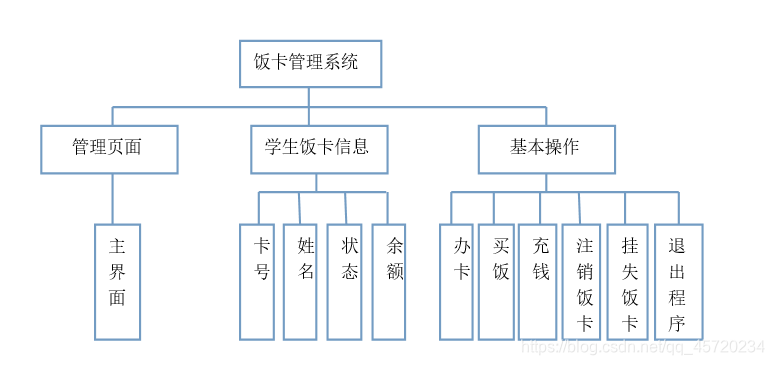
**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于jQuery的学生饭卡管理系统的分析与设计 | | |
| 课 程 | 软件工程 | | |
| 学 院 | 信息工程学院 | | |
| 专 业 | 软件工程 | 年级 | 2023级 |
| 学生姓名 | 许琪俊 | 学号 | 112322100003 |

|  |  |  |  |
| --- | --- | --- | --- |
| **评分项** | **评分标准** | **满分** | **得分** |
| 文档整体 | 文档内容详实、规范，美观大方，内容: 可行性分析、需求分析、概要设计、详细设计。 | 20 |  |
| 用例图及规约 | 用例图完整，准确，能够完全体现需求 | 10 |  |
| 顺序图与活动图 | 顺序图能够完全描述用例的设计思路和业务流程，活动图能够描述部分算法的流程。 | 10 |  |
| 类图 | 类图能够完整，准确反映业务的需数据的组织结构 | 10 |  |
| 数据库 | 数据库表的设计来自于类图，合理 | 10 |  |
| 界面设计 | 界面设计美观，清晰，合理，能够完全反映用例图的设计思路 | 10 |  |
| API接口设计 | 为每个界面设计合理的API接口，以便进行前后端分离开发 | 15 |  |
| 内容一致性 | 用例图，类图、数据库、界面相互印证，相互依赖，环环相扣 | 15 |  |
| **得分合计** | | |  |

2024 年 12 月 19 日

# 2. 系统总体结构

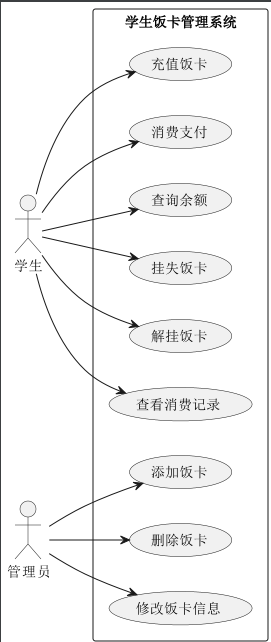




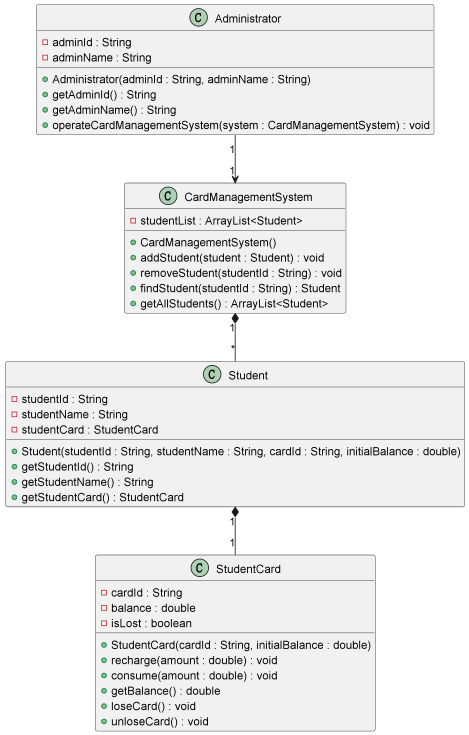




# 3. 用例图设计



# 4. 类图设计



# 5. 数据库设计

## STUDENTS表（学生表）

​

饭卡信息表（card\_info）



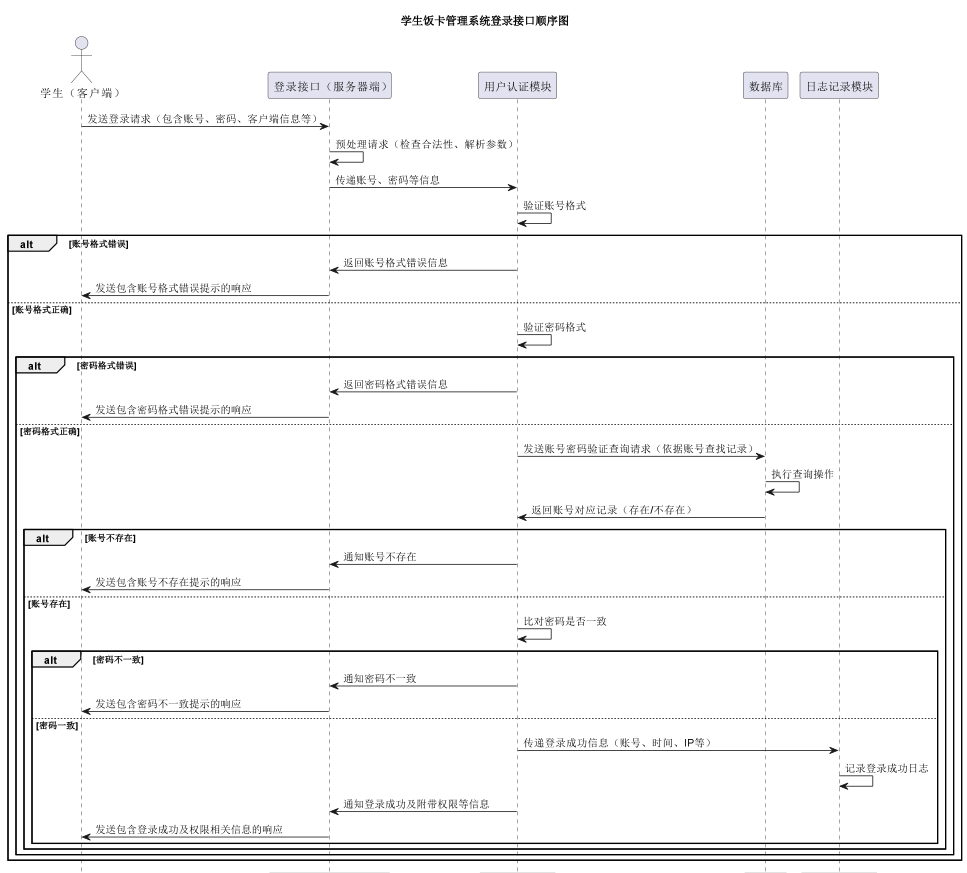
充值记录表（recharge\_record）：



消费记录表（consumption\_record）：



# 6. 用例及界面详细设计

1. **登录接口（/api/v1/auth/login）**
2. 
3. 

**功能描述**：用于验证用户身份信息，返回 JWT 令牌以便后续接口请求的授权。

**请求方法**：POST

**请求参数**：

username（字符串类型，必填）：用户的账号，对于学生为学号，管理员为特定的管理账号。

password（字符串类型，必填）：用户密码，需进行加密传输。

**响应数据**：

成功登录：

{

"code": 200,

"message": "登录成功",

"data": {

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV\_adQssw5c",

"user": {

"id": 12345,

"username": "2021001",

"role": "student",

"name": "张三"

}

}

}

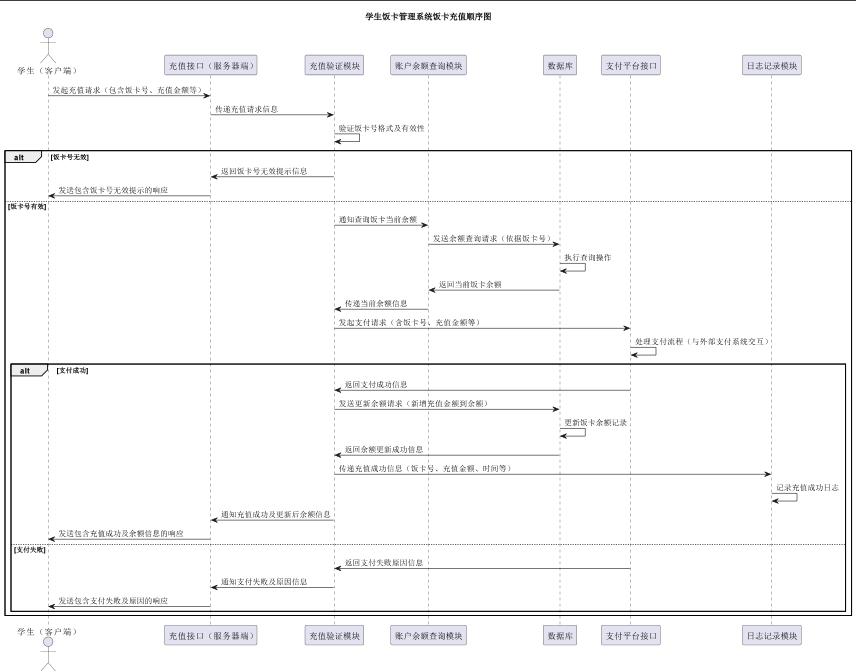
登录失败：

{

"code": 401,

"message": "用户名或密码错误"

**（二）饭卡管理接口**

1. **饭卡充值接口（/api/v1/card/recharge）**
2. 
3. 

**功能描述**：为指定饭卡进行充值操作，更新饭卡余额并记录充值信息。

**请求方法**：POST

**请求参数**：

card\_id（字符串类型，必填）：要充值的饭卡编号，需确保该饭卡存在且状态正常。

amount（数值类型，必填）：充值金额，需大于 0。

operator（字符串类型，必填）：操作员信息，记录进行充值操作的人员姓名或工号，用于操作日志记录。

**响应数据**：

充值成功：

{

"code": 200,

"message": "充值成功",

"data": {

"card\_id": "FK20241213001",

"balance": 100.00 // 假设充值前余额为 0，充值了 100 元后的余额

}

}

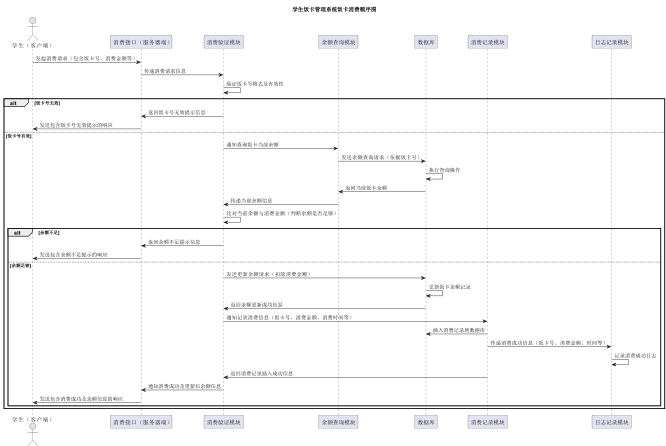
充值失败：

{

"code": 400,

"message": "饭卡不存在"

}

1. **饭卡消费接口（/api/v1/card/consume）**
2. 
3. 

**功能描述**：处理饭卡消费操作，验证饭卡余额并扣除消费金额，记录消费信息。

**请求方法**：POST

**请求参数**：

card\_id（字符串类型，必填）：消费的饭卡编号，需确保该饭卡存在且状态正常。

amount（数值类型，必填）：消费金额，需大于 0 且小于等于饭卡当前余额。

consumption\_location（字符串类型，必填）：消费地点信息，如食堂窗口编号或具体位置描述，用于消费记录和统计分析。

**响应数据**：

消费成功：

{

"code": 200,

"message": "消费成功",

"data": {

"card\_id": "FK20241213001",

"balance": 50.00 // 假设消费前余额为 100 元，消费了 50 元后的余额

}

}

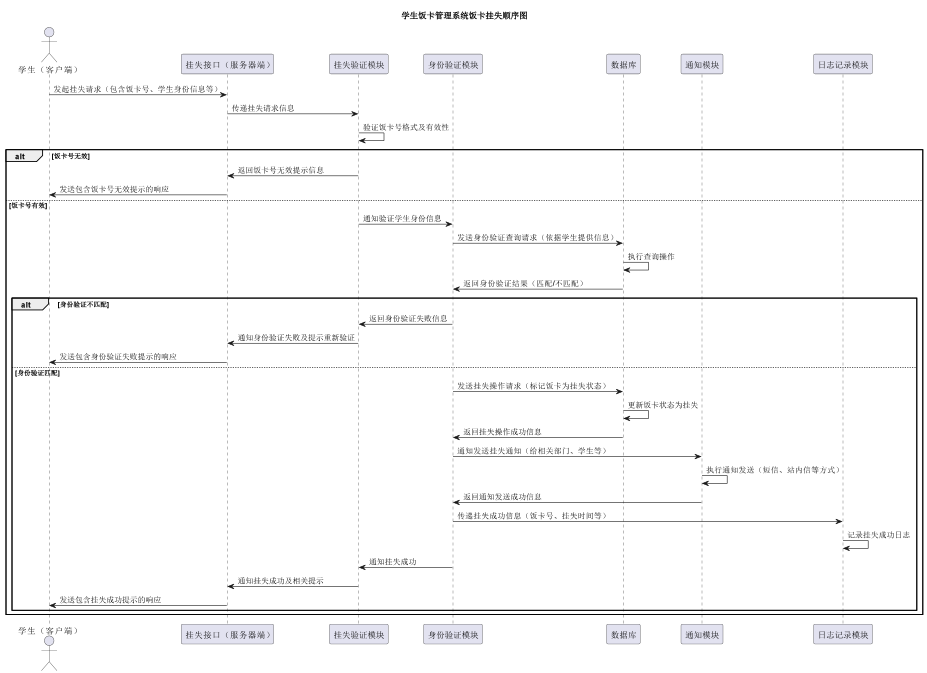
消费失败：

{

"code": 400,

"message": "饭卡余额不足"

}

1. **饭卡挂失接口（/api/v1/card/lost）**
2. 
3. 

**功能描述**：将指定饭卡设置为挂失状态，防止他人冒用。

**请求方法**：POST

**请求参数**：

card\_id（字符串类型，必填）：要挂失的饭卡编号，需确保该饭卡存在且未挂失。

operator（字符串类型，必填）：操作员信息，记录进行挂失操作的人员姓名或工号。

**响应数据**：

挂失成功：

{

"code": 200,

"message": "挂失成功"

}

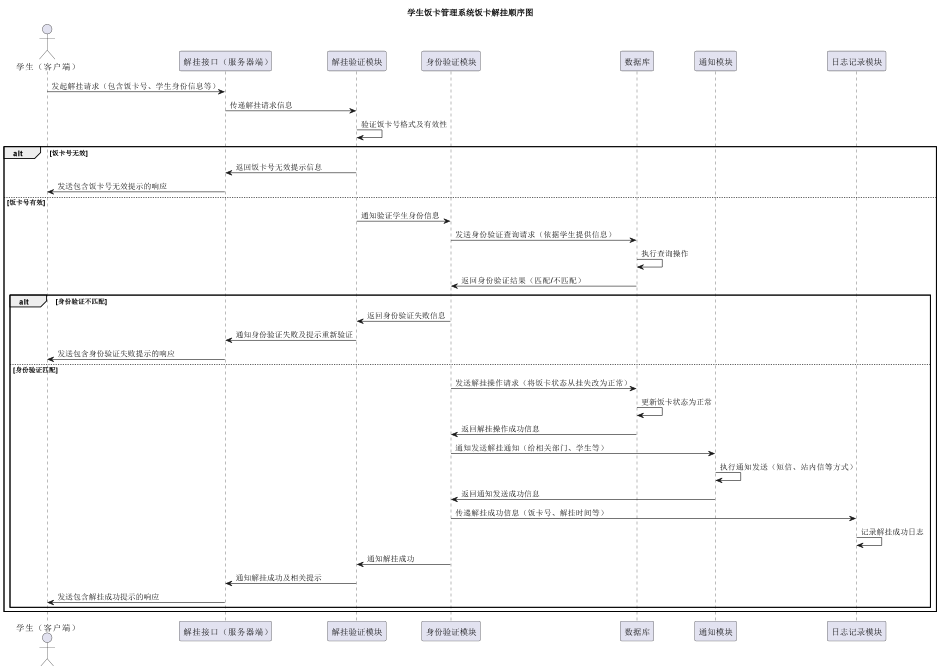
挂失失败：

{

"code": 400,

"message": "饭卡已挂失"

}

1. **饭卡解挂接口（/api/v1/card/unlost）**
2. 
3. 

**功能描述**：将已挂失的饭卡恢复为正常状态，允许正常使用。

**请求方法**：POST

**请求参数**：

card\_id（字符串类型，必填）：要解挂的饭卡编号，需确保该饭卡存在且已挂失。

operator（字符串类型，必填）：操作员信息，记录进行解挂操作的人员姓名或工号。

**响应数据**：

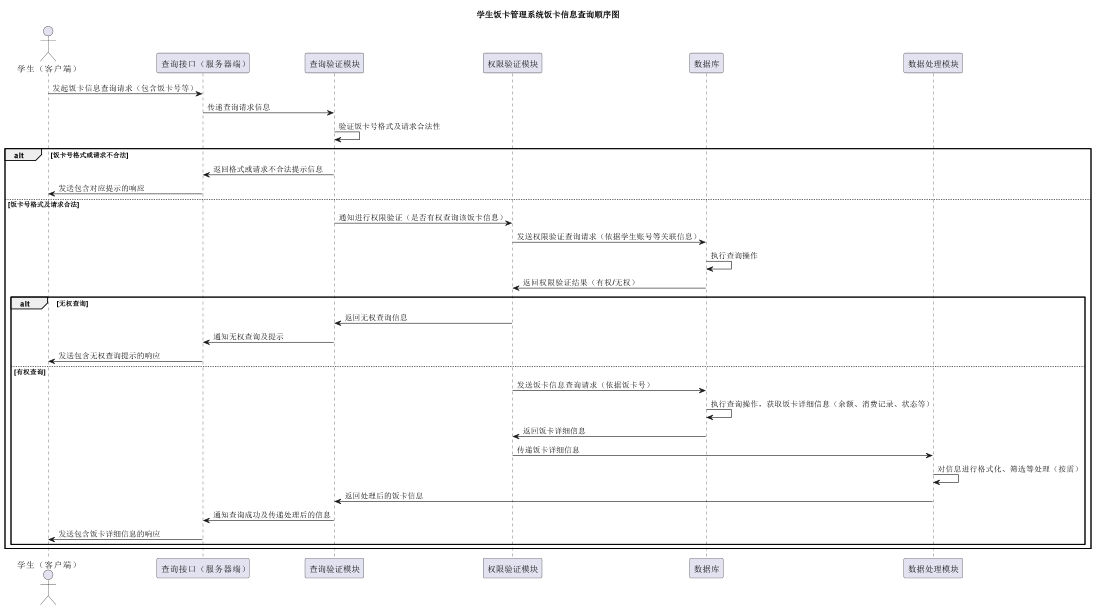
解挂成功：

`json { "code": 200, "message": "解挂成功" } `

解挂失败：

``json { "code": 400, "message": "饭卡未挂失" } ```

**（三）查询接口**

1. **饭卡信息查询接口（/api/v1/card/info）**
2. 
3. 

**功能描述**：根据饭卡编号或学生学号查询饭卡的详细信息，包括饭卡余额、状态、开户时间等。

**请求方法**：GET

**请求参数**：

card\_id（字符串类型，可选）：饭卡编号，可根据此编号查询特定饭卡信息。

student\_id（字符串类型，可选）：学生学号，可根据此学号查询该学生对应的饭卡信息。至少提供一个参数，若两个参数都提供则优先使用 card\_id 进行查询。

**响应数据**：

查询成功：

{

"code": 200,

"message": "查询成功",

"data": {

"card\_id": "FK20241213001",

"student\_id": "2024001",

"balance": 100.00,

"status": "正常",

"open\_date": "2024-12-13T09:00:00Z"

}

}

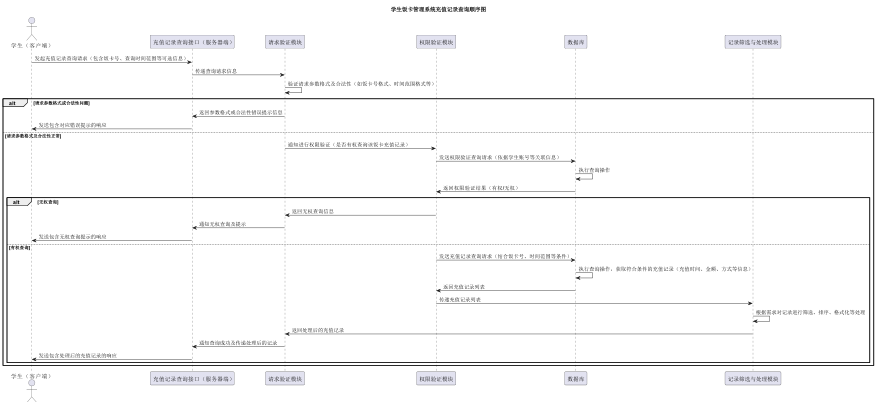
查询失败：

{

"code": 400,

"message": "未找到对应的饭卡信息"

}

1. **充值记录查询接口（/api/v1/card/recharge/history）**
2. 
3. 

**功能描述**：查询指定饭卡的充值历史记录，可根据时间范围等条件进行筛选。

**请求方法**：GET

**请求参数**：

card\_id（字符串类型，必填）：要查询充值记录的饭卡编号。

start\_date（日期时间类型，可选）：查询开始时间，用于筛选指定时间范围后的充值记录。

end\_date（日期时间类型，可选）：查询结束时间，用于筛选指定时间范围前的充值记录。若只提供 start\_date，则查询该时间之后的所有充值记录；若只提供 end\_date，则查询该时间之前的所有充值记录；若两个时间都提供，则查询指定时间范围内的充值记录。

**响应数据**：

查询成功：

{

"code": 200,

"message": "查询成功",

"data": [

{

"recharge\_id": "RC20241213001",

"card\_id": "FK20241213001",

"amount": 100.00,

"recharge\_date": "2024-12-13T10:00:00Z",

"operator": "管理员001"

},

{

"recharge\_id": "RC20241214002",

"card\_id": "FK20241213001",

"amount": 50.00,

"recharge\_date": "2024-12-14T15:30:00Z",

"operator": "管理员002"

}

]

}

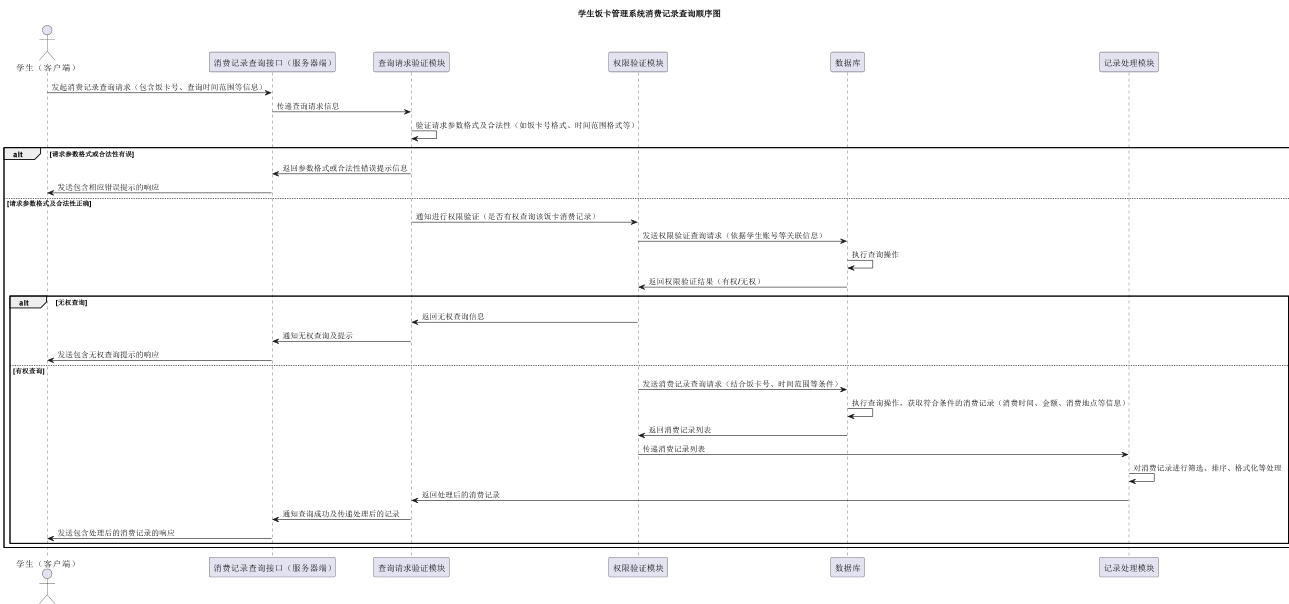
查询失败：

{

"code": 400,

"message": "未找到对应的充值记录"

}

1. **消费记录查询接口（/api/v1/card/consumption/history）**
2. 
3. 

**功能描述**：查询指定饭卡的消费历史记录，可根据时间范围、消费地点等条件进行筛选。

**请求方法**：GET

**请求参数**：

card\_id（字符串类型，必填）：要查询消费记录的饭卡编号。

start\_date（日期时间类型，可选）：查询开始时间，用于筛选指定时间范围后的消费记录。

end\_date（日期时间类型，可选）：查询结束时间，用于筛选指定时间范围前的消费记录。

consumption\_location（字符串类型，可选）：消费地点信息，用于筛选特定地点的消费记录。若同时提供时间范围和消费地点参数，则查询同时满足条件的消费记录；若只提供部分参数，则根据提供的参数进行相应筛选。

**响应数据**：

查询成功：

{

"code": 200,

"message": "查询成功",

"data": [

{

"consumption\_id": "CN20241213001",

"card\_id": "FK20241213001",

"amount": 30.00,

"consumption\_date": "2024-12-13T11:00:00Z",

"consumption\_location": "食堂一楼窗口 1"

},

{

"consumption\_id": "CN20241214002",

"card\_id": "FK20241213001",

"amount": 20.00,

"consumption\_date": "2024-12-14T16:00:00Z",

"consumption\_location": "食堂二楼窗口 5"

}

]

}

查询失败：

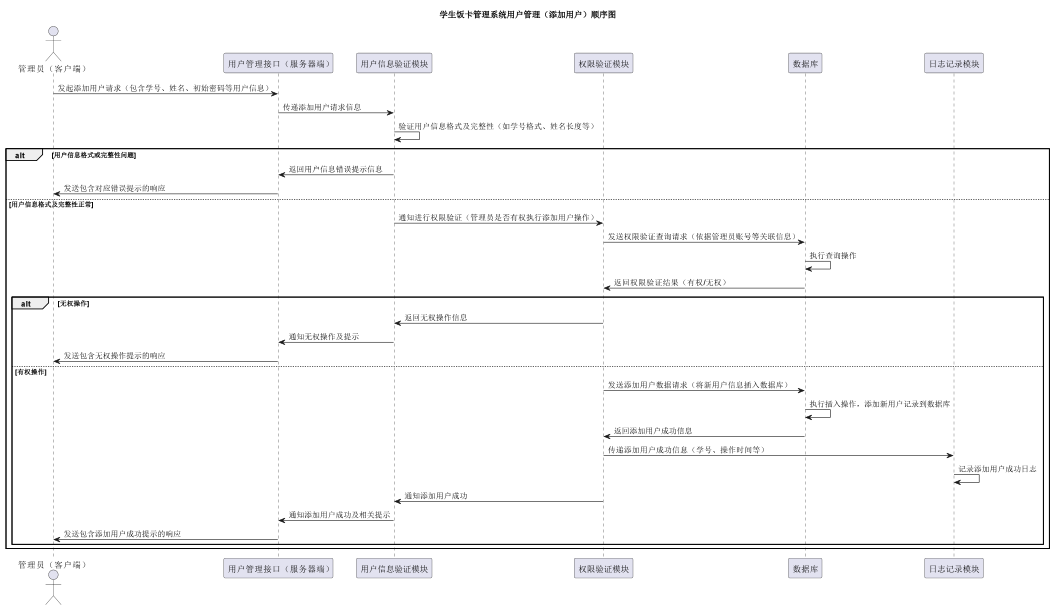
{

"code": 400,

"message": "未找到对应的消费记录"

}

**（四）系统管理接口（仅管理员可访问）**

1. **用户管理接口（/api/v1/admin/user）**
2. 
3. 

**功能描述**：对系统用户（学生和管理员）进行管理操作，包括创建、查询、修改和删除用户信息。

**请求方法**：根据具体操作而定（GET/POST/PUT/DELETE）

**请求参数**：

user\_id（字符串类型，可选）：用户编号，在查询、修改和删除操作时使用，用于指定特定用户。

username（字符串类型，可选）：用户名（学号或管理员账号），在查询操作时可作为筛选条件。

password（字符串类型，可选）：密码，在创建或修改用户密码时使用。

role（字符串类型，可选）：角色，取值为 "student" 或 "admin"，在创建或查询特定角色用户时使用。

其他用户信息字段（如学生的姓名、班级、性别，管理员的姓名、权限等）根据具体操作和需求而定，在创建或修改用户信息时使用。

**响应数据**：

操作成功：

code（整数类型）：200（查询操作）或 201（创建操作）或 204（删除操作），表示请求成功。对于修改操作，也可使用 200 表示成功。

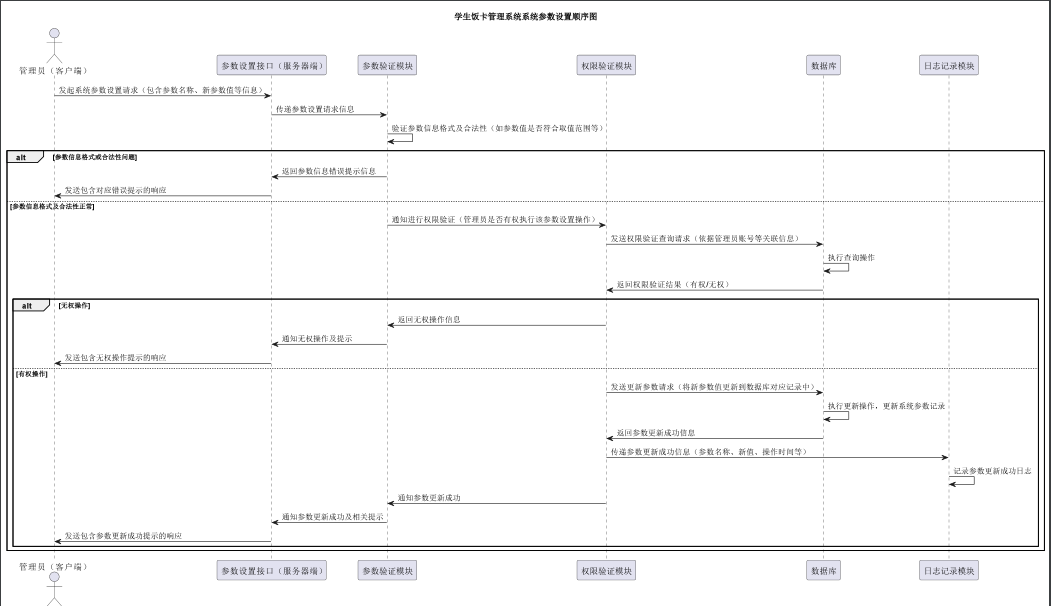
message（字符串类型）：相应的成功消息，如 "用户查询成功"、"用户创建成功"、"用户删除成功" 或 "用户信息修改成功"。

data（对象类型或数组类型）：根据操作而定，查询操作返回符合条件的用户信息数组或单个用户对象；创建操作可返回创建成功的用户信息对象；修改和删除操作可根据需求返回相应的提示信息或空数据。

操作失败：

code（整数类型）：400（请求参数错误）或 401（未授权）或 403（禁止访问）或 404（未找到资源）或 409（冲突，如用户名已存在）等，表示请求失败。

message（字符串类型）：错误原因，如 "参数错误"、"未授权访问"、"禁止执行该操作"、"未找到对应的用户" 或 "用户名已存在，创建失败"。

1. **系统参数设置接口（/api/v1/admin/system/parameter）**
2. 
3. 

**功能描述**：设置系统相关参数，如饭卡有效期、消费限额、密码规则等。

**请求方法**：PUT

**请求参数**：

param\_name（字符串类型，必填）：参数名称，如 "card\_validity\_period"（饭卡有效期）、"consumption\_limit"（消费限额）、"password\_rule"（密码规则）等。

param\_value（字符串类型或数值类型，必填）：参数值，根据参数名称的不同，值的类型和格式可能不同。例如，饭卡有效期可能是一个数字（表示天数），消费限额是一个数值，密码规则可能是一个字符串描述（如 "至少 6 位字符，包含数字和字母"）。

**响应数据**：

设置成功：

code（整数类型）：200，表示请求成功。

message（字符串类型）："系统参数设置成功"。

设置失败：

code（整数类型）：400，表示请求参数错误或其他业务逻辑错误。

message（字符串类型）：错误原因，如 "参数名称错误" 或 "参数值无效"。