

智慧医航

分工

学生姓名	子任务1：系统功能设计与数据库设计	子任务2：系统服务器端开发	子任务3：系统客户端设计	工作量占比
陈泓兆	想法讨论	ER 图绘制	病人界面，管理员界面，部分前后端连接，测试	28%
黄一轩	想法讨论，需求分析，数据流程分析	部分接口文档，数据流图绘制，接口敲定，在线图床搭建，服务器部署任务	登录界面和公用界面，病人全部界面，在线上传图片，将应用部署到服务器，部分前后端连接，测试	36%
周靖宇	想法讨论，需求分析，确定选题，数据流程分析，主题设计任务	后端代码全部内容，后端接口测试，数据库连接，数据库存储定义与实现后端架构设计与实现	前后端连接，测试	36%

一、需求分析

1功能设计

1.1用户相关功能

- 用户注册
- 用户分为两种：社区成员，非社区成员，社区成员输入学工号和密码，非社区成员则是输入用户名和密码。若信息合法，则注册成功，否则报相应的错误（社区成员用户名和学工号相同）
- 用户登录（通过用户名和密码）
- 如果是非社区用户，则需要绑定身份证号
- 删除账号，需要写对用户名，密码，学工号，身份证号等需要写“我确认删除用户”
- 输入自己的症状，获得推荐的挂号指南(通过文心一言api)
- 查询自己的化验结果
- 查询当前挂号是否有号以及排队情况，预测排到的时间
- 查询诊断证明
- 挂号
- 在线缴费

1.2医生相关功能

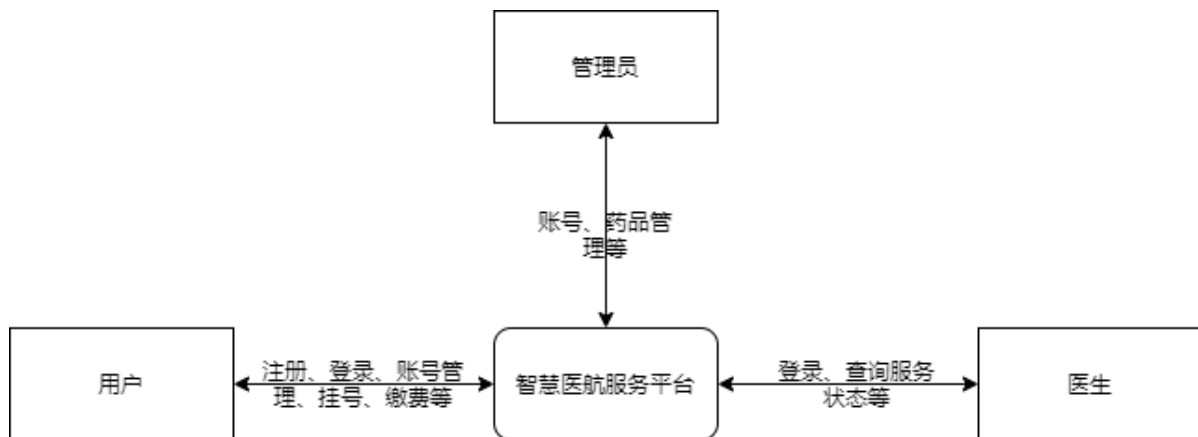
- 通过医生编号和密码进行登录
- 查询自己的排班状态
- 给病人开药
- 给病人设定化验订单
- 给病人下定诊断

1.3管理员相关功能

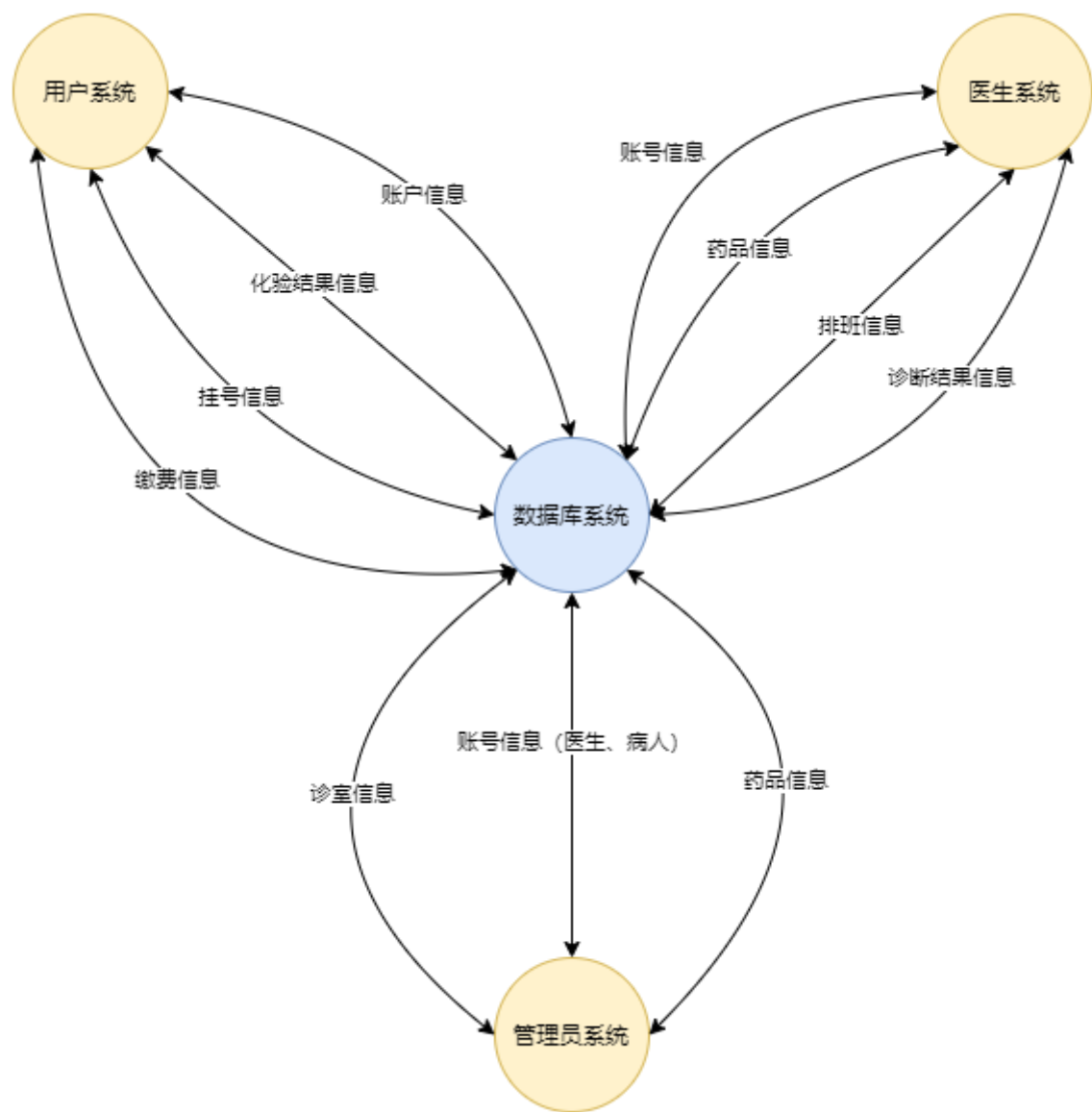
- 通过 admin 用户进行登录
- 查看药品剩余库存
- 添加新的医生账户
- 设定诊室是否可用

2.数据流图

2.1顶层简略数据流图

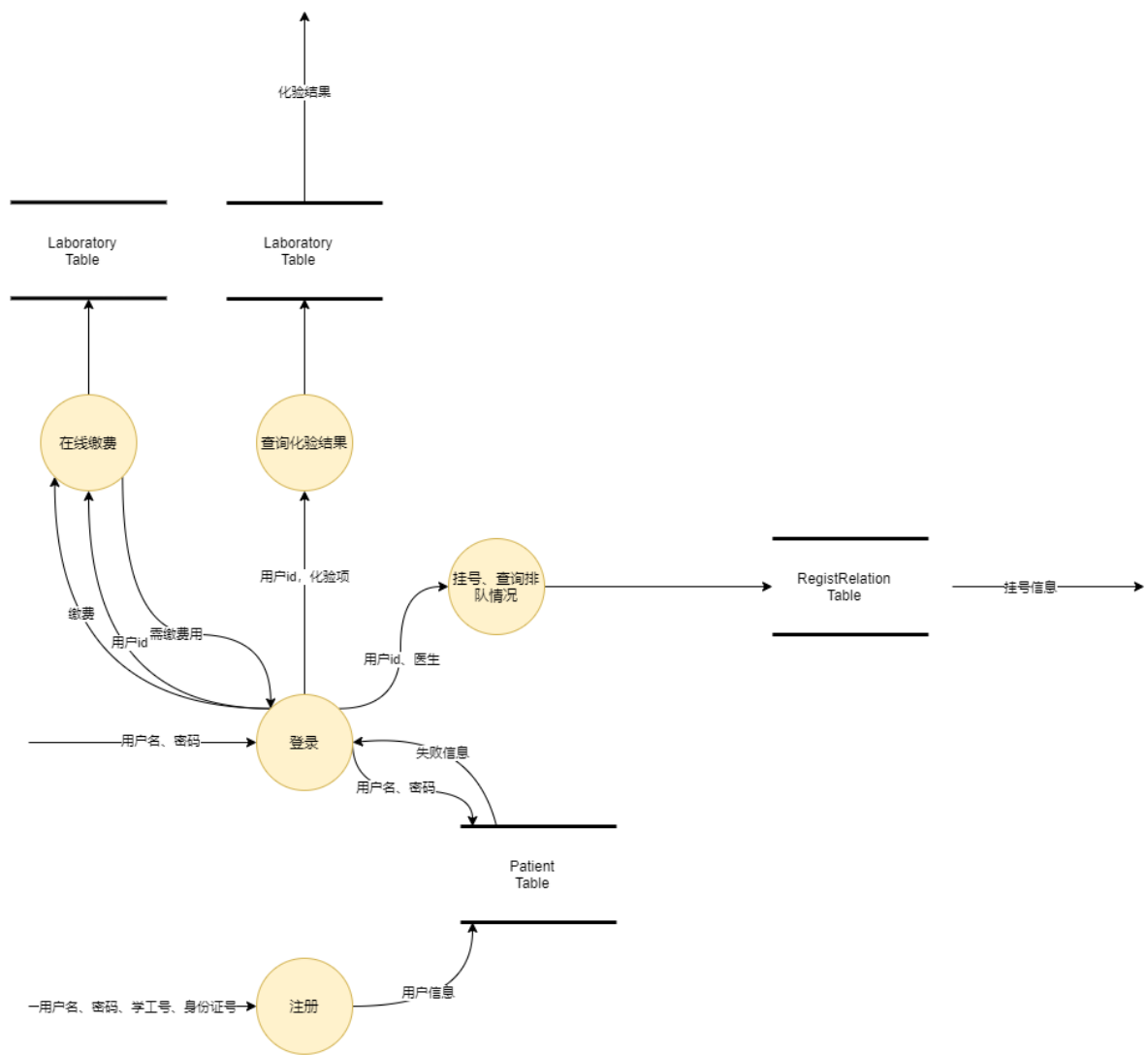


2.2零层数据流图

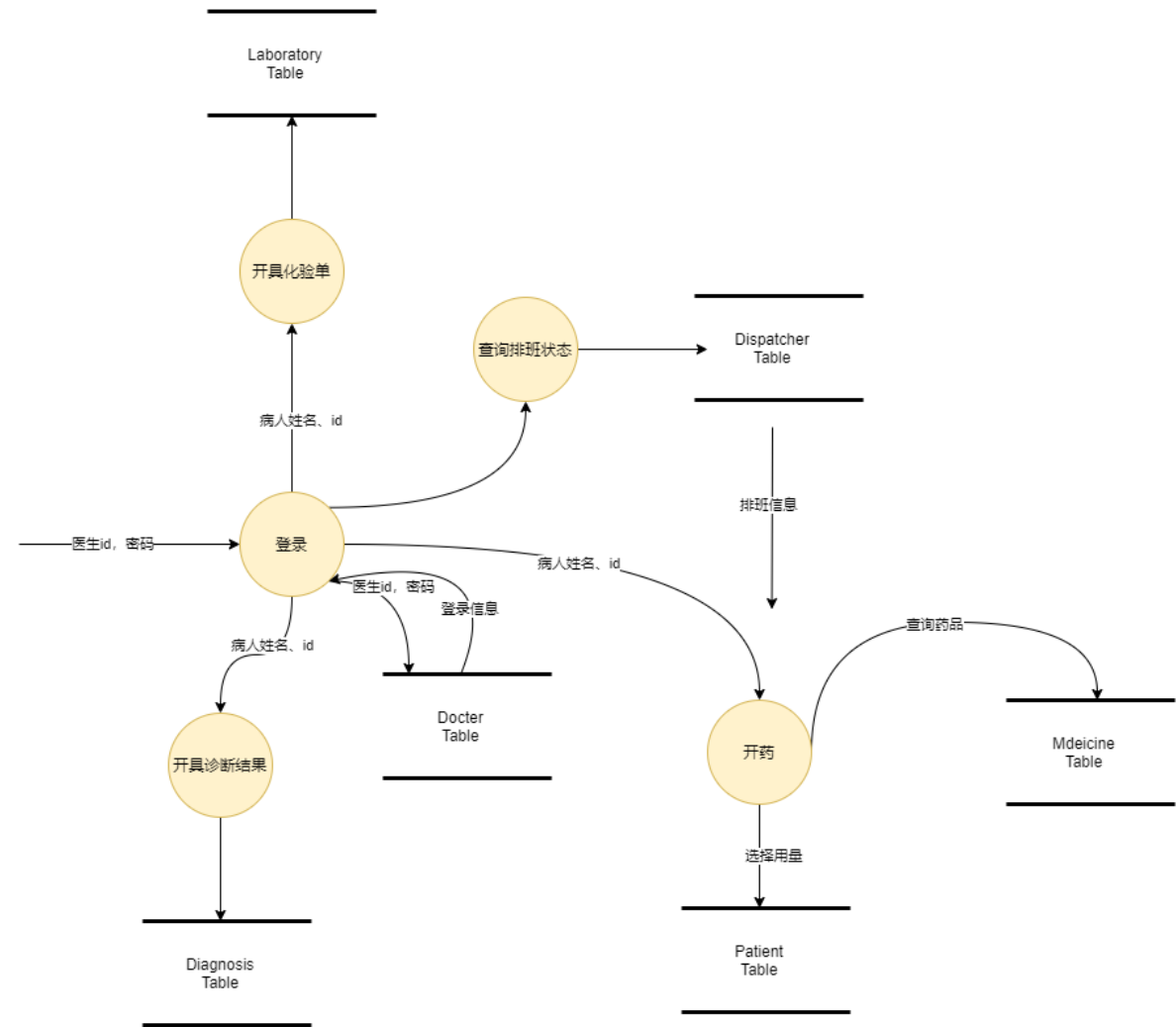


2.3一层数据流图

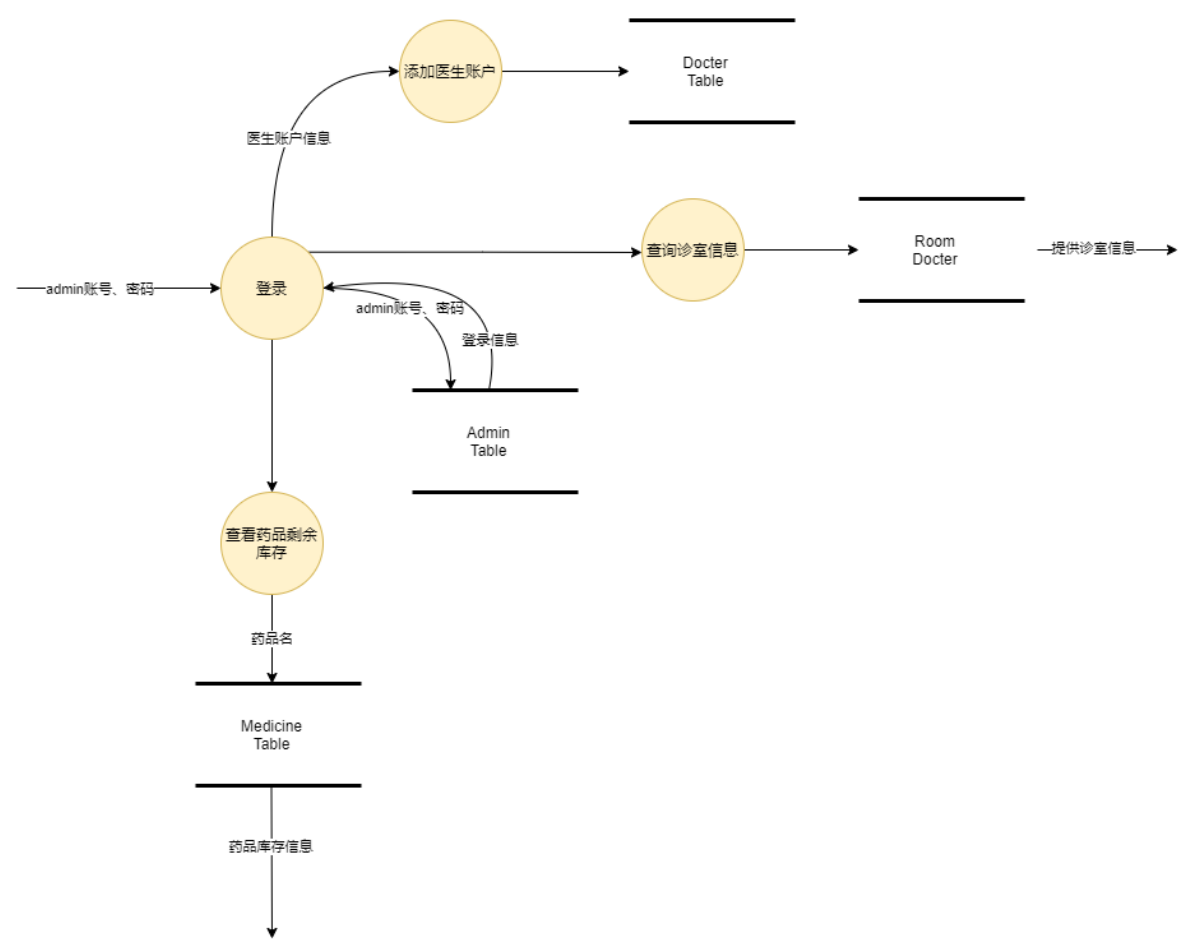
2.3.1用户系统数据流图



2.3.2医生系统数据流图



2.3.3管理员系统相关功能



3.数据元素表

3.0 User(用户表)

属性	中文	类型	备注
id	编号	VARCHAR(25)	NOT NULL，主码
username	用户名	VarCHAR(25)	NOT NULL
password	密码	VarCHAR(25)	NOT NULL
type	类型	VarCHAR(25)	NOT NULL

3.1 Patient(病人表)继承User

属性	中文	类型	备注
id	用户名	VARCHAR(25)	NOT NULL，外码
isComMem	是否为社区成员	BOOLEAN	NOT NULL
active	是否被删除	BOOLEAN	NOT NULL
idCard	身份证号	VarCHAR(25)	

3.2 Title(科室表)

属性	中文	类型	备注
id	编号	VARCHAR(25)	NOT NULL,主码
name	名称	VARCHAR(25)	NOT NULL

3.3 doctor(医生表)

属性	中文	类型	备注
id	医生编号	VARCHAR(25)	NOT NULL，外码继承User
active	是否活跃	BOOLEAN	NOT NULL
Tid	负责科室编号	VarCHAR(25)	NOT NULL

3.4 Drug(药品表)

属性	中文	类型	备注
id	药品编号	VARCHAR(25)	NOT NULL，主码
isBanned	是否被禁止	BOOLEAN	NOT NULL
price	药品价格	FLOAT	NOT NULL
Storage	存储量	FLOAT	NOT NULL
description	药品描述	TEXT	NOT NULL

3.5 Counter(订单表)

属性	中文	类型	备注
id	订单编号	VARCHAR(25)	NOT NULL, 主码
isPaid	是否已支付	BOOLEAN	NOT NULL
price	价格	FLOAT	NOT NULL
Pid	病人编号	VARCHAR(25)	NOT NULL，外码继承Patient
Did	医生编号	VARCHAR(25)	NOT NULL,外码继承DOCTOR
DATE	日期	DATE	NOT NULL
type	订单种类	VARCHAR(25)	NOT NULL

3.6 RegistRelation(登记表(挂号))

属性	中文	类型	备注
id	订单编号	VARCHAR(25)	NOT NULL, PRIMARY KEY, 外码继承Counter
ROOMID	诊室编号	VARCHAR(25)	NOT NULL
isFinished	是否被完成	BOOLEAN	NOT NULL

3.7 MedicinePurchase

属性	中文	类型	备注
id	订单编号	VARCHAR(25)	NOT NULL,主码
drugid	药品编码	VARCHAR(25)	NOT NULL, 外码继承Drug
amount	购买数量	FLOAT	NOT NULL

3.8 checkItems(检查项目)

属性	中文	类型	备注
id	检查项目编号	VARCHAR(25)	NOT NULL, 主码
price	检查项目价格	FLOAT	NOT NULL
description	检查项目描述	VARCHAR(255)	NOT NULL
MinResult	最小值	FLOAT	NOT NULL
MaxResult	最大值	FLOAT	NOT NULL

3.9 LaboratoryShee(化验订单)

属性	中文	类型	备注
id	订单编号	VARCHAR(25)	NOT NULL, 主码
checkName	化验名称	VARCHAR(25)	NOT NULL
beginTime	检查时间	DATETIME	NOT NULL
OutputTime	结果时间	DATETIME	NOT NULL
itemId	检查项目编号	VARCHAR(25)	NOT NULL,主码
result	检查结果	VARCHAR(255)	NOT NULL

3.10 ROOM(诊室表)

属性	中文	类型	备注
id	诊室编号	VARCHAR(25)	NOT NULL, 主码
isOccupied	是否被占用	BOOLEAN	NOT NULL
QueueLen	排队人数	int	NOT NULL

3.11 Dispatcher(医生排班调度器)

属性	中文	类型	备注
TimePeriod	时间段	VARCHAR(25)	NOT NULL, 主码
ROOMID	诊室编号	VARCHAR(25)	NOT NULL, 主码
doctorId	医生编号	VARCHAR(25)	
TitleId	负责的科室或活动名称编号	VARCHAR(25)	
DATE	日期（星期几）	VARCHAR(15)	NOT NULL

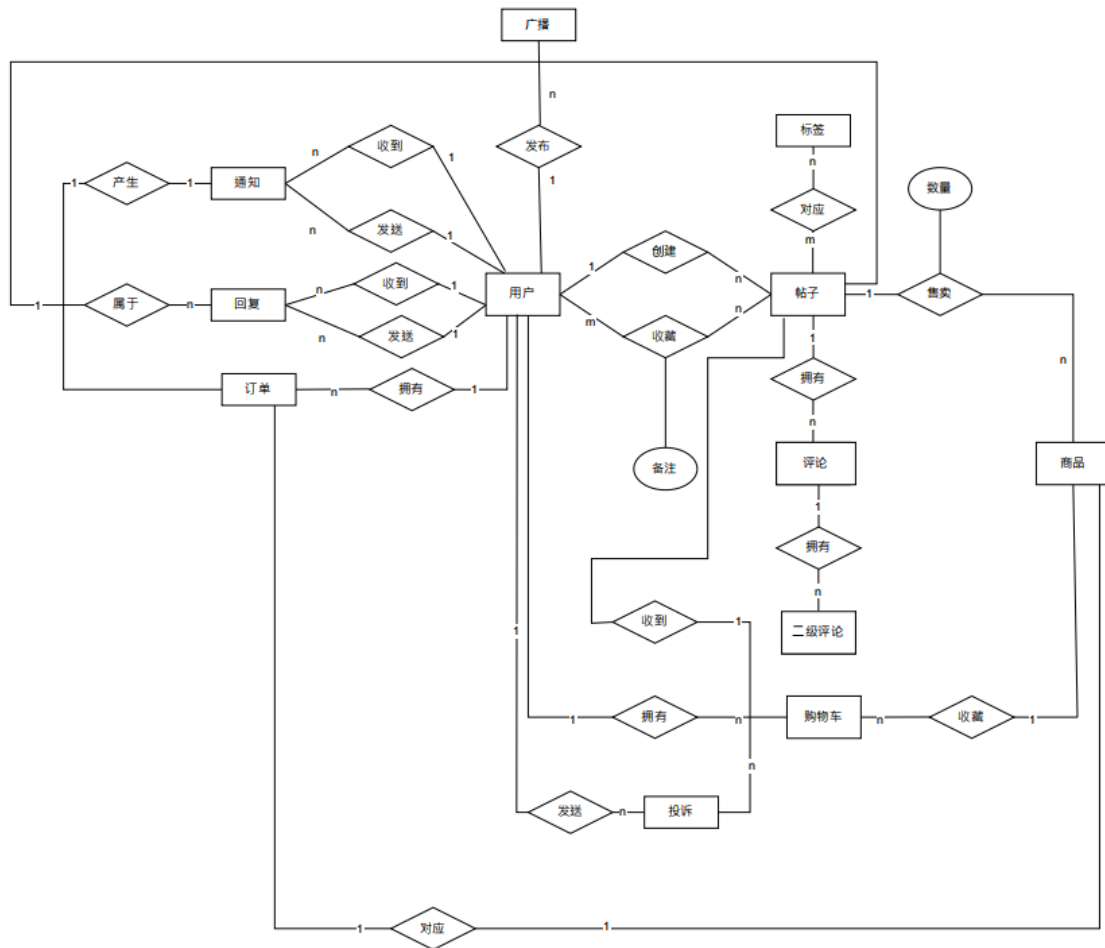
3.12 diagnosis(诊断订单)

属性	中文	类型	备注
id	订单编号	VARCHAR(25)	NOT NULL,主码
patientId	病人编号	VARCHAR(25)	NOT NULL
doctorId	医生编号	VARCHAR(25)	NOT NULL
diagnosis	诊断说明	TEXT	NOT NULL
time	诊断时间	DATETIME	NOT NULL

3.13 CheckCombine(常用检查组合)

属性	中文	属性	备注
id	编号	VARCHAR(25)	NOT NULL,主码
itemId	检查项编号	VARCHAR(25)	NOT NULL, 主码, 外码继承checkItems
description	名称	VARCHAR(255)	NOT NULL

二、数据库概念模式设计



三、数据库逻辑模式设计

关系模式

1. USER表:

- 主键: id
- 字段: username, password, type, AVATAR
- 外键: 无
- 关系模式: `USER(id, username, password, type, AVATAR)`

2. Patient表:

- 主键: id
- 字段: isComMem, idCard, active
- 外键: 无
- 关系模式: Patient(id, isComMem, idCard, active)

3. Titles表:

- 主键: id
- 字段: name
- 外键: 无
- 关系模式: Titles(id, name)

4. Doctor表:

- 主键: id
- 字段: Tid, active
- 外键: Tid (参考Titles表的id)
- 关系模式: Doctor(id, Tid, active)

5. Drug表:

- 主键: id
- 字段: price, Description, isBanned, Storage
- 外键: 无
- 关系模式: Drug(id, price, Description, isBanned, Storage)

6. Counter表:

- 主键: id
- 字段: Pid, Did, isPaid, price, DATE, type
- 外键: Pid (参考Patient表的id), Did (参考Doctor表的id)
- 关系模式: Counter(id, Pid, Did, isPaid, price, DATE, type)

7. RegistRelation表:

- 主键: id
- 字段: ROOMID, isFinished
- 外键: id (参考Counter表的id)
- 关系模式: RegistRelation(id, ROOMID, isFinished)

8. MedicinePurchase表:

- 主键: id, drugId
- 字段: amount
- 外键: drugId (参考Drug表的id), id (参考Counter表的id)
- 关系模式: MedicinePurchase(id, drugId, amount)

9. checkItems表:

- 主键: id
- 字段: price, description, MinResult, MaxResult
- 外键: 无
- 关系模式: checkItems(id, price, description, MinResult, MaxResult)

10. LaboratorySheet表:

- 主键: id, itemID
- 字段: checkName, beginTime, OutputTime, result
- 外键: itemID (参考checkItems表的id), id (参考Counter表的id)
- 关系模式: LaboratorySheet(id, itemID, checkName, beginTime, OutputTime, result)

11. ROOM表:

- 主键: id

- 字段: isOccupied, QueueLen
- 外键: 无
- 关系模式: `ROOM(id, isOccupied, QueueLen)`

12. Dispatcher表:

- 主键: TimePeriod, ROOMID, DATE
- 字段: doctorId, TitleId
- 外键: doctorId (参考Doctor表的id), ROOMID (参考ROOM表的id)
- 关系模式: `Dispatcher(TimePeriod, ROOMID, DATE, doctorId, TitleId)`

13. diagnosis表:

- 主键: id
- 字段: patientId, doctorId, diagnosis, time
- 外键: patientId (参考Patient表的id), doctorId (参考Doctor表的id)
- 关系模式: `diagnosis(id, patientId, doctorId, diagnosis, time)`

14. CheckCombine表:

- 主键: id, itemId
- 字段: checkName
- 外键: itemId (参考checkItems表的id)
- 关系模式: `CheckCombine(id, itemId, checkName)`

范式证明

针对给定的表结构，我们逐个进行3NF的分析：

1. 用户表 (USER):

- 主键: `id`
- 非主属性依赖于主键: `username, email, password, type, AVATAR`
- 3NF: 满足，因为非主属性完全依赖于主键。

2. 患者表 (Patient):

- 主键: `id`
- 非主属性依赖于主键: `isComMem, idCard, active`
- 3NF: 满足，因为非主属性完全依赖于主键。

3. 职称表 (Titles):

- 主键: `id`
- 非主属性依赖于主键: `name`
- 3NF: 满足，因为非主属性完全依赖于主键。

4. 医生表 (Doctor):

- 主键: `id`
- 非主属性依赖于主键: `Tid, active`
- 3NF: 满足，因为非主属性完全依赖于主键。

5. 药品表 (Drug):

- 主键: `id`
- 非主属性依赖于主键: `price`, `Description`, `isBanned`, `Storage`
- 3NF: 满足, 因为非主属性完全依赖于主键。

6. 订单表 (Counter):

- 主键: `id`
- 非主属性依赖于主键: `Pid`, `Did`, `isPaid`, `price`, `DATE`, `type`
- 3NF: 满足, 因为非主属性完全依赖于主键。

7. 挂号关系表 (RegistRelation):

- 主键: `id`
- 非主属性依赖于主键: `ROOMID`, `isFinished`
- 3NF: 满足, 因为非主属性完全依赖于主键。

8. 药品购买表 (MedicinePurchase):

- 主键: `id`, `drugId`
- 非主属性依赖于主键: `amount`
- 3NF: 满足, 因为非主属性完全依赖于主键。

9. 检查项目表 (checkItems):

- 主键: `id`
- 非主属性依赖于主键: `price`, `description`, `MinResult`, `MaxResult`
- 3NF: 满足, 因为非主属性完全依赖于主键。

10. 化验报告表 (LaboratorySheet):

- 主键: `id`, `itemID`
- 非主属性依赖于主键: `checkName`, `beginTime`, `OutputTime`, `result`
- 3NF: 满足, 因为非主属性完全依赖于主键。

11. 房间表 (ROOM):

- 主键: `id`
- 非主属性依赖于主键: `isOccupied`, `QueueLen`
- 3NF: 满足, 因为非主属性完全依赖于主键。

12. 调度员表 (Dispatcher):

- 主键: `TimePeriod`, `ROOMID`, `DATE`
- 非主属性依赖于主键: `doctorId`, `TitleId`
- 3NF: 满足, 因为非主属性完全依赖于主键。

13. 诊断表 (diagnosis):

- 主键: `id`
- 非主属性依赖于主键: `patientId`, `doctorId`, `diagnosis`, `time`
- 3NF: 满足, 因为非主属性完全依赖于主键。

14. 检查组合表 (CheckCombine):

- 主键: `id`, `itemId`

- 非主属性依赖于主键：checkName
- 3NF: 满足，因为非主属性完全依赖于主键。

总体来说，每个表都满足第三范式（3NF），因为非主属性完全依赖于主键，而且没有传递依赖。

数据库优化

使用索引

由于我们的所有关系都满足3NF关系，因此我们对数据库进行了索引优化，加快查询。

```
CREATE INDEX `index_id` ON `USER` (`id`)
CREATE INDEX `index_id` ON `Patient` (`id`)
CREATE INDEX `index_id` ON `Titles` (`id`)
CREATE INDEX `index_id` ON `Doctor` (`id`)
CREATE INDEX `index_id` ON `Drug` (`id`)
CREATE INDEX `index_id` ON `Counter` (`id`)
CREATE INDEX `index_id` ON `RegistRelation` (`id`)
CREATE INDEX `index_id` ON `MedicinePurchase` (`id`)
CREATE INDEX `index_id` ON `checkItems` (`id`)
CREATE INDEX `index_id` ON `LaboratorySheet` (`id`)
CREATE INDEX `index_id` ON `ROOM` (`id`)
CREATE INDEX `index_TimePeriod` ON `Dispatcher` (`TimePeriod`)
CREATE INDEX `index_id` ON `diagnosis` (`id`)
CREATE INDEX `index_id` ON `CheckCombine` (`id`)
```

设置触发器

我们对删除操作增加了触发器

```
CREATE TRIGGER `DELETE_DRUG` BEFORE DELETE ON `drug` FOR EACH ROW
BEGIN
    DELETE FROM `checkcombine` WHERE `id` = OLD.id;
END

CREATE TRIGGER `DELETE_PATIENT` BEFORE DELETE ON `patient` FOR EACH ROW
BEGIN
    IF `patient`.`isComMem` = 1 THEN
        DELETE FROM `counter` WHERE `Pid` = OLD.id;
    END IF;
END
```