

检查即通过实验手段获取患者生理或病理信息,用于诊断疾病、监测治疗效果及预防并发病。检查涵盖了辅助检查和病理学检查等多个层面。处置即基于医学手段的干预措施,包括药物使用、康复治疗、保健措施等。治疗即针对病因或症状的医疗干预,包括药物、手术、放疗等。检查、处置及治疗等内容的完成需要多个科室的医师通力合作。医师在填写相关信息时需要调取 HIS 智慧医疗系统内其他模块的内容,例如医师在开药时需要先在 HIS 智慧医疗系统中检索所需药物,然后导入药物名称、数量等信息。在检查、处置及治疗过程中,医师可以将检查、处置及治疗结果附件上传至电子病历中,用于日后对病案进行管理。检查、处置及治疗结果附件可使用表格、文字或图像等形式。除此之外,HIS 智慧医疗系统亦可清晰地反映检查是否已经完成。

HIS 智慧医疗系统的整体高效运行需要医院门诊、检验科、处置科及药局等各个专业科室充分协同配合、通力合作。只有各科室之间的高效协作,才能确保 HIS 智慧医疗系统的全面运行。在这一过程中,信息的及时传递至关重要,需要专人负责各个环节的信息传递与操作执行。同时,信息的及时更新也是保证 HIS 智慧医疗系统高效运转的关键所在,如果信息更新缓慢,则势必会给患者带来不便,因此 HIS 智慧医疗系统的更新维护也需要每位医务人员根据实际情况进行及时处理,并及时总结反馈,这样才能对 HIS 智慧医疗系统的运行进行优化与完善。总体来讲,HIS 智慧医疗系统的高效运行,需要各个专业科室之间的高度协同,信息传递的及时性和 HIS 智慧医疗系统维护的持续性得到共同保证,只有这样才能确保 HIS 智慧医疗系统最大化发挥其应有的作用,真正为患者提供优质高效的医疗服务。

5.1 辅助检查

辅助检查是医学检查中的一部分,主要用于协助医师诊断和评估患者的健康状况。常见的辅助检查如下。

1. 血液检查

血液检查包括血常规、尿常规、空腹血糖、血脂、肝功能(血清谷草转氨酶、谷丙转氨酶和

总胆红素)、肾功能(血清肌酐和血尿素氮)等。

2. 心电图检查

心电图检查包括常规心电图检查、药物试验心电图、负荷试验心电图、动态心电图等。

3. 影像学检查

影像学检查包括头部、胸部、腹部、盆部、四肢等常规 X 线片,以及 CT 检查和超声检查等。

4. 实验室检查

实验室检查包括各种化验检查,如乙肝系列、微量元素、肿瘤标志物等。

5. 特殊检查

特殊检查包括内分泌功能检查(甲状腺功能)、性腺六项检查、精液常规检查等。

6. 女性专科检查

女性专科检查包括宫颈 TCT 和 HPV 检查、阴道分泌物检查、经阴道或经腹部 B 超检查等。

医师在询问患者的基础信息后,通常需要借助现代科学技术手段进行辅助检查,以协助医师对患者的健康状况进行准确诊断和全面评估。在医师开处方后,患者可以通过医院的收费窗口选择支付方式并支付检查费用。在缴费完成后,患者需要携带处方和缴费小票到指定的检查科室进行各项检查。检验医师完成样本检查后需要先将检查结果手动或自动上传至 HIS 智慧医疗系统,然后医师即可在 HIS 智慧医疗系统中查看患者的检查结果,并结合临床表现对病情进行评估,制定诊疗方案,并开展下一步治疗流程。HIS 智慧医疗系统实现了患者检验结果的实时共享,大大地提高了医患沟通效率。

当检查结束后,除患者可以通过检验科人员打印或医院所设打印机打印出纸质报告外,其检查结果也会以文字、图片或表格的形式展示,在辅助检查的检查结果处展现给医师,医师可在检查状态中对于已检查的项目进行查看。辅助检查报告的表头部分主要包括患者的基础信息,如姓名、性别、年龄等。这些信息可直接从 HIS 智慧医疗系统中提取,或从医师开设的具体辅助检查项目中调用。HIS 智慧医疗系统后台可以实现辅助检查报告与患者信息的自动关联和提取。

报告的表体部分则需要检验科的检验人员根据实际检验情况填写。表体中应包含该检验项目的名称、检验方法、检验结果、参考区间等信息。正常结果一般不需要填写在表体中,表体应主要列出异常结果。对于危急值结果,则需要在表体中特别标注,提示医师及时注意。危急值指可能危及生命的极端异常结果,如血糖 $<2.2\text{mmol/L}$ 、血钾 $>6.5\text{mmol/L}$ 等。

以血常规检查为例,血常规主要反映造血功能和炎症状态,包括红细胞系列、白细胞系列和血小板的定量及比例等指标。在报告表中需要把异常结果特别标注出来,如计数增高或降低,或比例不在参考区间内等。对于严重异常的危急值,需要在备注中特别提示,其他常见辅助检查如生化检验、尿常规、便常规等,报告格式与内容要点类似。关键信息包括检

查项目、结果、异常标注、危急值提示等。结合具体检查的临床意义,检验报告应提供清晰、准确、规范的结果,为医师的后续诊断和治疗提供依据。HIS 智慧医疗系统实现了检验申请、样本采集、结果录入、报告生成及传输的信息化流程管理。规范的电子报告格式有助于医护人员快速地获取关键信息,提高工作效率,同时也为未来报告数据的二次利用提供了结构化支持,如数据分析、研究等。

除此之外,诸如血常规、尿常规等主要以数据或表格形式体现结果的项目,当其中的某个指标出现明显升高或降低时,HIS 智慧医疗系统可以通过特殊的视觉标记(如颜色、闪烁等形式)突显出来,使医师可以更加便捷、准确地分析这一异常结果。特殊的视觉标记可以让医师快速地判断哪些项目需要优先关注,节省医师审阅报告的时间。

在当前的信息时代,大数据分析已经深入应用于医疗保健领域。利用大数据平台可以收集和分析海量的医疗数据,建立数据之间的复杂关联,为医师提供更加系统且全面的分析结果,辅助医师进行精准诊断。大数据还可以应用于患者个人化医疗。同时,HIS 智慧医疗系统可以配合各类医疗设备使用,而且可以发挥医疗设备的更大效用,例如在有朝一日的未来,配合人工智能算法,医学影像可以实现异常病灶的自动识别和定位。HIS 智慧医疗系统就像医师忠实的助手,可以在海量图像中找出可疑区域,过滤掉大量正常图像,使医师可以更加专注于真正需要判读的病变影像,从而提高工作效率。总之,在 HIS 智慧医疗系统中,大数据分析、医疗设备等现代科技的运用可以有效地辅助医师工作,提高医疗诊断与治疗的速度和准确性。

5.1.1 辅助检查列表

医师可以通过患者的病案查看辅助检查列表。辅助检查列表弹窗如图 5-1 所示。

辅助检查列表弹窗的代码如下:

```
<!-- # 第 4 章/辅助检查列表.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="智慧医疗" @click="goTo('http://cnoctave.cn')">
      </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label="挂号" @click="goTo('http://cnoctave.cn')">
    </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label="病历" @click="goTo('http://cnoctave.cn')">
    </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label="检查" @click="goTo('http://cnoctave.cn')">
    </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label="处置" @click="goTo('http://cnoctave.cn')">
    </q-btn>
  </q-header>
</div>
```

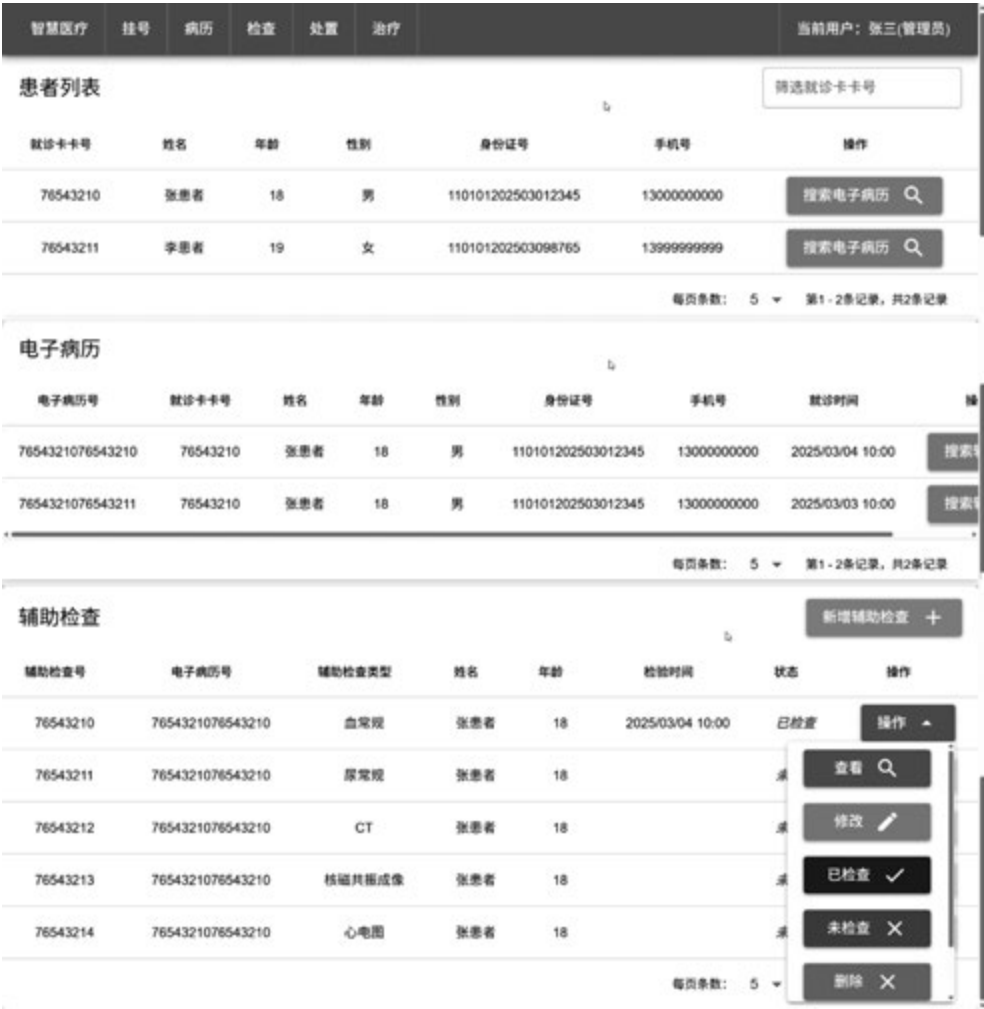


图 5-1 辅助检查列表弹窗

```
<q-separator dark vertical></q-separator>
<q-btn stretch flat label = "治疗" @click = "goTo('http://cnoctave.cn')">
</q-btn>

<q-separator dark vertical></q-separator>
<q-space></q-space>
<q-separator dark vertical></q-separator>
<q-btn stretch flat label = "当前用户: 张三(管理员)" @click =
"goTo('http://cnoctave.cn')"></q-btn>
</q-toolbar>
</q-header>
<q-page-container>
<div>
<q-table title = "患者列表" :rows = "患者列表 rows" :columns = "患者列表
columns" color = "dark" row-key = "id">
```

```

        no-data-label = "无数据" :pagination-label = "setPaginationLabel"
rows-per-page-label = "每页条数: ">
        <template v-slot:top-right>
            <q-input placeholder="筛选就诊卡卡号" clearable:dense="true"
outlined v-model="search_jiu_zhen_ka_id"></q-input>
        </template>
        <template v-slot:body-cell-options="props">
            <q-td:props="props">
                <q-btn color="green" icon-right="search" label="搜索电
子病历" no-caps @click="search(props)"></q-btn>
            </q-td>
        </template>
    </q-table>
    <q-table title="电子病历" :rows="电子病历 rows" :columns="电子病历
columns" color="dark" row-key="id"
        no-data-label="无数据" :pagination-label="setPaginationLabel"
rows-per-page-label="每页条数: ">
        <template v-slot:body-cell-options="props">
            <q-td:props="props">
                <q-btn color="green" label="搜索辅助检查" no-caps
@click="search(props)"></q-btn>
            </q-td>
        </template>
    </q-table>
    <q-table title="辅助检查" :rows="辅助检查 rows" :columns="辅助检查
columns" color="dark" row-key="id"
        no-data-label="无数据" :pagination-label="setPaginationLabel"
rows-per-page-label="每页条数: ">
        <template v-slot:top-right>
            <q-btn color="yellow-10" icon-right="add" label="新增辅助
检查" no-caps @click="add(props)"></q-btn>
        </template>
        <template v-slot:body-cell-state="props">
            <q-td:props="props">
                <div style="color:red;" v-if="props.row.state ===
0"><b><i>未检查</i></b></div>
                <div style="color:green;" v-else><b><i>已检查</i></b></div>
            </q-td>
        </template>
        <template v-slot:body-cell-options="props">
            <q-td:props="props">
                <q-btn-dropdown color="primary" label="操作">
                    <q-list>
<!-- 省略以下代码 -->

```

辅助检查 ORM 的代码如下：

```

# 第 5 章/辅助检查 ORM.py
from sqlalchemy import Column, String, Date, Integer, BigInteger, \
    SmallInteger, ForeignKey, DateTime, TIMESTAMP, Boolean, func
from sqlalchemy.dialects.PostgreSQL import JSONB, ARRAY, JSON

```

```

from sqlalchemy.orm import sessionmaker
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import relationship

class ORM(declarative_base()):
    __tablename__ = '辅助检查'

    id = Column(String(), primary_key=True, nullable=False)
    dzbl_id = Column(String())
    department = Column(String())
    type = Column(String())
    name = Column(String())
    age = Column(Integer())
    state = Column(Integer())
    note = Column(String())
    jian_yan_shi_jian = Column(TIMESTAMP())
    jian_yan_ren = Column(String())
    create_time = Column(TIMESTAMP())
    modify_time = Column(TIMESTAMP())
    op_id = Column(String())
    op_name = Column(String())

```

辅助检查 Model 的代码如下：

```

# 第 5 章/辅助检查 Model.py
from typing import Any
from pydantic import BaseModel as PydanticBaseModel

class BaseModel(PydanticBaseModel):
    class Config:
        arbitrary_types_allowed = True

class Model(BaseModel):
    id: str
    dzbl_id: str | None = None
    department: str | None = None
    type: str | None = None
    name: str | None = None
    age: int | None = None
    state: int | None = None
    note: str | None = None
    jian_yan_shi_jian: str | None = None
    jian_yan_ren: str | None = None
    create_time: str | None = None
    modify_time: str | None = None
    op_id: str | None = None
    op_name: str | None = None

```

辅助检查列表 API 的代码如下：

```

# 第 5 章/辅助检查列表.py
import uvicorn
from typing import Union

```

```

from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from 辅助检查 ORM import ORM
from 辅助检查 Model import Model
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
import datetime
from starlette.responses import JSONResponse
from sqlalchemy import desc

engine = create_engine("替换为数据库连接字符串")
DBSession = sessionmaker(bind=engine)
session = DBSession()

def get_list(page_number, page_size):
    query_result = session.query(ORM).order_by(desc(ORM.create_time))
    count = query_result.count()
    if page_size != 0:
        result = query_result.slice(
            (page_number - 1) * page_size, page_number * page_size).all()
    else:
        result = query_result.all()
    return result, count

app = FastAPI()

@app.get("/fzjc/list")
async def 辅助检查列表(page_number: int = Query(1, description="第 x 页"),
    page_size: int = Query(10, description="每页的大小")):
    try:
        (dzbl_list, length) = get_list(page_number, page_size)
        result = list()
        for md in dzbl_list:
            result.append(
                {"id": md.id,
                 "dzbl_id": md.dzbl_id,
                 "department": md.department,
                 "type": md.type,
                 "name": md.name,
                 "age": md.age,
                 "state": md.state,
                 "note": md.note,
                 "jian_yan_shi_jian": md.jian_yan_shi_jian,
                 "jian_yan_ren": md.jian_yan_ren,
                 "create_time": str(md.create_time).replace("-", "/").replace("None", ""),
                 "modify_time": str(md.modify_time).replace("-", "/").replace("None", ""),
                 "op_id": md.op_id,
                 "op_name": md.op_name})

        return Result(retcode=0, text="成功",
            result={"list": result, "page_number": page_number,
                "page_size": page_size, "total": length}
        )

```

```
except APIException as e:
    return JsonResponse(
        status_code = e.status,
        content = {"retcode": e.retcode, "text": e.text, "result": e.result},
    )
```

5.1.2 新增辅助检查

辅助检查需要送检科室、检验类型、患者姓名、年龄、处方医师姓名、检验医师姓名、检验时间和备注等信息。处方医师需要填写送检科室、检验类型、患者姓名、年龄、处方医师姓名和备注,检验医师需要填写检验医师姓名和检验时间。新增辅助检查弹窗如图 5-2 所示。



图 5-2 新增辅助检查弹窗

新增辅助检查弹窗的代码如下：

```
<!-- # 第 5 章/新增辅助检查.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="智慧医疗" @click="goTo('http://cnoctave.cn')">
      </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="挂号" @click="goTo('http://cnoctave.cn')">
      </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="病历" @click="goTo('http://cnoctave.cn')">
      </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="检查" @click="goTo('http://cnoctave.cn')">
      </q-btn>
        <q-separator dark vertical></q-separator>
  </q-header>
  <q-main>
    <q-form>
      <q-input type="text" value="采血室" />
      <q-input type="text" value="血常规" />
      <q-input type="text" value="张患者" />
      <q-input type="text" value="60" />
      <q-input type="text" value="张三" />
      <q-input type="text" value="张医师" />
      <q-input type="text" value="2025" />
      <q-input type="text" value="3" />
      <q-input type="text" value="3" />
      <q-input type="text" value="10" />
      <q-input type="text" value="0" />
      <q-input type="text" value="一般血常规 乙肝五项" />
    </q-form>
  </q-main>
</q-layout>
</div>
```

```

        <q-btn stretch flat label = "处置" @click = "goTo('http://cnoctave.cn')">
    </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label = "治疗" @click = "goTo('http://cnoctave.cn')">
    </q-btn>
        <q-separator dark vertical></q-separator>
        <q-space></q-space>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label = "当前用户: 张三(管理员)" @click =
"goTo('http://cnoctave.cn')"></q-btn>
        </q-toolbar>
    </q-header>
    <q-page-container>
        <div class = "q-ma-md">
            <div class = "text-h6">新增辅助检查</div>
            <q-btn color = "green" icon-right = "save" label = "保存" no-caps @click =
"add()"></q-btn>
            <q-btn color = "red" icon-right = "close" label = "取消" no-caps @click =
"cancel()"></q-btn>
            <div> &nbsp;</div>
            <div class = "row">
                <div class = "col">
                    <p>送检科室</p>
                </div>
                <div class = "col">
                    <p>检验类型</p>
                </div>
                <div class = "col">
                    <p>患者姓名</p>
                </div>
                <div class = "col">
                    <p>年龄</p>
                </div>
                <div class = "col">
                    <p>处方医师姓名</p>
                </div>
                <div class = "col">
                    <p>检验医师姓名</p>
                </div>
            </div>
            <div class = "row">
                <div class = "col">
                    <q-select outlined :model-value = "md.department" use-input
hide-selected fill-input
                    input-debounce = "0" :options = "department_list">
                        <template v-slot:no-option>
                            <q-item>
                                <q-item-section class = "text-grey">
                                    无数据匹配, 请输入正确的科室
                                </q-item-section>
                            </q-item>
                        </template>
                    </q-select>
                </div>
            </div>
        </div>
    </q-page-container>
<!-- 省略以下代码 -->

```

新增辅助检查 API 的代码如下：

```
# 第 5 章/新增辅助检查.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from 辅助检查 ORM import ORM
from 辅助检查 Model import Model
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
import datetime
from starlette.responses import JSONResponse

engine = create_engine("替换为数据库连接字符串")
DBSession = sessionmaker(bind=engine)
session = DBSession()

def add(md):
    session.add(md)
    session.commit()

app = FastAPI()

@app.post("/fzjc/add")
async def 新增辅助检查(md: Annotated[Model, Body(description="新增辅助检查")]):
    try:
        if not md.name:
            return Result(retcode=1, text="必须填写姓名")
        id_ = IdGenerator.gen()
        this_time = datetime.datetime.now()
        md.create_time = this_time
        md.modify_time = this_time
        add(ORM(id=id_,
                dzbl_id=md.dzbl_id,
                department=md.department,
                type=md.type,
                name=md.name,
                age=md.age,
                state=md.state,
                note=md.note,
                jian_yan_shi_jian=md.jian_yan_shi_jian,
                jian_yan_ren=md.jian_yan_ren,
                create_time=this_time,
                modify_time=this_time,
                op_id=md.op_id,
                op_name=md.op_name))
        return Result(retcode=0, text="成功",
                      result={
                          "id": id_,
```

```
        "dzbl_id": md.dzbl_id,
        "department": md.department,
        "type": md.type,
        "name": md.name,
        "age": md.age,
        "state": md.state,
        "note": md.note,
        "jian_yan_shi_jian": str(md.jian_yan_shi_jian).replace("-", "/").
replace("None", ""),
        "jian_yan_ren": md.jian_yan_ren,
        "create_time": str(this_time).replace("-", "/").replace("None", ""),
        "modify_time": str(this_time).replace("-", "/").replace("None", ""),
        "op_id": md.op_id,
        "op_name": md.op_name
    })
except APIException as e:
    return JSONResponse(
        status_code=e.status,
        content={"retcode": e.retcode, "text": e.text, "result": e.result},
    )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)
```

5.1.3 修改辅助检查

修改辅助检查弹窗如图 5-3 所示。

智慧医疗

挂号

病历

检查

处置

治疗

当前用户: 张三(管理员)

修改辅助检查

保存

取消

送检科室	检验类型	患者姓名	年龄	处方医师姓名	检验医师姓名
采血室	血常规	张患者	60	张三	张医师

检验时间

2025

年

3

月

3

日

10

时

0

分

备注

一般血常规 乙肝五项

图 5-3 修改辅助检查弹窗

修改辅助检查弹窗的代码如下：

```
<!-- #第5章/修改辅助检查.html -->
<div id="q-app">
```

```

    <q-layout view="hHh LpR fFf">
      <q-header elevated class="bg-primary text-white">
        <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
          <q-btn stretch flat label="智慧医疗" @click="goTo('http://cnoctave.cn')">
</q-btn>
          <q-separator dark vertical></q-separator>
          <q-btn stretch flat label="挂号" @click="goTo('http://cnoctave.cn')">
</q-btn>
          <q-separator dark vertical></q-separator>
          <q-btn stretch flat label="病历" @click="goTo('http://cnoctave.cn')">
</q-btn>
          <q-separator dark vertical></q-separator>
          <q-btn stretch flat label="检查" @click="goTo('http://cnoctave.cn')">
</q-btn>
          <q-separator dark vertical></q-separator>
          <q-btn stretch flat label="处置" @click="goTo('http://cnoctave.cn')">
</q-btn>
          <q-separator dark vertical></q-separator>
          <q-btn stretch flat label="治疗" @click="goTo('http://cnoctave.cn')">
</q-btn>
          <q-separator dark vertical></q-separator>
          <q-space></q-space>
          <q-separator dark vertical></q-separator>
          <q-btn stretch flat label="当前用户: 张三(管理员)" @click =
"goTo('http://cnoctave.cn')"></q-btn>
        </q-toolbar>
      </q-header>
      <q-page-container>
        <div class="q-ma-md">
          <div class="text-h6">修改辅助检查</div>
          <q-btn color="green" icon-right="save" label="保存" no-caps @click =
"add()"></q-btn>
          <q-btn color="red" icon-right="close" label="取消" no-caps @click =
"cancel()"></q-btn>
          <div>&nbsp;</div>
          <div class="row">
            <div class="col">
              <p>送检科室</p>
            </div>
            <div class="col">
              <p>检验类型</p>
            </div>
            <div class="col">
              <p>患者姓名</p>
            </div>
            <div class="col">
              <p>年龄</p>
            </div>
            <div class="col">
              <p>处方医师姓名</p>
            </div>
            <div class="col">
              <p>检验医师姓名</p>
            </div>
          </div>
        </div>
      </q-page-container>
    </q-layout>

```

```

        </div>
        <div class="row">
            <div class="col">
                <q-select outlined :model="md.department" use="input
hide-selected fill="input
                input-debounce="0" :options="department_list">
                <template v-slot:no-option>
                    <q-item>
                        <q-item-section class="text-grey">
                            无数据匹配,请输入正确的科室
                        </q-item-section>
                    </q-item>
                </template>
            </q-select>
        </div>
    <!-- 省略以下代码 -->

```

修改辅助检查 API 的代码如下：

```

# 第 5 章/修改辅助检查.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from 辅助检查 ORM import ORM
from 辅助检查 Model import Model
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
import datetime
from starlette.responses import JSONResponse

engine = create_engine("替换为数据库连接字符串")
DBSession = sessionmaker(bind=engine)
session = DBSession()

def get_orm(id):
    result = session.query(ORM).get(id)
    return result

def merge(md):
    session.merge(md)
    session.commit()

app = FastAPI()

@app.post("/fzjc/mod")
async def 修改辅助检查(md: Annotated[Model, Body(description="修改辅助检查")]):
    try:
        if not md.id:
            return Result(retcode=1, text="ID 无效")

```

```

result = get_orm(md.id)
if not result:
    return Result(retcode=1, text="辅助检查不存在")
if not md.name:
    return Result(retcode=1, text="必须填写姓名")
this_time = datetime.datetime.now()
md.modify_time = this_time
merge(ORM(id=md.id,
          dzbl_id=md.dzbl_id,
          department=md.department,
          type=md.type,
          name=md.name,
          age=md.age,
          state=md.state,
          note=md.note,
          jian_yan_shi_jian=md.jian_yan_shi_jian,
          jian_yan_ren=md.jian_yan_ren,
          create_time=md.create_time,
          modify_time=this_time,
          op_id=md.op_id,
          op_name=md.op_name))
return Result(retcode=0, text="成功",
              result={
                  "id": md.id,
                  "dzbl_id": md.dzbl_id,
                  "department": md.department,
                  "type": md.type,
                  "name": md.name,
                  "age": md.age,
                  "state": md.state,
                  "note": md.note,
                  "jian_yan_shi_jian": str(md.jian_yan_shi_jian).replace("-", "/").
replace("None", ""),
                  "jian_yan_ren": md.jian_yan_ren,
                  "create_time": str(md.create_time).replace("-", "/").
replace("None", ""),
                  "modify_time": str(this_time).replace("-", "/").
replace("None", ""),
                  "op_id": md.op_id,
                  "op_name": md.op_name
              })
except APIException as e:
    return JSONResponse(
        status_code=e.status,
        content={"retcode": e.retcode, "text": e.text, "result": e.result},
    )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)

```

5.1.4 删除辅助检查

删除辅助检查弹窗如图 5-4 所示。



图 5-4 删除辅助检查弹窗

删除辅助检查弹窗的代码如下：

```
<!-- # 第 5 章/删除辅助检查.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="智慧医疗" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="挂号" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="病历" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="检查" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="处置" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="治疗" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-space></q-space>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="当前用户: 张三(管理员)" @click="
goTo('http://cnoctave.cn')"></q-btn>
      </q-toolbar>
    </q-header>
  </q-page-container>
</div>
```

```

        <q-dialog v-model="is_show" :full-width="true">
          <q-card>
            <q-card-section>
              <div class="text-h6"><q-icon name="warning" color="
"warning" size="2em"></q-icon>删除辅助检查</div>
            </q-card-section>
            <q-card-section class="q-pt-none">
              <p>确定要删除吗?</p>
              <div>&nbsp;</div>
            </q-card-section>
            <q-card-actions align="right">
              <q-btn color="primary" icon-right="check" label="确定"
no-caps
              @click="del(id_)"></q-btn>
              <q-btn color="red" icon-right="close" label="取消"
no-caps
              @click="close()"></q-btn>
            </q-card-actions>
          </q-card>
        </q-dialog>
      </div>
    </q-page-container>
  </q-layout>
</div>

```

删除辅助检查 API 的代码如下：

```

# 第 5 章/删除辅助检查.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from 辅助检查 ORM import ORM
from 辅助检查 Model import Model
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
import datetime
from starlette.responses import JSONResponse

engine = create_engine("替换为数据库连接字符串")
DBSession = sessionmaker(bind=engine)
session = DBSession()

def get_orm(id):
    result = session.query(ORM).get(id)
    return result

def delete(md):
    session.delete(md)
    session.commit()

app = FastAPI()

```

```

@app.post("/fzjc/delete")
async def 删除辅助检查(id: Annotated[str | None, Query(description="删除辅助检查")]):
    try:
        if not id:
            return Result(retcode=1, text="ID 无效")
        md = get_orm(id)
        delete(md)
        return Result(retcode=0, text="成功",
            result={
                "id": md.id,
                "dzbl_id": md.dzbl_id,
                "department": md.department,
                "type": md.type,
                "name": md.name,
                "age": md.age,
                "state": md.state,
                "note": md.note,
                "jian_yan_shi_jian": str(md.jian_yan_shi_jian).replace("-", "/").
replace("None", ""),
                "jian_yan_ren": md.jian_yan_ren,
                "create_time": str(md.create_time).replace("-", "/").
replace("None", ""),
                "modify_time": str(md.modify_time).replace("-", "/").
replace("None", ""),
                "op_id": md.op_id,
                "op_name": md.op_name
            })
    except APIException as e:
        return JSONResponse(
            status_code=e.status,
            content={"retcode": e.retcode, "text": e.text, "result": e.result},
        )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)

```

5.1.5 辅助检查标记

在检查完成后,检验医师需要将辅助检查标记为已检查。HIS 智慧医疗系统也允许检验医师将辅助检查标记为未检查。辅助检查标记 API 的代码如下:

```

# 第5章/辅助检查标记.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from 辅助检查 ORM import ORM
from 辅助检查 Model import Model
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
import datetime

```

```

from starlette.responses import JSONResponse

engine = create_engine("替换为数据库连接字符串")
DBSession = sessionmaker(bind=engine)
session = DBSession()

def get_orm(id):
    result = session.query(ORM).get(id)
    return result

def merge(md):
    session.merge(md)
    session.commit()

app = FastAPI()

@app.post("/fzjc/mod_state")
async def 辅助检查标记(md: Annotated[Model, Body(description="辅助检查标记")],
                        state: Annotated[int, Body(description="辅助检查标记")]):
    try:
        md = get_orm(id)
        if not md.id:
            return Result(retcode=1, text="ID 无效")
        result = get_orm(md.id)
        if not result:
            return Result(retcode=1, text="辅助检查不存在")
        if not md.name:
            return Result(retcode=1, text="必须填写姓名")
        this_time = datetime.datetime.now()
        md.modify_time = this_time
        merge(ORM(id=md.id,
                  dzbl_id=md.dzbl_id,
                  department=md.department,
                  type=md.type,
                  name=md.name,
                  age=md.age,
                  state=state,
                  note=md.note,
                  jian_yan_shi_jian=md.jian_yan_shi_jian,
                  jian_yan_ren=md.jian_yan_ren,
                  create_time=md.create_time,
                  modify_time=this_time,
                  op_id=md.op_id,
                  op_name=md.op_name))
        return Result(retcode=0, text="成功",
                      result={
                          "id": md.id,
                          "dzbl_id": md.dzbl_id,
                          "department": md.department,
                          "type": md.type,

```

```

        "name": md.name,
        "age": md.age,
        "state": state,
        "note": md.note,
        "jian_yan_shi_jian": str(md.jian_yan_shi_jian).replace("-", "/").
replace("None", ""),
        "jian_yan_ren": md.jian_yan_ren,
        "create_time": str(md.create_time).replace("-", "/").
replace("None", ""),
        "modify_time": str(this_time).replace("-", "/").replace("None", ""),
        "op_id": md.op_id,
        "op_name": md.op_name
    })
except APIException as e:
    return JSONResponse(
        status_code=e.status,
        content={"retcode": e.retcode, "text": e.text, "result": e.result},
    )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)

```

5.1.6 辅助检查打印标签

在进行辅助检查时常常需要先打印不干胶标签并粘贴在样本容器上,然后检验仪器可以通过条码识别的方式识别样本。不干胶标签需要包含条形码、患者姓名和样本编号。辅助检查打印标签 API 的代码如下:

```

# 第5章/辅助检查打印标签.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from 辅助检查 ORM import ORM
from 辅助检查 Model import Model
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
import datetime
from starlette.responses import JSONResponse, FileResponse
import json
import barcode
from barcode.writer import ImageWriter
from PIL import Image, ImageDraw, ImageFont
import cups
import os

engine = create_engine("替换为数据库连接字符串")
DBSession = sessionmaker(bind=engine)
session = DBSession()

```

```

def get_orm(id):
    result = session.query(ORM).get(id)
    return result

app = FastAPI()

@app.post("/fzjc/tag")
async def 辅助检查打印标签(id: Annotated[str | None, Query(description = "辅助检查打印标签")]):
    try:
        if not id:
            return Result(retcode = 1, text = "ID 无效")
        conn = cups.Connection()
        printers = conn.getPrinters()
        is_printer_ok = False
        for printer in printers:
            if printer["name"] == "替换为打印机名称":
                is_printer_ok = True
                result = get_orm(id)
                id_ = IdGenerator.gen()
                ean = barcode.get_barcode_class('code128')
                barcode_obj = ean(id_, writer = ImageWriter())
                barcode_obj.save(f'{id_}.png')
                barcode_image = Image.open(f'{id_}.png')
                new_image = Image.new('RGB', (barcode_image.width, barcode_image.height + 50),
color = (255, 255, 255))
                new_image.paste(barcode_image, (0, 0))
                draw = ImageDraw.Draw(new_image)
                font = ImageFont.truetype("苹方特粗.ttf", 20)
                text_width, text_height = ImageDraw.textsize(f"{result.name} {id_}", font =
font)

                draw.text(((barcode_image.width - text_width) / 2,
barcode_image.height),
f"{result.name} {id_}",
font = font,
fill = (0, 0, 0))
                new_image.save(f'barcode_{id_}.png')
                with open(f'barcode_{id_}.png', "rb") as f:
                    data = f.read()
                    print_job = conn.printFile(
printer["name"], data, f'barcode_{id_}.png', "PNG", {})
                    conn.printJob(print_job)
                if not is_printer_ok:
                    return Result(retcode = 1, text = f"打印机"{printer['name']}"异常")
                return Result(retcode = 0, text = "成功",
result = {})
        except APIException as e:
            return JSONResponse(
status_code = e.status,
content = {"retcode": e.retcode, "text": e.text, "result": e.result},
)

if __name__ == "__main__":
    uvicorn.run(app, host = "127.0.0.1", port = 8000)

```

5.2 病理学检查

病理学检查指用以检查机体器官、组织或细胞中的病理改变的病理形态学方法。为探讨器官、组织或细胞所发生的疾病过程,医师可采用某种病理形态学检查的方法检查所发生的病变,探讨病变产生的原因、发病机理和病变的发生发展过程,最后做出病理诊断。病理形态学的检查方法,首先观察大体标本的病理改变,然后切取一定大小的病变组织,接着用病理组织学方法制成病理切片,最后用显微镜进一步检查病变。

病理学检查可以分为常规组织病理检查、术中冰冻病理检查、细胞病理检查、免疫组化病理检查和特殊染色检查。

5.2.1 常规组织病理检查

在常规组织病理检查时,临床医师送检病变标本,病理科医师在接收到组织标本后对其进行编号,经过取材、脱水、包埋、切片、染色及封片等一系列处理之后制成病理切片,对病理切片进行观察后给出最终病理诊断结果,确定病变性质及类型。常规组织病理检查是临床最常用的检查方法。

5.2.2 常规组织病理检查报告模板

常规组织病理检查报告模板弹窗如图 5-5 所示。

智慧医疗

挂号

病历

检查

处置

治疗

当前用户: 张三(管理员)

常规组织病理检查报告模板

下载报告 ↓ 清空 ×

病理切片

▼

病变性质

▼

病变类型

▼

诊断

▼

图 5-5 常规组织病理检查报告模板弹窗

常规组织病理检查报告模板弹窗的代码如下:

```
<!-- # 第 5 章/常规组织病理检查报告模板.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
```

```

    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="智慧医疗" @click="goTo('http://cnoctave.cn')">
      </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="挂号" @click="goTo('http://cnoctave.cn')">
      </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="病历" @click="goTo('http://cnoctave.cn')">
      </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="检查" @click="goTo('http://cnoctave.cn')">
      </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="处置" @click="goTo('http://cnoctave.cn')">
      </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="治疗" @click="goTo('http://cnoctave.cn')">
      </q-btn>
        <q-separator dark vertical></q-separator>
        <q-space></q-space>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="当前用户: 张三(管理员)" @click="goTo(
('http://cnoctave.cn'))"></q-btn>
      </q-toolbar>
    </q-header>
    <q-page-container>
      <div class="q-ma-md">
        <div class="text-h6">常规组织病理检查报告模板</div>
        <q-btn color="green" icon-right="south" label="下载报告" no-caps
@click="add()"></q-btn>
        <q-btn color="red" icon-right="close" label="清空" no-caps @click="
"cancel()"></q-btn>
      <div>&nbsp;</div>
      <q-expansion-item expand-separator icon="edit" label="病理切片"
class="bg-grey-3">
        <q-card>
          <q-card-section>
            <div v-for="(v, k) in md.blqp_img_urls" :key="k">
              <q-img :src="v" spinner-color="white" style="
"max-height: 30vh; max-width: 30vw">
            </q-img>
          </div>
        </q-card-section>
      </q-card>
    </q-expansion-item>
    <div>&nbsp;</div>
    <q-expansion-item expand-separator icon="edit" label="病变性质"
class="bg-grey-3">
      <q-card>
        <q-card-section>
          <q-input :dense="true" outlined v-model="md.bbzx" :
autogrow="true"></q-input>
        </q-card-section>
      </q-card>
    </q-expansion-item>

```

```

        <div> &nbsp;</div>
        <q-expansion-item expand-separator icon = "edit" label = "病变类型"
class = "bg-grey-3">
            <q-card>
                <q-card-section>
                    <q-input :dense = "true" outlined v-model = "md.bblx" :
autogrow = "true"></q-input>
                </q-card-section>
            </q-card>
        </q-expansion-item>
        <div> &nbsp;</div>
        <q-expansion-item expand-separator icon = "edit" label = "诊断" class =
"bg-grey-3">
            <q-card>
                <q-card-section>
                    <q-input :dense = "true" outlined v-model = "md.zd" :autogrow =
"true"></q-input>
                </q-card-section>
            </q-card>
        </q-expansion-item>
    <div> &nbsp;</div>
</div>
</q-page-container>
</q-layout>
</div>

```

5.2.3 下载常规组织病理检查报告

医师可以单击“下载报告”按钮下载 Word 格式的常规组织病理检查报告。常规组织病理检查报告如图 5-6 所示。

常规组织病理检查报告

*姓名: 性别: 年龄:

*病理切片

*病变性质
癌变

*病变类型
浸润性乳腺癌

*诊断
图 1 中可见 3mm×5mm 结节，发现癌组织侵犯周围神经。初步诊断为浸润性乳腺癌

*医师签字:

*报告日期: 年 月 日

图 5-6 常规组织病理检查报告

常规组织病理检查报告 Model 的代码如下：

```
# 第 5 章/常规组织病理检查报告 Model.py
from typing import Any
from pydantic import BaseModel as PydanticBaseModel

class BaseModel(PydanticBaseModel):
    class Config:
        arbitrary_types_allowed = True

class Model(BaseModel):
    blqp_img_urls: list | None = None
    bbxz: str | None = None
    bblx: str | None = None
    zd: str | None = None
```

下载常规组织病理检查报告 API 的代码如下：

```
# 第 5 章/下载常规组织病理检查报告.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from 常规组织病理检查报告 Model import Model
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
import datetime
from starlette.responses import JSONResponse, FileResponse
from docx import Document
from docx.shared import Inches

engine = create_engine("替换为数据库连接字符串")
DBSession = sessionmaker(bind=engine)
session = DBSession()

app = FastAPI()

@app.post("/download/cgzzbljcbg")
async def 下载常规组织病理检查报告(
    md: Annotated[Model | None, Query(description="下载常规组织病理检查报告")]):
    try:
        if not md:
            return Result(retcode=1, text="常规组织病理检查报告不能为空")
        id_ = IdGenerator.gen()
        document = Document('template.docx')
        document.add_heading('常规组织病理检查报告', 0)
        document.add_heading(
            '姓名: _____ 性别: _____ 年龄: _____ ',
            level=1)
        document.add_heading('病理切片', level=1)
        for img_url in md['blqp_img_urls']:
            document.add_picture(img_url, width=Inches(1.25))
        document.add_heading('病变性质', level=1)
```

```

document.add_paragraph(md[ 'bbxz'])
document.add_heading('病变类型', level = 1)
document.add_paragraph(md[ 'bblx'])
document.add_heading('诊断', level = 1)
document.add_paragraph(md[ 'zd'])
document.add_heading('医师签字: _____', level = 1)
document.add_heading('报告日期: ____年____月____日', level = 1)
document.add_page_break()
document.save(f'{id_}.docx')
return FileResponse(path = f'{id_}.docx')
except APIException as e:
    return JSONResponse(
        status_code = e.status,
        content = {"retcode": e.retcode, "text": e.text, "result": e.result},
    )

if __name__ == "__main__":
    uvicorn.run(app, host = "127.0.0.1", port = 8000)

```

5.2.4 术中冰冻病理检查

术中冰冻病理检查是一项极为重要的手术辅助检查技术。在手术过程中,首先医师会切除疑似病变组织,然后立即送到病理科进行快速病理检查,以明确病变的性质和严重程度。术中冰冻病理检查的具体流程如下:

(1) 将新鲜切除的组织迅速置于一20℃左右的冰冻切片机中进行冰冻固定,等待大约一分钟直到组织完全被冰冻。

(2) 将其切成薄片,进行染色等后续处理,制成可以在显微镜下观察的病理切片。整个过程必须在30min内完成。

(3) 病理科医师在显微镜下观察冰冻切片,判断病变是否为肿瘤,如果是肿瘤,则判断其良恶性、扩散和淋巴结受累情况等。

(4) 做出病理诊断报告。

术中冰冻病理的主要目的是指导手术方案。根据病理报告,外科医师可以判断肿瘤的生物行为,明确手术切除范围,决定是否清扫淋巴结,从而避免过度或不足切除。如果发现肿瘤已经浸润周围正常组织或向远处转移,则医师可及时终止手术,避免过度创伤。总之,术中冰冻病理检查可以即时提供病变性质和严重程度的评估,极大地帮助外科医师制定个体化的手术治疗方案。术中冰冻病理检查要求病理科工作迅速、准确,外科医师则需要根据病理报告调整手术策略,两者之间顺畅地进行信息沟通至关重要。同时,样本的取得和处理也需要标准化操作,避免冰冻造成的组织学偏差影响判断。尽管术中冰冻病理检查的准确度不如常规石蜡切片病理检查,但是术中冰冻病理检查仍是手术过程中不可或缺的重要辅助手段。

5.2.5 术中冰冻病理检查报告模板

术中冰冻病理检查报告模板弹窗如图5-7所示。

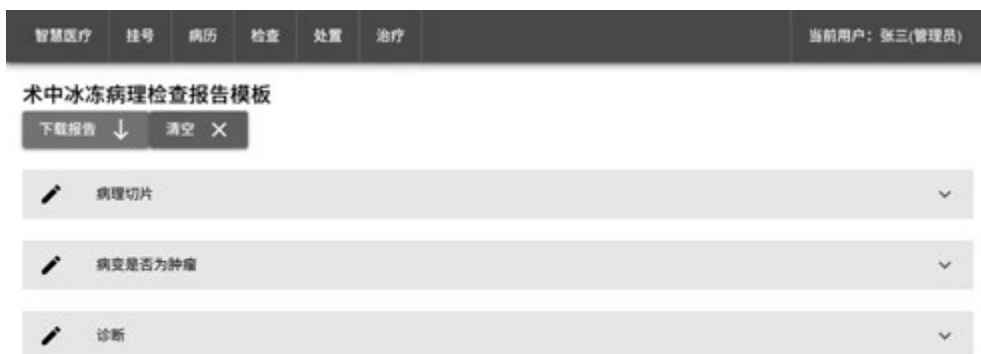


图 5-7 术中冰冻病理检查报告模板弹窗

术中冰冻病理检查报告模板弹窗的代码如下：

```
<!-- # 第 5 章/术中冰冻病理检查报告模板.html -->
<div id = "q-app">
  <q-layout view = "hHh LpR fFf">
    <q-header elevated class = "bg-primary text-white">
      <q-toolbar class = "bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label = "智慧医疗" @click = "goTo('http://cnoctave.cn')">
      </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label = "挂号" @click = "goTo('http://cnoctave.cn')">
      </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label = "病历" @click = "goTo('http://cnoctave.cn')">
      </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label = "检查" @click = "goTo('http://cnoctave.cn')">
      </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label = "处置" @click = "goTo('http://cnoctave.cn')">
      </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label = "治疗" @click = "goTo('http://cnoctave.cn')">
      </q-btn>
      <q-separator dark vertical></q-separator>
      <q-space></q-space>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label = "当前用户: 张三(管理员)" @click =
"goTo('http://cnoctave.cn')"></q-btn>
    </q-toolbar>
  </q-header>
  <q-page-container>
    <div class = "q-ma-md">
```

```

        <div class="text-h6">术中冰冻病理检查报告模板</div>
        <q-btn color="green" icon-right="south" label="下载报告" no-caps
@click="add()"></q-btn>
        <q-btn color="red" icon-right="close" label="清空" no-caps @click=
"cancel()"></q-btn>
        <div> &nbsp;</div>
        <q-expansion-item expand-separator icon="edit" label="病理切片"
class="bg-grey-3">
            <q-card>
                <q-card-section>
                    <div v-for="(v, k) in md.blqp_img_urls" :key="k">
                        <q-img :src="v" spinner-color="white" style=
"max-height: 30vh; max-width: 30vw">
                    </q-img>
                </div>
            </q-card-section>
        </q-card>
    </q-expansion-item>
    <div> &nbsp;</div>
    <q-expansion-item expand-separator icon="edit" label="病变是否为肿
瘤" class="bg-grey-3">
        <q-card>
            <q-card-section>
                <q-input :dense="true" outlined v-model="md.bbsfwzl" :
autogrow="true"></q-input>
            </q-card-section>
        </q-card>
    </q-expansion-item>
    <div> &nbsp;</div>
    <q-expansion-item expand-separator icon="edit" label="诊断" class=
"bg-grey-3">
        <q-card>
            <q-card-section>
                <q-input :dense="true" outlined v-model="md.zd" :autogrow=
"true"></q-input>
            </q-card-section>
        </q-card>
    </q-expansion-item>
    <div> &nbsp;</div>
</div>
</q-page-container>
</q-layout>
</div>

```

5.2.6 下载术中冰冻病理检查报告

医师可以单击“下载报告”按钮下载 Word 格式的术中冰冻病理检查报告。术中冰冻病理检查报告如图 5-8 所示。

术中冰冻病理检查报告 Model 的代码如下：

术中冰冻病理检查报告

*姓名: _____ 性别: _____ 年龄: _____

*病理切片



*病变是否为肿瘤
是

*诊断
3mm×1.2mm 良性肿瘤、扩散不严重、淋巴结无明显受累

*医师签字: _____

*报告日期: ____年__月__日

图 5-8 术中冰冻病理检查报告

```
# 第 5 章/术中冰冻病理检查报告 Model.py
from typing import Any
from pydantic import BaseModel as PydanticBaseModel

class BaseModel(PydanticBaseModel):
    class Config:
        arbitrary_types_allowed = True

class Model(BaseModel):
    blqp_img_urls: list | None = None
    bbsfwzl: str | None = None
    zd: str | None = None
```

下载术中冰冻病理检查报告 API 的代码如下:

```
# 第 5 章/下载术中冰冻病理检查报告.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from 术中冰冻病理检查报告 Model import Model
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
import datetime
from starlette.responses import JSONResponse, FileResponse
from docx import Document
from docx.shared import Inches
```

```

engine = create_engine("替换为数据库连接字符串")
DBSession = sessionmaker(bind=engine)
session = DBSession()

app = FastAPI()

@app.post("/download/szbdbljcbg")
async def 下载术中冰冻病理检查报告(
    md: Annotated[Model | None, Query(description="下载术中冰冻病理检查报告")]):
    try:
        if not md:
            return Result(retcode=1, text="术中冰冻病理检查报告不能为空")
        id_ = IdGenerator.gen()
        document = Document('template.docx')
        document.add_heading('术中冰冻病理检查报告', 0)
        document.add_heading(
            '姓名: _____ 性别: _____ 年龄: _____ ',
            level=1)
        document.add_heading('病理切片', level=1)
        for img_url in md['blqp_img_urls']:
            document.add_picture(img_url, width=Inches(1.25))
        document.add_heading('病变是否为肿瘤', level=1)
        document.add_paragraph(md['bbsfwzl'])
        document.add_heading('诊断', level=1)
        document.add_paragraph(md['zd'])
        document.add_heading('医师签字: _____', level=1)
        document.add_heading('报告日期: ___年___月___日', level=1)
        document.add_page_break()
        document.save(f'{id_}.docx')
        return FileResponse(path=f'{id_}.docx')
    except APIException as e:
        return JSONResponse(
            status_code=e.status,
            content={"retcode": e.retcode, "text": e.text, "result": e.result},
        )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)

```

5.2.7 细胞病理检查

细胞病理学检查是一项重要的病理学辅助检查技术,主要针对各种细胞学标本进行形态学检查,通过观察标本中的细胞形态、数量、排列等变化来检查细胞是否存在肿瘤性病变。细胞病理学检查的主要标本来源如下:

- (1) 痰液。
- (2) 尿液。
- (3) 脑脊液。
- (4) 胸腹水。
- (5) 擦拭涂片(如宫颈涂片)。

(6) 细针穿刺标本。

标本中可能含有病变部位脱落的异常细胞。与组织病理学相比,细胞病理学检查取材更简便,对病人创伤小,操作过程快速、经济,并且可反复进行动态监测。

细胞病理学检查在许多肿瘤的筛查中发挥着重要作用,尤其是对妇科肿瘤的筛查。以宫颈癌的涂片筛查为例,每年进行 1~2 次即可提高宫颈癌和癌前病变的检出率,降低宫颈癌的发病率和死亡率。因为在癌变过程中肿瘤细胞更易脱落进入各种体液,所以细胞学检查可检查出异常增生的癌细胞,从而实现早期发现、早期诊断、早期治疗。细胞病理学检查以结果主要分为阴性、不典型的鳞状上皮细胞、不典型的腺上皮细胞和明细胞癌等,对于阳性和可疑结果还需要进行组织病理学检查以进一步明确。尽管细胞学检查存在假阴性的可能,但细胞病理学具有无创、经济、快速等优点,可作为多种恶性肿瘤的首选筛查手段,对提高肿瘤诊治水平意义重大。

5.2.8 细胞病理检查报告模板

细胞病理检查报告模板弹窗如图 5-9 所示。

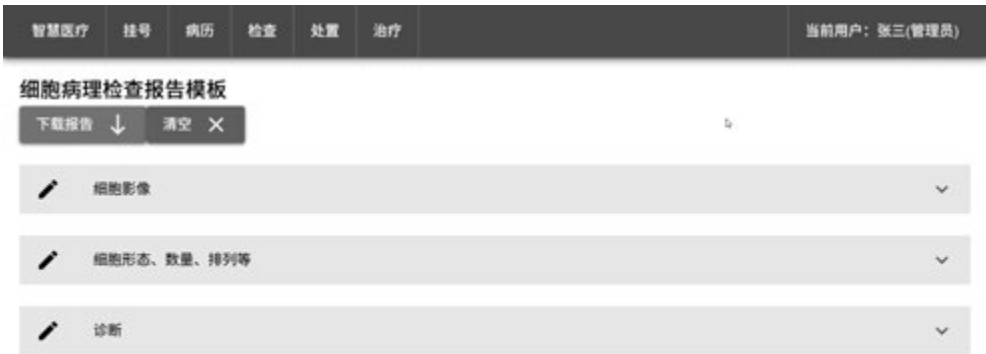


图 5-9 细胞病理检查报告模板弹窗

细胞病理检查报告模板弹窗的代码如下：

```
<!-- # 第 5 章/细胞病理检查报告模板.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="智慧医疗" @click="goTo('http://cnoctave.cn')">
      </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label="挂号" @click="goTo('http://cnoctave.cn')">
    </q-btn>
  </q-header>
  <q-main>
    <q-section>细胞影像</q-section>
    <q-section>细胞形态、数量、排列等</q-section>
    <q-section>诊断</q-section>
  </q-main>
</q-layout>
</div>
```

```

        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label = "病历" @click = "goTo('http://cnoctave.cn')">
    </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label = "检查" @click = "goTo('http://cnoctave.cn')">
    </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label = "处置" @click = "goTo('http://cnoctave.cn')">
    </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label = "治疗" @click = "goTo('http://cnoctave.cn')">
    </q-btn>
        <q-separator dark vertical></q-separator>
        <q-space></q-space>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label = "当前用户: 张三(管理员)" @click =
"goTo('http://cnoctave.cn')"></q-btn>
        </q-toolbar>
    </q-header>
    <q-page-container>
        <div class = "q-ma-md">
            <div class = "text-h6">细胞病理检查报告模板</div>
            <q-btn color = "green" icon-right = "south" label = "下载报告" no-caps
@click = "add()"></q-btn>
            <q-btn color = "red" icon-right = "close" label = "清空" no-caps @click
= "cancel()"></q-btn>
            <div> &nbsp;</div>
            <q-expansion-item expand-separator icon = "edit" label = "细胞影像" class =
"bg-grey-3">
                <q-card>
                    <q-card-section>
                        <div v-for = "(v, k) in md.xbyx_img_urls" :key = "k">
                            <q-img :src = "v" spinner-color = "white" style =
"max-height: 30vh; max-width: 30vw">
                                </q-img>
                            </div>
                        </q-card-section>
                    </q-card>
                </q-expansion-item>
                <div> &nbsp;</div>
                <q-expansion-item expand-separator icon = "edit" label = "细胞形态、数量、
排列等" class = "bg-grey-3">
                    <q-card>
                        <q-card-section>
                            <q-input :dense = "true" outlined v-model = "md.xb" :autogrow =
"true"></q-input>
                        </q-card-section>
                    </q-card>
                </q-expansion-item>
                <div> &nbsp;</div>
                <q-expansion-item expand-separator icon = "edit" label = "诊断" class =
"bg-grey-3">
                    <q-card>
                        <q-card-section>

```

```

        <q-input :dense="true" outlined v-model="md.zd" :autogrow
= "true"></q-input>
        </q-card-section>
    </q-card>
    </q-expansion-item>
    <div> &nbsp;</div>
</div>
</q-page-container>
</q-layout>
</div>

```

5.2.9 下载细胞病理检查报告

医师可以单击“下载报告”按钮下载 Word 格式的细胞病理检查报告。细胞病理检查报告如图 5-10 所示。

细胞病理检查报告

*姓名: _____ 性别: _____ 年龄: _____

*细胞影像



*细胞形态、数量、排列等
红细胞数量 $6.7 \times 10^{12}/L$ 。数量偏多，形态无明显异常

*诊断
不存在肿瘤性病变

*医师签字: _____

*报告日期: ____年__月__日

图 5-10 细胞病理检查报告

细胞病理检查报告 Model 的代码如下：

```

# 第 5 章/细胞病理检查报告 Model.py
from typing import Any
from pydantic import BaseModel as PydanticBaseModel

class BaseModel(PydanticBaseModel):
    class Config:
        arbitrary_types_allowed = True

class Model(BaseModel):
    xbyx_img_urls: list | None = None
    xb: str | None = None
    zd: str | None = None

```

下载细胞病理检查报告 API 的代码如下：

```
# 第 5 章/下载细胞病理检查报告.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from 细胞病理检查报告 Model import Model
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
import datetime
from starlette.responses import JSONResponse, FileResponse
from docx import Document
from docx.shared import Inches

engine = create_engine("替换为数据库连接字符串")
DBSession = sessionmaker(bind=engine)
session = DBSession()

app = FastAPI()

@app.post("/download/xbbljcjb")
async def 下载细胞病理检查报告(
    md: Annotated[Model | None, Query(description="下载细胞病理检查报告")]):
    try:
        if not md:
            return Result(retcode=1, text="细胞病理检查报告不能为空")
        id_ = IdGenerator.gen()
        document = Document('template.docx')
        document.add_heading('细胞病理检查报告', 0)
        document.add_heading(
            '姓名: _____ 性别: _____ 年龄: _____ ',
            level=1)
        document.add_heading('细胞影像', level=1)
        for img_url in md['xbyx_img_urls']:
            document.add_picture(img_url, width=Inches(1.25))
        document.add_heading('细胞形态、数量、排列等', level=1)
        document.add_paragraph(md['xb'])
        document.add_heading('诊断', level=1)
        document.add_paragraph(md['zd'])
        document.add_heading('医师签字: _____', level=1)
        document.add_heading('报告日期: ____年__月__日', level=1)
        document.add_page_break()
        document.save(f'{id_}.docx')
        return FileResponse(path=f'{id_}.docx')
    except APIException as e:
        return JSONResponse(
            status_code=e.status,
            content={"retcode": e.retcode, "text": e.text, "result": e.result},
        )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)
```

5.2.10 免疫组化病理检查

免疫组化技术是病理诊断中的一项重要辅助检查手段,它利用抗原抗体特异性识别原理在组织切片上定位特定的细胞组分和功能分子,为病理诊断提供关键的分子病理学信息。免疫组化的基本原理如下:

- (1) 先通过一抗与组织切片中的特定靶抗原发生特异性结合。
- (2) 再辅以二抗、酶标记物等进行信号放大。
- (3) 最后与底物反应形成可视化的颜色沉淀,从而在细胞和组织结构内显示目标抗原的定位。

不同的一抗可以识别各种蛋白质抗原,如细胞循环蛋白、激素受体、肿瘤抑制基因等。免疫组化技术的主要应用如下:

- (1) 确定细胞起源和分型,如鉴别肿瘤的上皮细胞起源或淋巴细胞起源。
- (2) 判断恶性程度,如 Ki-67 指标反映肿瘤增殖活性。
- (3) 预测预后,如错义突变蛋白 p53 过表达常提示预后不良。
- (4) 指导治疗,如雌激素受体和 HER-2 的表达可预测患者对激素治疗和靶向治疗的反应性。

免疫组化可提供常规 HE 染色无法提供的重要生物学信息,是病理科确定诊断、鉴别诊断的关键技术手段,可减少误诊和漏诊,提高病理诊断的准确性,然而,免疫组化结果的解读需要综合分析病例的临床资料、显微镜下形态改变等整体情况,这样才能避免单一指标的误导。

5.2.11 免疫组化病理检查报告模板

免疫组化病理检查报告模板弹窗如图 5-11 所示。



图 5-11 免疫组化病理检查报告模板弹窗

免疫组化病理检查报告模板弹窗的代码如下：

```
<!-- # 第 5 章/免疫组化病理检查报告模板.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="智慧医疗" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="挂号" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="病历" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="检查" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="处置" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="治疗" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="当前用户: 张三(管理员)" @click="
"goTo('http://cnoctave.cn')"></q-btn>
      </q-toolbar>
    </q-header>
    <q-page-container>
      <div class="q-ma-md">
        <div class="text-h6">免疫组化病理检查报告模板</div>
        <q-btn color="green" icon-right="south" label="下载报告" no-caps
@click="add()"></q-btn>
        <q-btn color="red" icon-right="close" label="清空" no-caps @click="
"cancel()"></q-btn>
        <div> &nbsp;</div>
        <q-expansion-item expand-separator icon="edit" label="病理切片"
class="bg-grey-3">
          <q-card>
            <q-card-section>
              <div v-for="(v, k) in md.blqp_img_urls" :key="k">
                <q-img :src="v" spinner-color="white" style="
"max-height: 30vh; max-width: 30vw">
              </q-img>
            </div>
          </q-card-section>
        </q-card>
      </q-expansion-item>
      <div> &nbsp;</div>
      <q-expansion-item expand-separator icon="edit" label="诊断" class =
"bg-grey-3">
```

```

        <q-card>
          <q-card-section>
            <q-input :dense = "true" outlined v-model = "md.zd" :autogrow =
"true"></q-input>
          </q-card-section>
        </q-card>
      </q-expansion-item>
      <div> &nbsp;</div>
    </div>
  </q-page-container>
</q-layout>
</div>

```

5.2.12 下载免疫组化病理检查报告

医师可以单击“下载报告”按钮下载 Word 格式的免疫组化病理检查报告。免疫组化病理检查报告如图 5-12 所示。

免疫组化病理检查报告

* 姓名: _____ 性别: _____ 年龄: _____

* 病理切片



* 诊断
错义突变蛋白 p53 过表达, 有预后不良的风险

* 医师签字: _____

* 报告日期: ____年__月__日

图 5-12 免疫组化病理检查报告

免疫组化病理检查报告 Model 的代码如下:

```

# 第 5 章/免疫组化病理检查报告 Model.py
from typing import Any
from pydantic import BaseModel as PydanticBaseModel

class BaseModel(PydanticBaseModel):
    class Config:
        arbitrary_types_allowed = True

class Model(BaseModel):
    blqp_img_urls: list | None = None
    zd: str | None = None

```

下载免疫组化病理检查报告 API 的代码如下:

```

# 第5章/下载免疫组化病理检查报告.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from 免疫组化病理检查报告 Model import Model
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
import datetime
from starlette.responses import JSONResponse, FileResponse
from docx import Document
from docx.shared import Inches

engine = create_engine("替换为数据库连接字符串")
DBSession = sessionmaker(bind=engine)
session = DBSession()

app = FastAPI()

@app.post("/download/myzhbljcbg")
async def 下载免疫组化病理检查报告(
    md: Annotated[Model | None, Query(description="下载免疫组化病理检查报告")] :
    try:
        if not md:
            return Result(retcode=1, text="免疫组化病理检查报告不能为空")
        id_ = IdGenerator.gen()
        document = Document('template.docx')
        document.add_heading('免疫组化病理检查报告', 0)
        document.add_heading(
            '姓名: _____ 性别: _____ 年龄: _____ ',
            level=1)
        document.add_heading('病理切片', level=1)
        for img_url in md['blqp_img_urls']:
            document.add_picture(img_url, width=Inches(1.25))
        document.add_heading('诊断', level=1)
        document.add_paragraph(md['zd'])
        document.add_heading('医师签字: _____', level=1)
        document.add_heading('报告日期: ____年____月____日', level=1)
        document.add_page_break()
        document.save(f'{id_}.docx')
        return FileResponse(path=f'{id_}.docx')
    except APIException as e:
        return JSONResponse(
            status_code=e.status,
            content={"retcode": e.retcode, "text": e.text, "result": e.result},
        )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)

```

5.2.13 特殊染色检查

特殊染色技术应用不同的染色方法和试剂,通过染色反应使组织切片中的特定成分产

生差异显色,从而显示其形态分布和定性定量变化,为病理诊断提供重要的形态学和化学信息。特殊染色的基本原理是利用不同组织成分的化学组成和亲和力差异,经优化的染色条件使目标组分与特异性染料发生选择性结合并着色。常用的特殊染色方法如下:

- (1) 组织学染色。Mallory 三色联合染色可分别染色细胞核、细胞浆和胶原纤维,显示组织的总体结构。Masson 三色染色可同时显示细胞组分、胶原纤维和肌细胞。
- (2) 糖原染色。PAS 染色可显示黏液、糖原等含糖组分。
- (3) 脂肪染色。油红 O 染色可显示组织中的中性脂肪。
- (4) 钙盐染色。佛诺氏钙染色可显示骨化和其他钙化灶。
- (5) 胶原染色。马里克染色可区分胶原纤维的类型。
- (6) 微生物染色。抗酸染色可显示抗酸杆菌。GMS 染色可显示真菌。

特殊染色可提供组织化学信息,对肿瘤的诊断分类、炎症性疾病的评估及微生物检出等有价值,但其临床应用也存在一定限制,结果解读需结合形态学改变进行整体分析。特殊染色技术的应用需根据具体病例需求进行选择,是组织病理学诊断的有益补充手段。

5.2.14 特殊染色检查报告模板

特殊染色检查报告模板弹窗如图 5-13 所示。

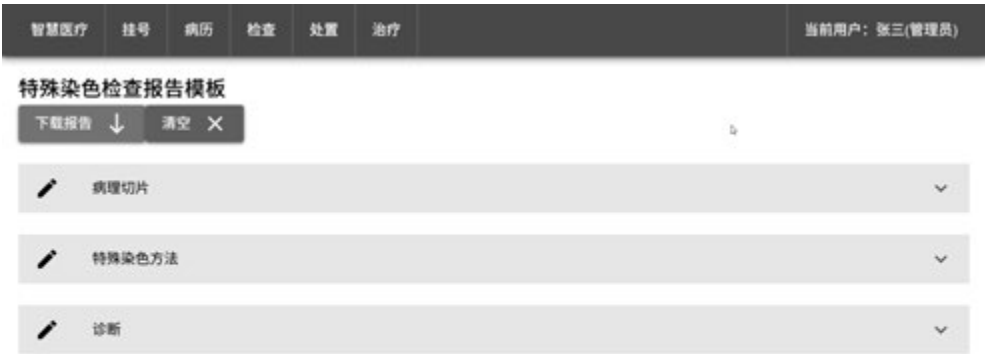


图 5-13 特殊染色检查报告模板弹窗

特殊染色检查报告模板弹窗的代码如下:

```
<!-- # 第 5 章/特殊染色检查报告模板.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="智慧医疗" @click="goTo('http://cnoctave.cn')">
      </q-toolbar>
    </q-header>
  </q-layout>
</div>
```

```

        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label = "挂号" @click = "goTo('http://cnoctave.cn')">
    </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label = "病历" @click = "goTo('http://cnoctave.cn')">
    </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label = "检查" @click = "goTo('http://cnoctave.cn')">
    </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label = "处置" @click = "goTo('http://cnoctave.cn')">
    </q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label = "治疗" @click = "goTo('http://cnoctave.cn')">
    </q-btn>
        <q-separator dark vertical></q-separator>
        <q-space></q-space>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label = "当前用户: 张三(管理员)" @click =
"goTo('http://cnoctave.cn')"></q-btn>
    </q-toolbar>
</q-header>
<q-page-container>
    <div class = "q-ma-md">
        <div class = "text-h6">特殊染色检查报告模板</div>
        <q-btn color = "green" icon-right = "south" label = "下载报告" no-caps
@click = "add()"></q-btn>
        <q-btn color = "red" icon-right = "close" label = "清空" no-caps @click =
"cancel()"></q-btn>
        <div> &nbsp;</div>
        <q-expansion-item expand-separator icon = "edit" label = "病理切片"
class = "bg-grey-3">
            <q-card>
                <q-card-section>
                    <div v-for = "(v, k) in md.blqp_img_urls" :key = "k">
                        <q-img :src = "v" spinner-color = "white" style =
"max-height: 30vh; max-width: 30vw">
                            </q-img>
                    </div>
                </q-card-section>
            </q-card>
        </q-expansion-item>
        <div> &nbsp;</div>
        <q-expansion-item expand-separator icon = "edit" label = "特殊染色方法"
class = "bg-grey-3">
            <q-card>
                <q-card-section>
                    <q-input :dense = "true" outlined v-model = "md.tsrsff" :
autogrow = "true"></q-input>
                </q-card-section>
            </q-card>
        </q-expansion-item>
        <div> &nbsp;</div>
        <q-expansion-item expand-separator icon = "edit" label = "诊断" class =
"bg-grey-3">

```

```

        <q-card>
          <q-card-section>
            <q-input :dense = "true" outlined v-model = "md.zd" :autogrow =
"true"></q-input>
          </q-card-section>
        </q-card>
      </q-expansion-item>
      <div> &nbsp;</div>
    </div>
  </q-page-container>
</q-layout>
</div>

```

5.2.15 下载特殊染色检查报告

医师可以单击“下载报告”按钮下载 Word 格式的特殊染色检查报告。特殊染色检查报告如图 5-14 所示。

特殊染色检查报告

*姓名: _____ 性别: _____ 年龄: _____

*病理切片



*特殊染色方法
微生物染色

*诊断
GMS 染色显示真菌感染

*医师签字: _____

*报告日期: ____年__月__日

图 5-14 特殊染色检查报告

特殊染色检查报告 Model 的代码如下：

```

# 第 5 章/特殊染色检查报告 Model.py
from typing import Any
from pydantic import BaseModel as PydanticBaseModel

class BaseModel(PydanticBaseModel):
    class Config:

```

```

        arbitrary_types_allowed = True

class Model(BaseModel):
    blqp_img_urls: list | None = None
    tsrsff: str | None = None
    zd: str | None = None

```

下载特殊染色检查报告 API 的代码如下：

```

# 第 5 章/下载特殊染色检查报告.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from 特殊染色检查报告 Model import Model
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
import datetime
from starlette.responses import JSONResponse, FileResponse
from docx import Document
from docx.shared import Inches

engine = create_engine("替换为数据库连接字符串")
DBSession = sessionmaker(bind=engine)
session = DBSession()

app = FastAPI()

@app.post("/download/tsrsjcbg")
async def 下载特殊染色检查报告(
    md: Annotated[Model | None, Query(description="下载特殊染色检查报告")]):
    try:
        if not md:
            return Result(retcode=1, text="特殊染色检查报告不能为空")
        id_ = IdGenerator.gen()
        document = Document('template.docx')
        document.add_heading('特殊染色检查报告', 0)
        document.add_heading(
            '姓名: _____ 性别: _____ 年龄: _____ ',
            level=1)
        document.add_heading('病理切片', level=1)
        for img_url in md['blqp_img_urls']:
            document.add_picture(img_url, width=Inches(1.25))
        document.add_heading('特殊染色方法', level=1)
        document.add_paragraph(md['tsrsff'])
        document.add_heading('诊断', level=1)
        document.add_paragraph(md['zd'])
        document.add_heading('医师签字: _____', level=1)
        document.add_heading('报告日期: ____年____月____日', level=1)
        document.add_page_break()
        document.save(f'{id_}.docx')
        return FileResponse(path=f'{id_}.docx')
    except APIException as e:

```

```

return JsonResponse(
    status_code = e.status,
    content = {"retcode": e.retcode, "text": e.text, "result": e.result},
)

if __name__ == "__main__":
    uvicorn.run(app, host = "127.0.0.1", port = 8000)

```

5.2.16 分子病理学检查

分子病理学是基于分子生物学技术检查体内遗传物质的变化以实现疾病的精准诊断和个体化治疗的一门新兴交叉学科。分子病理学的主要技术如下：

(1) 原位杂交。使用配基特异性的 DNA 或 RNA 探针,检查细胞和组织中目标核酸序列的存在及表达水平。常用于检查基因扩增、重排及特异性突变。

(2) 聚合酶链式反应(PCR)。通过引物特异性扩增目标 DNA 序列,可实现病原体检查及肿瘤特异性基因突变的检查。实时荧光 PCR 技术具有高通量、高灵敏度的优势。

(3) 高通量测序。作为新一代测序技术,可对全部基因组或外显子组进行深度测序,实现基因表达谱分析及肿瘤的全基因组分析,有利于发现新型生物标志物。

(4) 流式细胞技术。应用抗体标记、细胞计数及分选技术,可检查循环肿瘤细胞及免疫细胞群的变化,评价预后和指导免疫治疗。

分子病理学检查可精确定位病变的分子特征,提供传统病理学无法提供的遗传信息,有助于疾病精确诊断、进展监测、预后评估、指导个体化治疗方案选择和疗效预测。分子病理学将推动精准医学模式的深入应用,但其临床转化还面临检验规范化、结果解读和数据共享等方面的挑战。

5.2.17 分子病理学检查报告模板

分子病理学检查报告模板弹窗如图 5-15 所示。

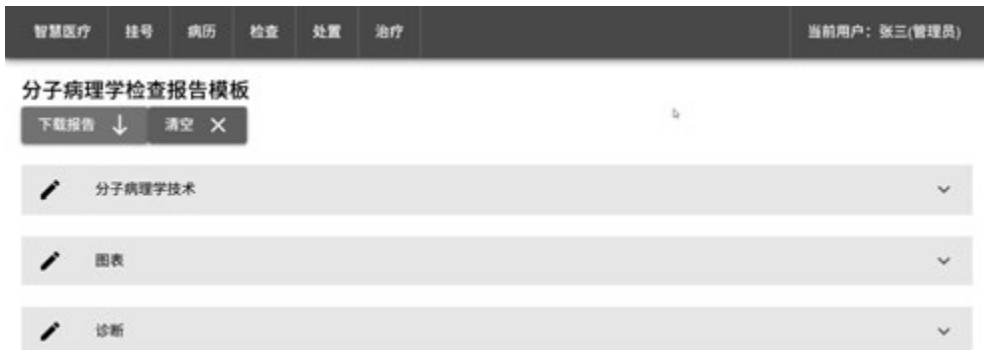


图 5-15 分子病理学检查报告模板弹窗

分子病理学检查报告模板弹窗的代码如下：

```
<!-- # 第 5 章/分子病理学检查报告模板.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="智慧医疗" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="挂号" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="病历" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="检查" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="处置" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="治疗" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="当前用户：张三(管理员)" @click="
"goTo('http://cnoctave.cn')"></q-btn>
      </q-toolbar>
    </q-header>
    <q-page-container>
      <div class="q-ma-md">
        <div class="text-h6">分子病理学检查报告模板</div>
        <q-btn color="green" icon-right="south" label="下载报告" no-caps
@click="add()"></q-btn>
        <q-btn color="red" icon-right="close" label="清空" no-caps @click="
"cancel()"></q-btn>
        <div>&nbsp;</div>
        <q-expansion-item expand-separator icon="edit" label="分子病理学技
术" class="bg-grey-3">
          <q-card>
            <q-card-section>
              <q-input :dense="true" outlined v-model="md.fzblxjs" :
autogrow="true"></q-input>
            </q-card-section>
          </q-card>
        </q-expansion-item>
        <div>&nbsp;</div>
        <q-expansion-item expand-separator icon="edit" label="图表" class="
"bg-grey-3">
          <q-card>
            <q-card-section>
              <div v-for="(v, k) in md.tb_img_urls" :key="k">
```

```

        <q-img :src = "v" spinner-color = "white" style =
"max-height: 30vh; max-width: 30vw">
        </q-img>
    </div>
    </q-card-section>
</q-card>
</q-expansion-item>
<div> &nbsp;</div>
<q-expansion-item expand-separator icon = "edit" label = "诊断" class =
"bg-grey-3">
    <q-card>
        <q-card-section>
            <q-input :dense = "true" outlined v-model = "md.zd" :autogrow =
"true"></q-input>
            </q-card-section>
        </q-card>
    </q-expansion-item>
    <div> &nbsp;</div>
</div>
</q-page-container>
</q-layout>
</div>

```

5.2.18 下载分子病理学检查报告

医师可以单击“下载报告”按钮下载 Word 格式的分子病理学检查报告。分子病理学检查报告如图 5-16 所示。



图 5-16 分子病理学检查报告

分子病理学检查报告 Model 的代码如下：

```
# 第 5 章/分子病理学检查报告 Model.py
from typing import Any
from pydantic import BaseModel as PydanticBaseModel

class BaseModel(PydanticBaseModel):
    class Config:
        arbitrary_types_allowed = True

class Model(BaseModel):
    tb_img_urls: list | None = None
    fzblxjs: str | None = None
    zd: str | None = None
```

下载分子病理学检查报告 API 的代码如下：

```
# 第 5 章/下载分子病理学检查报告.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from 分子病理学检查报告 Model import Model
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
import datetime
from starlette.responses import JSONResponse, FileResponse
from docx import Document
from docx.shared import Inches

engine = create_engine("替换为数据库连接字符串")
DBSession = sessionmaker(bind=engine)
session = DBSession()

app = FastAPI()

@app.post("/download/tsrsjcbg")
async def 下载分子病理学检查报告(
    md: Annotated[Model | None, Query(description="下载分子病理学检查报告")]):
    try:
        if not md:
            return Result(retcode=1, text="分子病理学检查报告不能为空")
        id_ = IdGenerator.gen()
        document = Document('template.docx')
        document.add_heading('分子病理学检查报告', 0)
        document.add_heading(
            '姓名: _____ 性别: _____ 年龄: _____ ',
            level=1)
        document.add_heading('分子病理学技术', level=1)
        document.add_paragraph(md['fzblxjs'])
        document.add_heading('图表', level=1)
        for img_url in md['tb_img_urls']:
            document.add_picture(img_url, width=Inches(1.25))
```

```

document.add_heading('诊断', level = 1)
document.add_paragraph(md['zd'])
document.add_heading('医师签字: _____', level = 1)
document.add_heading('报告日期: ____年__月__日', level = 1)
document.add_page_break()
document.save(f'{id_}.docx')
return FileResponse(path=f'{id_}.docx')
except APIException as e:
    return JSONResponse(
        status_code=e.status,
        content={"retcode": e.retcode, "text": e.text, "result": e.result},
    )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)

```

5.3 处置及治疗方案

面对疾病,医师需要先对患者的病情进行全面评估,然后制定系统而精准的治疗与处置方案。

5.3.1 处置方案

处置方案是 HIS 智慧医疗系统中的重要组成部分。处置通常指基于医学原理的各类救治性措施,包括康复处置和保健处置等,其目标在于改善和提升患者的身体健康状况。处置方案的制定和执行,需要医师根据患者的实际病情及其他相关指标,结合自身的专业知识和临床经验,制定出针对性强、切合实际的处置措施。同时,处置方案的实施还需要医院各相关科室的通力合作,如检验科、手术室、药局等,这样才能确保各项处置行为得以顺利开展。在具体实施中,医师还需密切关注患者的反馈信息,适时调整处置方案,确保取得预期的治疗效果。

5.3.2 处置方案模板

处置方案模板弹窗如图 5-17 所示。



图 5-17 处置方案模板弹窗

处置方案模板弹窗的代码如下：

```

<!-- # 第 5 章/处置方案模板.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="智慧医疗" @click="goTo('http://cnoctave.cn')">
      </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label="挂号" @click="goTo('http://cnoctave.cn')">
    </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label="病历" @click="goTo('http://cnoctave.cn')">
    </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label="检查" @click="goTo('http://cnoctave.cn')">
    </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label="处置" @click="goTo('http://cnoctave.cn')">
    </q-btn>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label="治疗" @click="goTo('http://cnoctave.cn')">
    </q-btn>
      <q-separator dark vertical></q-separator>
      <q-space></q-space>
      <q-separator dark vertical></q-separator>
      <q-btn stretch flat label="当前用户：张三(管理员)" @click="
"goTo('http://cnoctave.cn')"></q-btn>
    </q-toolbar>
  </q-header>
  <q-page-container>
    <div class="q-ma-md">
      <div class="text-h6">处置方案模板</div>
      <q-btn color="green" icon-right="south" label="下载记录" no-caps
@click="add()"></q-btn>
      <q-btn color="red" icon-right="close" label="清空" no-caps @click=
"cancel()"></q-btn>
      <div>&nbsp;</div>
      <q-expansion-item expand-separator icon="edit" label="处置方案"
class="bg-grey-3">
        <q-card>
          <q-card-section>
            <q-input :dense="true" outlined v-model="md.czfa" :
autogrow="true"></q-input>
          </q-card-section>
        </q-card>
      </q-expansion-item>
      <div>&nbsp;</div>
      <q-expansion-item expand-separator icon="edit" label="诊断" class=
"bg-grey-3">
        <q-card>
          <q-card-section>

```

```

        <q-input :dense = "true" outlined v-model = "md.zd" :autogrow =
"true"></q-input>
        </q-card-section>
    </q-card>
    </q-expansion-item>
    <div> &nbsp;</div>
</div>
</q-page-container>
</q-layout>
</div>

```

5.3.3 下载处置记录处方笺

医师可以单击“下载记录”按钮下载 Word 格式的处置记录处方笺。处置记录处方笺如图 5-18 所示。

处方笺

*姓名: _____ 性别: _____ 年龄: _____

*处置方案
 输血科: 输血 400 毫升, 输血后复查血红蛋白水平

*诊断
 外伤后贫血, 血红蛋白降低

*医师签字: _____

*处置日期: ____年__月__日

图 5-18 处置记录处方笺

处置记录 Model 的代码如下:

```

# 第 5 章/处置记录 Model.py
from typing import Any
from pydantic import BaseModel as PydanticBaseModel

class BaseModel(PydanticBaseModel):
    class Config:
        arbitrary_types_allowed = True

class Model(BaseModel):
    czfa: list | None = None
    zd: str | None = None

```

下载处置记录处方笺 API 的代码如下:

```

# 第5章/下载处置记录.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from 处置记录 Model import Model
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
import datetime
from starlette.responses import JSONResponse, FileResponse
from docx import Document
from docx.shared import Inches

engine = create_engine("替换为数据库连接字符串")
DBSession = sessionmaker(bind=engine)
session = DBSession()

app = FastAPI()

@app.post("/download/czjl")
async def 下载处置记录(
    md: Annotated[Model | None, Query(description="下载处置记录")]):
    try:
        if not md:
            return Result(retcode=1, text="处置记录不能为空")
        id_ = IdGenerator.gen()
        document = Document('template.docx')
        document.add_heading('处方笺', 0)
        document.add_heading(
            '姓名: _____ 性别: _____ 年龄: _____ ',
            level=1)
        document.add_heading('处置方案', level=1)
        document.add_paragraph(md['czfa'])
        document.add_heading('诊断', level=1)
        document.add_paragraph(md['zd'])
        document.add_heading('医师签字: _____', level=1)
        document.add_heading('处置日期: ____年____月____日', level=1)
        document.add_page_break()
        document.save(f'{id_}.docx')
        return FileResponse(path=f'{id_}.docx')
    except APIException as e:
        return JSONResponse(
            status_code=e.status,
            content={"retcode": e.retcode, "text": e.text, "result": e.result},
        )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)

```

5.3.4 治疗方案

治疗方案通常包含药物治疗、手术治疗、物理治疗及生活方式调整等多种组合。

药物治疗是最常见的疾病干预手段。医师会根据病灶部位、病理类型、严重程度、病程阶段等因素开适宜的药物,这些药物通过不同的作用机制发挥治疗效果,如抑制病原体生长、调节人体生物化学反应、增强免疫功能等。医师会根据患者的具体情况精准设计药物种类、剂量、用法用量,进行动态调整,以达到治疗的最佳效果。

对于严重疾病如肿瘤或器官衰竭,手术治疗往往是必要且关键的一环。手术可以切除恶性肿瘤、修复损伤组织、重建功能等。成功的手术需要医师团队的精心设计、充分准备和紧密配合。相对于门诊,手术更多在住院治疗中应用。在 HIS 智慧医疗系统中,不同科室的医师可以通过网络协作平台进行病例讨论,以及手术方案设计;手术过程中可以采集各项生命体征并进行监测;手术结束后通过总结分析继续优化流程,以提高手术的精准性和效果。

物理治疗利用推拿、针灸、理疗等非药物手段,帮助调节人体生理功能。这些治疗通过刺激穴位、增加血液循环、放松肌肉等方式起到止痛、活血、增强机体抵抗力的作用。在 HIS 智慧医疗系统中,医师可以开物理治疗处方,患者到相应的治疗室进行推拿、热疗等治疗,过程数据可以被录入 HIS 智慧医疗系统。医师可以根据过程数据评估治疗效果并调整方案。

此外,医师还会根据患者的生活方式与疾病的关系,为患者制定饮食、运动等生活调整方案。这些调整有助于减轻症状、延缓病程进展、提高患者的生活质量。医师可以在 HIS 智慧医疗系统中记录患者生活方式评估和建议的改变措施,以便患者在生活中做出调整。这需要医患双方的紧密配合。可以看出,面对复杂的疾病情况,医师需要全面评估患者病情,制定系统、精准、个性化的治疗方案,其中融合药物治疗、手术治疗、物理治疗及生活方式调整等多种治疗手段,相互配合,以达到最佳的治疗效果。在 HIS 智慧医疗系统的辅助下,这些治疗过程被有效记录、优化,使疾病治疗更加精准高效。

总之,治疗方案是一种针对疾病或健康问题制定的综合治疗措施,旨在帮助患者恢复健康或减轻病痛。患者在接受治疗时应积极配合医师,并配合医师定期进行治疗效果评估和治疗方案调整。对于某些病情较为严重的患者,普通的药物治疗及物理治疗很难起到明显的效果,这时就需要进行门诊输液治疗或住院治疗。对于需要输液治疗的病历,医师需要在 HIS 智慧医疗系统中进行特殊注明,例如哪几样药是输液注射的。对于需要住院治疗的病历,医师需要在 HIS 智慧医疗系统中标注出住院治疗。

5.3.5 治疗方案模板

治疗方案模板弹窗如图 5-19 所示。

治疗方案模板弹窗的代码如下:



图 5-19 治疗方案模板弹窗

```
<!-- # 第 5 章/治疗方案模板.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="智慧医疗" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="挂号" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="病历" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="检查" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="治疗" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="治疗" @click="goTo('http://cnoctave.cn')">
</q-btn>
        <q-separator dark vertical></q-separator>
        <q-space></q-space>
        <q-separator dark vertical></q-separator>
        <q-btn stretch flat label="当前用户: 张三(管理员)" @click="
"goTo('http://cnoctave.cn')"></q-btn>
      </q-toolbar>
    </q-header>
    <q-page-container>
```

```

        <div class = "q - ma - md">
            <div class = "text - h6">治疗方案模板</div>
            <q - btn color = "green" icon - right = "south" label = "下载记录" no - caps
@click = "add()"></q - btn>
            <q - btn color = "red" icon - right = "close" label = "清空" no - caps @click =
"cancel()"></q - btn>
            <div> &nbsp;</div>
            <q - expansion - item expand - separator icon = "edit" label = "药物治疗"
class = "bg - grey - 3">
                <q - card>
                    <q - card - section>
                        <q - input :dense = "true" outlined v - model = "md. ywzl" :
autogrow = "true"></q - input>
                    </q - card - section>
                </q - card>
            </q - expansion - item>
            <div> &nbsp;</div>
            <q - expansion - item expand - separator icon = "edit" label = "手术治疗"
class = "bg - grey - 3">
                <q - card>
                    <q - card - section>
                        <q - input :dense = "true" outlined v - model = "md. sszl" :
autogrow = "true"></q - input>
                    </q - card - section>
                </q - card>
            </q - expansion - item>
            <div> &nbsp;</div>
            <q - expansion - item expand - separator icon = "edit" label = "物理治疗"
class = "bg - grey - 3">
                <q - card>
                    <q - card - section>
                        <q - input :dense = "true" outlined v - model = "md. wlzl" :
autogrow = "true"></q - input>
                    </q - card - section>
                </q - card>
            </q - expansion - item>
            <div> &nbsp;</div>
            <q - expansion - item expand - separator icon = "edit" label = "生活方式调整"
class = "bg - grey - 3">
                <q - card>
                    <q - card - section>
                        <q - input :dense = "true" outlined v - model = "md. shfstz" :
autogrow = "true"></q - input>
                    </q - card - section>
                </q - card>
            </q - expansion - item>
            <div> &nbsp;</div>
            <q - expansion - item expand - separator icon = "edit" label = "其他治疗"
class = "bg - grey - 3">
                <q - card>
                    <q - card - section>
                        <q - input :dense = "true" outlined v - model = "md. qtzl" :
autogrow = "true"></q - input>
                    </q - card - section>

```

```

        </q-card>
      </q-expansion-item>
    <div> &nbsp;</div>
    <q-expansion-item expand-separator icon="edit" label="诊断" class =
"bg-grey-3">
      <q-card>
        <q-card-section>
          <q-input :dense="true" outlined v-model="md.zd" :autogrow =
"true"></q-input>
        </q-card-section>
      </q-card>
    </q-expansion-item>
  <div> &nbsp;</div>
</div>
</q-page-container>
</q-layout>
</div>

```

5.3.6 下载治疗记录处方笺

医师可以单击“下载记录”按钮下载 Word 格式的治疗记录处方笺。治疗记录处方笺如图 5-20 所示。

处方笺

*姓名: _____ 性别: _____ 年龄: _____

*
药物治疗
阿司匹林: 每日三次, 每次一片, 餐后服用

*
诊断
高血压

*医师签字: _____

*
治疗日期: ____年__月__日

图 5-20 治疗记录处方笺

治疗记录 Model 的代码如下:

```

# 第 5 章/治疗记录 Model.py
from typing import Any
from pydantic import BaseModel as PydanticBaseModel

class BaseModel(PydanticBaseModel):
    class Config:
        arbitrary_types_allowed = True

class Model(BaseModel):
    ywzl: str | None = None
    sszl: str | None = None

```

```
wlzl: str | None = None
shfstz: str | None = None
qtzl: str | None = None
zd: str | None = None
```

下载治疗记录 API 的代码如下：

```
# 第 5 章/下载治疗记录.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from 治疗记录 Model import Model
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
import datetime
from starlette.responses import JSONResponse, FileResponse
from docx import Document
from docx.shared import Inches

engine = create_engine("替换为数据库连接字符串")
DBSession = sessionmaker(bind=engine)
session = DBSession()

app = FastAPI()

@app.post("/download/zljl")
async def 下载治疗记录(
    md: Annotated[Model | None, Query(description="下载治疗记录")]):
    try:
        if not md:
            return Result(retcode=1, text="治疗记录不能为空")
        id_ = IdGenerator.gen()
        document = Document('template.docx')
        document.add_heading('处方笺', 0)
        document.add_heading(
            '姓名: _____ 性别: _____ 年龄: _____ ',
            level=1)
        if md['ywzl']:
            document.add_heading('药物治疗', level=1)
            document.add_paragraph(md['ywzl'])
        if md['sszl']:
            document.add_heading('手术治疗', level=1)
            document.add_paragraph(md['sszl'])
        if md['wlzl']:
            document.add_heading('物理治疗', level=1)
            document.add_paragraph(md['wlzl'])
        if md['shfstz']:
            document.add_heading('生活方式调整', level=1)
            document.add_paragraph(md['shfstz'])
        if md['qtzl']:
            document.add_heading('其他治疗', level=1)
```

```
        document.add_paragraph(md[ 'qtzl'])
        document.add_heading('诊断', level = 1)
        document.add_paragraph(md[ 'zd'])
        document.add_heading('医师签字: _____', level = 1)
        document.add_heading('治疗日期: ___年___月___日', level = 1)
        document.add_page_break()
        document.save(f'{id_}.docx')
        return FileResponse(path = f'{id_}.docx')
    except APIException as e:
        return JSONResponse(
            status_code = e.status,
            content = {"retcode": e.retcode, "text": e.text, "result": e.result},
        )

if __name__ == "__main__":
    uvicorn.run(app, host = "127.0.0.1", port = 8000)
```