目录

**[概要设计说明书 1](#_Toc22663)**

**[一、引言 1](#_Toc32232)**

[1、编写目的 1](#_Toc1625)

[2、项目背景 1](#_Toc3960)

[3、定义： 2](#_Toc10298)

[4、参考资料 3](#_Toc12101)

**[驾校管理系统项目可行性分析报告 3](#_Toc7455)**

**[二、任务概述 3](#_Toc9674)**

[1、目标 3](#_Toc17296)

[2、 运行环境 3](#_Toc30622)

[3、 需求概述 3](#_Toc21829)

[4、条件与限制 4](#_Toc10508)

**[三、总体设计 4](#_Toc6382)**

[1、设计思路 4](#_Toc13140)

[3、 设计实现 5](#_Toc25332)

[4、系统数据模型 6](#_Toc24786)

[6、以上实体之间存在以下关系： 10](#_Toc25964)

**[四、数据结构设计： 10](#_Toc12207)**

[2、算法设计： 13](#_Toc4258)

**[五、运行设计 13](#_Toc5829)**

[1、运行模块组合 13](#_Toc4313)

[2、运行控制 14](#_Toc9939)

[3、运行时间 14](#_Toc6968)

**[六、出错处理设计 14](#_Toc4670)**

[1、出错输出信息 14](#_Toc2815)

[2、由于其他问题，如网络传输超时等，产生的问题，称为硬错误。 14](#_Toc14320)

[3、出错处理对策 14](#_Toc6515)

**[七、维护设计 15](#_Toc29204)**

# 

# 概要设计说明书

### 一、引言

1、编写目的

本设计书是驾校管理系统项目的研发概要设计，将项目开发进程中或者项目结束后提供给双方人员使用，同时也可以作为实施后期的维护人员使用。

2、项目背景

作为软件开发的前期文档，可以帮助程序设计人员和管理人员提供清晰的思路，在软件开发后期的维护阶段也起到至关重要的作用。

委托单位：吉首大学张家界学院 开发单位：理工农学院 负责人：田硕

随着我国经济不断的发展，人们的物质生活水平不断的提高，开始对生活物质条件提出高的要求，我国汽车工业迅猛的发展，汽车成为一种便捷、高效、通常的交通工具进入了人们的日常生活中，人们大量开始学习汽车驾驶技术。为了满足人们对汽车驾驶的要求，汽车驾驶培训行业蓬勃发展，大大小小的汽车驾驶培训学校应运而生，使得汽车驾驶培训市场异常火爆。自从交通部最新出台的《机动车驾驶员培训资格条件》正式实施后，驾校的日常工作变得繁杂。学员从报名到最后取得驾照需要一系列学习过程，其间学员的数量庞大、车辆多、档案的存放与管理、学员的考试安排、教练与学员教学间的调度使得驾校人工管理日益复杂。在早期的管理方式中，很多驾校为了节约成本，采用EXCEL文档进行管理，导致查询、统计、分析繁琐复杂。这中管理方式，工作效率低下，相关资料数据容易丢失。随着时代的发展，计算机技术已经深入到各行各业，为广大用户提供了更为周到便捷的服务，人工已经无法满足爆炸似的信息量繁杂劳动;硬件的高速发展，使计算机的性价比越来越高，为了提高驾校工作效率，保证准确完整的数据，采用计算机来管理驾校的信息和调度日常的教学安排，操作简单，功能简单实用，便于日常操作管理，可以快速处理所需信息，大大节约时间、人力和物力资源。

3、定义：

SOA:面向服务架构（Service Oriented)

CS:客户端服务架构（Client-Sever)

WS:Web服务（Web Service)

GIS:地理信息服务（Geographic Information Service）

WCF:使用托管代码建立和运行面向服务（Service Oriented）应用程序的统一框架。（Windows Communication Foundation)

4、参考资料

驾校管理系统项目可行性分析报告

驾校管理系统项目需求分析说明书

《软件设计模式》 2001/11/1 清华出版社

《android 移动开发入门与进阶》 2009 人民邮电出版社

《java 编程思想》2007 机械工业出版社

《MYSQL 5权威指南（第3版）》2006/12 人民邮电出版社

### 二、任务概述

1、目标

致力于为广大驾考人士提供简洁的驾考在线报名、在线模拟答题、练习功能。

2、 运行环境

系统将由两部分程序组成，安装在客户机上的客户程序及公司内的数据服务器程序。

根据调研得知所有旅行社的计算机配置均在Pentium 133级别以上，客户程序应能够在Pentium 133级别以上, Win NT环境下运行。

3、 需求概述

驾校方面为方便广大驾考人士，需要开发一个驾校信息管理系统。驾考人士可以通过手机或电脑客户端登录本系统，了解驾校的基本信息，并可以进行线上报名，在线模拟答题。

要求系统能有效、快速、安全、可靠和无误的完成上述操作。并要求客户机的界面要简单明了，易于操作，服务器程序利于维护。

4、条件与限制

有一些技术组内成员均未完全掌握；时间限制在两个月以内。

### 三、总体设计

1、设计思路

（1）需求分析 在设计之前，我们需要进行需求分析，确定系统的功能和用户需求，以便于后续的设计、开发和测试。

功能需求：根据驾校的实际需求，确定系统的功能模块，包括但不限于学员管理、教练管理、课程管理、排班管理、教练车管理、教学资源管理、考试管理、财务管理等模块。

用户需求：确定用户对系统的需求，包括但不限于用户体验、界面友好、操作便捷、数据安全等方面。

2、总体设计 在需求分析的基础上，我们进行系统的总体设计，主要包括以下几个方面：

（1）系统架构设计：确定系统的整体架构，包括前端、后端、数据库等。

技术选型：根据系统的功能需求和性能需求，选择合适的技术栈，包括但不限于前端框架、后端框架、数据库等。

（2）数据库设计：设计数据库的表结构，确定数据之间的关系，以及对数据进行的操作，包括增、删、改、查等操作。

接口设计：确定系统的接口设计，包括前端和后端的接口设计，以及接口的参数和返回值等。

（3）用户界面设计：根据用户需求，设计用户界面，包括但不限于登录界面、主页面、菜单栏、数据展示、数据操作等。

（4）安全设计：设计系统的安全策略，包括但不限于用户认证、权限控制、数据加密、防范攻击等。

（5）性能优化设计：针对系统的性能瓶颈进行优化，包括但不限于数据库优化、代码优化等方面。

1. 设计实现

（1）系统定位

驾校系统的目标是提供一个在线的、全面的驾校管理系统，帮助驾校管理者和教练员更有效地管理学员和教学活动。该系统应该提供全面的管理工具，包括学员管理、教练员管理、教学计划和进度管理、车辆管理、考试管理、财务管理等。同时，该系统还应该提供在线学习功能，以帮助学员更好地掌握驾驶技能。

在系统定位中，需要考虑到该系统的目标用户是驾校管理者、教练员和学员。针对不同的用户，需要提供不同的功能和操作界面，以便他们能够更好地使用系统。

（2）数据来源

学员信息：包括学员的个人基本信息、联系方式、身份证信息、驾照类型等；

教练信息：包括教练的个人基本信息、联系方式、身份证信息、教练证等；

车辆信息：包括车辆的基本信息、车牌号、车型、车辆状态等；

排班信息：包括教练和车辆的排班信息，以及学员的预约情况；

教学进度信息：包括学员的学车进度、科目考试情况、学习评价等。

（3）在获取这些数据时，可以通过以下途径：

手动输入：对于一些基本信息，可以由管理员手动输入到系统中；

从第三方数据源获取：例如，从身份证识别系统中获取身份证信息，从车辆管理系统中获取车辆信息；

自动采集：例如，通过摄像头识别车牌号等信息。

为了保证数据的准确性和完整性，需要在系统中建立相应的数据校验和数据管理机制，例如，对于学员信息的输入，需要进行身份证号码、电话号码等信息的格式校验，对于排班信息，需要保证排班的合法性等。同时，需要保证数据的安全性，例如，通过加密技术保护学员的隐私信息。

4、系统数据模型

在本系统中，建议采用MySQL关系型数据库来存储数据。

下面是本系统的数据模型设计：

（1）学员表（student）：

- 学员编号（student\_id）

- 学员姓名（name）

- 学员性别（gender）

- 联系电话（phone）

- 身份证号（id\_card）

- 学员状态（status）

- 注册时间（register\_time）

（2） 教练表（coach）：

- 教练编号（coach\_id）

- 教练姓名（name）

- 教练性别（gender）

- 联系电话（phone）

- 身份证号（id\_card）

- 教练状态（status）

- 注册时间（register\_time）

（3） 车辆表（car）：

- 车辆编号（car\_id）

- 车辆品牌（brand）

- 车辆型号（model）

- 车牌号码（license\_plate）

- 车辆状态（status）

- 注册时间（register\_time）

（4） 课程表（course）：

- 课程编号（course\_id）

- 课程名称（name）

- 课程类型（type）

- 课程价格（price）

- 课程状态（status）

- 创建时间（create\_time）

（5） 预约表（appointment）：

- 预约编号（appointment\_id）

- 学员编号（student\_id）

- 教练编号（coach\_id）

- 车辆编号（car\_id）

- 预约日期（appointment\_date）

- 预约时间（appointment\_time）

- 预约状态（status）

（6）学习进度表（learning\_progress）：

- 学习进度编号（progress\_id）

- 学员编号（student\_id）

- 教练编号（coach\_id）

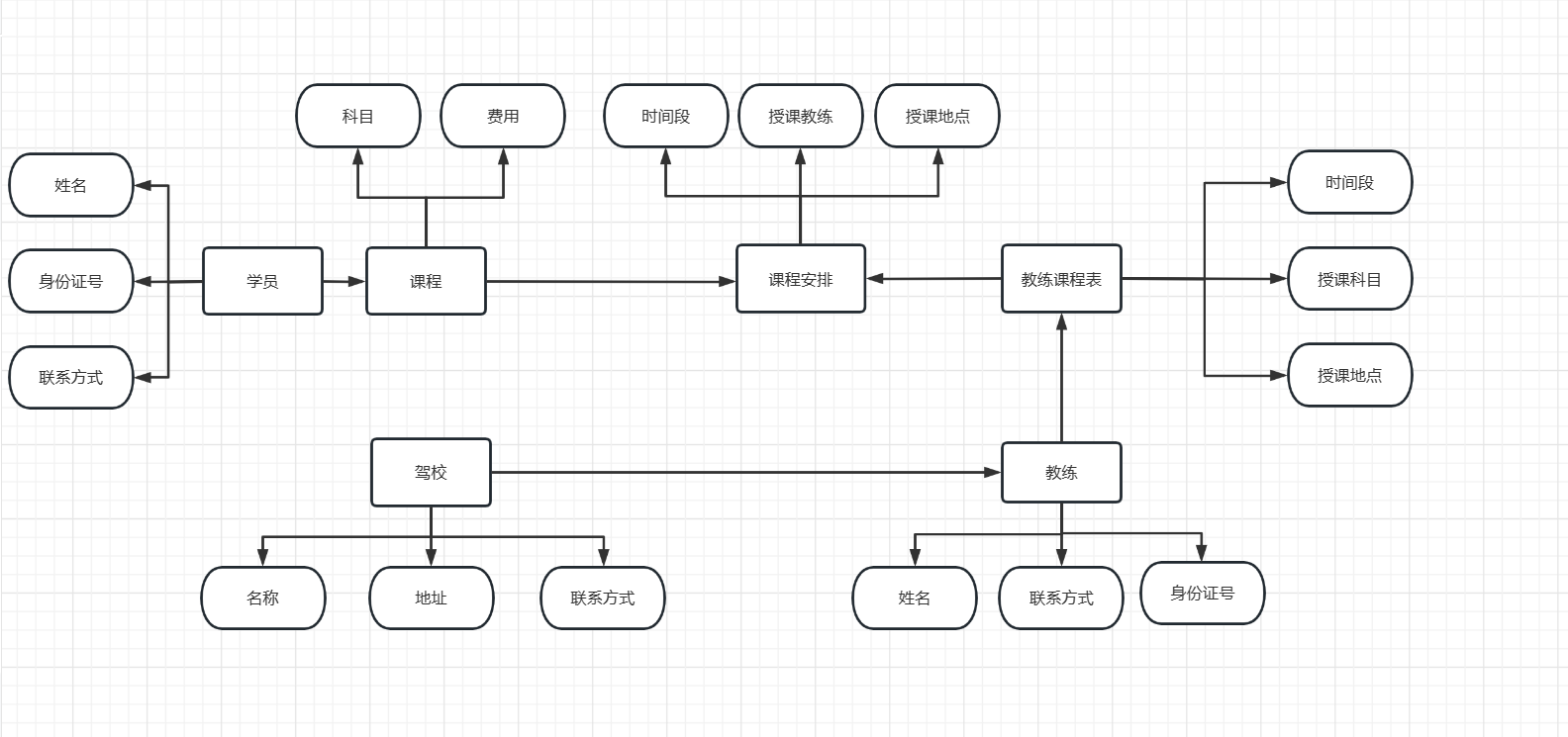
- 车辆编号（car\_id）

- 课程编号（course\_id）

- 学习状态（status）

- 学习时间（learning\_time）

5、系统数据模型可以采用实体-关系（ER）图的形式表示，该图用于描述系统中的实体以及它们之间的关系。下面是该驾校系统的数据模型：



该数据模型包括以下实体：

（1） 学员（Student）- 该实体包含学员的个人信息，如姓名、身份证号、联系方式等。

（2）教练（Instructor）- 该实体包含教练的个人信息，如姓名、身份证号、联系方式等。

（3）教练课程表（Instructor Schedule）- 该实体记录每个教练的授课时间表，包括时间段、授课科目、所在驾校等信息。

（4） 课程（Course）- 该实体包含课程信息，如科目、费用等。

（5） 课程安排（Course Schedule）- 该实体记录每个课程的授课时间表，包括时间段、授课教练、授课地点等信息。

（6） 驾校（Driving School）- 该实体包含驾校的基本信息，如名称、地址、联系方式等。

6、以上实体之间存在以下关系：

（1）学员和教练之间是多对多的关系，因为一个学员可以选多个教练，一个教练也可以教多个学员。

（2） 教练和教练课程表之间是一对多的关系，因为一个教练有多个授课时间表。

（3） 教练和课程安排之间是多对多的关系，因为一个教练可以教授多个课程，一个课程也可以由多个教练教授。

（4）课程和课程安排之间是一对多的关系，因为一个课程有多个授课时间表。

（5） 驾校和教练之间是一对多的关系，因为一个驾校可以有多个教练，一个教练也只能在一个驾校工作。

（6） 驾校和课程之间是多对多的关系，因为一个驾校可以提供多个课程，一个课程也可以由多个驾校提供。

通过该数据模型，我们可以清楚地了解系统中各个实体之间的关系，方便后续的数据库设计和实现。

### 四、数据结构设计：

1、在接口设计中，我们需要考虑输入和输出的数据格式，通常采用JSON格式作为数据交换的标准格式。因此，在数据结构设计上，我们需要定义JSON对象的属性名称和数据类型，如下所示：

（1） 学生信息结构体

学生编号：int

学生姓名：string

学生性别：string

学生身份证号：string

学生电话：string

学生地址：string

学生年龄：int

学生照片：byte[]

（2） 教练信息结构体

教练编号：int

教练姓名：string

教练性别：string

教练身份证号：string

教练电话：string

教练地址：string

教练年龄：int

教练入职时间：datetime

教练照片：byte[]

（3） 车辆信息结构体

车辆编号：int

车牌号：string

车辆类型：string

车辆品牌：string

车辆颜色：string

座位数：int

车辆照片：byte[]

（4）预约信息结构体

预约编号：int

学生编号：int

教练编号：int

车辆编号：int

预约时间：datetime

预约状态：int

（5）培训记录结构体

记录编号：int

学生编号：int

教练编号：int

车辆编号：int

训练开始时间：datetime

训练结束时间：datetime

训练里程：float

2、算法设计：

（1）排序算法

排序算法是对学生信息、教练信息等进行排序，便于查找和统计。常用的排序算法有冒泡排序、选择排序、插入排序、快速排序、归并排序等。

（2）查找算法

在学生信息、教练信息等大量数据中快速查找所需信息。常用的查找算法有顺序查找、二分查找、哈希查找等。

（3）图算法

对于驾校管理系统中的地图信息，可能需要使用图算法来求解最短路径、最小生成树等问题。常用的图算法有Dijkstra算法、Bellman-Ford算法、Floyd算法、Prim算法、Kruskal算法等。

（4）贪心算法

在一些优化问题中，可能会用到贪心算法来求解最优解。比如在排课问题中，根据某些规则贪心地选择课程安排，可以得到较好的效果。

（5）动态规划算法

在一些需要最优解的问题中，可能会用到动态规划算法。比如在学员分班问题中，可以用动态规划来得到分班方案。

### 五、运行设计

1、运行模块组合

本程序主要是以一个窗口为模块，一般一个窗口完成一个特定的功能，主窗口通过打开另一个子窗口来实现个模块之间不同功能的连接和组合。各模块之间相对独立，程序的可移植性好。各模块之间主要以传递数据项的引用来实现模块之间的合作和数据共享。

2、运行控制

运行控制将严格按照各模块间的函数调用关系来实现。

在网络传输方面，客户机再发送数据后，将等待服务器的确认到信号，收到后，在此等待服务器发送数据，然后对数据进行确认服务器再接收到数据后发送确认信号，在对数据处理、访问数据库后，将返回信息送回客户机，并等待确认。

3、运行时间

各模块运行时间不定，这也跟用户的操作以及数据的大小有关。

### 六、出错处理设计

1、出错输出信息

程序在运行时主要会出现两种错误：1、由于输入信息，或无法满足要求时产生的错误，称为软错误。

2、由于其他问题，如网络传输超时等，产生的问题，称为硬错误。

对于软错误，须在登录/注册操作时判断及输入数据验证模块由数据进行数据分析，判断错误类型，再生成相应的错误提示语句，送到输出模块中。

对与硬错误，可在出错的相应模块中输出简单的出错语句，并将程序重置。返回输入阶段。

3、出错处理对策

所有的客户机及服务器都必须安装不间断电源以防止停电或电压不稳造成的数据丢失的损失。若真断电时，客户机上将不会有太大的影响，主要是服务器上：在断电后恢复过程可采用 SQL SERVER 的日志文件，对其进行ROLLBACK 处理，对数据进行恢复。

在网络传输方面，可考虑建立一条成本较低的后备网络，以保证当主网络断路时数据的通信。

在硬件方面要选择较可靠、稳定的服务器机种，保证系统运行时的可靠性。

### 七、维护设计

维护方面主要为对服务器上的数据库数据进行维护。可使用 SQL SERVER 的数据库维护功能机制。例如，定期为数据库进行Backup，维护管理数据库死锁问题和维护数据库内数据的一致性等。