**J I A N G S U U N I V E R S I T Y**

**本 科 毕 业 论 文**

个人医学资料管理软件的设计与实现

**Design and Implementation of Personal Medical Data**

**Management Software**

学 院 名 称 ： 计算机科学与通信工程

专 业 班 级 ：

学 生 姓 名 ：

指导教师姓名 ： 年 轶

指导教师职称 ： 讲 师

年 月

个人医学资料管理软件的设计与实现

专业班级： 学生姓名：

指导教师：年轶 职 称 ：讲师

摘要 随着人们生活水平的提高，越来越多的人开始关注自身的健康状况，看病就医和定期体检已经成为人们日常生活方式的一部分，这就会产生许多重要的关乎个人健康的医学资料，越来越多的资料不仅在存放、查找等方面有诸多不便，也浪费了不少人力以及空间。目前，随着科技的进步，许多人选择使用计算机存储和管理自己的个人资料，相比较于传统的纸张，计算机系统具有安全、高效、便于检索和统计等优点。个人医学资料管理软件，能方便用户将自己的个人医学资料存储在计算机系统中，即解决了纸质资料储藏的一系列问题，又能为个人健康和后续诊疗提供科学的依据。

本文设计并实现了一个基于.NET的个人医学资料管理软件，系统采用C#语言编写，后台使用SQL Server数据库，并使用了三层架构将用户界面和数据操作分离，体现了高内聚，低耦合的特点。根据软件工程的基本原理，系统的开发包括系统分析，设计和实现三个阶段。在系统分析阶段，本文提出了系统目标，从技术、操作、经济、法律、环境等多方面进行了可行性分析，并通过用例图和数据流图对系统功能进行了详细的分析；在系统设计阶段，按照模块化设计的思想，给出了系统总体模块图，同时使用E-R图对数据库进行了设计，并使用类图和序列图介绍了每个模块的设计原理和实现流程；在系统实现阶段，本文给出了系统功能描述和系统界面。

本系统分为个人信息模块，电子类资料模块，图像类资料模块和提醒模块。其中，电子类资料模块实现了对个人病历、处方、体检报告和检验报告的增加、查询、删除和修改；图像类资料模块可以将CT片、X光片、核磁共振片等图片存储在系统中，并实现了增删改查等功能；提醒模块允许用户设置提醒，提示用户及时参加复诊、体检等；此外，用户还能将系统中的资料以表格或图片的形式保存到本地计算机中，方便打印和携带。测试结果表明，本系统实现了个人医学资料管理的主要功能，基本满足用户管理自己的医学资料的需求。

关键词：个人医学资料管理 C# 三层架构 SQL Server

**Design and Implementation of Personal Medical Data Management Software**

**Abstract** With the improvement of people's living standards, more and more people began to pay attention to their own health status, medical treatment and regular physical examination has become a part of people's daily life, which will produce a lot of important medical data that related to personal health. The increasing amounts of data are not only inconvenient for storing and searching, but also wasting a lot of manpower and space. At present, with the advancement of science and technology, many people choose to store and manage their own personal data with computer, which is safe, efficient, easy to retrieve and statistics compared to traditional methods. With the personal medical data management software, users can store their own personal medical data in the computer system, which not only solves a series of problems in the paper data storage, but also provides a scientific basis for personal health and follow-up diagnosis and treatment.

In this paper, we designed and implemented a personal medical data management software based on .NET platform. The system uses SQL Server database and C# as the programing language, and uses the three-tier architecture to separate the user interface and data operation, which embodies the characteristics of high cohesion and low coupling. According to the basic principles of software engineering, the system development includes three stages —system analysis, design and implementation. In the stage of system analysis, the system goal is introduced, and we not only analyze the feasibility of the system from technology, operation, economy, law and environment but also make a detailed analysis of the system function through the use case diagram and data flow diagram. In the system design stage, we give the general module according to the idea of modular design and we design the database with E-R diagram. What’s more, the design principle and implementation process of each module are introduced with class diagram and sequence diagram. In the system implementation stage, this paper gives the system function description and system interface.

The system is divided into personal information module, electronic data module, image data module and reminder module. Among them, the electronic data module implements the increase, query, deletion and modification of individual medical records, prescriptions, physical examination reports and inspection reports and the image data implements the same function of CT images, X-ray images, NMR images and so on. The reminder module allows users to set reminders so that they can return visit or participate in physical examination in time; In addition, the user can also save any data in the system to their local computer in the form of table or picture, which is easy to print and carry. The test results show that the system implements the main functions of personal medical data management, and basically meets the needs of users to manage their own medical data.

**Key words** personal medical data management C# three-tier architecture SQL Server

目 录

[第一章 绪论 1](#_Toc484429520)

[1.1 研究背景 1](#_Toc484429521)

[1.2 国内外研究现状 1](#_Toc484429522)

[1.3 研究内容 2](#_Toc484429523)

[1.4 研究的目的与意义 2](#_Toc484429524)

[1.5 论文的结构 3](#_Toc484429525)

[第二章 相关技术介绍 4](#_Toc484429526)

[2.1.NET平台和C#语言 4](#_Toc484429527)

[2.2 SQL Server技术 4](#_Toc484429528)

[2.3 三层架构 5](#_Toc484429529)

[第三章 系统分析 7](#_Toc484429530)

[3.1 系统目标 7](#_Toc484429531)

[3.2 可行性分析 7](#_Toc484429532)

[3.2.1 技术可行性 7](#_Toc484429533)

[3.2.2 操作可行性 7](#_Toc484429534)

[3.2.3 经济可行性 8](#_Toc484429535)

[3.2.4 法律可行性 8](#_Toc484429536)

[3.2.5 环境可行性 8](#_Toc484429537)

[3.3 功能需求 8](#_Toc484429538)

[3.4.1 用户信息管理 9](#_Toc484429539)

[3.4.2 电子资料管理 11](#_Toc484429540)

[3.4.3 图像资料管理 15](#_Toc484429541)

[3.4.4 提醒管理 18](#_Toc484429542)

[3.4 数据流分析 20](#_Toc484429543)

[3.5 性能需求 22](#_Toc484429544)

[3.6 开发和运行环境 23](#_Toc484429545)

[3.6.1 开发环境 23](#_Toc484429546)

[3.6.2 运行环境 23](#_Toc484429547)

[第四章 系统总体设计 24](#_Toc484429548)

[4.1 系统模块划分 24](#_Toc484429549)

[4.2 数据库设计 25](#_Toc484429550)

[4.2.1 概念结构设计 25](#_Toc484429551)

[4.2.3 逻辑结构设计 30](#_Toc484429552)

[第五章 系统详细设计 36](#_Toc484429553)

[5.1 系统架构设计 36](#_Toc484429554)

[5.2 模块设计 37](#_Toc484429555)

[5.2.1 数据库操作类 37](#_Toc484429556)

[5.2.2 个人信息管理模块 37](#_Toc484429557)

[5.2.3 病历管理模块 39](#_Toc484429558)

[5.2.4 处方管理模块 42](#_Toc484429559)

[5.2.5 体检报告管理模块 44](#_Toc484429560)

[5.2.6 检验报告管理模块 46](#_Toc484429561)

[5.2.7 图像资料管理模块 47](#_Toc484429562)

[5.2.8 提醒管理模块 48](#_Toc484429563)

[第六章 系统实现 50](#_Toc484429564)

[6.1 三层架构搭建 50](#_Toc484429565)

[6.2 模块功能实现 50](#_Toc484429566)

[4.2.1 个人信息管理模块 50](#_Toc484429567)

[4.2.2 病历管理模块 52](#_Toc484429568)

[4.2.3 处方管理模块 54](#_Toc484429569)

[4.2.4 体检报告管理模块 57](#_Toc484429570)

[4.2.5 检验报告管理模块 59](#_Toc484429571)

[4.2.6 图像资料管理模块 60](#_Toc484429572)

[4.2.7 提醒管理模块 62](#_Toc484429573)

[总 结 64](#_Toc484429574)

[致 谢 65](#_Toc484429575)

[参考文献 66](#_Toc484429576)

# 第一章 绪论

## 1.1 研究背景

随着人们的生活水平和文化素质的不断提高，人们越来越关注自己的生活质量，从观念上已经转化为对自身健康状况关注度的提高，一旦身体有任何不适，都会第一时间选择去医院就诊，而现代医学的进步会针对大多数疾病给病人做出详细的检查和具体的诊治方案。随着各种健康保险的发展，人们的疾病预防意识也逐步提高，健康体检也为越来越多的人所熟知，通过体检，人们能及早地发现健康状况出现的危机，消除疾病产生的危险因素，也能较早地发现疾病并进行早期及时的治疗，看病就医和参加体检已经成为人们生活方式的一部分[1]。

在看病就医和健康体检等一系列过程中，会产生许多相关的资料，比如病历、处方、体检报告等。这些资料记录着人们生命体征的变化，以及自身从事的与健康相关的行为，包括个人生活习惯、家族病史、既往病史、现病史、体检结果以及疾病产生的原因、发展过程、诊治方案和治疗效果[2]，对个人健康状况的发展有重要意义。然而，普通人所拥有的这些个人医学资料一般都是纸质资料，少量的纸质资料在整理、存放方面并没有太大的问题，但是时间的积累会带来越来越多的纸质资料，这些资料如果没有及时整理，在需要时查找起来异常困难。所以，个人医学资料想要在个人健康方面发挥作用，分门别类地整理、存放显得很重要。而纸质资料的整理，存放会浪费许多的人力和空间，某种程度上成为人们生活的累赘。

随着科技的进步，计算机的诞生大大方便了人们的生活，也使各种资料的存储和检索变得更加便利。采用计算机存储资料，颠覆了以往纸质资料手工分类存档、手工检索的格局，避免了存储不当带来的隐患，同时也节省了纸质资料存放的空间。基于此，我们设想采用计算机存储个人的医学资料，开发一个个人医学资料管理软件，能将个人医学资料按类别存储在计算机中，需要时可以快速地检索，解决纸质资料存储的一系列问题的同时也能提高人们的生活质量。

## 1.2 国内外研究现状

目前网上的资料管理软件，较为人所熟知有为知笔记，针式PKM等。它们都能实现信息的收集，整理，以及快速查找，不过，它们最主要的用处还是知识管理。其余的还有消防档案管理系统，健康档案管理系统，人事档案管理系统等，它们是对于某一个特定的方面专门设计的资料管理系统，有很强的针对性，并不适用于医学资料管理。在已有的这些资料管理软件中，医学资料管理软件并不多，而且绝大部分还是针对诊所、医生或医学工作者设计，不能满足普通用户的需求。

在国外，这类软件较为著名的有MyPatients(临床医生个人患者群管理系统)，这依旧是一个针对医生设计的患者资料管理软件，满足不了普通用户的需求；另外有一款软件叫做MyMedical，它是一个针对家庭设计的个人医学资料管理软件，它将所有的医疗记录保存在服务器端，医生和家人可以跟踪这些数据掌握用户的健康状况，与本课题所研究内容较为接近。不过，它只能在Mac和Iphone端使用，并且需要花费不少的金钱购买，而且需要医生和患者共同使用才能发挥其最大作用，比较适合国外那种私人医生模式，国内的普通用户可能不会因为它在资料管理方面带来的一点便利去支付高昂的成本。总的来说，目前并没有一款很好地满足普通用户在个人医学资料管理方面需求的软件。

## 1.3 研究内容

本文目的在于设计并实现一个个人医学资料管理软件。个人医学资料，主要可分为电子类资料（包括病历、处方、体检报告、检验报告等）和图片类资料（包括X光片、CT片、核磁共振片等），设计并实现该软件，主要内容是实现对上述资料的管理。本文主要研究如何针对上诉几类资料设计完善的数据表以及数据存储和查询机制，既要方便用户将个人医学资料存入系统中，又要在需要时能快速地检索出所要的资料。另外，对于存入的资料，要方便用户随时修改，并且只能由用户本人操作，保证资料的完整性和安全性。最后，通过添加定时提醒功能，提醒用户及时参加体检或复诊，完善系统功能，保证用户体验。

## 1.4 研究的目的与意义

个人医学资料对个人健康保健有及其重要的作用，通过比较某段时间的检查资料和数据，人们可以直观地了解这段时间的健康趋势，从而制定出更合理的锻炼或保健计划。然而，纸质资料的保存，需要一定的空间，特别是不同的资料大小不一，保存起来更不方便，另外还要防止纸张磨损、老化，以及注意防潮，防火，防蛀等，有许多限制条件。另一方面，如果这些资料没有合理地进行分类，查找起来也有不小的难度。所以，设计本软件的目的旨在解决上述问题。该软件能将病历、处方、体检报告、检验报告以及其他图像类资料按类别存储在计算机中，软件设置的数据项合理，能涵盖资料中的大部分有用信息，并且支持按医院、科室、时间等检索查询，使用方便。

设计并实现个人医学资料管理软件，能将用户所有的个人医学资料存储在计算机中，解决了纸质资料存取的一系列限制，而且，该软件的存储体系和备份机制能实现大量存储和及时存取的统一，占用空间小，存储容量大。随着人们健康观念的提升和各种健康信息管理手段的丰富，传统以疾病为中心的医疗模式也逐渐被以健康为中心、以人为中心的医疗模式取代[3]。从被动的有病求医到主动的预防保健的转变，更需要个人以往的医学资料提供有力的支持。反映出用户在某个时间段内身体状况的变化趋势，为个人健康状况分析和后续的诊疗提供依据，及时提醒用户参加复诊、体检等，是本软件设计的意义所在。

## 1.5 论文的结构

本文介绍了一个个人医学资料管理软件的开发过程，基于软件工程的基本原理，系统的开发分为分析、设计和实现三个阶段。

第一章简要介绍了课题背景，国内外的研究现状，研究的目的和意义。

第二章介绍了与系统开发相关的开发工具和技术。

第三章为系统的分析，提出了系统开发的目标，并进行了可行性分析，业务分析，功能需求分析、数据流分析和性能能需求分析。

第四章为系统的总体设计，介绍了系统的模块划分和数据库设计。

第五章为系统的详细设计，介绍了系统架构设计，并用类图和序列图对每个模块设计做了详细介绍。

第六章为系统的实现，给出了系统功能描述和系统界面。

# 第二章 相关技术介绍

## 2.1.NET平台和C#语言

.NET就是微软用来实现XML，Web Services，SOA（面向服务的体系结构service-oriented architecture）和敏捷性的技术[4]。技术人员一般将微软看成一个平台厂商。微软搭建技术平台，而技术人员在这个技术平台之上创建应用系统。从这个角度，.NET也可以如下来定义：.NET是微软的新一代技术平台，为敏捷商务构建互联互通的应用系统，这些系统是基于标准的，联通的，适应变化的，稳定的和高性能的。从技术的角度，一个.NET应用是一个运行于.NET Framework之上的应用程序。（更精确的说，一个.NET应用是一个使用.NET Framework类库来编写，并运行于公共语言运行时Common Language Runtime之上的应用程序[4]），即.NET提供了系统运行所必须的.NET Framework类库和公共语言运行时。好比于我们人类的生存必须有氧气，程序的运行也需要一些特定的条件，这就是.NET平台。

C#语言是微软公司推出的运行在.NET平台之下的一种简洁、类型安全的面向对象的编程语言。它使得程序员可以快速地编写各种基于.NET平台的应用程序，.NET提供了一系列的工具和服务来最大化地开发利用计算与通信领域[6]。C#语言语法简洁，不允许直接操作内存，去掉了C++中的指针操作；它与Web紧密结合，支持绝大多数Web标准，如HTML、XML、SOAP；因为C#遵循.NET公共语言规范(CLS)，从而保证与其他语言开发的组件兼容；此外，C#提供了完善的错误处理和异常处理机制，能使程序更加健壮。C#语言是.NET平台的代表语言，.NET平台可以把C#语言翻译成机器语言以达到计算机可以正确运行我们所写的程序的目的。

## 2.2 SQL Server技术

SQL（Structured Query Language），即结构化查询语言，是关系型数据库的标准语言，是一个通用的、功能极强的关系数据库语言。它的主要功能就是同各种数据库建立联系，进行沟通[7]。SQL集数据查询（Data Query）、数据操纵（Data Manipulation）、数据定义（Data Definition）和数据控制（Data Control）功能与一体，例如，SQL语句可以用于更新数据库中的数据，从数据库中提取数据等。目前，绝大多数流行的关系型数据库管理系统，如Oracle、Sybase、Microsoft SQL Server、Access等都采用了SQL语言标准。虽然很多数据库都对SQL语句进行了再开发和扩展，但是包括select、insert、update、delete、create以及drop在内的标准SQL命令仍然可以被用来完成几乎所有的数据库操作[7]。

SQL Server 是一个关系数据库管理系统。它最初是由Microsoft、Sybase 和Ashton-Tate三家公司共同开发的，于1988 年推出了第一个OS/2版本[7]。在Windows NT 推出后，Microsoft与Sybase 在SQL Server 的开发上就分道扬镳了，Microsoft 将SQL Server移植到Windows NT系统上，专注于开发推广SQL Server 的Windows NT 版本。Sybase 则较专注于SQL Server在UNⅨ操作系统上的应用。本系统的开发采用的SQL Server版本为SQL Server 2008，这是一个重大的产品版本，它推出了许多新的特性和关键性的改进，使得它成为一个强大而且全面的SQL Server版本。

## 2.3 三层架构

在软件体系架构设计中，分层式结构是最常见也是最重要的一种结构。微软推荐的分层式结构一般分为三层，从下到上依次是：数据访问层(DAL)、业务逻辑层（又或称为领域层）(BLL)、表示层(UI)[8]，如图2-1所示。



图 2- 1 三层系统的分层结构

1.数据访问层：主要是对原始数据（数据库或者文本文件等存放数据的形式）的操作层，而不是指原始数据，也就是说，是对数据的操作，而不是数据库，具体为业务逻辑层或表示层提供数据服务。

2.业务逻辑层：主要是针对具体的问题的操作，也可以理解成对数据层的操作，对数据业务逻辑处理，如果说数据层是积木，那逻辑层就是对这些积木的搭建。

3.表示层：主要表示Web方式，也可以表示成WinForm方式，表示层主要指的是用户界面。

在三层架构中，每一层（UI→BLL→DAL）之间的数据传递（单向）都是通过参数来传递的，这样就构造了三层之间的联系，完成了功能的实现。一般参数传递的方式为通过变量进行传递，但对于大量的数据来说，用变量作参数有些复杂，因为参数量太多，容易混淆。比如要把员工信息传递到下层，信息包括员工号、姓名、年龄、性别、工资等等，用变量作参数的话，那么方法中的参数就会很多，极有可能在使用时将参数匹配搞混。因此，引入实体（Entity）的概念，实体是对属性的封装，一个实体映射了数据库中的一张表，一个属性映射了表中的一个字段，例如，将员工封装成一个实体Person，其属性包括员工号、姓名、性别、工资等员工的信息。在层与层之间传递数据时，将实体作为参数进行传递，就不用考虑参数匹配的问题，需要用到实体的哪个属性就直接从实体中取出即可，这样做不仅方便，也提高了效率，所以，三层架构逐渐演变成如图2-2所示的结构。



图 2- 2 三层架构的演变

使用三层架构设计应用程序，程序结构清晰，系统可维护性和可扩展性高，体现了“高内聚，低耦合”的特点。利于开发任务同步进行，容易适应需求变化，但也降低了系统的性能，这是不言而喻的：如果不采用分层结构，很多业务可以直接访问数据库，以此获取相应的数据，而如今却必须通过中间层来完成[8]。

# 第三章 系统分析

## 3.1 系统目标

通过对一般资料管理软件的分析，本系统计划实现的目标分为以下方面[15]：

1.界面设计美观友好，页面简洁大方，资料信息显示直观，完整，准确，通过简单的鼠标点击和键盘输入即可使用该软件。

2.用户能够对自己的个人医学资料进行增加、删除、修改等操作，并能快速查找到某一资料。

3.用户能够增加提醒，包括设置提醒时间、地点等，软件要能在提醒时间到来时提示用户。

4.建立数据库，不同的资料建立不同的数据表，保证资料存储快捷，准确；对用户建立用户表，只有注册后的用户才能使用该软件，保证资料的安全性。

5.系统运行安全可靠，最大地实现易维护性和易操作性。

## 3.2 可行性分析

可行性分析是在进行初步调查后所进行的对系统开发必要性和可能性的研究，是在当前组织内外的具体条件下，这个信息系统的研制工作是否具备必要的资源及其他条件。本系统的可行性分析可从以下几个方面考虑：

### 3.2.1 技术可行性

技术可行性是指在当前条件下，系统开发拟采用的开发工具或开发技术能否满足系统的需求。本系统拟采用C#语言编写，采用SQL Server数据库，使用微软推荐的三层架构，所用的开发技术均数当前流行的技术，且技术成熟，安全，效率高，能够保证所有系统要求的功能都被实现。因此，本系统在技术上是完全可行的。

### 3.2.2 操作可行性

操作可行性是指系统在用户操作方面的便利性。本系统采用菜单式操作，将所有数据交互简化为按钮操作，界面设计简单，美观，大方，用户无需具有数据库或其他相关计算机知识也能快速上手该软件，因此，本系统在操作上完全可行。

### 3.2.3 经济可行性

经济可行性是指系统开发的成本是否合理和能否带来一定的经济效益。在人力成本上，本系统功能简单，开发技术成熟，少量的人力在较短的时间内即可完成系统的开发，没有太大的人力成本；在开发工具上，本软件采用的Microsoft Visual Studio 2013社区版允许学生免费使用，SQL Server 2008使用方便，功能完善，效率较高，一定程度上节约了开发成本。除此之外，在其他方面不会有很大的开销，相反，若软件能够投入使用，还可以带来一定的经济效益，在经济上完全可行。

### 3.2.4 法律可行性

法律可行性是指系统的开发是否会违反相关的法律，是否存在侵权等责任问题。本系统采用的开发工具均为正版，且软件为原创，不会产生版权纠纷。而且，本个人医学资料管理软件的开发、使用没有政策和法律方面的限制，因此本系统在法律上完全可行。

### 3.2.5 环境可行性

环境可行性是指系统的开发、使用是否会对环境造成破坏。由于本系统在开发、使用时采用的均是计算机资源，除此之外没有其他物理需求，不会造成环境污染或破坏。相反，本系统可提醒人们关注自身健康，倡导绿色生活，因此，本系统在环境上完全可行。

## 3.3 功能需求

本软件的主要功能是用户对自己的个人医学资料进行管理，系统的参与者只有用户。用户的个人医学资料包括像病历、处方这样的纸质资料，也包括像X光片、CT片这样的图像类资料，所以软件要能实现对这两类资料进行增加、删除、修改和查找等功能。对于图片类资料，可不必细分，他们在软件中的存储方式都是图片；而对于纸质资料，因为每种资料的格式各不相同，所以要进行划分，使用不同的数据表存储不同类别的资料。因此，资料管理可分为电子资料管理和图像资料管理两类，而电子资料管理具体又划分为病历管理，处方管理，体检报告管理和检验报告管理，考虑到医学资料众多，无法全部考虑，因此只选取了四种用户接触较多的类型，对于其他资料，可以以图片的方式存储。

此外，用户还要能管理自己的用户名、密码和基本信息。软件要实现提醒功能，需要用户能对提醒进行添加、删除等，基于此，可以得到系统总用例如图3-1所示。



图 3- 1 系统总用例图

下面分别介绍总用例图中的具体用例。

### 3.4.1 用户信息管理

用户信息管理的用例图如图3-2所示，主要包括用户注册、登录以及修改个人信息。



图 3- 2 用户信息管理用例图

1.用户注册

|  |  |
| --- | --- |
| 用例名称 | 用户注册 |
| 功能描述 | 用户输入个人信息，完成注册 |
| 前置条件 | 无 |
| 后置条件 | 系统将用户信息添加至数据库 |
| 主事件流 | 1.用户在注册界面输入用户名、密码及个人信息，并提交  2.系统判断用户信息是否完整，合法  A1：用户信息不完整或不合法  3.系统判断用户名是否存在  A2：用户名已存在  4.注册成功，返回登录界面 |
| 扩展事件流 | A1：用户信息不完整或不合法  (1)系统给出相应提示信息  (2)返回主事件流第1步  A2：用户名已存在  (1)系统给出相应提示信息  (2)返回主事件流第1步 |

2.用户登录

|  |  |
| --- | --- |
| 用例名称 | 用户登录 |
| 功能描述 | 用户输入用户名和密码，登录系统 |
| 前置条件 | 用户注册成功 |
| 后置条件 | 用户登录成功 |
| 主事件流 | 1.用户输入用户名和密码，并提交  2.系统判断用户名是否存在  A1：用户名不存在  3.系统判断密码是否正确  A2：密码错误  4.登录成功，显示系统主界面 |
| 扩展事件流 | A1：用户名不存在  (1)给出相应提示信息  (2)返回登录界面  A2：密码错误  (1)给出相应提示信息  (2)返回登录界面 |

3.查看个人信息

|  |  |
| --- | --- |
| 用例名称 | 查看个人信息 |
| 功能描述 | 用户查看自己的基本信息 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统显示用户的个人信息 |
| 主事件流 | 1.用户选择查看个人信息选项  2.系统读取用户的个人信息  3.系统显示用户个人信息 |
| 扩展事件流 | 无 |

4.修改个人信息

|  |  |
| --- | --- |
| 用例名称 | 修改个人信息 |
| 功能描述 | 用户修改用户名或自己的基本信息 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统将数据库中用户信息更新 |
| 主事件流 | 1.用户选择修改个人信息选项  2.系统显示用户的个人信息  3.用户修改需要修改的信息，并提交  4.系统判断用户信息是否完整，合法  A1：用户信息不完整或不合法  5.系统判断用户名是否被别的用户使用  A2：用户名已被别的用户使用  6.注册成功，返回登录界面 |
| 扩展事件流 | A1：用户信息不完整或不合法  (1)系统给出相应提示信息  (2)返回主事件流第2步  A2：用户名已被别的用户使用  (1)系统给出相应提示信息  (2)返回主事件流第2步 |

5.修改密码

|  |  |
| --- | --- |
| 用例名称 | 修改密码 |
| 功能描述 | 用户修改自己的密码 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统更新数据库中的用户密码 |
| 主事件流 | 1.用户选择修改密码选项  2.用户输入旧密码和新密码，并提交  3.系统验证旧密码是否正确  A1：旧密码错误  4.密码修改成功 |
| 扩展事件流 | A1：旧密码错误  (1)系统给出相应提示信息  (2)显示修改密码界面 |

### 3.4.2 电子资料管理

1.病历管理

病历管理是指用户对自己的病历资料进行管理，包括增加、删除、修改和查找等功能，其用例图如图3-3所示。



图 3- 3 病历管理用例图

2.处方管理

处方管理是指用户对自己的处方资料进行管理，包括增加、删除、修改和查找等功能，其用例图如图3-4所示。



图 3- 4 处方管理用例图

3.体检报告管理

体检报告管理是指用户对自己的体检报告进行管理，包括增加、删除、修改和查找等功能，其用例图如图3-5所示。



图 3- 5 体检报告管理用例图

4.检验报告管理

检验报告管理是指用户对自己的检验报告进行管理，包括增加、删除、修改和查找等功能，其用例图如图3-6所示。



图 3- 6 检验报告管理用例图

5.用例描述

由于病历、处方、体检报告和检验报告四类资料统称为电子资料，病历管理、处方管理、体检报告管理和检验报告管理功能类似，用例图也基本相同，因此对于图3-3至图3-6的用例统一进行描述。

（1）添加电子资料

|  |  |
| --- | --- |
| 用例名称 | 添加电子资料 |
| 功能描述 | 用户输入电子资料信息，添加电子资料 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统将电子资料信息添加至数据库 |
| 主事件流 | 1.用户选择添加电子资料选项  2.用户输入电子资料相关信息，并提交  3.系统验证电子资料必填信息是否输入完整  A1：电子资料必填信息未输入完整  4.电子资料添加成功 |
| 扩展事件流 | A1：电子资料必填信息未输入完整  (1)系统给出相应提示信息  (2)返回主事件流第2步 |

（2）查找电子资料

|  |  |
| --- | --- |
| 用例名称 | 查找电子资料 |
| 功能描述 | 用户根据一系列条件查找电子资料 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统显示所有符合条件的电子资料 |
| 主事件流 | 1.用户输入查找条件，并提交  2.系统根据条件检索电子资料  A1：没有找到符合条件的电子资料  3.系统显示检索到的所有电子资料基本信息 |
| 扩展事件流 | A1：没有找到符合条件的电子资料  (1)系统不显示任何电子资料信息 |

（3）查看电子资料信息

|  |  |
| --- | --- |
| 用例名称 | 查看电子资料信息 |
| 功能描述 | 用户选择某个电子资料，查看其详细信息 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统显示用户选择的电子资料详细信息 |
| 主事件流 | 1.用户选择某一电子资料，并选择查看详细信息选项  2.系统从数据库读取所选电子资料的详细信息  3.系统显示所选电子资料的详细信息 |
| 扩展事件流 | 无 |

（4）修改电子资料信息

|  |  |
| --- | --- |
| 用例名称 | 修改电子资料信息 |
| 功能描述 | 用户选择某个电子资料，并修改其详细信息 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统将电子资料信息更新至数据库 |
| 主事件流 | 1.用户选择某一电子资料，并选择修改选项  2.系统读取所选电子资料的详细信息  3.系统显示所选电子资料的详细信息  4.用户修改需要修改的信息，并提交  5.系统验证电子资料必填信息是否输入完整  A1：电子资料必填信息未输入完整  6.电子资料信息修改成功 |
| 扩展事件流 | A1：电子资料必填信息未输入完整  (1)系统给出相应提示信息  (2)返回主事件流第3步 |

（5）删除电子资料

|  |  |
| --- | --- |
| 用例名称 | 删除电子资料 |
| 功能描述 | 用户删除某个电子资料 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统将数据库中该电子资料删除 |
| 主事件流 | 1.用户选择某一电子资料，并选择删除选项  2.电子资料删除成功 |
| 扩展事件流 | 无 |

（6）保存到本地

|  |  |
| --- | --- |
| 用例名称 | 保存到本地 |
| 功能描述 | 用户将电子资料信息导出到excel表中 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统将电子资料信息保存到本地excel表 |
| 主事件流 | 1.用户选择某一电子资料，并选择查看详细信息选项  2.系统显示电子资料详细信息  3.用户选择保存到本地选项  4.电子资料信息导出成功 |
| 扩展事件流 | 无 |

### 3.4.3 图像资料管理

图像资料的主体图片，还包括资料的基本信息，不同资料的主要区别在于所包含图片的数量不同，而这可以通过对数据表的设计来实现，因此没必要对图像资料进行细分，图像资料管理的用例图如图3-7所示。



图 3- 7 图像资料管理用例图

1.添加图像资料

|  |  |
| --- | --- |
| 用例名称 | 添加图像资料 |
| 功能描述 | 用户输入图像资料信息，添加图像资料 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统将图像资料信息添加至数据库 |
| 主事件流 | 1.用户选择添加图像资料选项  2.用户输入图像资料相关信息，并提交  3.系统验证图像资料必填信息是否输入完整  A1：图像资料必填信息未输入完整  4.系统检验是否添加图片  A2：未添加图片  4.图像资料添加成功 |
| 扩展事件流 | A1：图像资料必填信息未输入完整  (1)系统给出相应提示信息  (2)返回主事件流第2步  A2：未添加图片  (1)系统给出相应提示信息  (2)返回主事件流第2步 |

2.查找图像资料

|  |  |
| --- | --- |
| 用例名称 | 查找图像资料 |
| 功能描述 | 用户根据一系列条件查找图像资料 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统显示所有符合条件的图像资料 |
| 主事件流 | 1.用户输入查找条件，并提交  2.系统根据条件检索图像资料  A1：没有找到符合条件的图像资料  3.系统显示检索到的所有图像资料基本信息 |
| 扩展事件流 | A1：没有找到符合条件的图像资料  (1)系统不显示任何图像资料信息 |

3.查看图像资料信息

|  |  |
| --- | --- |
| 用例名称 | 查看图像资料信息 |
| 功能描述 | 用户选择某个图像资料，查看其详细信息 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统显示用户选择的图像资料详细信息 |
| 主事件流 | 1.用户选择某一图像资料，并选择查看详细信息选项  2.系统从数据库读取所选图像资料的详细信息  3.系统显示所选图像资料的详细信息，包括文字信息和图片 |
| 扩展事件流 | 无 |

4.修改图像资料信息

|  |  |
| --- | --- |
| 用例名称 | 修改图像资料信息 |
| 功能描述 | 用户选择某个图像资料，并修改其详细信息 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统将图像资料信息更新至数据库 |
| 主事件流 | 1.用户选择某一图像资料，并选择修改选项  2.系统读取所选图像资料的详细信息  3.系统显示所选图像资料的详细信息  4.用户修改需要修改的信息，并提交  5.系统验证图像资料必填信息是否输入完整  A1：图像资料必填信息未输入完整  6.图像资料信息修改成功 |
| 扩展事件流 | A1：图像资料必填信息未输入完整  (1)系统给出相应提示信息  (2)返回主事件流第3步 |

5.删除图像资料

|  |  |
| --- | --- |
| 用例名称 | 删除图像资料 |
| 功能描述 | 用户删除某个图像资料 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统将数据库中该图像资料删除 |
| 主事件流 | 1.用户选择某一图像资料，并选择删除选项  2.图像资料删除成功 |
| 扩展事件流 | 无 |

6.保存到本地

|  |  |
| --- | --- |
| 用例名称 | 保存到本地 |
| 功能描述 | 用户将图像资料基本信息导出到excel表中，将图片保存成图片文件 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统将图像资料信息保存到本地excel表，将图片保存成文件 |
| 主事件流 | 1.用户选择某一图像资料，并选择查看详细信息选项  2.系统显示图像资料详细信息  3.用户选择保存到本地选项  A1：用户选择将文字保存到本地  A2：用户选择将图片保存到本地  4.图像资料信息导出成功 |
| 扩展事件流 | A1：用户选择将文字保存到本地  (1)系统将图像资料基本信息保存到excel表中  A2：用户选择将图片保存到本地  (2)系统将图片保存成文件 |

### 3.4.4 提醒管理

提醒功能主要是提醒用户参加复诊或定期体检、吃药等。所以提醒可以分为普通提醒和周期提醒。用户增加和删除普通提醒和周期提醒，因周期提醒包含的信息很多（如每一次提醒的具体时间），因此用户还要能查看周期提醒的信息，所以，可得到提醒管理的用例图如图3-8所示。



图 3- 8 提醒管理用例图

1.添加提醒

|  |  |
| --- | --- |
| 用例名称 | 添加提醒 |
| 功能描述 | 用户输入提醒时间、相应的事件等信息，添加一个提醒 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统将提醒信息存储到数据库中 |
| 主事件流 | 1.用户选择添加提醒选项，输入提醒信息，并提交  2.系统判断提醒信息是否输入完整  A1：提醒信息未输入完整  3.提醒添加成功 |
| 扩展事件流 | A1：提醒信息未输入完整  (1)系统给出相应提示信息  (2)返回主事件流第1步 |

2.添加周期提醒

|  |  |
| --- | --- |
| 用例名称 | 添加周期提醒 |
| 功能描述 | 用户输入周期提醒时间，周期，事件等信息，添加一个周期提醒 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统将周期提醒信息存储到数据库中 |
| 主事件流 | 1.用户选择添加周期提醒选项，输入周期提醒信息，并提交  2.系统判断周期提醒信息是否输入完整  A1：周期提醒信息未输入完整  3.周期提醒添加成功 |
| 扩展事件流 | A1：周期提醒信息未输入完整  (1)系统给出相应提示信息  (2)返回主事件流第1步 |

3.查找提醒

|  |  |
| --- | --- |
| 用例名称 | 查找提醒 |
| 功能描述 | 用户根据一系列条件查找提醒 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统显示所有符合条件的提醒 |
| 主事件流 | 1.用户输入查找条件，并提交  2.系统根据条件检索提醒  A1：没有找到符合条件的提醒信息  3.系统显示检索到的所有提醒信息 |
| 扩展事件流 | A1：没有找到符合条件的提醒信息  (1)系统不显示任何提醒信息 |

4.查看周期提醒信息

|  |  |
| --- | --- |
| 用例名称 | 查看周期提醒信息 |
| 功能描述 | 用户选择某个周期提醒，查看其详细信息 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统显示用户选择的周期提醒详细信息 |
| 主事件流 | 1.用户选择某一周期提醒，并选择查看详细信息选项  2.系统从数据库读取所选周期提醒的详细信息  3.系统显示所选周期提醒的详细信息，包括所有提醒的具体时间 |
| 扩展事件流 | 无 |

5.删除提醒

|  |  |
| --- | --- |
| 用例名称 | 删除提醒 |
| 功能描述 | 用户删除某个提醒 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统将数据库中该提醒删除 |
| 主事件流 | 1.用户选择某一提醒，并选择删除选项  2.提醒删除成功 |
| 扩展事件流 | 无 |

6.删除周期提醒

|  |  |
| --- | --- |
| 用例名称 | 删除周期提醒 |
| 功能描述 | 用户删除某个提周期醒 |
| 前置条件 | 用户登录成功 |
| 后置条件 | 系统将数据库中该周期提醒删除 |
| 主事件流 | 1.用户选择某一周期提醒，并选择删除选项  2.周期提醒删除成功 |
| 扩展事件流 | 无 |

## 3.4 数据流分析

系统顶层数据流图如图3-9所示，图中描述了整个系统的数据流向。用户与系统的信息交互可以从以下几个方面考虑：用户管理病历、用户管理处方、用户管理体检报告、用户管理检验报告、用户管理图像资料和用户管理提醒。用户可以将上述信息输入系统，系统进行存储；用户可以根据一系列查询条件从系统中查询某个资料，系统将查询结果返回给用户；用户可以在系统中设置提醒，系统在提醒时间到达时提醒用户。



图 3- 9 系统顶层数据流图

根据用户与系统交互的数据流，对系统功能进行再次分解。从用户输入的信息看，该系统包含对病历、处方、体检报告、检验报告、图像资料以及提醒的处理，因此，可得到系统0层数据流图如图3-10所示。



图 3- 10 系统0层数据流图

可对图3-10得到的0层数据流图中每个处理过程再次细化，得到以下1层数据流图。

1.病历管理

病历管理的1层数据流图如图3-11所示。当用户添加病历时，病历信息由用户输入并最终存入病历表；若用户需要将病历导出，则病历信息被从病历表中取出后导出给用户；当用户需要查找病历时，系统根据用户输入的查询条件从病历表中查找病历并将查找到的病历显示给用户。



图 3- 11 病历管理1层数据流图

由于处方管理、体检报告管理、检验报告管理以及图像资料管理的过程除了流通的信息与信息最终流入的数据表不同，其余过程均相同，因此其1层数据流图也基本相同，此处不再赘述。

2.提醒管理

提醒管理的1层数据流图如图3-12所示，当用户添加提醒时，提醒信息由用户输入并最终流入到提醒表；当提醒时间到达时，系统将提醒信息显示给用户以达到提醒用户的目的。



图 3- 12 提醒管理1层数据流图

## 3.5 性能需求

除以上功能需求外，系统还应满足一定的性能需求，以保证用户体验，具体包括：

1.界面。本软件界面简洁，简单易懂，信息分布合理，清晰。

2.可靠性和可用性。软件可以一天24小时使用，故障率小于等于5%。

3.安全性。本软件要求不同用户的数据相互独立，用户只有注册和登录后才能使用本软件，且只能查看和操作属于自己的资料，保证用户数据的安全。

4.容错性。用户输入错误信息时，软件应给予相应的提示，不能因此而崩溃。

5.易用性。软件适合12-70岁智力、体力、视力正常且具有计算机使用能力的人使用。

## 3.6 开发和运行环境

### 3.6.1 开发环境

本软件是在正常的计算机系统下开发的，具体开发工具和环境如下：

1.硬件环境：装有Windows 10操作系统的计算机

2.软件环境：.Net平台+SQL Server运行环境

3.开发工具：Microsoft Visual Studio 2013社区版+SQL Server 2008

### 3.6.2 运行环境

本软件能够在任何装有Windows 7及以上版本的计算机中安装、运行，软件运行需要.NET Framework 3.5及以上版本的环境支持。

# 第四章 系统总体设计

## 4.1 系统模块划分

系统总体模块划分如图4-1所示。



图 4- 1 系统总体模块图

综合第三章的分析，可以将整个系统分为上诉七大模块：个人信息模块、病历管理模块、处方管理模块、体检报告管理模块、检验报告管理模块，图像资料管理模块以及提醒管理模块。其中，个人信息模块包含登录、注册、修改用户信息和密码等功能；病历、处方、体检报告、检验报告和图像资料模块包含对相应资料的添加、查找、修改、删除等管理功能；提醒模块包含设置提醒、实现提醒和删除提醒等功能。

## 4.2 数据库设计

### 4.2.1 概念结构设计

将需求分析的用户需求抽象为信息结构即概念结构设计，它是整个数据库的设计关键[13]。

1.经对系统的分析可得到系统的实体包括用户实体、病历实体、处方实体、体检报告实体、检验报告实体、 图像资料实体以及提醒实体和周期提醒实体。但处方中包含有多条处方记录，每一条记录又可抽象成为一个处方明细实体，类似的，可抽象出体检报告明细实体、检验报告明细实体，因一份图像资料可包含多张图片，所以可抽象出图片实体。每个实体及其属性用E-R图描述如下：

（1）用户的属性包括用户ID、用户名、密码、真实姓名、性别、民族、血型、出生日期、职业和地址，其实体及属性描述如图4-2所示。



图 4- 2 用户实体及其属性

（2）病历的属性包括病历ID、医院、科室、就诊时间、主诉、现病史、既往史、系统回顾、个人史、婚姻史、体格检查、辅助检查、初步诊断、治疗意见、医生、复诊时间以及说明，其实体及属性描述如图4-3所示。



图 4- 3 病历实体及其属性

（3）处方的属性包括处方ID、医院、科室、开方日期、开方医师、初步诊断、医嘱以及其他说明，其实体及属性描述如图4-4所示。



图 4- 4 处方实体及其属性

（4）处方明细的属性包括处方明细ID、药品名称、规格、数量、用法、用量，其实体及属性描述如图4-5所示。



图 4- 5 处方明细实体及其属性

（5）体检报告的属性包括体检报告ID、体检单位、体检日期、体检结果、医生建议和其他说明，其实体及属性描述如图4-6所示。



图 4- 6 体检报告实体及其属性

（6）体检报告明细的属性包括体检报告明细ID、体检科室、项目、结果、单位、标识以及参考值，其实体及属性描述如图4-7所示。



图 4- 7 体检报告明细实体及其属性

（7）检验报告的属性包括检验报告ID、医院、检验样本、送检科室、检验日期，其实体及属性描述如图4-8所示。



图 4- 8 检验报告实体及其属性

（8）检验报告明细的属性包括检验报告明细ID、检验项目、检验结果、单位、标识以及参考值，其实体及属性描述如图4-9所示。



图 4- 9 检验报告明细实体及其属性

（9）图像资料的属性包括图像资料ID，医院、科室、日期、图像类型、资料名称、检查部位、影像所见以及检查结果，其实体及属性描述如图4-10所示。



图 4- 10 图像资料实体及其属性

（10）图片的属性包括图片ID和图片，其实体及属性描述如图4-11所示。



图 4- 11 图片实体及其属性

（11）提醒的属性包括提醒ID、地点、事件、提醒时间以及提前时间，其实体及属性描述如图4-12所示。



图 4- 12 提醒实体及其属性

（12）周期提醒实体的属性包括周期提醒ID、开始时间、结束时间、地点、事件以及周期，其实体及属性描述如图4-13所示。



图 4- 13 周期提醒实体及其属性

2.各实体之间的对应关系如下：

（1）一个用户拥有多个病历。

（2）一个用户拥有多个处方，一个处方对应多条处方明细。

（3）一个用户拥有多个体检报告，一个体检报告对应多条体检报告明细。

（4）一个用户拥有多个检验报告，一个检验报告对应多条检验报告明细。

（5）一个用户拥有多份图像资料，一份图像资料拥有多张图片。

（6）一个用户拥有多个提醒和周期提醒，一个周期提醒对应多个提醒。

（7）一个病历对应一个处方。

（8）一个病历对应多个检验报告。

所以可以得到实体之间联系的E-R图如图4-14所示。



图 4- 14 实体之间联系的E-R图

### 4.2.3 逻辑结构设计

根据以上概念模型，可得以下数据表：

1.用户表(c\_user)

用户表包括用户id、用户名、密码、真实姓名等用户基本信息，其逻辑结构如表4-1所示。

表4- 1 用户表c\_user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **约束** | **非空** | **备注** |
| id | int | 主键，自增 | 是 | 用户id |
| username | varchar(50) |  | 是 | 用户名 |
| password | varchar(50) |  | 是 | 密码 |
| name | varchar(50) |  | 是 | 姓名 |
| sex | varchar(20) |  | 是 | 性别 |
| blood | varchar(20) |  | 是 | 血型 |
| birthday | datetime |  | 是 | 出生日期 |
| nation | varchar(20) |  | 是 | 民族 |
| job | varchar(max) |  |  | 职业 |
| address | varchar(max) |  |  | 住址 |

2.病历表(case\_history)

病历表包括病历id，所属的用户id、就诊医院、就诊科室、就诊时间等信息，其逻辑结构如表4-2所示。

表4- 2 病历表case\_history

| **字段** | **类型** | **约束** | **非空** | **备注** |
| --- | --- | --- | --- | --- |
| id | int | 主键，自增 | 是 | 病历id |
| uid | int | 外键 | 是 | 用户id |
| hospital | varchar(50) |  | 是 | 就诊医院 |
| department | varchar(50) |  | 是 | 就诊科室 |
| cureTime | datetime |  | 是 | 就诊时间 |
| tell | varchar(max) |  | 是 | 主诉 |
| presentIllHis | varchar(max) |  |  | 现病史 |
| previousIllHis | varchar(max) |  |  | 既往史 |
| review | varchar(max) |  |  | 系统回顾 |
| personalHis | varchar(max) |  |  | 个人史 |
| marryHis | varchar(max) |  |  | 婚姻史 |
| physicalExam | varchar(max) |  |  | 体格检查 |
| auxilaryExam | varchar(max) |  |  | 辅助检查 |
| primaryDia | varchar(max) |  |  | 初步诊断 |
| treatment | varchar(max) |  |  | 治疗意见 |
| doctor | varchar(20) |  |  | 医生 |
| returnTime | datetime |  |  | 复诊时间 |
| other | varchar(max) |  |  | 其他 |

3.处方表(prescription)

处方表包括病历id、所属用户id、对应的病历id、医院、科室、开方日期等信息，其逻辑结构如表4-3所示。

表4- 3 处方表prescription

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **约束** | **非空** | **备注** |
| id | int | 主键，自增 | 是 | 处方id |
| uid | int | 外键 | 是 | 用户id |
| cid | int |  | 是 | 处方id |
| hospital | varchar(50) |  | 是 | 医院 |
| department | varchar(50) |  | 是 | 科室 |
| date | datetime |  | 是 | 开方日期 |
| doctor | varchar(50) |  |  | 医生 |
| primaryDia | varchar(max) |  |  | 初步诊断 |
| advice | varchar(max) |  |  | 医嘱 |
| other | varchar(max) |  |  | 其他 |

4.处方明细表(rp\_record)

处方明细表包括处方明细id、所属的处方id、药品名称、药品规格、药品数量的该信息，其逻辑结构如表4-4所示。

表4- 4 处方明细表rp\_record

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **约束** | **非空** | **备注** |
| id | int | 主键，自增 | 是 | 处方记录id |
| pid | int | 外键 | 是 | 处方id |
| name | varchar(max) |  | 是 | 药品名称 |
| norms | varchar(max) |  |  | 规格 |
| num | varchar(max) |  |  | 数量 |
| usage | varchar(max) |  |  | 用法 |
| dosage | varchar(max) |  |  | 用量 |

5.体检报告表(medical\_exam)

体检报告表包括体检报告id、所属的用户id、体检单位、体检日期、体检结果等信息，其逻辑结构如表4-5所示。

表4- 5 体检报告表medical\_exam

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **约束** | **非空** | **备注** |
| id | int | 主键，自增 | 是 | 体检报告id |
| uid | int | 外键 | 是 | 用户id |
| hospital | varchar(50) |  | 是 | 体检单位 |
| examDate | datetime |  | 是 | 体检日期 |
| result | varchar(max) |  | 是 | 体检结果 |
| advice | varchar(max) |  |  | 医生建议 |
| other | varchar(max) |  |  | 其他 |

6.体检报告明细表(medical\_record)

体检报告明细表包括体检报告明细id、所属的体检报告id、科室、项目、结果等信息，其逻辑结构如表4-6所示。

表4- 6 体检报告明细表medical\_record

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **约束** | **非空** | **备注** |
| id | int | 主键，自增 | 是 | 体检报告记录id |
| mid | int | 外键 | 是 | 体检报告id |
| department | varchar(max) |  | 是 | 科室 |
| item | varchar(max) |  | 是 | 项目 |
| result | varchar(max) |  |  | 结果 |
| unit | varchar(max) |  |  | 单位 |
| hint | varchar(max) |  |  | 标识 |
| reference | varchar(max) |  |  | 参考值 |

7.检验报告表(assay)

检验报告表包括检验报告id、所属的用户id、对应的病历id、医院、送检科室、送检样本、送检日期，其逻辑结构如表4-7所示。

表4- 7 检验报告表assay

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **约束** | **非空** | **备注** |
| id | int | 主键，自增 | 是 | 检验报告id |
| uid | int | 外键 | 是 | 用户id |
| cid | int |  | 是 | 病历id |
| hospital | varchar(max) |  | 是 | 医院 |
| sample | varchar(max) |  | 是 | 送检样本 |
| examDate | datetime |  | 是 | 送检日期 |
| department | varchar(max) |  |  | 送检科室 |

8.检验报告明细表(assay\_record)

检验报告明细表包括检验报告明细id、所属的检验报告id、检验项目、检验结果等信息，其逻辑结构如表4-8所示。

表4- 8 检验报告明细表assay\_record

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **约束** | **非空** | **备注** |
| id | int | 主键，自增 | 是 | 检验报告记录id |
| aid | int | 外键 | 是 | 检验报告id |
| item | varchar(max) |  | 是 | 项目 |
| result | varchar(max) |  |  | 结果 |
| unit | varchar(max) |  |  | 单位 |
| hint | varchar(max) |  |  | 标识 |
| reference | varchar(max) |  |  | 参考值 |

9.图像资料表(picture)

图像资料表包括图像资料id、所属的用户id、医院、科室、资料类型、资料名称等信息，其逻辑结构如表4-9所示。

表4- 9 图像资料表picture

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **约束** | **非空** | **备注** |
| id | int | 主键，自增 | 是 | 检验报告记录id |
| uid | int | 外键 | 是 | 用户id |
| hospital | varchar(max) |  | 是 | 医院 |
| department | varchar(max) |  | 是 | 科室 |
| examDate | datetime |  | 是 | 检查日期 |
| type | varchar(max) |  | 是 | 资料类型 |
| name | varchar(max) |  |  | 资料名称 |
| part | varchar(max) |  |  | 检查部位 |
| scene | varchar(max) |  |  | 影像所见 |
| result | varchar(max) |  |  | 检查结果 |

10.图片表(picture\_image)

图片表包括图片id、所属的图像资料id以及图片，其逻辑结构如表4-10所示。

表4- 10 图片表picture\_image

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **约束** | **非空** | **备注** |
| id | int | 主键，自增 | 是 | 图片id |
| pid | int | 外键 | 是 | 图像资料id |
| img | image |  | 是 | 图片 |

11.周期提醒表(cycle\_remind)

周期提醒表包括周期提醒id、所属的用户id、开始时间、结束时间等信息，其逻辑结构如表4-11所示。

表4- 11 周期提醒表cycle\_remind

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **约束** | **非空** | **备注** |
| id | int | 主键，自增 | 是 | 周期提醒id |
| uid | int | 外键 | 是 | 用户id |
| startTime | datetime |  | 是 | 开始时间 |
| endTime | datetime |  | 是 | 结束时间 |
| cycle | varchar(max) |  | 是 | 周期 |
| hospital | varchar(max) |  |  | 地点 |
| event | varchar(max) |  |  | 事件 |

12.提醒表(remind)

提醒表包括提醒id、所属的用户id、对应的周期提醒id、提醒时间、事件等信息，其逻辑结构如表4-12所示。

表4- 12 提醒表remind

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **约束** | **非空** | **备注** |
| id | int | 主键，自增 | 是 | 检验报告记录id |
| uid | int | 外键 | 是 | 用户id |
| cycleID | int | 外键 | 是 | 周期提醒id |
| hospital | varchar(amx) |  |  | 地点 |
| event | varchar(amx) |  |  | 事件 |
| interval | varchar(amx) |  |  | 提前间隔 |
| eventTime | datetime |  | 是 | 事件时间 |
| remindTime | datetime |  | 是 | 提醒时间 |

# 第五章 系统详细设计

## 5.1 系统架构设计

本系统采用三层架构进行开发，基于三层架构的结构，对系统架构进行如下设计：

1.系统设计数据访问层CDAL，其主要作用是操作数据库，将上一层传过来的数据插入到数据库中，或者根据上一层传过来的数据对数据表进行相应的处理，并返回处理结果。所有与数据库相关的操作都在这一层，系统每一个与数据库相关的小功能最后都会转化为一个此层的函数，如用户登录功能可由此层的login函数实现：其接受上层传递的用户名和密码，将字符串拼接成SQL语句，然后连接数据库，执行SQL语句，并将查询结果（登录成功结果即为用户id）返回给上一层。

2.业务逻辑层CBLL，其主要作用是处理相关的业务逻辑，但因为本系统功能简单，系统本身与业务的关系结合的不紧密，该层的作用只是在表示层和数据访问层之间传递数据。对该层每个函数都存在一个数据访问层的函数与之对应，例如用户登录函数login对应数据访问层的login函数，其接受表示层传递的用户名和密码，调用数据访问层的login函数，并将数据访问层的login函数的执行结果返回给表示层。因本层功能过于简单，本系统的实现过程并没有对其进行过多的介绍，其实从功能上将完全可以将本层去除，但从三层架构的完整性以及系统的易维护性方面考虑，还是将其保留。

3.表示层UI，该层也是最重要的一层，所有用户与系统的交互都在该层实现。该层包括系统界面和与界面相关的数据处理，以及数据准备，其不涉及数据库操作，所有需要操作数据库的功能都是通过调用业务逻辑层的函数实现。例如用户登录界面，用户点击登录按钮后，该层首先对用户输入的用户名和密码进行基本的合法性判断（用户名和密码是否输入完整等），如合法，则调用业务逻辑层的login函数，并根据其返回的结果进行下一步操作。

4.实体层Model，该层严格意义上不属于三层架构的基本层，其存在的主要意义是方便层与层之间的数据传输，相当于一种数据格式。例如用户注册时要将用户信息写入数据库，此时需要把用户名、密码、姓名、性别等信息从表示层传到业务逻辑层，再从业务逻辑层传到数据访问层，这些参数在传递过程中很容易混淆，实现起来很不方便。如果在此层新增UserInfo类，该类的属性包含用户名、密码、姓名、性别等用户的基本信息，在用户注册时将这些信息封装至该类的一个实例userInfo中，直接把userInfo作为表示层、业务逻辑层和数据访问层之间传递的参数，并在数据访问层将用户名等信息从userInfo中取出，这种方法要比直接传递参数更加直观有效。对于数据库的每个实体，该层都有相应的类与之对应，类中的属性即为实体的属性，如UserInfo对应了用户实体。

## 5.2 模块设计

本系统采用C#语言实现，因为C#语言是典型的面向对象编程语言，系统实现主要是通过类，所以系统设计主要是对类的设计，本节主要介绍了对系统每个模块设计的类，通过类图展现一个模块所有类之间的联系，通过顺序图介绍某个功能实现的步骤。

### 5.2.1 数据库操作类

数据库操作主要指连接数据库和执行SQL语句，对所有功能来说，这两部分都是基本一样的，不同之处在于执行的SQL语句。所以，可以在数据访问层CDAL中提取出数据库操作公共类DbHelper，该类主要包含以下方法：

1. ExecuteDataTable，执行SQL语句并返回执行结果，主要执行查询语句，接受参数包括SQL语句commandText、SQL语句参数SqlParameter、SQL语句类型commandType，返回值为DataTable类型，根据需要参数个数的不同进行重载。

2. ExecuteReader，执行SQL语句并返回执行结果，主要执行查询语句，接受参数与上相同，返回值为SqlDataReader类型，根据需要参数个数的不同进行重载。

3. ExecuteScalar，执行SQL语句并返回执行结果的第一行第一列，主要执行查询语句，接受参数与上相同，返回值为Object类型，根据需要参数个数的不同进行重载。

4. ExecuteNonQuery，执行SQL语句并返回受影响的行数，主要执行增删改语句，接受参数与上相同，返回值为int类型，根据需要参数个数的不同进行重载。

### 5.2.2 个人信息管理模块

个人信息管理模块的功能包括用户登录，用户注册，查看个人信息，修改个人信息，修改密码，这些功能主要靠界面类和数据操作类实现，所以本模块设计以下类：

1.数据操作类（User）：位于数据访问层，描述了所有与数据库操作相关的方法。

2.用户实体类（User）：位于实体层，用于描述用户信息。

3.登录界面类（Login）：位于表示层，描述了登录界面。

4.注册界面类（Register）：位于表示层，描述了注册界面。

5.查看个人信息界面类（UserInfo）：位于表示层，描述了个人信息查看界面。

6.修改个人信息界面类（ChangeUserInfo）：位于表示层，描述了修改个人信息界面。

7.修改密码界面类（ChangePassword）：位于表示层，描述了修改密码界面。

该模块系统类图如图5-1所示。



图 5- 1 个人信息管理类图

该模块序列图如图5-2所示，图中描述了各功能实现过程中消息在类的实例中传递的过程以及类的调用关系。当用户注册时，登录界面调用注册界面类构造函数并显示注册界面，注册界面验证用户输入信息后将用户信息传递到数据库操作类User，并执行该类的注册函数将用户信息写入数据库；当用户登录时，登录界面首先检查用户输入信息，然后调用User类下的登录函数验证用户名和密码是否正确，如正确，则将用户id返回；当用户查看个人信息时，查看个人信息页面根据用户id调用User类下的查看个人信息函数获得用户信息，再将信息显示在页面上；当用户修改个人信息时，修改个人信息页面首先使用查看个人信息的方法将信息显示在页面上，用户修改后再调用User类下修改个人信息方法将个人信息更新至数据库；当用户修改密码时，修改密码页面首先验证用户输入的旧密码，若旧密码正确，则调用User类下修改密码方法将新密码更新至数据库。



图 5- 2 个人信息管理序列图

### 5.2.3 病历管理模块

病历管理模块的功能包括添加病历、查看病历信息、修改病历信息、查找病历、删除病历以及导出病历。本模块设计以下类来实现上述功能：

1.数据操作类（Case\_history）：位于数据访问层，描述了所有与数据库操作相关的方法。

2.病历实体类（Case\_history）：位于实体层，用于描述病历信息。

3.病历管理主界面类（AllCase）：位于表示层，描述了病历管理主界面，提供各功能的入口。

4.添加病历界面类（AddCase）：位于表示层，描述了添加病历的界面。

5.查看病历信息界面类（CaseInfo）：位于表示层，描述了查看病历信息的界面。

6.修改病历信息界面类（ChangeCase）：位于表示层，描述了修改病历信息的界面。

该模块类图如图5-3所示。



图 5- 3 病历管理类图

病历管理的时序图如图5-4所示，图中描述了病历管理中添加病历、查看病历信息、删除病历和修改病历信息的执行过程。用户选择病历管理选项后，进入病历管理主界面，该页面根据用户id调用数据操作类Case\_history下的getAllCase()方法获取用户所有病历的基本信息，用户可以选择某个病历进行操作。当用户选中某个病历并选择查看病历信息时，主界面调用病历信息界面的构造函数，病历信息界面调用数据操作类getCase()方法获取选中病历的详细信息并显示在页面上；当用户选中某个病历并选择修改病历信息时，主界面调用修改病历信息界面的构造函数，修改病历信息界面首先使用显示病历信息的方法显示病历详情，用户修改病历信息后调用数据操作类的changeCase()方法将修改后的病历信息更新至数据库；当用户选中某个病历并选择删除病历信息时，主界面调用数据操作类的deleteCase()方法从数据库删除该病历；当用户选择添加病历时，主界面调用添加病历界面的构造函数，添加病历界面调用数据操作类下的addCase()方法将病历信息添加至数据库。



图 5- 4 病历管理顺序图

病历管理主界面使用C#自带控件DataGridview显示所有病历的基本信息，因该控件有自带的检索功能，所以查找病历直接使用DataGridView控件的自带的功能实现。其流程图如图5-5所示，因以下所有查找功能实现原理一致，所以后面不再重复。



图 5- 5 查找病历功能流程图

### 5.2.4 处方管理模块

处方管理模块包括增加处方、查看处方信息、修改处方信息、查找处方、删除处方和导出处方六个功能，本模块设计一下类来实现上述功能：

1.数据库操作类（Prescription）：位于数据访问层，描述了所有与数据库操作相关的方法。

2.处方实体类（Prescription）：位于实体层，描述了处方的基本信息。

3.处方明细实体类（Rp\_record）：位于实体层，描述了处方中每一条记录的信息。

4.处方管理主界面类（AllPrescription）：位于表示层，描述了处方管理主界面，提供各功能的入口。

5.添加处方界面类（AddPrescription）：位于表示层，描述了添加处方界面。

6.查看处方信息界面类（PrescriptionInfo）：位于表示层，描述了查看处方信息界面。

7.修改处方信息界面类（ChangePrescription）：位于表示层，描述了修改处方信息界面。

该模块类图如图5-6所示。



图 5- 6 处方管理类图

处方管理时序图如图5-7所示，图中描述了添加处方、查看处方、修改处方信息和删除处方的执行过程。当用户选择添加处方选项时，调用处方管理主界面类下的构造函数构造处方管理主界面，主界面又调用数据操作类下的getAllPre()方法根据用户id获取所有处方的基本信息并显示在页面上；当用户选择添加处方选项时，主界面调用添加处方界面的构造函数构造添加处方界面，用户输入处方信息后，添加处方界面首先调用数据操作类下的addPre()方法将处方基本信息写入处方表，根据返回的id再调用addRecord()方法将所有处方明细写入处方明细表；当用户选中某个处方并选择查看处方信息时，查看处方信息界面调用数据操作类下的getPre()和gedRecord()方法获得处方基本信息和所有处方明细并显示出来；当用户选中某个处方并选择修改处方信息时，修改处方信息界面首先使用查看处方信息的方法将处方基本信息和处方明细显示，用户修改后该界面首先调用数据操作类下的changePre()方法将处方基本信息更新至数据库，然后调用deleteRecord()方法删除所有旧的处方明细，再调用addRecord()方法添加新的处方明细；当用户选中某个处方并选择删除处方时，主界面调用数据操作类下的deletePre()方法将处方从处方信息表删除，因处方信息表和处方明细表之间的外键关系指定删除时的级联操作，该处方对应的处方明细会自动删除。查找处方和导出处方分别在处方管理主界面和查看处方信息界面实现。



图 5- 7 处方管理顺序图

### 5.2.5 体检报告管理模块

体检报告管理模块包括添加体检报告、查看体检报告信息、修改体检报告信息、查找体检报告、删除体检报告和导出体检报告功能，该模块设计以下类：

1.数据操作类（Exam）：位于数据访问层，描述了所有与数据库操作相关的方法。

2.体检报告实体类（Exam）：位于实体层，描述了体检报告的基本信息。

3.体检报告明细实体类（Exam\_record）：位于实体层，描述了体检报告每一项的信息。

4.体检报告管理主界面类（AllExam）：位于表示层，描述了体检报告管理主界面，提供了各功能入口。

5.添加体检报告界面类（AddExam）：位于表示层，描述了添加体检报告界面。

6.查看体检报告信息界面类（ExamInfo）：位于表示层，描述了查看体检报告信息界面。

7.修改体检报告信息界面类（ChangeExam）：位于表示层，描述了修改体检报告信息界面。

该模块类图如图5-8所示，由于体检报告管理与处方管理设计的类大致相同，每一类实现的方法和各功能实现的流程基本一致，因此模块序列图也基本类似，此处不再给出，各功能实现的步骤也不再赘述。



图 5- 8 体检报告管理类图

### 5.2.6 检验报告管理模块

检验报告管理模块功能包括添加检验报告、查看检验报告信息、修改检验报告信息、查找检验报告、删除检验报告和导出检验报告。与处方管理和体检报告管理类似，本模块设计以下类：

1.数据操作类（Assay）：位于数据访问层，描述了所有与数据库操作相关的方法。

2.检验报告实体类（Assay）：位于实体层，描述了检验报告的基本信息。

3.检验报告明细实体类（Assay\_record）：位于实体层，描述了检验报告每一项的信息。

4.检验报告管理主界面类（AllAssay）：位于表示层，描述了检验报告管理主界面，提供了各功能入口。

5.添加检验报告界面类（AddAssay）：位于表示层，描述了添加检验报告界面。

6.查看检验报告信息界面类（AssayInfo）：位于表示层，描述了查看检验报告信息界面。

7.修改检验报告信息界面类（ChangeAssay）：位于表示层，描述了修改检验报告信息界面。

模块类图如图5-9所示，同样，本模块功能的实现流程与处方管理和体检报告管理实现流程类似，所以不再给出模块顺序图。



图 5- 9 检验报告管理类图

### 5.2.7 图像资料管理模块

图像资料管理模块的功能包括添加图像资料，查看图像资料信息，修改图像资料信息，查找图像资料，删除图像资料以及导出图像资料。本模块功能与处方管理、体检报告管理和检验报告管理功能类似，只是具体记录显示的形式不同：本模块具体记录以图片方式显示，而另外三个模块具体记录以表格形式显示。所以，本模块类设计也与他们类似，功能实现的过程亦基本相同，不再给出顺序图。本模块设计以下类，类图如图5-10所示。

1.数据操作类（Picture）：位于数据访问层，描述了所有与数据库操作相关的方法。

2.图像资料实体类（Picture）：位于实体层，描述了图像资料的基本信息。

3.图像资料管理主界面类（AllPicture）：位于表示层，描述了图像资料管理主界面，提供了各功能入口。

4.添加图像资料界面类（AddPicture）：位于表示层，描述了添加图像资料界面。

5.查看图像资料信息界面类（PictureInfo）：位于表示层，描述了查看图像资料信息界面。

6.修改图像资料信息界面类（ChangePicture）：位于表示层，描述了修改图像资料信息界面。



图 5- 10 图像资料管理类图

### 5.2.8 提醒管理模块

提醒管理模块的功能包括添加提醒、添加周期提醒、查看周期提醒信息、删除提醒和实现提醒。本模块设计以下类实现上诉功能：

1.数据操作类（Remind）：位于数据访问层，描述了与数据操作相关的方法。

2.提醒实体类（Remind）：位于实体层，描述了提醒的基本信息。

3.周期提醒实体类（Cycle\_remind）：位于实体层，描述了周期提醒的基本信息。

4.提醒管理主界面类（AllRemind）：位于表示层，描述了周期提醒管理的主界面，提供了各功能入口。

5.添加提醒界面类（AddReming）：位于表示层，描述了添加提醒的界面。

6.添加周期提醒界面类（AddCycleRemind）：位于表示层，描述了添加周期提醒的界面。

7.查看周期提醒界面类（CycleRemindInfo）：位于表示层，描述了查看周期提醒信息的界面。

8.提醒界面类（Clock）：位于表示层，描述了提醒时间到来时提示用户的界面。

该模块类图如图5-11所示。



图 5- 11 提醒管理类图

该模块顺序图如图5-12所示，图中描述了提醒管理模块各功能的实现过程。当用户选择提醒管理选项时，调用提醒管理主页面构造函数构造提醒管理主页面，提醒管理主页面调用数据操作类下的getAllRemind()方法获取所有提醒并显示；当用户选中某个提醒（或周期提醒）并选择删除时，主页面调用数据操作类下的deleteRemind()（或deleteCycleRemind()）方法将提醒从数据库删除；当用户选中某个周期提醒并选择查看详情时，主界面调用查看周期提醒界面构造函数构造查看周期提醒界面，该界面调用数据操作类下的getRemind()方法获取被选中的周期提醒对应的所有提醒并显示；当用户选择添加提醒时，主页面调用添加提醒界面构造函数构造添加提醒界面，用户输入提醒信息后添加提醒界面调用数据操作类下的addRemind()方法将提醒信息写入数据库；当用户选择添加周期提醒时，主界面调用添加周期提醒界面构造函数构造添加周期提醒界面，用户输入周期提醒信息后，添加周期提醒界面调用数据操作类下的AddCycleRemind()方法将周期提醒信息写入数据库，并计算周期提醒对应的每一次提醒，调用addRemind()方法将每一次提醒信息写入数据库。



图 5- 12 提醒管理顺序图

# 第六章 系统实现

基于以上的分析和设计，整个系统结构已经非常清晰，本章主要介绍了系统的实现，包括系统界面，部分关键代码和部分功能的程序流程图。

## 6.1 三层架构搭建

本系统采用VisualStudio 2013进行开发，系统实现的第一步是搭建基础架构。在VS2013中新建C# Windows窗体应用程序，在新建的项目中添加四个类库，分别命名为CDAL、CBLL、UI和Model，分别作为三层架构的数据访问层、业务逻辑层、表示层和实体层。为了使高层能引用低层的方法，需要为他们添加引用：为表示层添加对业务逻辑层和实体层的引用，为业务逻辑层添加对数据访问层和实体层的引用，为数据访问层添加对实体层的引用。

在数据访问层CDAL中新建数据操作公共类DbHelper，根据5.2.1的设计，该类中共添加四大类共16个方法，这些方法供SQL语句执行时调用。

## 6.2 模块功能实现

### 4.2.1 个人信息管理模块

1.用户登录。启动软件后，首先进入登录界面，如图6-1所示，用户输入用户名和密码点击登录即可登入系统，系统主界面如图6-2所示。若用户未输入用户名或密码直接点击登录，系统会提示用户输入完整；若用户名不存在或密码错误，系统也会给出相应提示信息。该软件还有托盘功能，用户登录成功后会在任务栏中显示系统托盘，用户关闭软件界面时软件会最小化到托盘，用户可以通过托盘退出软件。

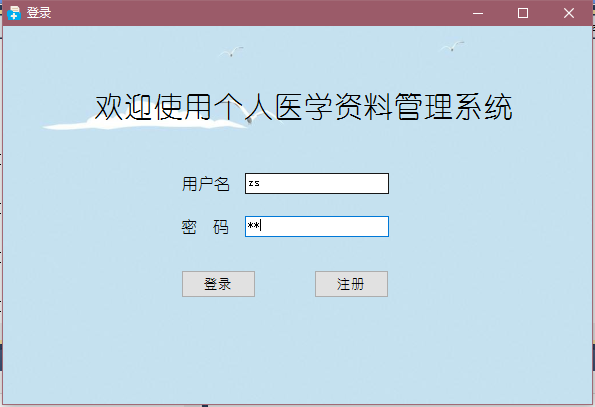


图 6- 1 登录界面

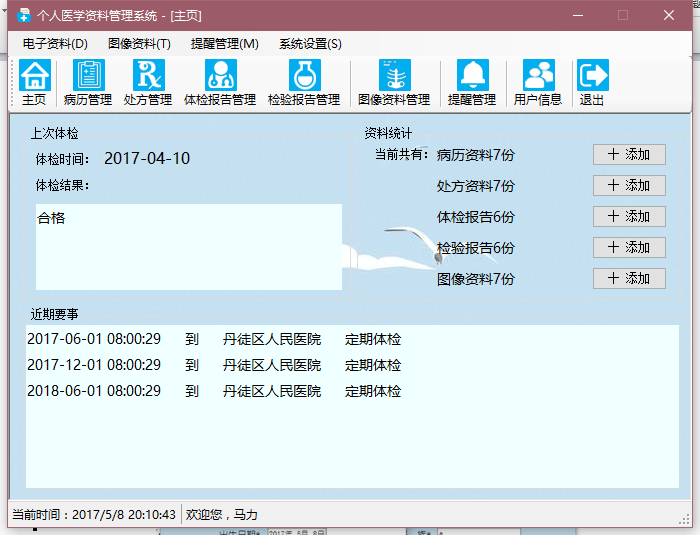


图 6- 2 系统主界面

2.用户注册。在登录界面，点击注册按钮，进入注册界面，如图6-3所示。用户输入用户名、密码等信息，点击注册，即可注册成功。如果用户输入的用户名已存在或密码不一致或者信息没有输入完整，会有相应的提示信息。



图 6- 3 注册界面

3.修改个人信息。用户登录成功后，在主界面点击用户信息选项，可以查看用户信息；在用户信息界面点击修改按钮，可进入修改用户信息界面，用户修改完毕后点击保存修改按钮，即可完成个人信息的修改，如图6-4所示。同注册一样，若修改后的用户名已存在或用户信息输入不完整，也会有相应的提示信息。

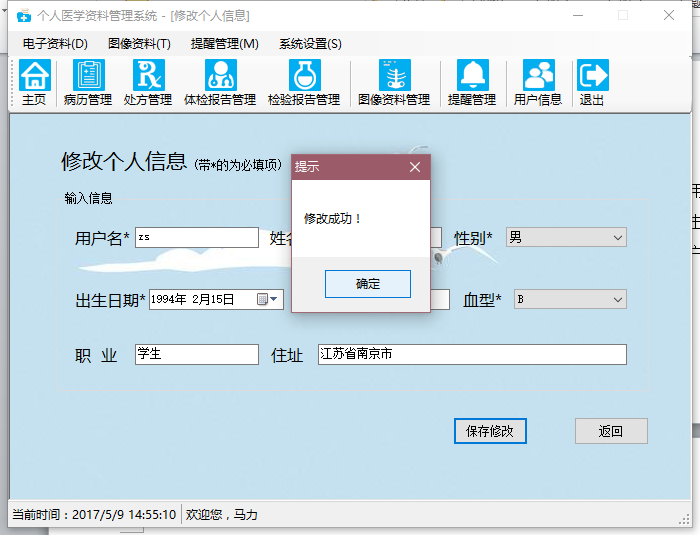


图 6- 4 修改个人信息界面

### 4.2.2 病历管理模块

1.查找病历。用户在软件主界面点击病历管理选项后，进入病历管理主界面，在病历管理主界面中显示当前用户所有病历的基本信息，包括就诊医院、科室、时间、医生等。用户可在右侧搜索区域相应的输入框内输入查找条件，点击查找，即可查找出所有符合条件的病历。如查找“急诊科”的结果如图6-5所示。

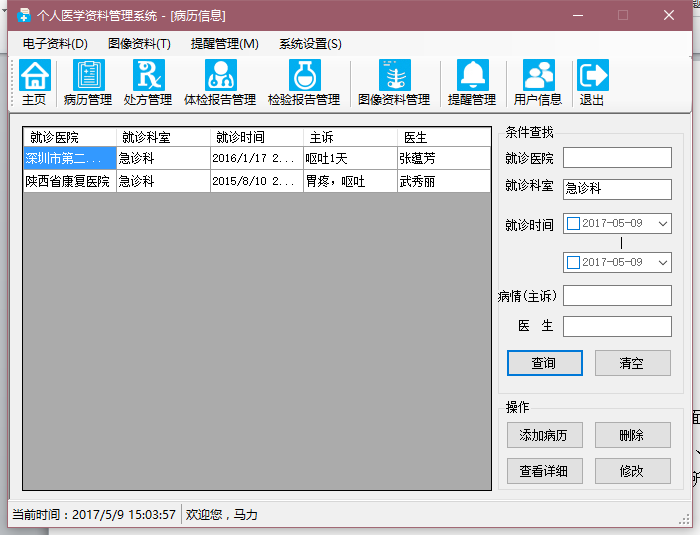


图 6- 5 查找病历

2.添加病历。用户在病历管理点击添加病历按钮后，会弹出添加病历界面，用户输入病历信息后点击添加，即可在软件中添加一份病历，如图6-6所示。当病历必填信息输入不完整时系统会有相应的提示信息。

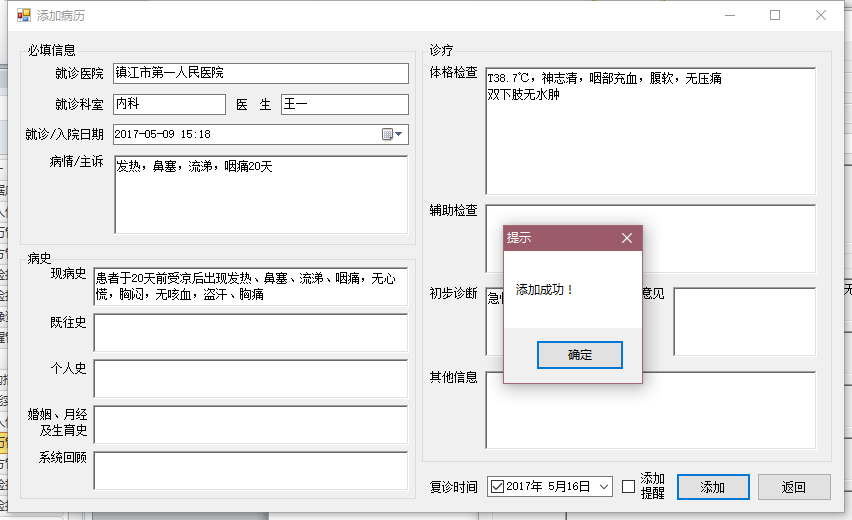


图 6- 6 添加病历界面

3.查看病历信息。在病历管理首页，选中某个病历，点击查看详细按钮，或使用右键菜单，点击查看详细，即可进入查看病历信息界面，如图6-7所示。

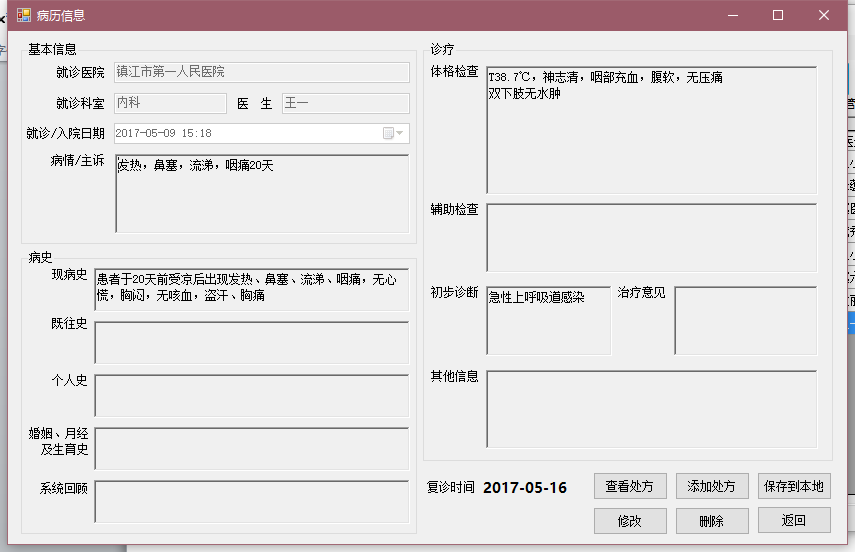


图 6- 7 查看病历信息

4.修改病历和导出病历。在病历详细信息界面点击修改按钮，进入修改病历信息按钮。其界面与添加病历界面相同，只是输入框内已有原病历信息，用户直接修改后点保存按钮即可修改成功。在病历详细信息界面点击保存到本地按钮，可将病历信息导出至本地Excel表中，如图6-8所示。

导出功能主要是使用微软提供的Excel组件和文件操作，用户点击导出后，软件首先将系统内的模板文件复制到用户的导出目录，然后打开新复制的文件，将信息写入后关闭文件并保存，即实现了导出功能。以下所有导出功能实现的原理相同，所以不再重复。

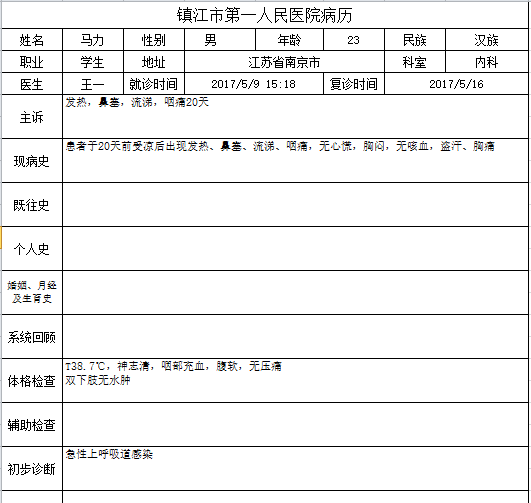


图 6- 8 导出的病历

5.删除病历。在病历管理主界面选中某个病历，点击删除按钮，或直接右键菜单选择删除，或者在病历详情页面点击删除按钮，会弹出删除确认对话框，确认后即可删除病历。

### 4.2.3 处方管理模块

1.查找处方。用户在软件主界面点击处方管理后，进入处方管理主界面，该界面显示当前所有处方的基本信息，包括医院、科室、开方日期、医师等，如图6-9所示。在界面右侧搜索区域输入框内输入相应的搜索条件，即可搜索出所有符合条件的处方。

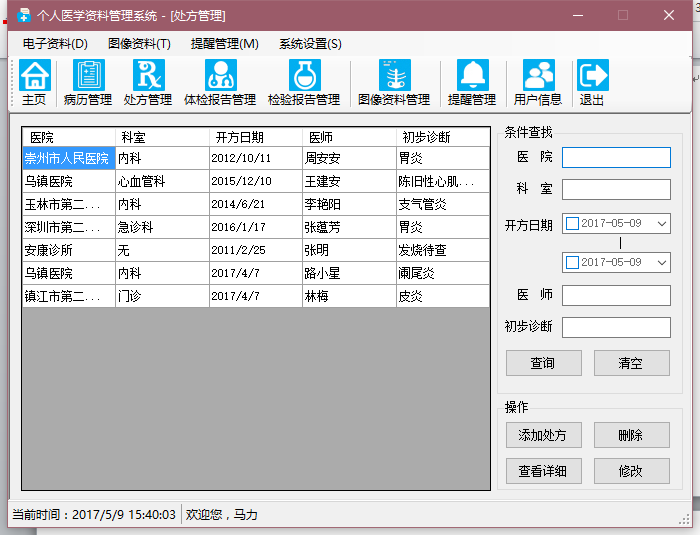


图 6- 9 处方管理主界面

2.添加处方。在处方管理主界面点击添加处方按钮，弹出添加处方界面，如图6-10所示，在该界面输入处方基本信息和所有处方记录信息，点击添加按钮，即可添加一份处方。若处方必填信息或处方记录的药品名称未输入完整，软件会作出相应的提示。



图 6- 10 添加处方界面

3.查看处方信息和修改处方信息。在处方管理主界面选择处方然后点击查看按钮或通过右键菜单查看处方基本信息，可进入处方信息查看界面，该界面显示了选中处方的基本信息和其所有处方记录，且所有信息不可编辑，如图6-11所示。在查看处方信息界面点修改按钮，可对处方信息进行修改，修改界面与添加处方界面相同，修改完成后点击保存，即可完成处方信息的修改。

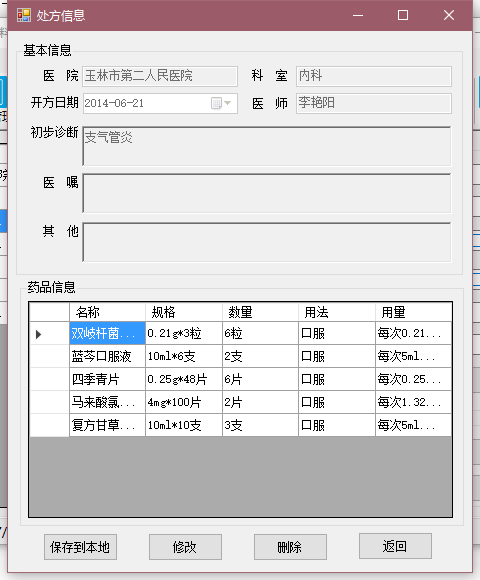


图 6- 11 查看处方信息

4.删除处方和导出处方。在查看处方信息页面点击删除按钮或在处方管理主界面选中处方点击删除或者右键菜单删除处方，经确认后可删除处方。在查看处方信息界面点击保存到本地按钮，可将处方信息导出到Excel表中，如图6-12所示。



图 6- 12 导出的处方

### 4.2.4 体检报告管理模块

1.查找体检报告。用户在软件主界面点击体检报告管理后，进入体检报告管理主页面，该页面显示了当前用户所有体检报告的基本资料。同病历管理和处方管理相同，用户可以在右侧搜索区域输入搜索条件对体检报告进行检索。

2.添加体检报告。在体检报告管理主页面点击添加报告按钮，进入添加体检报告界面，如图6-13所示。在该页面输入体检报告的基本信息，并按科室输入体检记录信息，点击保存按钮，可添加一份体检报告。同样，若有信息输入不完整，软件会有相应的提示信息。本界面已为用户预设了一些体检项目，用户可以自行添加或删除，若某一项用户没有输入信息，保存时将不会保存该项。

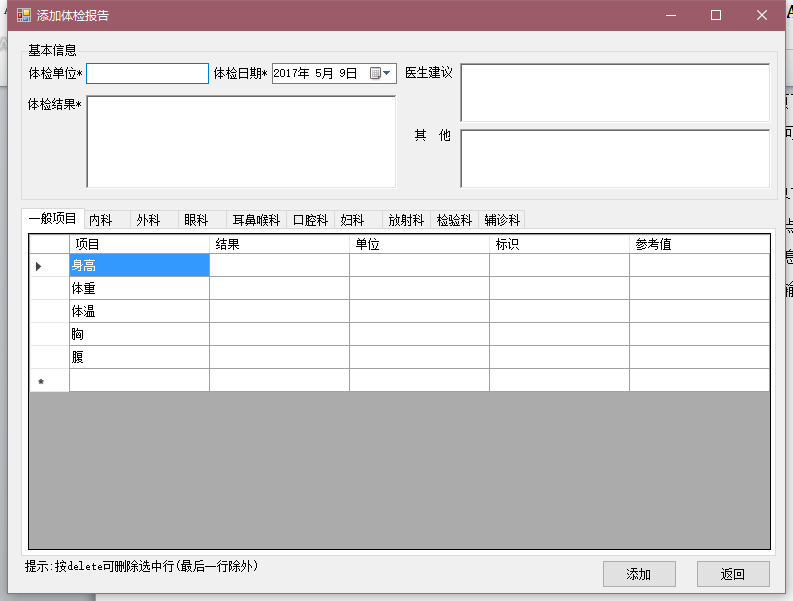


图 6- 13 添加体检报告

3.查看体检报告信息和修改体检报告信息。在体检报告管理主界面选中某个体检报告，点击查看详细按钮或通过右键菜单查看详细，可进入查看体检报告详细信息界面，如图6-14所示，该界面与添加体检报告界面类似，只是所有信息不可编辑；点击修改按钮，可进入修改体检报告信息界面，该界面与查看体检报告信息类似，只是用户可以对信息进行修改，修改后点击保存按钮即可保存修改后的体检报告。

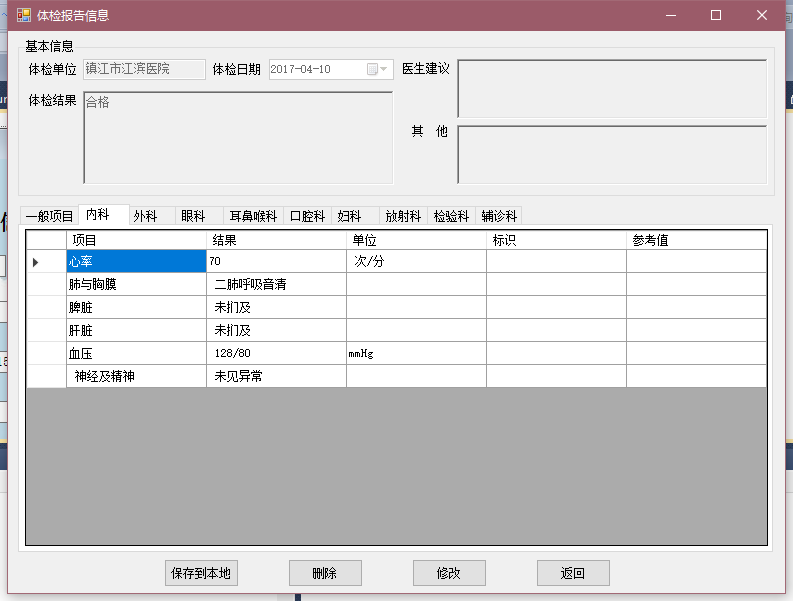


图 6- 14 查看体检报告

4.删除体检报告和导出体检报告。在体检报告管理主界面选中某个体检报告点击删除，或者通过邮件菜单删除报告，或者在查看体检报告详情页面点击删除按钮，确认后可删除体检报告。在查看体检报告详细信息界面点击保存到本地按钮，可将体检报告信息导出到Excel表，如图6-15所示。



图 6- 15 导出的体检报告

### 4.2.5 检验报告管理模块

1.查找检验报告。用户选择软件主界面检验报告管理选项后，进入检验报告管理主界面。该界面显示了当前用户所有检验报告的基本信息，在界面右侧搜索区域输入查询条件并点击查找按钮后，将显示出所有符合条件的检验报告基本信息。

2.添加检验报告。用户在检验报告管理主界面点击添加报告按钮，进入添加检验报告界面，如图6-16所示，用户在该界面输入检验报告基本信息和所有检验报告记录，点击添加按钮，则成功添加一份检验报告。

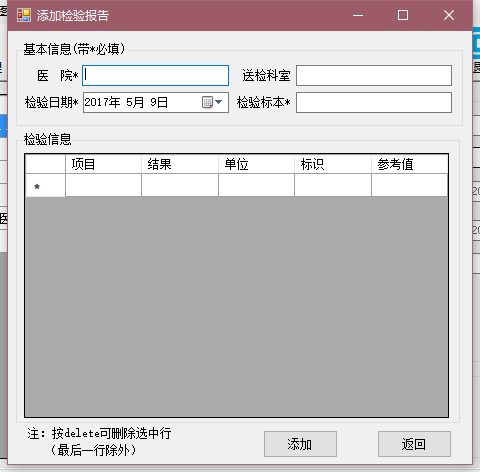


图 6- 16 添加检验报告

3.查看检验报告信息和编辑检验报告信息。用户在检验报告管理主界面选中某个检验报告点击查看详细按钮或者直接通过右键菜单查看详细，则可进入查看检验报告详细信息界面，如图6-17所示，该界面显示用户选中的检验报告的基本信息和所有检验报告记录，其界面与添加检验报告界面类似，只是无法编辑；用户在查看检验报告界面点击修改按钮，进入修改检验报告信息界面，用户修改完成后点击保存按钮，可保存修改后的检验报告信息。

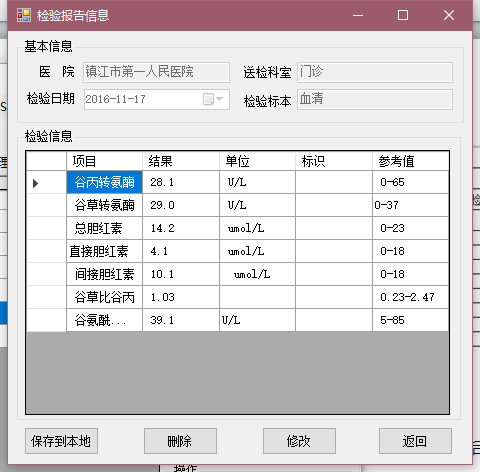


图 6- 17 查看检验报告

4.删除检验报告和导出检验报告。用户可通过检验报告管理主界面或检验报告详情页面经确认后删除某个检验报告。在检验报告详情页面，用户点击保存到本地按钮，可将检验报告信息导出至Excel表中，如图6-18所示。



图 6- 18 导出的检验报告

### 4.2.6 图像资料管理模块

1.查找图像资料。用户在软件主界面选择图像资料管理后，进入图像资料管理主界面。该界面会显示用户所有图像资料的基本信息。同样，用户在右侧搜索区域输入查询条件，会查询出所有符合条件的图像资料。

2.添加图像资料。用户在图像资料管理主界面点击添加按钮，会打开添加图像资料界面，如图6-19所示。用户输入图像资料基本信息，并添加图片后，点击添加按钮，即可添加图像资料。



图 6- 19 添加图像资料

3.查看图像资料信息和编辑图像资料。用户在图像资料管理主界面选中某个图像资料，点击查看详细按钮或直接通过右键菜单可打开查看图像资料界面，如图6-20所示。该界面显示了图像资料的基本信息和所包含的图片，通过点击上一张和下一张按钮可查看所有图片。用户点击修改按钮后，可进入修改图像资料界面对该资料进行修改，修改后点保存按钮即可保存修改。



图 6- 20 查看图像资料信息

4.删除图像资料和导出图像资料。用户可以在图像资料管理主界面或查看图像资料信息界面删除某一份图像资料。在查看图像资料信息界面，点击图像资料基本信息下的保存到本地按钮可将图像资料基本信息导出到Excel表中，点击图片下的保存到本地按钮可将当前图片保存到本地计算机中。

### 4.2.7 提醒管理模块

1.查看所有提醒和查找提醒。用户在软件主界面选择提醒管理选项，进入提醒管理主界面，如图6-21所示。该界面显示了所有提醒和周期提醒，用户同样可以通过条件查找某个提醒。

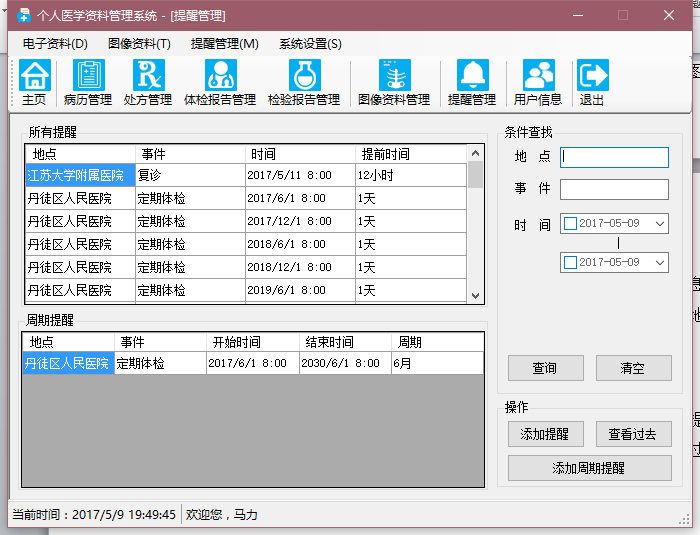


图 6- 21 提醒管理主界面

2.添加提醒。用户点击提醒管理主界面上的添加提醒按钮，进入添加提醒界面，如图6-22所示，用户输入提醒时间、地点等信息后，点击添加按钮，即可添加一条提醒。

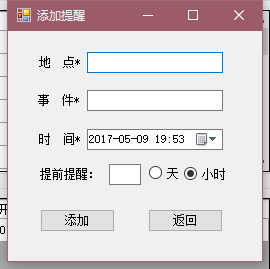


图 6- 22 添加提醒

3.添加周期提醒。用户点击提醒管理主界面上的添加周期提醒按钮，进入添加周期提醒界面，如图6-23所示，用户输入周期提醒的开始时间、结束时间等信息后，点击添加按钮，即可添加一条周期提醒，同时会对应地添加所有周期提醒中的每一次提醒。

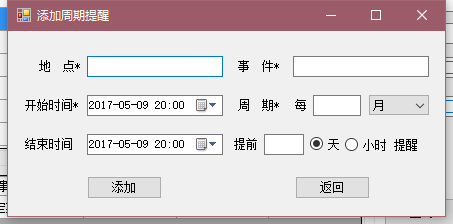


图 6- 23 添加周期提醒

4.查看周期提醒信息。用户在提醒管理主界面通过右键菜单点击某一条周期提醒，然后选择查看详情，可查看该周期提醒的详细信息，包括开始时间、结束时间、剩余次数和每次提醒的具体时间，如图6-24所示。



图 6- 24 查看周期提醒详细信息

5.删除（周期）提醒。用户在提醒管理主界面或查看周期提醒详细信息界面可以使用右键菜单经确认后删除某个提醒，也可以在提醒管理主界面通过右键菜单删除某个周期提醒，或者在查看周期提醒详细信息界面点击删除按钮直接删除当前周期提醒。

6.实现提醒。当当前时间到达某个提醒的时间时，软件会弹出提醒框提醒用户，同时软件托盘闪烁并伴有闹铃声响起，用户关闭提醒框后声音停止。若用户未关闭提醒框，闹铃声会在1分钟后停止，并在5分钟后再次响起，共响铃3次停止。提醒框如图6-25所示。

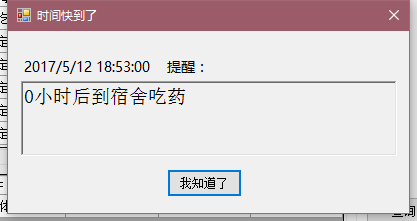


图 6- 25 提醒框

# 总 结

通过这次毕业设计，我设计实现了一个个人医学资料管理软件，本软件能将用户的病历、处方、体检报告、检验报告和其他图像资料分门别类地存储在数据库中，这些资料可供用户随时查看，或者根据需要导出，软件支持多重条件组合查询，方便用户快速找到某个资料；此外，软件还实现了提醒功能，可根据用户需要对要事做出提醒；另外，软件需注册登录后才能使用，保证了用户数据的安全性。

本软件模块划分合理，功能基本实现，并且界面简洁，容易上手。软件采用三层架构搭建，将用户界面和数据操作分离，体现了高内聚，低耦合的特点，即便于软件维护，又能很容易对软件功能进行扩展。经过测试，软件运行稳定，能够满足用户的基本需求。

总的来说，最后的成果能够达到毕业设计的要求，但是，软件还有一些不足之处：

1.软件功能相对来说还是比较简单，可以在此基础扩展出许多功能，如根据用户的体检报告生成某段时间内某个指标的变化趋势，就能更加体现出软件在个人健康方面发挥的作用。

2.如果能增加服务器端，将数据存取操作放在服务器端，这样用户在任何安装了本软件且接入互联网的计算机上都能查看、管理自己的资料，提高了软件使用的广泛性。

3.虽然软件使用了三层架构技术，但业务逻辑层只起到了传递数据的作用，如果能将一些页面逻辑处理移到业务逻辑层，将使软件结构更加清晰，分层也更合理。

# 致 谢

转眼间，大学生活即将结束，感谢学校和老师们对我的教导，我才能从刚入学时的一无所知成长到能独立完成此次毕业设计。特别要感谢我的指导老师年轶老师，她对我要求严格，在软件从分析到设计阶段都给予了我很大帮助，在我遇到困难时也给了我正确的方向。在最后写毕业论文时，年老师严谨的治学精神也让我印象深刻。正是年老师的积极督促和认真负责的态度，才使我能按时完成毕设每个阶段的任务，最终圆满完成毕业设计。

# 参考文献

[1]吴迪.建立个人电子病历管理系统[J].中国医疗设备信息,2005,20(4):47-48.

[2]顾宝红.卫生院健康档案管理信息系统设计与实现[J]. 才智,2012,(27).

[3]吴思静,郭清,赵发林,等.国内外健康状况管理现状[J].健康研究,2010,(5):321-323.

[4](美)内格尔. .NET技术[M].北京：清华大学出版社,2013.

[5]严蔚敏,吴伟民.Windows项目开发[M].北京:清华大学出版社,2011.

[6]李乃文,刘好增.C#实践教程(第2版)[M].北京:清华大学出版社，2016.

[7]Karle Watson，Christian Nagel,齐立波,等.SQL Server数据库编程[M].北京：清华大学出版社,2008.

[8]曾宪权,李梅莲.C#应用开发与实践[M].北京:清华大学出版社，2015.

[9]何涛.基于SQL Server用C#实现文件夹中图片存储到数据库的研究与应用[J].计算机

与信息技术,2009,(10):56-58.

[10]赖均,陶春梅.软件工程[M].北京:清华大学出版社，2016.

[11]骆斌.需求分析——软件建模与分析[M].北京:高等教育出版社，2009.

[12]胡荷芬,高雯.UML面向对象分析与设计教程[M].北京:清华大学出版社，2012.

[13](美)内格尔.关于概念模型[M].北京:清华大学出版社,2012.

[14]王珊,萨师煊.数据库系统概论[M].北京:高等教育出版社，2011.

[15] LD Tan, RZ Guo, S Wang,等.Design and implementation of equipments′ electronic archive system based on C# and SQL Server[J]. Modern Electronics Technique,2014,12(3):21-23.