

数据库应用综合专题学期实训

**项目名称： IB智慧公交系统**

**专 业： 物 联 网 工 程**

**班 级： 1706**

**学 号： 2017115050614**

**姓 名： 杨雨婷**

**指导老师： 董元和**

**日 期： 2019年6月25日**

目录

[1 引言 1](#_Toc12443463)

[1.1 项目描述 1](#_Toc12443464)

[1.2 编写目的 1](#_Toc12443465)

[1.3 用户角色描述 1](#_Toc12443466)

[2 任务概述 1](#_Toc12443467)

[2.1 目标 1](#_Toc12443468)

[2.2 运行环境 2](#_Toc12443469)

[2.3 工具 2](#_Toc12443470)

[2.4 系统功能 3](#_Toc12443471)

[3 系统需求分析 3](#_Toc12443472)

[3.1 用例图 3](#_Toc12443473)

[3.2 概念结构设计 4](#_Toc12443474)

[3.2.1 分析四种元素 4](#_Toc12443475)

[3.2.2 顶层数据流图 5](#_Toc12443476)

[3.2.3 主要功能 6](#_Toc12443477)

[3.2.4 三层结构 6](#_Toc12443478)

[3.3 数据字典 7](#_Toc12443479)

[4 数据库的设计 8](#_Toc12443480)

[4.1 概念模型 8](#_Toc12443481)

[4.1.1 E-R 图（关系） 8](#_Toc12443482)

[4.1.2 E- R图（具体） 10](#_Toc12443483)

[4.2 逻辑模型 10](#_Toc12443484)

[4.2.1 关系模式 10](#_Toc12443485)

[4.2.2 E-R 图 10](#_Toc12443486)

[4.3 物理模型 11](#_Toc12443487)

[5 数据库的实现 12](#_Toc12443488)

[5.1 创建用户表 12](#_Toc12443489)

[5.2 创建线路表 12](#_Toc12443490)

[5.3 创建权限表 13](#_Toc12443491)

[5.4 创建角色表 13](#_Toc12443492)

[5.5 创建排班表 13](#_Toc12443493)

[5.6 创建站点表 14](#_Toc12443494)

[5.7 创建用户表 14](#_Toc12443495)

[5.8 创建视图权限视图 14](#_Toc12443496)

[5.9 创建线路站点关系 15](#_Toc12443497)

[5.10 创建角色权限关系表 15](#_Toc12443498)

[6 项目实现 15](#_Toc12443499)

[6.1 连接数据库 15](#_Toc12443500)

[6.2 实现增添 16](#_Toc12443501)

[6.3 实现删除 17](#_Toc12443502)

[6.4 实现查看 17](#_Toc12443503)

[6.5 实现修改 18](#_Toc12443504)

[7 项目总结 19](#_Toc12443505)

# 

# 1 引言

## 1.1 项目描述

智慧公交项目（简称IB）是由大连智森科技有限公司为某城市公交公司开发的应用软件。旨在提高员工的日常工作便捷性、工作效率，降低管理及运营成本。本软件系统将为该公司提供车辆信息管理、员工信息管理、线路信息管理、站点信息管理、排班方案管理等功能，同时提供了近场通讯功能（NFC）。利用这些功能充分满足该公交公司日常工作中的管理及信息同步需求。

## 1.2 编写目的

本文档旨为某城市公交公司的智慧公交软件的开发提供开发和实施依据，明确项目开发目标。通过本文档对此系统的业务流程做详细、准确的描述，为开发、设计、部署及运维人员提供指导，为计算开发工时和编制计划进度提供基础，同时便于日后对项目内容的扩充、移植和推广。

## 1.3 用户角色描述

|  |  |  |
| --- | --- | --- |
| 用户角色 | | 平台中用户的角色描述 |
| 内部用户 | 管理员 | 平台最大的权限角色，可以进行平台管理 |
| 调度员 | 可以管理排班方案 |
| 驾驶员 | 只能够查询排班方案 |

# 2 任务概述

## 2.1 目标

通过互联网和辽宁科技大学校园网进行各学院毕业设计选题、中期、答辩和后期的流程管理。各阶段都要教务长来开通和关闭，对整个毕业设计的流程进行管理。其中系统的用户信息来自于现教务管理系统。

## 2.2 运行环境

1. 客户端及服务器：

CPU：Intel i3 以上

内存：4G DDR3以上

硬盘：80G以上

显示器：1366\*768以上分辨率

1. 数据通信设备：

本系统平台是建立在INTENET的基础之上，组建INTERNET的设备数量，包括交换机的数量，网线的数量，网络终端的数量等等会随着需求方员工数量的增加或场地的变更而改变，以上改变并不包含在本系统的考虑范围之内。

1. 功能键和专用硬件：

无

## 2.3 工具

1. 设计工具：Visio 2016、MindManager 18、Power Designer 15.1
2. 开发工具：Eclipse Oxygen.1a Release (4.7.1a)
3. 数据库系统：无
4. 操作系统：Windows 7

## 2.4 系统功能

表格 2-1 系统功能表



# 3 系统需求分析

## 用例图

用例图(Use Case Diagram)是 由软件需求分析到最终实现的第一步，它描述人们如何使用一个系统。 用例图包含六个元素，分别是：参与者 (Actor)、用例(Use Case)、关联关系(Association)、包含关系(Include)、扩展关系(Extend)以及泛化关系 (Generalization)。

3-1 用例图



## 3.2 概念结构设计

### 3.2.1 分析四种元素

表格 3-1 数据流图分析

|  |  |
| --- | --- |
| **源点/终点** | **处理** |
| 管理员  调度员 | 处理事务  排班信息 |
| **数据流** | **数据存储** |
| 人员报表  用户编号  登录名  登录密码  姓名  手机号  身份证  角色  驾龄  状态  车辆报表  车辆编号  车牌号  车型  状态  起运时间  线路报表  线路编号  线路名称  状态  开线时间  运行方向  站点报表  站点编号  站点名称  经度  纬度  所在区域  所在街道  排班报表  排班编号  线路编号  车牌号  趟次  每趟时间 | 信息表  用户信息表 车辆信息表  站点信息表 线路信息表  排班表 |

### 3.2.2 顶层数据流图

管理员可以拥有数据库的所有管理权限，包括查询所有信息表

调度员可以有管理排班信息，并且查询排班信息的权限

驾驶员有录入信息和查询排班信息的权限

图3-2 顶层数据流图分析



### 3.2.3 主要功能

智能公交系统可以分为处理事物和排班两种主要功能

图3-3 主要功能数据流图



### 3.2.4 三层结构

事物处理包括：录入人员信息，录入车辆基本信息，录入站点相关信息和录入线路信息

录入信息的时候将信息存储在用户信息表

录入车辆的时候将信息存储在车辆信息表

录入站点的时候将信息存储在站点信息表

录入线路信息的时候将信息存储在线路信息表

图3-4 第三层数据流图



## 3.3 数据字典

名字: 人员报表

描述：存储用户的属性信息

定义：人员报表 = 用户编号 + 登录名 +登录密码 + 姓名 +手机号+ 身份证 + 角色　+ 驾龄　+ 状态

位置：排班表

名字: 车辆报表

描述: 存储线路的属性信息

定义：线路报表 = 车辆编号 + 车牌号 + 车型 + 状态 + 起运时间

位置：排班表

名

名字：线路表

描述: 存储线路的属性信息

定义：线路报表 =线路编号 +线路名称 +状态 +开线时间+ 运行方向

位置：排班表

名字: 站点报表

描述: 存储站点的属性信息

定义：站点报表 =站点编号 +站点名称 +经度+纬度+所在区域+所在街道

位置：排班表

名字: 排班报表

描述: 存储排班的属性信息

定义：排班报表 =排班编号 +线路编号 +车牌号 +每趟时间+ 趟次

位置：反馈给调度人员

# 4 数据库的设计

## 概念模型

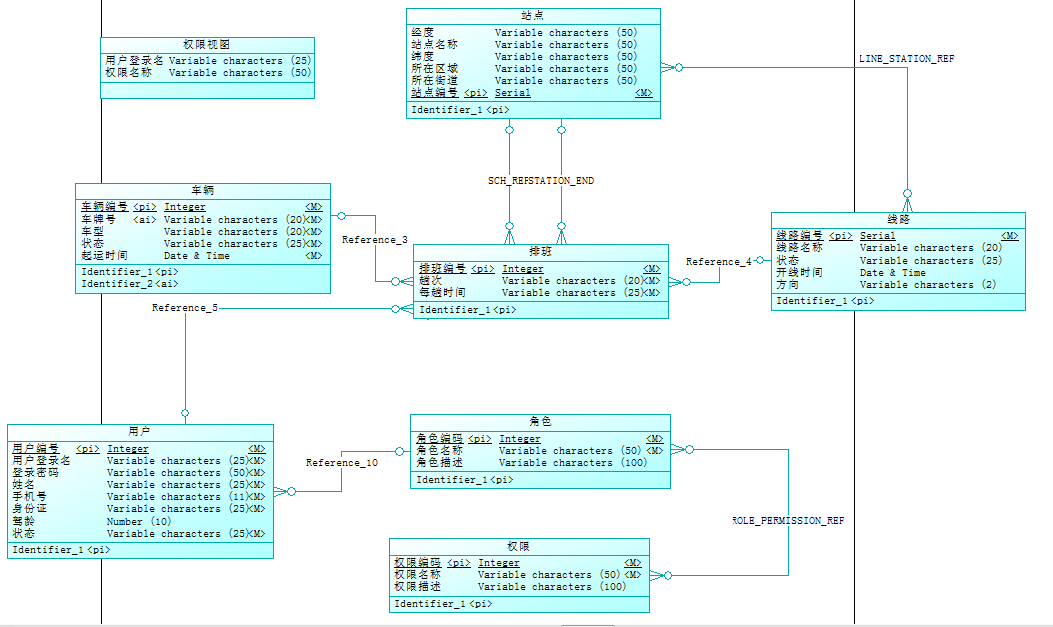
### E-R 图（关系）

概念模型是对真实世界中问题域内的事物的描述，不是对软件设计的描述。  
表示概念模型最常用的是"实体-关系"图。  
E-R图主要是由实体、属性和关系三个要素构成的。

图4-1 关系概念模型 

### 4.1.2 E- R图（具体）

图4-2 具体概念模型



## 逻辑模型

### 4.2.1 关系模式

用户(用户编号,登录名,登录密码,姓名,手机号,身份证, 身份证,角色,驾龄,状态)

角色（角色编码，角色名称，角色描述）

权限（权限编码，权限名称，权限描述）

角色权限关系（关系编码，角色编码，权限编码）

车辆（车辆编号 ， 车牌号 ， 车型，状态，起运时间）

线路（线路编号， 线路名称 ，状态 ，开线时间，运行方向）

站点（站点编号，站点名称 ， 经度，纬度 ， 所在区域， 所在街道）

线路站点关系（编码，线路编号 ， 站点编号 ）

排班（排班编号， 编号， 车牌号，趟次 ，每趟时间，司机编号，始发站 ，终点站 　）

### 4.2.2 E-R 图

逻辑数据模型反映的是系统分析设计人员对数据存储的观点，是对概念数据模型进一步的分解和细化。

图4-3 逻辑模型



## 物理模型

物理模型是对真实数据库的描述。数据库中的一些对象如下：表，视图，字段，数据类型、长度、主键、外键、索引、是否可为空，默认值。  
概念模型到物理模型的转换即是把概念模型中的对象转换成物理模型的对象。

图4-4 物理模型



# 5 数据库的实现

## 5.1 创建用户表

create table BUS

(

busCode int not null,

busLicense varchar(20) not null,

busType varchar(20) not null,

busStatus varchar(25) not null,

startTime datetime not null,

primary key (busCode),

key AK\_busLicense\_nuique (busLicense)

);

## 5.2 创建线路表

create table LINE

(

lineCode int not null auto\_increment,

lineName varchar(20),

status varchar(25),

startLineTime datetime,

direction varchar(2),

primary key (lineCode)

);

## 5.3 创建权限表

create table PERMISSION

(

permissionCode int not null,

permissionName varchar(50) not null,

permissionDescribe varchar(100),

primary key (permissionCode)

);

## 5.4 创建角色表

create table ROLE

(

roleCode int not null,

roleName varchar(50) not null,

roledescribe varchar(100),

primary key (roleCode)

);

## 5.5 创建排班表

create table SCHEDULING

(

pcode int not null,

stationCode int,

STA\_stationCode int,

lineCode int,

code int,

busLicense varchar(20),

cNumber varchar(20) not null,

cTime varchar(25) not null,

primary key (pcode)

);

## 5.6 创建站点表

create table STATION

(

stationCode int not null auto\_increment,

longitude varchar(50),

stationName varchar(50),

latitude varchar(50),

region varchar(50),

street varchar(50),

primary key (stationCode)

);

## 5.7 创建用户表

create table SYSUSER

(

code int not null,

roleCode int,

loginName varchar(25) not null,

password varchar(50) not null,

name varchar(25) not null,

phone varchar(11) not null,

idCard varchar(25) not null,

driving numeric(10,0),

status varchar(25) not null,

primary key (code)

);

## 5.8 创建视图权限视图

create table V\_PERMISSION

(

loginName varchar(25),

permissionName varchar(50)

);

## 5.9 创建线路站点关系

create table 线路站点关系

(

id int not null,

lineCode int not null,

stationCode int not null,

primary key (id)

);

## 5.10 创建角色权限关系表

create table 角色权限关系

(

relationCode int not null,

roleCode int not null,

permissionCode int not null,

primary key (relationCode)

)

# 6 项目实现

## 6.1 连接数据库

当登录用户的时候就要练数据库

**public** **void** connect(User user) {

DaoFactory fac = **new** DaoFactory() ;

**try** {

Connection connect = (Connection) fac.*getConnect*();

Statement stmt = (Statement) connect.createStatement();

ResultSet rs = stmt.executeQuery("select \* from sysuser");

**while** (rs.next()) {

**if**(rs.getString("loginName").equals(userName))

**return** rs.getString("password");

}

}

**catch** (Exception e) {

System.***out***.print("get data error!");

e.printStackTrace();

}

**return** **null**;

}

## 6.2 实现增添

以用户的添加为例

**public** **void** save(User user) {

**try** {

//1.获取数据库连接

DaoFactory fac = **new** DaoFactory() ;

Connection connect = (Connection) fac.*getConnect*();

//3、准备SQl语句

System.***out***.println(user);

//4.1 获取用于执行SQl语句的对象statement 调用connection的creatStatement（）

Statement stmt = (Statement) connect.createStatement();

System.***out***.println("-----------------");

ResultSet rs = stmt.executeQuery("select \* from sysuser");

**int** t = 1;

**while** (rs.next()) {

t ++;

}

//4.2 调用Statement对象的executeUpdate（sql）执行SQL语句进行插入

**int** ttt = 1;

String u = user.getRole();

**if**(u.equals("调度员[1]")) {

ttt = 2;

}

**else** **if**(u.equals("驾驶员[2]")){

ttt = 3;

}

String SQL = "insert into sysuser values (" + t + ", '"

+ user.getLoginName()+ "','"

+ user.getPassword() + "','"+ user.getName()

+ "','" + user.getPhone()

+ "','" + user.getIdCard()

+ "','" + ttt

+ "','" + user.getDriving()

+ "','" + user.getStatus()+"')";

System.***out***.println(SQL);

**int** n = stmt.executeUpdate(SQL);

System.***out***.println(n);

//2.关闭数据库连接

connect.close();

}

**catch** (Exception e) {

System.***out***.print("save data error!");

e.printStackTrace();

}

}

## 6.3 实现删除

以用户的删除为例

**public** **void** dropUser(User user){

//1.获取数据库连接

DaoFactory fac = **new** DaoFactory() ;

Connection connect = (Connection) fac.*getConnect*();

**try** {

//3、准备SQl语句

System.***out***.println(user);

String SQL = "delete from sysuser where code = " + user.getCode();

//4.1 获取用于执行SQl语句的对象statement 调用connection的creatStatement（）

Statement stmt = (Statement) connect.createStatement();

//4.2 调用Statement对象的executeUpdate（sql）执行SQL语句进行插入

**int** n = stmt.executeUpdate(SQL);

System.***out***.println(n);

// }

//2.关闭数据库连接

connect.close();

}

**catch** (Exception e) {

System.***out***.print("get data error!");

e.printStackTrace();

}

}

## 6.4 实现查看

以用户的查看为例

**public** List<User> find(User condition) {

List<User> users = **new** ArrayList<User>();

**try** {

DaoFactory fac = **new** DaoFactory() ;

Connection connect = (Connection) fac.*getConnect*();

Statement stmt = (Statement) connect.createStatement();

ResultSet rs = stmt.executeQuery("select \* from sysuser");

**while** (rs.next()) {

System.***out***.println(condition); **if**( condition.getName().equals("")||rs.getString("loginName").equals(condition.getName()) ) {

User user = **new** User();

user.setCode(rs.getString("code"));

user.setDriving(rs.getInt("driving"));

user.setIdCard(rs.getString("idCard"));

user.setLoginName(rs.getString("loginName"));

user.setName(rs.getString("name"));

user.setPassword(rs.getString("password"));

user.setPhone(rs.getString("phone"));

user.setRole(rs.getString("role"));

user.setStatus(rs.getString("status"));

users.add(user);

}

}

}

**catch** (Exception e) {

System.***out***.print("get data error!");

e.printStackTrace();

}

**return** users;

}

## 6.5 实现修改

**public** **void** update(User user) {

//1.获取数据库连接

DaoFactory fac = **new** DaoFactory() ;

Connection connect = (Connection) fac.*getConnect*();

//3、准备SQl语句

System.***out***.println(user);

String SQL1 = "update sysuser " + " set loginName = '"+ user.getLoginName() +"' where code = "+ user.getCode();

**try** {

//4.1 获取用于执行SQl语句的对象statement 调用connection的creatStatement（）

Statement stmt = (Statement) connect.createStatement();

//4.2 调用Statement对象的executeUpdate（sql）执行SQL语句进行插入

System.***out***.println(SQL1);

stmt.execute(SQL1);

//2.关闭数据库连接

connect.close();

}

**catch** (Exception e) {

System.***out***.print("get data error!");

e.printStackTrace();

}

}

# 7 项目总结

通过这次实训，我们在学习数据库理论的同时，也提高了自己的动手能力，将课本上的知识用到实践中，真正做到了学以致用，让我们也真正的体验了一次做项目的大致流程，对于我们这样的初学者阶段来说，实训让我们对前边的理论知识的一种检测，也是让我们对做项目，做软件有了初步的认识。“不积跬步，无以至千里，不积小流，无以成江河”，其实我们就应该多去参见一些这样的实训，去积累一些经验，经验多了对以后自己的发展，会大有所用，对数据库的应用也会得心应手。其实，不仅仅是数据库，以及下一阶段的java实现都有进一步的巩固，将数据库的设计与代码实现综合起来，使所有所学的只是串联起起来。

通过这次实训，让我们在实践的过程中巩固了理论学习，认识到实现软件开发的所有过程。同时，也让软件工程理论课的学习得到了很好的实践。实践里出真知。软件工程的学习使我们的开发过程更加规范，减少了软件危机，使后期的维护更加容易一点。