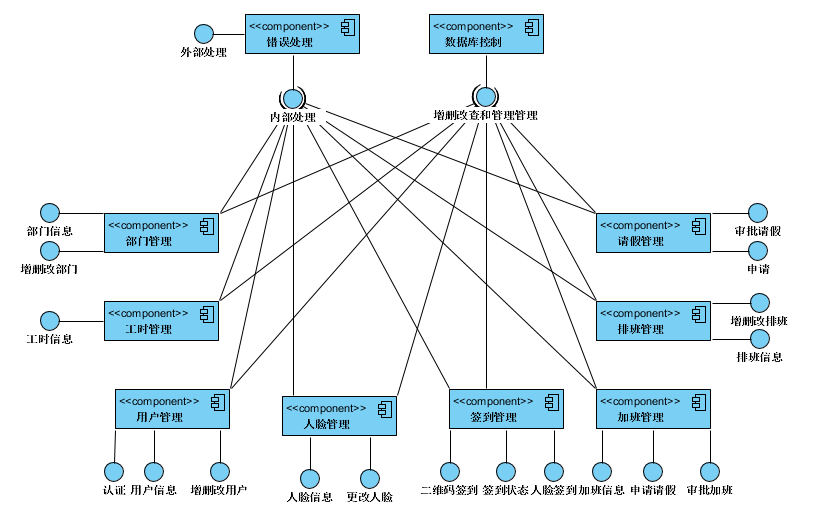
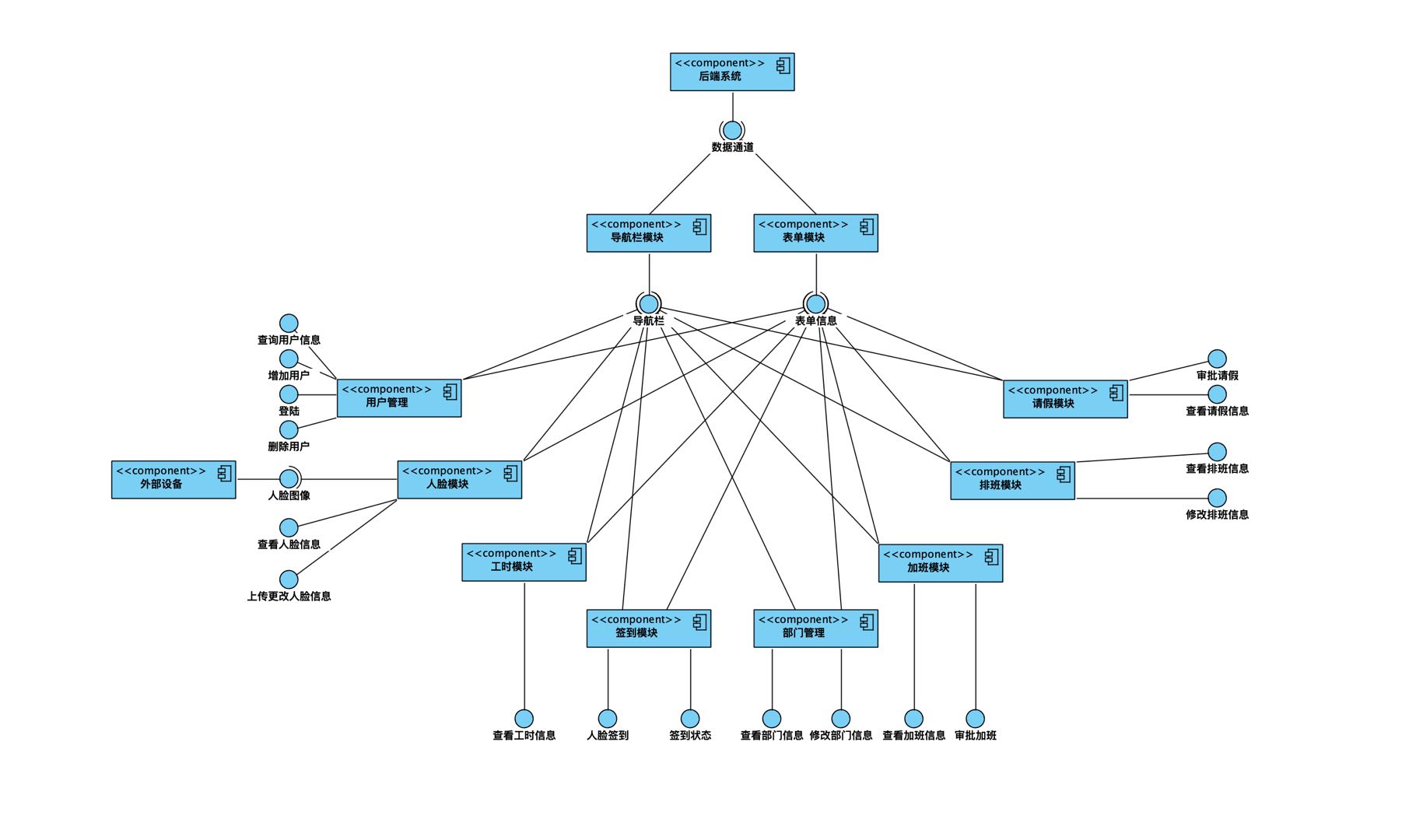
**整体设计部门：**

后端：



前端：



**数据接口部分设计：**

|  |  |  |  |
| --- | --- | --- | --- |
| 接口 | URL | 附加 | 响应 |
| 错误格式响应 |  |  | {  "status": 4xx/5xx,  "message": "error message"  } |
| 用户认证 | POST `/user/auth` | JSON参数：  name:用户姓名  password:用户密码 | {  "status": 200,  "token": "{Token}"  } |
| 用户列表 | GET `/user` | Header：  Authorization: Bearer {Token}  URL Query 参数：  page: 分页 | {  "status": 200,  "total": 100,  "current\_page": 1,  "per\_page": 20,  "data": [ {  "id": 1,  "name": "test",  "role": "user/manager/master",  "department": 1,  "hours": 10  }, ...  ]  } |
| 添加用户 | POST `/user` | Header:  Authorization: Bearer {Token}  JSON 参数：  name: 姓名  password: 密码  department: 部门 | {  "status": 201,  "user\_id": 2  } |
| 修改用户 | PUT `/user/{user\_id}` | Header：  Authorization: Bearer {Token}  JSON参数：  name: 姓名  department: 部门  role: 身份  hours: 工时 | {  “status”: 200  } |
| 获取指定用户信息 | GET `/user/{user\_id}` | Header：  Authorization: Bearer {Token} | {  "status": 200,  "data": {  "name": "test",  ...  }  } |
| 删除用户 | DELETE `/user/{user\_id}` | Header：  Authorization: Bearer {Token} | {  “status”: 200  } |
| 部门列表 | GET `/department` | Header：  Authorization: Bearer {Token}  URL Query 参数：  page: 分页 | {  "status": 200,  "total": 100,  "current\_page": 1,  "per\_page": 20,  "data": [  {  "id": 1,  "name": "X 部门",  "manager": 1  },  ...  ]  } |
| 添加部门 | POST `/department` | Header：  Authorization: Bearer {Token}  JSON参数：  name: 部门名称  manager: 主管 ID | {  “status”:201,  “department\_id”:2  } |
| 编辑部门 | PUT `/department/{department\_id}` | Header：  Authorization: Bearer {Token}  JSON参数：  name:部门名称  manager：主管ID | {  “status”: 200  } |
| 获取指定部门信息 | GET `/department/{department\_id}` | Header：  Authorization: Bearer {Token} | {  “status”:200,  “data”:{  “id”:1,  ...  }  } |
| 删除部门 | DELETE `/department/{department\_id}` | Header：  Authorization: Bearer {Token} | {  “status”: 200  } |
| 获取指定用户可用的人脸信息 | GET `/face/user/{user\_id}` | Header：  Authorization: Bearer {Token} | {  "status": 200,  "data": [  {  "id": 1,  "user\_id": 1,  "info": "xxxxxxx",  "status": "wait/available/discarded"  },  ...  ]  } |
| 更新指定用户人脸信息 | POST `/face/user/{user\_id}` | Header：  Authorization: Bearer {Token}  JSON 参数：  info:人脸信息 | {  “status”: 201，  “face\_id”:2  } |
| 审批人脸信息 | PUT `/face/{face\_id}` | Header：  Authorization: Bearer {Token}  JSON参数：  status:人脸状态 | {  “status”: 200  } |
| 获取工时记录 | GET `/hours` | Header：  Authorization: Bearer {Token}  URL Query 参数：  user\_id: 用户 ID，可选  start\_at: 开始日期，可选  end\_at: 结束日期，可选  page: 分页 | {  "status": 200,  "total": 100,  "current\_page": 1,  "per\_page": 20,  "data": [  {  "id": 1,  "user\_id": 1,  "date": "2019-02-02 11:11:11",  "hours": 12  },  ...  ]  } |
| 排班列表 | GET `/shift` | Header：  Authorization: Bearer {Token}  URL Query 参数：  user\_id: 用户 ID，可选  department\_id: 部门 ID，可选  page: 分页 | {  "status": 200,  "total": 100,  "current\_page": 1,  "per\_page": 20,  "data": [  {  "id": 1,  "user\_id": 1,  "start\_at": "2019-02-02 11:11:11",  "end\_at": "2019-02-02 11:11:11",  "type": "normal/overtime/allovertime",  "status": "no/on/off/leave"  },  ...  ]  } |
| 添加排班 | POST `/shift/user/{user\_id}` | Header：  Authorization: Bearer {Token}  JSON参数：  start\_at: 开始时间  end\_at: 结束时间  type: 排班类型 | {  "status": 201,  "shift\_id": 2  } |
| 部门排班 | POST `/shift/department/{department\_id}` | Header：  Authorization: Bearer {Token}  JSON参数：  start\_at: 开始时间  end\_at: 结束时间  type: 排班类型 | {  "status": 201,  "shift\_ids": [2, 3]  } |
| 更新排班状态 | PUT `/shift/{shift\_id}` | Header：  Authorization: Bearer {Token}  JSON状态：  status: 排班状态 | {  "status": 200  } |
| 删除排班 | DELETE `/shift/{shift\_id}` | Header：  Authorization: Bearer {Token} | {  "status": 200  } |
| 请假列表 | GET `/leave` | Header：  Authorization: Bearer {Token}  URL Query 参数：  page: 分页 | {  "status": 200,  "total": 100,  "current\_page": 1,  "per\_page": 20,  "data": [  {  "id": 1,  "user\_id": 1,  "start\_at": "2019-02-02 11:11:11",  "end\_at": "2019-02-02 11:11:11",  "remark": "身体原因",  "status": "wait/pass/reject"  },  ...  ]  } |
| 获取指定用户请假 | GET `/leave/user/{user\_id}` | Header：  Authorization: Bearer {Token}  URL Query 参数：  page: 分页 | {  "status": 200,  "page": 1,  "data": [  {  "id": 1,  "user\_id": 1,  "start\_at": "2019-02-02 11:11:11",  "end\_at": "2019-02-02 11:11:11",  "remark": "身体原因",  "status": "wait/pass/reject"  },  ...  ]  } |
| 申请请假 | POST `/leave/user/{user\_id}` | Header：  Authorization: Bearer {Token}  JSON参数：  start\_at: 开始时间  end\_at: 结束时间  remark: 请假理由 | {  "status": 201,  "leave\_id": 2  } |
| 审批请假 | PUT `/leave/{leave\_id}` | Header：  Authorization: Bearer {Token}  JSON参数：  status: 状态 | {  "status": 200  } |
| 加班申请列表 | GET `/overtime` | Header：  Authorization: Bearer {Token}  URL Query 参数：  page: 分页 | {  "status": 200,  "total": 100,  "current\_page": 1,  "per\_page": 20,  "data": [  {  "id": 1,  "user\_id": 1,  "start\_at": "2019-02-02 11:11:11",  "end\_at": "2019-02-02 11:11:11",  "remark": "任务未完成",  "status": "wait/pass/reject"  },  ...  ]  } |
| 获取指定用户加班 | GET `/overtime/user/{user\_id}` | Header：  Authorization: Bearer {Token}  URL Query 参数：  page: 分页 | {  "status": 200,  "total": 100,  "current\_page": 1,  "per\_page": 20,  "data": [  {  "id": 1,  "user\_id": 1,  "start\_at": "2019-02-02 11:11:11",  "end\_at": "2019-02-02 11:11:11",  "remark": "任务未完成",  "status": "wait/pass/reject"  },  ...  ]  } |
| 申请加班 | POST `/overtime/user/{user\_id}` | Header：  Authorization: Bearer {Token}  JSON参数：  start\_at: 开始时间  end\_at: 结束时间  remark: 理由 | {  "status": 201,  "overtime\_id": 2  } |
| 审批加班 | PUT `/overtime/{overtime\_id}` | Header：  Authorization: Bearer {Token}  JSON参数：  status: 状态 | {  "status": 200  } |
| 获取用户当前签到状态 | GET `/sign/user/{user\_id}` | Header：  Authorization: Bearer {Token} | no/leave = {  "status": 204  }  on/off = {  "status": 200,  "data": {  "id": 1,  "shift\_id": 1,  "start\_at": "2019-02-02 11:11:11",  "end\_at": "2019-02-02 11:11:11"  }  } |
| 获取二维码 | GET `/sign/qrcode` | Header：  Authorization: Bearer {Token}  说明：二维码解析后为签到使用的Tocken | {  "status": 200,  "data": {  "qrcode": "data:image/png;...",  "count": 10,  "expired\_at": "2019-02-02 11:11:11"  }  } |
| 二维码签到 | POST `/sign/qrcode/user/{user\_id}` | Header：  Authorization: Bearer {Token}  JSON参数：  token: 签到 Token | {  "status": 200,  "sign\_id: 3  } |
| 人脸签到 | POST `/sign/face/user/{user\_id}` | Header：  Authorization: Bearer {Token}  JSON：  - face: 人脸图片的 base64 编码 |  |
| 签退 | POST `/off/sign/{sign\_id}` | Header：  Authorization: Bearer {Token} | {  "status": 200  } |

**人机交互部门设计：**

**控制驱动部分设计：**

数据库采用gorm框架，程序运行时直接将models对应进行数据库迁移。

后端采用gin框架自动进行整体的控制，大体过程如下：

具体过程通过gin框架内部实现。

