# **总体设计**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能 | 输入 | 处理 | 输出 |
| 登录 | 管理员账号/医生账号/患者账号、密码 | 查询数据库中对应的管理员/医生/患者信息，匹配则返回Token，存储到本地localStorage中 | 失败:提示错误  成功:进入系统页面 |
| 信息管理（医生、患者、挂号信息等） | 1.查询特定用户信息。  2.点击“修改”用户按钮。  3.点击“增加”用户按钮。  4.点击“删除”用户按钮 | 1.从数据库中查询相应的用户信息。  2.更新数据库中该用户的信息。  3.新增数据库中该角色的信息。  4.删除数据库中该用户的信息。 | 跳回到当前用户信息列表首页 |
| 排班安排 | 1.点击排班日期。  2.选择排班科室。  3.对该科室的医生进行排班。 | 1.把排班日期存储到本地localStorage中。  2.根据排班科室在数据库中查询出该科室的所有医生。  3.将排班日期和该医生ID存储到Redis中，并限制每个时间段的挂号人数为40个，再将该信息存储到MySQL中。 | 该医生已排班:提示“该医生已排班”  该医生未排班：提示“已成功排班” |
| 预约挂号 | 1.选择挂号的科室。  2.选择挂号日期。  3.选择挂号时间段。 | 1.根据科室在数据库中找出所有该科室的医生。  2.根据日期和科室查询出该科室该天的排班医生。  3.查询Redis数据库，若剩余号源不为零，将剩余号源减1，最后把挂号信息插入到MySQL中。 | 成功或者失败都会有相应的提示 |
| 处理挂号 | 1.查看当天挂号信息。  2.处理某个挂号信息。 | 1.在Token中解析出该医生的ID，根据该医生ID查询出当天所有未处理的挂号信息。  2.根据填写的病因，所开的药物以及所开的检查项目在数据库中更新该挂号信息。 | 成功或者失败都会有相应的提示 |
| 退出登录 | 点击“退出登录” | 清除本地localStorage中的Token | 退出到登录页面 |

后续每周跟进各个模块的详细设计