

程序设计报告

（2018/2019 学年 第 二 学期）

题 目：**个人账本管理系统**

|  |  |
| --- | --- |
| **专 业** | **计算机科学与技术** |
| **组长 学号姓名** | **B18030406 张颖** |
| **组员 学号姓名** | **B18030408 胡欣然** |
|  | **B18030405 杨路玥** |
| **指 导 教 师** | **吴家皋** |
| **指 导 单 位** | **计算机学院、软件学院、网络空间安全学院** |
| **日 期** | **2019.3.29** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **成员分工** | | | | | | | | | |
| 组长（张颖） | | 开发：记账列表明细界面（增删查排序）  对后端数据库的调用  前后端交互  架构选用  文档：编写概念设计  测试：删改查功能测试 | | | | | | | |
| 组员（胡欣然） | | 开发：利用数据可视化来统计用户整体数据的图表分析  首页设计  显示功能  文档：编写测试数据及其结果分析  使用visio画流程图  编写需求分析文档  测试：统计图表测试和需求分析 | | | | | | | |
| 组员（杨路玥） | | 开发：增加账目数据功能  首页设计  文档：编写调试过程中的问题  PPT制作  测试：增加功能测试 | | | | | | | |
| **教师评价** | | | | | | | | | |
| 评价准则 | | | | 优秀 | 良好 | | 中等 | 合格 | 不合格 |
| 该课题组掌握程序设计开发的相关工程基础知识，并能够针对求解的工程问题，进行合理的分析与综合。 | | | |  |  | |  |  |  |
| 该课题组能够结合计算机软硬件资源，合理选用算法、数据结构、数据存储方式等技术手段，对求解的工程问题进行系统设计、建模及预测。 | | | |  |  | |  |  |  |
| 该课题组能够选择合适的程序设计语言与编程开发平台，对求解的工程问题进行系统模拟、仿真与预测。 | | | |  | √ | |  |  |  |
| 该课题组熟练掌握调试方法与工具，对程序开发过程中出现的问题进行分析、跟踪与调试，并在开发平台上对最终作品进行充分测试。 | | | |  |  | |  |  |  |
| 该课题组具备初步的开发能力，完成了课题要求的各项开发任务，功能实现全面。 | | | |  |  | |  |  |  |
| 该课题组具备初步的界面设计意识。用户使用界面布局合理，交互友好，操作简便。 | | | |  |  | |  |  |  |
| 该课题组正确、完整的回答了指导教师关于课题作品的提问，反映出其对课题内容，以及相关的工程基础知识具有较好的理解和掌握。 | | | |  |  | |  |  |  |
| 该课题组具备一定的表达能力与文字处理能力，能够熟练使用文字处理软件完成课题报告一份。报告完整记录了课题组的工作及总结，内容详实，格式规范。 | | | |  |  | |  |  |  |
| 该课题组具有较好的团队协作精神，能够努力沟通解决遇到的各种开发、工程问题。 | | | |  |  | |  |  |  |
| 该课题组具备一定自学能力与探索创新意识，能够充分利用各种网络与图书馆资源自学新知识与新技能。 | | | |  |  | |  |  |  |
| 该课题组在上机过程中遵守机房规章制度，出勤与平时表现规范。 | | | |  |  | |  |  |  |
| 最终作品和报告反映出该课题组具备较好的解决工程问题的能力。 | | | |  |  | |  |  |  |
| **本次程序设计能力达成评价** | | | | | | | | | |
| 学号姓名 | 张颖B18030406 | | 胡欣然B18030408 | | | 杨路玥B18030405 | | | |
| 贡献度及个人表现 |  | |  | | |  | | | |
| 最终成绩 |  | |  | | |  | | | |
| **备注：**  贡献度及个人表现：好、较好、中等、较差、差  最终成绩：优秀、良好、中等、及格、不及格 | | | | | | | | | |
| **指导教师： 年 月 日** | | | | | | | | | |

**个人账本管理系统**

**一、课题内容和要求**

个人账本管理系统要求实现一个简单、实用的个人账本管理程序，主要功能包括账目数据的录入、查找、修改、删除、排序、显示等。

**二、需求分析**

客户管理系统的功能框架图如图1所示。



图1 功能框架图

（1）提供可操作的主菜单：在PC端浏览器打开网站页面，输出记账本功能菜单，用于显示以从数据库中加载的总账目信息和若干个可选的功能选项。根据用户输入的选项来运行不同的功能，运行不同的函数。

（2）查找功能函数：分为按消费类型查找和按用户消费账户类型以及搜索备注中的关键字进行账目查询。

（3）插入功能函数：进行账目数据分类的存储，包括消费类型、消费账户、时间、备注以及金额等数据。

（4）删除功能函数：进行账目数据的删除。

（5）修改功能函数：分别进行账目数据中消费类型、账户、时间等的数据的修改。

（6）排序功能函数：将账目数据用时间等类型进行排序。

（7）统计功能函数：将数据聚合利用饼图，折线图和条形图显示。

**三、概要设计**

1 主要存储结构

Data{

Id； //每条账目数据的id Number

time； //每条账目数据的记录时间 String

type； //每条账目数据的消费类型 String

pay\_type；//每条账目数据的账户类型 String

num； //每条账目数据的金额 Number

note； //每条账目数据的备注 String

pay； //每条账目数据支出/收入 String

}

2 主要函数流程

（1）数据初始化函数：用于从数据库中调入测试demo，此处以ajax来调用接口方式读入数据，数据初始化函数如图2所示。



图2 数据初始化函数

（2）增加函数：利用用户输入的新数据增加账目记录，如图3所示。



图3 增加功能函数

（3）删除和修改函数：利用用户点击的对象的id或文本值判断用户的需求为删除/更改，修改和删除功能函数如图4所示。



图4 删除和修改功能函数

（4）查询函数：利用用户点击对象的文本值判断用户需求为按消费类型筛选/按账户类型筛选/按备注关键字筛选，查找功能函数如图5所示。



图5 查找功能函数

（5）排序功能函数：判断用户点击时账目顺序是否按时间排好，排序功能函数如图6所示。



图6 排序功能函数

（6）统计功能函数：利用用户的全部数据访问后端接口，进行饼图、折线表、条形图的图表分析，统计功能函数如图7所示。



图7 统计函数

（3）技术架构

1、使用html/css/JavaScript开发网页版个人记账管理系统以达成构建图形化界面目的

2、使用Amaze UI前端框架进行组件化设计

3、构建单页面网页以达到更加友好的用户体验

4、运用数据可视化进行数据的图表分析

5、使用JavaScript写后端与前端实现交互

6、运用mongoed来进行数据库管理

**四、源程序代码**

（1）HTML文件

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<meta http-equiv="X-UA-Compatible" content="ie=edge" />

<title>个人记账本</title>

<link rel="stylesheet" href="./css/reset.css" />

<link rel="stylesheet" href="./css/index.css" />

<link rel="stylesheet" href="./AmazeUI-2.7.2/assets/css/amazeui.css" />

<link rel="stylesheet" href="./css/font\_1106087\_gqgq3hkolf4/iconfont.css" />

</head>

<body>

<div class="container">

<div class="homePage">

<div class="navigation flex">

<div class="nav-left">

<span class="iconfont youzi"></span>

<span id="name">柚子记账</span>

</div>

<div class="nav-right">

<ul class="am-nav am-nav-pills">

<li class="am-active"><a href="">首页</a></li>

<li><a href="">帮助</a></li>

<li><a href="">登陆</a></li>

<li><a href="">注册</a></li>

<li data-am-smooth-scroll="{position: 8000}"><a href="">关于我们</a></li>

</ul>

</div>

</div>

<div class="logoBox">

<div class="am-slider am-slider-default" data-am-flexslider="{playAfterPaused: 8000}">

<ul class="am-slides">

<li> <img src="./images/banner\_01.jpg" alt="" /></li>

<li><img src="./images/banner\_02.jpg" alt="" /></li>

<li><img src="./images/banner\_03.jpg" alt="" /></li>

</ul>

</div>

</div>

<div class="navFuction flex">

<button data-am-smooth-scroll="{position: 183}" class="am-btn am-btn-secondary"> <span class="iconfont "></span> <p class="fuctionTitle">记账</p> <p class="fuctionView">最简便的记账方式</p> </button>

<button data-am-smooth-scroll="{position:1820}" class="am-btn am-btn-secondary"> <span class="iconfont "></span> <p class="fuctionTitle">图表</p> <p class="fuctionView">百分比展现消费</p> </button>

<button data-am-smooth-scroll="{position: 950}" class="am-btn am-btn-secondary"> <span class="iconfont "></span> <p class="fuctionTitle">明细</p> <p class="fuctionView">具体详细的消费清单</p> </button>

<button data-am-smooth-scroll="{position: 189}" class="am-btn am-btn-secondary"> <span class="iconfont "></span> <p class="fuctionTitle">我的</p> <p class="fuctionView">个人信息界面</p> </button>

</div>

</div>

<div class="specificFunction">

<div class="account">

<div class="accountBook">

<ol class="am-breadcrumb">

<li class="am-active"><a href="#">生活账本</a></li>

<li> <a href="#"></a> </li>

</ol>

</div>

<div class="accountList flex">

<div class="detail">

<div class="addList flex">

<div class="sort">

<select data-am-selected=""> <option value="a" selected="">餐饮费</option> <option value="b">服饰美容</option> <option value="c"> 生活日用</option> <option value="d">住房缴费</option> <option value="e">交通出行</option> <option value="f">通讯物流</option> <option value="g">文教娱乐</option> <option value="h">健康</option> <option value="i">其他消费</option> </select>

</div>

<div class="money">

<select data-am-selected=""> <option value="a" selected="">现金</option> <option value="b">银行卡</option> <option value="c">支付宝</option> <option value="d">信用卡</option> <option value="e">余额宝</option> </select>

</div>

<div class="figure">

<div class="text">

<p><input type="number" class="am-form-field am-radius" placeholder="金额" /></p>

</div>

</div>

<div class="time">

<div class="am-input-group am-datepicker-date" data-am-datepicker="{format: 'yyyy-mm-dd'}">

<input type="text" class="am-form-field" placeholder="日历组件" readonly="" />

<span class="am-input-group-btn am-datepicker-add-on"> <button class="am-btn am-btn-default" type="button"><span class="am-icon-calendar"></span> </button> </span>

</div>

</div>

<div class="note flex">

<p><input type="text" class="am-form-field am-radius" placeholder="点击添加备注" /></p>

<button type="button" class="am-btn am-btn-default am-radius add">保存</button>

</div>

</div>

<div class="specialList">

<div class="specialListNav flex">

<span class="iconfont"></span>

<button type="button" class="am-btn am-btn-primary" id="sort">按时间排序</button>

<select data-am-selected=""> <option value="a" selected="">餐饮费</option> <option value="b">服饰美容</option> <option value="c"> 生活日用</option> <option value="d">住房缴费</option> <option value="e">交通出行</option> <option value="f">通讯物流</option> <option value="g">文教娱乐</option> <option value="h">健康</option> <option value="i">其他消费</option> </select>

<select data-am-selected=""> <option value="a" selected="">现金</option> <option value="b">银行卡</option> <option value="c">支付宝</option> <option value="d">信用卡</option> <option value="e">余额宝</option> </select>

<div class="typeNote">

<p><input type="text" class="am-form-field am-radius" placeholder="搜索备注关键字" id="searchNote" /></p>

<button type="button" class="am-btn am-btn-primary" id="search">搜索</button>

</div>

</div>

<div class="List">

<div>

<ul class="am-nav am-nav-pills " id="listnav">

<li><a href="">序号</a></li>

<li><a href="">时间</a></li>

<li><a href="">消费类</a></li>

<li><a href="">金额</a></li>

<li><a href="">账户</a></li>

<li><a href="">备注</a></li>

<li><a href="">删除</a></li>

</ul>

</div>

<div class="list1">

<div class="am-panel-group" id="accordion"></div>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="chart" id="chart" style="

height: 80vh;

position: absolute;

top: 242vh;

width: 60vw;

margin: 0 20vw;"></div>

<div class="personage" id="zxb" style="

height: 80vh;

position: absolute;

top: 331vh;

width: 60vw;

margin: 0 20vw;"></div>

<div id="txt" style="

height: 43vh;

position: absolute;

top: 410vh;

width: 60vw;

margin: 0 20vw;"></div>

</div>

<footer class="flex">

<div>

<img src="./images/二维码.png" alt="" />

</div>

<div>

<ul>

<li><i class="am-icon-qq am-icon-fw"></i>389746410</li>

<li><i class="am-icon-github am-icon-fw"></i> GitHub</li>

<li>copyright@HKFPouu</li>

</ul>

</div>

<div>

<img src="./images/二维码.png" alt="" />

</div>

</footer>

<span class="iconfont" data-am-smooth-scroll="" id="top"></span>

</div>

<script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.js"></script>

<script src="https://gw.alipayobjects.com/os/lib/antv/g2/3.4.10/dist/g2.min.js"></script>

<script src="https://gw.alipayobjects.com/os/antv/pkg/\_antv.data-set-0.10.1/dist/data-set.min.js"></script>

<script src="./js/index.js"></script>

<script src="./js/antv.js"></script>

<script src="./AmazeUI-2.7.2/assets/js/amazeui.js"></script>

</body>

</html>

（2）index.js

let data;

//数据初始化

$.ajax({

url: "http:/127.0.0.1:8022/find",

//127.0.0.1：8022/sort

type: "get",

data: {},

async: false,

success: function(res) {

data = res.data

},

error: function(err) {

}

})

window.onload = function() {

//导入初始化数值

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

$('.am-panel-group').append(` <div class = &quot;am-panel am-panel-default&quot; &gt;

<div class="am-panel-hd">

<h4 class="am-panel-title" data="" -="" am="" collapse="{parent: '#accordion', target: '#do-not-say-${data[i].id}'}">

<ul class="list flex">

<li> $ { data[i].id } &lt; /li&gt; </li>

<li>${data[i].time}</li>

<li class="value type"> $ { data[i].type } &lt; /li&gt; </li>

<li class="value num">${data[i].num}</li>

<li class="value pay\_type"> $ { data[i].pay\_type } &lt; /li&gt; </li>

<li class="value note">${data[i].note}</li>

<li style="display:inline-block"> <button type="button" class="am-close" id="${data[i].id}"> &times; &lt; /button&gt;</button></li>

</ul> </h4>

</div>

<div id="do-not-say-${data[i].id}" class="am-panel-collapse am-collapse">

<div class="am-panel-bd flex">

<div class="sortBtn">

<select data-am-selected=""> <option value="a">餐饮费</option> <option value="b"> 服饰美容 &lt; /option&gt; </option><option value="c"> 生活日用</option> <option value="d"> 住房缴费 &lt; /option&gt; </option><option value="e">交通出行</option> <option value="f"> 通讯物流 &lt; /option&gt; </option><option value="g">文教娱乐</option> <option value="h"> 健康 &lt; /option&gt; </option><option value="i">其他消费</option> </select>

</div>

<div class="moneyBtn">

<select data="" -="" am="" selected=""> <option value="a"> 现金 &lt; /option&gt; </option><option value="b">银行卡</option> <option value="c"> 支付宝 &lt; /option&gt; </option><option value="d">信用卡</option> <option value="e"> 余额宝 &lt; /option&gt; </option></select>

</div>

<div class="figureBtn">

<div class="text">

<p><input type="number" class="am-form-field am-radius" value="${data[i].num}" /> </p>

</div>

</div>

<div class="time">

<div class="am-input-group am-datepicker-date" data-am-datepicker="{format: 'yyyy-mm-dd'}">

<input type="text" class="am-form-field" value="${data[i].time}" readonly="" />

<span class="am-input-group-btn am-datepicker-add-on"> <button class="am-btn am-btn-default" type="button"><span class="am-icon-calendar"></span> </button> </span>

</div>

</div>

<div class="note flex">

<p> <input type="text" class="am-form-field am-radius" value="${data[i].note}" /></p>

</div>

<button type="button" class="am-btn am-btn-primary am-btn-block alter" id="${data[i].id}"> 修改 &lt; /button&gt; </button>

</div>

</div>

</body>

</html> `);

}

//增加

$(".add").click(function() {

let id = Number(data[data.length - 1].id) + 1,

type = $('.sort').find('option:selected').text(),

pay\_type = $('.money').find('option:selected').text(),

num = $('.figure').find('input').val(),

note = $('.note').find('input').val(),

time = $('.time').find('input').val();

data.push({

id: `$ {

id

}`,

time: $('.time').find('input').val(),

type: $('.sort').find('option:selected').text(),

pay\_type: $('.money').find('option:selected').text(),

num: $('.figure').find('input').val(),

note: $('.note').find('input').val()

}) console.log($('.time').find('input').val()) $('.am-panel-group').append(` <div class = &quot;am-panel am-panel-default&quot; &gt;

<div class="am-panel-hd">

<h4 class="am-panel-title" data="" -="" am="" collapse="{parent: '#accordion', target: '#do-not-say-${data[i].id}'}">

<ul class="list flex">

<li> $ { data[i].id } &lt; /li&gt; </li>

<li>${data[i].time}</li>

<li class="value type"> $ { data[i].type } &lt; /li&gt; </li>

<li class="value num">${data[i].num}</li>

<li class="value pay\_type"> $ { data[i].pay\_type } &lt; /li&gt; </li>

<li class="value note">${data[i].note}</li>

<li style="display:inline-block"> <button type="button" class="am-close" id="${data[i].id}"> &times; &lt; /button&gt;</button></li>

</ul> </h4>

</div>

<div id="do-not-say-${data[i].id}" class="am-panel-collapse am-collapse">

<div class="am-panel-bd flex">

<div class="sortBtn">

<select data-am-selected=""> <option value="a">餐饮费</option> <option value="b"> 服饰美容 &lt; /option&gt; </option><option value="c"> 生活日用</option> <option value="d"> 住房缴费 &lt; /option&gt; </option><option value="e">交通出行</option> <option value="f"> 通讯物流 &lt; /option&gt; </option><option value="g">文教娱乐</option> <option value="h"> 健康 &lt; /option&gt; </option><option value="i">其他消费</option> </select>

</div>

<div class="moneyBtn">

<select data="" -="" am="" selected=""> <option value="a"> 现金 &lt; /option&gt; </option><option value="b">银行卡</option> <option value="c"> 支付宝 &lt; /option&gt; </option><option value="d">信用卡</option> <option value="e"> 余额宝 &lt; /option&gt; </option></select>

</div>

<div class="figureBtn">

<div class="text">

<p><input type="number" class="am-form-field am-radius" value="${data[i].num}" /> </p>

</div>

</div>

<div class="time">

<div class="am-input-group am-datepicker-date" data-am-datepicker="{format: 'yyyy-mm-dd'}">

<input type="text" class="am-form-field" value="${data[i].time}" readonly="" />

<span class="am-input-group-btn am-datepicker-add-on"> <button class="am-btn am-btn-default" type="button"><span class="am-icon-calendar"></span> </button> </span>

</div>

</div>

<div class="note flex">

<p> <input type="text" class="am-form-field am-radius" value="${data[i].note}" /></p>

</div>

<button type="button" class="am-btn am-btn-primary am-btn-block alter" id="${data[i].id}"> 修改 &lt; /button&gt; </button>

</div>

</div>

</body>

</html> `);

$.ajax({

url: "http:/127.0.0.1:8022/add",

type: "get",

data: {

id: parseInt(id),

time: time,

type: type,

pay\_type: pay\_type,

num: num,

note: note

},

async: false,

success: function(res) {

data = res.data

},

error: function(err) {

}

})

});

//删除和修改

document.querySelector(".list1").onclick = function(ev) {

var ev = ev || window.event;

var target = ev.target || ev.srcElement;

if (target.className == "am-close") {

if (data.find(function(x) {

return x.id == target.id

})) {

data.splice(data.findIndex(function(x) {

return x.id == target.id

}), 1);

target.parentNode.parentNode.parentNode.parentNode.parentNode.parentNode.removeChild(target.parentNode.parentNode.parentNode.parentNode.parentNode);

$.ajax({

url: "http:/127.0.0.1:8022/delete",

type: "get",

data: {

id: parseInt(target.id),

},

async: false,

success: function(res) {},

error: function(err) {}

})

}

} else if (target.innerText == "修改") {

if (data.find(function(x) {

return x.id == target.id

})) {

data[data.findIndex(function(x) {

return x.id == target.id

})] = {

id: `$ {

target.id

}`,

type: target.parentNode.parentNode.querySelector('.sortBtn').querySelector('select')[target.parentNode.parentNode.querySelector('.sortBtn').querySelector('select').selectedIndex].text,

pay\_type: target.parentNode.parentNode.querySelector('.moneyBtn').querySelector('select')[target.parentNode.parentNode.querySelector('.sortBtn').querySelector('select').selectedIndex].text,

num: Number($('.figureBtn').find('input').val()),

note: target.parentNode.querySelector('.note').querySelector('input').value,

time: target.parentNode.querySelector('.time').querySelector('input').value

};

console.log(target.parentNode.ElementChild) target.parentNode.parentNode.parentNode.firstElementChild.firstElementChild.firstElementChild.innerHTML = ` < li > $ {

data[data.findIndex(function(x) {

return x.id == target.id

})].id

} < /li>

<li>${data[data.findIndex(function(x) { return x.id == target.id })].time}</li > <li class = "value type" > $ {

data[data.findIndex(function(x) {

return x.id == target.id

})].type

} < /li>

<li class="value num">${data[data.findIndex(function(x) { return x.id == target.id })].num}</li > <li class = "value pay\_type" > $ {

data[data.findIndex(function(x) {

return x.id == target.id

})].pay\_type

} < /li>

<li class="value note">${data[data.findIndex(function(x) { return x.id == target.id })].note}</li > <li style = "display:inline-block" > <button type = "button"class = "am-close"id = "${data[data.findIndex(function(x) { return x.id == target.id })].id}" > &times; < /button></li > `;

console.log(target.parentNode.parentNode.parentNode.firstElementChild.firstElementChild.firstElementChild.innerHTML) console.log(target.parentNode.parentNode.querySelector('.sortBtn').querySelector('select')[target.parentNode.parentNode.querySelector('.sortBtn').querySelector('select').selectedIndex].text) $.ajax({

url: "http:/127.0.0.1:8022/updata",

type: "get",

data: {

id: `$ {

target.id

}`,

type: target.parentNode.parentNode.querySelector('.sortBtn').querySelector('select')[target.parentNode.parentNode.querySelector('.sortBtn').querySelector('select').selectedIndex].text,

pay\_type: target.parentNode.parentNode.querySelector('.moneyBtn').querySelector('select')[target.parentNode.parentNode.querySelector('.sortBtn').querySelector('select').selectedIndex].text,

num: Number($('.figureBtn').find('input').val()),

note: target.parentNode.querySelector('.note').querySelector('input').value,

time: target.parentNode.querySelector('.time').querySelector('input').value

},

async: false,

success: function(res) {

//data = res.data

console.log(res);

},

error: function(err) {

}

})

}

}

}

//搜索

var key1 = '餐饮费',

key2 = '现金';

document.querySelector(".specialListNav").onclick = function() {

var ev = ev || window.event;

var target = ev.target || ev.srcElement;

console.log(target.innerText) if (target.innerText == key1 || target.innerText == key2) {} else {

switch (target.innerText) {

case '餐饮费':

case '服饰美容':

case '生活日用':

case '交通出行':

case '通讯物流':

case '住房缴费':

case '文教娱乐':

case '健康':

case '其他消费':

$.ajax({

url:

"http:/127.0.0.1:8022/find",

type: "get",

data: {

type: target.innerText

},

async: false,

success: function(res) {

data = res.data;

key1 = target.innerText;

document.querySelector("#accordion").innerHTML = '';

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

$('.am-panel-group').append(` <div class = &quot;am-panel am-panel-default&quot; &gt;

<div class="am-panel-hd">

<h4 class="am-panel-title" data="" -="" am="" collapse="{parent: '#accordion', target: '#do-not-say-${data[i].id}'}">

<ul class="list flex">

<li> $ { data[i].id } &lt; /li&gt; </li>

<li>${data[i].time}</li>

<li class="value type"> $ { data[i].type } &lt; /li&gt; </li>

<li class="value num">${data[i].num}</li>

<li class="value pay\_type"> $ { data[i].pay\_type } &lt; /li&gt; </li>

<li class="value note">${data[i].note}</li>

<li style="display:inline-block"> <button type="button" class="am-close" id="${data[i].id}"> &times; &lt; /button&gt;</button></li>

</ul> </h4>

</div>

<div id="do-not-say-${data[i].id}" class="am-panel-collapse am-collapse">

<div class="am-panel-bd flex">

<div class="sortBtn">

<select data-am-selected=""> <option value="a">餐饮费</option> <option value="b"> 服饰美容 &lt; /option&gt; </option><option value="c"> 生活日用</option> <option value="d"> 住房缴费 &lt; /option&gt; </option><option value="e">交通出行</option> <option value="f"> 通讯物流 &lt; /option&gt; </option><option value="g">文教娱乐</option> <option value="h"> 健康 &lt; /option&gt; </option><option value="i">其他消费</option> </select>

</div>

<div class="moneyBtn">

<select data="" -="" am="" selected=""> <option value="a"> 现金 &lt; /option&gt; </option><option value="b">银行卡</option> <option value="c"> 支付宝 &lt; /option&gt; </option><option value="d">信用卡</option> <option value="e"> 余额宝 &lt; /option&gt; </option></select>

</div>

<div class="figureBtn">

<div class="text">

<p><input type="number" class="am-form-field am-radius" value="${data[i].num}" /> </p>

</div>

</div>

<div class="time">

<div class="am-input-group am-datepicker-date" data-am-datepicker="{format: 'yyyy-mm-dd'}">

<input type="text" class="am-form-field" value="${data[i].time}" readonly="" />

<span class="am-input-group-btn am-datepicker-add-on"> <button class="am-btn am-btn-default" type="button"><span class="am-icon-calendar"></span> </button> </span>

</div>

</div>

<div class="note flex">

<p> <input type="text" class="am-form-field am-radius" value="${data[i].note}" /></p>

</div>

<button type="button" class="am-btn am-btn-primary am-btn-block alter" id="${data[i].id}"> 修改 &lt; /button&gt; </button>

</div>

</div>

</body>

</html> `);

}

},

error: function(err) {

}

}) break;

case '现金':

case '信用卡':

case '银行卡':

case '支付宝':

case '余额宝':

$.ajax({

url:

"http:/127.0.0.1:8022/find",

type: "get",

data: {

pay\_type: target.innerText

},

async: false,

success: function(res) {

data = res.data;

key2 = target.innerText;

document.querySelector("#accordion").innerHTML = '';

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

$('.am-panel-group').append(` <div class = &quot;am-panel am-panel-default&quot; &gt;

<div class="am-panel-hd">

<h4 class="am-panel-title" data="" -="" am="" collapse="{parent: '#accordion', target: '#do-not-say-${data[i].id}'}">

<ul class="list flex">

<li> $ { data[i].id } &lt; /li&gt; </li>

<li>${data[i].time}</li>

<li class="value type"> $ { data[i].type } &lt; /li&gt; </li>

<li class="value num">${data[i].num}</li>

<li class="value pay\_type"> $ { data[i].pay\_type } &lt; /li&gt; </li>

<li class="value note">${data[i].note}</li>

<li style="display:inline-block"> <button type="button" class="am-close" id="${data[i].id}"> &times; &lt; /button&gt;</button></li>

</ul> </h4>

</div>

<div id="do-not-say-${data[i].id}" class="am-panel-collapse am-collapse">

<div class="am-panel-bd flex">

<div class="sortBtn">

<select data-am-selected=""> <option value="a">餐饮费</option> <option value="b"> 服饰美容 &lt; /option&gt; </option><option value="c"> 生活日用</option> <option value="d"> 住房缴费 &lt; /option&gt; </option><option value="e">交通出行</option> <option value="f"> 通讯物流 &lt; /option&gt; </option><option value="g">文教娱乐</option> <option value="h"> 健康 &lt; /option&gt; </option><option value="i">其他消费</option> </select>

</div>

<div class="moneyBtn">

<select data="" -="" am="" selected=""> <option value="a"> 现金 &lt; /option&gt; </option><option value="b">银行卡</option> <option value="c"> 支付宝 &lt; /option&gt; </option><option value="d">信用卡</option> <option value="e"> 余额宝 &lt; /option&gt; </option></select>

</div>

<div class="figureBtn">

<div class="text">

<p><input type="number" class="am-form-field am-radius" value="${data[i].num}" /> </p>

</div>

</div>

<div class="time">

<div class="am-input-group am-datepicker-date" data-am-datepicker="{format: 'yyyy-mm-dd'}">

<input type="text" class="am-form-field" value="${data[i].time}" readonly="" />

<span class="am-input-group-btn am-datepicker-add-on"> <button class="am-btn am-btn-default" type="button"><span class="am-icon-calendar"></span> </button> </span>

</div>

</div>

<div class="note flex">

<p> <input type="text" class="am-form-field am-radius" value="${data[i].note}" /></p>

</div>

<button type="button" class="am-btn am-btn-primary am-btn-block alter" id="${data[i].id}"> 修改 &lt; /button&gt; </button>

</div>

</div>

</body>

</html> `);

}

},

error: function(err) {

}

}) break;

case '搜索':

$.ajax({

url:

"http:/127.0.0.1:8022/find",

type: "get",

data: {

note: document.querySelector("#searchNote").value

},

async: false,

success: function(res) {

data = res.data;

document.querySelector("#accordion").innerHTML = '';

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

$('.am-panel-group').append(` <div class = &quot;am-panel am-panel-default&quot; &gt;

<div class="am-panel-hd">

<h4 class="am-panel-title" data="" -="" am="" collapse="{parent: '#accordion', target: '#do-not-say-${data[i].id}'}">

<ul class="list flex">

<li> $ { data[i].id } &lt; /li&gt; </li>

<li>${data[i].time}</li>

<li class="value type"> $ { data[i].type } &lt; /li&gt; </li>

<li class="value num">${data[i].num}</li>

<li class="value pay\_type"> $ { data[i].pay\_type } &lt; /li&gt; </li>

<li class="value note">${data[i].note}</li>

<li style="display:inline-block"> <button type="button" class="am-close" id="${data[i].id}"> &times; &lt; /button&gt;</button></li>

</ul> </h4>

</div>

<div id="do-not-say-${data[i].id}" class="am-panel-collapse am-collapse">

<div class="am-panel-bd flex">

<div class="sortBtn">

<select data-am-selected=""> <option value="a">餐饮费</option> <option value="b"> 服饰美容 &lt; /option&gt; </option><option value="c"> 生活日用</option> <option value="d"> 住房缴费 &lt; /option&gt; </option><option value="e">交通出行</option> <option value="f"> 通讯物流 &lt; /option&gt; </option><option value="g">文教娱乐</option> <option value="h"> 健康 &lt; /option&gt; </option><option value="i">其他消费</option> </select>

</div>

<div class="moneyBtn">

<select data="" -="" am="" selected=""> <option value="a"> 现金 &lt; /option&gt; </option><option value="b">银行卡</option> <option value="c"> 支付宝 &lt; /option&gt; </option><option value="d">信用卡</option> <option value="e"> 余额宝 &lt; /option&gt; </option></select>

</div>

<div class="figureBtn">

<div class="text">

<p><input type="number" class="am-form-field am-radius" value="${data[i].num}" /> </p>

</div>

</div>

<div class="time">

<div class="am-input-group am-datepicker-date" data-am-datepicker="{format: 'yyyy-mm-dd'}">

<input type="text" class="am-form-field" value="${data[i].time}" readonly="" />

<span class="am-input-group-btn am-datepicker-add-on"> <button class="am-btn am-btn-default" type="button"><span class="am-icon-calendar"></span> </button> </span>

</div>

</div>

<div class="note flex">

<p> <input type="text" class="am-form-field am-radius" value="${data[i].note}" /></p>

</div>

<button type="button" class="am-btn am-btn-primary am-btn-block alter" id="${data[i].id}"> 修改 &lt; /button&gt; </button>

</div>

</div>

</body>

</html> `);

}

},

error: function(err) {

}

}) break;

}

}

}

//排序

$("#sort").click(function() {

$.ajax({

url: "http:/127.0.0.1:8022/sort",

data: {},

type: "get",

async: false,

success: function(res) {

data = res.data;

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

$('.am-panel-group').append(`<div class = &quot;am-panel am-panel-default&quot; &gt;

<div class="am-panel-hd">

<h4 class="am-panel-title" data="" -="" am="" collapse="{parent: '#accordion', target: '#do-not-say-${data[i].id}'}">

<ul class="list flex">

<li> $ { data[i].id } &lt; /li&gt; </li>

<li>${data[i].time}</li>

<li class="value type"> $ { data[i].type } &lt; /li&gt; </li>

<li class="value num">${data[i].num}</li>

<li class="value pay\_type"> $ { data[i].pay\_type } &lt; /li&gt; </li>

<li class="value note">${data[i].note}</li>

<li style="display:inline-block"> <button type="button" class="am-close" id="${data[i].id}"> &times; &lt; /button&gt;</button></li>

</ul> </h4>

</div>

<div id="do-not-say-${data[i].id}" class="am-panel-collapse am-collapse">

<div class="am-panel-bd flex">

<div class="sortBtn">

<select data-am-selected=""> <option value="a">餐饮费</option> <option value="b"> 服饰美容 &lt; /option&gt; </option><option value="c"> 生活日用</option> <option value="d"> 住房缴费 &lt; /option&gt; </option><option value="e">交通出行</option> <option value="f"> 通讯物流 &lt; /option&gt; </option><option value="g">文教娱乐</option> <option value="h"> 健康 &lt; /option&gt; </option><option value="i">其他消费</option> </select>

</div>

<div class="moneyBtn">

<select data="" -="" am="" selected=""> <option value="a"> 现金 &lt; /option&gt; </option><option value="b">银行卡</option> <option value="c"> 支付宝 &lt; /option&gt; </option><option value="d">信用卡</option> <option value="e"> 余额宝 &lt; /option&gt; </option></select>

</div>

<div class="figureBtn">

<div class="text">

<p><input type="number" class="am-form-field am-radius" value="${data[i].num}" /> </p>

</div>

</div>

<div class="time">

<div class="am-input-group am-datepicker-date" data-am-datepicker="{format: 'yyyy-mm-dd'}">

<input type="text" class="am-form-field" value="${data[i].time}" readonly="" />

<span class="am-input-group-btn am-datepicker-add-on"> <button class="am-btn am-btn-default" type="button"><span class="am-icon-calendar"></span> </button> </span>

</div>

</div>

<div class="note flex">

<p> <input type="text" class="am-form-field am-radius" value="${data[i].note}" /></p>

</div>

<button type="button" class="am-btn am-btn-primary am-btn-block alter" id="${data[i].id}"> 修改 &lt; /button&gt; </button>

</div>

</div>

</body>

</html> `);

}

},

error: function(err) {

}

})

})

}

**五、测试数据及其结果分析**

运行后，程序的主菜单界面如图8所示。

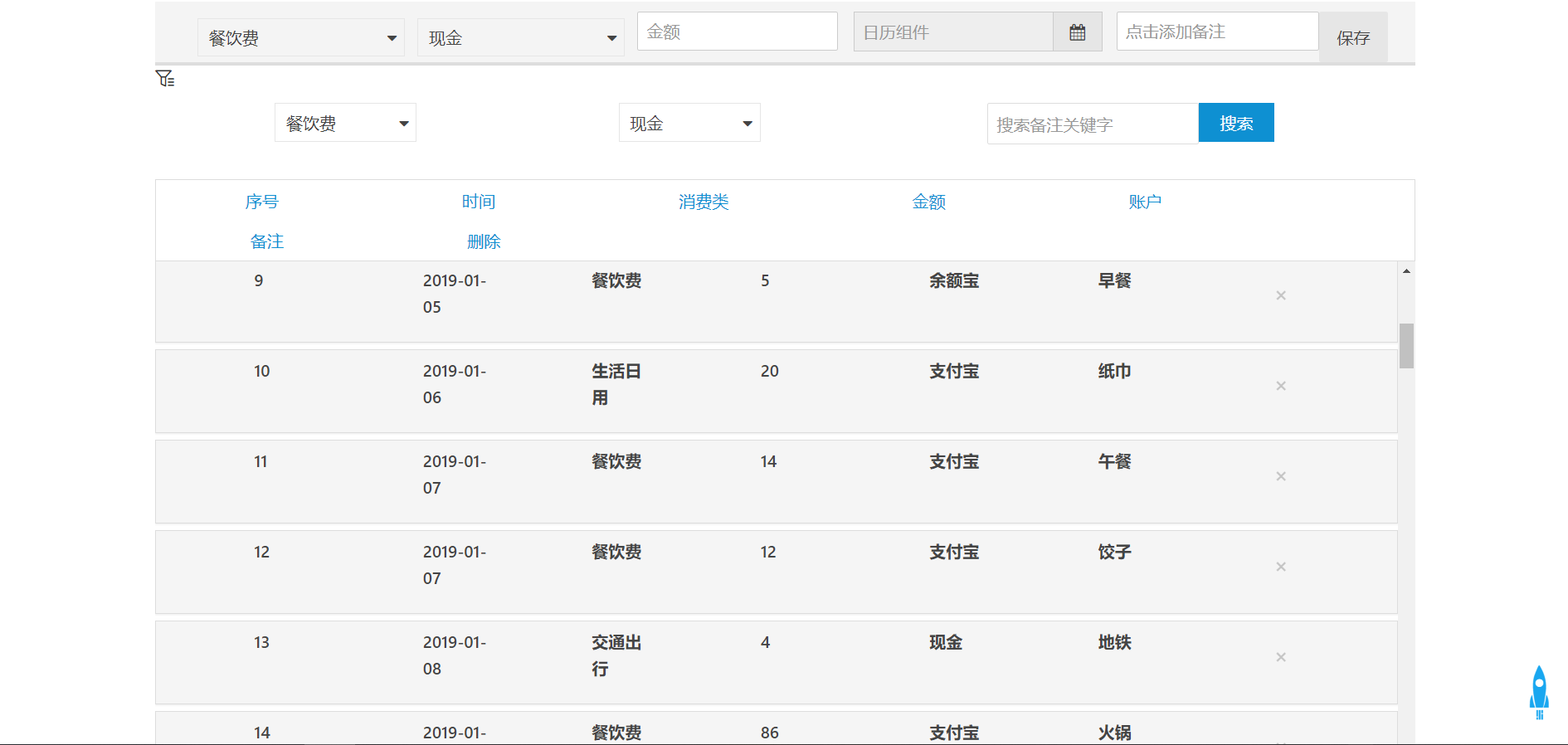


图8 程序的主菜单界面

（1）查找用户测试

输入：鼠标点击消费类型按钮，滚轮滑动选择筛选的消费类型“住房缴费”

输出：找到的所有消费类型为“住房缴费”的明细



图9 按消费类型的查找用户测试

输入：鼠标点击消费类型按钮，滚轮滑动选择筛选的账户类型“支付宝”

输出：找到的所有账户类型为“支付宝”的明细



图10 按账户类型的查找用户测试

输入：鼠标点击搜索关键字，输入关键字“一点点”

输出：找到的所有关键字“一点点”的明细



图11 按关键字的查找用户测试

（2）插入客户测试

输入：鼠标点击消费类型按钮，滚轮滑动选择增加的消费类型“餐饮费”——鼠标点击账户类型按钮，滚轮滑动选择增加的账户类型“现金”——直接输入/点击金额输入框旁的按钮，若收入则数据为正，若支出数据为负“-10”——点击日历进行时间记录——“2019-03-29”点击备注输入备注“测试demo1”——点击保存

输出：明细表中增加新增账目类型第121条



图12 新增账目类型第121条

输入：鼠标点击消费类型按钮，滚轮滑动选择增加的消费类型“餐饮费”——鼠标点击账户类型按钮，滚轮滑动选择增加的账户类型“现金”——直接输入/点击金额输入框旁的按钮，若收入则数据为正，若支出数据为负“-6”——点击日历进行时间记录——“2019-03-29”点击备注输入备注“测试demo2”——点击保存

输出：明细表中增加新增账目类型第124条

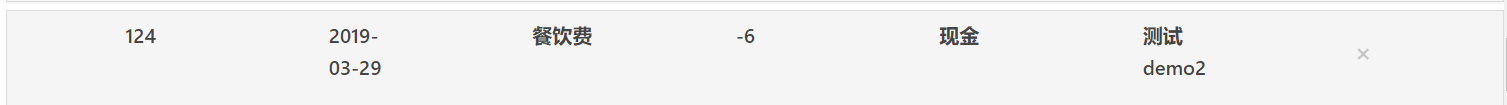


图13 新增账目类型第124条

（3）客户排序测试

点击排序按钮后数据按时间排序，如图14所示。



图14 数据按时间排序

（4）统计测试

输出：饼图形式展现用户所有账目的消费类型所占比

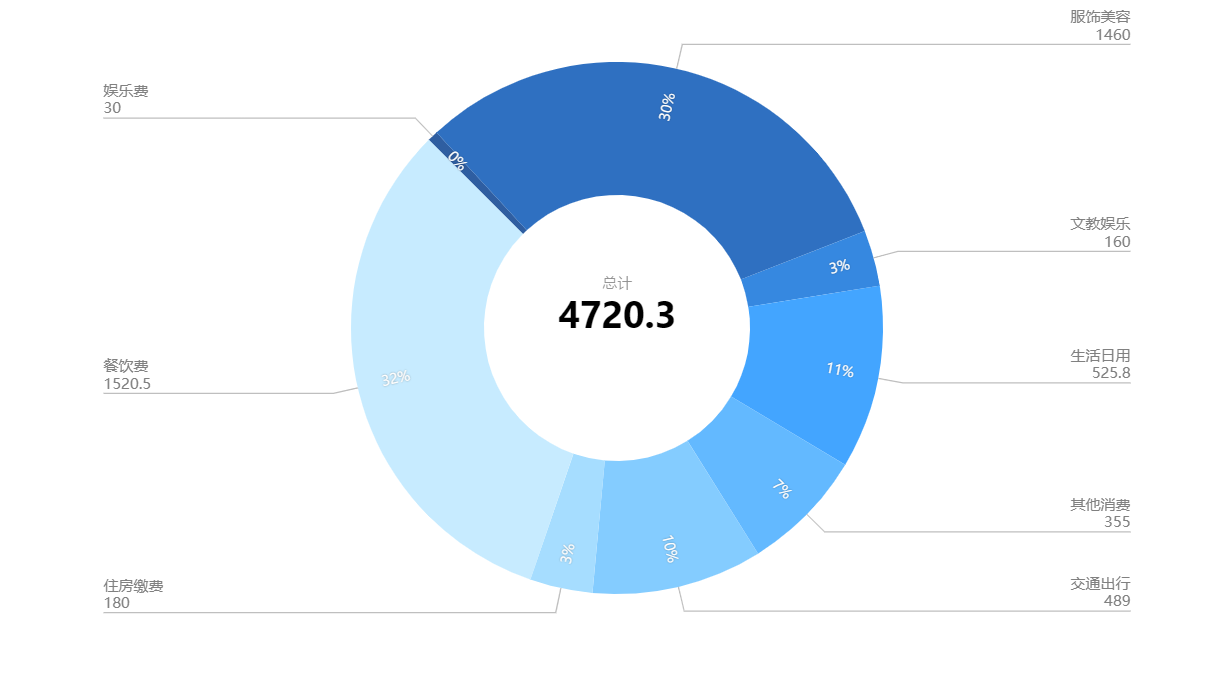


图15 用户所有账目的消费类型饼图

输出：折线图形式展现用户所有账目的每日的消费类型总额

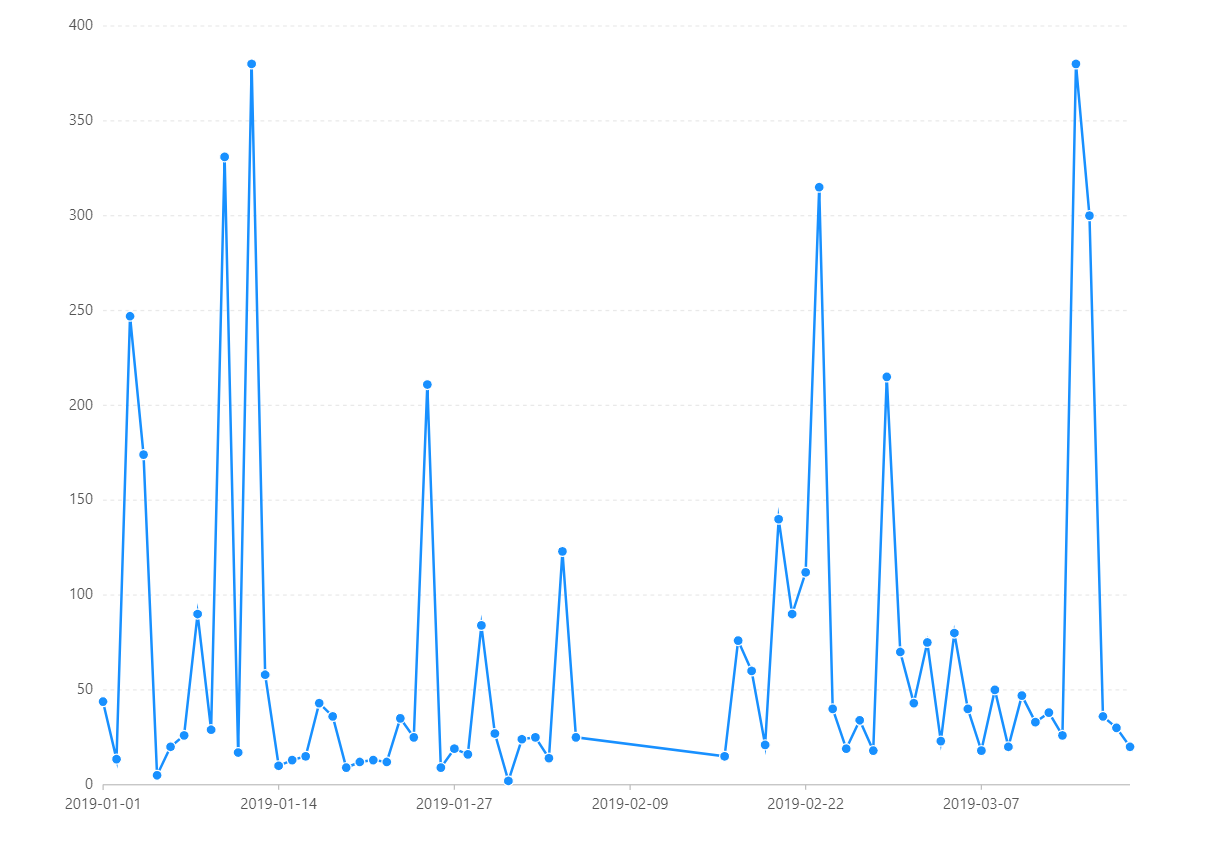


图15 用户所有账目的每日的消费类型总额折线图

输出：条形图形式展现用户所有账目的账户类型所占比

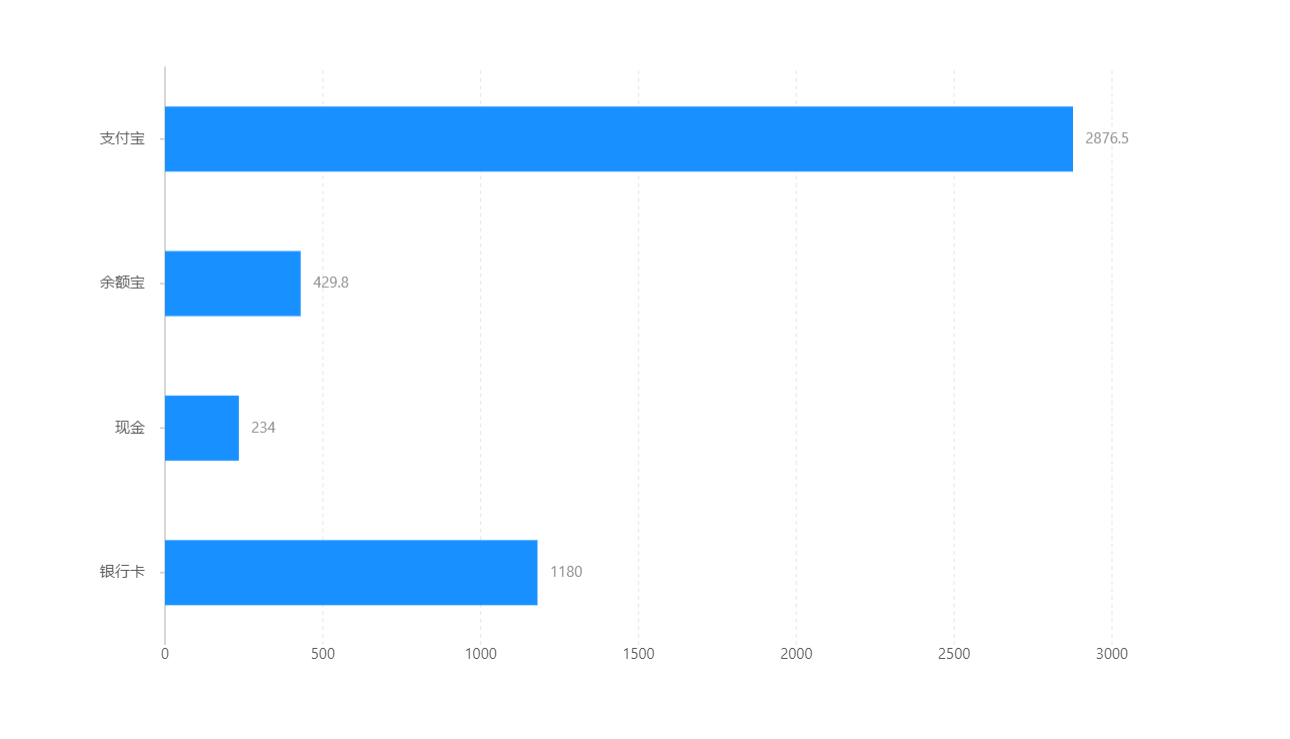


图16 用户所有账目的账户类型条形图

（5）退出测试

关闭网页界面。

结果分析如下。

（1）如果用户没有点击金额就进行保存，会弹出提示框“没有保存新纪录”。

（2）如果用户对数据进行操作时，前后端没有达到同步时，会弹出提示框“未同步”。

**六、调试过程中的问题**

（1）问题：前后端数据没有同步更新

解决方法：利用ajax调用接口传入的数据取错，应传入id。

（2）问题：数据添加位置错乱

解决方法：因为要跟后端数据库保持同步，所以在添加数据时应当是先有数据，再创建dom树，而不是直接对dom树进行操作。

（3）问题：删除功能错乱

解决方法：原本直接让各条数据列表与各自列表的修改按钮的id绑定，但这样会给删除按钮绑上很多函数，且删除以后，数据库中各条数据的id并不会重新排列，因此删除功能只能实现一次。后改为利用用户当前点击对象去寻找对应需要操作的数据列表进而达到删除正确数据的目的。

（4）问题：修改后出现的文本错乱

解决方法：原本是用选择器来选择对应的数据类型，但获得到的是所有的消费类型按钮的被选中值，因而修改后会出现多个文本，后改用选择器达到目的。

（5）问题：图表没有与预想的那样实时更新

解决方法：因为传入数据的时候忘记更改数据来源，后改正。

（6）问题：修改后时间与金额不变

解决方法：因为选择器用错了，而且格式错误导致没有取到修改的值，后改正。

（7）问题：手写出来的页面色彩搭配差距大，不美观

解决方法：调Amaze UI库引用组件使界面更为简洁美观。

（8）问题：查找时只能出现同时满足多条筛选需求而不能满足单一筛选需求

解决方法：原因是因为实现时是取到数据列表下的每一个被选中的筛选类型值，因而只能多重条件筛选。后改用switch语句并使用截流达到目的。

**七、课程总结**

组长：B18030406张颖：在这次程序设计过程中，经历了前后端数据没有同步更新、数据添加位置错乱、删除功能错乱、修改后文本错乱、查找时只能出现同时满足多条筛选需求而不能满足单一筛选需求等等一系列的问题。但比较庆幸的是，这些问题都解决了。这一次程序设计是我第一次以组长的身份，去决定整体布局、框架、色调等问题；去考虑组员的实际能力而分别分工，大体上做到了合理分工。虽然这个题目本身仅是实现增删改查的功能，但我们自己增加了一些比较创新的东西：比如图形化界面、数据可视化、数据库管理数据、前后端同步更新等。不过也正是因为这些创新，我也学会了许多新技术，也算是充实了自己吧！

组员：B18030408胡欣然：通过这次的程序设计周，不仅让我有机会巩固了以前所学的知识，还以此为契机学习到了很多新东西。由于我们组选择用HTML、CSS和JS来实现个人账本管理系统的页面设计和功能实现，而我分配到的任务是首页的设计和图表的构建，所以也让我有机会对我之前所学的网页设计有了进一步的了解和实际应用的实现。我还学会了利用Amaze UI来做前端框架，引用里面的组件来让页面更加丰富和美观。但对于图表的构建我先前并不是很熟悉，也是这次机会让我有机会认识和学习了数据可视化。为了和后端实现实时更新，所以在组长的建议下我又去学习了Ajax，并让组长教了我如何访问后端接口。总而言之，程序设计周对我而言既是充满了挑战的一周又是开拓我眼界的一次难得的机会！

组员：B18030405杨路玥：首先这次的实验周大大增加了我对后端的理解。由于我写的是账本的增删改查中的增加记录部分，原来我只会在网页中直接对dom树执行操作，在这次程序设计中，我原先会的将新数据直接添加到dom树的方式，并不能让后端数据库接收到这份数据，仅仅只是网页展示出来而已；所以我这次学会了先有数据，后创建dom树，也就是创建一个虚拟dom树来对数据进行操作，而不是对dom树进行操作，这样才能保证与后端数据库的实时更新。还有一点是我原先在写网页时，只能自己手写样式，但由于总体色调差距过大导致网页不够美观。后来我学会了如何调用Amaze UI，利用它来做整体框架，从中引用组件使网页更加简洁。