第四节、图书借阅数据分析系统客户端设计与实现

4.1.hbuilder开发工具下载与安装

4.2.node服务器下载与安装

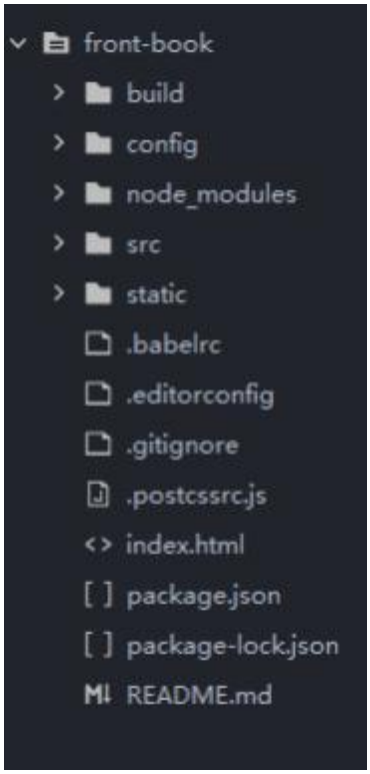
1.3.1.node.js服务器安装 访问网址:http://nodejs.cn/

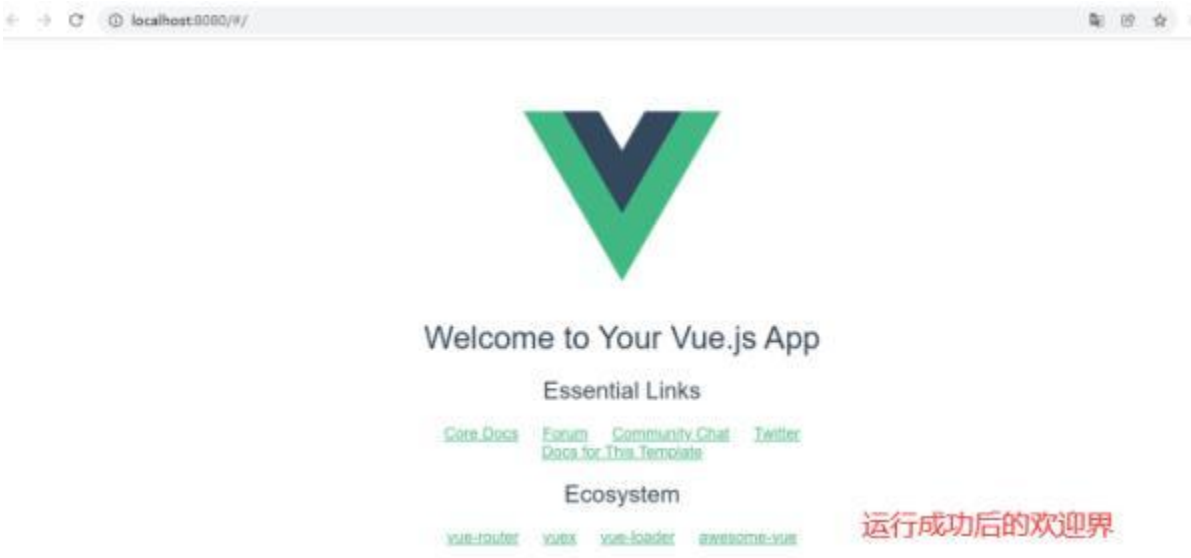
4.3.vueCli安装

```
/**npm简介
node.js服务器安装 访问网址:http://nodejs.cn/9.3.vueCli简介
NPM是随同NodeJS一起安装的包管理工具，能解决NodeJS代码部署上的很多问题，常见的使用场景有以下几种：
允许用户从NPM服务器下载别人编写的第三方包到本地使用。
允许用户从NPM服务器下载并安装别人编写的命令行程序到本地使用。
允许用户将自己编写的包或命令行程序上传到NPM服务器供别人使用。
由于新版的nodejs已经集成了npm，所以之前npm也一并安装好了。
同样可以通过输入 "npm -v" 来测试是否成功安 装。命令如下，出现版本提示表示安装成功*/
```

```
/**
Vue CLI 是一个基于 Vue.js 进行快速开发的完整系统，通过 @vue/cli 快速创建前端工程
配置阿里加速地址： npm install -g cnpm --registry=https://registry.npm.taobao.org
安装命令 npm install -g @vue/cli
查看版本命令vue --version
使用webpack创建项目
步骤1 进入要创建项目的盘符或者文件夹输入如下命令: vue init webpack 工程名称
*/
```

4.4客户端工程目录结构





第五节、图书借阅数据分析系统工程前端架构搭建

5.1.Element概述

```
Element，一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库，官网：
https://element.eleme.cn/#/zh-CN
```

5.2.Element开发环境构建

```
//开发环境安装进入工程目录运行如下命令: npm i element-ui -S
//在main.js中引入如下命令
import ElementUI from 'element-ui';
import 'element-ui/lib/theme-chalk/index.css';
Vue.use(ElementUI);
```

5.3.axios概述

```
//axios是vue框架提倡使用的ajax库
//axios安装 进入工程目录使用 npm install axios -S命令安装
//main.js中引用
import axios from 'axios'
Vue.prototype.$http=axios
```

5.4.样式引入

```
//在工程静态资源中编写main.css
html,
body,
#app {
margin: 0px;
height: 100%;
padding: 0px;
}
/
/在main.js中引入
import './assets/css/main.css'
```

5.5.局部组件Main.vue创建

```
<template>
<el-container>
<el-header>疫苗预约管理系统</el-header>
<el-container>
<el-aside width="200px">导航菜单</el-aside>
<el-main>主体</el-main>
</el-container>
</el-container>
</template>
<script>
export default {
data() {
return {}
}
}
```

```
</script>
<style scoped>
.el-header {
background-color: #2B4B6B;
display: flex;
justify-content: space-between;
padding-left: 10px;
align-items: center;
color: white;
font-size: 18px;
} .
el-aside {
color: white;
background-color: #2B4B6B;
} .
el-main {
background-color: aliceblue;
} .
el-container {
height: 100%;}
</style>
```

5.6.导航菜单设计与实现

```
//在路由(router)文件夹中的index.js中编写如下代码
import Vue from 'vue'
import Router from 'vue-router'
import Main from '../components/Main.vue'
import Category from '../components/Category.vue'
import Book from '../components/Book.vue'
import Logs from '../components/Logs.vue'
import Borrow from '../components/Borrow.vue'
import Users from '../components/Users.vue'
import Welcome from '../components/Welcom.vue'
Vue.use(Router)

export default new Router({
  routes: [
    {
```

```
      path: '/',
      name: 'Main',
      component: Main,
      redirect : '/welcome',
      children:[
        {
          path:'/welcome',
          name: Welcome,
          component: Welcome
        },
        {
        path: '/category',
        name: 'Category',
        component: Category,
      },{
        path: '/book',
        name: 'book',
        component: Book,
      },{
        path: '/borrow',
        name: 'Borrow',
        component: Borrow,
      },{
        path: '/users',
        name: 'Users',
        component: Users,
      },{
        path: '/logs',
        name: 'Logs',
        component: Logs,
      }]
    }
  ]
})
```

```
<template>
  <el-container>
    <el-header>图书借阅管理系统</el-header>
    <el-container>
      <el-aside width="200px">
        <el-menu router background-color="#2B4B6B" text-color="#fff"
         active-text-color="#409EFF" unique-opened>
         <el-submenu index="3">
           <template slot="title">
             <i class="el-icon-s-tools"></i>
             <span>馆藏类别管理</span>
           </template>
           <el-menu-item index="/category">
             <i class="el-icon-menu"></i>
             <span slot="title">类别信息管理</span>
           </el-menu-item>=
           <el-menu-item index="/book">
             <i class="el-icon-reading"></i>
             <span slot="title">馆藏图书管理</span>
```

```
            </el-menu-item>
          </el-submenu>
          <el-submenu index="2">
            <template slot="title">
              <i class="el-icon-s-cooperation"></i>
              <span>借阅信息管理</span>
            </template>
            <el-menu-item index="/borrow">
              <i class="el-icon-files"></i>
              <span slot="title">借阅信息管理</span>
            </el-menu-item>
          </el-submenu>
           <el-submenu index="1">
             <template slot="title">
               <i class="el-icon-s-platform"></i>
               <span>系统信息管理</span>
             </template>
             <el-menu-item index="/users">
               <i class="el-icon-user"></i>
               <span slot="title">用户信息管理</span>
             </el-menu-item>
             <el-menu-item index="/logs">
               <i class="el-icon-document"></i>
               <span slot="title">日志信息管理</span>
             </el-menu-item>
           </el-submenu>
        </el-menu>
      </el-aside>
      <el-main>
        <!-- 路由占位符 -->
        <router-view/>
      </el-main>
    </el-container>
  </el-container>
</template>

<script>
  export default {
    data() {
      return {

      }
    }
  }
</script>

<style scoped>
  .el-header {
    background-color: #2B4B6B;
    display: flex;
    justify-content: space-between;
    padding-left: 10px;
    align-items: center;
    color: white;
    font-size: 18px;
```

```
  }

  .el-aside {
    color: white;
    background-color: #2B4B6B;
  }

  .el-main {
    background-color: aliceblue;
  }

  .el-container {
    height: 100%;
  }
</style>
```

第六节、图书借阅数据分析系统前端各个UI模块设计与实现

6.1.馆藏图书模块UI设计与实现


```
<template>
  <!-- 卡片组件 -->
  <el-card>
    <!-- 面包屑组件 -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/welcome' }">首页</el-breadcrumb-item>
      <el-breadcrumb-item>馆藏类别管理</el-breadcrumb-item>
      <el-breadcrumb-item>图书信息管理</el-breadcrumb-item>
    </el-breadcrumb>
    <el-row :gutter="10">
      <el-col :span="6">
        <el-input clearable @change="listPage()" v-model="book.bname"
placeholder="请输入图书名称进行查询"></el-input>
      </el-col>
      <el-col :span="6">
        <el-input clearable @change="listPage()" v-model="book.author"
placeholder="请输入图书作者进行查询"></el-input>
      </el-col>
      <el-col :span="12">
        <el-button icon="el-icon-plus" @click="openAddDialog" type="success">添加
</el-button>
      </el-col>
    </el-row>
    <!-- 表格组件 -->
    <el-table :data="tableData">
      <el-table-column type="expand" prop="ddesc">
        <template slot-scope="scope">
          <el-tag>简介：{{scope.row.ddesc}}</el-tag>
        </template>
      </el-table-column>
      <el-table-column prop="bnum" label="书号">
      </el-table-column>
      <el-table-column prop="bname" label="图书书名">
        <template slot-scope="scope">
          <el-popover trigger="hover" placement="top">
            <p>图书简介: {{ scope.row.ddesc }}</p>
            <p>库存数量: {{ scope.row.quantity }}</p>
            <div slot="reference" class="name-wrapper">
```

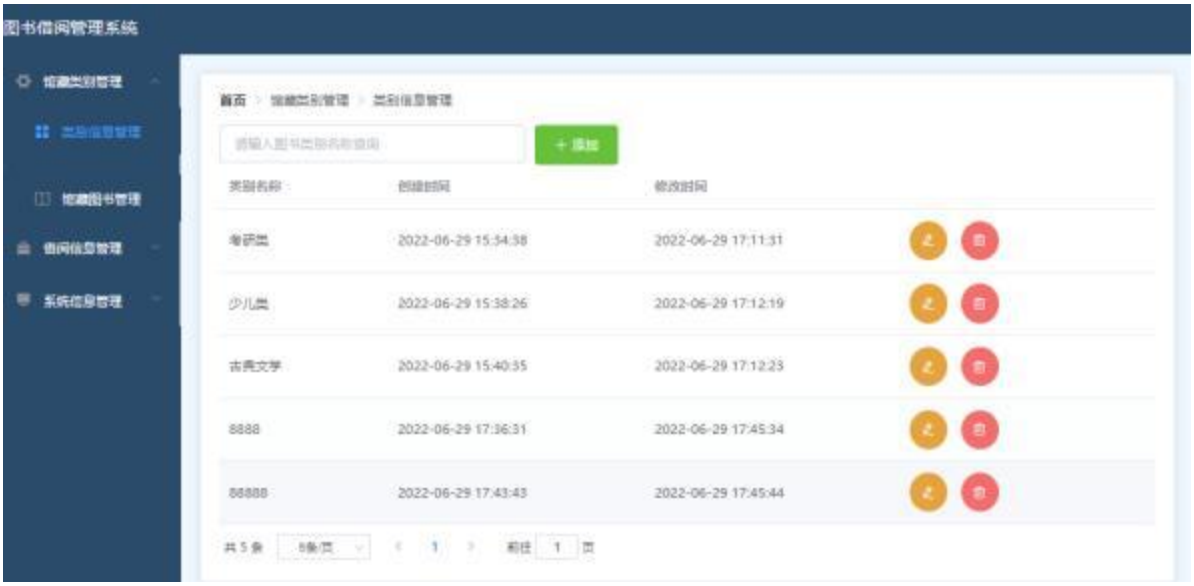
```
              <el-tag size="medium">{{ scope.row.bname }}</el-tag>
            </div>
          </el-popover>
        </template>
      </el-table-column>
      <el-table-column prop="author" label="图书作者">
      </el-table-column>

      <el-table-column prop="price" label="单价">
      </el-table-column>

      <el-table-column prop="category" label="所属类别">
        <template slot-scope="scope">
          {{scope.row.category.catename}}
        </template>
      </el-table-column>
      <el-table-column lable="操作">
        <!-- 作用域插槽 -->
        <template slot-scope="scope">
          <el-tooltip content="点击修改" placement="bottom">
            <el-button type="warning" @click="openEditDialog(scope.row)"
icon="el-icon-edit" circle></el-button>
          </el-tooltip>
          <el-tooltip content="点击删除" placement="bottom">
            <el-button type="danger" @click="remove(scope.row)" icon="el-icon-
delete" circle></el-button>
          </el-tooltip>
        </template>
      </el-table-column>
    </el-table>
    <!-- 分页组件定义 -->
    <el-pagination @size-change="handleSizeChange" @current-
change="handleCurrentChange" :current-page="pager.page"
      :page-sizes="[5, 10, 15, 20]" :page-size="pager.size" layout="total,
sizes, prev, pager, next, jumper"
      :total="pager.total">
    </el-pagination>

    <!-- 添加对话框 -->
    <el-dialog title="图书类别添加" :show-close="false" :close-on-click-
modal="false" :visible.sync="addDialogFlag">
      <el-form :model="book" :rules="rules" ref="book">
        <el-form-item label="类别名称" prop="catename">
          <el-input v-model="book.catename" placeholder="请输入类别名称"></el-
input>
        </el-form-item>
      </el-form>
      <div slot="footer" class="dialog-footer">
        <el-button @click="addDialogFlag= !addDialogFlag">取 消</el-button>
        <el-button type="primary" @click="save">保存</el-button>
      </div>
    </el-dialog>

    <!-- 修改对话框 -->
```

```
    <el-dialog title="图书类别修改" :show-close="false" :close-on-click-
modal="false" :visible.sync="editDialogFlag">

      <div slot="footer" class="dialog-footer">
        <el-button @click="editDialogFlag= !editDialogFlag">取 消</el-button>
        <el-button type="primary" @click="update">保存</el-button>
      </div>
    </el-dialog>

  </el-card>



</template>

<script>
  export default {
    data() {
      return {
        rules: {
          catename: [{
            required: true,
            message: '请输入类别名称 ',
            trigger: 'blur'
          }]
        }, //定义表单验证规则
        editDialogFlag: false, //控制修改对话框显示
        book: { //封装数据
          bid: 0,
          bname: '',
          author: '',
          price: '',
          quantity: '',
          ddesc: '',
          cateid: 0
        },
        addDialogFlag: false, //控制对话框是否显示
        cateName: '', //条件查询
        tableData: [],
        pager: { //分页封装对象
          page: 1,
          size: 5,
          total: 0
        }
      }
    },
    //该属性是vue对象中的成员变量，作用编写用户自己定的方法
    methods: {
      openAddDialog() {
        this.addDialogFlag = !this.addDialogFlag; //将打开添加对话框标记改成true
      },
      openEditDialog(row) {
        this.category = row;
        console.info(this.category)
        this.editDialogFlag = !this.editDialogFlag;
```

```
    },
    update() {
      this.$refs['category'].validate(valid => {
        if (valid) {
          this.$http.post('http://127.0.0.1/category/update', this.category)
            .then(res => {
              if (res.data.code == 200) {
                this.$message({
                  message: res.data.message,
                  type: 'success'
                });
                this.listPage();
                this.editDialogFlag = !this.editDialogFlag;
              }
            }).catch(error => {
              this.$message.error('错了哦，网络异常 ');
            })
        }
      })

    },
    save() {
      // let parmas = new URLSearchParams();
      // parmas.append("catename", this.category.catename)
      this.$refs['category'].validate(valid => {
        if (valid) {
          this.$http.post('http://127.0.0.1/category/add', this.category)
            .then(res => {
              if (res.data.code == 200) {
                this.$message({
                  message: res.data.message,
                  type: 'success'
                });
                this.listPage();
                this.addDialogFlag = !this.addDialogFlag;
              }
            }).catch(error => {
              this.$message.error('错了哦，网络异常 ');
            })
        }

      })

    },
    remove(row) {
      this.$http.post('http://127.0.0.1/category/remove?cateid=' + row.cateid)
        .then(res => {
          if (res.data.code == 200) {
            this.$message({
              message: res.data.message,
              type: 'success'
            });
            this.listPage();
          }
```

```
            }).catch(error => {
              this.$message.error('错了哦，网络异常 ');
            })
        },

        handleCurrentChange(val) {
          this.pager.page = val;
          this.listPage();
        },
        handleSizeChange(val) {
          this.pager.size = val;
          this.listPage(); //重新调用查询方法

        },
        //用户定义的前端请求方法
        listPage() {
          this.$http.get('http://127.0.0.1/book/listPage', {
            params: {
              page: this.pager.page,
              size: this.pager.size,
              bname: this.book.bname,
              author: this.book.author
            }
          }).then(res => {

            this.tableData = res.data.data.data;
            this.pager.total = res.data.data.total
          }).catch(error => {
            this.$message.error('错了哦，网络异常 ');
          })
        }

      },
      //vue声明周期函数
      mounted() {
        this.listPage();
      }
    }
</script>

<style>
    .el-breadcrumb {
      padding-bottom: 20px;
    }

    .el-pagination {
      padding-top: 10px;
    }
</style>
```

6.2.图书类别模块UI设计与实现




```
<template>
  <!-- 卡片组件 -->
  <el-card>
    <!-- 面包屑组件 -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/welcome' }">首页</el-breadcrumb-item>
      <el-breadcrumb-item>馆藏类别管理</el-breadcrumb-item>
      <el-breadcrumb-item>类别信息管理</el-breadcrumb-item>
    </el-breadcrumb>
    <el-row :gutter="10">
      <el-col :span="8">
        <el-input clearable @change="listPage()" v-model="cateName"
placeholder="请输入图书类别名称查询"></el-input>
      </el-col>
      <el-col :span="12">
        <el-button icon="el-icon-plus" @click="openAddDialog" type="success">添加
</el-button>
      </el-col>
    </el-row>
    <!-- 表格组件 -->
    <el-table :data="tableData" style="width: 100%">
      <el-table-column prop="catename" label="类别名称" width="180">
      </el-table-column>
      <el-table-column prop="createtime" label="创建时间">
      </el-table-column>
      <el-table-column prop="modifytime" label="修改时间">
      </el-table-column>
      <el-table-column lable="操作">
        <!-- 作用域插槽 -->
        <template slot-scope="scope">

          <el-tooltip content="点击修改" placement="bottom">
            <el-button type="warning" @click="openEditDialog(scope.row)"
icon="el-icon-edit" circle></el-button>
          </el-tooltip>
          <el-tooltip content="点击删除" placement="bottom">
```

```
            <el-button type="danger" @click="remove(scope.row)" icon="el-icon-delete" circle></el-button>
          </el-tooltip>
        </template>
      </el-table-column>
    </el-table>
    <!-- 分页组件定义 -->
    <el-pagination @size-change="handleSizeChange" @current-change="handleCurrentChange" :current-page="pager.page"
      :page-sizes="[5, 10, 15, 20]" :page-size="pager.size" layout="total, sizes, prev, pager, next, jumper"
      :total="pager.total">
    </el-pagination>

    <!-- 添加对话框 -->
    <el-dialog title="图书类别添加" :show-close="false" :close-on-click-modal="false" :visible.sync="addDialogFlag">
      <el-form :model="category" :rules="rules" ref="category">
        <el-form-item label="类别名称" prop="catename">
          <el-input v-model="category.catename" placeholder="请输入类别名称"></el-input>
        </el-form-item>
      </el-form>
      <div slot="footer" class="dialog-footer">
        <el-button @click="addDialogFlag= !addDialogFlag">取 消</el-button>
        <el-button type="primary" @click="save">保存</el-button>
      </div>
    </el-dialog>

    <!-- 修改对话框 -->
    <el-dialog title="图书类别修改" :show-close="false" :close-on-click-modal="false" :visible.sync="editDialogFlag">
      <el-form :model="category" :rules="rules" ref="category">
        <el-form-item label="类别名称" prop="catename">
          <el-input v-model="category.catename" placeholder="请输入类别名称"></el-input>
        </el-form-item>
      </el-form>
      <div slot="footer" class="dialog-footer">
        <el-button @click="editDialogFlag= !editDialogFlag">取 消</el-button>
        <el-button type="primary" @click="update">保存</el-button>
      </div>
    </el-dialog>

  </el-card>



</template>

<script>
  export default {
    data() {
      return {
        rules: {
```

```
        catename: [{
          required: true,
          message: '请输入类别名称',
          trigger: 'blur'
        }]
      }, //定义表单验证规则
      editDialogFlag: false, //控制修改对话框显示
      category: { //封装数据
        cateid: 0,
        catename: ''
      },
      addDialogFlag: false, //控制对话框是否显示
      cateName: '', //条件查询
      tableData: [],
      pager: { //分页封装对象
        page: 1,
        size: 5,
        total: 0
      }
    }
  },
  //该属性是vue对象中的成员变量，作用编写用户自己定的方法
  methods: {
    openAddDialog() {
      this.addDialogFlag = !this.addDialogFlag; //将打开添加对话框标记改成true
    },
    openEditDialog(row) {
      this.category = row;
      console.info(this.category)
      this.editDialogFlag = !this.editDialogFlag;
    },
    update() {
      this.$refs['category'].validate(valid => {
        if (valid) {
          this.$http.post('http://127.0.0.1/category/update', this.category)
            .then(res => {
              if (res.data.code == 200) {
                this.$message({
                  message: res.data.message,
                  type: 'success'
                });
                this.listPage();
                this.editDialogFlag = !this.editDialogFlag;
              }
            }).catch(error => {
              this.$message.error('错了哦，网络异常');
            })
        }
      })

    },
    save() {
      // let parmas = new URLSearchParams();
      // parmas.append("catename", this.category.catename)
      this.$refs['category'].validate(valid => {
```

```
        if (valid) {
          this.$http.post('http://127.0.0.1/category/add', this.category)
            .then(res => {
              if (res.data.code == 200) {
                this.$message({
                  message: res.data.message,
                  type: 'success'
                });
                this.listPage();
                this.addDialogFlag = !this.addDialogFlag;
              }
            }).catch(error => {
              this.$message.error('错了哦，网络异常 ');
            })
        }

      })

    },
    remove(row) {
      this.$http.post('http://127.0.0.1/category/remove?cateid=' + row.cateid)
        .then(res => {
          if (res.data.code == 200) {
            this.$message({
              message: res.data.message,
              type: 'success'
            });
            this.listPage();
          }

        }).catch(error => {
          this.$message.error('错了哦，网络异常 ');
        })
    },

    handleCurrentChange(val) {
      this.pager.page = val;
      this.listPage();
    },
    handleSizeChange(val) {
      this.pager.size = val;
      this.listPage(); //重新调用查询方法

    },
    //用户定义的前端请求方法
    listPage() {
      this.$http.get('http://127.0.0.1/category/listPage', {
        params: {
          page: this.pager.page,
          size: this.pager.size,
          cateName: this.cateName
        }
      }).then(res => {

        this.tableData = res.data.data.data;
```

```
            this.pager.total = res.data.data.count
          }).catch(error => {
            this.$message.error('错了哦，网络异常 ');
          })
        }

    },
    //vue声明周期函数
    mounted() {
      this.listPage();
    }
  }
</script>

<style>
  .el-breadcrumb {
    padding-bottom: 20px;
  }

  .el-pagination {
    padding-top: 10px;
  }
</style>
```

6.3.借阅模块UI设计与实现

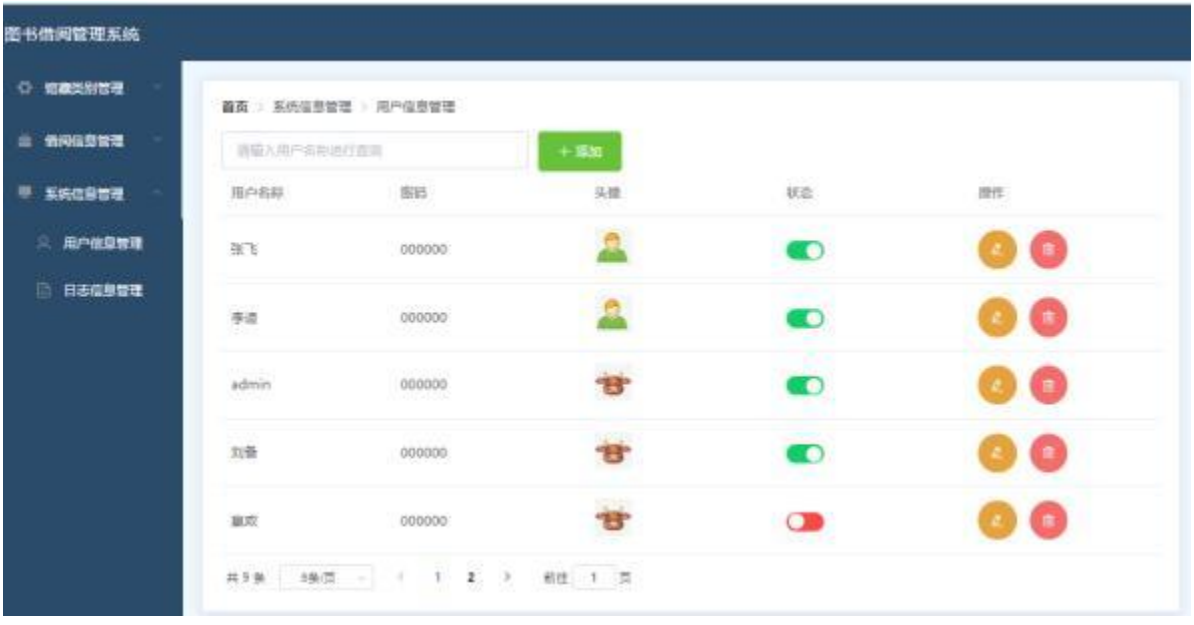

```
<template>
  <!-- 卡片组件 -->
  <el-card>
    <!-- 面包屑组件 -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/welcome' }">首页</el-breadcrumb-item>
      <el-breadcrumb-item>借阅信息管理</el-breadcrumb-item>
      <el-breadcrumb-item>借阅信息管理</el-breadcrumb-item>
    </el-breadcrumb>
    <el-row :gutter="10">
      <el-col :span="8">
        <el-input clearable @change="listPage()" v-model="borrow.book.bname"
placeholder="请输入借阅图书称查询"></el-input>
```

```
      </el-col>
      <el-col :span="12">
        <el-button icon="el-icon-plus" @click="openAddDialog" type="success">借阅
</el-button>
      </el-col>
    </el-row>
    <!-- 表格组件 -->
    <el-table :data="tableData">
      <el-table-column prop="book" label="图书名称">
        <template slot-scope="scope">
          {{scope.row.book.bname}}
        </template>
      </el-table-column>
      <el-table-column prop="users" label="借阅人">
        <template slot-scope="scope">
          {{scope.row.users.uname}}
        </template>
      </el-table-column>
      <el-table-column prop="otime" label="借出时间">
      </el-table-column>
      <el-table-column prop="itime" label="归还时间">
      </el-table-column>
      <el-table-column prop="quantity" label="数量">
      </el-table-column>
      <el-table-column prop="quantity" label="状态">
        <template slot-scope="scope">
          <el-tag type="success" v-show="scope.row.stat==1">已归还</el-tag>
          <el-tag type="danger" v-show="scope.row.stat==0">已借出</el-tag>
        </template>
      </el-table-column>
      <el-table-column label="操作">
        <!-- 作用域插槽 -->
        <template slot-scope="scope">
          <el-button size="mini" type="danger" v-show="scope.row.stat==0">归还
</el-button>
        </template>
      </el-table-column>
    </el-table>
    <!-- 分页组件定义 -->
    <el-pagination @size-change="handleSizeChange" @current-
change="handleCurrentChange" :current-page="pager.page"
      :page-sizes="[5, 10, 15, 20]" :page-size="pager.size" layout="total,
sizes, prev, pager, next, jumper"
      :total="pager.total">
    </el-pagination>
  </el-card>



</template>

<script>
  export default {
    data() {
      return {
```

```
      rules: {
        catename: [{
          required: true,
          message: '请输入类别名称',
          trigger: 'blur'
        }]
      }, //定义表单验证规则
      editDialogFlag: false, //控制修改对话框显示
      category: { //封装数据
        cateid: 0,
        catename: ''
      },
      borrow: {
        boid: 0,
        otime: '',
        itime: '',
        quantity: 0,
        book: {
          bid: 0,
          bname: ''
        },
        users: {
          uid: 0 ,
          uname: ''
        }
      },
      addDialogFlag: false, //控制对话框是否显示
      cateName: '', //条件查询
      tableData: [],
      pager: { //分页封装对象
        page: 1,
        size: 5,
        total: 0
      }
    }
  },
  //该属性是vue对象中的成员变量，作用编写用户自己定的方法
  methods: {
    openAddDialog() {
      this.addDialogFlag = !this.addDialogFlag; //将打开添加对话框标记改成true
    },
    openEditDialog(row) {
      this.category = row;
      console.info(this.category)
      this.editDialogFlag = !this.editDialogFlag;
    },

    handleCurrentChange(val) {
      this.pager.page = val;
      this.listPage();
    },
    handleSizeChange(val) {
      this.pager.size = val;
      this.listPage(); //重新调用查询方法
```

```
      },
      //用户定义的前端请求方法
      listPage() {
        this.$http.get('http://127.0.0.1/borrow/listPage', {
          params: {
            page: this.pager.page,
            size: this.pager.size,
            bname: this.borrow.book.bname
          }
        }).then(res => {

          this.tableData = res.data.data.data;
          this.pager.total = res.data.data.count
        }).catch(error => {
          this.$message.error('错了哦，网络异常 ');
        })
      }

    },
    //vue声明周期函数
    mounted() {
      this.listPage();
    }
  }
</script>

<style>
  .el-breadcrumb {
    padding-bottom: 20px;
  }

  .el-pagination {
    padding-top: 10px;
  }
</style>
```

6.4.用户模块UI设计与实现



```
<template>
  <!-- 卡片组件 -->
  <el-card>
    <!-- 面包屑组件 -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/welcome' }">首页</el-breadcrumb-item>
      <el-breadcrumb-item>系统信息管理</el-breadcrumb-item>
      <el-breadcrumb-item>用户信息管理</el-breadcrumb-item>
    </el-breadcrumb>
    <el-row :gutter="10">
      <el-col :span="8">
        <el-input clearable @change="listPage()" v-model="uname"
         placeholder="请输入用户名称进行查询"></el-input>
      </el-col>
      <el-col :span="12">
        <el-button icon="el-icon-plus" @click="openAddDialog" type="success">添加
</el-button>
      </el-col>
    </el-row>
    <!-- 表格组件 -->
    <el-table :data="tableData" style="width: 100%">
      <el-table-column prop="uname" label="用户名称" width="180">
      </el-table-column>
      <el-table-column prop="password" label="密码">
      </el-table-column>
      <el-table-column prop="images" label="头像">
        <template slot-scope="scope">
          <img  :src="scope.row.images" style="width: 40px;height: 40x;"/>
        </template>
      </el-table-column>
      <el-table-column prop="stats" label="状态">
        <template slot-scope="scope">
          <el-switch
            v-model="scope.row.stats"
            :active-value="0"
            :inactive-value="1"
            active-color="#13ce66"
            inactive-color="#ff4949"  @change="changStats(scope.row)">
          </el-switch>
        </template>
      </el-table-column>
      <el-table-column label="操作">
        <!-- 作用域插槽 -->
        <template slot-scope="scope">

          <el-tooltip content="点击修改" placement="bottom">
            <el-button type="warning" @click="openEditDialog(scope.row)"
icon="el-icon-edit" circle></el-button>
          </el-tooltip>
          <el-tooltip content="点击删除" placement="bottom">
            <el-button type="danger" @click="remove(scope.row)" icon="el-icon-
delete" circle></el-button>
          </el-tooltip>
        </template>
      </el-table-column>
```

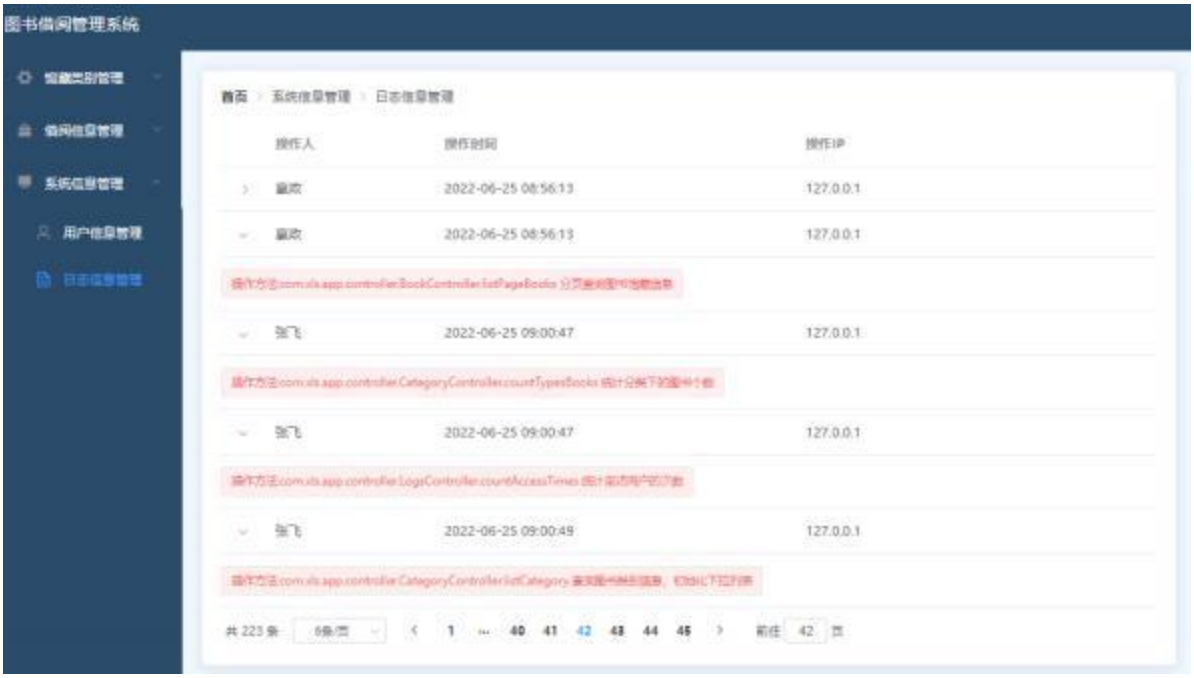
```
    </el-table>
    <!-- 分页组件定义 -->
    <el-pagination @size-change="handleSizeChange" @current-
change="handleCurrentChange" :current-page="pager.page"
      :page-sizes="[5, 10, 15, 20]" :page-size="pager.size" layout="total,
sizes, prev, pager, next, jumper"
      :total="pager.total">
    </el-pagination>

    <!-- 添加对话框 -->
    <el-dialog title="用户添加" :show-close="false" :close-on-click-modal="false"
:visible.sync="addDialogFlag">
      <el-form :model="users" :rules="rules" ref="users">
        <el-form-item label="用户名称" prop="catename">
          <el-input v-model="users.uname" placeholder="请输入用户名称"></el-input>
        </el-form-item>
        <el-form-item label="用户密码" prop="catename">
          <el-input v-model="users.password" placeholder="请输入用户密码"></el-
input>
        </el-form-item>
        <el-form-item label="用户头像" prop="catename">
          <el-input v-model="users.images" placeholder="请输入用户头像"></el-input>
        </el-form-item>
      </el-form>
      <div slot="footer" class="dialog-footer">
        <el-button @click="addDialogFlag= !addDialogFlag">取 消</el-button>
        <el-button type="primary" @click="save">保存</el-button>
      </div>
    </el-dialog>

    <!-- 修改对话框 -->
    <el-dialog title="用户修改" :show-close="false" :close-on-click-modal="false"
:visible.sync="editDialogFlag">
      <el-form :model="users" :rules="rules" ref="users">
       <el-form-item label="用户名称" prop="catename">
         <el-input v-model="users.uname" placeholder="请输入用户名称"></el-input>
       </el-form-item>
       <el-form-item label="用户密码" prop="catename">
         <el-input v-model="users.password" placeholder="请输入用户密码"></el-
input>
       </el-form-item>
       <el-form-item label="用户头像" prop="catename">
         <el-input v-model="users.images" placeholder="请输入用户头像"></el-input>
       </el-form-item>
      </el-form>
      <div slot="footer" class="dialog-footer">
        <el-button @click="editDialogFlag= !editDialogFlag">取 消</el-button>
        <el-button type="primary" @click="update">保存</el-button>
      </div>
    </el-dialog>

  </el-card>
```

```
</template>

<script>
  export default {
    data() {
      return {
        uname:'',
        rules: {
          uname: [{
            required: true,
            message: '请输入用户名称',
            trigger: 'blur'
          }],
          password: [{
            required: true,
            message: '请输入密码',
            trigger: 'blur'
          }],
          images: [{
            required: true,
            message: '请输入头像连接地址',
            trigger: 'blur'
          }]
        }, //定义表单验证规则
        editDialogFlag: false, //控制修改对话框显示
        users: { //封装数据
          uid: 0,
          uname: '',
          images:'',
          password:''
        },
        addDialogFlag: false, //控制对话框是否显示
        tableData: [],
        pager: { //分页封装对象
          page: 1,
          size: 5,
          total: 0
        }
      }
    },
    //该属性是vue对象中的成员变量，作用编写用户自己定的方法
    methods: {
      changStats(row){ //改变用户状态
        this.users=row;

 this.$http.post('http://127.0.0.1/users/changStats',this.users).then(res=>{
          if (res.data.code == 200) {
            this.$message({
              message: res.data.message,
              type: 'success'
            });
            this.listPage();
          }

        })
```

```
    },
    openAddDialog() {
      this.addDialogFlag = !this.addDialogFlag; //将打开添加对话框标记改成true
    },
    openEditDialog(row) {
      this.users = row;
      this.editDialogFlag = !this.editDialogFlag;
    },
    update() {
      this.$refs['users'].validate(valid => {
        if (valid) {
          this.$http.post('http://127.0.0.1/users/update', this.users)
            .then(res => {
              if (res.data.code == 200) {
                this.$message({
                  message: res.data.message,
                  type: 'success'
                });
                this.listPage();
                this.editDialogFlag = !this.editDialogFlag;
              }
            }).catch(error => {
              this.$message.error('错了哦，网络异常 ');
            })
        }
      })

    },
    save() {
      this.$refs['users'].validate(valid => {
        if (valid) {
          this.$http.post('http://127.0.0.1/users/save', this.users)
            .then(res => {
              if (res.data.code == 200) {
                this.$message({
                  message: res.data.message,
                  type: 'success'
                });
                this.listPage();
                this.addDialogFlag = !this.addDialogFlag;
              }
            }).catch(error => {
              this.$message.error('错了哦，网络异常 ');
            })
        }

      })

    },
    remove(row) {
      this.$http.post('http://127.0.0.1/users/remove?uid=' + row.uid)
        .then(res => {
          if (res.data.code == 200) {
            this.$message({
              message: res.data.message,
```

```
                type: 'success'
              });
              this.listPage();
            }

          }).catch(error => {
            this.$message.error('错了哦，网络异常 ');
          })
      },

      handleCurrentChange(val) {
        this.pager.page = val;
        this.listPage();
      },
      handleSizeChange(val) {
        this.pager.size = val;
        this.listPage(); //重新调用查询方法

      },
      //用户定义的前端请求方法
      listPage() {
        this.$http.get('http://127.0.0.1/users/listPage', {
          params: {
            page: this.pager.page,
            size: this.pager.size,
            uname:this.uname
          }
        }).then(res => {

          this.tableData = res.data.data.data;
          this.pager.total = res.data.data.total
        }).catch(error => {
          this.$message.error('错了哦，网络异常 ');
        })
      }

    },
    //vue声明周期函数
    mounted() {
      this.listPage();
    }
  }
</script>

<style>
  .el-breadcrumb {
    padding-bottom: 20px;
  }

  .el-pagination {
    padding-top: 10px;
  }
</style>
```

6.5.日志模块UI设计与实现




```
<template>
  <!-- 卡片组件 -->
  <el-card>
    <!-- 面包屑组件 -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/welcome' }">首页</el-breadcrumb-item>
      <el-breadcrumb-item>系统信息管理</el-breadcrumb-item>
      <el-breadcrumb-item>日志信息管理</el-breadcrumb-item>
    </el-breadcrumb>
    <!-- 表格组件 -->
    <el-table :data="tableData" >
      <el-table-column type="expand">
        <template slot-scope="scope">
         <el-tag type="danger">操作方法:{{scope.row.methods}} {{scope.row.ddesc}}
</el-tag>
        </template>
      </el-table-column>
      <el-table-column prop="opername" label="操作人" width="180">
      </el-table-column>
      <el-table-column prop="opertime" label="操作时间">
      </el-table-column>
      <el-table-column prop="ip" label="操作IP">
      </el-table-column>

    </el-table>
    <!-- 分页组件定义 -->
    <el-pagination @size-change="handleSizeChange" @current-
change="handleCurrentChange" :current-page="pager.page"
      :page-sizes="[5, 10, 15, 20]" :page-size="pager.size" layout="total,
sizes, prev, pager, next, jumper"
      :total="pager.total">
    </el-pagination>
  </el-card>

</template>
```

```
<script>
  export default {
    data() {
      return {
        rules: {
          catename: [{
            required: true,
            message: '请输入类别名称',
            trigger: 'blur'
          }]
        }, //定义表单验证规则
        editDialogFlag: false, //控制修改对话框显示
        category: { //封装数据
          cateid: 0,
          catename: ''
        },
        addDialogFlag: false, //控制对话框是否显示
        cateName: '', //条件查询
        tableData: [],
        pager: { //分页封装对象
          page: 1,
          size: 5,
          total: 0
        }
      }
    },
    //该属性是vue对象中的成员变量，作用编写用户自己定的方法
    methods: {

      handleCurrentChange(val) {
        this.pager.page = val;
        this.listPage();
      },
      handleSizeChange(val) {
        this.pager.size = val;
        this.listPage(); //重新调用查询方法

      },
      //用户定义的前端请求方法
      listPage() {
        this.$http.get('http://127.0.0.1/logs/listPage', {
          params: {
            page: this.pager.page,
            size: this.pager.size,
          }
        }).then(res => {

          this.tableData = res.data.data.data;
          this.pager.total = res.data.data.count
        }).catch(error => {
          this.$message.error('错了哦，网络异常');
        })
      }
```

```
    },
    //vue声明周期函数
    mounted() {
      this.listPage();
    }
  }
</script>

<style>
  .el-breadcrumb {
    padding-bottom: 20px;
  }

  .el-pagination {
    padding-top: 10px;
  }
</style>
```