<template>

<el-container>

<el-aside width="400px">

<el-card

class="box-card"

shadow="always"

body-style="{width: '100%',height: '100%'}"

>

<div slot="header" class="aside-header">

<span>今日就诊</span>

</div>

<el-row>

<el-input

placeholder="请输入患者姓名/联系方式/身份证号"

v-model="SearchPatientInfo"

class="input-with-select"

>

<!-- <el-select v-model="select" slot="prepend" placeholder="请选择搜索方式" style="width:100px;">

<el-option label="患者姓名" value="1"></el-option>

<el-option label="联系方式" value="2"></el-option>

<el-option label="身份证号" value="3"></el-option>

<el-option label="健康卡号" value="4"></el-option>

</el-select> -->

<el-button

style="width: 100px"

slot="append"

icon="el-icon-search"

@click="handlePatientClick"

>搜索</el-button

>

</el-input>

</el-row>

<!-- <el-row>

<el-button style="width:100%;">快速接诊</el-button>

</el-row> -->

<el-row>

<el-tabs

v-model="TodayActiveName"

@tab-click="handlePatientClick"

stretch="true"

>

<el-tab-pane name="prepare">

<span slot="label">

待就诊

<el-badge

:value="PreparePatientTotal"

:max="99"

class="mark"

style="line-height: normal"

></el-badge>

</span>

<el-scrollbar

view-style="height:calc(100vh - 308px);padding-top: 15px;"

>

<div

v-for="(item, index) in PreparePatientList"

:key="item"

class="text item"

@click="clickPreparePatient(item)"

>

<el-descriptions :column="2" :colon="false">

<el-descriptions-item

labelStyle="margin-right: 0px;"

contentStyle="font-weight: bold;"

>{{ index + 1 }}. {{ item.patientId.name }} /

{{ item.patientId.gender.name }} /

{{ item.patientId.birthdayText }}岁</el-descriptions-item

>

<el-descriptions-item></el-descriptions-item>

<el-descriptions-item labelStyle="margin-right: 0px;"

>{{ item.doctor.name }} /

<span style="color: #018cb7">{{

item.source.name

}}</span></el-descriptions-item

>

<el-descriptions-item contentStyle="margin-left: 90px;">{{

item.formatUpdateDate

}}</el-descriptions-item>

</el-descriptions>

<el-divider></el-divider>

</div>

</el-scrollbar>

<el-pagination

small

background

@current-change="handleCurrentPreparePatientChange"

:page-size="SearchPreModel.limit"

layout="prev, pager, next"

:total="PreparePatientTotal"

>

</el-pagination>

</el-tab-pane>

<el-tab-pane name="already">

<span slot="label">

已就诊

<el-badge

:value="AlreadyPatientListTotal"

:max="99"

type="primary"

style="line-height: normal"

></el-badge>

</span>

<el-scrollbar view-style="height:calc(100vh - 293px);">

<el-collapse

v-model="activeSelAlreadyPatientItem"

accordion

@change="AlreadyPatientDeletilQuery()"

>

<div v-for="(item, index) in AlreadyPatientList" :key="item">

<el-collapse-item class="already-medocal" :name="item">

<template slot="title">

<span

style="

font-weight: bold;

font-size: 12px;

color: #606266;

flex: 1;

"

>{{ index + 1 }}. {{ item.patientId.name }} /

{{ item.patientId.gender.name }} /

{{ item.patientId.birthdayText }}岁</span

>

<span style="font-size: 12px; color: #606266">{{

item.formatReceptionEndDate

}}</span>

</template>

<div

v-for="o in AlreadyPatientDescriptions"

:key="o"

@click="AlreadyPatientDescriptionsQuery(o)"

>

<el-descriptions

:column="1"

:colon="false"

style="

border-top: 1px dashed #dcdfe6;

margin: 0px 30px 0 15px;

"

>

<el-descriptions-item

contentStyle="font-weight: bold;margin-top: 6px;"

>{{ o.name }}</el-descriptions-item

>

<el-descriptions-item

>{{ o.code }}

<span style="position: relative">

<div v-if="o.status === -1" class="refundBox">

已作废

</div>

<div

v-if="o.status === 1 && o.chargeStatus === 0"

class="refundBox"

>

未收费

</div>

<div

v-if="o.status === 1 && o.chargeStatus === 1"

class="refundBox"

>

已收费

</div>

<div

v-if="o.status === 1 && o.chargeStatus === -1"

class="refundBox"

>

已退费

</div>

</span>

</el-descriptions-item>

</el-descriptions>

</div>

</el-collapse-item>

</div>

</el-collapse>

</el-scrollbar>

<el-pagination

small

background

@current-change="handleCurrentAlreadyPatientChange"

:page-size="SearchAlreadyModel.limit"

layout="prev, pager, next"

:total="AlreadyPatientListTotal"

>

</el-pagination>

</el-tab-pane>

</el-tabs>

</el-row>

</el-card>

</el-aside>

<el-main>

<el-card class="box-card" shadow="always">

<el-scrollbar

view-style="height:calc(100vh - 210px);padding-right: 20px;"

>

<el-card class="box-card main-card">

<div

slot="header"

style="padding: 5px 0px; font-size: 14px; font-weight: bold"

>

<span>基础信息</span>

<!-- <el-button v-if="isReadOnly==false" style="float: right; padding: 3px 0" type="text">清空</el-button> -->

</div>

<el-form

:model="BasicInfoModel"

ref="BasicInfoForm"

label-width="auto"

:rules="BasicInfoFormRules"

>

<el-row :gutter="24">

<el-col :span="8">

<el-form-item label="患者姓名" prop="name">

<el-input

:disabled="isBaseReadOnly"

v-model="BasicInfoModel.name"

></el-input>

</el-form-item>

</el-col>

<el-col :span="8">

<el-form-item label="性别" prop="gender">

<el-select

:disabled="isBaseReadOnly"

v-model="BasicInfoModel.gender.name"

placeholder="请选择性别"

style="width: 100%"

>

<el-option label="男" value="男"></el-option>

<el-option label="女" value="女"></el-option>

</el-select>

</el-form-item>

</el-col>

<el-col :span="8">

<el-form-item label="年龄" prop="name">

<el-input

:disabled="isBaseReadOnly"

placeholder="请输入"

v-model="BasicInfoModel.age"

style="width: 49%"

>

<template slot="append">岁</template>

</el-input>

<el-input

:disabled="isBaseReadOnly"

placeholder="请输入"

v-model="BasicInfoModel.month"

style="width: 49%"

>

<template slot="append">月</template>

</el-input>

</el-form-item>

</el-col>

</el-row>

<el-row :gutter="24">

<el-col :span="6">

<el-form-item label="体温" prop="temperature">

<div class="ipt-box">

<el-input

:maxlength="45"

:disabled="isBaseReadOnly"

v-model="BasicInfoModel.temperature"

></el-input>

<span>&nbsp;℃</span>

</div>

</el-form-item>

</el-col>

<el-col :span="6">

<el-form-item label="脉搏" prop="pulse">

<div class="ipt-box">

<el-input

:maxlength="45"

:disabled="isBaseReadOnly"

v-model="BasicInfoModel.pulse"

></el-input>

<span>&nbsp;次/min</span>

</div>

</el-form-item>

</el-col>

<el-col :span="6">

<el-form-item label="呼吸" prop="breathe">

<div class="ipt-box">

<el-input

:maxlength="45"

:disabled="isBaseReadOnly"

v-model="BasicInfoModel.breathe"

></el-input>

<span>&nbsp;次/min</span>

</div>

</el-form-item>

</el-col>

<el-col :span="6">

<el-form-item label="血压" prop="bloodPressure">

<div class="ipt-box">

<el-input

:maxlength="45"

:disabled="isBaseReadOnly"

v-model="BasicInfoModel.bloodPressure"

></el-input>

<span>&nbsp;mmHg</span>

</div>

</el-form-item>

</el-col>

</el-row>

<el-row :gutter="24">

<el-col :span="8">

<el-form-item label="联系方式" prop="phone">

<el-input

:disabled="isBaseReadOnly"

v-model="BasicInfoModel.phone"

></el-input>

</el-form-item>

</el-col>

<el-col :span="8">

<el-form-item label="身份证号" prop="card">

<el-input

:disabled="isBaseReadOnly"

v-model="BasicInfoModel.card"

></el-input>

</el-form-item>

</el-col>

<el-col :span="8">

<!-- <el-form-item label="治疗类型" prop="treatType">

<el-radio-group v-model="MedicalRecordModel.registration.treatType">

<el-radio :disabled="isReadOnly==true" v-for="typeItem in TreatTypeOption" :key="typeItem" :label="typeItem.name"></el-radio>

</el-radio-group>

</el-form-item> -->

<el-form-item label="治疗类型">

<el-radio

:disabled="isReadOnly"

v-model="registration.treatType.value"

label="treatType\_0"

>初诊</el-radio

>

<el-radio

:disabled="isReadOnly"

v-model="registration.treatType.value"

label="treatType\_1"

>复诊</el-radio

>

</el-form-item>

</el-col>

</el-row>

</el-form>

</el-card>

<el-card class="box-card main-card">

<div

slot="header"

style="padding: 5px 0px; font-size: 14px; font-weight: bold"

>

<span>病历</span>

<div style="float: right; margin-top: -3px; display: flex">

<!-- <el-upload

class="upload-demo"

ref="upload"

action="https://jsonplaceholder.typicode.com/posts/"

:on-preview="handleFilePreview"

:on-remove="handleFileRemove"

:on-progress="handleFileProgress"

:on-change="handleFileBeforeUpload"

:file-list="fileList"

:show-file-list="true"

:auto-upload="false">

<el-button type="text" slot="trigger"><i class="el-icon-upload2 el-icon--right"></i>上传附件</el-button>

</el-upload> -->

<!-- <el-button type="text"><i class="el-icon-upload2 el-icon--right"></i>上传附件</el-button> -->

<div>

<!-- <el-button type="text"><i class="el-icon-document el-icon--right"></i>模板调用</el-button>

<el-button type="text"><i class="el-icon-document-add el-icon--right"></i>存为模板</el-button> -->

<el-popover

placement="bottom-end"

width="400"

popper-class="medical-edit-popover"

trigger="click"

>

<el-card class="box-card">

<div slot="header" class="clearfix">

<span>病历设置</span>

<el-button

style="float: right; padding: 3px 0"

type="text"

@click="ClearMedicineCofigure()"

>恢复默认</el-button

>

</div>

<div>

<el-row>选择需要的病历字段</el-row>

<el-checkbox-group v-model="ChooseRecordList">

<el-checkbox

:disabled="isReadOnly"

v-for="item in MedicineRecordLabelList"

:label="item"

:key="item"

>{{ item.label }}</el-checkbox

>

</el-checkbox-group>

</div>

</el-card>

<el-button

:disabled="isReadOnly"

slot="reference"

type="text"

><i class="el-icon-setting el-icon--right"></i

>配置</el-button

>

</el-popover>

</div>

</div>

</div>

<el-form

:disabled="isReadOnly"

:model="MedicalRecordModel"

ref="BasicInfoForm"

:rules="BasicInfoFormRules"

label-width="auto"

label-suffix="："

>

<el-row :gutter="24">

<el-col :span="24">

<el-form-item label="主诉" prop="patientTell">

<el-input

v-model="MedicalRecordModel.patientTell"

></el-input>

</el-form-item>

</el-col>

</el-row>

<el-row

:gutter="24"

v-for="(item, index) in ChooseRecordList"

:key="index"

>

<el-col :span="24">

<el-form-item :label="item.label" :prop="item.name">

<el-input

v-model="MedicalRecordModel[`${item.name}`]"

></el-input>

</el-form-item>

</el-col>

</el-row>

<el-row :gutter="24">

<el-col :span="24">

<el-form-item label="西医诊断" prop="westernDiagnose">

<el-input

v-model="MedicalRecordModel.westernDiagnose"

></el-input>

</el-form-item>

</el-col>

</el-row>

<el-row :gutter="24">

<el-col :span="24">

<el-form-item label="中医诊断" prop="chinaDiagnose">

<el-input

v-model="MedicalRecordModel.chinaDiagnose"

></el-input>

</el-form-item>

</el-col>

</el-row>

</el-form>

</el-card>

<el-card class="box-card main-card">

<div

slot="header"

style="padding: 5px 0px; font-size: 14px; font-weight: bold"

>

<span>开具处方</span>

</div>

<div style="min-height: 400px">

<el-tabs

v-model="medicalEditTabsValue"

type="card"

@tab-remove="removeMedicalEditTab"

@tab-click="clickMedicalEditTab"

>

<el-tab-pane

v-for="(item, index) in medicalEditTabs"

:key="item"

:label="item.title"

:name="item"

:closable="item.closable"

>

<span slot="label">

<!-- <el-checkbox :disabled="isReadOnly"></el-checkbox> -->

{{ item.title }}

</span>

<!-- 西药处方 -->

<div

v-if="item.type === 'recipelType\_0'"

style="margin-top: 10px"

>

<el-row>

<el-divider content-position="left">处方信息</el-divider>

<el-select

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

v-model="item.content.recipelInfo.smallType"

placeholder="请选择处方分类"

style="width: 100px"

>

<el-option

v-for="smallTypeItem in RecipelSmallTypeList"

:key="smallTypeItem.value"

:label="smallTypeItem.name"

:value="{

name: smallTypeItem.name,

value: smallTypeItem.value,

}"

>

</el-option>

</el-select>

<el-popover

placement="top-start"

v-if="

!isReadOnly &&

item.content.recipelInfo.chargeStatus == 0 &&

item.content.recipelInfo.status != -1

"

width="700"

trigger="focus"

>

<el-table

:data="WesternMedicineTable"

:height="300"

border

highlight-current-row

@row-click="RowClickWesternTable"

>

<el-table-column

prop="type.name"

label="药品类型"

></el-table-column>

<el-table-column

prop="goodsName"

label="药品名称"

></el-table-column>

<el-table-column prop="gg" label="规格" width="120">

<template slot-scope="scope">

{{ scope.row.dosis

}}{{ scope.row.dosisUnit.name }} \*

{{ scope.row.preparation

}}{{ scope.row.preparationUnit.name }}/{{

scope.row.pack.name

}}

</template>

</el-table-column>

<el-table-column

prop="factory.name"

label="厂家"

width="100"

></el-table-column>

<el-table-column label="包装价" width="80">

<template slot-scope="scope">

{{ scope.row.price + "/" + scope.row.pack.name }}

</template>

</el-table-column>

<el-table-column label="零售价" width="80">

<template slot-scope="scope">

{{

scope.row.isUnpackSell == 1

? scope.row.retailPrice +

"/" +

scope.row.preparationUnit.name

: "--"

}}

</template>

</el-table-column>

<el-table-column

prop="inventory"

label="库存"

width="100"

>

<template slot-scope="scope">

{{

Math.floor(

scope.row.inventory / scope.row.preparation

) > 0

? Math.floor(

scope.row.inventory /

scope.row.preparation

) +

scope.row.pack.name +

(scope.row.inventory %

scope.row.preparation >

0

? (scope.row.inventory %

scope.row.preparation) +

scope.row.preparationUnit.name

: "")

: scope.row.inventory +

scope.row.preparationUnit.name

}}

</template>

</el-table-column>

</el-table>

<el-input

prefix-icon="el-icon-plus"

suffix-icon="el-icon-search"

style="width: 30%"

slot="reference"

ref="WesternInput"

v-model="SearchWesternInput"

@input="GetWesternTable"

@focus="GetWesternTable"

placeholder="输入药品名称或拼音码"

></el-input>

</el-popover>

</el-row>

<el-row>

<el-table

:data="

getDataFilterTable(

item.content.recipelDetailEvtList,

0,

item

)

"

border

style="width: 100%"

class="tableStyle"

@cell-click="checkInventory"

>

<el-table-column

type="index"

label="序号"

width="50"

align="center"

></el-table-column>

<el-table-column

prop="drugStuffId"

label="药品名称"

align="center"

>

<template slot-scope="scope">

{{ scope.row.drugStuffId.name }}

<span style="color: forestgreen">

{{

scope.row.drugStuffId.drug.dosis +

scope.row.drugStuffId.drug.dosisUnit.name +

"\*" +

scope.row.drugStuffId.drug.preparation +

scope.row.drugStuffId.drug.preparationUnit

.name +

"/" +

scope.row.drugStuffId.drug.pack.name

}}

</span>

</template>

</el-table-column>

<el-table-column

prop="singleDosage"

label="单次用量"

align="center"

width="100"

>

<template slot-scope="scope">

<el-input

v-model="scope.row.singleDosage"

oninput="value=value.replace(/[^\d.]/g,'')"

@input="MedicalCalculate()"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

>

<template slot="append">{{

scope.row.drugStuffId.preparationUnit.name

}}</template>

</el-input>

</template>

</el-table-column>

<el-table-column

prop="westernMedicineUse"

label="用法"

align="center"

width="120"

>

<template slot-scope="scope">

<el-select

v-model="scope.row.westernMedicineUse"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

placeholder=""

@change="changeUse(scope.row.westernMedicineUse)"

>

<el-option

v-for="item in WesternUseOption"

:key="item.value"

:label="item.name"

:value="{ name: item.name, value: item.value }"

>

</el-option>

</el-select>

</template>

</el-table-column>

<el-table-column

prop="frequency"

label="频次"

align="center"

width="120"

>

<template slot-scope="scope">

<el-select

v-model="scope.row.frequency"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

@change="MedicalCalculate()"

placeholder=""

>

<el-option

v-for="item in FrequencyOption"

:key="item.value"

:label="item.name"

:value="{ name: item.name, value: item.value }"

>

</el-option>

</el-select>

</template>

</el-table-column>

<el-table-column

prop="days"

label="天数"

align="center"

width="100"

>

<template slot-scope="scope">

<el-select

v-model="scope.row.days"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

@change="MedicalCalculate()"

placeholder=""

>

<el-option

v-for="item in DayNumOption"

:key="item.value"

:label="item.name"

:value="{ name: item.name, value: item.value }"

>

</el-option>

</el-select>

</template>

</el-table-column>

<el-table-column

prop="total"

label="总量"

align="center"

width="100"

>

<template slot-scope="scope">

{{

Math.floor(

scope.row.total /

scope.row.drugStuffId.drug.preparation

) > 0

? Math.floor(

scope.row.total /

scope.row.drugStuffId.drug.preparation

) +

scope.row.drugStuffId.pack.name +

(scope.row.total %

scope.row.drugStuffId.drug.preparation >

0

? (scope.row.total %

scope.row.drugStuffId.drug

.preparation) +

scope.row.drugStuffId.preparationUnit.name

: "")

: scope.row.total +

scope.row.drugStuffId.preparationUnit.name

}}

</template>

</el-table-column>

<el-table-column

prop="isUnpackSell"

label="单价"

width="110"

align="center"

>

<template slot-scope="scope">

<el-select

v-model="scope.row.isUnpackSell"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

placeholder="请选择"

@change="MedicalCalculate()"

style="width: 80px"

>

<el-option

:label="

scope.row.drugStuffId.price +

'/' +

scope.row.drugStuffId.pack.name

"

:value="0"

></el-option>

<el-option

:label="

scope.row.drugStuffId.retailPrice +

'/' +

scope.row.drugStuffId.preparationUnit.name

"

:value="1"

></el-option>

</el-select>

</template>

</el-table-column>

<el-table-column

prop="allFee"

label="金额"

width="100"

align="center"

></el-table-column>

<el-table-column

v-if="

!isReadOnly &&

item.content.recipelInfo.chargeStatus == 0 &&

item.content.recipelInfo.status != -1

"

label="操作"

fixed="right"

align="center"

>

<template slot-scope="scope">

<i

class="el-icon-circle-close"

@click="DeleteMedicalRow(scope.$index, scope.row)"

></i>

<span style="display:none">{{changeData}}</span>

</template>

</el-table-column>

</el-table>

</el-row>

</div>

<!-- 中药处方 -->

<div

v-if="item.type === 'recipelType\_1'"

style="margin-top: 15px"

>

<el-row>

<el-divider content-position="left">处方信息</el-divider>

<el-select

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

v-model="item.content.recipelInfo.smallType"

placeholder="请选择处方分类"

style="width: 100px"

>

<el-option

v-for="smallTypeItem in RecipelSmallTypeList"

:key="smallTypeItem.value"

:label="smallTypeItem.name"

:value="{

name: smallTypeItem.name,

value: smallTypeItem.value,

}"

>

</el-option>

</el-select>

<el-popover

placement="top-start"

v-if="

!isReadOnly &&

item.content.recipelInfo.chargeStatus == 0 &&

item.content.recipelInfo.status != -1

"

width="700"

trigger="focus"

>

<el-table

:data="ChineseMedicineTable"

:height="300"

border

highlight-current-row

@row-click="RowClickChineseTable"

>

<el-table-column

prop="type.name"

label="药品类型"

></el-table-column>

<el-table-column

prop="goodsName"

label="药品名称"

></el-table-column>

<el-table-column prop="gg" label="规格" width="120">

<template slot-scope="scope">

{{ scope.row.dosis

}}{{ scope.row.dosisUnit.name }} \*

{{ scope.row.preparation

}}{{ scope.row.preparationUnit.name }}/{{

scope.row.pack.name

}}

</template>

</el-table-column>

<el-table-column

prop="factory.name"

label="厂家"

width="100"

></el-table-column>

<el-table-column label="包装价" width="80">

<template slot-scope="scope">

{{ scope.row.price + "/" + scope.row.pack.name }}

</template>

</el-table-column>

<!--

<el-table-column label="零售价" width="80">

<template slot-scope="scope">

{{scope.row.isUnpackSell === "1" ? (scope.row.retailPrice + "/" + scope.row.preparationUnit.name) : "--"}}

</template>

</el-table-column>

-->

<el-table-column

prop="inventory"

label="库存"

width="100"

>

<template slot-scope="scope">

{{

Math.floor(

scope.row.inventory / scope.row.preparation

) > 0

? Math.floor(

scope.row.inventory /

scope.row.preparation

) +

scope.row.pack.name +

(scope.row.inventory %

scope.row.preparation >

0

? (scope.row.inventory %

scope.row.preparation) +

scope.row.preparationUnit.name

: "")

: scope.row.inventory +

scope.row.preparationUnit.name

}}

</template>

</el-table-column>

</el-table>

<el-input

prefix-icon="el-icon-plus"

suffix-icon="el-icon-search"

style="width: 30%"

slot="reference"

ref="WesternInput"

v-model="SearchChineseInput"

@input="GetChineseTable"

@focus="GetChineseTable"

placeholder="输入药品名称或拼音码"

></el-input>

</el-popover>

</el-row>

<el-row :gutter="24" style="min-height: 100px">

<el-col

:span="6"

v-for="(citem, index) in getDataFilterTable(

item.content.recipelDetailEvtList,

0,

item

)"

:key="citem"

:offset="0"

>

<el-card

:index="index"

class="box-card chinese-medicine-card"

style="width: 240px; margin-bottom: 10px"

>

<div

slot="header"

class="clearfix"

style="padding: 5px 0px; font-size: 14px"

>

<span>{{ citem.drugStuffId.name }}</span>

<el-button

v-if="

!isReadOnly &&

item.content.recipelInfo.chargeStatus == 0 &&

item.content.recipelInfo.status != -1

"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

@click="DeleteMedicalRow(index, citem,item)"

icon="el-icon-close"

style="float: right; padding: 3px 0"

type="text"

></el-button>

</div>

<div>

<el-input

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

placeholder="数量"

size="mini"

oninput="value=value.replace(/[^\d.]/g,'')"

@input="changeSingleDosage(citem.singleDosage,index,item)"

v-model="citem.singleDosage"

style="width: 80px"

>

<template slot="append">{{

citem.drugStuffId.pack.name

}}</template>

</el-input>

<el-select

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

size="mini"

v-model="citem.chineseMedicineUse"

placeholder=""

style="width: 105px"

>

<el-option

v-for="oitem in ChineseUseTimeOption"

:key="oitem.value"

:label="oitem.name"

:value="{

name: oitem.name,

value: oitem.value,

}"

>

</el-option>

</el-select>

</div>

</el-card>

</el-col>

</el-row>

<el-row>

<el-input

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

v-model="item.content.recipelInfo.dosage"

oninput="value=value.replace(/[^\d.]/g,'')"

@input="MedicalCalculate()"

style="width: 60px"

></el-input

>

&nbsp;剂 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;用法：

<el-select

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

v-model="item.content.recipelInfo.recipelUse"

placeholder="请选择"

style="width: 110px"

>

<el-option

v-for="pitem in ChineseUseOption"

:key="pitem.value"

:label="pitem.name"

:value="{ name: pitem.name, value: pitem.value }"

>

</el-option>

</el-select>

<el-select

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

v-model="item.content.recipelInfo.frequency"

placeholder="请选择"

style="width: 110px"

>

<el-option

v-for="pitem in ChineseTimeOption"

:key="pitem.value"

:label="pitem.name"

:value="{ name: pitem.name, value: pitem.value }"

>

</el-option>

</el-select>

<el-select

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

v-model="item.content.recipelInfo.takeFrequency"

placeholder="请选择"

style="width: 110px"

>

<el-option

v-for="pitem in JSON.parse(

JSON.stringify(FrequencyOption)

)"

:key="pitem.value"

:label="pitem.name"

:value="{ name: pitem.name, value: pitem.value }"

>

</el-option>

</el-select>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; 一次&nbsp;<el-input

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

v-model="item.content.recipelInfo.singleDosage"

style="width: 60px"

></el-input

>&nbsp;<span v-if="isSpecial">格</span><span v-else>ml</span>

<span style="float: right"

>中药金额:{{

item.content.recipelInfo.medicalAmount

? item.content.recipelInfo.medicalAmount

: 0

}}元</span

>

</el-row>

</div>

<!-- 输液处方 -->

<div

v-if="item.type === 'recipelType\_2'"

style="margin-top: 15px"

>

<div >

<el-row>

<el-divider content-position="left">处方信息</el-divider>

<el-select

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

v-model="item.content.recipelInfo.smallType"

placeholder="请选择处方分类"

style="width: 100px"

>

<el-option

v-for="smallTypeItem in RecipelSmallTypeList"

:key="smallTypeItem.value"

:label="smallTypeItem.name"

:value="{

name: smallTypeItem.name,

value: smallTypeItem.value,

}"

>

</el-option>

</el-select>

<el-button

v-if="

!isReadOnly &&

item.content.recipelInfo.chargeStatus == 0 &&

item.content.recipelInfo.status != -1

" @click="addGroup(item)" type="primary" plain>加一组</el-button>

<!-- <button v-if="

!isReadOnly &&

item.content.recipelInfo.chargeStatus == 0 &&

item.content.recipelInfo.status != -1

" @click="minusGroup(item)">减一组</button> -->

</el-row>

</div>

<div style="margin-left:-10px;">

<div v-for="(items,index) in item.infusion.zushu" :key="index" style="border:1px solid #DCDCDC;padding:10px;margin:10px;">

<el-row>

<div>

<el-popover

placement="top-start"

v-if="

!isReadOnly &&

item.content.recipelInfo.chargeStatus == 0 &&

item.content.recipelInfo.status != -1

"

width="700"

trigger="focus"

>

<el-table

:data="InfusionTable"

:height="300"

border

highlight-current-row

@row-click="(row,column,e)=>RowClickInfusionTable(row,column,e,index)"

>

<el-table-column

prop="type.name"

label="药品类型"

></el-table-column>

<el-table-column

prop="goodsName"

label="药品名称"

></el-table-column>

<el-table-column prop="gg" label="规格" width="120">

<template slot-scope="scope">

{{ scope.row.dosis

}}{{ scope.row.dosisUnit.name }} \*

{{ scope.row.preparation

}}{{ scope.row.preparationUnit.name }}/{{

scope.row.pack.name

}}

</template>

</el-table-column>

<el-table-column

prop="factory.name"

label="厂家"

width="100"

></el-table-column>

<el-table-column label="包装价" width="80">

<template slot-scope="scope">

{{ scope.row.price + "/" + scope.row.pack.name }}

</template>

</el-table-column>

<el-table-column label="零售价" width="80">

<template slot-scope="scope">

{{

scope.row.isUnpackSell == 1

? scope.row.retailPrice +

"/" +

scope.row.preparationUnit.name

: "--"

}}

</template>

</el-table-column>

<el-table-column

prop="inventory"

label="库存"

width="100"

>

<template slot-scope="scope">

{{

Math.floor(

scope.row.inventory / scope.row.preparation

) > 0

? Math.floor(

scope.row.inventory /

scope.row.preparation

) +

scope.row.pack.name +

(scope.row.inventory %

scope.row.preparation >

0

? (scope.row.inventory %

scope.row.preparation) +

scope.row.preparationUnit.name

: "")

: scope.row.inventory +

scope.row.preparationUnit.name

}}

</template>

</el-table-column>

</el-table>

<el-input

prefix-icon="el-icon-plus"

suffix-icon="el-icon-search"

style="width: 20%"

slot="reference"

ref="WesternInput"

v-model="SearchInfusion[index]"

@input="GetInfusionTable(index)"

@focus="GetInfusionTable(index)"

placeholder="输入药品名称或拼音码"

></el-input>

</el-popover>

<span >&nbsp;&nbsp;第{{index+1}}组</span>

<span style="float:right;height:25px;line-height:25px;font-size:14px;">&nbsp;&nbsp;</span>

<el-button style="float:right;" type="primary" plain icon="el-icon-delete" circle

v-if="

!isReadOnly &&

item.content.recipelInfo.chargeStatus == 0 &&

item.content.recipelInfo.status != -1

" @click="minusGroup(item,index)"

></el-button>

<span style="float:right;height:25px;line-height:25px;font-size:14px;">&nbsp;&nbsp;</span>

<span style="float:right;height:25px;line-height:25px;font-size:14px;">&nbsp;&nbsp;滴/分钟</span>

<el-input

style="float:right;width:10%"

v-model="item.infusion.drippingSpeed[index]"

placeholder="请选择滴速"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

>

</el-input>

<span style="float:right;height:25px;line-height:25px;font-size:14px;">&nbsp;&nbsp;天&nbsp;&nbsp;</span>

<el-select

style="float:right;width:10%"

v-model="item.infusion.days[index]"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

@change="MedicalCalculate()"

placeholder="请选择天数"

>

<el-option

v-for="item in DayNumOption"

:key="item.value"

:label="item.name"

:value="{ name: item.name, value: item.value }"

>

</el-option>

</el-select>

<span style="float:right;height:25px;line-height:25px;font-size:18px;">&nbsp;&nbsp;</span>

<el-select

style="float:right;width:10%;"

v-model="item.infusion.frequency[index]"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

@change="MedicalCalculate()"

placeholder="请选择频次"

>

<el-option

v-for="item in FrequencyOption"

:key="item.value"

:label="item.name"

:value="{ name: item.name, value: item.value }"

>

</el-option>

</el-select>

<span style="float:right;height:25px;line-height:25px;font-size:18px;">&nbsp;&nbsp;</span>

<el-select

style="float:right;width:10%"

v-model="item.infusion.infuseUse[index]"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

placeholder="请选择用法"

>

<el-option

v-for="item in infuseUseOption"

:key="item.value"

:label="item.name"

:value="{ name: item.name, value: item.value }"

>

</el-option>

</el-select>

</div>

<div style="border:7px solid #0000"></div>

<el-table

:data="

getDataFilterTable(

item.infusion.infusionProject[index],

0,

item

)

"

border

style="width: 100%"

class="tableStyle"

@cell-click="checkInventory"

>

<el-table-column

type="index"

label="序号"

width="50"

align="center"

></el-table-column>

<!-- <el-table-column

v-show="true"

prop="infuseGroup"

label="组数"

width="100"

align="center"

></el-table-column> -->

<el-table-column

prop="drugStuffId"

label="药品名称"

align="center"

>

<template slot-scope="scope">

{{ scope.row.drugStuffId.name }}

<span style="color: forestgreen">

{{

scope.row.drugStuffId.drug.dosis +

scope.row.drugStuffId.drug.dosisUnit.name +

"\*" +

scope.row.drugStuffId.drug.preparation +

scope.row.drugStuffId.drug.preparationUnit

.name +

"/" +

scope.row.drugStuffId.drug.pack.name

}}

</span>

</template>

</el-table-column>

<el-table-column

prop="skinTest"

label="皮试"

align="center"

width="120"

>

<template slot-scope="scope">

<el-select

v-model="scope.row.skinTest"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

placeholder=""

>

<el-option

v-for="item in InfusionOption"

:key="item.value"

:label="item.name"

:value="{ name: item.name, value: item.value }"

>

</el-option>

</el-select>

</template>

</el-table-column>

<el-table-column

prop="singleDosage"

label="单次用量"

align="center"

width="100"

>

<template slot-scope="scope">

<el-input

v-model="scope.row.singleDosage"

oninput="value=value.replace(/[^\d.]/g,'')"

@input="MedicalCalculate()"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

>

<template slot="append">{{

scope.row.drugStuffId.preparationUnit.name

}}</template>

</el-input>

</template>

</el-table-column>

<!-- <el-table-column

prop="drippingSpeed"

label="滴速"

align="center"

width="80"

>

<template slot-scope="scope">

<el-input

v-model="scope.row.drippingSpeed"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

>

</el-input>

<span >

滴/分钟

</span>

</template>

</el-table-column> -->

<!-- <el-table-column

prop="infuseUse"

label="用法"

align="center"

width="120"

>

<template slot-scope="scope">

<el-select

v-model="scope.row.infuseUse"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

placeholder=""

>

<el-option

v-for="item in infuseUseOption"

:key="item.value"

:label="item.name"

:value="{ name: item.name, value: item.value }"

>

</el-option>

</el-select>

</template>

</el-table-column> -->

<!-- <el-table-column

prop="frequency"

label="频次"

align="center"

width="120"

>

<template slot-scope="scope">

<el-select

v-model="scope.row.frequency"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

@change="MedicalCalculate()"

placeholder=""

>

<el-option

v-for="item in FrequencyOption"

:key="item.value"

:label="item.name"

:value="{ name: item.name, value: item.value }"

>

</el-option>

</el-select>

</template>

</el-table-column> -->

<!-- <el-table-column

prop="days"

label="天数"

align="center"

width="100"

>

<template slot-scope="scope">

<el-select

v-model="scope.row.days"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

@change="MedicalCalculate()"

placeholder=""

>

<el-option

v-for="item in DayNumOption"

:key="item.value"

:label="item.name"

:value="{ name: item.name, value: item.value }"

>

</el-option>

</el-select>

</template>

</el-table-column> -->

<el-table-column

prop="total"

label="总量"

align="center"

width="100"

>

<template slot-scope="scope">

<!-- <el-input v-model="scope.row.total" type="number" disabled>

<template slot="append">

{{scope.row.isUnpackSell === '1' ? scope.row.drugStuffId.preparationUnit.name : scope.row.drugStuffId.pack.name}}

</template>

</el-input> -->

{{

Math.floor(

scope.row.total /

scope.row.drugStuffId.drug.preparation

) > 0

? Math.floor(

scope.row.total /

scope.row.drugStuffId.drug.preparation

) +

scope.row.drugStuffId.pack.name +

(scope.row.total %

scope.row.drugStuffId.drug.preparation >

0

? (scope.row.total %

scope.row.drugStuffId.drug

.preparation) +

scope.row.drugStuffId.preparationUnit.name

: "")

: scope.row.total +

scope.row.drugStuffId.preparationUnit.name

}}

</template>

</el-table-column>

<el-table-column

prop="isUnpackSell"

label="单价"

width="110"

align="center"

>

<template slot-scope="scope">

<el-select

v-model="scope.row.isUnpackSell"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

placeholder="请选择"

@change="MedicalCalculate()"

style="width: 80px"

>

<el-option

:label="

scope.row.drugStuffId.price +

'/' +

scope.row.drugStuffId.pack.name

"

:value="0"

></el-option>

<el-option

:label="

scope.row.drugStuffId.retailPrice +

'/' +

scope.row.drugStuffId.preparationUnit.name

"

:value="1"

></el-option>

</el-select>

</template>

</el-table-column>

<el-table-column

prop="allFee"

label="金额"

width="100"

align="center"

></el-table-column>

<el-table-column

v-if="

!isReadOnly &&

item.content.recipelInfo.chargeStatus == 0 &&

item.content.recipelInfo.status != -1

"

label="操作"

fixed="right"

align="center"

width="100"

>

<template slot-scope="scope">

<i

class="el-icon-circle-close"

@click="DeleteMedicalRow(scope.$index, scope.row)"

></i>

</template>

</el-table-column>

</el-table>

</el-row>

</div>

</div>

</div>

<!-- 诊疗项目 -->

<div

v-if="item.type === 'recipelType\_3'"

style="margin-top: 15px"

>

<el-row>

<el-divider content-position="left">处方信息</el-divider>

<el-popover

placement="top-start"

v-if="

!isReadOnly &&

item.content.recipelInfo.chargeStatus == 0 &&

item.content.recipelInfo.status != -1

"

width="540"

trigger="focus"

>

<el-table

:data="TreatmentTable"

:height="300"

border

highlight-current-row

@row-click="RowClickTreatmentTable"

>

<el-table-column

prop="itemName"

label="项目名称"

width="200"

></el-table-column>

<el-table-column

prop="itemType.name"

label="项目类型"

width="100"

></el-table-column>

<el-table-column

prop="isPackage"

label="套餐项目"

width="100"

>

<template slot-scope="scope">

{{ scope.row.isPackage === "1" ? "是" : "否" }}

</template>

</el-table-column>

<el-table-column

prop="salePrice"

label="单价"

width="100"

>

<template slot-scope="scope">

{{ scope.row.salePrice }}/{{

scope.row.unit.name

}}

</template>

</el-table-column>

</el-table>

<el-input

prefix-icon="el-icon-plus"

suffix-icon="el-icon-search"

style="width: 30%"

slot="reference"

ref="TreatmentInput"

v-model="SearchCostItemInput"

@input="GetCostItemTable"

@focus="GetCostItemTable"

placeholder="输入诊疗项目名称或拼音码"

></el-input>

</el-popover>

</el-row>

<el-row>

<el-table

:data="

getDataFilterTable(

item.content.recipelDetailEvtList,

0,

item

)

"

border

style="width: 100%"

class="tableStyle"

>

<el-table-column

type="index"

label="序号"

width="50"

align="center"

></el-table-column>

<el-table-column

prop="drugStuffId.name"

label="项目名称"

align="center"

></el-table-column>

<el-table-column

prop="drugStuffId.costItem.itemType.name"

label="项目类别"

align="center"

width="100"

></el-table-column>

<el-table-column

prop="drugStuffId.costItem.isPackage"

label="套餐项目"

align="center"

width="100"

>

<template slot-scope="scope">

{{

scope.row.drugStuffId.costItem.isPackage === "1"

? "是"

: "否"

}}

</template>

</el-table-column>

<el-table-column

prop="total"

label="数量"

align="center"

width="120"

>

<template slot-scope="scope">

<el-input

v-model="scope.row.total"

oninput="value=value.replace(/[^\d.]/g,'')"

@input="MedicalCalculate()"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

>

<template slot="append">{{

scope.row.drugStuffId.pack.name

}}</template>

</el-input>

</template>

</el-table-column>

<el-table-column

label="单价"

width="110"

align="center"

>

<template slot-scope="scope">

{{ scope.row.drugStuffId.price }}/{{

scope.row.drugStuffId.pack.name

}}

</template>

</el-table-column>

<el-table-column

prop="allFee"

label="合计金额"

width="120"

align="center"

></el-table-column>

<el-table-column

v-if="

!isReadOnly &&

item.content.recipelInfo.chargeStatus == 0 &&

item.content.recipelInfo.status != -1

"

label="操作"

fixed="right"

align="center"

width="80"

>

<template slot-scope="scope">

<i

class="el-icon-circle-close"

@click="DeleteMedicalRow(scope.$index, scope.row)"

></i>

</template>

</el-table-column>

</el-table>

</el-row>

</div>

<div style="margin-top: 10px">

<el-row>

<el-divider content-position="left">附加费</el-divider>

<el-popover

placement="top-start"

width="550"

trigger="click"

v-if="

!isReadOnly &&

item.content.recipelInfo.chargeStatus == 0 &&

item.content.recipelInfo.status != -1

"

>

<el-table

:data="SurchargeTable"

:max-height="300"

border

highlight-current-row

@row-click="RowClickSurchargeTable"

>

<el-table-column label="费用名称">

<template slot-scope="scope">

{{

scope.row.stuffType === "4"

? scope.row.name

: scope.row.itemName

}}

</template>

</el-table-column>

<el-table-column label="费用类型" width="130">

<template slot-scope="scope">

{{

scope.row.stuffType === "4"

? scope.row.type.name

: "诊疗（其它费用）"

}}

</template>

</el-table-column>

<el-table-column label="销售价" width="100">

<template slot-scope="scope">

{{

scope.row.stuffType === "4"

? scope.row.priceOutSell +

"/" +

scope.row.packUnit.name

: scope.row.salePrice +

"/" +

scope.row.unit.name

}}

</template>

</el-table-column>

<el-table-column label="零售价" width="100">

<template slot-scope="scope">

{{

scope.row.stuffType === "4" &&

scope.row.isUnpackSell == "1"

? scope.row.retailPrice +

"/" +

scope.row.minUnit.name

: "--"

}}

</template>

</el-table-column>

<el-table-column label="库存" width="100">

<template slot-scope="scope">

{{

scope.row.stuffType === "4"

? Math.floor(

scope.row.inventory / scope.row.packNumber

) > 0

? Math.floor(

scope.row.inventory /

scope.row.packNumber

) +

scope.row.packUnit.name +

(scope.row.inventory %

scope.row.packNumber >

0

? (scope.row.inventory %

scope.row.packNumber) +

scope.row.minUnit.name

: "")

: scope.row.inventory +

scope.row.minUnit.name

: "不限制"

}}

</template>

</el-table-column>

</el-table>

<el-input

prefix-icon="el-icon-plus"

suffix-icon="el-icon-search"

style="width: 30%"

slot="reference"

v-model="SearchSurchargeInput"

@input="GetSurchargeTable"

@focus="GetSurchargeTable"

placeholder="输入附加费名称或拼音码"

>

</el-input>

</el-popover>

</el-row>

<div v-if="item.type!='recipelType\_2'">

<el-row>

<el-table

:data="

getDataFilterTable(

item.content.recipelDetailEvtList,

1,

item

)

"

border

style="width: 100%"

class="tableStyle"

>

<el-table-column

type="index"

label="序号"

width="50"

align="center"

></el-table-column>

<el-table-column

prop="drugStuffId.name"

label="费用名称"

width="200"

align="center"

></el-table-column>

<el-table-column

label="费用类型"

width="130"

align="center"

>

<template slot-scope="scope">

{{

scope.row.stuffType === "4"

? scope.row.drugStuffId.stuff.type.name

: "诊疗（其它费用）"

}}

</template>

</el-table-column>

<el-table-column

prop="price"

label="单价"

width="100"

align="center"

>

<template slot-scope="scope">

<span v-if="scope.row.stuffType === '3'">

{{

scope.row.drugStuffId.price +

"/" +

scope.row.drugStuffId.pack.name

}}

</span>

<span v-if="scope.row.stuffType === '4'">

{{

scope.row.isUnpackSell == 0

? scope.row.drugStuffId.price +

"/" +

scope.row.drugStuffId.pack.name

: scope.row.drugStuffId.retailPrice +

"/" +

scope.row.drugStuffId.preparationUnit.name

}}

</span>

</template>

</el-table-column>

<el-table-column

label="数量"

align="center"

width="200"

>

<template slot-scope="scope">

<el-input

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

v-model="scope.row.singleDosage"

@input="MedicalCalculate()"

style="width: 60px"

></el-input>

<el-select

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

v-model="scope.row.isUnpackSell"

@change="MedicalCalculate()"

placeholder="请选择"

style="width: 80px"

>

<el-option

:label="scope.row.drugStuffId.pack.name"

:value="0"

></el-option>

<el-option

v-if="scope.row.drugStuffId.isUnpackSell == 1"

:label="

scope.row.drugStuffId.preparationUnit.name

"

:value="1"

></el-option>

</el-select>

</template>

</el-table-column>

<el-table-column

prop="allFee"

label="金额"

align="center"

></el-table-column>

<el-table-column

v-if="

!isReadOnly &&

item.content.recipelInfo.chargeStatus == 0 &&

item.content.recipelInfo.status != -1

"

label="操作"

fixed="right"

width="100"

align="center"

>

<template slot-scope="scope">

<i

class="el-icon-circle-close"

@click="DeleteMedicalRow(scope.$index, scope.row)"

></i>

</template>

</el-table-column>

</el-table>

</el-row>

</div>

<div v-else>

<el-row>

<el-table

:data="

getDataFilterTable(

item.infusion.excharge,

1,

item

)

"

border

style="width: 100%"

class="tableStyle"

>

<el-table-column

type="index"

label="序号"

width="50"

align="center"

></el-table-column>

<el-table-column

prop="drugStuffId.name"

label="费用名称"

width="200"

align="center"

></el-table-column>

<el-table-column

label="费用类型"

width="130"

align="center"

>

<template slot-scope="scope">

{{

scope.row.stuffType === "4"

? scope.row.drugStuffId.stuff.type.name

: "诊疗（其它费用）"

}}

</template>

</el-table-column>

<el-table-column

prop="price"

label="单价"

width="100"

align="center"

>

<template slot-scope="scope">

<span v-if="scope.row.stuffType === '3'">

{{

scope.row.drugStuffId.price +

"/" +

scope.row.drugStuffId.pack.name

}}

</span>

<span v-if="scope.row.stuffType === '4'">

{{

scope.row.isUnpackSell == 0

? scope.row.drugStuffId.price +

"/" +

scope.row.drugStuffId.pack.name

: scope.row.drugStuffId.retailPrice +

"/" +

scope.row.drugStuffId.preparationUnit.name

}}

</span>

</template>

</el-table-column>

<el-table-column

label="数量"

align="center"

width="200"

>

<template slot-scope="scope">

<el-input

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

v-model="scope.row.singleDosage"

@input="MedicalCalculate()"

style="width: 60px"

></el-input>

<el-select

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

v-model="scope.row.isUnpackSell"

@change="MedicalCalculate()"

placeholder="请选择"

style="width: 80px"

>

<el-option

:label="scope.row.drugStuffId.pack.name"

:value="0"

></el-option>

<el-option

v-if="scope.row.drugStuffId.isUnpackSell == 1"

:label="

scope.row.drugStuffId.preparationUnit.name

"

:value="1"

></el-option>

</el-select>

</template>

</el-table-column>

<el-table-column

prop="allFee"

label="金额"

align="center"

></el-table-column>

<el-table-column

v-if="

!isReadOnly &&

item.content.recipelInfo.chargeStatus == 0 &&

item.content.recipelInfo.status != -1

"

label="操作"

fixed="right"

width="100"

align="center"

>

<template slot-scope="scope">

<i

class="el-icon-circle-close"

@click="DeleteExMedicalRow(scope.$index, scope.row)"

></i>

</template>

</el-table-column>

</el-table>

</el-row>

</div>

<el-row style="margin-top:10px">

<el-divider content-position="left">医嘱事项</el-divider>

<el-input

type="textarea"

:rows="4"

placeholder="请输入医嘱事项"

v-model="item.content.recipelInfo.entrust"

:disabled="

isReadOnly ||

item.content.recipelInfo.chargeStatus != 0 ||

item.content.recipelInfo.status == -1

"

>

</el-input>

</el-row>

</div>

</el-tab-pane>

<el-tab-pane

:closable="false"

v-if="!isReadOnly"

disabled

key="add"

name="add"

>

<span slot="label">

<el-popover

placement="bottom-start"

popper-class="medical-type-popover"

trigger="click"

>

<ul class="medical-type-ul">

<li

v-for="(item, index) in medicalTypeList"

:key="item"

>

<el-button

@click="addMedicalEditTab(item)"

plain

size="mini"

style="border: none"

>{{ item.name }}</el-button

>

</li>

</ul>

<el-button type="text" slot="reference"

><i class="el-icon-plus el-icon--right"></i

>开处方</el-button

>

</el-popover>

</span>

</el-tab-pane>

</el-tabs>

</div>

</el-card>

<div style="height: 10px"></div>

</el-scrollbar>

</el-card>

<el-card class="box-card sub-card" style="height: 50px">

<span style="line-height: 27px">总金额：{{ payAmount }}元</span>

<div style="float: right">

<el-button

type="primary"

v-if="

medicalEditTabsValue.content &&

medicalEditTabsValue.content.recipelInfo.status == 1 &&

medicalEditTabsValue.content.recipelInfo.chargeStatus == 0 &&

TodayActiveName == 'already'

"

@click="invalidMedical"

>作废</el-button

>

<el-button

:type="timeOut?'info':'primary'"

v-if="isReadOnly && TodayActiveName == 'already'"

@click="edit"

>编辑</el-button

>

<el-button

type="primary"

v-if="!isReadOnly && TodayActiveName == 'already'"

@click="save"

>保存</el-button

>

<el-button type="primary" v-if="hasBtnPrint" @click="print"

>打印</el-button

>

<el-button v-show="hasBtnFinishVisit" type="primary" @click="save"

>完成接诊</el-button

>

</div>

</el-card>

</el-main>

</el-container>

</template>

<script>

import { listDictItemAll } from "@/api/sys/dictItem";

import { listAllStock } from "@/api/stock/drug";

import { listCostItemAll } from "@/api/treatment/costItem";

import { listStuffAll } from "@/api/stock/stuff";

import {

listRegistrationPage,

getRegistrationById,

updateStatus,

} from "@/api/outpatient/registration";

import {

allSaveMedicalRecord,

allQueryMedicalRecord,

ureport,

editSave,

} from "@/api/outpatient/medicalRecord";

import {

listRecipelInfoAll,

invalidStatus,

} from "@/api/outpatient/recipelInfo";

import { getPatientById } from "@/api/outpatient/patient";

import NP from "number-precision";

import { BigNumber } from "bignumber.js";

import moment from "moment";

export default {

name: "index",

data() {

return {

registration: {

treatType:{

}

},

savelock:false,

timeOut:false,

medicalEditTabsValue: {},

medicalClickTabsValue: {},

medicalTypeList: [

{

name: "西药处方",

type: "recipelType\_0",

},

{

name: "中药处方",

type: "recipelType\_1",

},

{

name: "输液处方",

type: "recipelType\_2"

},

{

name: "诊疗项目",

type: "recipelType\_3",

},

],

//是否快速接诊

isInstant: 0,

ChineseButton: false,

//tabs默认状态 prepare:待就诊 already:已就诊

TodayActiveName: "prepare",

//今日待就诊患者列表

PreparePatientList: [],

//今日待就诊患者总数

PreparePatientTotal: 0,

//今日已就诊患者列表

AlreadyPatientList: [],

//今日已就诊患者总数

AlreadyPatientListTotal: 0,

//搜索患者信息

SearchPatientInfo: "",

//就诊搜索条件

SearchPreModel: {

columnName: "",

limit: 20,

offset: 0,

order: "",

},

SearchAlreadyModel: {

columnName: "reception\_end\_date",

limit: 20,

offset: 0,

order: "desc",

},

activeSelAlreadyPatientItem: "",

AlreadyPatientDescriptions: [],

//处方小类选项

RecipelSmallTypeList: [],

//中药用法下拉列表

ChineseUseOption: [],

ChineseTimeOption: [],

ChineseUseTimeOption: [],

//西药频度下拉列表

FrequencyOption: [],

//西药用法下拉列表

WesternUseOption: [],

//西药天数下拉列表

DayNumOption: [],

infuseUseOption: [],

InfusionOption: [],

TreatTypeOption: [],

//西药处方表格主数据

WesternMedicineTable: [],

InfusionTable:[],

//中药处方表格数据

ChineseMedicineTable: [],

//诊疗项目表格

TreatmentTable: [],

//附加费可选表格数据

SurchargeTable: [],

//搜索西药

SearchWesternInput: "",

//搜索中药

SearchChineseInput: "",

//搜索诊疗项目

SearchCostItemInput: "",

//搜索附加费

SearchSurchargeInput: "",

//搜索输液

SearchInfusion:[""],

//搜索药品条件

SearchWesternModel: {

columnName: "",

limit: 10,

offset: 0,

order: "",

},

SearchChineseModel: {

columnName: "",

limit: 10,

offset: 0,

order: "",

},

fileIdFile: {

fileId: "",

action: "add",

deletes: [], // 待删除（已上传）的文件列表

uploads: [], // 待上传的文件列表

},

//病历配置字段

MedicineRecordLabelList: [

{

label: "个人史",

name: "personalHistory",

value: "",

},

{

label: "过敏史",

name: "allergyHistory",

value: "",

},

{

label: "疾病史",

name: "diseaseHistory",

value: "",

},

{

label: "传染病史",

name: "infectiousHistory",

value: "",

},

{

label: "手术史",

name: "surgeryHistory",

value: "",

},

{

label: "输血史",

name: "transfusionHistory",

value: "",

},

{

label: "体格检查",

name: "physiqueCheck",

value: "",

},

{

label: "急诊诊断",

name: "emergencyDiagnose",

value: "",

},

{

label: "急诊效果",

name: "emergencyEffect",

value: "",

},

{

label: "家族史",

name: "familyHistory",

value: "",

},

{

label: "月经史",

name: "lunariaHistory",

value: "",

},

{

label: "婚育史",

name: "pregnancyHistory",

value: "",

},

{

label: "辅助检查",

name: "assistCheck",

value: "",

},

{

label: "现病史",

name: "nowHistory",

value: "",

},

{

label: "既往史",

name: "beforeHistory",

value: "",

},

{

label: "其他检查",

name: "otherCheck",

value: "",

},

],

//选择的病例配置

ChooseRecordList: [],

//开具处方Tabs标签页

medicalEditTabs: [],

//基础信息

BasicInfoModel: {

name: "",

gender: {

name: "",

},

age: "",

month: "",

phone: "",

card: "",

withPatientNexus: {

name: "",

},

healthCardNo: "",

pulse: "",

breathe: "",

temperature: "",

},

//病历表单

MedicalRecordModel: this.ReturnMedicalModel(),

//患者表单验证

BasicInfoFormRules: {

treatType: [

{ required: true, message: "请选择治疗类型", trigger: "blur" },

],

name: [{ required: true, message: "请输入患者姓名", trigger: "blur" }],

gender: [{ required: true, message: "请选择性别", trigger: "blur" }],

age: [{ required: true, message: "请输入年龄", trigger: "blur" }],

phone: [{ required: true, message: "请输入联系方式", trigger: "blur" }],

card: [{ required: true, message: "请输入身份证号", trigger: "blur" }],

patientTell: [

{ required: true, message: "请输入主诉", trigger: "blur" },

],

},

//开具处方数据集合

recipelInfoEvtList: [],

payAmount: 0,

//打印时候的处方ID

recipelInfoId: "",

//是否显示完成接诊按钮

hasBtnFinishVisit: true,

//是否显示打印按钮

hasBtnPrint: false,

//只读

isReadOnly: true,

isBaseReadOnly: true,

//特殊诊所

isSpecial: false,

BASE\_API: process.env.BASE\_API,

defaultNumber:1,//默认组号

infusionProject:[[]],

drippingSpeed:[""], //滴速

days:[{}], //天数

frequency:[{}], //频次

infuseUse:[{}], //用法

zushu:[1],

excharge:[], //输液附加费

changeData:{},

hint:false, //用户切换提示

clickId:'', //点击id

chineseTest:[[]]

};

},

methods: {

changesDosage(row,index){

if(row==""){

this.medicalEditTabs[index].recipelInfo.dosage=" "

}else{

this.medicalEditTabs[index].recipelInfo.dosage=row-0

}

this.MedicalCalculate()

},

changeSingleDosage(row,index,item){

this.medicalEditTabs.forEach((tabElement) => {

if (tabElement.type === "recipelType\_1"&&tabElement.key==item.key) {

//for (let i = 0; i < tabElement.content.recipelDetailEvtList.length; i++) {

console.log(tabElement.content.recipelDetailEvtList[index],'就是看看');

if(row==""){

tabElement.content.recipelDetailEvtList[index].singleDosage=" "

}else{

tabElement.content.recipelDetailEvtList[index].singleDosage=row

}

//}

}

})

this.MedicalCalculate()

},

changeUse(row){

this.changeData = row

},

addGroup(item){

item.infusion.defaultNumber=item.infusion.defaultNumber+1

item.infusion.zushu.push(item.infusion.defaultNumber)

item.infusion.infusionProject.push([])

item.infusion.drippingSpeed.push("")

item.infusion.days.push({})

item.infusion.frequency.push({})

item.infusion.infuseUse.push({})

this.SearchInfusion.push("")

console.log(item,'添加');

},

minusGroup(item,index){

if(item.infusion.defaultNumber>1){

item.infusion.defaultNumber=item.infusion.defaultNumber-1

item.infusion.zushu.splice(index,1)

item.infusion.infusionProject.splice(index,1)

for (let i = 0; i < item.infusion.infusionProject.length; i++) {

for (let j = 0; j < item.infusion.infusionProject[i].length; j++) {

item.infusion.infusionProject[i][j].infuseGroup=i+1

}

}

item.infusion.drippingSpeed.splice(index,1)

item.infusion.days.splice(index,1)

item.infusion.frequency.splice(index,1)

item.infusion.infuseUse.splice(index,1)

this.SearchInfusion.splice(index,1)

console.log(item,'添加');

}

},

indexMethod(index){

index+1

return this.defaultNumber

},

//编辑保存

save() {

if(this.savelock){

this.$message.error('保存中，请稍后')

return

}

let FormCheck = true;

this.$refs["BasicInfoForm"].validate((valid) => {

if (valid) {

//alert('submit!');

} else {

FormCheck = false;

return false;

}

});

if (!FormCheck) {

this.$message.error("请填写主诉")

return;

}

let flage1=false;

let flage2=false;

let treatment=false;

let chargeflage=false;

let flageChinese1=false;

let flageChinese2=false;

let recipelInfoEvtList = [];

let returnForEach = false;

let hasSelsmallType = false;

let numberCheck = false;

let totalCheck = false;

let paramsCheck = false

if (this.medicalEditTabs.length == 0) {

this.$message.error("处方数量不能为0");

return;

}

this.medicalEditTabs.forEach((tabElement) => {

console.log(tabElement,'真奇怪');

//西药处方

if (

tabElement.type == "recipelType\_0" &&

!this.MedicalRecordModel.westernDiagnose

) {

returnForEach = true;

}

if (

tabElement.type == "recipelType\_0"

) {

let total = 0

tabElement.content.recipelDetailEvtList.forEach(item=>{

if(item.isExtra==0){

if(item.singleDosage&&item.frequency&&item.frequency.value&&item.days&&item.days.name){

total = Math.ceil(

BigNumber(item.singleDosage)

.multipliedBy(item.frequency.value.split("\_")[1])

.multipliedBy(item.days.name)

.toNumber()

);

if(item.drugStuffId.drug.total!=0&&total>item.drugStuffId.drug.total){

this.$message.error(item.drugStuffId.name+'总量上限小于功效说明计算总量')

totalCheck = true

}

}else{

this.$message.error(item.drugStuffId.name+'信息未填写完整')

paramsCheck = true

}

}

})

}

//中药处方

if (

tabElement.type == "recipelType\_1" &&

!this.MedicalRecordModel.chinaDiagnose

) {

returnForEach = true;

}

if(tabElement.type == "recipelType\_1"){

if(tabElement.content.recipelInfo.dosage==""){

flageChinese1=true

}

for (let i = 0; i < tabElement.content.recipelDetailEvtList.length; i++) {

if(tabElement.content.recipelDetailEvtList[i].isExtra==0){

if(tabElement.content.recipelDetailEvtList[i].singleDosage==" "||tabElement.content.recipelDetailEvtList[i].singleDosage==""){

flageChinese2=true

}

}

}

}

//输液处方

if(tabElement.type == "recipelType\_2"){

tabElement.content.recipelDetailEvtList=[]

let arr=0;

for (let i = 0; i < tabElement.infusion.infusionProject.length; i++) {

for (let j = 0; j < tabElement.infusion.infusionProject[i].length; j++) {

tabElement.infusion.infusionProject[i][j].drippingSpeed=tabElement.infusion.drippingSpeed[tabElement.infusion.infusionProject[i][j].infuseGroup-1]

tabElement.infusion.infusionProject[i][j].days=tabElement.infusion.days[tabElement.infusion.infusionProject[i][j].infuseGroup-1]

tabElement.infusion.infusionProject[i][j].frequency=tabElement.infusion.frequency[tabElement.infusion.infusionProject[i][j].infuseGroup-1]

tabElement.infusion.infusionProject[i][j].infuseUse=tabElement.infusion.infuseUse[tabElement.infusion.infusionProject[i][j].infuseGroup-1]

tabElement.content.recipelDetailEvtList[arr]= tabElement.infusion.infusionProject[i][j]

arr++;

if(tabElement.infusion.days[tabElement.infusion.infusionProject[i][j].infuseGroup-1].name==""||tabElement.infusion.days[tabElement.infusion.infusionProject[i][j].infuseGroup-1].name==undefined){

flage1=true

}

if(tabElement.infusion.frequency[tabElement.infusion.infusionProject[i][j].infuseGroup-1].name==""||tabElement.infusion.frequency[tabElement.infusion.infusionProject[i][j].infuseGroup-1].name==undefined){

flage2=true

}

}

}

for (let i = 0; i < tabElement.infusion.excharge.length; i++) {

tabElement.content.recipelDetailEvtList.push(tabElement.infusion.excharge[i])

}

}

//诊疗项目

if(tabElement.type == "recipelType\_3"){

for (let i = 0; i < tabElement.content.recipelDetailEvtList.length; i++) {

if(tabElement.content.recipelDetailEvtList[i].isExtra==0){

if(tabElement.content.recipelDetailEvtList[i].total==""||tabElement.content.recipelDetailEvtList[i].singleDosage==""||tabElement.content.recipelDetailEvtList[i].total==0||tabElement.content.recipelDetailEvtList[i].singleDosage==0){

treatment=true

}

}

}

}

//附加费

for (let i = 0; i < tabElement.content.recipelDetailEvtList.length; i++) {

if(tabElement.content.recipelDetailEvtList[i].isExtra==1){

if(tabElement.content.recipelDetailEvtList[i].total==""||tabElement.content.recipelDetailEvtList[i].singleDosage==""||tabElement.content.recipelDetailEvtList[i].total==0||tabElement.content.recipelDetailEvtList[i].singleDosage==0){

chargeflage=true

}

}

}

if (

!tabElement.content.recipelInfo.smallType &&

tabElement.type != "recipelType\_3"

) {

hasSelsmallType = true;

}

if (tabElement.content.recipelDetailEvtList.length == 0) {

numberCheck = true;

}

recipelInfoEvtList.push(tabElement.content);

});

if (returnForEach) {

this.$message.error("存在西药处方或中药处方，请填写对应诊断说明。");

return;

}

if (hasSelsmallType) {

this.$message.error("请检查每个处方里的分类是否已选择。");

return;

}

if (numberCheck) {

this.$message.error("请检查每个处方里是否有药品。");

return;

}

if(totalCheck){

return

}

if(paramsCheck){

return

}

if(flage1){

this.$message.error("请填写输液天数")

return;

}

if(flage2){

this.$message.error("请填写输液频次")

return;

}

if(flageChinese1){

this.$message.error("请填写中药剂量")

return;

}

if(flageChinese2){

this.$message.error("请填写中药药品数量")

return;

}

if(treatment){

this.$message.error("请填写诊疗处方的项目数量")

return;

}

if(chargeflage){

this.$message.error("请填写附加费的项目数量")

return;

}

this.MedicalRecordModel.company = this.Company;

this.MedicalRecordModel.doctor.id = this.UserId;

let medicalRecipelEvt = {

id: this.registration.id,

registration: this.registration,

medicalRecord: this.MedicalRecordModel,

patient: this.BasicInfoModel,

recipelInfoEvtList: recipelInfoEvtList,

type: 0,

};

console.log(medicalRecipelEvt,'怎么回事');

//return

let formData = new FormData();

formData.append("entity", JSON.stringify(medicalRecipelEvt));

this.savelock = true

editSave(formData).then((responseData) => {

if (responseData.code == 100) {

this.$message.success("操作成功！");

this.hasBtnFinishVisit = false;

this.hasBtnPrint = false;

this.isReadOnly = true;

this.medicalEditTabs = []

this.medicalEditTabsValue = {}

this.medicalClickTabsValue = {}

this.BasicInfoModel = {

name: "",

gender: {

name: "",

},

age: "",

month: "",

phone: "",

card: "",

withPatientNexus: {

name: "",

},

healthCardNo: "",

pulse: "",

breathe: "",

temperature: "",

},

this.MedicalRecordModel = this.ReturnMedicalModel();

this.GetPreparePatientList();

this.GetAlreadyPatientList();

} else {

this.$message.error("更新失败！");

}

this.savelock = false

});

},

edit() {

if (this.medicalClickTabsValue&&this.medicalClickTabsValue.content.recipelInfo.id) {

console.log(this.medicalEditTabs,'获取值');

let now = new Date()

let newDate = new Date(this.activeSelAlreadyPatientItem.createDate)

now = Date.parse(now)

newDate = Date.parse(newDate)

if(this.timeOut){

this.$message.error('超过一天的数据无法编辑')

return

}

this.isReadOnly = false;

// this.chineseTest=[[]]

// for (let i = 1; i < this.medicalEditTabs.length; i++) {

// if(this.medicalEditTabs[i].type=="recipelType\_1"){

// this.chineseTest.push([])

// }

// }

// for (let i = 0; i < this.medicalEditTabs.length; i++) {

// if(this.medicalEditTabs[i].type=="recipelType\_1"){

// for (let j = 0; j < this.medicalEditTabs[i].content.recipelDetailEvtList.length; j++) {

// this.chineseTest[this.medicalEditTabs[i].key-1].push(this.medicalEditTabs[i].content.recipelDetailEvtList[j].drugStuffId.name)

// }

// }

// }

// console.log(this.chineseTest,'参数');

} else {

this.$message.error("未选择用户");

}

},

print() {

if (!this.activeSelAlreadyPatientItem) {

return;

}

let filterArr = this.medicalEditTabs.filter(

(item) => item.content.recipelInfo.status != -1

);

let arr = []

filterArr.forEach((item, index) => {

let params = {

recipelInfoId: item.content.recipelInfo.id,

type: item.content.recipelInfo.recipelType.value,

};

let str = "";

let type =

params.type == "recipelType\_0"

? "westMedicine"

: params.type == "recipelType\_1"

? "chineseMedicine"

: params.type == "recipelType\_2"

? "infuse"

: "costItem";

if (params.recipelInfoId) {

str = "&recipelInfoId=" + params.recipelInfoId;

}

arr[index] = window.open(

this.BASE\_API +

"/ureport/preview?\_u=Newtouch:" +

type +

".ureport.xml&\_t=1,6,7,8,9" +

str +

"&type=0",

index

);

arr[index].onload=()=>{

arr[index].document.title = "新致云诊所"

}

});

},

handleFileRemove(file, fileList) {

console.log(12);

},

handleFileProgress(event, file, fileList) {

console.log(11);

},

handleFileBeforeUpload(file) {

console.log(13);

},

handlePatientClick(tab, event) {

if (this.TodayActiveName === "prepare") {

this.isReadOnly = true;

this.ChineseButton = false;

this.$refs["BasicInfoForm"].clearValidate();

this.GetPreparePatientList();

} else {

this.isReadOnly = true;

this.ChineseButton = false;

this.$refs["BasicInfoForm"].clearValidate();

this.GetAlreadyPatientList();

}

console.log("---------:" + this.TodayActiveName);

},

//获取待就诊患者列表

GetPreparePatientList() {

let params = [

{

columnName: "company\_id",

queryType: "=",

value: this.Company.id,

},

{

columnName: "status",

queryType: "=",

value: "",

},

{

columnName: "doctor\_id",

queryType: "=",

value: this.UserId,

},

];

if (this.SearchPatientInfo) {

params.push({

logic: "AND",

queryType: "(",

});

params.push({

logic: "",

columnName: "patientId.name",

queryType: "like",

value: this.SearchPatientInfo,

});

params.push({

logic: "OR",

columnName: "patientId.phone",

queryType: "like",

value: this.SearchPatientInfo,

});

params.push({

logic: "OR",

columnName: "patientId.card",

queryType: "like",

value: this.SearchPatientInfo,

});

params.push({

logic: "",

queryType: ")",

});

}

this.SearchPreModel.params = params;

this.SearchPreModel.params[1].value = "registrationStatus\_0";

listRegistrationPage(this.SearchPreModel).then((responseData) => {

if (responseData.code == 100) {

this.PreparePatientTotal = responseData.data.total;

if (responseData.data.rows) {

responseData.data.rows.forEach((element) => {

let birthday = element.patientId.birthday;

if (birthday) {

const duration = moment.duration(moment().diff(birthday));

element.patientId.birthdayText = duration.years();

} else {

element.patientId.birthdayText = "--";

}

element.formatUpdateDate = moment(element.updateDate).format(

"MM/DD HH:mm"

);

});

}

this.PreparePatientList = responseData.data.rows;

}

});

},

//获取已就诊患者列表

GetAlreadyPatientList() {

let params = [

{

columnName: "company\_id",

queryType: "=",

value: this.Company.id,

},

{

columnName: "status",

queryType: "in",

value: ["registrationStatus\_1", "registrationStatus\_4"],

},

{

columnName: "doctor\_id",

queryType: "=",

value: this.UserId,

},

];

if (this.SearchPatientInfo) {

params.push({

logic: "AND",

queryType: "(",

});

params.push({

logic: "",

columnName: "patientId.name",

queryType: "like",

value: this.SearchPatientInfo,

});

params.push({

logic: "OR",

columnName: "patientId.phone",

queryType: "like",

value: this.SearchPatientInfo,

});

params.push({

logic: "OR",

columnName: "patientId.card",

queryType: "like",

value: this.SearchPatientInfo,

});

params.push({

logic: "",

queryType: ")",

});

}

this.SearchAlreadyModel.params = params;

listRegistrationPage(this.SearchAlreadyModel).then((responseData) => {

if (responseData.code == 100) {

this.AlreadyPatientListTotal = responseData.data.total;

if (responseData.data.rows) {

responseData.data.rows.forEach((element) => {

let birthday = element.patientId.birthday;

if (birthday) {

const duration = moment.duration(moment().diff(birthday));

element.patientId.birthdayText = duration.years();

} else {

element.patientId.birthdayText = "--";

}

element.formatReceptionEndDate = moment(

element.receptionEndDate

).format("MM/DD HH:mm");

});

}

this.AlreadyPatientList = responseData.data.rows;

}

});

},

handleCurrentPreparePatientChange(pageNum) {

this.SearchPreModel.offset = (pageNum - 1) \* this.SearchPreModel.limit;

this.GetPreparePatientList();

},

handleCurrentAlreadyPatientChange(pageNum) {

this.SearchAlreadyModel.offset =

(pageNum - 1) \* this.SearchAlreadyModel.limit;

this.GetAlreadyPatientList();

},

AlreadyPatientDeletilQuery() {

console.log(this.activeSelAlreadyPatientItem, "我的");

if (!this.activeSelAlreadyPatientItem) {

return;

}

let now = new Date()

let newDate = new Date(this.activeSelAlreadyPatientItem.createDate)

now = Date.parse(now)

newDate = Date.parse(newDate)

if(now-newDate>86400000){

this.timeOut = true

}else{

this.timeOut = false

}

let itemCopy = JSON.parse(

JSON.stringify(this.activeSelAlreadyPatientItem)

);

this.registration = JSON.parse(

JSON.stringify(this.activeSelAlreadyPatientItem)

);

this.$nextTick(() => {

this.MedicalRecordModel = this.ReturnMedicalModel();

this.hasBtnFinishVisit = false;

this.hasBtnPrint = true;

this.isReadOnly = true;

this.recipelInfoId = "";

// let model = {

// company: {

// id: this.Company.id,

// },

// doctor: {

// id: this.UserId,

// },

// id: itemCopy.id,

// registrationId:this.registration.id

// };

// console.log(model,'hahah');

this.AlreadyPatientDescriptions = [];

allQueryMedicalRecord(this.registration.id).then((responseData) => {

let arr=responseData.data

console.log(arr,'这是怎么回事');

if (responseData.code == 100) {

let recipelInfoEvtList = responseData.data.recipelInfoEvtList;

recipelInfoEvtList.forEach((element) => {

this.AlreadyPatientDescriptions.push(

JSON.parse(JSON.stringify(element.recipelInfo))

);

});

this.AlreadyPatientDescriptions.sort(function (a, b) {

a.seq = a.seq ? a.seq : 0;

b.seq = b.seq ? b.seq : 0;

return a.seq - b.seq;

});

console.log(this.AlreadyPatientDescriptions, "utiy");

console.log(responseData.data, "poiu");

this.MedicalRecordModel = responseData.data.medicalRecord;

this.MedicalRecordModel.registration.id = itemCopy.id;

this.MedicalRecordModel.registration.treatType = itemCopy.treatType;

this.BasicInfoModel = responseData.data.patient;

this.BasicInfoModel.breathe = this.registration.breathe;

this.BasicInfoModel.temperature =

this.registration.temperature;

this.BasicInfoModel.pulse = this.registration.pulse;

this.BasicInfoModel.bloodPressure = this.registration.bloodPressure;

this.GetMedicineCofigure();

this.createMedicalEditTab(recipelInfoEvtList);

}

});

// this.GetPatient(itemCopy);

console.log(this.MedicalRecordModel, "nuide");

});

},

AlreadyPatientDescriptionsQuery(item) {

console.log(item, "dshafhfkjsafhkj");

this.hasBtnFinishVisit = false;

this.medicalEditTabs.forEach((element) => {

if (element.content.uuid == item.id) {

this.medicalEditTabsValue = element;

this.medicalClickTabsValue = this.medicalEditTabsValue;

}

});

this.recipelInfoId = item.id;

},

clickPreparePatient(item) {

// if(item.id===this.clickId){

// return

// }

console.log(item, "点击用户");

if(!this.hint){

this.clickId=item.id

let itemCopy = JSON.parse(JSON.stringify(item));

this.registration = JSON.parse(JSON.stringify(item));

this.$nextTick(() => {

this.hasBtnFinishVisit = true;

this.hasBtnPrint = false;

this.isReadOnly = false;

this.ChineseButton = true;

this.MedicalRecordModel = this.ReturnMedicalModel();

this.MedicalRecordModel.registration.id = itemCopy.id;

this.MedicalRecordModel.registration.treatType = itemCopy.treatType;

this.GetPatient(itemCopy);

this.GetMedicineCofigure();

this.medicalEditTabs = [];

this.addMedicalEditTab(this.medicalTypeList[0]);

});

this.hint=true

}else{

this.open(item)

}

},

open(item) {

this.$confirm('当前患者还未完成接诊是否确定切换?', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning'

}).then(() => {

this.hint=false

this.clickPreparePatient(item)

}).catch(() => {

this.hint=true

});

},

GetPatient(patientId) {

getPatientById(patientId.patientId.id).then((responseData) => {

if (responseData.code == 100) {

console.log(responseData.data, "查询用户");

this.BasicInfoModel = responseData.data;

this.BasicInfoModel.breathe = patientId.breathe;

this.BasicInfoModel.temperature = patientId.temperature;

this.BasicInfoModel.pulse = patientId.pulse;

this.BasicInfoModel.bloodPressure = patientId.bloodPressure;

}

});

},

ReturnMedicalModel() {

return {

company: this.Company,

doctor: {

id: this.UserId,

},

patientTell: "",

nowHistory: "",

beforeHistory: "",

diagnose: "",

otherCheck: "",

personalHistory: "",

allergyHistory: "",

diseaseHistory: "",

infectiousHistory: "",

surgeryHistory: "",

transfusionHistory: "",

physiqueCheck: "",

emergencyDiagnose: "",

emergencyEffect: "",

familyHistory: "",

lunariaHistory: "",

pregnancyHistory: "",

assistCheck: "",

registration: {

treatType: {

name: "初诊",

value: "treatType\_0",

},

},

};

},

//开具处方新增Tab标签页

addMedicalEditTab(tabTypeData) {

let tabType = tabTypeData.type;

let typeMaxTabIndex = 0;

this.medicalEditTabs.forEach((element) => {

if (element.type === tabType) {

typeMaxTabIndex++;

}

});

let isFollowUp =

this.MedicalRecordModel.registration.treatType.value === "treatType\_0"

? 0

: 1;

let recipelInfoEvt = {};

if (tabType == "recipelType\_1") {

recipelInfoEvt = {

uuid: require("uuid").v1(),

recipelInfo: {

company: this.Company,

recipelType: {

name: tabTypeData.name,

value: tabTypeData.type,

},

remarks: tabTypeData.name + (typeMaxTabIndex + 1),

fee: 0,

status: 1,

chargeStatus: 0,

dispensionStatus: 0,

isFollowUp: isFollowUp,

recipelUse: this.isSpecial

? {

name: this.ChineseUseOption[0].name,

value: this.ChineseUseOption[0].value,

}

: {

name: "",

value: "",

},

smallType:this.RecipelSmallTypeList[0]

},

recipelDetailEvtList: [],

};

} else {

recipelInfoEvt = {

uuid: require("uuid").v1(),

recipelInfo: {

company: this.Company,

recipelType: {

name: tabTypeData.name,

value: tabTypeData.type,

},

remarks: tabTypeData.name + (typeMaxTabIndex + 1),

fee: 0,

status: 1,

chargeStatus: 0,

dispensionStatus: 0,

isFollowUp: isFollowUp,

smallType:this.RecipelSmallTypeList[0]

},

recipelDetailEvtList: [],

};

}

// if (tabTypeData.type === "recipelType\_1") {

// recipelInfoEvt.recipelInfo["dosage"] = 1;

// }

let infusion={}

if (tabTypeData.type === "recipelType\_2") {

infusion={

// infusionProject:this.infusionProject,

// drippingSpeed:this.drippingSpeed, //滴速

// days:this.days, //天数

// frequency:this.frequency, //频次

// infuseUse:this.infuseUse, //用法

// zushu:this.zushu,

defaultNumber:1,//默认组号

infusionProject:[[]],

drippingSpeed:[""], //滴速

days:[{}], //天数

frequency:[{}], //频次

infuseUse:[{}], //用法

zushu:[1],

excharge:[]

}

}

let costDebug={

chineseTest:[]

}

this.recipelInfoEvtList.push(recipelInfoEvt);

let tab = {

title: tabTypeData.name + (typeMaxTabIndex + 1),

key: this.medicalEditTabs.length + 1,

content: recipelInfoEvt,

closable: true,

type: tabType,

infusion:infusion,

costDebug:costDebug,

};

// if(tab.type=="recipelType\_1"){

// this.chineseTest[tab.key-1]=[]

// }

this.SearchWesternInput = "";

this.SearchChineseInput = "";

this.SearchCostItemInput = "";

this.SearchSurchargeInput = "";

this.SearchInfusion=[""]

this.medicalEditTabs.push(tab);

this.medicalEditTabsValue = tab;

this.medicalClickTabsValue = this.medicalEditTabsValue;

console.log(this.medicalEditTabsValue,'新增');

console.log(require("uuid").v1());

},

//开具处方删除Tab标签页

removeMedicalEditTab(targetName) {

console.log(targetName);

let tabs = this.medicalEditTabs;

let activeName = this.medicalEditTabsValue;

if (activeName === targetName) {

tabs.forEach((tab, index) => {

if (tab === targetName) {

let nextTab = tabs[index + 1] || tabs[index - 1];

if (nextTab) {

activeName = nextTab;

}

}

});

}

this.medicalEditTabsValue = activeName;

this.medicalEditTabs = tabs.filter((tab) => tab !== targetName);

this.SearchWesternInput = "";

this.SearchChineseInput = "";

this.SearchInfusion=[""]

if (this.medicalEditTabs.length > 0) {

this.medicalClickTabsValue = this.medicalEditTabsValue;

} else {

this.medicalClickTabsValue = {};

}

this.MedicalCalculate();

this.medicalEditTabs = this.medicalEditTabs;

},

//开具处方切换Tab标签页触发事件

clickMedicalEditTab() {

if (this.medicalEditTabsValue === this.medicalClickTabsValue) {

return;

}

this.SearchWesternInput = "";

this.medicalClickTabsValue = this.medicalEditTabsValue;

console.log(this.medicalEditTabsValue);

},

createMedicalEditTab(recipelInfoEvtList) {

console.log("---------recipelInfoEvtList------------");

console.log(recipelInfoEvtList);

this.medicalEditTabs = [];

if (recipelInfoEvtList == undefined || recipelInfoEvtList == null) {

return;

}

recipelInfoEvtList.forEach((element) => {

element.uuid = element.recipelInfo.id;

let costDebug={

chineseTest:[]

}

for (let i = 0; i < this.medicalEditTabs.length; i++) {

for (let j = 0; j < this.medicalEditTabs[i].content.recipelDetailEvtList.length; j++) {

costDebug.chineseTest.push(this.medicalEditTabs[i].content.recipelDetailEvtList[j].drugStuffId.name)

}

}

if(element.recipelInfo.recipelType.value=="recipelType\_2"){

let infusion={

defaultNumber:1,//默认组号

infusionProject:[[]],

drippingSpeed:[""], //滴速

days:[{}], //天数

frequency:[{}], //频次

infuseUse:[{}], //用法

zushu:[1],

excharge:[]

}

let count=0

for (let i = 0; i < element.recipelDetailEvtList.length; i++) {

if(element.recipelDetailEvtList[i].isExtra===0){

if(count<element.recipelDetailEvtList[i].infuseGroup){

count=element.recipelDetailEvtList[i].infuseGroup

}

}

}

for (let i = 1; i < count; i++) {

infusion.defaultNumber=infusion.defaultNumber+1;

infusion.zushu.push(infusion.defaultNumber)

infusion.infusionProject.push([])

infusion.drippingSpeed.push("")

infusion.days.push({})

infusion.frequency.push({})

infusion.infuseUse.push({})

}

//this.excharge=null

let arr=[]

for (let i = 0; i < element.recipelDetailEvtList.length; i++) {

if(element.recipelDetailEvtList[i].isExtra!=1){

infusion.infusionProject[element.recipelDetailEvtList[i].infuseGroup-1].push(element.recipelDetailEvtList[i])

infusion.drippingSpeed[element.recipelDetailEvtList[i].infuseGroup-1]=element.recipelDetailEvtList[i].drippingSpeed

infusion.days[element.recipelDetailEvtList[i].infuseGroup-1]=element.recipelDetailEvtList[i].days

infusion.frequency[element.recipelDetailEvtList[i].infuseGroup-1]=element.recipelDetailEvtList[i].frequency

infusion.infuseUse[element.recipelDetailEvtList[i].infuseGroup-1]=element.recipelDetailEvtList[i].infuseUse

}else{

infusion.excharge.push(element.recipelDetailEvtList[i])

//arr.push(element.recipelDetailEvtList[i])

}

}

// this.excharge=arr

let tab = {

title: element.recipelInfo.name,

key: element.recipelInfo.seq,

content: element,

closable: false,

type: element.recipelInfo.recipelType.value,

infusion:infusion,

costDebug:costDebug

};

this.medicalEditTabs.push(tab);

}else{

let tab = {

title: element.recipelInfo.name,

key: element.recipelInfo.seq,

content: element,

closable: false,

type: element.recipelInfo.recipelType.value,

costDebug:costDebug

};

this.medicalEditTabs.push(tab);

}

});

// this.infusionProject=[]

// this.infusionProject[0].push([])

// console.log(this.infusionProject,'集合查看');

// this.medicalEditTabs.forEach(items=>{

// if(items.type=="recipelType\_2"){

// let count=0

// for (let i = 0; i < items.content.recipelDetailEvtList; i++) {

// if(this.medicalEditTabs[i].type=='recipelType\_2'){

// if(count<this.medicalEditTabs[i].content.recipelDetailEvtList.infuseGroup){

// count=this.medicalEditTabs[i].content.recipelDetailEvtList.infuseGroup

// }

// }

// }

// for (let i = 1; i < count; i++) {

// this.zushu.push(i+1)

// }

// console.log(this.zushu,'组数');

// items.content.recipelDetailEvtList.forEach(infusionItem=>{

// console.log(infusionItem,'集合查看')

// if(infusionItem.infuseGroup-1==0){

// this.infusionProject[0].push(infusionItem)

// }else{

// this.infusionProject[infusionItem.infuseGroup-1].push(infusionItem)

// }

// // items.content.recipelDetailEvtList.infuseGroup

// console.log(infusionItem,'kankanqiqi')

// })

// }

// })

// for (let i = 0; i < this.infusionProject.length; i++) {

// for (let j = 0; j < this.infusionProject[i].length; j++) {

// this.drippingSpeed[this.infusionProject[i][j].infuseGroup-1]=this.infusionProject[i][j].drippingSpeed

// this.days[this.infusionProject[i][j].infuseGroup-1]=this.infusionProject[i][j].days

// this.frequency[this.infusionProject[i][j].infuseGroup-1]=this.infusionProject[i][j].frequency

// this.infuseUse[this.infusionProject[i][j].infuseGroup-1]=this.infusionProject[i][j].infuseUse

// }

// }

this.medicalEditTabsValue = this.medicalEditTabs[0];

this.medicalClickTabsValue = this.medicalEditTabsValue;

this.MedicalCalculate();

},

//获取数据字典的值

GetAllOption() {

this.GetOption("1014474470772899981");

this.GetOption("1014474470772899990");

this.GetOption("1014474470772899985");

this.GetOption("1014474470772900028");

this.GetOption("1014474470772900052");

this.GetOption("1014474470772900062");

this.GetOption("1014474470772900068");

this.GetOption("1014474470772900058");

this.GetOption("1088952622484547519");

this.GetOption("1008534118685450388");

},

GetOption(optionId) {

console.log(optionId);

let model = {

params: [

{

columnName: "dict\_type\_id",

queryType: "=",

value: optionId,

},

],

};

listDictItemAll(model).then((responseData) => {

if (optionId == "1014474470772899981"){

this.ChineseUseOption = responseData.data;

if (this.isSpecial) {

this.ChineseUseOption = this.ChineseUseOption.filter(

(item) => item.name == "水冲"

);

}

}else if (optionId == "1014474470772899990")

this.FrequencyOption = responseData.data;

else if (optionId == "1014474470772899985")

this.ChineseTimeOption = responseData.data;

else if (optionId == "1014474470772900028")

this.WesternUseOption = responseData.data;

else if (optionId == "1014474470772900052")

this.DayNumOption = responseData.data;

else if (optionId == "1014474470772900062")

this.infuseUseOption = responseData.data;

else if (optionId == "1014474470772900068")

this.InfusionOption = responseData.data;

else if (optionId == "1014474470772900058")

this.ChineseUseTimeOption = responseData.data;

else if (optionId == "1088952622484547519")

this.RecipelSmallTypeList = responseData.data;

else if (optionId == "1008534118685450388") {

this.TreatTypeOption = [];

if (responseData.data) {

responseData.data.forEach((element) => {

this.TreatTypeOption.push({

name: element.name,

value: element.value,

});

});

}

}

});

},

//病历配置恢复默认

ClearMedicineCofigure() {

console.log(this.MedicalRecordModel);

this.ChooseRecordList.map((item) => {

this.MedicalRecordModel[`${item.name}`] = "";

});

this.ChooseRecordList = [];

},

GetMedicineCofigure() {

this.ChooseRecordList = [];

this.MedicineRecordLabelList.map((item) => {

if (this.MedicalRecordModel[`${item.name}`]) {

this.ChooseRecordList.push(item);

}

});

},

GetMedicalRecordAll() {

let params = [

{

columnName: "company\_id",

queryType: "=",

value: this.Company.id,

},

{

columnName: "type",

queryType: "in",

value: [],

},

{

columnName: "goods\_name",

queryType: "like",

value: "",

},

];

this.SearchWesternModel.params = JSON.parse(JSON.stringify(params));

this.SearchChineseModel.params = JSON.parse(JSON.stringify(params));

//this.GetWesternTable();

//this.GetChineseTable();

},

GetWesternTable() {

this.SearchWesternModel.params[1].value = [

"medicalType\_0",

"medicalType\_2",

];

//判断是否输入的是英文

var pattern2 = new RegExp("[A-Za-z]+");

if (pattern2.test(this.SearchWesternInput)) {

// console.log(this.SearchWesternInput,'字符');

this.SearchWesternModel.params[2].value =

this.SearchWesternInput.toUpperCase();

this.SearchWesternModel.params[2].columnName = "pinyin\_code";

} else {

this.SearchWesternModel.params[2].value = this.SearchWesternInput;

this.SearchWesternModel.params[2].columnName = "goods\_name";

}

listAllStock(this.SearchWesternModel).then((responseData) => {

if (responseData.code == 100) {

// responseData.data.forEach((element) => {

// let isUnpackSell = element.isUnpackSell; //允许拆零销售

// let stockNumber = element.stockNumber; //库存数量

// let packName = element.pack.name; //包装（单位）

// let preparation = element.preparation; //制剂

// let preparationUnit = element.preparationUnit.name; //制剂单位

// let price = element.price; //售价

// let retailPrice = element.retailPrice; //拆开后零售价

// let inventory = element.inventory; //库存

// let stockText = "";

// //向下取整

// let packNumber = Math.floor(inventory / preparation);

// //取余

// let retailNumber = inventory % preparation;

// if(packNumber > 0)

// {

// stockText = packNumber + packName;

// }

// if(retailNumber > 0)

// {

// if(packNumber > 0)

// {

// stockText = stockText + retailNumber + preparationUnit;

// }

// else

// {

// stockText = retailNumber + preparationUnit;

// }

// }

// element["stockText"] = stockText;

// });

this.WesternMedicineTable = responseData.data;

}

});

},

GetInfusionTable(index) {

this.SearchWesternModel.params[1].value = [

"medicalType\_0",

"medicalType\_2",

];

//判断是否输入的是英文

var pattern2 = new RegExp("[A-Za-z]+");

if (pattern2.test(this.SearchInfusion[index])) {

// console.log(this.SearchWesternInput,'字符');

this.SearchWesternModel.params[2].value =

this.SearchInfusion[index].toUpperCase();

this.SearchWesternModel.params[2].columnName = "pinyin\_code";

} else {

this.SearchWesternModel.params[2].value = this.SearchInfusion[index];

this.SearchWesternModel.params[2].columnName = "goods\_name";

}

listAllStock(this.SearchWesternModel).then((responseData) => {

if (responseData.code == 100) {

// responseData.data.forEach((element) => {

// let isUnpackSell = element.isUnpackSell; //允许拆零销售

// let stockNumber = element.stockNumber; //库存数量

// let packName = element.pack.name; //包装（单位）

// let preparation = element.preparation; //制剂

// let preparationUnit = element.preparationUnit.name; //制剂单位

// let price = element.price; //售价

// let retailPrice = element.retailPrice; //拆开后零售价

// let inventory = element.inventory; //库存

// let stockText = "";

// //向下取整

// let packNumber = Math.floor(inventory / preparation);

// //取余

// let retailNumber = inventory % preparation;

// if(packNumber > 0)

// {

// stockText = packNumber + packName;

// }

// if(retailNumber > 0)

// {

// if(packNumber > 0)

// {

// stockText = stockText + retailNumber + preparationUnit;

// }

// else

// {

// stockText = retailNumber + preparationUnit;

// }

// }

// element["stockText"] = stockText;

// });

this.InfusionTable = responseData.data;

}

});

},

GetChineseTable() {

this.SearchChineseModel.params[1].value = ["medicalType\_1"];

var pattern2 = new RegExp("[A-Za-z]+");

if (pattern2.test(this.SearchChineseInput)) {

// console.log(this.SearchWesternInput,'字符');

this.SearchChineseModel.params[2].value =

this.SearchChineseInput.toUpperCase();

this.SearchChineseModel.params[2].columnName = "pinyin\_code";

} else {

this.SearchChineseModel.params[2].value = this.SearchChineseInput;

this.SearchChineseModel.params[2].columnName = "goods\_name";

}

listAllStock(this.SearchChineseModel).then((responseData) => {

if (responseData.code == 100) {

this.ChineseMedicineTable = responseData.data;

}

});

},

//西药表格行点击选择添加到已选择西药处方的表格

RowClickWesternTable(row) {

let recipelDetailEvtList =

this.medicalClickTabsValue.content.recipelDetailEvtList;

let count = 0;

let hasFlag = false;

if(row.days.value&&row.frequency.value&&row.singleDosage&&row.total!=0){

let total = Math.ceil(

BigNumber(row.singleDosage)

.multipliedBy(row.frequency.value.split("\_")[1])

.multipliedBy(row.days.name)

.toNumber()

);

if(total>row.total){

this.$message.error('该药品设置的总量上限小于默认功效说明计算总量')

return

}

}

recipelDetailEvtList.forEach((element) => {

if (element.isExtra === 0) {

count++;

if (row.id === element.drugStuffId.drugStuffId) {

hasFlag = true;

}

}

});

if (hasFlag) {

this.$message.error("已有该药品");

return;

}

if (count == 5) {

this.$message.error("每个西药处方最多只能添加5个药品");

return;

}

let drugStuff = {

drugStuffId: row.id,

name: row.goodsName,

price: row.price,

retailPrice: row.retailPrice,

isUnpackSell: row.isUnpackSell,

dosisUnit: row.dosisUnit,

preparationUnit: row.preparationUnit,

pack: row.pack,

drug: row,

};

let recipelDetailEvt = {

allFee: 0,

total: 0,

singleDosage:0,

company: this.Company,

stuffType: 0,

isExtra: 0,

isUnpackSell: BigNumber(row.isUnpackSell).toNumber(),

drugStuffId: drugStuff,

};

this.medicalClickTabsValue.content.recipelDetailEvtList.push(

JSON.parse(JSON.stringify(recipelDetailEvt))

);

setTimeout(() => {

this.MedicalCalculate()

}, 0);

},

//输液表格行点击选择添加到已选择西药处方的表格

RowClickInfusionTable(row, column, event,index) {

console.log(this.medicalClickTabsValue.content,'就像');

let content = this.medicalClickTabsValue.content;

let recipelDetailEvtList =

this.medicalClickTabsValue.content.recipelDetailEvtList;

let count = 0;

let hasFlag = false;

console.log(row);

recipelDetailEvtList.forEach((element) => {

if (element.isExtra === 0) {

count++;

if (row.id === element.drugStuffId.drugStuffId) {

hasFlag = true;

}

}

});

// if (hasFlag) {

// this.$message.error("已有该药品");

// return;

// }

if (count == 5) {

this.$message.error("每个输液处方最多只能添加5个药品");

return;

}

let drugStuff = {

drugStuffId: row.id,

name: row.goodsName,

price: row.price,

retailPrice: row.retailPrice,

isUnpackSell: row.isUnpackSell,

dosisUnit: row.dosisUnit,

preparationUnit: row.preparationUnit,

pack: row.pack,

drug: row,

};

let recipelDetailEvt = {

allFee:0,

total:0,

company: this.Company,

stuffType:0,

isExtra:0,

isUnpackSell: BigNumber(row.isUnpackSell).toNumber(),

singleDosage:0,

skinTest:{},

drugStuffId: drugStuff,

infuseGroup:index+1,

drippingSpeed:"",

days:{},

frequency:{},

infuseUse:{}

};

// this.medicalClickTabsValue.content.recipelDetailEvtList.push(

// JSON.parse(JSON.stringify(recipelDetailEvt))

// );

this.medicalClickTabsValue.infusion.infusionProject[index].push(

JSON.parse(JSON.stringify(recipelDetailEvt))

);

console.log(drugStuff);

console.log(content);

},

//点击西药表格后的删除按钮删除指定下标的数据

DeleteMedicalRow(index,row,item) {

// console.log(

// this.chineseTest,

// "就是"

// );

// if(row.stuffType==1){

// for (let i = 0; i < this.chineseTest[item.key-1].length; i++) {

// if(row.drugStuffId.drug.name==this.chineseTest[item.key-1][i]){

// this.chineseTest[item.key-1].splice(i,1)

// //i--;

// }

// }

// }

this.medicalClickTabsValue.content.recipelDetailEvtList =

this.medicalClickTabsValue.content.recipelDetailEvtList.filter(

(item) => item !== row

);

if(row.infuseGroup>0){

this.medicalClickTabsValue.content.recipelDetailEvtList =

this.medicalClickTabsValue.content.recipelDetailEvtList.filter(

(item) => item.infuseGroup!==row.infuseGroup && this.medicalClickTabsValue.content.recipelDetailEvtList.indexOf(item)!==index

);

this.medicalClickTabsValue.infusion.infusionProject[row.infuseGroup-1].splice(index,1)

}

this.MedicalCalculate();

// console.log(row,'传说');

},

DeleteExMedicalRow(index,row){

this.medicalClickTabsValue.content.recipelDetailEvtList =

this.medicalClickTabsValue.content.recipelDetailEvtList.filter(

(item) => item.id!==row.id

);

this.medicalClickTabsValue.infusion.excharge.splice(index,1)

this.MedicalCalculate();

},

RowClickChineseTable(row) {

let recipelDetailEvtList =

this.medicalClickTabsValue.content.recipelDetailEvtList;

let hasFlag = false;

recipelDetailEvtList.forEach((element) => {

if (element.isExtra === 0) {

if (row.id === element.drugStuffId.drugStuffId) {

hasFlag = true;

}

}

});

if (hasFlag) {

this.$message.error("已有该药品");

return;

}

let drugStuff = {

drugStuffId: row.id,

name: row.goodsName,

price: row.price,

retailPrice: row.retailPrice,

isUnpackSell: row.isUnpackSell,

dosisUnit: row.dosisUnit,

preparationUnit: row.preparationUnit,

pack: row.pack,

drug: row,

};

let recipelDetailEvt = {

allFee: 0,

total: 0,

company: this.Company,

stuffType: 1,

isExtra: 0,

isUnpackSell: 0, //中药处方默认按不拆零销售开处方

drugStuffId: drugStuff,

singleDosage: 0,

};

this.medicalClickTabsValue.content.recipelDetailEvtList.push(

JSON.parse(JSON.stringify(recipelDetailEvt))

);

setTimeout(() => {

this.MedicalCalculate()

}, 0);

},

GetChineseFee(tabData, rowData) {

console.log(tabData);

console.log(tabData.content.recipelInfo,'1212');

if (rowData) {

let recipelInfo = tabData.content.recipelInfo;

tabData.content.recipelInfo.dosage = recipelInfo.dosage

? recipelInfo.dosage

: "";

rowData.singleDosage = rowData.singleDosage ? rowData.singleDosage : 0;

rowData.allFee =

rowData.drugStuffId.price \* rowData.singleDosage \* recipelInfo.dosage;

}

this.GetChineseSumFee(tabData);

},

GetCostItemTable() {

let SearchModel = {

columnName: "",

limit: 10,

offset: 0,

order: "",

params: [

{

columnName: "company\_id",

queryType: "=",

value: this.Company.id,

},

{

columnName: "item\_name",

queryType: "like",

value: "",

},

{

columnName: "item\_type",

queryType: "<>",

value: "treatmentItemType\_4",

},

],

};

var pattern2 = new RegExp("[A-Za-z]+");

if (pattern2.test(this.SearchCostItemInput)) {

// console.log(this.SearchWesternInput,'字符');

SearchModel.params[1].value = this.SearchCostItemInput.toUpperCase();

SearchModel.params[1].columnName = "pinyin\_code";

} else {

SearchModel.params[1].value = this.SearchCostItemInput;

SearchModel.params[1].columnName = "item\_name";

}

listCostItemAll(SearchModel).then((responseData) => {

if (responseData.code == 100) {

this.TreatmentTable = responseData.data;

}

});

},

RowClickTreatmentTable(row) {

let recipelDetailEvtList =

this.medicalClickTabsValue.content.recipelDetailEvtList;

let hasFlag = false;

recipelDetailEvtList.forEach((element) => {

if (element.isExtra === 0) {

if (row.id === element.drugStuffId.drugStuffId) {

hasFlag = true;

}

}

});

if (hasFlag) {

return;

}

let drugStuff = {

drugStuffId: row.id,

name: row.itemName,

price: row.salePrice,

isUnpackSell: 0,

pack: row.unit,

costItem: row,

};

let recipelDetailEvt = {

allFee: 0,

total: 0,

company: this.Company,

stuffType: 3,

isExtra: 0,

isUnpackSell: 0, //诊疗处方默认按不拆零销售开处方

drugStuffId: drugStuff,

};

this.medicalClickTabsValue.content.recipelDetailEvtList.push(

JSON.parse(JSON.stringify(recipelDetailEvt))

);

},

GetCostItemFee(index, row) {

row.allFee = row.drugStuffId.price \* row.total;

},

GetSurchargeTable() {

this.SurchargeTable = [];

this.GetSurchargeCostItemTable();

this.GetSurchargeStuffTable();

console.log(this.SurchargeTable);

},

GetSurchargeCostItemTable() {

let SearchModel = {

columnName: "",

limit: 10,

offset: 0,

order: "",

params: [

{

columnName: "company\_id",

queryType: "=",

value: this.Company.id,

},

{

columnName: "item\_name",

queryType: "like",

value: "",

},

{

columnName: "item\_type",

queryType: "=",

value: "treatmentItemType\_4",

},

],

};

var pattern3 = new RegExp("[A-Za-z]+");

if (pattern3.test(this.SearchSurchargeInput)) {

SearchModel.params[1].value = this.SearchSurchargeInput.toUpperCase();

SearchModel.params[1].columnName = "pinyin\_code";

} else {

//console.log(this.SearchSurchargeInput,'字符');

SearchModel.params[1].value = this.SearchSurchargeInput;

SearchModel.params[1].columnName = "item\_name";

}

listCostItemAll(SearchModel).then((responseData) => {

if (responseData.code == 100) {

responseData.data.forEach((element) => {

element["stuffType"] = "3";

this.SurchargeTable.push(element);

});

}

});

},

GetSurchargeStuffTable() {

let SearchModel = {

columnName: "",

limit: 10,

offset: 0,

order: "",

params: [

{

columnName: "company\_id",

queryType: "=",

value: this.Company.id,

},

{

columnName: "name",

queryType: "like",

value: "",

},

{

columnName: "is\_out\_sell",

queryType: "=",

value: "1",

},

{

columnName: "inventory",

queryType: ">",

value: 0,

},

],

};

var pattern2 = new RegExp("[A-Za-z]+");

if (pattern2.test(this.SearchSurchargeInput)) {

// console.log(this.SearchWesternInput,'字符');

SearchModel.params[1].value = this.SearchSurchargeInput.toUpperCase();

SearchModel.params[1].columnName = "pinyin\_code";

} else {

SearchModel.params[1].value = this.SearchSurchargeInput;

SearchModel.params[1].columnName = "name";

}

listStuffAll(SearchModel).then((responseData) => {

if (responseData.code == 100) {

responseData.data.forEach((element) => {

element["stuffType"] = "4";

this.SurchargeTable.push(element);

});

}

});

},

RowClickSurchargeTable(row) {

let recipelDetailEvtList =

this.medicalClickTabsValue.content.recipelDetailEvtList;

let hasFlag = false;

recipelDetailEvtList.forEach((element) => {

if (element.isExtra === 1) {

if (row.id === element.drugStuffId.drugStuffId) {

hasFlag = true;

}

}

});

if (hasFlag) {

return;

}

let stuffType = row.stuffType;

let drugStuff = {};

if (stuffType === "3") {

drugStuff = {

drugStuffId: row.id,

name: row.itemName,

price: row.salePrice,

isUnpackSell: 0,

pack: row.unit,

costItem: row,

};

} else if (stuffType === "4") {

drugStuff = {

drugStuffId: row.id,

name: row.name,

price: row.priceOutSell,

retailPrice: row.retailPrice,

isUnpackSell: row.isUnpackSell

? BigNumber(row.isUnpackSell).toNumber()

: 0,

preparationUnit: row.minUnit,

pack: row.packUnit,

stuff: row,

};

}

let recipelDetailEvt = {

allFee: 0,

total: 0,

singleDosage: 0,

company: this.Company,

stuffType: stuffType,

isExtra: 1,

isUnpackSell: 0,

drugStuffId: drugStuff,

};

if(this.medicalClickTabsValue.type=="recipelType\_2"){

this.medicalClickTabsValue.infusion.excharge.push(JSON.parse(JSON.stringify(recipelDetailEvt)))

}else{

this.medicalClickTabsValue.content.recipelDetailEvtList.push(

JSON.parse(JSON.stringify(recipelDetailEvt))

);

}

},

getDataFilterTable(data, isExtra,item) {

let arr=[]

if(data.length==undefined){

arr.push(data)

}else if(data.length>=1){

for (let i = 0; i < data.length; i++) {

if(data[i].isExtra==0&&data[i].drugStuffId.drug){

if(data[i].drugStuffId.drug.type.value=='medicalType\_0'){

// if(data[i].singleDosage==0&&data[i].singleDosage==" "&&data[i].drugStuffId.drug.singleDosage){

// data[i].singleDosage = data[i].drugStuffId.drug.singleDosage

// }

//console.log(item,'这是一个新址');

if(data[i].singleDosage==0&&data[i].singleDosage==" "&&data[i].drugStuffId.drug.singleDosage){

let flages=false;

for (let i = 0; i < item.costDebug.chineseTest.length; i++) {

if(item.costDebug.chineseTest[i]==data[i].drugStuffId.drug.name){

flages=true;

break;

}

}

if(!flages){

data[i].singleDosage = data[i].drugStuffId.drug.singleDosage?data[i].drugStuffId.drug.singleDosage:""

item.costDebug.chineseTest.push(data[i].drugStuffId.drug.name)

}

}

if(!data[i].frequency&&data[i].drugStuffId.drug.frequency.value){

data[i].frequency = data[i].drugStuffId.drug.frequency

}

if(!data[i].days&&data[i].drugStuffId.drug.days.value){

data[i].days = data[i].drugStuffId.drug.days

}

if(!data[i].westernMedicineUse&&data[i].drugStuffId.drug.westernMedicineUse.value){

data[i].westernMedicineUse = data[i].drugStuffId.drug.westernMedicineUse

}

}

if(data[i].drugStuffId.drug.type.value=='medicalType\_1'){

let flages=false

if(data[i].singleDosage==0&&data[i].singleDosage==" "&&data[i].drugStuffId.drug.singleDosage){

for (let i = 0; i < item.costDebug.chineseTest.length; i++) {

if(item.costDebug.chineseTest[i]==data[i].drugStuffId.drug.name){

flages=true;

break;

}

}

if(!flages){

data[i].singleDosage = data[i].drugStuffId.drug.singleDosage?data[i].drugStuffId.drug.singleDosage:""

item.costDebug.chineseTest.push(data[i].drugStuffId.drug.name)

}

}

if(!data[i].chineseMedicineUse&&data[i].drugStuffId.drug.chineseMedicineUse.value){

data[i].chineseMedicineUse = data[i].drugStuffId.drug.chineseMedicineUse

}

}

}

arr.push(data[i])

}

}

// this.MedicalCalculate()

// if(data.length>1){

// if(data[0].infuseGroup!=undefined){

// return data.filter((item) => item.isExtra === isExtra);

// }else{

// return data.filter((item) => item.isExtra === isExtra);

// }

// }

return arr.filter((item) => item.isExtra === isExtra);

},

GetSurchargeFee(index, row) {

row.singleDosage = row.singleDosage ? row.singleDosage : 0;

if (row.isUnpackSell == "1") {

row.total = row.singleDosage;

row.allFee = row.total \* row.drugStuffId.retailPrice;

} else {

if (row.stuffType === "4") {

row.total = row.singleDosage \* row.drugStuffId.stuff.packNumber;

} else if (row.stuffType === "3") {

row.total = row.singleDosage;

}

row.allFee = row.singleDosage \* row.drugStuffId.price;

}

},

MedicalCalculate() {

let n1 = 0; //西药处方个数

let n2 = 0; //中药处方个数

let n3 = 0; //输液处方个数

let n4 = 0; //诊疗项目处方个数

let seq = 0; //处方顺序

let fullAmount = 0; //所有处方的费用汇总

this.medicalEditTabs.forEach((tabElement) => {

let medicalFullAmount = 0; //处方所有费用 = 处方费用汇总 + 附加费汇总

let medicalAmount = 0; //处方费用汇总

let surchargeAmount = 0; //附加费汇总

let detailSeq = 0; //处方明细顺序

let recipelInfo = tabElement.content.recipelInfo;

let recipelTypeName = recipelInfo.recipelType.name;

let recipelDetailEvtList = tabElement.content.recipelDetailEvtList

? tabElement.content.recipelDetailEvtList

: [];

seq++;

//西药处方

if (tabElement.type === "recipelType\_0") {

n1++;

tabElement.title = recipelTypeName + n1;

recipelDetailEvtList.forEach((rowElement) => {

if (rowElement.isExtra === 0) {

detailSeq++;

rowElement.seq = detailSeq;

rowElement.singleDosage = rowElement.singleDosage

? rowElement.singleDosage

: "";

rowElement.total = rowElement.total ? rowElement.total : 0;

rowElement.allFee = rowElement.allFee ? rowElement.allFee : 0;

if (rowElement.frequency && rowElement.days) {

let total = Math.ceil(

BigNumber(rowElement.singleDosage-0)

.multipliedBy(rowElement.frequency.value.split("\_")[1])

.multipliedBy(rowElement.days.name)

.toNumber()

);

if (rowElement.isUnpackSell == "1") {

rowElement.unitPrice = rowElement.drugStuffId.retailPrice;

rowElement.total = total;

rowElement.allFee = BigNumber(rowElement.total)

.multipliedBy(rowElement.drugStuffId.retailPrice)

.toNumber();

} else {

rowElement.unitPrice = rowElement.drugStuffId.price;

rowElement.total = BigNumber(

Math.ceil(

BigNumber(total)

.dividedBy(rowElement.drugStuffId.drug.preparation)

.toNumber()

)

)

.multipliedBy(rowElement.drugStuffId.drug.preparation)

.toNumber();

rowElement.allFee = BigNumber(

Math.ceil(

BigNumber(total)

.dividedBy(rowElement.drugStuffId.drug.preparation)

.toNumber()

)

)

.multipliedBy(rowElement.drugStuffId.price)

.toNumber();

}

}

if (rowElement.drugStuffId.drug.inventory < rowElement.total&&!this.isReadOnly&&tabElement.content.recipelInfo.dispensionStatus==0) {

this.$message.error(

rowElement.drugStuffId.name +

"库存不足，请修改数量或者添加库存！"

);

}

medicalAmount = BigNumber(medicalAmount)

.plus(rowElement.allFee)

.toNumber();

}

});

medicalFullAmount = BigNumber(medicalFullAmount)

.plus(medicalAmount)

.toNumber();

}

//中药处方

else if (tabElement.type === "recipelType\_1") {

n2++;

tabElement.title = recipelTypeName + n2;

recipelInfo.dosage = recipelInfo.dosage ? recipelInfo.dosage : "";

recipelDetailEvtList.forEach((rowElement) => {

if (rowElement.isExtra === 0) {

detailSeq++;

rowElement.seq = detailSeq;

if(rowElement.singleDosage==" "||rowElement.singleDosage==""){

if(recipelInfo.dosage==""){

rowElement.singleDosage=""

rowElement.total = BigNumber(0)

.multipliedBy(0)

.multipliedBy(rowElement.drugStuffId.drug.preparation)

.toNumber();

rowElement.allFee = BigNumber(rowElement.drugStuffId.price)

.multipliedBy(0)

.multipliedBy(0)

.toNumber();

rowElement.unitPrice = rowElement.drugStuffId.price;

medicalAmount = BigNumber(medicalAmount)

.plus(rowElement.allFee)

.toNumber();

}else{

rowElement.singleDosage=""

rowElement.total = BigNumber(0)

.multipliedBy(recipelInfo.dosage)

.multipliedBy(rowElement.drugStuffId.drug.preparation)

.toNumber();

rowElement.allFee = BigNumber(rowElement.drugStuffId.price)

.multipliedBy(0)

.multipliedBy(recipelInfo.dosage)

.toNumber();

rowElement.unitPrice = rowElement.drugStuffId.price;

medicalAmount = BigNumber(medicalAmount)

.plus(rowElement.allFee)

.toNumber();

}

}else{

if(recipelInfo.dosage==""){

rowElement.singleDosage = rowElement.singleDosage

? rowElement.singleDosage-0

: 0;

rowElement.total = BigNumber(rowElement.singleDosage)

.multipliedBy(0)

.multipliedBy(rowElement.drugStuffId.drug.preparation)

.toNumber();

rowElement.allFee = BigNumber(rowElement.drugStuffId.price)

.multipliedBy(rowElement.singleDosage)

.multipliedBy(0)

.toNumber();

rowElement.unitPrice = rowElement.drugStuffId.price;

medicalAmount = BigNumber(medicalAmount)

.plus(rowElement.allFee)

.toNumber();

}else{

rowElement.singleDosage = rowElement.singleDosage

? rowElement.singleDosage-0

: 0;

rowElement.total = BigNumber(rowElement.singleDosage)

.multipliedBy(recipelInfo.dosage)

.multipliedBy(rowElement.drugStuffId.drug.preparation)

.toNumber();

rowElement.allFee = BigNumber(rowElement.drugStuffId.price)

.multipliedBy(rowElement.singleDosage)

.multipliedBy(recipelInfo.dosage)

.toNumber();

rowElement.unitPrice = rowElement.drugStuffId.price;

medicalAmount = BigNumber(medicalAmount)

.plus(rowElement.allFee)

.toNumber();

}

}

if (rowElement.drugStuffId.drug.inventory < rowElement.total&&!this.isReadOnly&&tabElement.content.recipelInfo.dispensionStatus==0) {

this.$message.error(

rowElement.drugStuffId.name +

"库存不足，请修改数量或者添加库存！"

);

}

}

});

medicalFullAmount = BigNumber(medicalFullAmount)

.plus(medicalAmount)

.toNumber();

}

//输液处方

else if (tabElement.type === "recipelType\_2") {

n3++;

tabElement.title = recipelTypeName + n3;

// tabElement.content.recipelDetailEvtList=this.infusionProject

let arr=0;

tabElement.content.recipelDetailEvtList=[]

for (let i = 0; i < tabElement.infusion.infusionProject.length; i++) {

for (let j = 0; j < tabElement.infusion.infusionProject[i].length; j++) {

tabElement.infusion.infusionProject[i][j].drippingSpeed=tabElement.infusion.drippingSpeed[tabElement.infusion.infusionProject[i][j].infuseGroup-1]

tabElement.infusion.infusionProject[i][j].days=tabElement.infusion.days[tabElement.infusion.infusionProject[i][j].infuseGroup-1]

tabElement.infusion.infusionProject[i][j].frequency=tabElement.infusion.frequency[tabElement.infusion.infusionProject[i][j].infuseGroup-1]

tabElement.infusion.infusionProject[i][j].infuseUse=tabElement.infusion.infuseUse[tabElement.infusion.infusionProject[i][j].infuseGroup-1]

tabElement.content.recipelDetailEvtList[arr]= tabElement.infusion.infusionProject[i][j]

arr++;

}

}

// let countArr=0

// countArr=arr

// for (let i = 0; i < this.excharge.length; i++) {

// tabElement.content.recipelDetailEvtList.push(this.excharge[i])

// // tabElement.content.recipelDetailEvtList[countArr+1]=this.excharge[i]

// //countArr++;

// }

for (let i = 0; i < tabElement.infusion.excharge.length; i++) {

tabElement.content.recipelDetailEvtList.push(tabElement.infusion.excharge[i])

}

recipelDetailEvtList = tabElement.content.recipelDetailEvtList

? tabElement.content.recipelDetailEvtList

: [];

console.log(this.medicalEditTabs,'kankan');

// recipelInfo.frequency = recipelInfo.frequency ? recipelInfo.frequency : 0;

let infusionRecipelDetailEvtList=tabElement.content.recipelDetailEvtList

infusionRecipelDetailEvtList.forEach((rowElement) => {

if (rowElement.isExtra === 0) {

detailSeq++;

rowElement.seq = detailSeq;

rowElement.singleDosage = rowElement.singleDosage

? rowElement.singleDosage

: "";

rowElement.total = rowElement.total ? rowElement.total : 0;

rowElement.allFee = rowElement.allFee ? rowElement.allFee : 0;

if (rowElement.frequency.value && rowElement.days.name) {

let total = Math.ceil(

BigNumber(rowElement.singleDosage-0)

.multipliedBy(rowElement.frequency.value.split("\_")[1])

.multipliedBy(rowElement.days.name)

.toNumber()

);

if (rowElement.isUnpackSell == "1") {

rowElement.unitPrice = rowElement.drugStuffId.retailPrice;

rowElement.total = total;

rowElement.allFee = BigNumber(rowElement.total)

.multipliedBy(rowElement.drugStuffId.retailPrice)

.toNumber();

} else {

rowElement.unitPrice = rowElement.drugStuffId.price;

rowElement.total = BigNumber(

Math.ceil(

BigNumber(total)

.dividedBy(rowElement.drugStuffId.drug.preparation)

.toNumber()

)

)

.multipliedBy(rowElement.drugStuffId.drug.preparation)

.toNumber();

rowElement.allFee = BigNumber(

Math.ceil(

BigNumber(total)

.dividedBy(rowElement.drugStuffId.drug.preparation)

.toNumber()

)

)

.multipliedBy(rowElement.drugStuffId.price)

.toNumber();

}

}

if (rowElement.drugStuffId.drug.inventory < rowElement.total&&!this.isReadOnly&&tabElement.content.recipelInfo.dispensionStatus==0) {

this.$message.error(

rowElement.drugStuffId.name +

"库存不足，请修改数量或者添加库存！"

);

}

medicalAmount = BigNumber(medicalAmount)

.plus(rowElement.allFee)

.toNumber();

}

});

medicalFullAmount = BigNumber(medicalFullAmount)

.plus(medicalAmount)

.toNumber();

}

//诊疗项目处方

else if (tabElement.type === "recipelType\_3") {

n4++;

tabElement.title = recipelTypeName + n4;

recipelDetailEvtList.forEach((rowElement) => {

if (rowElement.isExtra === 0) {

detailSeq++;

rowElement.seq = detailSeq;

rowElement.total = rowElement.total ? rowElement.total : "";

rowElement.allFee = BigNumber(rowElement.drugStuffId.price)

.multipliedBy(rowElement.total-0)

.toNumber();

rowElement.unitPrice = rowElement.drugStuffId.price;

medicalAmount = BigNumber(medicalAmount)

.plus(rowElement.allFee)

.toNumber();

}

});

medicalFullAmount = BigNumber(medicalFullAmount)

.plus(medicalAmount)

.toNumber();

}

//每个处方的附加费

recipelDetailEvtList.forEach((rowElement) => {

console.log('精神不好估计按时归还健康卡');

// if(rowElement.infuseGroup==undefined){

if (rowElement.isExtra === 1) {

detailSeq++;

rowElement.seq = detailSeq;

rowElement.singleDosage = rowElement.singleDosage&&rowElement.singleDosage!=0

? rowElement.singleDosage

: "";

if (rowElement.isUnpackSell == "1") {

rowElement.total = rowElement.singleDosage-0;

rowElement.allFee = BigNumber(rowElement.drugStuffId.retailPrice)

.multipliedBy(rowElement.total)

.toNumber();

rowElement.unitPrice = rowElement.drugStuffId.retailPrice;

console.log(rowElement, "奇怪");

if (rowElement.drugStuffId.stuff.inventory < rowElement.total&&!this.isReadOnly&&tabElement.content.recipelInfo.dispensionStatus==0) {

this.$message.error(

rowElement.drugStuffId.name +

"库存不足，请修改数量或者添加库存！"

);

}

} else {

//材料

if (rowElement.stuffType == "4") {

rowElement.total = BigNumber(rowElement.singleDosage-0)

.multipliedBy(rowElement.drugStuffId.stuff.packNumber)

.toNumber();

if (rowElement.drugStuffId.stuff.inventory < rowElement.total&&!this.isReadOnly&&tabElement.content.recipelInfo.dispensionStatus==0) {

this.$message.error(

rowElement.drugStuffId.name +

"库存不足，请修改数量或者添加库存！"

);

}

}

//诊疗项目

else if (rowElement.stuffType == "3") {

rowElement.total = rowElement.singleDosage-0;

}

rowElement.unitPrice = rowElement.drugStuffId.price;

rowElement.allFee = BigNumber(rowElement.singleDosage-0)

.multipliedBy(rowElement.drugStuffId.price)

.toNumber();

}

surchargeAmount = BigNumber(surchargeAmount)

.plus(rowElement.allFee)

.toNumber();

}

//}

});

medicalFullAmount = BigNumber(medicalFullAmount)

.plus(surchargeAmount)

.toNumber();

recipelInfo.name = tabElement.title;

recipelInfo.seq = seq;

recipelInfo.medicalAmount = medicalAmount;

recipelInfo.surchargeAmount = surchargeAmount;

recipelInfo.fee = medicalFullAmount;

fullAmount = BigNumber(fullAmount).plus(recipelInfo.fee).toNumber();

});

this.payAmount = fullAmount;

},

FinishVisit() {

let FormCheck = true;

this.$refs["BasicInfoForm"].validate((valid) => {

if (valid) {

//alert('submit!');

} else {

console.log("error submit!!");

FormCheck = false;

return false;

}

});

if (!FormCheck) {

return;

}

let recipelInfoEvtList = [];

console.log(this.medicalEditTabs);

let returnForEach = false;

let hasSelsmallType = false;

let numberCheck = false;

if (this.medicalEditTabs.length == 0) {

this.$message.error("处方数量不能为0");

return;

}

this.medicalEditTabs.forEach((tabElement) => {

//西药处方

if (

tabElement.type == "recipelType\_0" &&

!this.MedicalRecordModel.westernDiagnose

) {

returnForEach = true;

}

//中药处方

if (

tabElement.type == "recipelType\_1" &&

!this.MedicalRecordModel.chinaDiagnose

) {

returnForEach = true;

}

if (

!tabElement.content.recipelInfo.smallType &&

tabElement.type != "recipelType\_3"

) {

hasSelsmallType = true;

}

if (tabElement.content.recipelDetailEvtList.length == 0) {

numberCheck = true;

}

recipelInfoEvtList.push(tabElement.content);

});

if (returnForEach) {

this.$message.error("存在西药处方或中药处方，请填写对应诊断说明。");

return;

}

if (hasSelsmallType) {

this.$message.error("请检查每个处方里的分类是否已选择。");

return;

}

if (numberCheck) {

this.$message.error("请检查每个处方里是否有药品。");

return;

}

this.MedicalRecordModel.company = this.Company;

this.MedicalRecordModel.doctor.id = this.UserId;

let medicalRecipelEvt = {

medicalRecord: this.MedicalRecordModel,

patient: this.BasicInfoModel,

recipelInfoEvtList: recipelInfoEvtList,

isInstant: this.isInstant,

};

console.log(medicalRecipelEvt);

let formData = new FormData();

formData.append("entity", JSON.stringify(medicalRecipelEvt));

this.hint=false

allSaveMedicalRecord(formData).then((responseData) => {

if (responseData.code == 100) {

this.$message.success("操作成功！");

this.hasBtnFinishVisit = false;

this.hasBtnPrint = false;

this.isReadOnly = true;

this.GetPreparePatientList();

this.GetAlreadyPatientList();

} else {

this.$message.error("完成接诊失败！");

}

});

},

invalidMedical() {

console.log(this.medicalClickTabsValue);

if (!this.medicalClickTabsValue.content||!this.medicalClickTabsValue.content.recipelInfo.id) {

this.$message.error("未选择处方");

return;

}

let id = this.medicalClickTabsValue.content.recipelInfo.id;

invalidStatus(id).then((responseData) => {

if (responseData.code == 100) {

this.$message.success("处方作废成功！");

this.AlreadyPatientDeletilQuery();

} else {

this.$message.error("处方作废失败！");

}

});

},

checkInventory(row) {

console.log(row.total, "当前行");

// alert('hahahahahhahaha')

if (row.total > row.drugStuffId.drug.inventory) {

//

this.$message.error("药品超出库存总量，请重新输入或者入库药品!");

}

},

},

computed: {

Company() {

let company = JSON.parse(sessionStorage.getItem("currentUser")).company;

return {

id: company.id,

label: company.label,

name: company.name,

};

},

UserId() {

return JSON.parse(sessionStorage.getItem("currentUser")).id;

},

},

mounted() {

let specialCompany = JSON.parse(sessionStorage.getItem("currentCompany"));

if (specialCompany && specialCompany.id == "1088657523871555640") {

this.isSpecial = true;

}

this.GetAllOption();

this.GetMedicalRecordAll();

this.GetPreparePatientList();

this.GetAlreadyPatientList();

this.TodayActiveName = "prepare";

this.addMedicalEditTab(this.medicalTypeList[0]);

console.log(BigNumber(6).multipliedBy(2.8).toNumber());

console.log(BigNumber(6).multipliedBy("2.8").toNumber());

},

watch: {

checkInventory: function (row) {

console.log(row, "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa");

},

},

};

</script>

<style >

.el-input-number{

width: 1px;

line-height:10px;

height: 28px;

}

</style>

<style scoped>

/deep/ .el-aside {

/\* background-color: #e6d6d3; \*/

color: #333;

/\* text-align: center; \*/

/\* line-height: 200px; \*/

margin-right: 5px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.12), 0 0 6px rgba(0, 0, 0, 0.04);

height: calc(100vh - 118px);

}

/deep/ .el-main {

/\* background-color: #E9EEF3; \*/

color: #333;

/\* text-align: center; \*/

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.12), 0 0 6px rgba(0, 0, 0, 0.04);

/\* line-height: 200px; \*/

height: calc(100vh - 118px);

}

/deep/ .el-card\_\_header {

padding: 12px 20px;

}

/deep/ .el-form-item--mini {

margin-bottom: 8px !important;

}

/deep/ .main-card {

margin-bottom: 10px;

}

/deep/ .main-card > .el-card\_\_header {

padding: 0px 20px;

}

/deep/ .main-card > .el-card\_\_body {

font-size: 14px;

}

/deep/ .sub-card > .el-card\_\_body {

padding: 12px 20px;

font-size: 16px;

font-weight: bold;

color: midnightblue;

}

/deep/ .chinese-medicine-card > .el-card\_\_header {

padding: 0px 20px;

}

/deep/ .el-divider--horizontal {

margin: 5px 0 10px 0px;

}

/deep/ .el-tabs\_\_header {

margin: 0 0 0;

}

/deep/ .el-collapse-item\_\_content {

padding-bottom: 0;

}

/deep/ .el-divider--vertical {

width: 5px;

height: 2em;

margin-left: 0;

background-color: #409eff;

}

/deep/ .el-input-group\_\_append {

padding: 4px;

}

.medical-type-ul {

list-style-type: none;

font-size: 13px;

color: #303133;

margin: 5px;

padding: 0;

}

.aside-header {

font-weight: bold;

line-height: 10px;

height: 10px;

font-size: 14px;

}

/\* .box-card {

width: 100%;

height: 100%;

} \*/

.el-row {

margin-bottom: 8px;

}

.el-row:last-child {

margin-bottom: 0;

}

.already-medocal {

border-bottom: 1px solid #dcdfe6;

margin-right: 15px;

}

.refundBox {

position: absolute;

top: -23px;

right: -80px;

width: 80px;

height: 38px;

border: 2px solid #e6a23c;

text-align: center;

font-size: 18px;

border-radius: 80%;

color: #e6a23c;

transform: scale(0.5);

}

</style>

<style>

.el-popover.medical-type-popover {

min-width: 85px !important ;

margin: 0;

padding: 0;

}

.el-popover.medical-edit-popover {

margin: 0;

padding: 0;

}

.ipt-box .el-input--mini {

width: 70%;

}

.ipt-box {

display: flex;

align-items: center;

}

</style>