极大的关注[2][3][4]。因此，找到一种能够快速、准确地检测道路病害的方法，具有重要的实际意义。

近年来，随着智能手机和移动互联网的普及，越来越多的人习惯使用手机处理各种事务，在当今这个信息时代，手机APP已成为人们生活中不可或缺的一部分，此外由于城市化的进程，人们对道路病害的管理变得更加复杂和繁琐。综上所述，开发一种能够实时提供道路病害信息的APP，对于保障道路安全和方便公众出行具有重要的现实意义。

本研究旨在设计和实现一种基于高德地图的道路病害检测APP，通过利用高德地图API获取道路信息和病害数据，结合机器学习等技术，实现道路病害的检测和识别，并为用户提供实时的道路病害信息。这将对道路安全和公众出行带来重要的实际应用价值。

### 智能手机平台现状分析

随着新势力的不断崛起，智能手机操作系统的发展可谓是百花齐放，这也促进了智能手机操作系统的不断进步。在当前移动计算技术发展中，研究与开发智能手机操作系统是极为重要的一个领域[1]，这一领域的竞争日趋激烈，各个厂商纷纷推出自己的操作系统，如苹果的iOS、谷歌的Android、微软的Windows Phone等[2]。这些操作系统都具备各自的特点和优势，以及一系列的应用程序和服务，为用户提供更为便捷的移动体验。随着移动互联网的不断发展和智能手机的普及，对于操作系统的要求也越来越高，如更快的响应速度、更流畅的用户界面、更加安全的数据保护等。因此，操作系统的研究与开发具有重要的意义和广阔的发展前景。

开发内嵌地图应用的软件相当困难，通常还需要支付高额的地图厂商版权费用。此外，由于手机的功能限制，基于当前位置开发功能的应用程序非常少见[3]。在此背景下，由运营商、手机制造商等各种企业组成的手机开放联盟诞生了，积极推动Android的开发工作[4]。Android是由Google开发的开源操作系统，其拥有众多优秀的特性，如多任务处理能力、广泛的设备支持、强大的通知系统和开放性的应用程序生态系统等，在Android平台上开发应用程序的开发人员使用Java语言编写应用程序，尽管Java如今面临着许多挑战，但由于其流行和广泛应用，以及不断的创新和改进，我们可以相信，Java的发展前景仍然是非常乐观的[5]。而我们开发出来的应用程序会在Dalvik虚拟机上运行，Dalvik虚拟机时类似于Java虚拟机的一个存在，正如前面所说，Android程序是由Java编写，但是Java编写后的程序需要经由转换得到.dex格式后才能在Dalvik虚拟机上运行，一个Android应用会由一个唯一Dalvik虚拟机与之对应[6]，同时Android底层采用开放性的Linux kernel作为其架构[7]。这种架构的优势在于，它不仅提供了强大的性能和稳定性，同时还为开发人员提供了广泛的自由度和灵活性[8]，使得他们可以创造出各种各样的应用程序，从而进一步推动了Android生态系统的发展。

### Android与iOS

值得一提的是，Android强有力的竞争对手苹果公司有着iOS，iOS的体系架构相对朴实无华，其系统核心、基础服务、应用框架和应用都是采用C/C++或Object-C开发。由于这种开发方式，对硬件的要求相对较低，因此它具有优异的运行效率和人机操作体验。然而苹果公司的技术创新模式具有典型的封闭式创新特征，众所周知，不论是Mac OS还是iOS，其开发升级的资源、软件生态都是苹果公司自家的。相比之下，Android的创新模式则更具开放性，各式各样的APP都可经由各式各样的开发商所提供，其系统可供所有移动设备厂商使用[9]。因而，本次研究是基于Android平台进行的。

### 主流地图现状分析

在介绍主流地图之前，我们需要了解一下什么是GPS，这是一种卫星导航系统，通过接受来自卫星的信号可以准确定位地球上的具体位置，最初是由美国政府开发应用的[10]，目前已经成为了全球范围内最常用的定位系统之一，经度已经达到了亚米级别。而正是GPS系统技术的飞速发展使得行业内手机导航系统产品的不断涌现，给人们提供：即时定位、线路导航、地图浏览、信息查询等功能。[11]而目前三大地图产品供应者分别是高德、百度、谷歌。在国内，高德地图与百度地图手中人群是高于Google Map的，尤其是高德地图，其应用渗透到了各行各业。同时高德地图在移动平台方面表现更出色，特别是其能够在Android、iOS两大主流移动平台上流畅使用，并且只有高德地图的API具有实时交通查询的功能[12]。综上所述，高德地图在一定方面更满足我的需求，故本项目基于高德地图API进行开发。

### 研究目标和内容

本文的研究内容主要包括以下几个方面：

高德地图API的应用研究：介绍高德地图API的基本功能和应用特点，探讨如何利用高德地图API构建道路病害导航APP。其中，需要详细探讨如何实现高德地图API的集成、地图显示、地图标注、地图搜索和导航等功能。

道路病害数据管理系统的设计与实现：分析道路病害数据的来源和特点，设计并实现一个数据管理系统，实现数据的采集、存储和处理功能，并通过本地数据库技术实现数据的可靠性和安全性。其中，需要详细介绍本地数据库的设计和实现，包括数据库表结构的设计、数据表之间的关系、数据的增删改查等功能的实现。

道路病害检测研究目标：利用计算机视觉技术对道路病害进行自动化检测，从而提高道路病害检测的效率和精度。

数据可视化设计与实现：基于高德地图API，实现道路病害数据的可视化展示，包括道路病害的类型、位置、程度等信息的展示，同时实现道路病害信息的查询和上传功能。其中，需要详细介绍如何利用高德地图API实现道路病害数据的可视化展示和查询功能，并实现数据上传功能。

系统性能评估与应用推广：对道路病害导航APP进行性能评估和测试，包括系统的稳定性、实时性和用户体验等方面，同时探讨道路病害导航APP的应用前景和推广策略。其中，需要详细介绍如何进行系统性能评估和测试，并根据测试结果对系统进行优化和改进，同时探讨道路病害导航APP的应用前景和推广策略。

通过以上研究内容的探讨，本文旨在提供一种基于高德地图API的道路病害导航APP解决方案。该方案可以有效地提高驾驶者对道路病害的认知和警惕，避免道路病害对行车安全和车辆损坏带来的潜在风险，并为提升道路交通安全和保障驾驶者权益做出一定的贡献。

### 研究方法和技术路线

根据该项目的具体情况，将采用以下研究方法：

调查研究法：对道路病害维修相关企业和政府部门进行深入访谈，了解他们在道路病害维修过程中所面临的问题，收集相关需求，以此为基础确定项目需求和功能。

文献研究法：研究现有的道路病害检测和识别技术，评估其在本项目中的可行性和应用前景，选择适合本项目的技术。

实践研究法：通过对自己相关病害接口、用户接口、处理接口进行单元测试，验证我接口的正确性、可用性，通过对所有Activity进行功能测试，验证我编写的页面的正确性，最后对整体应用进行运行测试，验证系统的性能和稳定性。同时，将道路病害数据导入系统中，测试其在道路病害管理和维修中的实际应用效果，收集反馈信息进行改进和优化。

本研究的技术路线主要包括以下几个方面：

高德地图API的应用研究：本文采用了高德地图API作为导航APP的地图展示和路线规划工具。在应用研究中，我主要学习了高德地图API的使用方法，包括地图AMap的添加与显示、标记Marker与数据的绑定与添加、路线规划、POI搜索等功能。

道路病害数据管理系统的设计与实现：本文设计了一个本地SQLite数据库，用于存储道路病害数据和用户信息以及处理状况，从而为我的应用程序提供数据支撑。在数据管理系统的设计和实现中，主要包括数据表结构设计、数据相关性、数据增删改查功能的实现、不同Activity之间的关联性以及高德地图模块的继承工作。

数据可视化设计与实现：本文采用了高德地图API的可视化功能，将道路病害数据与高德地图marker标记绑定，从而根据经纬度信息展示在地图上。在可视化设计和实现中，我们主要包括数据获取、数据标记、地图展示等功能。

APP系统性能评估与应用推广：在APP开发完成后，我进行了系统性能评估和应用推广。在性能评估中，主要测试系统不同Activity独立功能是否完备、数据是否，同时也测试系统整体的稳定性，是否会出现系统宕机异常，系统画面卡死，请求反馈过慢等问题。在应用推广中，通过各种渠道将APP推广给目标用户，同时采用问卷调查等方式了解用户的反馈和建议。

通过以上技术路线的实践和应用，我可以开发出一款基于高德地图的道路病害导航APP，并且在维持系统准确性和稳定性的基础上，提高App使用者的用户体验。

# 正文

## 需求分析

### 技术可行性

基于高德地图的道路病害导航app需要集成高德地图API，需要相应的算法，包括识别道路病害识别的算法和路径规划的算法[11]。目前，这些技术都已经相当成熟，并且在高德地图开放平台上有提供相应的技术支持，因此技术实现上是可行的。

### 市场可行性

道路病害导航app的市场潜力很大，因为路面损坏问题一直存在，而且车主对车辆的保养越来越重视[10]，故路面修复的重要性不言而喻。此外，道路病害导航app可以为政府部门提供路面维护的数据支持，政府可以通过使用app，获悉本市全部道路病害的具体分布情况，病害本身的各种信息，从而对病害有一个统筹全局的规划。综上，本款app市场需求很大，而且具有可持续性，能够长期投入使用。

### 经济可行性

道路病害导航app主要需要投入开发成本、服务器成本、市场推广成本等，但这些成本可以通过收费模式、广告等方式进行收回[9]。同时，高德地图已经建立了广泛的用户基础和商业合作伙伴，这为道路病害导航app的商业化运营提供了有利条件。

### 法律可行性

开发道路病害导航app需要遵守相关的法律法规，其中不乏包括数据隐私、地图版权等方面的规定。我能够严格遵守相关法规，从而保证法律可行性。

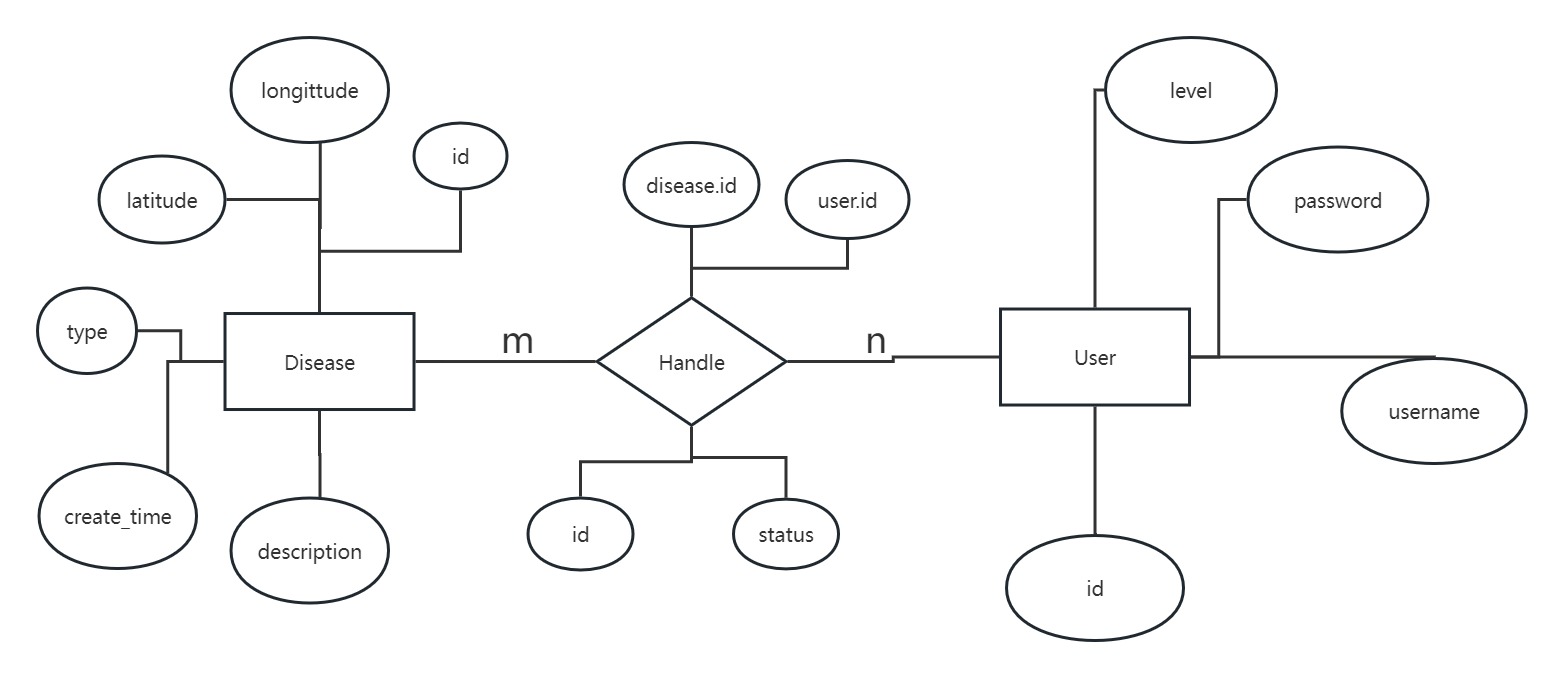
综上所述，基于高德地图的道路病害导航app在技术、市场、经济和法律等方面都具备可行性。

## 数据库设计

在本研究中，与传统的 Web 开发使用 MySQL 数据库不同，我在该应用中选择了 SQLite 作为后端数据库来存储道路病害信息。SQLite数据库不需要像MySQL或Oracle等数据库管理系统一样需要安装独立的服务器进程，同时由于开发app的原因，我需要一款占据更小内存、更快响应速度和更高稳定性的数据库来支撑我的app，而SQLite无需独立服务器进程和配置的方式，而是直接通过SQL语句操作数据库文件，十分符合我的需求。因此SQLite是我当然不让的选择。

### E-R图设计

为了方便我描述数据之间的关系，我在数据库系统设计阶段初采用E-R图的方式。制作E-R的过程能够帮助我梳理系统不同实体之间的关系，最后的E-R能够对整个系统的结构进行直观呈现，有助于理解数据模型。本数据库的E-R图设计如下：



* 1. E-R图

### 逻辑设计

为了便于我后期数据库的设计实现，通过逻辑设计能够对数据库表进行逻辑上的构建。数据库表之间存在的关系，包括主码和外码，在这一阶段将得到落实，逻辑设计如下：

Disease(id, longitude, latitude, type, create\_time, description)

User(id, username, password, level)

Handel(id, status, disease.id, user.id)

Disease的主码为id

User的主码为id

Handle的主码为id，外码是disease.id和user.id

### 范式

根据以上部分的探讨，可以看出本数据库设计满足第一范式、第二范式和第三范式。

在该设计中，每个表中的列都是原子值，不再有进一步可分解的属性，满足第一范式要求。此外每个表都有自己独特的主键，而相应非主键列完全依赖于自身主键，故满足第二范式要求。每个表中的非主键列都只和主键相关，在第二范式的基础上，不存在传递依赖，因而满足第三范式。

综上所述，本数据库符合数据库设计的基本要求。

### 病害表

病害表是我用于存储病害信息的表，主要用于实现对道路病害信息的检索分析，例如基于地图的可视化展示、数据分析筛选等功能。通常会与用户表、处理表等其他相关表一起被使用，用于实现综合的道路病害管理系统。具体的：我创建一个名为“RoadDisease”的表，用来存储道路病害的具体信息。这个表设置“id”字段用于标识每条道路病害的信息，“longitude”、“latitude”字段分别用于记录道路病害发生的经纬度，“type”字段用于记录道路病害的类型，如：路面坑洞、路面裂缝、桥头跳车等问题，“description”字段用于自定义对病害的一些描述信息，可由用户任意输入，“create\_time”字段描述病害的创建时间。表结构大体如下：

* 1. 病害表

|  |  |
| --- | --- |
| RoadDisease | |
| 列名 | 数据类型 |
| id | INTEGER |
| longitude | REAL |
| latitude | REAL |
| type | INTEGER |
| description | TEXT |
| create\_time | DATETIME |
| place | TEXT |

### 用户表

用户表是十分重要的表，用于验证用户身份、限制用户权限。我设计的User包括id字段描述唯一表示、username字段描述用户名、password字段描述密码、level字段描述用户类型，用户类型可以分为0：管理员，1：普通用户。例如管理员可以查看所有病害数据并进行编辑和删除操作，而普通用户只能编辑自己所负责的数据，而不能对非属于自己的数据进行操作。用户表还可以与其他表进行关联，例如Handle表中的userId字段就是通过与用户表的id字段进行关联得到的。

* 1. 用户表

|  |  |
| --- | --- |
| User | |
| id | INTEGER |
| username | TEXT |
| password | TEXT |
| level | INTEGER |

### 修复情况表

Handle表的设计目的是记录某个用户负责对某一条病害记录进行处理的情况，因此，表中记录的每一行都对应着一个用户对一条病害记录的处理记录。用户可以对同一条病害记录进行多次处理

Handle表中的外键user\_id和record\_id分别与User表和Record表建立关联，以实现表之间的关联查询和数据完整性约束。Status字段用于表示这条处理记录的状态，取值只要有3个：未分配、处理中、已处理。

* 1. 处理表

|  |  |
| --- | --- |
| Handle | |
| id | INTEGER |
| user\_id | INTEGER |
| disease\_id | INTEGER |
| status | INTEGER |

## 系统实现

### 相关技术和工具

高德地图API：该API主要提供地图添加与显示、POI搜索、导航规划等功能，我们将利用它来展示道路病害信息以及提供导航服务。

Android Studio：我使用它来编写本项目，这款应用智能提示补全功能强大，且具有安卓开发所需要的大量工具，内置模拟器，自动生成打包apk。

Java语言：本项目主要使用Java语言开发，辅之以一些xml技术用作布局文件等相关配置文件。

SQLite数据库：SQLite是一款轻量级的嵌入式关系型数据库，能够运行在手机上且兼具性能和体积，同时功能完备[13]，我将使用它来存储本项目中的病害信息以及相关的用户和处理记录信息。

Git版本控制：我将使用广为人知的git工具来管理项目代码的版本，保证代码的稳定性和可维护性。

Gradle：我主要通过Gradle构建工具来构建、编译、和部署我的项目。

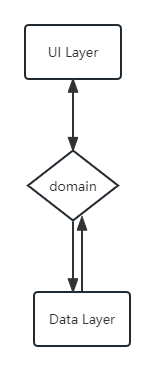
Room：一种Android平台提供的SQLite对象映射库，主要是对SQLite数据库的进一步封装，使得使用者不用编写大量繁琐冗杂的sql语句，而是通过注解的方式自动生成相应的工作，可以帮助开发者快速构建并与SQLite数据库进行交互，使得应用程序的数据库操作更加高效和可靠。

### 数据管理系统设计与实现

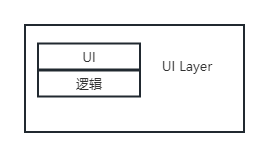
当今的数据管理系统大多采用客户端-服务器（Client-Server）架构[14]或者分布式（Distributed）架构。在分布式架构中，系统的各个组成部分分布在不同的计算节点上，通过网络通信进行协同工作。在客户端-服务器架构中，客户端和服务器之间通过网络通信，客户端提供用户界面和一些应用逻辑，服务器则负责处理数据请求和管理数据库。例如，在一个在线购物网站上，客户端是指用户在自己的计算机上运行的Web浏览器，而服务器是指运行在网站后台的应用程序，它处理用户的订单和付款请求，并将相关信息存储在数据库中。客户端和服务器之间通过Internet连接通信，以实现用户与网站的交互。

在本毕业论文所涉及的数据管理系统中，采用了客户端-服务器架构，主要组成部分包括前端界面、后端数据库和数据访问层。前端界面是用户与系统交互的主要方式，它负责收集用户输入，展示数据查询结果，以及向用户呈现一些辅助信息。后端数据库是整个系统的数据存储核心，它负责存储、管理和维护系统所需的各种数据。数据访问层则是前端界面和后端数据库之间的中间层，它负责处理前端界面的数据请求，同时封装了与数据库交互的底层细节，以提供更加简洁和高效的数据访问接口。

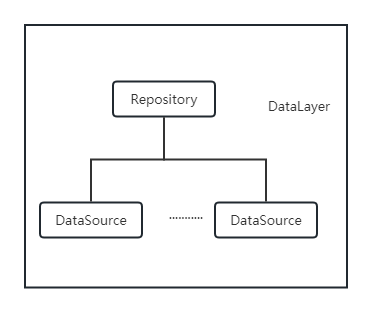
为了实现数据管理系统的功能，需要使用到多种技术和工具。在本毕业论文中，我们采用了Android开发框架、高德地图API、SQLite数据库和Room库等技术和工具。其中，Android开发框架提供了基础的应用程序开发环境和相关的组件，使我们能够更加方便地构建移动应用程序。高德地图API提供了强大的地图服务，使我们能够在应用程序中集成地图显示、搜索和导航等功能。SQLite数据库是一款轻量级的嵌入式数据库，适合于移动设备等资源受限环境下的数据存储和管理。Room库则是一款为SQLite数据库提供了更加便捷的数据访问接口和对象关系映射（ORM）功能的库。



* 1. 总应用架构



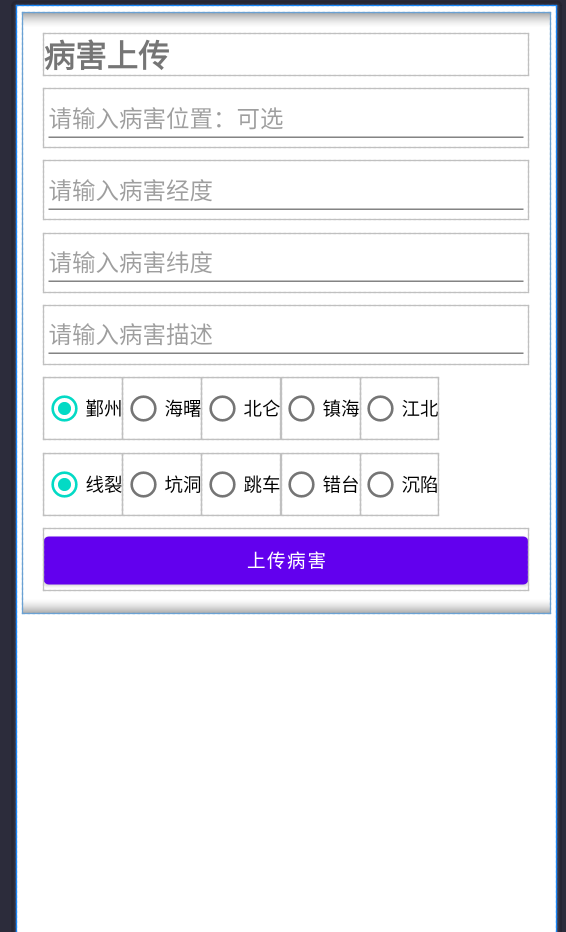
* 1. UI Layer



* 1. Data Layer

病害数据的管理和维护主要通过前后端的交互完成，用户可以通过APP的添加病害信息功能来添加新的病害记录，该功能实现了数据的输入验证和格式化，保证了数据的准确性和一致性。

为此我专门制作了一个病害上传的Activity，在其布局文件中放置了要给TextView用户展示标题，4个EditText用户获得用户的输入信息，两组RadioGroup用于实现单选，一个Button用于实现病害封装上传。



* 1. 病害上传UI

通过编写validateInput方法，用于验证用户输入的数据是否合法。该方法通过获取UploadActivity中的两个EditText输入框，判断两者的内容是否存在Null的情况，如果存在Null，则意味着用户的输入非法，则需要重新进行输入。

public boolean validateInput()  
{  
 String s = mEtLatitude.getText().toString()**;** String s1 = mEtLongitude.getText().toString()**;** if (s.isEmpty() || s1.isEmpty())  
 return false**;** return true**;**}

当用户需要修改或删除已有的病害记录时，可以通过APP的修改病害信息和删除病害信息功能来完成操作，这两个功能同样实现了数据的验证和格式化，以避免用户误操作导致的数据错误。

通过给Button按钮添加点击事件监听，在监听中，对用户输入的数据进行封装，制作成病害信息，并使用DiseaseDao. insertDisease(disease)方法上传信息至数据库。核心部分代码如下：

mBtnGo.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (!validateInput())  
 {  
 ToastUtil.*showMsg*(UploadActivity.this**,** "请填写经纬度")**;** return**;** }  
 makeData()**;** Long aLong = db.diseaseDao().insertDisease(disease)**;**

**}**

其中DiseaseDao的insertDisease方法用于将一个或多个Disease对象插入到与之关联的SQLite数据库中。具体代码如下：

@Insert  
Long[] insertDiseases(Disease... diseases)**;**

为了能够实时响应用户的选择请求并做出相应的应答，我们需要在应用中添加相应的筛选条件和查询功能。用户可以根据道路病害的位置、类型、状态等条件进行筛选，并且可以根据时间范围来查询道路病害信息。在进行查询时，我们需要从数据库中读取数据并根据用户的筛选条件进行过滤和排序，最终将符合条件的病害信息展示给用户。通过这种方式，我们可以帮助用户快速、准确地找到所需要的信息。

UI界面方面，通过置放spinner控件实现对实现下拉选择框，便于用户条件的筛选，独立编写一个日期选择的fragment，用于进行日期选择。用RecycleView控件来实现数据列表的展示。



* 1. 病害查询（修复）



* 1. 病害查询（未分配）

实现数据的筛选功能核心在于获取用户选择的栏目进行相应的条件判断，根据用户所选进行相应的SQL语句编写。核心代码如下：

switch (parent.getTag().toString())  
{  
 case "place":  
 disease.setPlace(parent.getItemAtPosition(position).toString())**;** break**;** case "type":  
 int tmp = 0**;** tmp = MyUtil.*ConvertType\_toInt*(parent.getItemAtPosition(position).toString())**;** disease.setType(tmp)**;** break**;** case "status":  
 int status = 0**;** status = MyUtil.*convertStatus\_toInt*(parent.getItemAtPosition(position).toString())**;** disease.setStatus(status)**;** break**;**}

其中由于呈现在用户面前的都是字符串类型的数据，而数据库中一部分数据是其他类型的，因而我自行编写了工具类MyUtil来实现数据格式的处理与转换。

为了做到实时响应用户的选择请求，从而做出应答，我在下拉列表状态发生变化的时候就进行数据的更新提交操作，将数据更新到recycleView中，从而使用户看到的始终是最新的消息。

List<DiseaseReturnPojo> data = diseaseDao.getDiseaseByConditional(sqLiteQuery)**;**

**…**

setRecycleView(recycleView)**;**

设置RecycleView核心的就是实现一个属于自己的Adapter，这里采用线性布局，用于绘制我所需要的数据。Adapter中相关核心代码如下：

@Override  
public int getItemCount() {  
 if (list.size() == 0)  
 ToastUtil.*showMsg*(mContext**,** "没有数据")**;** return list.size()**;**}

绘制数据前，首先会调用这个方法得到需要绘制的行数，根据行数去绘制数据，具体方法参照如下：

mHolder.getPlace().setText(disease.getPlace())**;**

逐步以此方式给Holder的属性绑定我们需要的数据，就能够实现我们的Adapter了。

此外，为了保证数据的完整性和一致性，系统还提供了事务支持。通过在方法中添加@Transaction注解，Room库会将整个方法作为一个事务进行处理，从而确保操作的原子性。这样，当一个操作失败时，整个事务都会被回滚，以保证数据的完整性和一致性。

在本系统中，管理员和普通用户具有不同的权限和操作范围。管理员可以对所有的病害信息进行添加、删除、修改和查询操作，而普通用户只能对自己负责的病害信息进行操作。

为了实现这一功能，本系统使用了基于角色的访问控制[15]（Role-Based Access Control，RBAC）模型。通过该模型，管理员和普通用户被赋予不同的角色，不同的角色拥有不同的权限，从而达到权限控制的目的。具体是通过分配权限id来控制用户对病害信息的操作。系统根据用户的权限id来判断是否能对数据进行相应的增删改查操作。

以用户对自己负责的病害数据管理为例子，假设一名用户分配到多个病害数据，则该用户处理完病害后应能够将病害信息就行修改，标记为处理完毕，并将数据更新到数据库中去。具体实现我们可以通过以下方式来实现：获取当前用户的id和当前待处理对象的id，发起请求更新数据库handle表中记录的status状态。核心代码如下：

private void setDiseaseTarget(int id)  
{  
 HandleDao handleDao = db.handleDao()**;** handleDao.updateDiseasesStatus(id**,** 1)**;**}

@Query("update Handle set status = :dStatus where disease\_id = :dId")  
int updateDiseasesStatus(int dId**,** int dStatus)**;**

其中获取病害id可以通过点击事件，用户点击病害的同时，获取到病害id传回给主活动。

setDiseaseTarget(id)**;**

而获取用户id，则可以通过MeFragment调用主Activity方法获取用户的方式来得到。

private int getUserId()  
{  
 MainActivity activity = (MainActivity)getActivity()**;** return activity.getUserId()**;**}

同样，为了及时能够响应用户请求，需要在用户点击按钮后即时刷新页面，获取最新数据。

public void refresh()  
{  
 FragmentTransaction transaction = getParentFragmentManager().beginTransaction()**;** transaction.replace(R.id.*fragment\_container\_view***,** new MeFragment())**;** transaction.commit()**;**}

此外，由于用户可能会有误操作的问题，亦或病害处理不全面需要反工，故系统提供了标记未处理的功能，代码逻辑同上。用户修改板块效果如下：



* 1. 用户界面

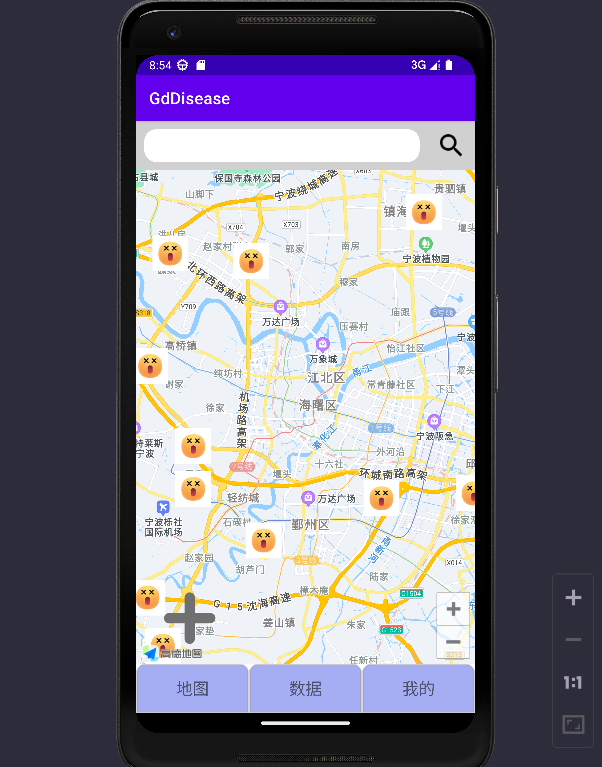
为保证数据的安全性与可靠性，我采用定期备份，导出数据库的方式对数据进行冗余存储，故当发生故障的时候可以通过冗余备份的数据进行数据的恢复。此外，我采用定期开展数据库维护工作的方式，包括优化查询语句、清理垃圾数据、更新数据库版本等，以保证系统的稳定性和可靠性。

### 数据可视化设计与实现

数据可视化是将数据以图形、图表、图像等可视化的形式呈现出来，以便用户更加直观地理解和分析数据。其主要目的是将复杂的数据变得更加易于理解和使用，提高数据的可读性和可视化效果，帮助用户更加深入地了解数据背后的信息和趋势，从而更好地进行数据分析和决策。

一般我们进行数据可视化，主要需要遵循的原则包括简单明了、数据精确、可交互性、美观、创新性。不同的数据可视化技术适用于不同的场景和需求，根据实际情况选择合适的技术可以帮助我们更好地展现数据的特征和规律。当涉及到地理信息的可视化时，使用地图组件能够提供更直观的展示方式，而高德地图提供的地图组件不仅提供了基本的地图显示功能，还提供了丰富的交互和控制选项，例如缩放、平移、标记、信息窗口等，这些功能可以让用户更方便地了解和操作地图上的信息。此外，高德地图还给我们提供了丰富的API接口，可以极大的满足不同场景下的定制化需求。

在我这款app当中，主要采用的就是地图可视化组件的方式。高德地图对外提供了添加标记点的功能，且自定义强度很高，我需要通过将数据库中的病害数据和标记点进行绑定，从而添加标记点至地图组件上，实现对病害数据的分布情况可视化呈现。



* 1. 地图与标记

这种可视化方式可以更加直观地展示病害的分布情况，帮助用户快速了解病害的地理位置信息和数量分布，具体实现如下：

mapView = (MapView) view.findViewById(R.id.*map*)**;**

mapView.onCreate(savedInstanceState)**;**

if (aMap == null) {  
 aMap = mapView.getMap()**;**}

这部分代码调用了高德地图给我们提供的sdk工具包，实现了创建地图，将地图添加到我们自定义的布局当中去。目标位置如下：

<com.amap.api.maps.MapView  
  
 android:id="@+id/map"  
  
 android:layout\_width="match\_parent"  
  
 android:layout\_height="match\_parent"/>

使用地图前需要根据高德地图的官方手册进行工具包的下载、配置文件的相关配置等。

SDK的下载可以通过高德开放平台Android SDK手册中的相关下载页面进行下载。根据需求的不同，分为了3D地图、2D地图、定位服务、搜索服务等工具，对于本项目，主要下载3D地图和定位服务即可。



* 1. 高德资源下载

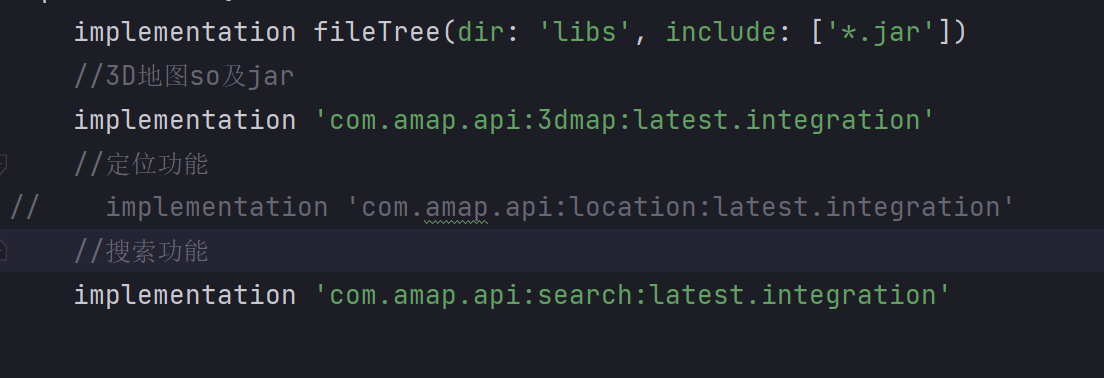
下载完资源后，要去高德控制台进行Key配置，不配置Key是没法使用高德地图相关API的。配置key的时候会要求我们获取SHA1，可以通过执行命令得到

keytool -v -list -keystore keystore



* 1. Key配置

引入高德地图SDK也可以采用Gradle引入的方式，这需要我们在Gradle的build文件中加入以下代码：



* 1. Gradle引入高德

代码中定位功能引入可能会出现冲突，如若出现可以不引入定位功能的包。

高德地图需要获取一部分用户权限，为了确保高德地图的正常使用，需要在Manifest文件中加入授权代码：



* 1. 授权

最后，在正式使用地图之前，需要开启合规性检查。

MapsInitializer.*updatePrivacyShow*(getContext()**,**true**,**true)**;**MapsInitializer.*updatePrivacyAgree*(getContext()**,**true)**;**

为了实现在地图上进行标记点的标注，我们需要获得数据库中的病害信息，对得到的病害信息进行遍历添加到地图上，核心代码如下：

public void drawMarker(String place)  
{  
 bindData(place)**;** for (Disease disease : list) {  
 addMarker(disease)**;** }  
}

获取数据的自定义函数bindData函数实现如下：

private void bindData(String place)  
{  
 if (place == "")  
 list = diseaseDao.getDiseaseAll()**;** else{  
 list = diseaseDao.getDiseaseByPlace(place)**;** }  
}

默认情况下，数据是不经筛选得到的全体病害数据，但是这边提供了一个地区筛选的选择，可以设置病害所在地区得到该区域的所有病害来实现一个在地图上显示特定区域病害的功能。两个查询的功能实现如下：

@Query("select \* from disease")  
List<Disease> getDiseaseAll()**;**@Query("select \* from disease where place = :place")  
List<Disease> getDiseaseByPlace(String place)**;**

AddMarker函数实现的逻辑在于：往地图添加病害所需要的最重要的数据就是经纬度，有了经纬度便可以定位到所添加的位置。创建经纬度可以通过高德地图提供的类LatLng，对每一个遍历出来的disease对象，创建一个LatLng类，设置其经纬度为disease对象的经纬度。高德地图组件良好的交互性和自定义也为用户提供了更加丰富的数据展示功能，用户可以自定义消息面板来实现用户对标记的点击事件，轻松查看病害详细信息的同时也可以利用地图的缩放、拖动等操作对数据进行交互式探索和分析，帮助用户更好地理解数据，发现数据中隐藏的模式和趋势，从而更好地做出决策。

public void addMarker(Disease disease)  
{  
 double lat = disease.getLatitude()**;** double log = disease.getLongitude()**;** LatLng latLng = new LatLng(lat**,** log)**;** MarkerOptions markerOption = new MarkerOptions()**;** markerOption.position(latLng)**;** StringBuilder snippet = new StringBuilder()**;** snippet.append(disease.getId()).append(": 纬度-- " + disease.getLatitude())  
 .append(", 经度-- " + disease.getLongitude())**;** markerOption.title("编号").snippet(snippet.toString())**;** markerOption.draggable(true)**;**//设置Marker可拖动  
 markerOption.icon(BitmapDescriptorFactory.*fromBitmap*(BitmapFactory  
 .*decodeResource*(getResources()**,**R.drawable.*sleepy\_big*)))**;** // 将Marker设置为贴地显示，可以双指下拉地图查看效果  
 markerOption.setFlat(true)**;**//设置marker平贴地图效果  
 aMap.addMarker(markerOption)**;**}

## 系统测试

### 功能测试

功能测试部分旨在对系统的功能进行测试，包括病害信息管理、数据可视化、用户权限管理、数据备份与恢复、数据检索与上传等功能。通过模拟用户实际使用场景，检测系统在各种情况下是否能正常运行并满足用户需求。

本系统主要采用黑盒测试的方法进行测试：

病害信息管理功能测试：测试系统是否能正常添加、删除、修改和查询病害记录，以及是否能够正确处理各种数据异常的情况。

数据可视化功能测试：测试系统在地图上是否能够正确显示病害数据的分布情况，并对不同的病害类型进行分类和标记，同时检测系统在数据量较大时是否能保持稳定性和流畅性。

用户权限管理功能测试：测试系统是否能正确区分管理员和普通用户，做到对不同身份的用于授予不同程度的权限，同时检测系统在用户登录和注销等操作时的安全性和稳定性。

数据备份与恢复功能测试：测试系统是否能够正确备份和恢复数据，并保证数据的安全性和完整性，同时检测系统在数据备份和恢复过程中的稳定性和可靠性。

数据检索和上传功能测试：验证系统是否能够按照用户输入的筛选条件返回正确的结果，同时还需要测试系统在高并发情况下的查询性能和响应速度。上传功能则是测试数据在上传后，数据是否在数据库中以正确的的方式存在。

测试过程中，我进行了一系列测试，包括正常操作、非法操作等情况，同时结合用户需求和使用习惯，对系统进行了全面测试。测试结果表明，系统的各项功能均能够正常运行，满足用户需求。

### 用户体验测试

用户体验测试是为了验证系统在使用过程中用户的感受和反馈，通过用户调查和用户行为记录等方法，了解用户对系统界面设计、操作流程、交互体验等方面的感受和反馈，以及发现和解决用户在使用过程中遇到的问题和困难。

本研究在功能测试的基础上，进行了用户体验测试，主要包括问卷调查和用户行为记录等方法。通过问卷调查，我们了解了用户对系统界面设计、操作流程、交互体验等方面的满意度和不足之处，并根据反馈意见进行了相应的改进。同时，我们还记录了用户在使用过程中的行为，分析用户的操作习惯和使用情况，发现和解决了一些潜在的问题和困难，提高了系统的易用性和用户满意度。

通过功能测试和用户体验测试，本研究保证了系统的正确性、稳定性、性能和安全性，同时提高了系统的易用性和用户满意度，为系统的推广和应用奠定了良好的基础。

# 论文总结

在本次论文中，我设计和实现了一个基于高德地图的道路病害导航APP。该系统能够实现病害数据的录入、存储、查询、统计和可视化展示，并提供了数据备份和恢复、用户权限管理、数据检索和上传等功能。

在实现数据可视化方面，我采用了地图可视化组件的方式，通过高德地图组件来呈现病害数据分布情况进行数据可视化。通过将病害数据和标记点进行绑定，实现了病害数据在地图上的展示和交互。

在系统测试方面，我采用了功能测试和用户体验测试以保证系统的质量和可靠性。其中，功能测试主要测试系统各个功能是否符合需求，用户体验测试主要测试用户对系统的使用体验。通过这些测试，我发现系统的各个功能都得到了有效的验证和检验，用户体验也得到了较好的提升。

总结，我认为本次研究实现了一个功能完备、性能稳定、安全可靠、易于使用的道路病害管理系统。通过本次研究，我也深入理解了数据系统的设计和实现，熟悉了数据可视化技术的应用和实现，同时也对系统测试和用户体验测试等方面有了更深入的认识和体会。

# 参考文献

1. 李惠,丁革建.智能手机操作系统概述[J].电脑与电信,2009,No.157(03):67-68.
2. 胡彬文.浅析目前主流智能手机操作系统的现状与发展[J].电脑知识与技术,2015,11(15):154-155.DOI:10.14004/j.cnki.ckt.2015.1445.
3. 舒贤华. 基于Android平台的手机Web地图服务设计[D].大连海事大学,2009.
4. 刘亚丽.Android软件前景浅析[J].科技创新导报,2011,No.205(25):32.DOI:10.16660/j.cnki.1674-098x.2011.25.173.
5. 范凯.Java技术[J].程序员,2007(01):13.
6. 李晓亮. 基于Android的手机导航系统设计与实现[D].江西理工大学,2014.
7. 张仕成.基于Google Android平台的应用程序开发与研究[J].电脑知识与技术,2009,5(28):7959-7962.
8. Love R. Linux Kernel Development: Linux Kernel Development \_p3[M]. Pearson Education, 2010.
9. 马诚泽. Android与iOS的对比分析及Android安全技术的设计与实现[D].山东理工大学,2013.
10. 刘美生.全球定位系统及其应用综述(三)——GPS的应用[J].中国测试技术,2007,No.150(01):5-11.
11. 邵志举,崔光辉.手机导航系统研究与实例[J].计算机仿真,2008(03):142-144+170.
12. 向玉云,高爽,陈云红等.百度、高德及Google地图API比较研究[J].软件导刊,2017,16(09):19-21+25.
13. 万玛宁,关永,韩相军.嵌入式数据库典型技术SQLite和Berkeley DB的研究[J].微计算机信息,2006(02):91-93+272.
14. Schussel G. Client/server past, present, and future[J]. Formerly Available WWW< URL: http://news. dci. com/geos/dbsejava. htm, 1995.
15. Ferraiolo D, Cugini J, Kuhn D R. Role-based access control (RBAC): Features and motivations[C]//Proceedings of 11th annual computer security application conference. 1995: 241-48.

# 附录（居中，黑体三号）

大幅面的图纸单独装订，并在附录中说明。

某些重要的原始数据、数学推导、计算程序、框图、构造图、零件图、装配图等要在附录中出现（若原始数据较多要装订成册同论文一起提交）。所附图纸须准确、规范，符合技术要求。

附录中的图、表、公式、参考文献等另行编排序号，与正文分开，也一律用阿拉伯数字编号，但在数码前冠以附录序码。例如：图A.1等。

**正文汉字字体、字号：宋体小四号。**

**外文、数字等符号：Times New Roman 字体，字号与同行汉字相同。**

**行间距1.5倍行距，首行缩进2个字符**