

# 用例交互图

本文档描述了中山大学医生预约系统（sysu-doctor）的用例交互图，展示了系统中不同参与者（Actor）与各种用例（Use Case）之间的交互关系。

## ✧ 1. 系统参与者（Actors）

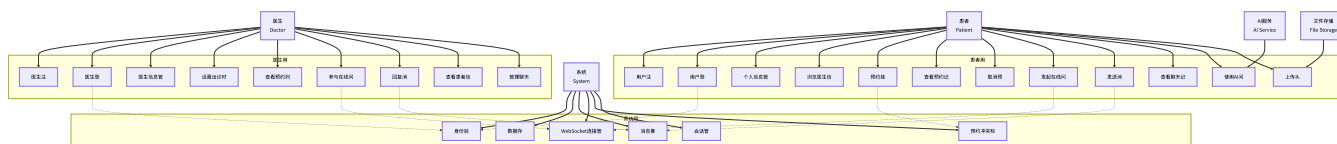
### 1.1 主要参与者

- 患者（Patient）：使用系统进行预约挂号、在线问诊、AI问诊等服务的用户
- 医生（Doctor）：提供医疗服务、管理个人信息、参与在线问诊的医疗专业人员
- 系统（System）：处理业务逻辑、数据存储、消息推送等后台服务

### 1.2 外部系统

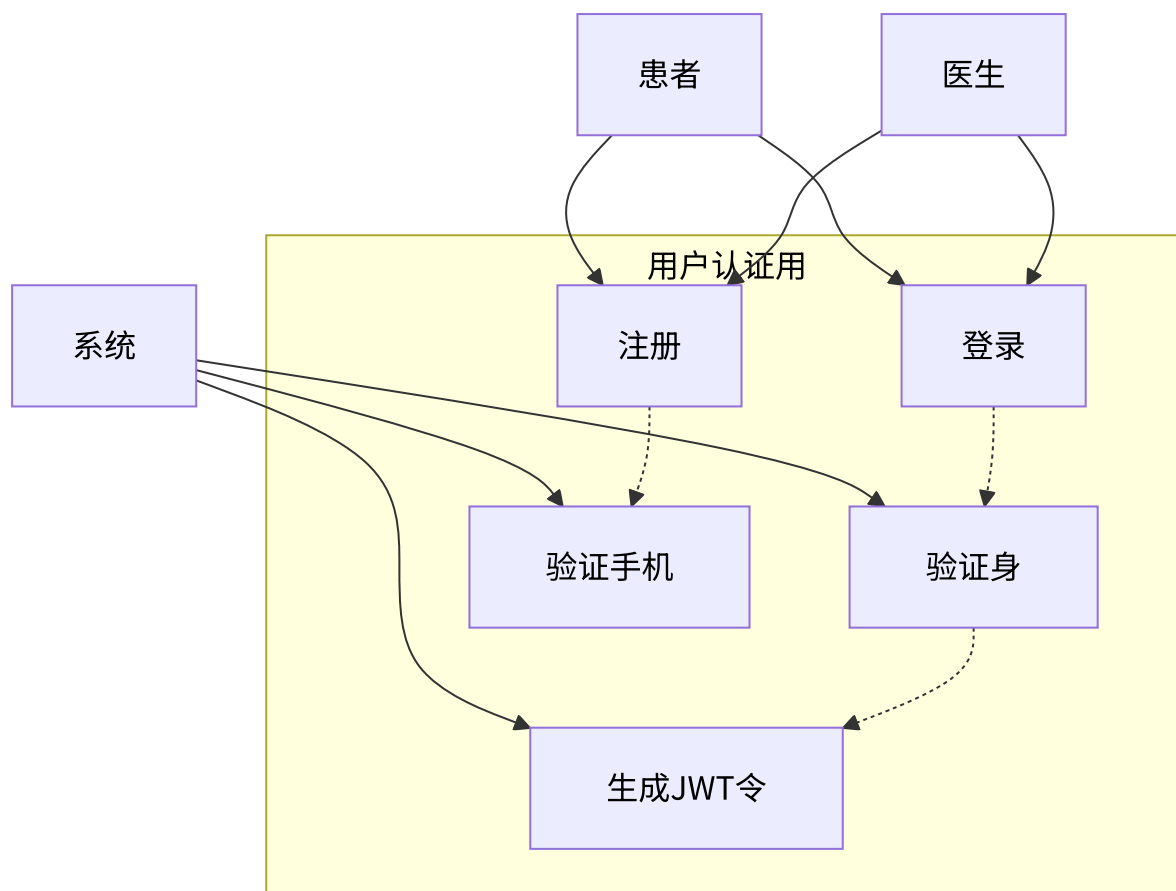
- AI服务（AI Service）：提供智能问诊服务的外部AI接口
- 文件存储服务（File Storage）：处理头像上传和图片存储的服务

## ✧ 2. 整体系统用例图



## \* 3. 详细用例交互图

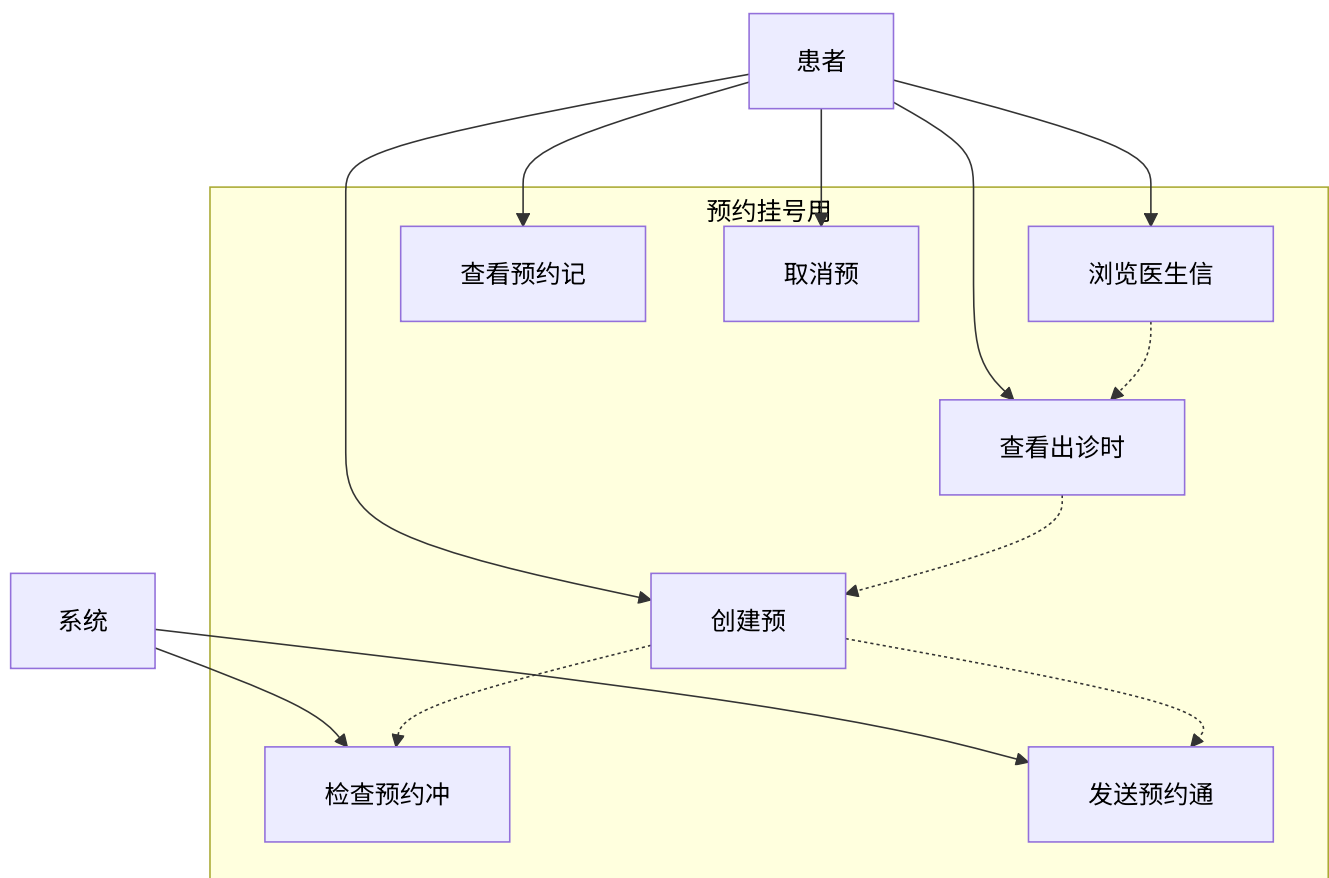
### 3.1 用户认证模块用例图



用例说明：

- 注册：用户提供手机号、密码、姓名进行系统注册
- 登录：用户使用手机号和密码登录系统
- 验证身份：系统验证用户提供的登录凭据
- 生成JWT令牌：为通过验证的用户生成访问令牌
- 验证手机号：检查手机号格式和唯一性

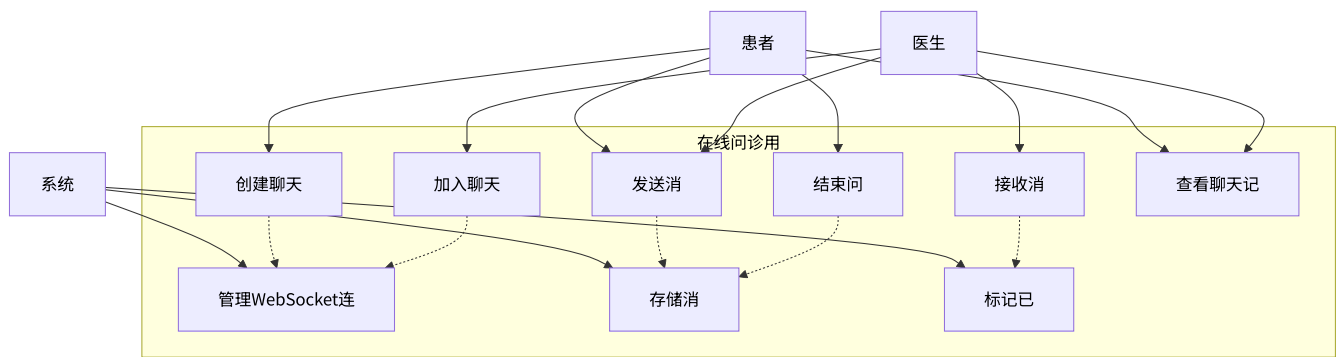
## 3.2 预约挂号模块用例图



用例说明：

- 浏览医生信息：患者查看可预约的医生列表和详细信息
- 查看出诊时间：患者查看医生的可预约时间段
- 创建预约：患者选择医生和时间段创建预约记录
- 检查预约冲突：系统验证是否存在重复预约
- 查看预约记录：患者查看自己的预约历史
- 取消预约：患者主动取消已创建的预约
- 发送预约通知：系统向相关方发送预约状态变更通知

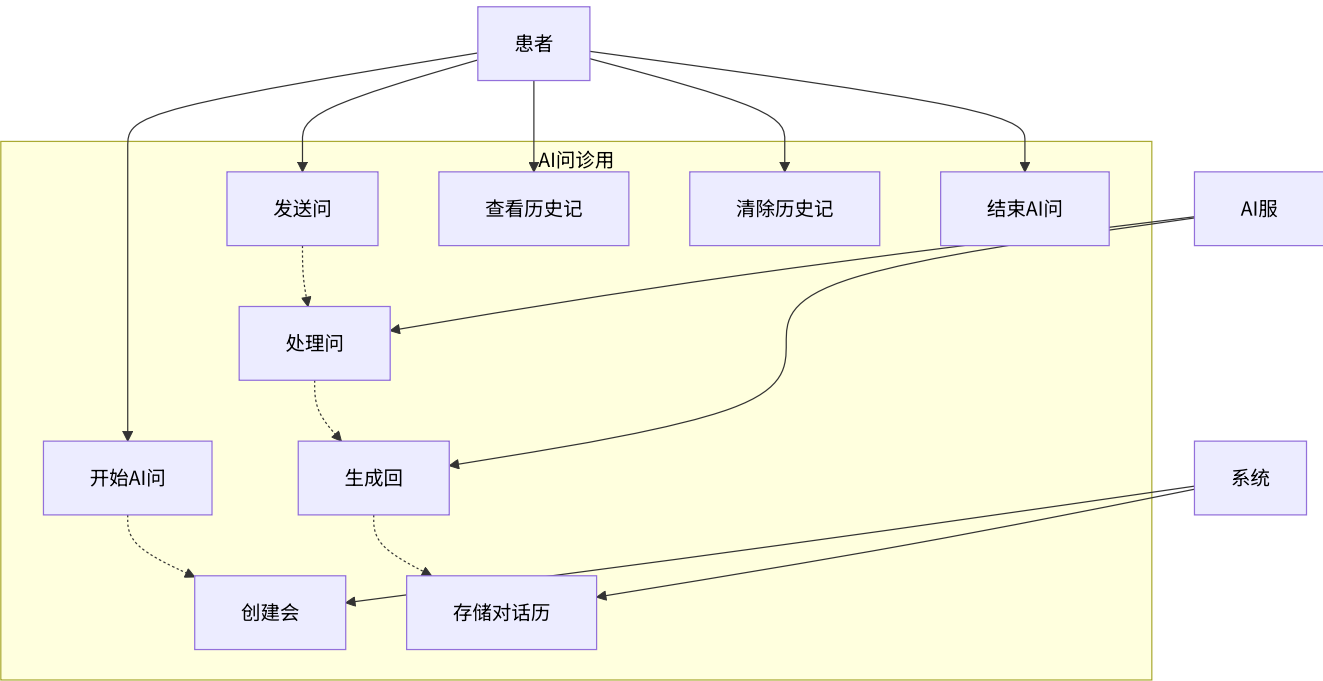
### 3.3 在线问诊模块用例图



用例说明：

- 创建聊天室：患者选择医生创建问诊聊天室
- 加入聊天室：医生接受问诊请求加入聊天室
- 发送消息：用户发送文本或图片消息
- 接收消息：用户接收来自对方的消息
- 管理WebSocket连接：系统管理实时通信连接
- 存储消息：系统将消息保存到数据库
- 标记已读：系统标记消息的已读状态
- 结束问诊：结束问诊并清理相关数据
- 查看聊天记录：查看历史问诊记录

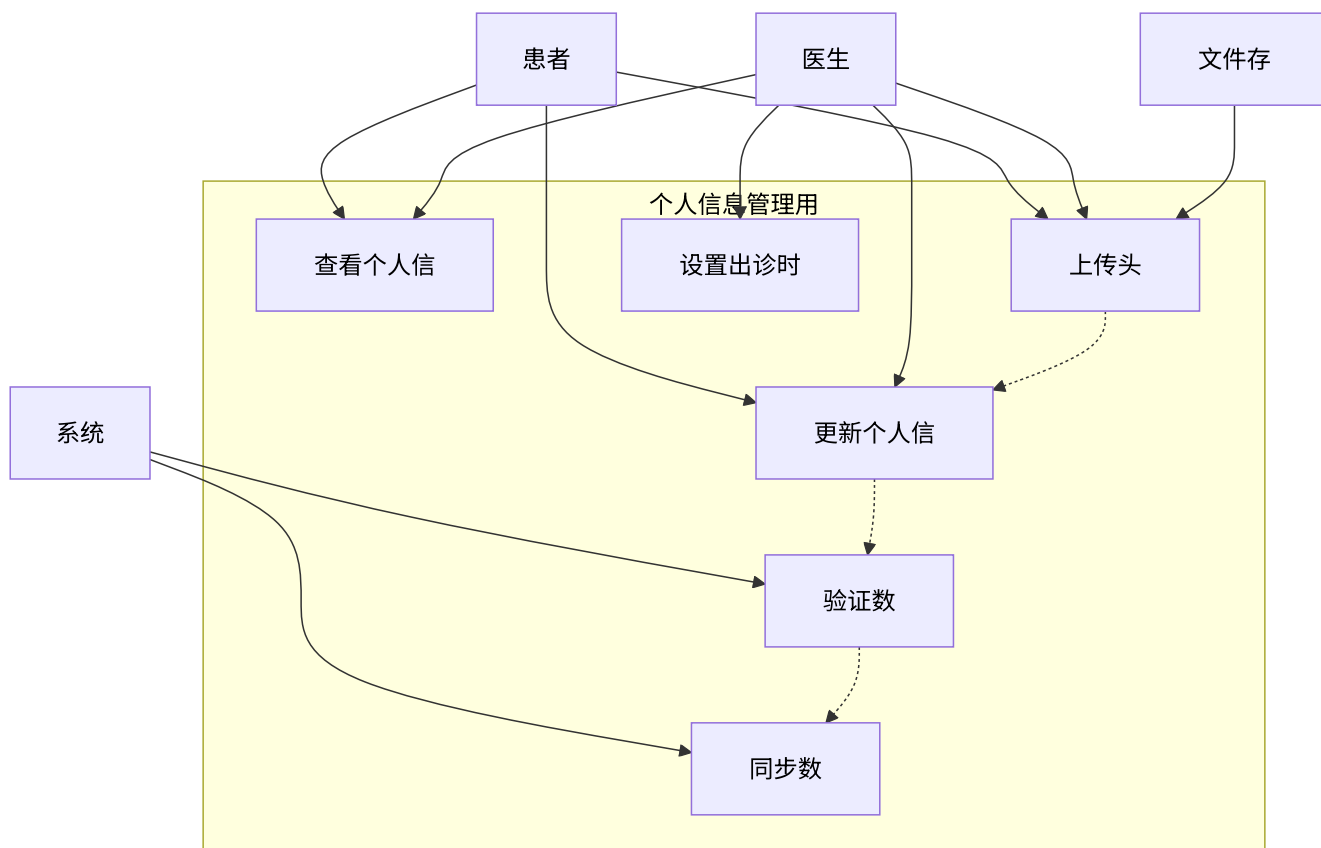
### 3.4 AI问诊模块用例图



用例说明：

- 开始AI问诊：患者启动AI问诊服务
- 创建会话：系统为患者创建AI问诊会话
- 发送问题：患者向AI发送健康相关问题
- 处理问题：AI服务处理患者的问题
- 生成回答：AI基于医学知识生成回答
- 存储对话历史：系统保存问答记录
- 查看历史记录：患者查看AI问诊历史
- 清除历史记录：患者清除AI问诊历史
- 结束AI问诊：结束AI问诊会话

### 3.5 个人信息管理模块用例图



用例说明：

- 查看个人信息：用户查看自己的个人资料
- 更新个人信息：用户修改个人资料信息
- 上传头像：用户上传或更换头像图片
- 验证数据：系统验证用户输入数据的有效性
- 设置出诊时间：医生配置自己的出诊时间安排
- 同步数据：系统在不同表之间同步用户数据

## ✧ 4. 用例关系说明

### 4.1 包含关系（Include）

- 登录用例 包含 身份验证用例

- 注册用例 包含 手机号验证用例
- 创建预约用例 包含 预约冲突检查用例
- 发送消息用例 包含 消息存储用例

## 4.2 扩展关系 (Extend)

- 消息推送用例 扩展 发送消息用例
- 预约通知用例 扩展 创建预约用例
- 文件上传用例 扩展 更新个人信息用例

## 4.3 泛化关系 (Generalization)

- 患者注册和医生注册 泛化 为用户注册
- 患者登录和医生登录 泛化 为用户登录
- 患者信息管理和医生信息管理 泛化 为个人信息管理

# \* 5. 系统边界和约束

---

## 5.1 系统边界

- 内部系统：用户认证、预约管理、消息处理、数据存储
- 外部系统：AI服务接口、文件存储服务、WebSocket服务

## 5.2 主要约束

- 安全约束：所有操作需要JWT令牌验证
- 数据约束：手机号唯一性、预约时间不冲突
- 业务约束：医生只能在设定的出诊时间被预约
- 技术约束：实时通信依赖WebSocket连接

## 总结

---

本用例交互图展示了sysu-doctor系统中各个参与者与用例之间的完整交互关系。系统主要围绕患者和医生两类用户，提供了完整的医疗服务流程，包括：

- 1 用户管理：注册、登录、个人信息管理
- 2 预约服务：医生信息浏览、预约挂号、预约管理
- 3 问诊服务：在线实时问诊、AI智能问诊
- 4 辅助功能：文件上传、消息推送、数据同步

每个用例都有明确的参与者、前置条件、后置条件和异常处理机制，确保系统的完整性和可靠性。