温州医科大学联想智慧医疗研究院

概要设计说明书

编号：LMP-TMP-HDESIGN

版本：1.0

文档控制

|  |  |  |  |
| --- | --- | --- | --- |
| **作者：** |  | **日期：** |  |
| **审批：** |  | **日期：** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **变更说明** | **作者** |
| 2014-8-11 | 1.0 | 创建 | 陈希章 |
| 2014-8-11 | 1.0 | 在需求讨论中，明确了员工卡是需要基于芯片感应扫描的，本文档针对设计想法进行了明确。参考：“2.2.2 设计思想” | 陈希章 |
| 2014-8-21 |  | 添加了数据模型的部分。参考：6.0 数据结构 | 陈希章 |
|  |  |  |  |

目录

[1 引言 5](#_Toc395793738)

[1.1 编写目的 5](#_Toc395793739)

[1.2 背景 5](#_Toc395793740)

[1.3 术语定义 5](#_Toc395793741)

[2 总体设计 6](#_Toc395793742)

[2.1 系统描述 6](#_Toc395793743)

[2.1.1 需求规格概述 6](#_Toc395793744)

[2.1.2 运行环境规定 6](#_Toc395793745)

[2.1.3 必须满足的国际/国内/企业标准 6](#_Toc395793746)

[2.2 总体设计说明 6](#_Toc395793747)

[2.2.1 基本设计概述 6](#_Toc395793748)

[2.2.2 设计思想 7](#_Toc395793749)

[2.3 系统结构和处理流程 7](#_Toc395793750)

[2.3.1 系统结构 7](#_Toc395793751)

[3 模块设计 8](#_Toc395793752)

[4 接口设计 9](#_Toc395793753)

[4.1 外部接口 9](#_Toc395793754)

[4.1.1 用户接口 9](#_Toc395793755)

[4.1.2 硬件接口 9](#_Toc395793756)

[4.1.3 软件接口 9](#_Toc395793757)

[4.1.4 通讯接口 9](#_Toc395793758)

[4.2 内部接口 9](#_Toc395793759)

[5 属性设计（系统设计原则的考虑） 10](#_Toc395793760)

[5.1 可靠性 10](#_Toc395793761)

[5.2 安全性 10](#_Toc395793762)

[5.3 可移植性 11](#_Toc395793763)

[5.4 可测试性 11](#_Toc395793764)

[5.5 其他元素 11](#_Toc395793765)

[6 系统数据结构 12](#_Toc395793766)

[6.1 逻辑结构设计要点 12](#_Toc395793767)

[6.2 物理结构设计要点 12](#_Toc395793768)

[6.3 数据结构与模块的关系 12](#_Toc395793769)

[7 运行设计 12](#_Toc395793770)

[8 系统出错处理 12](#_Toc395793771)

[9 开发环境 13](#_Toc395793772)

[9.1 硬件环境 13](#_Toc395793773)

[9.2 软件环境 13](#_Toc395793774)

[10 系统调试与测试方法 14](#_Toc395793775)

[10.1 调试方法 14](#_Toc395793776)

[10.2 测试方法 14](#_Toc395793777)

[11 参考资料 14](#_Toc395793778)

# 引言

## 编写目的

编写此文档的目的是结合“移动护理需求规格说明书”提出该系统的概要设计，作为后续详细设计的依据。此文档的目标阅读对象为：交互设计师，开发团队，测试团队。

## 背景

该项目的背景如下

A. 待开发软件系统名称：移动护理系统。

B. 商业应用背景介绍：因应移动化的技术趋势，我们开发的“移动护理系统”将对现有护理系统进行有益的补充，提高护理工作的效率和质量。

C. 该系统基本概念，如该系统的类型、从属地位等：这个系统是依托于医院原有的护理系统（住院护士工作站和护理电子病历系统）的，它不是一个独立的系统，而主要是提供了从移动设备这个角度来增加护理的功能。

D. 预期的使用环境、使用者: 这个系统的预期使用环境是在搭载了Android系统的手持设备上面，目标用户主要是护士。

## 术语定义

医嘱：就是医生根据病情和治疗的需要对病人在饮食、用药、化验等方面的指示。这个系统中所涉及的医嘱主要包含了如下几种：药品医嘱、皮试医嘱、治疗医嘱、特检医嘱、化验医嘱、输血医嘱。

# 总体设计

概述总体设计的功能结构、系统逻辑（分层）系统、网络拓朴结构。概述系统运行软/硬件开发环境与开发工具。

给出系统结构总体框图（包括软件、硬件结构框图），说明本系统的各模块的划分，扼要说明每个系统模块的标识符和功能，分层次地给出各模块之间的控制与被控制关系。

## 系统描述

### 需求规格概述

这个系统的核心两个功能需求是：医嘱执行确认和生命体征录入。需求规格详见“移动护理需求规格说明书”。

### 运行环境规定

这个系统需要运行在现代智能设备之上，包括手机或平板。当前这个版本只针对Android设备。

### 必须满足的国际/国内/企业标准

这个系统需要符合“联想智慧医疗软件UI（用户界面）设计规范”。

## 总体设计说明

### 基本设计概述

系统：这里所讲的系统，其实具体来说就是一个移动App。这个App能够运行在搭载了Android操作系统的智能手持设备上面。

### 设计思想

这个系统的设计思想主要有如下两点

1. 易于使用。作为对原有护理系统的有益补充，此系统必须充分考虑到手持设备的使用习惯，提供简单易用的界面，尽可能地结合扫描等方式来简化操作。
2. 易于扩展。“移动护理”本身是一个很好的产品方向，此系统在设计伊始就考虑到后期可能的扩展。我们需要考虑的因素主要有：不同屏幕尺寸的适应性；横向和纵向屏幕的适应性；不同操作系统（Android，iOS甚至其他）的互用性。对于员工卡感应扫描这个需求，因为对设备依赖较大，我们将其设计成一个可选的功能，可以根据需要打开或关闭。

## 系统结构和处理流程

### 系统结构

该系统主要有四个模块，分别是用户管理模块（M0001）、医嘱执行确认模块（M0002），生命体征录入模块（M0003）以及数据业务逻辑模块（M0004）.他们的关系如下

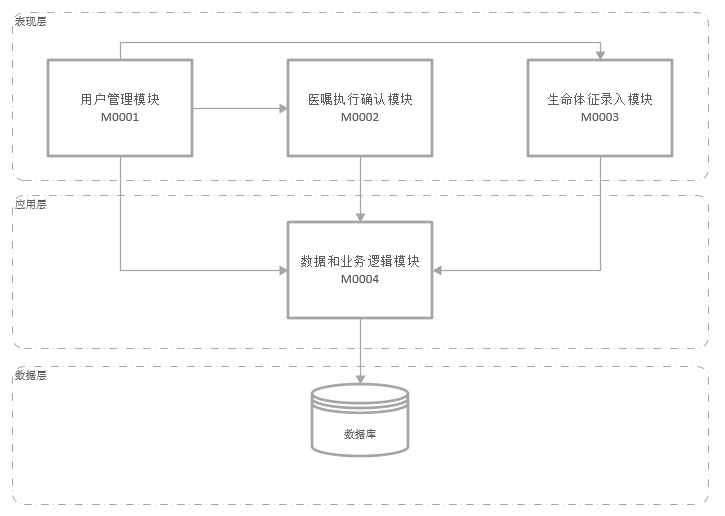


Figure 1系统结构和模块关系图

# 模块设计

该系统主要有四个模块，分别是用户管理模块（M0001）、医嘱执行确认模块（M0002），生命体征录入模块（M0003）以及数据业务逻辑模块（M0004）.

这些模块主要的设计考虑如下

【备注】详细的功能细节，以需求文档为准。

1. **用户管理模块（M0001）**: 这是该系统的一个基础模块。这里的用户通常指的是护士。使用该系统的用户必须首先登录系统。登录系统的方式可以使用账号和密码、也可以用员工卡扫描(芯片)的方式（这种方式是可选的，只有具备了相应的设备才会启用）。同时考虑到私密性，该系统在进入后台被唤醒，或锁屏重新进入、或特定的超时时间后，或者用户自己退出后，要求再次登录。该模块不包含用户注册的功能。这个模块是安装在前端的智能手持设备上面的。
2. **医嘱执行确认模块（M0002）**：这是该系统的一个核心业务模块。医嘱的执行确认，主要的方式分为两大类，有条码的和无条码的。有条码的，是需要先扫描对应的医嘱条形码，然后扫描病人的条形码，然后进行有关操作；无条码的就是先获取病人有关基本信息（通过扫描或者手工搜索得到），然后查询得到其相应的未执行医嘱，并且进行执行确认。在该系统中，医嘱分为六类，而某些类别中又有小类，某些执行确认过程中需要有双签名的机制保证安全性。这个模块是安装在前端的智能手持设备上面的。
3. **生命体征录入模块（M0003）**：这是该系统的一个核心业务模块。生命体征的录入分为常规生命体征和等级护理生命体征。这个模块是安装在前端的智能手持设备上面的。
4. **数据业务逻辑模块（M0004）**：这是考虑扩展性而单独设计的一个中间层模块，用来统一处理和交换数据。这个模块将采用服务来实现，部署在后端的应用服务器中。

# 接口设计

## 外部接口

### 用户接口

该系统对于用户来说，是一个标准的App，将遵循App用户接口（界面）设计的一般规范，以及“联想智慧医疗软件UI（用户界面）设计规范”的要求。具体的用户接口（界面）设计，以“交互设计”最终定稿为准。

### 硬件接口

该系统所依赖的硬件接口主要有：摄像头。主要的作用在于扫描一维码或者二维码。目前对该接口的最低要求是800万像素，设计的时候以智能手持设备自带的摄像头进行考虑。如果有精度较高、响应速度较高的专用硬件则是更好的。

### 软件接口

该系统所使用的数据绝大部分都是直接来自原有的护理系统（住院护士工作站、住院电子病历系统等），所以这部分的数据接口是该系统所主要依赖的软件接口。在整体设计上，这部分接口我们会采用服务的方式来实现。

该系统不提供对外的软件接口。

### 通讯接口

该系统需要有网络通讯接口支持，由于是部署在医院内部使用，所以采用的网络通讯接口为医院特定的无线网络（WIFI）。

## 内部接口

该系统内部的模块之间，M0001和M0002、M0003这三个模块的接口全部基于Android系统开发的内部机制来实现，没有专门的内部接口设计。M0001、M0002、M0003与M0004之间是通过标准的服务接口来实现的（具体的接口定义在详细设计中体现），服务接口所采用的开发技术是ASP.NET Web API。

# 属性设计（系统设计原则的考虑）

说明系统对以下五个方面的属性所作的特殊设计和考虑。

## 可靠性

我们从以下方面来保证可靠性：

1. 数据层的可靠性：由Oracle平台提供高可用性保障。（这个不属于该系统负责的范畴）。
2. 应用层的可靠性：该应用部署到医院的主要的应用服务器上面，提供了较好的软硬件配置，以及负载均衡的配置。
3. 表现层的可靠性：使用一套标准的框架（包含了预防崩溃的机制），调用接口时要求返回标准的JSON格式。

## 安全性

我们以下方面来保证安全性：

1. 用户是需要登录的：该系统使用内部认证过的用户账号和密码登录，确保了只有医院的合法用户才能使用。而且本地不保存密码。
2. 调用服务采用HTTPS，保证数据的安全性。
3. 重新登录的机制：在下列的情况下用户都是需要重新登录的
   1. 系统后台运行
   2. 锁屏
   3. 超过规定时间
4. 用户5次输入错误会强制退出。
5. 扫描患者的二维码的时候，语音提示患者的姓名和床位号，以作确认。
6. 部分操作采用双签名的机制保证操作的安全性。
7. 所有操作都是实时完成的，数据不保存在本地。

## 可移植性

我们从以下方面来保证可移植性：

1. 采用了分层设计，将数据和业务逻辑单独做成一个模块，这样就可以为移植到其他平台（例如iOS）提供了可能性。
2. 采用界面UI和界面逻辑的分离设计，为移植到不同屏幕尺寸的目标设备提供了可能性。

## 可测试性

1. 表现层的App的可测试性
   1. 目前使用内置的程序框架（自主研发的，拥有完整知识产权），包含了一个日志开关“Debug模式”，包含了必要的操作日志，以便在测试的时候发现问题的原因。还可以通过直接用开发工具附加这App，用logcat来进行查看。
   2. 与M0004(数据和业务模块) 采用松耦合的关系，App可以切换两种模式，一种是本地模式，不连接远程的这个服务，使用本地的模拟的数据（采用IoC设计模式；另外一种是远程模式，可以用于集成测试。
2. 应用层服务的可测试性
   1. 因为是完全基于HTTP的模式，输入和输出都是标准的JSON格式，适合做自动化测试。
   2. 同时可以做压力测试。

## 其他元素

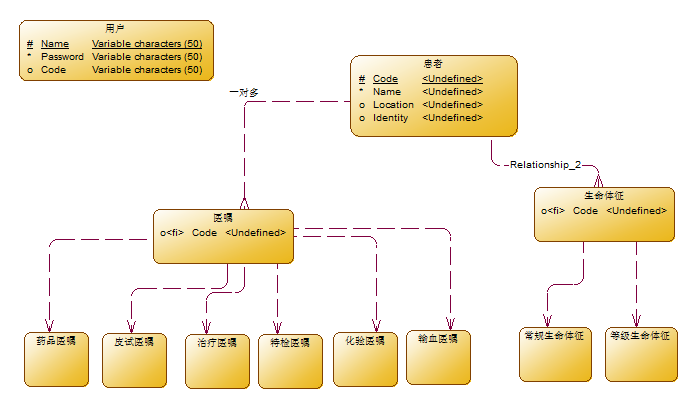
在具体设计开发时，我们需要保证如下的一些附加的属性

1. 一旦网络断开，系统处于离线状态，马上进行提醒
2. 扫描响应时间时间小于2秒；
3. 输入响应时间时间小于2秒；
4. 数据保存响应时间小于3秒；
5. 数据查询响应时间小于5秒；
   * 默认显示10条数据，如果需要查看更多，则进行翻页
6. 数据统计响应时间小于30秒。

# 系统数据结构

## 逻辑结构设计要点

逻辑模型如下：



## 物理结构设计要点

这些数据目前都在现有数据库中存在，本系统主要负责读取，个别的时候需要更新*。*

## 数据结构与模块的关系

可采用矩阵图形式, 并规定以下符号：

C 创建 U 更新 R 读出使用

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模块 | 用户 | 患者 | 医嘱 | 生命体征 |
| 用户管理 | R |  |  |  |
| 医嘱执行录入 |  | R | R/U |  |
| 生命体征录入 |  | R |  | R/U |

# 运行设计

参考需求文档中的流程图，以及交互设计的文档。

# 系统出错处理

|  |  |
| --- | --- |
| 出错类别 | 出错处理 |
| 数据服务调用接口出错 | 应该合理地向表现层发送错误消息,同时在服务端记录完整的日志。每个方法的返回值都是统一的下面的格式  {  HasError: true/false,  ErrorMessage:string,  Data:{  }  } |
| 应用服务器无法正常响应（宕机） | 根据timeout时间，主动地告诉用户目前服务不可用。 |
| 网络异常（无法连接） | 在App中弹出提示，用户必须进行响应，响应后又会检测网络连通状况，如果还是无法连接，则再次弹出提示。同时，如果不响应，后台自动监控事件，如果连接上了，则自动关闭提示。 |

# 开发环境

## 硬件环境

本系统开发所需要的硬件平台、工具如下

1. 开发用：PC 或者Mac
2. 测试用：服务器
3. 测试用设备：Android手机或平板

## 软件环境

本系统开发所需要的软件平台（操作系统、数据库、网络平台、相关软件配套版本等）、软件辅助工具及使用周期。

1. 操作系统：Windows 7 或者Mac OS
2. 数据库：Oracle 11g
3. 网络平台：无线网络或者内部网络
4. 开发工具：Eclipse（表现层） ，Visual Studio 2013（应用层）
5. 设计工具：Power Designer 16.5 (数据模型)

# 系统调试与测试方法

## 调试方法

1. 在Eclipse和Visual Studio 2013中利用内置的调试工具进行调试。

## 测试方法

1. 在App里面切换到远程模式（连接真实的测试数据库）。
2. 测试组按照测试计划和测试用例，逐一进行测试。
3. 如果Bug数量连续3天稳定在5个以下（而且没有严重的Bug）,发动需求部门和研究院内部员工进行测试。

# 参考资料

无。