

**浙江万里学院《移动应用开发管理》**  
**（浙江万里学院教务管理学生子系统概要设计书）**  
**封面**

学生填写：

姓名	邓骑林	学号	2021016112
所在学院	信息与智能工程学院	班级	通信 211
课程名称	移动应用开发管理		
任课教师	熊波	课程学分	2
上课时间	20 <u>23</u> 至 20 <u>24</u> 学年第二学期		
递交时间	2024 年 3 月 31 日		

教师评价：

得 分\_\_\_\_\_

教师签名\_\_\_\_\_

# 浙江万里学院教务管理学生子系统概要设计书

## 1. 引言

### 1.1 编写目的

本概要设计书的编写旨在对浙江万里学院教务管理学生子系统进行详细规划和设计，以确保系统开发过程中有明确的指导方向。通过概要设计，我们能够清晰地定义系统的功能需求、技术选型以及系统结构，为后续的详细设计和开发工作提供基础和指导。

### 1.2 背景

随着信息化技术的发展，教务管理系统在高校中扮演着重要的角色，为学校教务管理工作提供了便捷和高效的解决方案。浙江万里学院作为一所现代化的高校，需要一个完善的教务管理系统来支持学院的教学管理工作。因此，本概要设计书针对学院的教务管理学生子系统展开设计与规划。

### 1.3 参考资料

在编写本概要设计书时，我们参考了以下资料：

- 浙江万里学院教务管理系统需求文档
- 相关领域的教务管理系统设计文档和实践经验
- 现有教务管理系统的案例分析和评估报告
- 技术文档和书籍，包括 Java 开发、Spring 框架、数据库设计等相关内容

本文档将综合以上参考资料，结合浙江万里学院的实际需求，进行系统概要设计的详细阐述和规划。

## 2. 总体设计

### 2.1 需求规定

浙江万里学院教务管理学生子系统的需求规定如下：

- 系统应能够对学生的基本信息、课程信息和成绩信息进行有效管理和查询。
- 系统应具有良好的用户界面，方便用户进行操作。

- 系统应具有权限管理功能，保证不同用户有不同的操作权限。

## 2.2 运行环境

### 2.2.1 前端环境

- Web 浏览器：支持 HTML5、CSS3 和 JavaScript 的现代浏览器。

### 2.2.2 后端环境

- 服务器：支持 Java 运行环境的服务器，如 Tomcat。
- 数据库：MySQL 数据库。

## 2.3 基本设计概念和处理流程

### 2.3.1 基本设计概念

- MVC 架构：采用 MVC（Model-View-Controller）架构，将系统分为模型层、视图层和控制层，实现数据、显示和控制的分离。
- RESTful API：设计符合 RESTful 风格的 API 接口，实现前后端分离和系统的灵活性。

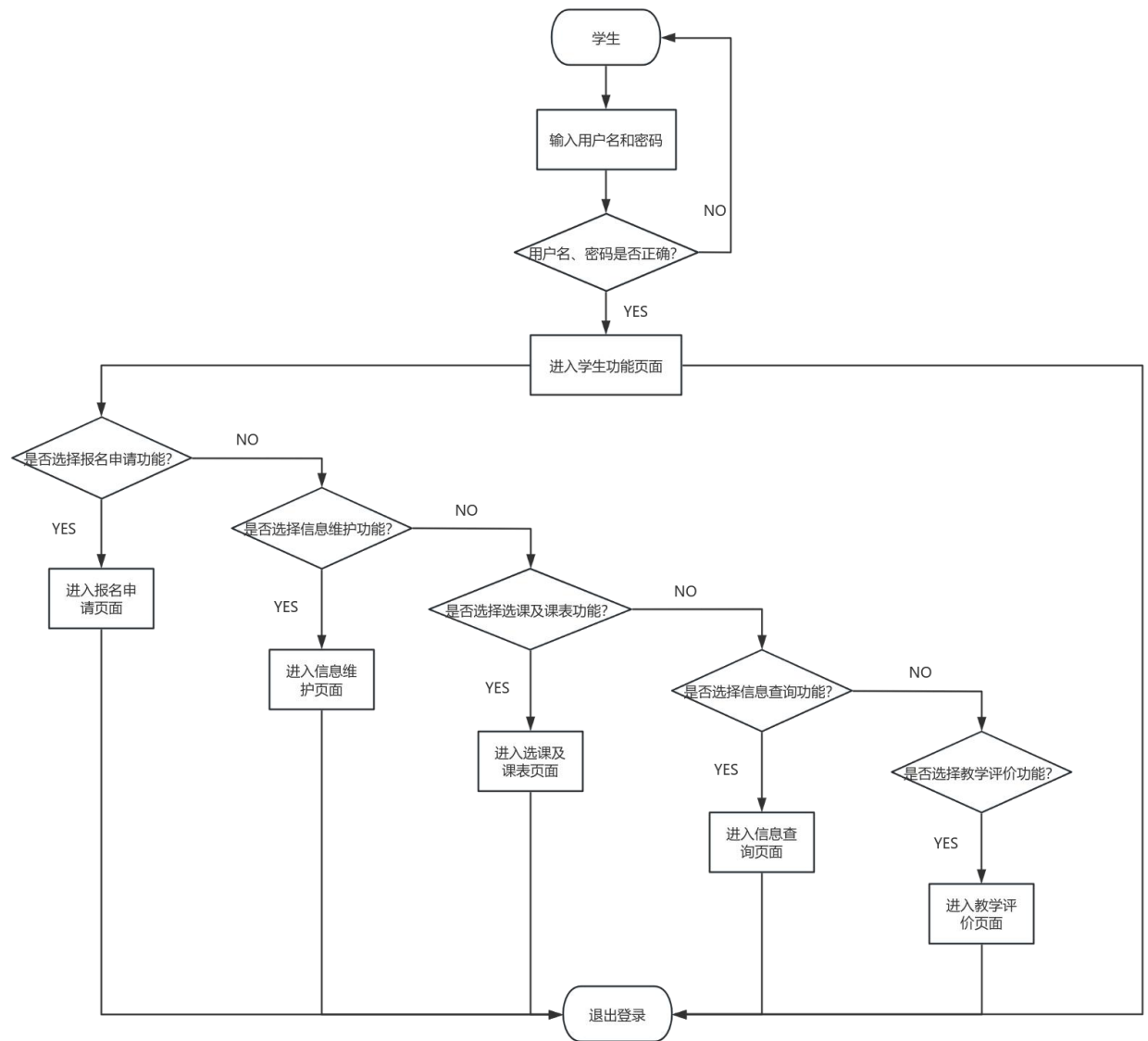
《浙江万里学院教务管理学生子系统》主要功能如下图所示



### 2.3.2 处理流程

1. 用户通过浏览器访问系统的前端页面。
2. 前端页面向后端发送请求，后端根据请求调用相应的控制器处理逻辑。
3. 控制器从模型层获取数据，并进行相应的处理和计算。
4. 控制器将处理后的数据返回给前端页面，页面进行展示。

处理流程（见下图）



## 2.4 结构

浙江万里学院教务管理学生子系统的整体结构包括：

- 前端页面：采用 HTML、CSS 和 JavaScript 构建，实现用户界面。
- 后端服务：采用 Java 语言和 Spring 框架构建，实现系统的业务逻辑和数据处理。
- 数据库：采用 MySQL 数据库存储系统的数据。

## 学生教务子系统

NO.	模块名称	功能需求	程序 ID
1	报名申请	学籍异动申请、学生证补办申请、辅修报名等	StudentEnrollment_1
2	信息维护	查询个人信息、专业（大类）分流确认等	StudentManage_1
3	选课及课表	学生个人课表查询、自主选课等	StudentCourse_1
4	信息查询	学生学业情况查询、学生成绩查询、考试安排信息查询等	StudentManage_2
5	教学评价	学生评价、选择喜欢的老师（学生）等	StudentAssessment_1

### 2.5 人工处理过程

系统中可能涉及到的人工处理过程包括：

- 管理员录入学生信息、课程信息和成绩信息。
- 教师录入学生成绩和发布课程信息。
- 学生查询个人信息和成绩信息。

### 2.6 尚未解决问题

目前在系统设计过程中尚未解决的问题包括：

- 如何确保系统的安全性和稳定性？
- 如何进行系统的性能优化？
- 如何处理并发访问和数据一致性问题？

### 3. 接口设计

#### 3.1 用户接口

- 用户接口主要包括系统的前端界面，以及用户与系统进行交互的方式。

##### 3.1.1 前端界面

- 系统的前端界面采用 Web 页面的形式，包括登录界面、主页面和各个功能模块的页面。用户通过浏览器访问这些页面，与系统进行交互。

##### 3.1.2 用户操作

- 用户通过前端页面进行各种操作，如输入账号密码登录系统、填写表单录入信息、点击按钮提交请求等。

#### 3.2 外部接口

- 系统需要与外部系统或服务进行数据交换和通信。

##### 3.2.1 数据库接口

- 系统与数据库之间的接口，用于进行数据的读写操作。采用 SQL 语言进行数据库操作。

##### 3.2.2 第三方服务接口

- 系统可能需要与第三方服务进行集成，如短信服务、邮件服务等，用于发送通知或验证信息。

#### 3.3 内部接口

- 系统内部不同模块之间需要进行通信和数据交换。

##### 3.3.1 后端接口

- 系统内部的后端接口，用于不同模块之间的数据传递和调用。采用 Java 语言的接口形式。

##### 3.3.2 RESTful API

- 系统提供的 RESTful 风格的 API 接口，用于与前端页面进行数据交换和通信。采用 HTTP 协议进行通信。

### 4. 运行设计

#### 4.1 运行模块组合

系统的运行模块组合包括前端界面、后端服务和数据库三部分。

- 前端界面：负责与用户进行交互，向用户展示页面并接收用户的操作请求。
- 后端服务：包括控制器、业务逻辑和数据访问层，负责处理用户的请求，执行相应的业务逻辑并与数据库进行交互。
- 数据库：存储系统的数据，包括学生信息、课程信息和成绩信息等。

#### 4.2 运行控制

系统的运行控制由后端服务来实现，具体包括以下方面：

- 请求处理：后端控制器接收到前端的请求后，根据请求的类型和参数进行相应的处理，并调用相应的业务逻辑处理器进行具体的业务逻辑处理。
- 数据交互：后端服务与数据库之间进行数据的读写交互，通过数据库访问对象（DAO）来实现对数据库的操作。
- 异常处理：后端服务需要处理各种可能出现的异常情况，如数据库连接异常、数据验证异常等，进行适当的异常处理并返回相应的错误信息给前端。

#### 4.3 运行时间

系统的运行时间主要取决于以下几个因素：

- 服务器性能：服务器的处理能力和资源配置会直接影响系统的响应速度和并发处理能力。
- 数据库访问：数据库的读写性能和响应速度会影响系统的数据交互效率。
- 网络延迟：用户访问系统的网络延迟也会影响系统的响应时间。

通常情况下，系统的运行时间应尽量优化，以保证用户体验和系统的高可用性。

### 5. 系统数据结构设计

#### 5.1 逻辑结构设计

- 系统的逻辑结构设计主要包括对数据进行抽象和组织的逻辑模型。

##### 5.1.1 学生信息逻辑结构

- 学生信息：包括学号、姓名、性别、班级等字段。

##### 5.1.2 课程信息逻辑结构

- 课程信息：包括课程代码、课程名称、学分、授课教师等字段。

### 5.1.3 成绩信息逻辑结构

- 成绩信息：包括学号、课程代码、平时成绩、考试成绩、总成绩等字段。

## 5.2 物理结构设计

- 系统的物理结构设计主要指数据在存储介质上的组织方式。

### 5.2.1 数据库表设计

- 学生信息表（Student）：包括学号（Primary Key）、姓名、性别、班级等字段。
- 课程信息表（Course）：包括课程代码（Primary Key）、课程名称、学分、授课教师等字段。
- 成绩信息表（Grade）：包括学号（Foreign Key）、课程代码（Foreign Key）、平时成绩、考试成绩、总成绩等字段。

## 5.3 数据结构与程序关系

- 系统的数据结构与程序关系主要体现在程序对数据的处理和操作方式上。

### 5.3.1 Java 实体类设计

- 系统将使用 Java 实体类来表示逻辑结构中的数据模型，如学生、课程和成绩等。

### 5.3.2 数据访问对象（DAO）设计

- 系统将设计数据访问对象（DAO）来实现程序与数据库之间的数据交互，包括数据的读取、写入和更新等操作。

### 5.3.3 业务逻辑处理器设计

- 系统将设计业务逻辑处理器来实现程序对逻辑结构中数据的处理和计算，如成绩统计、课程选课逻辑等。

## 6. 系统出错处理设计

### 6.1 出错信息

- 系统出错时，需要提供清晰明了的出错信息，以帮助用户和管理员快速定位和解决问题。

#### 6.1.1 前端错误提示

- 前端页面应具备友好的错误提示机制，当用户操作出错时，能够直观地提示用户出错的



原因和解决方法。

#### 6.1.2 后端异常信息

– 后端服务应记录异常信息，包括异常类型、异常发生时间、异常原因等，以便开发人员进行调试和定位问题。

#### 6.2 补救措施

系统出错后，需要采取相应的补救措施来尽快恢复系统的正常运行状态。

##### 6.2.1 数据回滚

– 在出现严重错误时，系统应能够自动回滚到出错前的状态，避免数据错误导致系统数据异常。

##### 6.2.2 异常处理

– 对于可预见的异常情况，系统应设计相应的异常处理机制，采取合适的措施进行处理，如重试、数据修复等。

#### 6.3 系统维护设计

– 系统的维护设计主要包括系统的更新、优化和监控等方面。

##### 6.3.1 系统更新

– 系统需要定期进行更新，修复已知的 BUG，并增加新功能和优化性能。

##### 6.3.2 系统优化

– 定期对系统进行性能优化，包括数据库索引优化、代码优化等，提升系统的运行效率和响应速度。

##### 6.3.3 系统监控

– 引入系统监控工具，对系统的运行状态进行实时监控，及时发现和处理异常情况，保证系统的稳定性和可用性。

#### 7. 总结

本文档提供了对浙江万里学院教务管理学生子系统的概要设计，包括功能模块、数据库设计、系统架构设计等方面的内容。该设计将为系统的详细设计和开发提供指导，确保系统能够满足用户的需求并具备良好的可维护性和扩展性。