

浙江大学

宁波理工学院

毕业设计（论文）



题 目 药材设施栽培远程灌溉控制系统安卓手机客户端软件设计

姓 名 谢智勇

学 号 3120403028

专业班级 12 通信工程 1 班

指导教师 蔡卫明

分 院 信息工程与工程分院

完成日期 2016 年 5 月 2 日

目录

摘 要.....	3
Abstract	4
第 1 章 概述.....	5
1.1 选题背景及研究意义.....	5
1.2 国内外研究现状.....	6
1.2.1 Android 开发平台技术研究现状.....	6
1.2.2 数据库技术国内外研究现状.....	8
1.3 研究内容及组织结构.....	9
1.4 本章小结.....	10
第 2 章 系统总体设计.....	11
2.1 系统需求分析.....	11
2.2 系统总体设计.....	12
2.2.1 Java 语言.....	12
2.2.2 Java 串口通信.....	12
2.2.3 MySQL 数据库.....	13
2.2.4 内网解析映射外网.....	13
2.2.5 Android 应用程序设计.....	13
2.2.6 系统的总体结构.....	15
2.3 本章小结.....	16
第 3 章 数据获取的设计与实现.....	17
3.1 数据的收集与存储设计与实现.....	17
3.1.1 串口通信程序编写.....	18
3.1.2 JDBC 数据存储程序编写.....	22
3.1.3 MySQL 数据库的搭建.....	23
3.2 内部服务器至外部网络映射.....	25
3.2.1 Php 网页的编写.....	25
3.2.2 Nat123 的映射.....	26
3.3 本章小结.....	28
第 4 章 Android 应用的设计与实现.....	29
4.1 设计总述.....	29

4.2	开发工具介绍.....	29
4.2.1	Android studio 编译器.....	29
4.2.2	Genymotion 模拟器.....	30
4.3	应用程序设计.....	32
4.3.1	登录注册页面.....	33
4.3.2	数据显示页面.....	38
4.3.3	折线图显示页面.....	44
4.3.4	记录查询页面.....	48
4.4	本章小结.....	50
第 5 章	总结与展望.....	51
5.1	总结.....	51
5.2	展望.....	51
参考文献	52
致 谢	54

摘 要

现如今淡水资源短缺已经是全世界面临的共同难题，国内也是如此，而灌溉用水却因技术实现原因造成大量浪费。本次设计以阳春砂的栽培为背景，采用 Android 手机客户端对 Zigbee 传感网络收集数据进行监控观察。第一部分使用 eclipse 进行串口通信部分的开发，收集并处理来自传感模块的数据，其次第二部分是安卓手机客户端的设计。第一部分采用了 Java 语言的串口通信类库，收集串口数据并使用 JDBC 将数据存储至 MySQL 数据库，使用 php 编写网页读取数据库，之后映射该网页至外网。

第二部分为 android 手机客户端 app 设计，通过 HttpURLConnection 类对之前的网络数据进行读取，之后对数据进行处理，再显示至客户端页面。第一时间对温湿度变化进行监控。

本设计针对阳春砂这一药用植物进行远程灌溉设施的搭建，采用 android 操作系统平台的进行应用设计，其市场范围广泛，开源，是现今使用最广泛的手机操作系统。对于传感模块的数据能够更加方便快捷的进行实时监测，易于推广使用。

关键词：Android，MySQL，Java，

Abstract

Nowadays, the world are facing a shortage of fresh water resources problems, China is also facing the problem too, the irrigation water because of technology causes a lot of waste. This design's background is planting YangChunSha .Use the Android application for Zigbee sensor network to collect data to monitor observation. The first part of the development using eclipse serial communication portion, collect and process data from the sensor module. The second part is the design of Android mobile client. The first part is a serial communication library of Java. Collected the serial data, and use JDBC to store data to MySQL database, than use Php write the web page to read the database, last, mapping this web page to network.

The second part is Android application development. This time use class of HttpURLConnection for read the web page data, and resolve the data, than display resolved data to application activity, real time to collect data to monitor observation.

The design of this medicinal plant for YangChunSha build remote irrigation facilities, and uses android operating system platform to application develop, Its wide range of markets, open source, and it is now the most widely used mobile operating system. For the sensor data can be more convenient and efficient real-time monitoring, easy to promote use.

Keywords: Android , MySQL , Java

第1章 概述

1.1 选题背景及研究意义

现如今淡水资源短缺已经是全世界面临的共同难题。农业灌溉所需的淡水量又非常巨大，加之我国的农业发展还处于传统的农业模式，水资源短缺已经成为制约农业发展的重要因素^[1]。

我国的淡水资源丰富，总量达到了 28000 亿 m^3 ，但人均水资源占有量只有 2300 m^3 ，只占世界人均水平 1/4，是世界上人均淡水资源最为缺乏的国家之一^[2]。此外，空间和水资源时空分布严重失衡，雨量东南多于西北，山区多，平原降水量少，一般由东南向西北减少。而我国是传统的农业大国，农业用水量巨大，占全国用水量的 70% 左右，其中又以农业灌溉为主。我国农业灌溉水的有效率仅为 30%~40%，而先进国家可达 70%~80%^[3]。这表明了我国节水农业技术集成各种应用水平还很低，与发达国家相比还有相当大的差距。这也使我们看到了节水农业在中国的巨大潜力和广阔的发展前景^[4]。

因此，必须大力发展节水灌溉技术，以应对日益凸显的水资源危机。节水灌溉是在作物能够正常生长的情况下，采用减少灌溉用水，能够使作物用最少的水资源而收获最大的效益^[5]。精细农业的发展就要求转变“灌溉是浇地”的传统观念，树立“灌溉是浇作物而不是浇地”的新理念，对作物进行精量灌溉^[6]。采用智能化的节水灌溉系统，既能节约农业用水同时提高作物收益，也能节省人力物力，提高劳动生产率。于是乎，智能节水灌溉系统，既能提升农业灌溉的自动化和智能化水平，又是应对水资源短缺和推进农业现代化的必然选择。

在庞大的设施农业灌溉及检测系统中，空气和土壤温湿度情况作为环境数据最重要参数，是植物生长不可缺少的条件。设施农业对植物生长环境的提供和保障也应以温湿度为重点。同时，由于设施农业区域集约化强，生产者和技术人员不能时刻准确测量所有大棚环境状况，因此，温湿度监测系统能够较大程度上节约农业生产人力资源，提高农业环境测量精准度，为设施农业环境控制提供数据

基础^[7]。

而另一方面，对普通用户而言，PC 机等管理平台实际应用并不理想，不仅价格昂贵、功能单一，而且便携性较差，导致其适用的人群十分有限，另一方面，目前监测系统开发平台多基于 windows 操作系统和 linux 操作系统，这使其在开元性、免费性和人机交互方面较差^[8]。因此在移动互联网快速发展的今天，设计一种更加便捷高效、价格较低廉且友好的监测系统对现代农业技术及时有效发挥现实作用有重要的意义^[9]。

近年来，随着全球特别是中国智能手机的快速发展，智能手机已经基本普及，最新统计数据显示，截止 2014 年 8 月份 Android 市场占有率高达 84.6%，平板市场也以 65.8% 的份额占据了大部。一方面 Android 智能手机是集通话、多媒体、上网等多功能于一体的智能终端，不仅价格低廉、携带方便，而且避免“二次投入”另一方面，开源、免费的 Android 操作系统不仅为软件设计者提供了更为灵活的自主设计空间，而且为研究和开发开源免费以及操作更具人性化的监控系统提供了较好的开发平台^[9]。

在物联网的实现过程中，需要更多可以与之相搭配的各类软硬件，一个合适的操作平台，一个简单的操作应用，时刻能够查看使用以及控制的工具，而此时最适合提供这样的平台的就是智能手机以及 Android 这样的完全开源的操作系统。这样的工具也正是一款 android 智能硬件中的应用，时时刻刻通过而 android 监控你需要监控的一切，这将是物联网的大趋势，未来生活的大方向，android 也将在这个即将崛起的产业中拥有一个更加开阔的前景。本文设计并实现了一种基于 Android 平台的药材设施栽培远程灌溉系统，可实现在移动设备实时显示温湿度数据及进行历史查询和数据分析等综合功能^[9]。

1.2 国内外研究现状

1.2.1 Android 开发平台技术研究现状

Android 一词本意是指“机器人”。Google 公司在 2007 年 11 月发布了一款智能手机操作系统 Android。它是一款基于 linux 开源操作系统并且应用于智能手

机的新平台。之后交由开放手持设备联盟（Open Handset Alliance）进行开发，是号称首个为智能手机打造的真正全开放移动软件。开放手持设备开发联盟是由 Google 与 HTC、T-Mobile、摩托罗拉、索尼、华为和中国移动等 30 多家世界移动通信领域的领军企业组成的，宗旨是打造产业内开源的生态系统。Android 操作系统最大的特点即是其完全开源。让 Android 应用程序开发者能够在有源码的基础上自由地对此程序进行二次开发，开发出各种具有独创性的应用程序，并且也可以避免专利和授权等问题的风险，这也打破了 iPhone 等智能手机在开发应用时的限制。所以在 Android 操作系统推出的五年间它得到了众多手机厂商、通信网络运营商、开发者们和消费者的喜爱，在此移动终端市场中掀起了一场新的革命^[20]。

在物联网中的应用，不得不提的便是智能家居，而在家庭环境检测监控系统的手机终端选用 Android 系统是基于以下的考虑：

- android 操作系统通过 3C 数码领域得到了广泛的使用和认可，有利于本系统的应用普及。
- android 操作系统是开源并且可以免费下载，减少了系统开发的成本和周期。
- 它具有很强的扩展性，系统的升级和应用程序的修改非常方便。
- 开发工具高效快捷技术成熟，通过 JAVA 平台 eclipse 的 ADT 开发包就可以进行软件开发。

近几年，中国国内智能手机产业快速发展，其普及率大大提升。最新统计显示，截止于 2015 年 8 月份 Android 智能设备在市场中占有率高达 82.4%。Android 智能手机是集多媒体、上网、通话等多功能于一体的智能手持设备，价格相较当前市场较低廉^[8]。另一方面，开源且免费的 android 平台为开发者们提供了自由、人性化的开发空间^[2]。

对于智能设备应用软件开发来说，比较关心的是手机操作平台的前景与市场。当今，移动操作系统平台的市场份额绝大多数被 IOS，Android 及少部分 Windows Mobile 所占领。自 2009 年以来，Android 操作系统已经成为开发者最受欢迎的操作系统，正因为其开源的特点，开发 Android 应用软件具有很好的市场价值与前景，开发 Android 系统手机打入市场也成为国内外手机厂商的重要

手段，如国内的华为、小米、魅族等以及国外的 Samsung 等，他们都以 android 系统为基础进行的二次开发并获得良好成效与增长的厂商。

Android 应用的开发的另一大特点是它只需要运用 java 语言就可以进行大部分的开发，一小部分涉及至底层的需要通过 JNI 来使用 C 或 C++ 语言生成一些库来使用，这让开发者们能够很容易的就熟悉其开发流程，android 操作系统是完全开放的，能够接纳第三方软件，所以开发者也可以利用网上众多的开源代码在 android 平台上开发出自己的应用，这让开发者们拥有了前所未有的开发设计空间，Android 近几年的井喷之势和其能够迅速占领市场的成绩也正是因为其开源性和免费的服务平台。而本课题将基于 android 平台的开发展开应用研究工作。

1.2.2 数据库技术国内外研究现状

随着 Linux 开源操作系统受到来自广大开发者与应用者的欢迎之后，开源的数据库也逐渐崭露头角。近些年来，越来越多的大企业开始使用开源的数据库，它们大多在企业总部采用闭源且收费的商业型数据库，而在其分部更多的是使用一些开源的软件^[10]。这些开源数据库除了低廉的价格甚至免费外，也拥有各自擅长的领域。相比于商业产品，开源数据库组成构造较为简单，但功能依旧强大。操作同样快捷，易于管理。

现今，MySQL 做为当前最流行的开源数据库之一，为开发者与用户提供了一个相对简单的方案，适用于广泛的应用程序部署，降低了用户的 TCO。MySQL 是一个多线程的、结构化的数据库服务器。MySQL 的运行速度快，执行效率高且易于使用^[11]。作为一个开源的小型关系型数据库管理系统，MySQL 从 4.0 版本中开始加入了 Spatial 模块，用于实现 OpenGIS 规定的几何数据类型和简单的空间数据运算^[12]。不过作为其缺点 MySQL Spatial 并没有得到持续地增强修改，MySQL 所支持的空间扩展性能仍然较弱^[22]。

而另一款轻量级的数据库 SQLite 已嵌入在 android 系统中。SQLite 数据库最初的设计目的是嵌入式设备。它是由 D.Richard Hipp 于 2000 年 5 月发布，其在设备中与应用程序中占有的空间十分少，在嵌入式设备运行时所需的内存可能只需要几百 KB，它非常适合应用在内存等资源较为有限的设备中，如手机、MP3、PDA 等。SQLite 同样支持在 Android、Linux、Unix、Windows 等操作系统中搭

建运行，同时也支持 C/C++、Java、PHP 等高级开发语言，同样的具有开源、执行速度快、可移植性好、高效稳定等特点，同时也支持基本 SQL 的语句。

1.3 研究内容及组织结构

本次设计的任务是通过 Java 语言编写串口通信模块与 JDBC 数据库通信模块，之后使用 MySQL 数据库对数据进行存储，编写 php 网页，用于读取数据内容，并显示与网页上，之后使用 nat123 对该网页进行映射，使其能够被外部网络访问。最后使用 Android studio 开发 android 手机客户端对数据进行实时的监控。

本文主要研究内容如下：

1. 使用 Java 语言的串口通信类库以及 JDBC 进行数据处理
2. MySQL 数据库对数据的存储，以及 php 网页的编写
3. Android 手机客户端的开发

以下是本文的组织结构：

第一章：为概述，讨论本课题的背景和意义，以及国内外对 android 平台，MySQL 数据库和 java 语言的研究现状，同时介绍了本课题主要研究的方向与内容。

第二章：对整个远程灌溉控制系统 android 手机客户端的总体设计方案进行叙述，首先进行需求分析，并根据需求对软件进行相对应的功能设计，并简要介绍 Android 与 Java 语言相关技术，以及本课题中涉及的众多相关开发软件，编程语言等都做一定的介绍，以一个具体的流程进行叙述。

第三章：串口通信部分的设计与实现，以及 android 手机客户端的设计与实现。

第四章：总结与展望。

1.4 本章小结

本章介绍了本课题的背景和意义，以及国内外对 android 平台，MySQL 数据库和 java 语言的研究现状，同时介绍了本课题主要研究的方向与内容。

第2章 系统总体设计

2.1 系统需求分析

近年来,随着智能农业、精准农业的发展,智能感知芯片、移动嵌入式系统等物联网技术在现代农业中的应用逐步拓宽。随着农业大棚面积不断扩大,规模不断增加,传统的有线布线方法和人工读取监控数据的生产方式,已经无法满足农业大棚管理人员的工作需要^[13]。现有设施农业测控系统能够较大程度上解放劳动力,使生产者远程测控温室环境。

学者们主要研究方向在于无线传感器网络的优化设计和应用,但对数据库数据的处理和使用模式仍存在一定限制和不完善性。在移动通讯端搭建远程监测平台,能够随时随地为生产者提供温室内实时环境信息,并能搭建远程控制系統。如今 IT 设备成本下降、性能提升,这些都为建立农业方面环境监测监控系统提供了条件,目前国内外在这些设施设备上发展迅速,并已具有相对完善的解决方案^[7]。使用远程灌溉控制系统能够根据农业生产特性和动植物生长规律较好的控制设施农业环境要素,真正提高农产品产量和质量^[14]。

本设计将采用符合 ZigBee 传感技术,定时采集药物阳春砂的土壤温湿度,再结合数据自动进行相应控制以调控环境参数,经由串口将数据传送至数据库,在 android 平台建立客户端,实现实时显示,使作物始终处于最佳生长环境,确保各项生产活动科学有序、规范持续的进行。

根据上述分析,本次设计需要实现如下功能:

- (1) 收集来自传感模块的数据。
- (2) 分析数据,并进行存储。
- (3) 建立外部网络,提供访问查看数据端口
- (4) 开发 android 端应用
- (5) 根据需求,应用需要具备登录注册,查看当前数据,查看温湿度变化曲线,记录查询等功能。

2.2 系统总体设计

2.2.1 Java 语言

Android 应用程序是由 Java 语言开发的，不仅在 Android 领域有着重要位置，在 Web 交互式应用程序中也是首选开发语言。Java 语言是由 SUN 公司（Stanford University Network 斯坦福大学网络公司）于 1995 年推出的一门高级编程语言，它完全面向对象，安全可靠，可实现跨平台，与操作系统无关，相较于 C、C++ 等编程语言，java 拥有的更为可靠的安全机制，异常处理，万物皆对象的理念^[19]。

java 语言具有三种技术架构：

- ——>1.J2EE/JAVAEE:企业版，是为开发于企业环境的应用程序所提供的一套解决方案，主要技术有 servlet, jsp 等，主要针对于 web 应用程序开发。
- ——>2.J2SE/JAVASE:标准版，是为开发于普通桌面和商务应用程序所提供的解决方案，可以开发一些桌面应用程序的开发。
- ——>3.J2ME/JAVAME:小型版，是为开发于电子消费产品的嵌入式设备所提供的解决方案，该技术体系主要应用在小型电子消费类产品。

本次设计也将使用 Java 语言的第三方串口通信的类库来编写串口通信程序，实现从传感网络获取温湿度数据等，而之后也将使用 java 语言的 JDBC 协议来操作 MySQL 数据库，将这些分析之后的温湿度数据存储到数据库中。

2.2.2 Java 串口通信

完成 MySQL 数据库与串口的数据通信交互，在 zigbee 传感模块通过串口传输以字符串为单位的温湿度等数据时，需要使用串口通信技术，而本次设计中采用了 JAVA 语言进行开发。C 语言是最适用于机械语言的开发语言，但 java 语言由于其广泛的开发者群体，拥有许多可以使用的类，相对 C 语言更加便捷，java 语言中提供了串口通信相关的类库：javax.comm.CommPort；javax.comm.CommPortIdentifier；javax.comm.SerialPort 等，运用这些接口，将开发的时间成本等降到最低，而且本次设计的 android 应用程序也同样使用 java 语言，这使得开发更加方便。

2.2.3 MySQL 数据库

本次设计中需要使用数据库来进行数据的存储，而本次将选用开源免费的 MySQL 数据库来担任这个角色。

现今数据库中有许多选择比如轻量级的 SQLite，它是 android 本地中采用的数据库，早在几年前就已经正式融入 android 应用程序中，但其在 PC 端搭建的使用较少，大多数小型数据库的选择都为 MySQL，作为开源且免费的数据库相对于 oracle 的大型且收费，SQL server 的收费来讲，MySQL 的优势依然很明显。摒除大型且收费以及不开源属性和技术不成熟等因素，MySQL 将是首选。本次设计应用 MySQL 数据库软件搭建数据库，以主机为服务器，创建与串口、外网以及手机客户端之间的连接。

2.2.4 内网解析映射外网

同类解析映射软件有多种，例如花生壳，而 nat123 的优势在于其应用简单，简洁明了的界面等。运用 nat123 的目的是为了使 app 获取外部接口，映射内部数据库至外网以提供 app 数据接口。

2.2.5 Android 应用程序设计

Android 是由 Google 公司开发的一款基于 linux 的操作系统，国内俗称“安卓”，也是现今全球应用最为广泛的智能手机操作系统，它使用了 Google 自主研发的 Dalvik Java 虚拟机，安卓系统承载了 linux 的完全开放源代码的性质，从底层至上层都不存在能够阻碍开发者与产业创新的专有权的障碍，也正是因此，android 逐渐拥有一支壮大的开发者队伍，且这支队伍还在不断扩大，在这样的互相促进中，Android 越发成熟与稳定。

Android 软件框图的四个层次结构自上而下分别为^[15]：

- 应用程序（Application）。
- 应用程序框架（Application Framework）。
- Android 的运行环境（RunTime）和各种库（Libraries）。

● 操作系统层面（Operation System）。

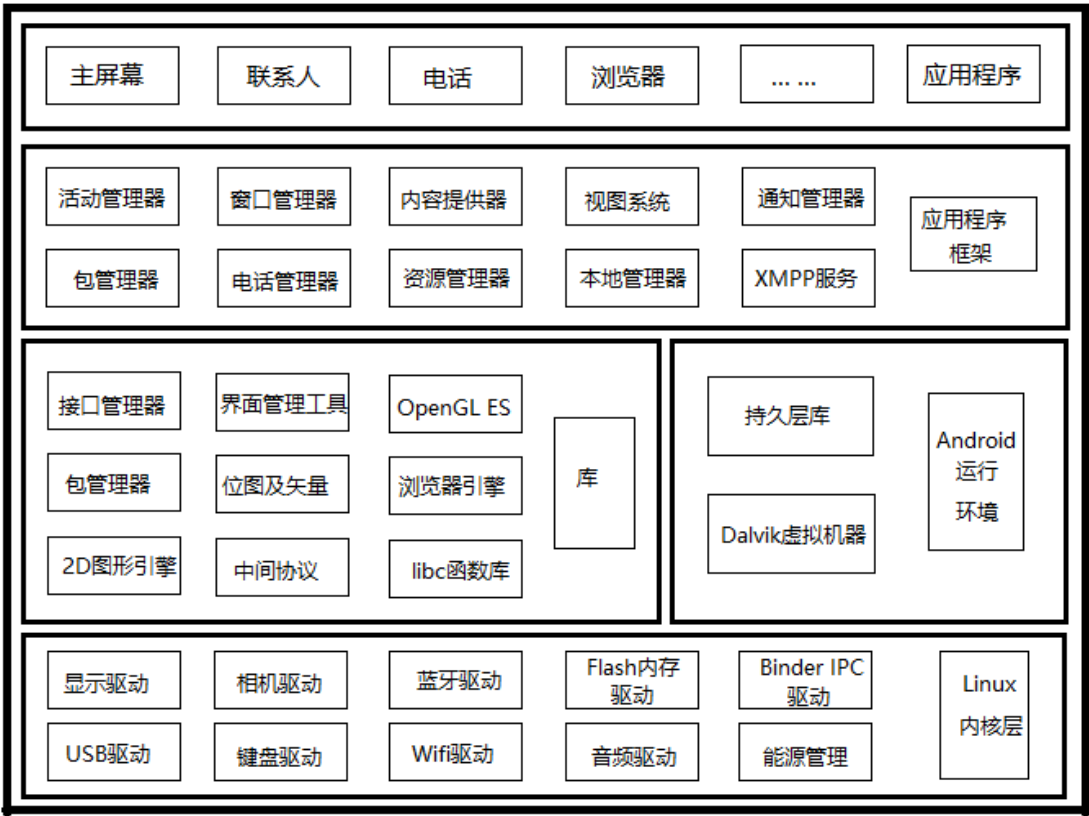


图 2.1 Android 操作系统组件结构图

本文设计的药材设施栽培远程灌溉控制系统安卓手机客户端软件设计具有客户端登录、实时温度监测、历史数据查询、温度趋势图绘制等功能，如下图所示。

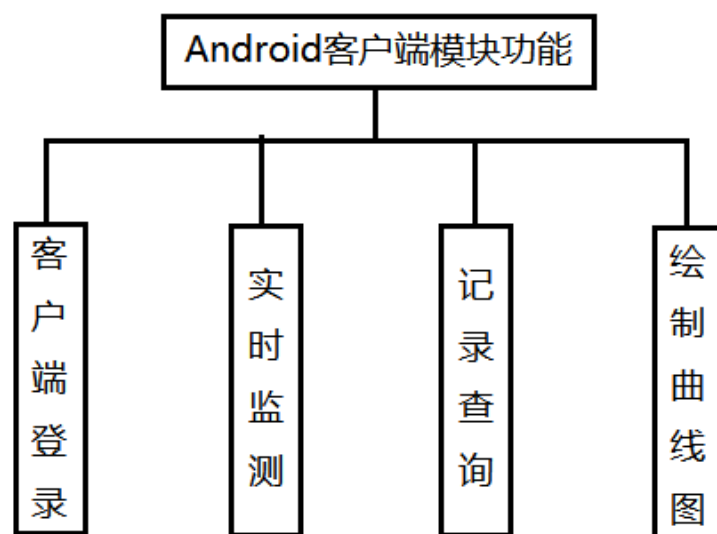


图 2.2 Android 客户端功能模块

2.2.6 系统的总体结构

综合以上设计，总体设计框图如下所示，由传感网络通过串口传输温湿度 10 位字符串数据至上位机，上位机使用 Java 语言编写的串口通信程序接收数据，分析数据，之后使用 JDBC 连接数据库，并将数据存入 MySQL 数据库，在这中间还讲使用 php 编写网页显示数据库数据，之后使用 nat123 将这个 php 网页映射至外部网络，而最后手机端的客户端使用 HttpURLConnection 协议访问该网页，获取网页上数据。在手机端将实现登录注册界面，实时显示数据的页面，变化曲线图的页面，以及记录查询功能。



图 2.3 系统总体设计结构图

2.3 本章小结

本章确定了系统的基本需求，也详细介绍了系统总体的设计方案，首先是 Java 语言上的使用，使用其中的 Comm 类库与 JDBC 协议，以及之后的 Android 客户端开发依旧使用 Java 语言，而数据收集部分由 MySQL 数据库进行，由 Java 的 JDBC 创建与 MySQL 之间的联系，之后使用 php 的网页，创建本地网页端的数据显示，使用 nat123 进行映射，最后是 android 收集客户端的设计方案，通过本章内容，可以对系统的整体设计有基本的了解。

第3章 数据获取的设计与实现

系统软件设计主要包括了数据的收集与存储，即 Comm 类库的使用，JDBC 的使用，MySQL 数据库的搭建，之后数据网页显示映射，即 php 网页的编写与 nat123 软件的映射。

3.1 数据的收集与存储设计与实现

数据收集部分将使用 Java 语言进行编写，使用 Comm 类库，之后进行数据解析，然后使用 JDBC 进行数据存储，这之间还有 MySQL 数据库的搭建。

本次程序流程如下：

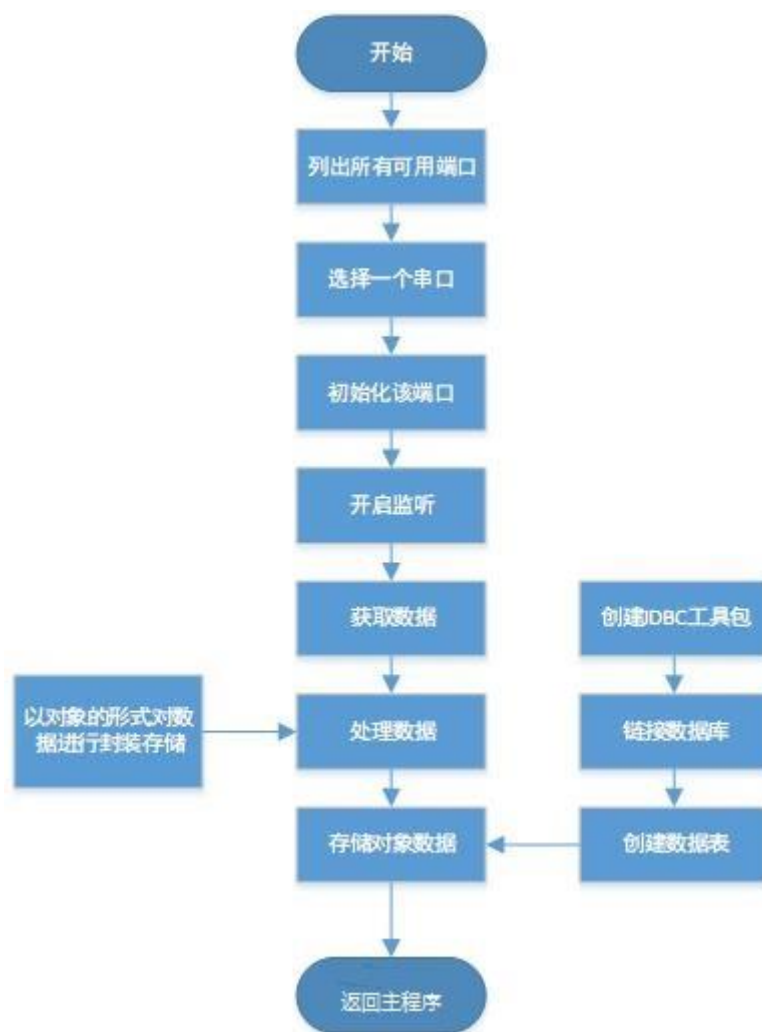


图 3.1 串口通信与 JDBC 程序流程图

3.1.1 串口通信程序编写

嵌入式设备与传感器网络许多测试或实现都需要通过 PC 端与其连接通信。这中间，使用最多的接口就是 RS-232 串口和并口。SUN 的 CommAPI 分别提供了对常用的 RS232 串行端口和 IEEE1284 并行端口通讯的支持^[16]。RS-232-C(又称 EIA RS-232-C，以下简称 RS232)是在 1970 年由美国电子工业协会(EIA)联合贝尔系统、调制解调器厂家及计算机终端生产厂家共同制定的用于串行通讯的标准^[17]。RS232 是一个全双工的通讯协议，它可以同时进行数据接收和发送的工作。

当下常见的 Java 串口包有 1998 年 Sun 公司发布的 comm2.0.jar 的 API。本次设计也将采用这个包进行使用。

(一) 根据流程图第一步，列出的可用串口之后，选择其中可用串口，程序如下：

```
/**
 * @方法名称 :selectPort
 * @功能描述 :选择一个端口
 * @返回值类型 :void
 * @param portName
 */
public void selectPort(String portName) {
    this.commPort = null;
    CommPortIdentifier cpid;
    Enumeration en = CommPortIdentifier.getPortIdentifiers();
    while (en.hasMoreElements()) {
        cpid = (CommPortIdentifier) en.nextElement();
        if (cpid.getPortType() == CommPortIdentifier.PORT_SERIAL
            && cpid.getName().equals(portName)) {
            this.commPort = cpid;
            break;
        }
    }
}
```

```

    }
    openPort();
}

```

选择过程即是匹配传入的串口名如 COM1 与可用串口中的进行匹配，然后选取串口，之后打开串口开始下一步。

（一）初始化串口核心代码：

```

serialPort.setSerialPortParams(115200,
SerialPort.DATABITS_8,SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);

```

第一个参数为波特率，由硬件方面决定，所以定义初始化波特率为 115200；第二个参数为数据位，为 8；第三个参数为停止位，为 1；最后一位为校验位，默认为 NONE。

初始化串口后便是开启监听了：

```

/**
 * @方法名称 :startRead
 * @功能描述 :开始监听从端口中接收的数据
 * @返回值类型 :void
 * @param time
 *           监听程序的存活时间，单位为秒，0 则是一直监听
 */
public void startRead(int time) {
    checkPort();
    try {
        inputStream = new BufferedInputStream(serialPort.getInputStream());
    } catch (IOException e) {
        throw new RuntimeException("获取端口的 InputStream 出错：" +
e.getMessage());
    }
    try {
        serialPort.addEventListener(this);

```

```

    } catch (TooManyListenersException e) {
        throw new RuntimeException(e.getMessage());
    }
    serialPort.notifyOnDataAvailable(true);
    if (time > 0) {
        Thread t = new Thread(this);
        t.start();
    }
}

```

首先是检查串口连接是否有效，之后尝试从串口获取输入流，为串口添加事件监听，即当串口发送数据就触发事件，而最初传入的时间数据是初始设定的监听总时长，设为 0 即不自动停止，本次将设定为 0，保持持续的监听。之后开启线程，保持监听的开启状态，用于获取数据。

（三）在开启线程后，获取串口的数据，本次设计中，首先对数据进行了对应处理，从 10 位数据位中提取相对应的数据值。

首先，创建了一个 Data 类，用于封装存储每一次串口传入的数据，当然是封装后的：

```

public class Data {
    public int id;
    public int soilTemperature;
    public int soilHumidity;
    public int airTemperature;
    public int airHumidity;
    public String currentTime;
    public Data(int soilTemperature, int soilHumidity, int airTemperature,
        int airHumidity, String currentTime) {
        super();
        this.soilTemperature = soilTemperature;
        this.soilHumidity = soilHumidity;
    }
}

```

```

        this.airTemperature = airTemperature;

        this.airHumidity = airHumidity;

        this.currentTime = currentTime;
    }
}

```

（四）该类有 soilTemperature, soilHumidity, airTemperature, airHumidity, currentTime 五项成员属性。从其名字中即可得知，其分别为，土壤温度，土壤湿度，空气温度，空气湿度，当前时间。而串口中获取的数据为 10 位字符串数据：

字节	1	2	3	2	2
含义	设备节点 ID	土壤温度	土壤湿度	空气温度	空气湿度

图 3.2 串口数据流字节分析

创建一个 ResolvedData 类进行数据的封装存储，核心代码为：

```

String strData = new String(portData);

byte[] resultPortData = strData.trim().getBytes();

int soilTemperature = (resultPortData[1] * 10) + resultPortData[2];

int soilHumidity = (resultPortData[3] * 100) + (resultPortData[4] * 10) +
resultPortData[5];

int airTemperature = resultPortData[6] * 10 + resultPortData[7];

int airHumidity = resultPortData[8] * 10 + resultPortData[9];

String currentTime = new SimpleDateFormat("HH:mm:ss").format(new
Date()).toString();

Data data = new Data(soilTemperature, soilHumidity, airTemperature, airHumidity,
currentTime);

```

首先是获取串口传入的数据流，将其转为字节数组，分别将第 9 位与第 8 位数据作为土壤温湿度，7、6、5 位作为土壤湿度，4、3 位为空气温度，2、1 位为空气湿度，并获取当前存储的时间，之后创建 Data 对象，传入上述 5 个数据将其封装。

而之后将 JDBC 进行下一步工作。

3.1.2 JDBC 数据存储程序编写

JDBC (Java Database Connection)，顾名思义，使用 java 技术连接数据库。Java 语言提前定义了和数据库交互的规则，使用接口来实现。而各个数据库厂商生产数据库软件的时候，都对 Java 提供的接口进行了实现。而 java 程序要操作数据库，首先是要在项目中导入响应的数据库驱动 Jar 包，本次设计采用的是 MySQL 数据库，使用的是：mysql-connector-java-5.1.7-bin.jar 包。

(一) 在使用该工具时，为方便使用，创建了一个 JDBCUtils 类，内部封装了数据库连接：

```
DriverManager.registerDriver(new Driver());  
String url = "jdbc:mysql://localhost:3306/data";  
String user = "root";  
String password = "";  
Connection conn = DriverManager.getConnection(url, user, password);
```

第一步为加载驱动，然后定义驱动管理器创建链接所需的三个参数，分别是数据库本地的端口和数据库，之后是账号与密码。

(二) 连接数据库为第一步，之后是在数据库中创建一个表，用于存放数据：

```
String sql = "insert into data(id, soilTemperature, soilHumidity, airTemperature,  
airHumidity, currentTime) values (" +  
sum + "," +  
data.soilTemperature + "," +  
data.soilHumidity + "," +  
data.airTemperature + "," +
```

```

        data.airHumidity + "," +
        data.currentTime + "));";

Statement stmt = conn.createStatement();

stmt.executeUpdate(sql);

```

JDBC 只是一个创建连接的工具，要操作数据库依旧需要使用 SQL 语言，在存储数据之前，在 MySQL 中已提前创建了一个名为 data 的表，上述程序进行了一条插入数据的操作，该方法首先传入了一个 data 对象，而插入语句中是分别将 data 中的 5 个数据以及一个自定义的 id 号，分别存入表中对应位置，id 号是为数据条目做一个唯一标记，id 号也为主键。之后使用 statement 类创建一个预编译 SQL 语句的对象，对该语句进行预编译，之后执行该语句。

（三）在这之间，为了防止数据库数据过多，之后定义了一个静态常量，每存入一条 data 的数据，该值就会加 1，即为存入的条数，而当该值超过 15 时将会进行删除当前表中第一条的操作：

```

if(sum > 15){

    String sql = "delete from data where id="+sum;

    PreparedStatement stmt = conn.prepareStatement(sql);

    stmt.executeUpdate(sql);

}

sum++;

```

3.1.3 MySQL 数据库的搭建

当开发中运行程序需要的数据量较大，并且需要重复使用时，IO 流等无法做到这样的数据存储，而这时就需要数据库来存储这些数据。数据库是一个软件，早期时的数据库都是关系型数据库，即数据之间存在相对应的关系。数据存入数据库后也需要去维护这样的关系 [18]。

当前数据库软件众多，在进行本次设计的需求分析时，也进行了逐一的分析和筛选。无疑的 Oracle 公司旗下的几款数据库占据了数据库市场的半壁江山。而 MySQL 数据库早期由

瑞典公司开发，之后被 sun 公司所收购。但在之后 sun 公司却也被 Oracle 收购，所以如今 MySQL 也属于 Oracle 公司，并计划跟随旗下几款数据库一样，从 MySQL 的 6.0 版本开始也将收费。

（一）本次开发使用的是 5.5 版本的 MySQL，使用的是 Navicat of MySQL 可视化工具，安装时选用 utf-8 的编码表，统一编码，方便使用。

使用 Navicat 首先进行数据库连接：

主机名或 IP 地址:

localhost

端口:

3306

用户名:

root

密码:

☒ 保存密码

图 3.3 Navicat 连接本地 MySQL 数据库

由于密码设置为空，所以本次没有密码，主机名与 IP 地址在本机上默认为 localhost 或 127.0.0.1，MySQL 数据库程序默认端口为 3306，用户名为 root，密码为空。

然后在本地创建一个名为 data 的数据库，在 data 数据库下再创建一个名为 data 的表：

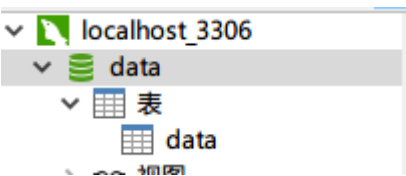


图 3.4 创建数据库与数据表

创建 data 表，定义其列属性：

名	类型	长度	小数点	不是 null
id	int	11	0	<input type="checkbox"/>
soilTemperature	int	11	0	<input type="checkbox"/>
soilHumidity	int	11	0	<input type="checkbox"/>
airTemperature	int	11	0	<input type="checkbox"/>
airHumidity	int	11	0	<input type="checkbox"/>
▶ currentTime	varchar	50	0	<input checked="" type="checkbox"/>

图 3.5 定义 data 数据表列属性

3.2 内部服务器至外部网络映射

这一部分包含了 php 读取 MySQL 表数据，nat123 映射 80 端口下 php 网页，配置外部网络 URL 地址。

3.2.1 Php 网页的编写

Php（Hypertext Preprocessor）是一种易于学习和使用的服务器端脚本语言。其语法混合了 C、Java、Perl 以及自身的一些语法。本身也是开源免费的，拥有跨平台性，面向对象等特点。本次使用 php 编写网页，用于创建与 MySQL 的简单连接，用于之后的映射。

代码如下：

```
<?php

$link = mysqli_connect('localhost:3306','root','') or die('数据库连接失败');

mysqli_select_db($link,'data') or die('数据库选择失败');

mysqli_query($link,'set names utf8');

$rs = mysqli_query($link,'select * from data');

?>

<table>

<?php

while($rows=mysqli_fetch_row($rs))

{

    echo '<tr>';

    echo

'<td>'.$rows[1].'~~'.$rows[2].'~~'.$rows[3].'~~'.$rows[4].'~~'.$rows[5].'</td>';

    echo '</tr>';

}

?>
```

与 JDBC 相同，php 连接数据库，第一步依旧是创建链接，php5 中使用函数 mysqli 进行这些工作，之后是选择数据表，定义编码，查询表中所有列。而下方的即是网页上的显示的数据了，为了方便之后的抓取，所以将数据库中的每一行数据组成一行显示，每个数据中间以波浪线隔开，该网页也仅用于之后应用端的

抓取，所以没有什么布局。

实现结果：

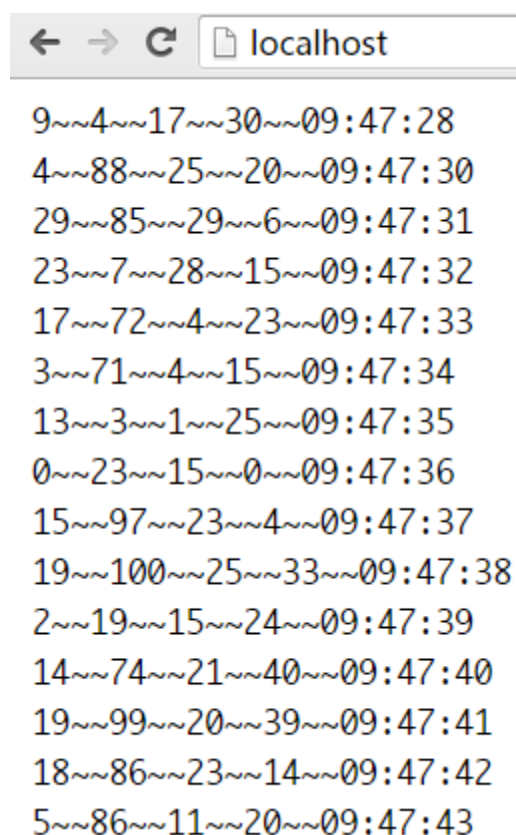


图 3.6 php 网页数据显示结果

上面显示的数据是测试用的随机生成的数据，与真实数据无关，但范围大小相同，为了有更清晰的显示效果，所以使用测试类以每秒一组的速度生成随机数据，存入数据库。数据显示数量上限为 15 条，当然为了配合实验数据，可以做一些适当的修改。

3.2.2 Nat123 的映射

Nat123 是一款解析映射软件，同类的软件有多种，例如花生壳。而 nat123 的优势在于其应用简单，简洁明了的界面等，当然首先他们都是收费的。

首先是配置：

您当前的位置：首页 > 端口映射 > 映射编辑

应用类型：	非80网站	非80网站映射适用不考虑百度收录的网站
映射线路：	nat123	免费VIP 如何切换VIP 我的世界免费满格信号
应用名称：	data	?
内网端口：	80	?
内网地址：	localhost	?
外网端口：	26169	?
外网域名：	15054375.nat123.net	?
自主域名：	将dns或ns设为:dns1.dns123.net/dns3.dns123.net (荐)或将cname(唯一)设为:054375.0349f5.1.c.cname123.net 映射怎么	
确认保存		剩余N币316 保存需1N币 猛击这里免费获取N币 T币N币转

图 3.7 nat123 基本设置

内网端口由 php 的开发环境 httpd 运行端口决定，默认为 80 端口，而浏览器在打开 80 端口时，不需要加上端口号，默认添加。内网地址也默认为 localhost。至于外网端口与外网域名，需要收费才能够设置自定义，本次使用默认配置。

由于 Android Studio 编译器以及 genymotion 模拟器与 nat123 映射方式存在冲突，两者无法同时开启，所以采用真机测试，nat123 正常情况下映射成功后显示为：

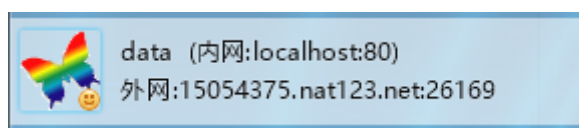


图 3.8 nat123 映射成功

访问该页面：

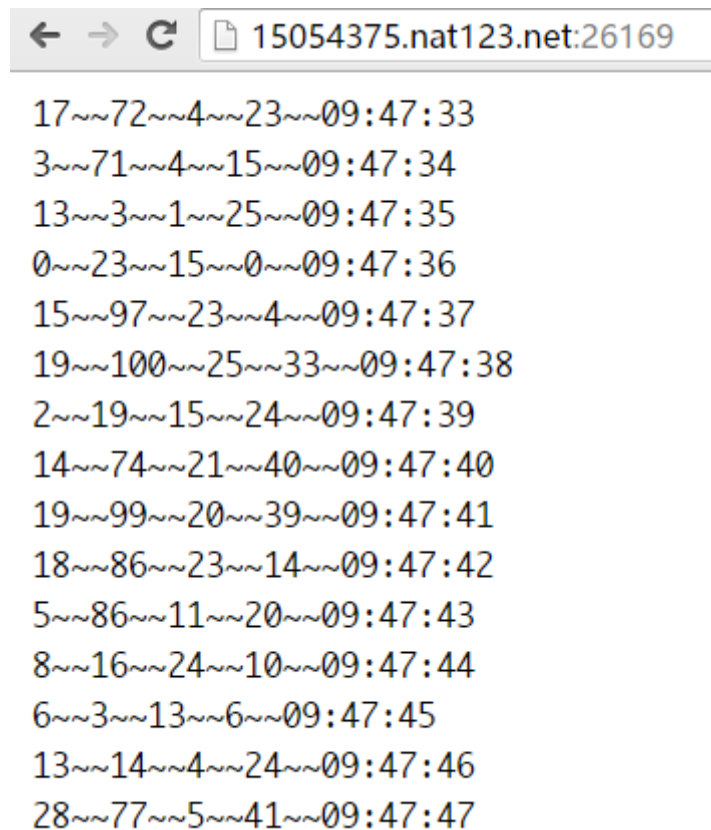


图 3.9 nat123 映射结果

3.3 本章小结

本章详细介绍了，使用 Java 非核心类库 Comm 库的使用，接收来自串口的数据，到使用 JDBC 传输数据至 MySQL 数据库，再到使用 nat123 映射本地端口网页至外部网络，为 app 提供数据接口。

第4章 Android 应用的设计与实现

4.1 设计总述

本次设计主题即是 Android 手机应用程序的开发，而上述步骤用于提供 app 数据。本次设计采用 Android studio 开发应用程序，使用 genymotion 作为模拟器进行开发调试。应用程序将分为登录注册页面，主要数据显示页面，折线图页面，以及记录查询页面。

4.2 开发工具介绍

4.2.1 Android studio 编译器

Android Studio 是 Google 官方基于 IntelliJ 开发的一款 Android 应用开发工具，而在这之前，国内最主流的 android 开发工具是 eclipse ADT。相比于如今的 Android studio，功能略显不足，经常容易出现异常问题，以及对 git，第三方库的支持交差。而 Android Studio 是 Google 为 android 开发者量身定做的编辑器，强大并专业，内置 6.0SDK，基于 Gradle 灵活编译。布局更加适用于 android 开发，有更多的工具用来捕捉应用的性能，版本兼容性问题。

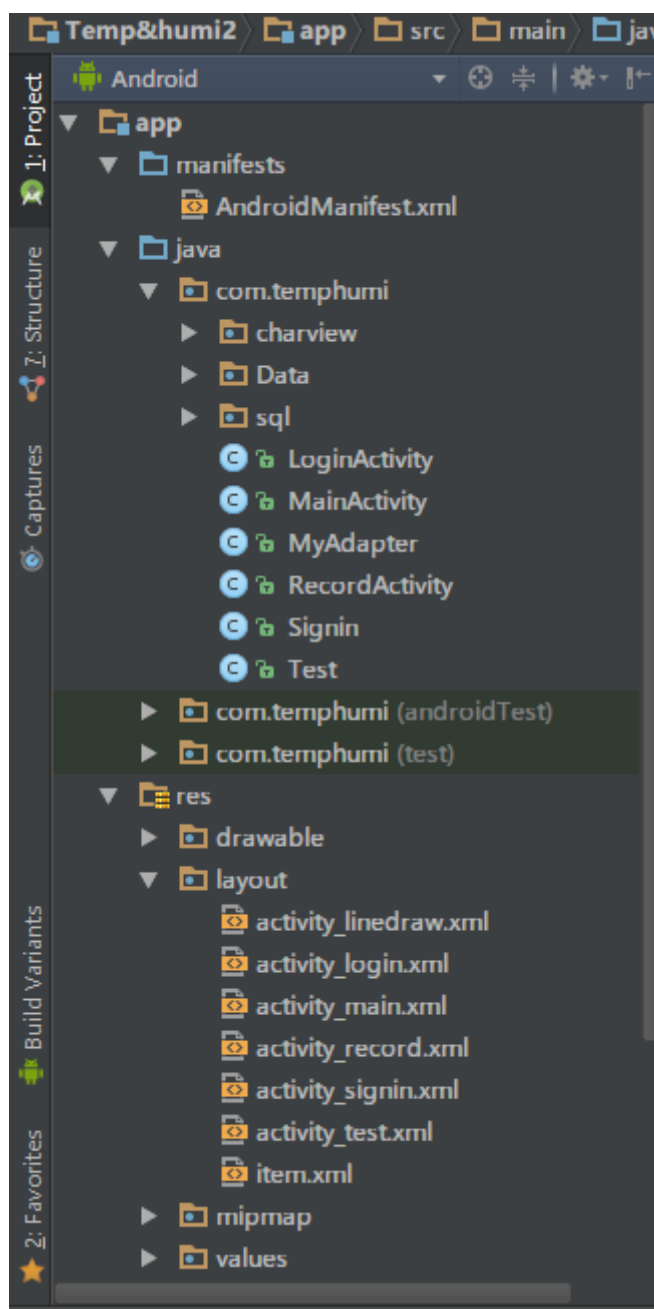


图 4.2 Android Studio 布局

直接定位 app 文件,从最上方开始分别是清单配置文件 `AndroidManifest.xml`, 然后是主要的活动类,工具类等 java 文件;下方为资源文件,有布局文件,图片等。

4.2.2 Genymotion 模拟器

由于原生模拟器硬件要求过高,且卡顿严重,所以使用第三方模拟器。Genymotion 提供了一套完整的 Android 测试环境。严格意义来说,他并不是模拟

器,他依赖于 virtual box,创建 android 虚拟机。支持同时开启多个模拟器,OpenGL 加速等。

首先需要至其官网注册下载, 安装后, 下载虚拟 android 版本的虚拟镜像:

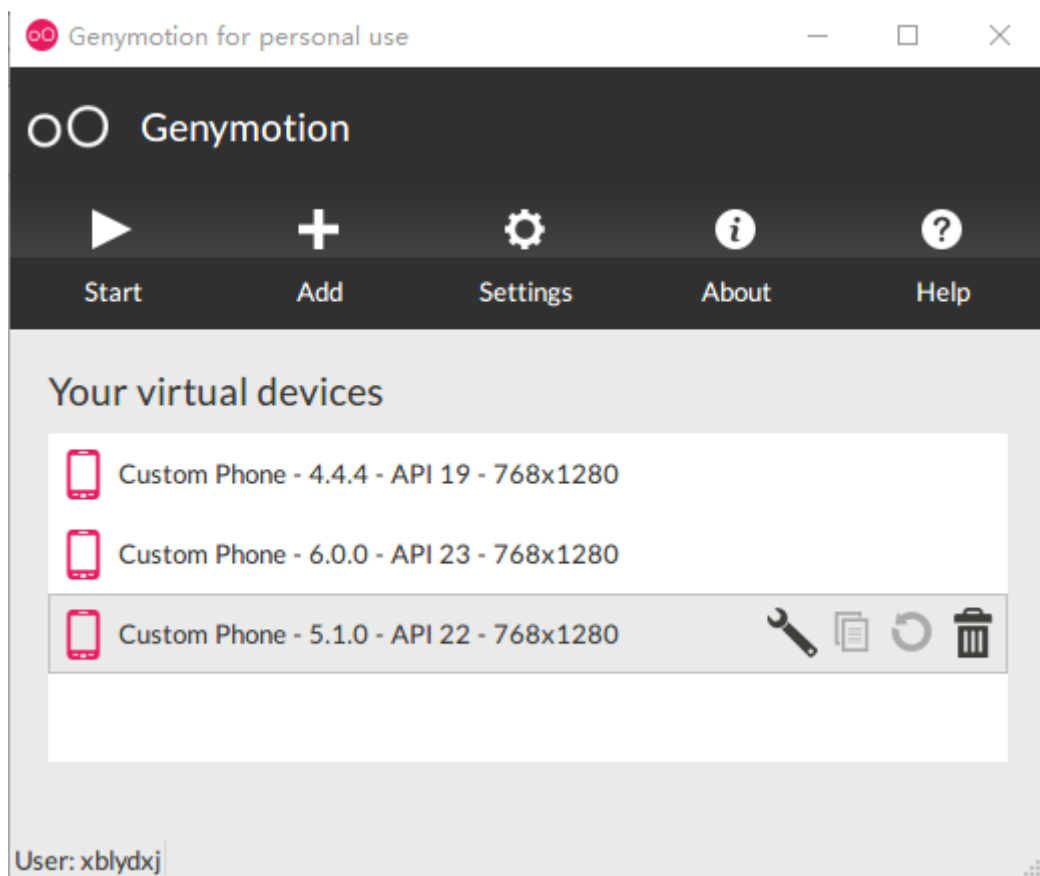


图 4.3 Genymotion 虚拟机

本次选用 Android 5.1 版本的模拟器, 开启之后:

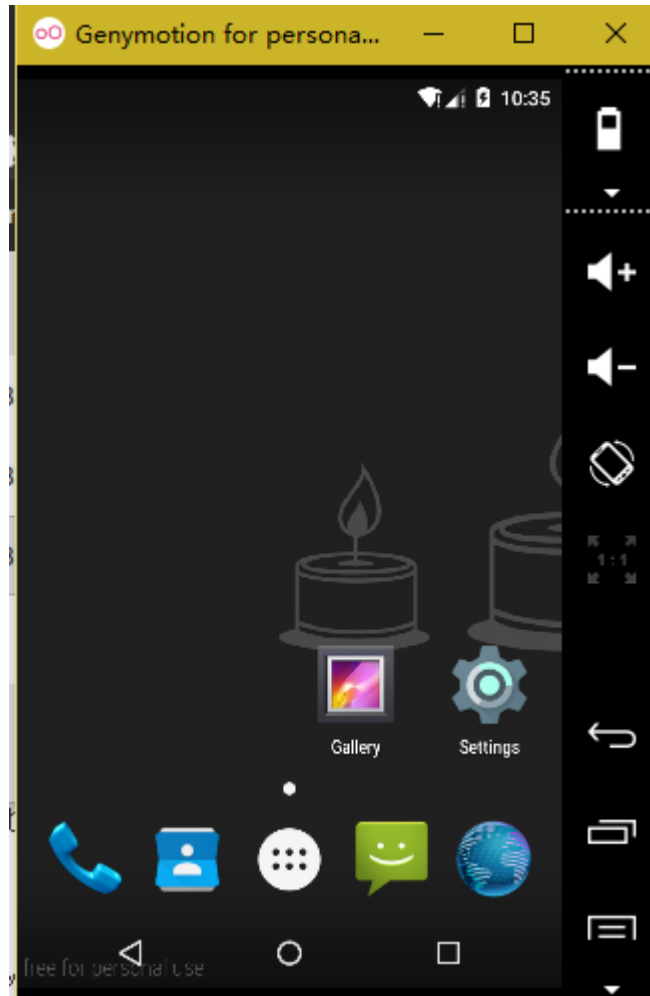


图 4.4 Genymotion Android 模拟器

开发中调试也将使用 5.1 版本，本次设计 Android 版本支持最小版本为 4.4，最大为 6.0.

4.3 应用程序设计

App 开发具体流程为：

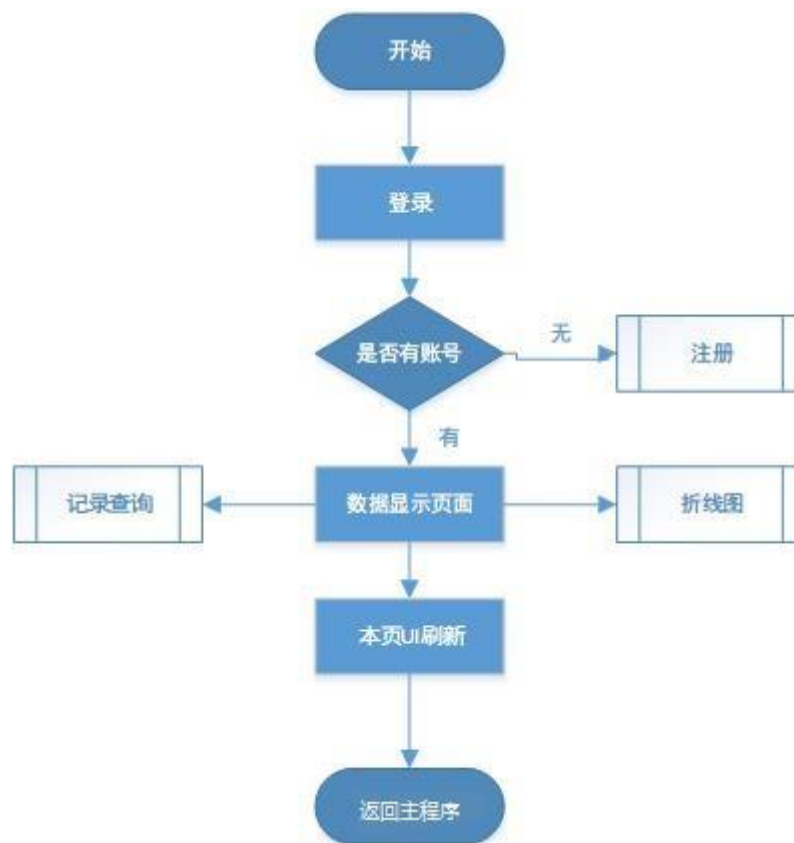


图 4.1 Android 应用开发步骤

4.3.1 登录注册页面

当 app 安装后，打开第一个页面即为登录页面，之后点击注册，进入注册页面：



图 4.5 登录页面



图 4.6 注册页面

登录页面核心代码：

```
public void onClickLogin(View view) {  
    String name = et_name.getText().toString().trim();  
    String password = et_password.getText().toString().trim();  
    if (TextUtils.isEmpty(name) || TextUtils.isEmpty(password)) {
```

```

        Toast.makeText(this, "用户名或密码为空", Toast.LENGTH_SHORT).show();
    } else {
        MySQLiteOpenHelper_New mySQLiteOpenHelper = new
        MySQLiteOpenHelper_New(this, VERSION);

        SQLiteDatabase database = mySQLiteOpenHelper.getWritableDatabase();
        Cursor cursor = database.query("user_name", null, null, null, null, null, null);
        while (cursor.moveToNext()) {
            String name_cursor = cursor.getString(cursor.getColumnIndex("name"));
            String password_cursor = cursor.getString(cursor.getColumnIndex("password"));
            if (name.equals(name_cursor) && password.equals(password_cursor)) {
                flag = 1;
            }
        }
        cursor.close();
        database.close();
        if (flag == 1) {
            Toast.makeText(this, "登录成功", Toast.LENGTH_SHORT).show();

            Intent intent = new Intent(this, MainActivity.class);
            startActivity(intent);
            finish();
        } else {
            Toast.makeText(this, "用户名或密码错误,请重新登录",
            Toast.LENGTH_SHORT).show();
        }
    }

    flag = 0;
}

```

代码分析:本身为登录按键绑定的点击事件,先是获取上方填入的账号或是密码,先对方两个 textview 进行非空判断,为空弹出 Toast 说明,账号密码不能

为空，之后创建 SQLiteOpenHelper 对象用于操作 SQLite，查询 SQLite 中是否有上方填入的账号与密码，判断是否匹配返回相对应的 Toast，当两者匹配，显示登录成功，跳转主页面并结束该页面。

注册页面核心代码：

```
public void register(View view) {  
    String name = et_name.getText().toString().trim();  
    String password = et_password.getText().toString().trim();  
    String newpassword = et_newpassword.getText().toString().trim();  
    if(TextUtils.isEmpty(name)||TextUtils.isEmpty(password)||TextUtils.isEmpty  
(newpassword)) {  
        Toast.makeText(this, " 用 户 名 或 密 码 不 能 为 空 ",  
Toast.LENGTH_SHORT).show();  
    }else {  
        if(!password.equals(newpassword)) {  
            Toast.makeText(this, " 密 码 不 一 致 , 请 重 新 输 入 密 码 ",  
Toast.LENGTH_SHORT).show();  
            et_password.setText("");  
            et_newpassword.setText("");  
        }else {  
            MySQLiteOpenHelper_New  mysqliteOpenHelper  =  new  
MySQLiteOpenHelper_New(this, VERSION);  
            SQLiteDatabase  database  =  
mysqliteOpenHelper.getWritableDatabase();  
            ContentValues values = new ContentValues();  
            values.put("name",name);  
            values.put("password",password);  
            database.insert("user_name", null, values);  
            database.close();  
        }  
    }  
}
```

```

        Toast.makeText(this, "    注    册    成    功    ",
        Toast.LENGTH_SHORT).show();

        Intent intent = new Intent(this, LoginActivity.class);
        startActivity(intent);
        finish();
    }
}
}

```

代码分析:依旧是注册按键绑定的监听事件,获得上方三条 `textView` 的内容,先判断其是否不为空的,有则弹出 `Toast` 用户名不能为空,之后判断新密码与确认密码框中是否相同,不同返回密码不一致,请重新输入密码,当上述条件不符合后跳入下一步,创建 `SQLiteHelper` 对象,创建本地 `SQLite` 数据库,用于存放账号与密码,用于登录时进行校验匹配,之后显示注册成功 `Toast`,跳转至登录页面并结束该页面。

4.3.2 数据显示页面

点击登录之后将进入数据显示页面,也是本应用的主页面:



图 4.7 数据显示主页面

核心代码：

（一）创建数据连接

```
public Message getData(String path, Context context) {
    try {
        URL url = new URL(path);
```

```

        HttpURLConnection connection = (HttpURLConnection)
url.openConnection();

        connection.setRequestMethod("GET");
        connection.setConnectTimeout(5000);
        connection.connect();
        int responseCode = connection.getResponseCode();
        if (200 == responseCode) {
            BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));

            ArrayList<Data> Datas = pattrenResult(bufferedReader,
context);

            Message msg = new Message();
            msg.obj = Datas;
            msg.what = MSG_RIGHT;
            return msg;
        } else {
            Message msg = Message.obtain();
            msg.obj = responseCode;
            msg.what = MSG_ERROR;
            return msg;
        }
    } catch (IOException e) {
        Message msg = Message.obtain();
        msg.obj = e;
        msg.what = MSG_ERROR;
        return msg;
    }
}

```

此时传入的 path 路径即为上方 nat123 映射的网页：

<http://15054375.nat123.net:26169/>，本次使用 `HttpURLConnection` 创建网络连接，先是创建 `URL` 对象，绑定 `path` 路径，获取 `HttpURLConnection` 对象，设置请求方式为 `GET`，请求时限 5000 毫秒，之后连接访问网页。获取响应码，当响应码为 200 时即是请求成功，之后开始获取数据，捕捉获取数据时的异常等。返回值为 `Handler` 的 `Message` 对象。

（二）数据处理

创建页面连接之后，获取上方数据，之后进行数据处理：

```
public ArrayList<Data> pattrenResult(BufferedReader bufferedReader, Context
context) throws IOException {
    StringBuilder stringBuilder = new StringBuilder();
    String line;
    while ((line = bufferedReader.readLine()) != null) {
        stringBuilder.append(line);
    }
    String result = stringBuilder.toString().trim();
    Pattern pattern = Pattern.compile("<td>([0-9]{1,2})~([0-9]{1,3})~([0-9]{1,2})~([0-9]{1,2})~([0-9]{1,2}):[0-9]{1,2}:[0-9]{1,2}</td>");
    Matcher matcher = pattern.matcher(result);
    ArrayList<Data> arrayList = new ArrayList<>();
    while (matcher.find()) {
        Data data = new Data();
        data.soilTemperature = Integer.parseInt(matcher.group(1));
        data.soilHumidity = Integer.parseInt(matcher.group(2));
        data.airTemperature = Integer.parseInt(matcher.group(3));
        data.airHumidity = Integer.parseInt(matcher.group(4));
        data.currentTime = matcher.group(5);
        arrayList.add(data);
        recordData2DB(data, context);
    }
}
```

```

        return arrayList;
    }

```

代码分析：获取数据之后，创建字符串缓冲区，将数据以字符串形式存放，之后创建 Pattern，Matcher 对象将该字符串与正则表达式匹配，将正则内部 5 个组的数据分别存入对应数据，在这之前创建了一个 Data 类用于存放这些数据，也创建了一个 ArrayList 用于存放这些 Data 对象，最后返回这个 ArrayList 集合。

（三）ListView 显示与 UI 更新

ListView 布局中的每一条 Item 显示为三条并排的数据：

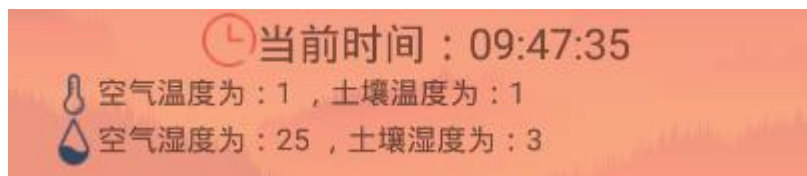


图 4.8 ListView 显示 Item 数据

当上方数据处理完后，传入数据为 ArrayList 的数据集，在更新 ListView 前需要先创建一个 Adapter 对数据进行整理。

核心代码：

```

public View getView(int position, View convertView, ViewGroup parent) {
    View view;
    if(convertView == null){
        view = View.inflate(parent.getContext(), R.layout.item, null);
    }else{
        view = convertView;
    }
    TextView textClock = (TextView) view.findViewById(R.id.Itemclock);
    TextView textTemp = (TextView) view.findViewById(R.id.ItemTextTemp);
    TextView textHumi = (TextView) view.findViewById(R.id.ItemTextHumi);
    Data data = datas.get(position);
    textClock.setText("当前时间： " + data.currentTime);
    textTemp.setText("空气温度为： " + data.airTemperature + " ， 土壤温度为： "
+ data.airTemperature);

```

```

        textHumi.setText("空气湿度为: " + data.airHumidity + " , 土壤湿度为: " +
data.soilHumidity);

        return view;
    }

```

代码分析：每一次 Listview 更新都将传入一次 Item 的 position 参数，根据这个参数，adapter 将更新这个 Item 数据，显示获取 Item 里的三个 textView 控件，然后将之前 ArrayList 里的 data 对象数据以固定格式设置进三个 TextView 控件，最后返回这个 view。

之后在主页面中 listView.setAdapter(adapter);将这个 adapter 设置入 listview，而原本这样更新 UI 的操作为耗时操作不允许在主线程进行，所以创建了一个子线程与 Handler 传递数据，子线程将上述操作运行，获取到数据，以 Message 的形式传递给 Handler，让 Handler 进行更新 UI 的操作：

```

Handler handler = new Handler(){

    @Override

    public void handleMessage(Message msg){

        switch (msg.what){

            case MSG_RIGHT:

                datas = (ArrayList<Data>) msg.obj;

                updateUI(datas);

                Toast.makeText(MainActivity.this, "刷新成功",
Toast.LENGTH_SHORT).show();

                break;

            case MSG_ERROR:

                Toast.makeText(MainActivity.this, "访问网络失败 " + msg.obj,
Toast.LENGTH_LONG).show();

                break;

            default:

                break;

        }

    }

}

```

};

当线程中传递的数据有异常时 `msg.what` 的数据与正常数据不同，此时 **Handler** 将进行不同的操作，弹出不一样的 **Toast**。是作为一种异常处理的方式。

（四）按键页面跳转

页面最上方为三个按键，页面的数据实时更新需要通过刷新按键触发，否则当数据更新较快时，页面不易于查看，影响用户体验，第二个按键为折线图页面，第三个为记录查询页面，点击即跳转，而本次设计为了体现数据页面为主页面，并提高其美观的页面效果，将记录查询与折线图页面采用 **Dialog** 形式显示。

4.3.3 折线图显示页面

Google 官方没有提供属于数据图表的控件，所以绘制折线图需要绘制第三方控件。

显示效果：

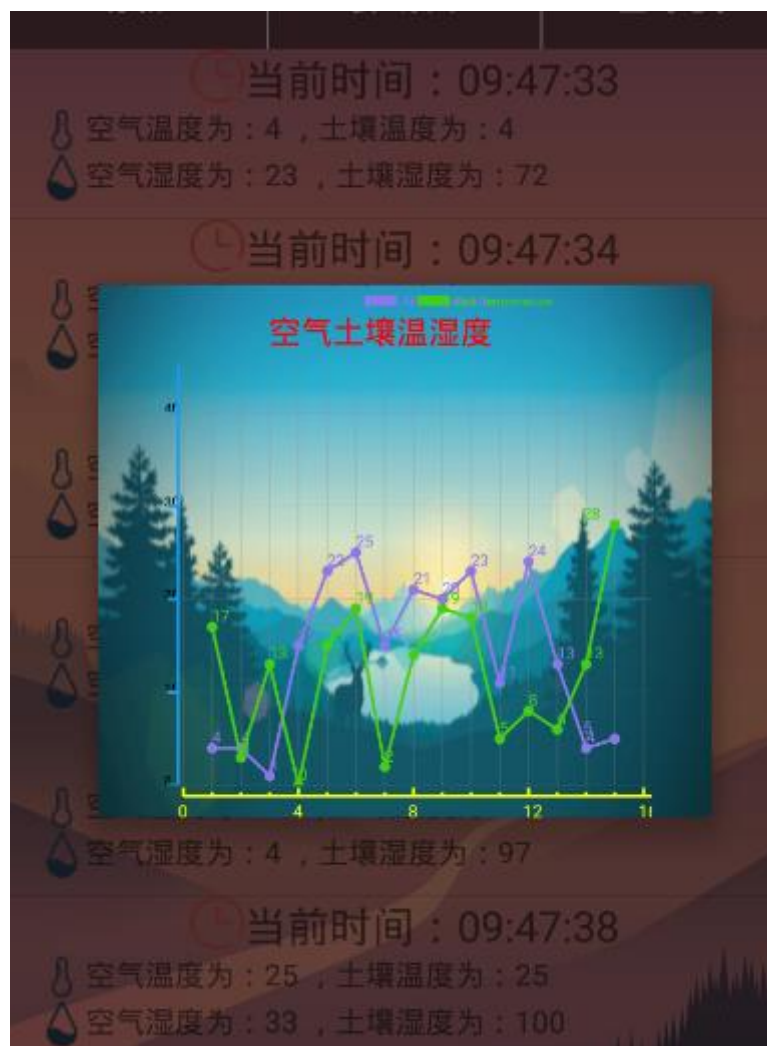


图 4.9 折线图

该图中显示为测试数据，为任意随机数据，显示内容为空气与土壤的温度变化曲线，下方黄线为数据条数，左侧为温度数据从 0 至 50 的纵坐标。

本次设计使用 Github 开源库中的 XCL-charts 第三方库，是为国人开发的 android 开源项目中较为出众的项目，能够绘制各种图表数据。

首先从 github 中下载其开源项目 jar 包：<https://github.com/xblydxj/XCL-Charts>，将其导入项目中，并使用其绘制折线图：

绘制流程：

init();

setTitle();

setAxis();

setData();

addView();

(一) init(): 将图表初始化, 查找自定义控件, 并绘制自定义控件:

```
axisY_2 = new AxisYView_NormalType(this);
```

```
axisX = new AxisXView(this);
```

```
lineView = new LineView(this);
```

```
titleView = new TitleView(this);
```

(二) setTitle();设置图标标题: Common.title = "空气土壤温湿度";

(三) setAxis();设置轴参数:

```
Common.xScaleArray = new  
String[]{"0","1","2","3","4","5","6","7","8","9","10","11","12","13","14","15",  
"16"};
```

```
Common.xScaleColor = Color.YELLOW;
```

```
Common.yScaleArray = new int[]{0,10,20,30,40,50,60};
```

```
Common.levelName = new String[]{"低温","低温","较低","正常","较  
高","高温"};
```

```
Common.yScaleColors = new  
int[]{0xff00ff00,0xffffffff,0xfffffa500,0xffff4500,0xffdc143c,0xffa52a2a};
```

首先是 X 轴的数据, 即为条目, 从 0 至 16 条, 轴颜色为黄色; 之后是 Y 轴, 数据为 0 至 60 的温度数据, 为其分配等级与等级的颜色。

(四) setData();添加数据部分, 这一部分, 由之前更新 ListView 部分的 ArrayList 携带的 data 对象来,

```
MyData data1 = new MyData();
```

```
data1.setName("Air Temperature");
```

```
data1.setData(airTempDatas);
```

```
data1.setColor(0xff8d77ea);
```

```
MyData data2 = new MyData();
```

```
data2.setName("Soil Temperature");
```

```
data2.setData(soilTempDatas);
```

```

data2.setColor(0xff43ce17);

Common.DataSeries = new ArrayList<MyData>();

Common.DataSeries.add(data1);

Common.DataSeries.add(data2);

```

由于是两条温度数据，所以添加数据 data1 与 data2，并为其设置数据名与对应的颜色。

（五）addView();将上述添加的数据等添加自定义控件至初始化出的 View 中

```

private void addView(){
    int width = 0;
    width = Common.viewWidth;
    xy.y = Common.viewHeight - Common.layoutHeight;
    lineView.initValue(width, Common.viewHeight, true);
    lineView.scrollTo(0, xy.y);
    axisY_2.initValue(Common.viewHeight);
    axisY_2.scrollTo(0, xy.y);
    axisX.initValue(width, Common.viewHeight);
    axisX.scrollTo(0, xy.y);
    axisYLayout.removeAllViews();
    axisYLayout.addView(axisY_2);
    axisXLayout.removeAllViews();
    axisXLayout.addView(axisX);
    threndLine_Layout.removeAllViews();
    threndLine_Layout.addView(lineView);
    title_layout.removeAllViews();
    title_layout.addView(titleView);
}

```

传入 Y 轴高度，与 X 轴的横向高度，之后分别将 X，Y，连线，以及标题传入原布局。

4.3.4 记录查询页面

记录查询页面同样为 Dialog 形式的 Activity，该设置在 AndroidManifest.xml 中配置：

```
<activity android:name=".RecordActivity"
```

```
android:theme="@style/Theme.AppCompat.Light.Dialog.MinWidth"/>
```

页面效果：

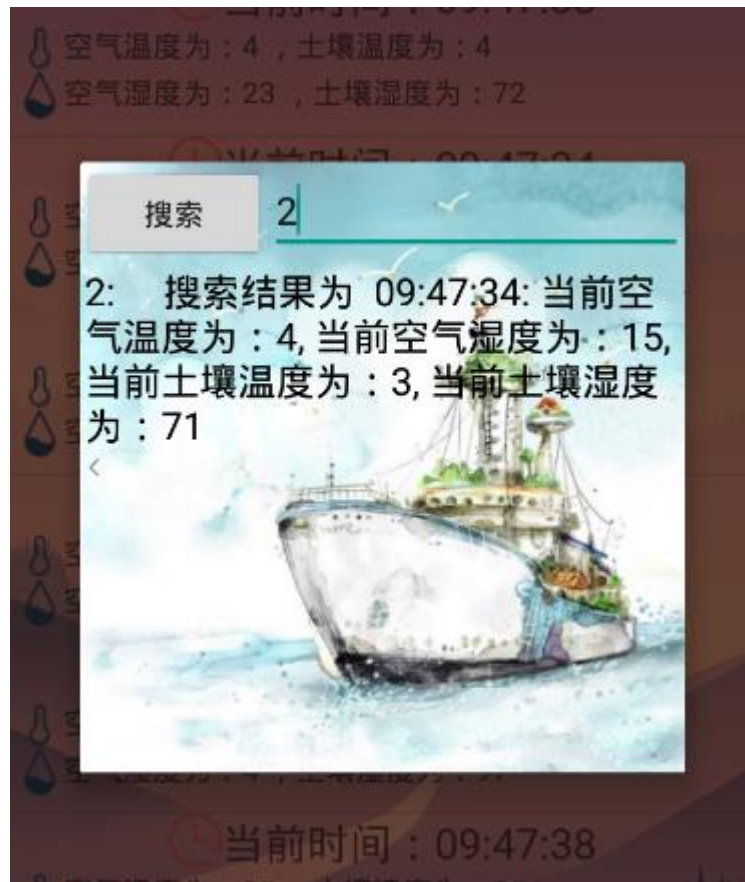


图 4.10 记录查询页面

查询条件设置为 id 号，即是在存储入 MySQL 时的每条数据所匹配的 Id 号，根据 id 号查询其数据。

```
public void search(View view) {  
    if (TextUtils.isEmpty(searchFor.getText())) {  
        Toast.makeText(this, " 搜 索 内 容 不 能 为 空 ",
```



```

Toast.LENGTH_SHORT).show();

    } else {

        String idstr = searchFor.getText().toString().trim();

        int id = Integer.parseInt(idstr);

        MySQLiteOpenHelper queryDataDB = new MySQLiteOpenHelper(this,
VERSION);

        SQLiteDatabase database = queryDataDB.getWritableDatabase();

        Cursor cursor = database.query("data", null, "id=?", new String[]{idstr}, null,
null, null);

        if (cursor.isNull(id)) {

            result.setText("无搜索结果");

            Toast.makeText(this, "无搜索结果", Toast.LENGTH_SHORT).show();

            return;

        }

        while (cursor.moveToNext()) {

            String currentTime = cursor.getString(cursor.getColumnIndex("currentTime"));
            int soilTemperature = cursor.getInt(cursor.getColumnIndex("soilTemperature"));
            int soilHumidity = cursor.getInt(cursor.getColumnIndex("soilHumidity"));
            int airTemperature = cursor.getInt(cursor.getColumnIndex("airTemperature"));
            int airHumidity = cursor.getInt(cursor.getColumnIndex("airHumidity"));

            resultStr = id + ":    搜索结果为  " + currentTime
                + ": 当前空气温度为: " + airTemperature
                + ", 当前空气湿度为: " + airHumidity
                + ", 当前土壤温度为: " + soilTemperature
                + ", 当前土壤湿度为: " + soilHumidity
                + "\r\n";

        }

        result.setText(resultStr);

        cursor.close();

        database.close();

```

```
}  
}
```

代码分析：依旧是 `search` 按键绑定的监听事件，匹配搜索框内容，为空弹出 `Toast`，`id` 号不能为空，在查询数据之前，在主页面更新 `UI` 部分时，也同时创建了一个 `SQLite` 用于存放每一条数据，并且配有其相对应的 `id` 号，而在记录查询页面，也同时操作 `SQLiteHelper` 来查询该数据库，匹配 `id` 号将内部数据显示在下方的 `TextView` 上，之后释放资源。

4.4 本章小结

本章详细介绍了 `Android` 手机应用程序的设计流程，从登陆注册到数据显示部分，再到使用第三方开源库的折线图页面，以及最后的记录查询页面，每一部分都加入了部分代码讲解介绍。

第5章 总结与展望

5.1 总结

本文以阳春砂要用植物种植栽培传感器网络为基础，设计了一个基于 Java 语言与 android 平台技术的后台数据系统与 Android 手机应用程序。当前众多智能灌溉系统依旧采用老旧的 PC 端检测监控数据，而本次设计也从新的，更加简便高效的方向去发展，开发智能手机端的应用程序对智能农业的设备进行实时检测监控。

论文主要研究内容和成果总结如下：

- (1) 介绍了 Java 的非核心类库串口通信，以及 Java 语言与众多数据库相联系的接口 JDBC，使用它们获取来自传感网络的数据，并存储于 MySQL 数据库。
- (2) 介绍了关于 MySQL 数据库的现状，以及当前的使用，构建了 Php 网页端数据接口，并提供向外部网络。
- (3) 设计开发了 Android 手机端应用程序，实时获取传感器网络的数据，美观简洁的显示于智能手机端中，并提供了登录注册功能。使用了开源项目用于绘制自定义控件折线图，以及一个能够查询过去记录的页面，内部依旧使用了 SQLite 数据库的相关开发。

5.2 展望

本系统的结构较为复杂，涉及面太泛，融合了几个不同领域的技术，而也由于这结构复杂，在学习过程中能够学习到更多的优化手段。简化本项目：

- (1) 本文采用了 Java 语言与 MySQL 数据库，并编写了 php 页面用于显示 MySQL 数据库内容，这样太过繁琐。在实际开发中更多会使用的是 Javaweb 端的 Tomcat 服务器，以及使用 eclipse 构建 javaweb 项目，使用 jsp 书写页面，但依旧会使用 Comm 与 JDBC 来获取数据与创建与数据库的连接，但内部网络将会使用 8080 端口的

Tomcat，之后再映射。

- (2) 上述过程主要针对数据源由串口输入的，正式项目中传感器网络将会使用网络网关传输数据，上传至云端服务器存储数据，这样也能够更好的提供 Android 应用程序数据接口。
- (3) 本次设计为该方向设计的雏形，Android 手机端的数据资源更容易获取之后能够将注册登录的账号密码匹配部分同样部署于远程云端的服务器，更加正式，也能够记录更多的数据，记录查询依照时间或者更多的信息进行查询，也能够查询更多更长时间的记录。本次设计数据量较小，并未实现该功能。

参考文献

- [1] 廖清棋. 浅谈有关节水农业的发展问题[J]. 福建水土保持, 2001(2):6 – 7.
- [2] 江铭. 海水淡化，为世界解渴——向大海要水[J]. 环境, 2012(7):5.
- [3] 刘玉乐. 天津市灌溉水利用率测算研究[J]. 中国农业科学院, 2010:1 – 2.
- [4] 孙景生, 康绍忠. 我国水资源利用现状与节水灌溉发展对策[J]. 农业工程学报, 2000(2) :6.
- [5] 张观山. 果园智能灌溉控制系统的研发[D]. 山东大学, 2014(6) :6 – 7.
- [6] 张兵, 袁寿其, 李红, 丛小青, 袁建平. 基于模糊决策理论的冬小麦精量灌溉智能系统的研究[J]. 中国农村水利水电, 2006(5) :11.
- [7] 曹金源. 基于无线传感器网络的设施农业温湿度检测系统设计与实现[J]. 中国农业科学院, 2013:6 – 7.

- [8] 杨林楠, 郇鲁涛, 林尔升, 彭琳, 李文峰. 基于 Android 系统手机的甜玉米病虫害智能诊断系统[J]. 农业工程学报, 2012:7.
- [9] 周立, 张书慧. 基于 Android 平台的茶园环境监控系统设计与实现[J]. 南京工业职业技术学院学报, 2014:7.
- [10] 施建忠, 张永. 开源数据库刀锋开刃[J]. 信息系统工程. 2005:45 – 47..
- [11] 高素宁, 张春云, 石俊峰, 刘燕. 利用 MySQL 数据库技术进行决策气象服务[J]. 河南气象, 2006(4) :7.
- [12] 陈锦伟, 基于 MySQL 的空间数据库关键技术研究[J]. 南京邮电大学, 2013:6 – 7.
- [13] 孙丽婷, 基于无线传感器网络的农业大棚监控系统设计[J]. 大连理工大学, 2013:7.
- [14] 陈杉. 试论设施农业远程监控技术[J]. 天津农林科技. 2012(2) :7.
- [15] 赵情融. 基于 Android 平台的个人移动地图应用[D]. 华中科技大学, 2011:8.
- [16] 郑蓉. 基于 Java 语言的串口通信[J]. 科技致富向导. 2014(11):1 - 4.
- [17] 周国华, 唐土生. 基于 Java 穿行通信研究[J]. 福建电脑, 2006(6):1 - 2.
- [18] 苏淑文. 扩展 NXD 的 GML 控件数据库圆形系统的设计与实现[J]. 北京航空航天大学, 2004. 5 – 6.
- [19] Bruce Eckel. Thinking in Java, Fourth Edition. 2007, 2(2):11 - 31.
- [20] Brestovac G, Grbac T G, Grgurina R. Development environment for Android application development: An experience report. MIPRO, 2011 Proceedings of the 34th International Convention, 2011:1 – 3.

致 谢

通过这一阶段的努力，我的毕业论文终于完成了，这意味着大学生活即将结束。回首既往，自己一生最宝贵的时光能于这样的校园之中，实是荣幸之极。在这四年的时间里，我在学习上和思想上都受益非浅。这除了自身努力外，与各位老师、同学和朋友的关心、支持和鼓励是分不开的

在本论文的写作过程中，我的导师蔡卫明老师，从最初的选题，到资料收集，到写作、修改，到论文定稿，严格把关，循循善诱，在此我表示衷心感谢。同时我还要感谢在我学习期间给我极大关心和支持的各位老师以及关心我的同学和朋友。正是由于他们，我才能在各方面取得显著的进步，在此向他们表示我由衷的谢意，并祝所有的老师培养出越来越多的优秀人才，桃李满天下！也祝愿我的同学的在以后的道路上越走越宽，前程似锦！

在这里，我也要借这个机会对我的父母说一声：“谢谢你们！”正是由于多年来有你们的鼓励和支持，我才能一路坚持下来走到今天这个地方。感激之情无法言语表达，只是想说：如今已长大成人，希望父母能够安享晚年，不再操劳！

时间的仓促及自身专业水平的不足，整篇论文肯定存在尚未发现的缺点和错误。恳请阅读此篇论文的老师、同学，多予指正，不胜感激！