

《移动应用软件开发》实验报告

年级、专业、班级	2021 级计算机科学与技术 3 班	姓名	刘成	学号	20215005
实验题目	收支记账 APP				
实验时间	2023. 11. 29	实验地点	DS1401		
学年学期	2023-2024(1)	实验性质	<input type="checkbox"/> 验证性 <input checked="" type="checkbox"/> 设计性 <input type="checkbox"/> 综合性		
<p>一、实验目的</p> <ol style="list-style-type: none">1. 掌握基于 HarmonyOS 平台，使用“声明式开发范式”开发 APP 的方法。2. 能够使用 ArkUI 对话框、各种布局、组件构建复杂界面，并且能够使用手机本地数据库表存放数据。或者把关系数据存放在网络后台系统。3. 在 HarmonyOS 平台上调试完成 APP，限定使用 ArkTS 语言编写代码。					
<p>二、实验项目内容</p> <p>完成的软件基本功能包括：（1）分类记录日常收入和支出数据，包括金额、用途、日期、分类等信息；能够增加、删除等，可以支持侧滑删除。（2）显示月、日收支流水记录，显示当月、当日收支总额；（3）按周、月、年进行汇总形成结果。（4）具备友好的用户数据输入界面和查询界面。</p> <p>评分点：（1）首页界面、功能、（2）信息输入界面、功能。（3）查询汇总显示界面、功能。（4）界面设计、布局合理。</p> <p>加分点：（1）能够根据输入条件查询，例如金额范围、用途、日期范围、分类等。（2）其他自创功能。（3）界面美观漂亮。（4）统计图表，如饼图，折线图等（至少具备其一可以加分）。</p> <p>提交：（1）本实验报告。（2）源代码压缩文件 zip，注意源代码加注释。（3）软件演示的 MP4 视频，视频大小不超过 20M，并用搜狗浏览器检查视频能否正常播放。分三个文件提交，注意文件名称的规范性。文件名：学号姓名 3.doc，学号姓名 3.zip，学号姓名 3.mp4。</p>					
<p>三、实验过程或算法（写明设计思想、程序的结构、功能关系图、类的说明和类之间的关系图、程序主要执行流程图，最后是核心源代码，截图等）</p> <p>1. 设计思想</p> <p>1.1 连接手机数据库实现方式</p> <p>引入 stage 模型中的关系型数据库，导入 ohos.data.relationalStore 库中的 relationalStore 类，在 Rdb.ts 中定义重写类方法，定义数据库的创建语句以及新建表的名称，构建 insertData, deleteDataup, dateData 和</p>					

query 增删查改方法。然后在 AccountTable.ets 中详细的定义了数据库表 ACCOUNT_TABLE 中的 id, accountType, typeText, amount, ps, year, month, day 各个键的名称和类型, 以及实现了增删查改每个方法的具体重载。

1.2 记账本首页的实现方式

首页中通过 Row-Column 布局, 首先定义了表头的“记账本”标题名称, 当日收入和当日支出的文本框, 其中通过 @State 方式绑定了当日的收入和支出金额, 通过 if-else 的动态渲染方式, 一旦发生金额变化, 就会同步发生改变。具体的, 首先在生命周期 aboutToAppear 函数中初始化数据库 AccountTable 表, 并通过 query 函数查询数据库具体的每一笔收支信息, 再将对应的值渲染到程序上。同时使用 DatePickerDialog 配合文本框的点击事件, 实现了更换日期的功能。下方的功能选项布局通过调整 margin 和 borderRadius 实现了悬浮在标题下方的效果, 其中通过 SpaceEvenly 布局使每个功能图标布局更加美观, 不同的图标绑定了对应的 router 路由, 跳转到不同页面。下方当日记账的明细, 通过 ForEach 函数, 为数据库中取得的每个 AccountData 项动态渲染在页面上, 并通过 if-else 结合 @State 类型的变量, 实现了有无备注的明细不同的显示方式。每一个明细通过 Scroll 和 scrollable (Scroll Direction.Horizontal) 相结合的方式, 实现了右滑删除的功能。点击删除按钮会弹出提示框提醒是否删除, 点击删除后对应的判断变量会置否, 从而达到不显示在页面上的效果。

1.3 支出和收入页面的实现方式

通过 TabsController 和 Tabs 组件, 让菜单栏置于页面顶端, 滑动菜单栏或者页面, 可以实现页面翻转。页面中每一个支出(收入)类型的标签首先封装在 ItemData 类中, 再使用 gridData 存放所有标签的 id、图片和文字信息, 使用 grid 布局, 将标签按照每行 4 个的方式排列在页面中。点击对应的标签, 通过 onClick 绑定点击事件, 标签的背景颜色会变成黄色, 同时弹出自定义弹窗, 输入对应的金额和选择日期。自定义弹窗用 CustomDialog 类进行复写, 通过 grid 布局实现 4*4 的输入框布局, 通过 if-else 绑定不同按钮的点击事件, 实现删除、清空和选择日期的功能。若点击页面蒙层, 使用回调函数将选择的标签背景变回原来的颜色。若输入正确的金额并点击确定后, 填写的参数会通过 @Link 类型的变量与父组件交互, 同时通过 insert 方式存储到数据库中, 页面跳转到首页并将记账明细重新渲染在页面上。

1.3 月账单的实现方式

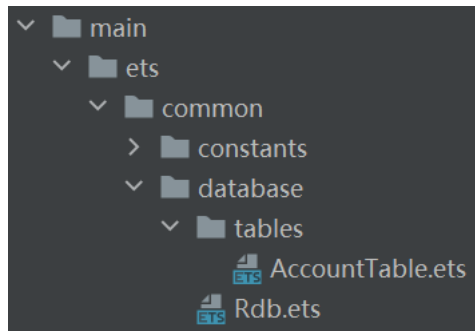
首先在 AboutToAppear 生命周期函数中查询到每个月收入和支出记录, 再统计当月的总收入和总支出, 计算月结余金额, 存储在 MonthCount 列表中, 同时就年收入、年支出和年支出, 用于展示在页面上方。在页面中通过 ForEach 函数便利列表中每一个月的明细记录, 渲染在页面上。

1.4 每月明细和年账单的实现方式

在月账单跳转到月明细页面时, router 中 params 参数传递对应的月份以及支出、收入、结余、记账天数和记账笔数信息, 在明细页面中通过 router.getParamas 函数接收对应的参数, 渲染在明细页面中对应的位置。在通过网页加载的方式, 传递月份的收入和支出排名前五的明细内容到 Web 组件中, Web 组件通过调用统计表图 api, 将返回的绘制的饼图、条形图和折线图通过调整 Web 组件的比例渲染在页面上。整个页面通过 Scroll 结合页面高度的调整, 实现滑动效果。

2. 程序结构

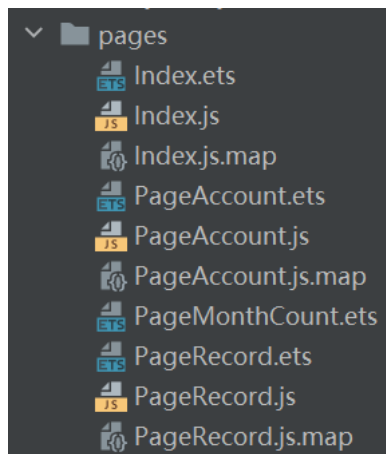
2.1 src.main.common.database 下的数据库配置文件



Rdb.ets 数据库的创建配置类

AccountTable.ets 数据库增删查改具体实现类

2.2 src.main.pages 下的页面布局文件



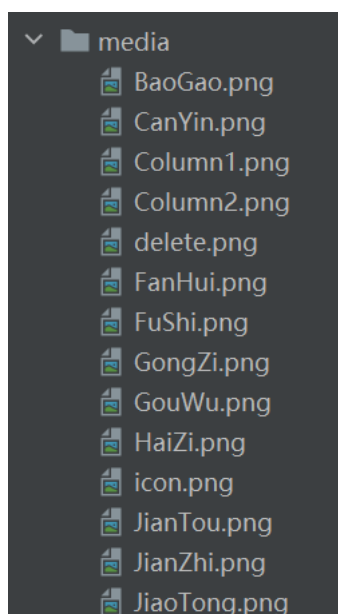
Index.ets 记账本页面首页

PageRecord.ets 记录支出和收入的记账页面

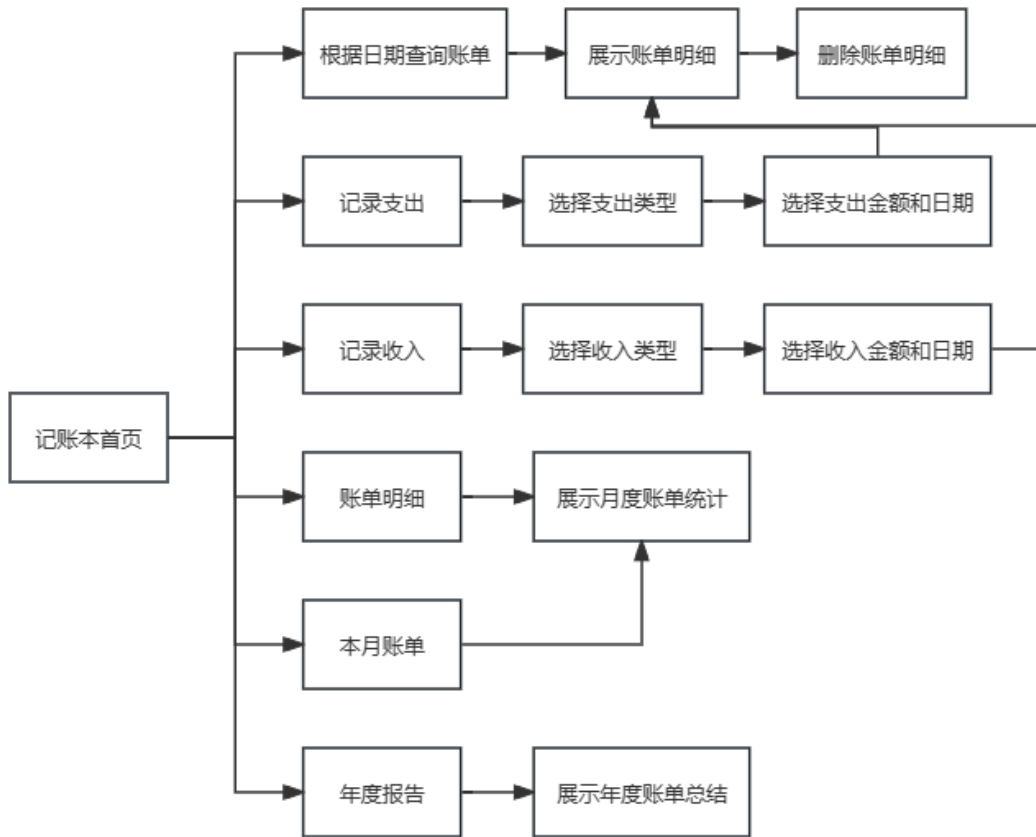
PageAccount.ets 月账单和年账单页面

PageMonthAccount.ets 月账单明细页面

2.3 main.resources.media 中的图片资源文件（仅列举部分）



3. 功能关系图



4. 类的说明和类之间的关系图

4.1 Index.ets

当日支出和收入的明细页面，定义了右滑删除记账明细方法，标题上方中记录当日支出和收入的总额，定义了通过日期查询金额的点击事件，同时定义了点击不同的功能跳转到对应的页面的功能。

4.2 PageRecord.ets

定义了支出和收入菜单栏，分别展示了不同的支出和收入类型，定义了不同标签的点击事件，定义了自定义弹窗中不同按钮的功能实现方式，定义了数据库插入对应记账明细的功能。

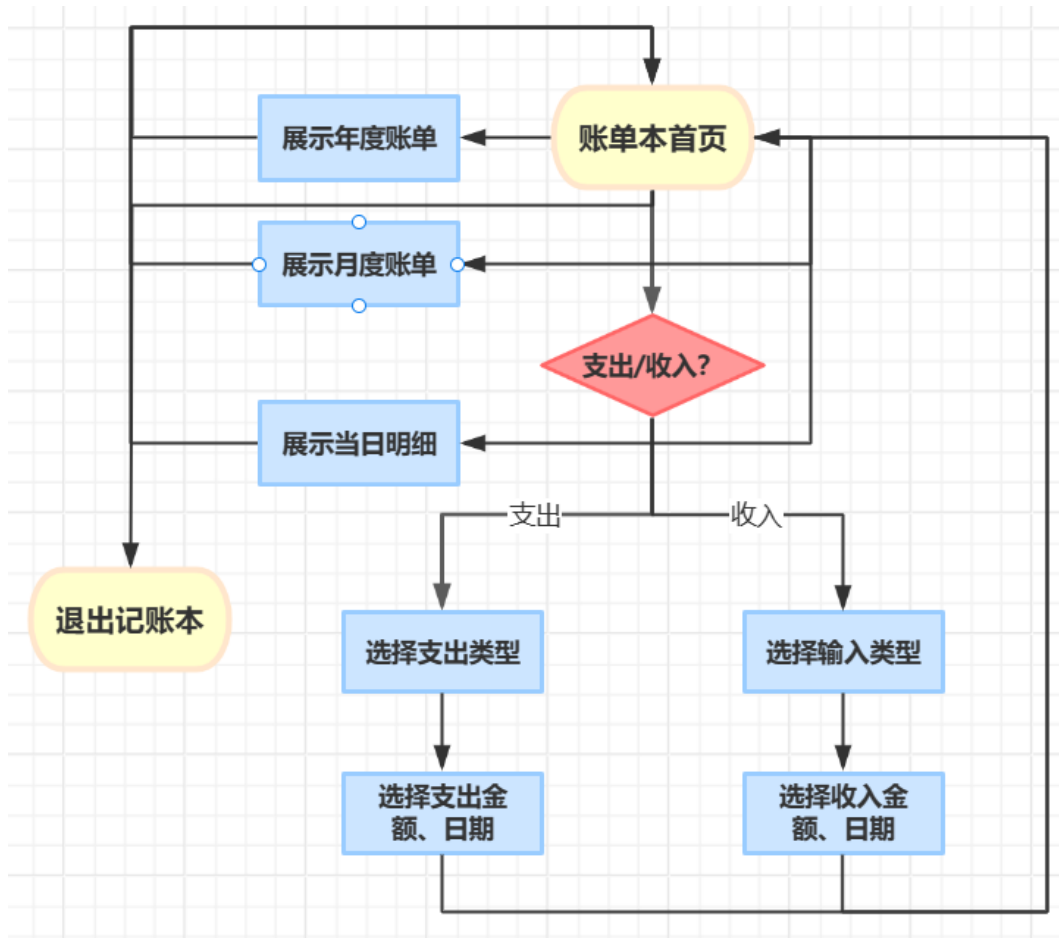
4.3 PageAccount.ets

定义了月账单和年账单菜单栏，展示了年支出、年收入、年结余和总支出、总收入、总结余，定义了通过查询数据库对应月份的明细渲染在页面中的方法，以及为每个页面定义了页面跳转的点击事件。

4.4 PageMonthCount.java

展示了具体月份的本月结余、上月结余以及支出、收入、记账天数和记账笔数，同时将支出类别、月支出对比、收入类别和月收入对比对应的统计图表渲染在页面上。

5. 程序主要流程图



6. 核心代码

6.1.1 数据库整体配置代码

```

export default class Rdb {
  private rdbStore: relationalStore.RdbStore | null = null;
  private tableName: string;
  private sqlCreateTable: string;
  private columns: Array<string>;

  constructor(tableName: string, sqlCreateTable: string, columns:
Array<string>) {
    this.tableName = tableName;
    this.sqlCreateTable = sqlCreateTable;
    this.columns = columns;
  }

  getRdbStore(callback: Function = () => {
  }) {
    if (!callback || typeof callback === 'undefined' || callback ===
undefined) {
      Logger.info(CommonConstants.RDB_TAG, 'getRdbStore() has no

```

```
callback!');
    return;
}
if (this.rdbStore !== null) {
    Logger.info(CommonConstants.RDB_TAG, 'The rdbStore exists.');
```

callback();

return

}

let context: Context = getContext(this) as Context;

relationalStore.getRdbStore(context, CommonConstants.STORE_CONFIG,

(err, rdb) => {

if (err) {

Logger.error(CommonConstants.RDB_TAG, `getRdbStore() failed, err: \${err}`);

return;

}

this.rdbStore = rdb;

this.rdbStore.executeSql(this.sqlCreateTable);

Logger.info(CommonConstants.RDB_TAG, 'getRdbStore() finished.');

callback();

});

}

insertData(data: relationalStore.ValuesBucket, callback: Function = ()

=> {

}) {

if (!callback || typeof callback === 'undefined' || callback ===

undefined) {

Logger.info(CommonConstants.RDB_TAG, 'insertData() has no

callback!');

return;

}

let resFlag: boolean = false;

const valueBucket: relationalStore.ValuesBucket = data;

if (this.rdbStore) {

this.rdbStore.insert(this.tableName, valueBucket, (err, ret) => {

if (err) {

Logger.error(CommonConstants.RDB_TAG, `insertData() failed,

err: \${err}`);

callback(resFlag);

return;

}

Logger.info(CommonConstants.RDB_TAG, `insertData() finished:

\${ret}`);

```
        callback(ret);
    });
}
}

deleteData(predicates: relationalStore.RdbPredicates, callback:
Function = () => {
    }) {
    if (!callback || typeof callback === 'undefined' || callback ===
undefined) {
        Logger.info(CommonConstants.RDB_TAG, 'deleteData() has no
callback!');
        return;
    }
    let resFlag: boolean = false;
    if (this.rdbStore) {
        this.rdbStore.delete(predicates, (err, ret) => {
            if (err) {
                Logger.error(CommonConstants.RDB_TAG, `deleteData() failed,
err: ${err}`);
                callback(resFlag);
                return;
            }
            Logger.info(CommonConstants.RDB_TAG, `deleteData() finished:
${ret}`);
            callback(!resFlag);
        });
    }
}

updateData(predicates: relationalStore.RdbPredicates, data:
relationalStore.ValuesBucket, callback: Function = () => {
    }) {
    if (!callback || typeof callback === 'undefined' || callback ===
undefined) {
        Logger.info(CommonConstants.RDB_TAG, 'updateDate() has no
callback!');
        return;
    }
    let resFlag: boolean = false;
    const valueBucket: relationalStore.ValuesBucket = data;
    if (this.rdbStore) {
        this.rdbStore.update(valueBucket, predicates, (err, ret) => {
            if (err) {
```

```

        Logger.error(CommonConstants.RDB_TAG, `updateData() failed,
err: ${err}`);
        callback(resFlag);
        return;
    }
    Logger.info(CommonConstants.RDB_TAG, `updateData() finished:
${ret}`);
    callback(!resFlag);
    });
}
}

query(predicates: relationalStore.RdbPredicates, callback: Function = ()
=> {
    }) {
    if (!callback || typeof callback === 'undefined' || callback ===
undefined) {
        Logger.info(CommonConstants.RDB_TAG, 'query() has no callback!');
        return;
    }
    if (this.rdbStore) {
        this.rdbStore.query(predicates, this.columns, (err, resultSet) => {
            if (err) {
                Logger.error(CommonConstants.RDB_TAG, `query() failed, err:
${err}`);
                return;
            }
            Logger.info(CommonConstants.RDB_TAG, 'query() finished. ');
            callback(resultSet);
            resultSet.close();
        });
    }
}
}

```

6.1.2 AccountTable 增删查改操作代码

```

export default class AccountTable {
    private accountTable = new Rdb(CommonConstants.ACCOUNT_TABLE.tableName,
CommonConstants.ACCOUNT_TABLE.sqlCreate,
CommonConstants.ACCOUNT_TABLE.columns);

    constructor(callback: Function = () => {
    }) {
        this.accountTable.getRdbStore(callback);
    }
}

```



```
}

getRdbStore(callback: Function = () => {
}) {
  this.accountTable.getRdbStore(callback);
}

insertData(account: AccountData, callback: Function) {
  const valueBucket: relationalStore.ValuesBucket =
generateBucket(account);
  this.accountTable.insertData(valueBucket, callback);
}

deleteData(account: AccountData, callback: Function) {
  let predicates = new
relationalStore.RdbPredicates(CommonConstants.ACCOUNT_TABLE.tableName);
  predicates.equalTo('id', account.id);
  this.accountTable.deleteData(predicates, callback);
}

updateData(account: AccountData, callback: Function) {
  const valueBucket: relationalStore.ValuesBucket =
generateBucket(account);
  let predicates = new
relationalStore.RdbPredicates(CommonConstants.ACCOUNT_TABLE.tableName);
  predicates.equalTo('id', account.id);
  this.accountTable.updateData(predicates, valueBucket, callback);
}

query(amount: number, callback: Function, isAll: boolean = true) {
  let predicates = new
relationalStore.RdbPredicates(CommonConstants.ACCOUNT_TABLE.tableName);
  if (!isAll) {
    predicates.equalTo('amount', amount);
  }
  this.accountTable.query(predicates, (resultSet:
relationalStore.ResultSet) => {
    let count: number = resultSet.rowCount;
    if (count === 0 || typeof count === 'string') {
      console.log(`${CommonConstants.TABLE_TAG}` + 'Query no
results!');
      callback([]);
    } else {
      resultSet.goToFirstRow();
    }
  });
}
```

```
const result: AccountData[] = [];
for (let i = 0; i < count; i++) {
  let tmp: AccountData = {
    id: 0,
    accountType: 0,
    typeText: '',
    amount: 0,
    ps: '',
    year: 0,
    month: 0,
    day: 0
  };
  tmp.id = resultSet.getDouble(resultSet.getColumnIndex('id'));
  tmp.accountType =
resultSet.getDouble(resultSet.getColumnIndex('accountType'));
  tmp.typeText =
resultSet.getString(resultSet.getColumnIndex('typeText'));
  tmp.amount =
resultSet.getDouble(resultSet.getColumnIndex('amount'));
  tmp.ps = resultSet.getString(resultSet.getColumnIndex('ps'));
  tmp.year =
resultSet.getDouble(resultSet.getColumnIndex('accountType'));
  tmp.month =
resultSet.getDouble(resultSet.getColumnIndex('typeText'));
  tmp.day =
resultSet.getDouble(resultSet.getColumnIndex('amount'));
  result[i] = tmp;
  resultSet.moveToNextRow();
}
callback(result);
});
}
}

function generateBucket(account: AccountData):
relationalStore.ValuesBucket {
  let obj: relationalStore.ValuesBucket = {};
  obj.accountType = account.accountType;
  obj.typeText = account.typeText;
  obj.amount = account.amount;
  obj.ps = account.ps;
  obj.year = account.year;
  obj.month = account.month;
```

```
obj.day = account.day;
return obj;
}
```

6.2 记账本首页代码

```
import router from '@ohos.router'
import AccountTable from '../common/database/tables/AccountTable';
import AccountData from '../viewmodel/AccountData';
```

@Entry

@Component

```
struct Index {
  private controller: TabsController = new TabsController();
  @State private year: string = '2023';
  @State private month: string = '11';
  @State private day: string = '29';
  @State private getIn: string = '00.00'
  @State private getOut: string = '70.00'
  selectedDate: Date = new Date("2023-11-29")
  @State private year2: string = '2023';
  @State accounts: Array<AccountData> = [];
  @State message: object = router.getParams();
  @State turn: number = -1;
  @State delete: number = 0;
```

```
onPageShow() {
  this.turn += 1;
  if (this.turn == 1) {
    this.getOut = "90.00";
  } else if (this.turn >= 2) {
    this.getOut = "90.00";
    this.getIn = "100.00";
  } else if (this.turn < 0) {
    this.getOut = "40.00";
    this.getIn = "100.00";
  }
}
```

```
build() {
  Column({ space: 50 }) {
    Column() {
      Column({ space: 20 }) {
        Row() {
          Image($r('app.media.title'))
        }
      }
    }
  }
}
```

```

    }
    .height("30%")
    .width("70%")

    Row() {
        Column({ space: 10 }) {
            Text(this.year + "年")
                .fontSize(20)
                .fontColor(Color.Brown)
            Text(this.month + "月" + this.day + "日▼")
                .fontSize(20)
                .onClick(() => {
                    DatePickerDialog.show({
                        start: new Date("2015-1-1"), // 设置选择器的起始日期
                        end: new Date("2024-12-31"), // 设置选择器的结束日期
                        selected: this.selectedDate, // 设置当前选中的日期
                        lunar: false,
                        onAccept: (value: DatePickerResult) => { // 点击弹窗
中的“确定”按钮时触发该回调
                            // 通过 Date 的 setFullYear 方法设置按下确定按钮时的
                            日期，这样当弹窗再次弹出时显示选中的是上一次确定的日期
                            this.selectedDate.setFullYear(value.year,
value.month, value.day)
                            this.year = value.year.toString()
                            this.month = (value.month + 1).toString()
                            this.day = value.day.toString()
                            if (this.day != "29") {
                                this.getIn = "00.00"
                                this.getOut = "00.00"
                            } else {
                                this.getIn = "00.00"
                                this.getOut = "70.00"
                            }
                            console.info("DatePickerDialog:onAccept()" +
JSON.stringify(value))
                        },
                        onCancel: () => { // 点击弹窗中的“取消”按钮时触发
该回调
                            console.info("DatePickerDialog:onCancel()")
                        },
                        onChange: (value: DatePickerResult) => { // 滑动弹窗
中的滑动选择器使当前选中项改变时触发该回调
                            console.info("DatePickerDialog:onChange()" +
JSON.stringify(value))

```

```
        }
      })
    })
  }
  .alignItems(HorizontalAlign.Start)
  .width("45%")

  Column({ space: 10 }) {
    Text("收入")
      .fontColor(Color.Gray)
      .fontSize(20)
      .backgroundColor(Color.White)
      .borderRadius(5)
    Text(this.getIn)
      .fontSize(25)
      .backgroundColor(Color.White)
      .borderRadius(5)
  }
  .alignItems(HorizontalAlign.Start)
  .width("25%")
  .padding(5)
  // .backgroundColor(Color.White)

  Column({ space: 10 }) {
    Text("支出")
      .fontColor(Color.Gray)
      .fontSize(20)
      .backgroundColor(Color.White)
      .borderRadius(5)
    Text(this.getOut)
      .fontSize(25)
      .backgroundColor(Color.White)
      .borderRadius(5)
  }
  .alignItems(HorizontalAlign.Start)
  .width("25%")
  .padding(5)
}
}
.width("100%")
.height("26%")
.padding(20)
.backgroundColor(Color.Orange)
.borderRadius(20)
```

```
Column({ space: 5 }) {
  Row() {
    Column() {
      Image($r('app.media.Jizhang'))
        .width("10%")
      Text("支出")
        .fontSize(15)
    }
    .onClick() => {
      router.pushUrl({
        url: "pages/PageRecord",
        params: {
          id: 0
        }
      })
    })
  })

  Column() {
    Image($r('app.media.ShouRu'))
      .width("11%")
    Text("收入")
      .fontSize(15)
  }
  .onClick() => {
    router.pushUrl({
      url: "pages/PageRecord",
      params: {
        id: 1
      }
    })
  })
})

  Column() {
    Image($r('app.media.MingXi'))
      .width("10%")
    Text("明细")
      .fontSize(15)
  }
  .onClick() => {
    router.pushUrl({
      url: "pages/PageAccount",
      params: {
        id: 0
      }
    })
  })
})
```

```
        }
      })
    })

    Column() {
      Image($r('app.media.RiLi2'))
        .width("10%")
      Text("本月")
        .fontSize(15)
    }
    .onClick(() => {
      router.pushUrl({
        url: "pages/PageMonthCount"
      })
    })

    Column() {
      Image($r('app.media.BaoGao'))
        .width("10%")
      Text("年度")
        .fontSize(15)
    }
    .onClick(() => {
      router.pushUrl({
        url: "pages/PageAccount",
        params: {
          id: 1
        }
      })
    })
  })
}
.width("100%")
.height(80)
.backgroundColor(Color.White)
.borderRadius(40)
.margin(-20)
.borderWidth(1)
.justifyContent(FlexAlign.SpaceEvenly)

Row() {
}
.margin(10)

if (this.day == "29") {
```

```

if (this.delete == 0) {
  Scroll() {
    Row() {
      Image($r('app.media.JuJia'))
        .width(50)
        .backgroundColor(Color.Yellow)
        .borderRadius(20)

      Column() {
        Text("居家")
          .fontSize(22)
          .width(230)
          .textAlign(TextAlign.Center)
      }

      Text("-50")
        .fontSize(18)
        .textAlign(TextAlign.End)
        .width(55)

      Row()
        .width(60)
      Row() {
        Text('删除')
      }
    }
    .onClick(() => {
      AlertDialog.show({
        title: "是否删除?",
        message: "删除这条记录",
        autoCancel: true, // 点击蒙层, 隐藏对话框
        cancel: () => {
        },
        alignment: DialogAlignment.Bottom, // 设置对话框底部对齐
        offset: { dx: 0, dy: -20 }, // 在 Y 轴方向上的偏移量

        confirm: {
          value: "确认删除",
          action: () => {
            this.delete = 1
            this.getOut = "40.00"
            this.turn = -100
          }
        }
      })
    })
  }
}

```

部对齐
量


```
        });
    })
    .width(60)
    .height("100%")
    .backgroundColor("#FFB266")
    .borderRadius(25)
    .justifyContent(FlexAlign.Center)
  }
  .height("10%")
  .padding(15)
  .borderRadius(15)
}
.width(700)
.scrollBar(BarState.Off)
.scrollable(ScrollDirection.Horizontal)

Divider()
}

Row() {
  Image($r('app.media.CanYin'))
    .width("15%")
    .backgroundColor(Color.Yellow)
    .borderRadius(20)
  if (true) {
    Column({ space: 3 }) {
      Text("餐饮")
        .fontSize(20)
      Text("备注：这是一条备注")
        .fontSize(15)
    }
    .width("60%")
  } else {
    Column() {
      Text("餐饮")
        .fontSize(22)
        .width("60%")
        .textAlign(TextAlign.Center)
    }
  }
}
Text("-20")
  .fontSize(18)
  .textAlign(TextAlign.End)
  .width("15%")
```

```
}
.width("100%")
.height("10%")
.borderRadius(15)
.padding(15)
.backgroundColor(Color.White)
.onClick(() => {
  AlertDialog.show({
    title: "是否删除?",
    message: "删除这条记录",
    autoCancel: true, // 点击蒙层, 隐藏对话框
    cancel: () => {
    },
    alignment: DialogAlignment.Bottom, // 设置对话框底部对齐
    offset: { dx: 0, dy: -20 }, // 在 Y 轴方向上的偏移量
    confirm: {
      value: "确认删除",
      action: () => {
      }
    }
  });
})

Divider()

// if (this.message) {
if (this.turn >= 1 || this.turn < 0) {
  Row() {
    Image($r('app.media.CanYin'))
      .width("15%")
      .backgroundColor(Color.Yellow)
      .borderRadius(20)
    if (true) {
      Column({ space: 3 }) {
        Text("餐饮")
          .fontSize(20)
        Text("备注: eating spends 20")
          .fontSize(15)
      }
      .width("60%")
    } else {
      Column() {
        Text("餐饮")
          .fontSize(22)
```

齐

```

        .width("60%")
        .textAlign(TextAlign.Center)
    }
}
Text("-20")
    .fontSize(18)
    .textAlign(TextAlign.End)
    .width("15%")
}
.width("100%")
.height("10%")
.borderRadius(15)
.padding(15)
.backgroundColor(Color.White)
.onClick(() => {
    AlertDialog.show({
        title: "是否删除?",
        message: "删除这条记录",
        autoCancel: true, // 点击蒙层, 隐藏对话框
        cancel: () => {
        },
        alignment: DialogAlignment.Bottom, // 设置对话框底部对

        offset: { dx: 0, dy: -20 }, // 在 Y 轴方向上的偏移量
        confirm: {
            value: "确认删除",
            action: () => {
            }
        }
    });
})

Divider()
}

if (this.turn >= 2 || this.turn < 0) {
    Row() {
        Image($r('app.media.JianZhi'))
            .width("15%")
            .backgroundColor(Color.Pink)
            .borderRadius(20)
        if (true) {
            Column({ space: 3 }) {
                Text("兼职")
            }
        }
    }
}

```

```

        .fontSize(20)
        Text("备注: part time job gets 100")
        .fontSize(15)
    }
    .width("60%")
} else {
    Column() {
        Text("餐饮")
        .fontSize(22)
        .width("60%")
        .textAlign(TextAlign.Center)
    }
}
Text("+100")
    .fontSize(18)
    .textAlign(TextAlign.End)
    .width("15%")
}
.width("100%")
.height("10%")
.borderRadius(15)
.padding(15)
.backgroundColor(Color.White)
.onClick(() => {
    AlertDialog.show({
        title: "是否删除?",
        message: "删除这条记录",
        autoCancel: true, // 点击蒙层, 隐藏对话框
        cancel: () => {
        },
        alignment: DialogAlignment.Bottom, // 设置对话框底部对
        offset: { dx: 0, dy: -20 }, // 在 Y 轴方向上的偏移量
        confirm: {
            value: "确认删除",
            action: () => {
            }
        }
    });
});
})

Divider()
}
} else { // 还没有添加数据

```

齐

```

        }

        }
    }
    .width('100%')
    .height('100%')
    // .backgroundColor("#aabbcc")
    .padding(3)
}
}

// .size({ width: "100%", height: "100%" })
// }
}

```

6.2 支出/输入界面代码

```

import router from '@ohos.router';
import AccountTable from '../common/database/tables/AccountTable';
import AccountData from '../viewmodel/AccountData';

@Entry
@Component
struct PageRecord {
    private controller: TabsController = new TabsController();
    @State message: object = router.getParams()
    @State index: number = 0;
    @State turn: number = 0;
    private gridData: ItemData[] = [
        new ItemData(1, "餐饮", $r("app.media.CanYin")),
        new ItemData(2, "购物", $r('app.media.GouWu')),
        new ItemData(3, "日用", $r("app.media.RiYong")),
        new ItemData(4, "交通", $r('app.media.JiaoTong')),
        new ItemData(5, "蔬菜", $r('app.media.ShuCai')),
        new ItemData(6, "水果", $r('app.media.ShuiGuo')),
        new ItemData(7, "零食", $r('app.media.LingShi')),
        new ItemData(8, "运动", $r('app.media.YunDong')),
        new ItemData(9, "娱乐", $r('app.media.YuLe')),
        new ItemData(10, "通讯", $r('app.media.TongXun')),
        new ItemData(11, "服饰", $r('app.media.FuShi')),
        new ItemData(12, "美容", $r('app.media.MeiRong')),
        new ItemData(13, "住房", $r('app.media.ZhuFang')),
        new ItemData(14, "居家", $r('app.media.JuJia')),
    ]
}

```

```

    new ItemData(15, "孩子", $r('app.media.HaiZi')),
    new ItemData(16, "长辈", $r('app.media.ZhangBei')),
    new ItemData(17, "社交", $r('app.media.SheJiao')),
    new ItemData(18, "烟酒", $r('app.media.YanJiu')),
    new ItemData(19, "旅行", $r('app.media.LvXing')),
    new ItemData(20, "旅行", $r('app.media.ShuMa'))
  ];
  private gridData2: ItemData2[] = [
    new ItemData2(1 + 20, "工资", $r("app.media.GongZi")),
    new ItemData2(2 + 20, "兼职", $r('app.media.JianZhi')),
    new ItemData2(3 + 20, "理财", $r("app.media.LiCai")),
    new ItemData2(4 + 20, "礼金", $r('app.media.LiJin')),
    new ItemData2(5 + 20, "其他", $r('app.media.QiTa')),
  ];
  customDialogController: CustomDialogController = new CustomDialogController({
    builder: CustomDialogWidget(),
    alignment: DialogAlignment.Bottom,
    customStyle: true,
    offset: { dx: 0, dy: -20 },
    cancel: () => {
      this.index = 0;
    }
  });

  build() {
    Row() {
      Column() {
        Tabs({
          barPosition: BarPosition.Start,
          controller: this.controller,
          index: this.message['id']
        }) {
          TabContent() {
            Column() {
              Grid() {
                ForEach(this.gridData, (item: ItemData) => {
                  GridItem() {
                    Column({ space: 8 }) {
                      Image(item.img)
                        .width("60%")
                        .backgroundColor(this.index == item.index ? '#E3CF57' :
                          '#dcdcdc')
                      .borderRadius(20)
                      .borderWidth(1)

```

```

        Text(item.title)
    }
    .onClick(() => {
        this.index = item.index
        this.customDialogController.open();
    })
}
}, item => JSON.stringify(item))
}
.columnsTemplate('1fr 1fr 1fr 1fr')
.rowsTemplate('1fr 1fr 1fr 1fr 1fr 1fr 1fr')
.columnsGap(8)
.rowsGap(12)
}
.width('100%')
.height('100%')
.backgroundColor("#CCFFE5")
.borderRadius(20)
}
.size({ width: "100%", height: "100%" })
.tabBar("支出")

TabContent() {
    Column() {
        Grid() {
            ForEach(this.gridData2, (item: ItemData2) => {
                GridItem() {
                    Column({ space: 8 }) {
                        Image(item.img)
                            .width("60%")
                            .borderWidth(1)
                            .borderRadius(20)
                            .backgroundColor(this.index == item.index ? '#E3CF57' :
' #dcdcdc')

                        Text(item.title)
                    }
                }
                .onClick(() => {
                    this.index = item.index;
                    this.customDialogController.open();
                })
            })
        }, item => JSON.stringify(item))
    }
    .columnsTemplate('1fr 1fr 1fr 1fr')

```

```
        .rowTemplate('1fr 1fr 1fr 1fr 1fr 1fr 1fr')
        .columnsGap(8)
        .rowsGap(12)
    }
    .width('100%')
    .height('100%')
    .borderRadius(20)
    .backgroundColor(Color.Pink)
    }
    .size({ width: "100%", height: "100%" })
    .tabBar("收入")
    }
    .barWidth("50%")
    }
    .width('100%')
    .padding(10)

    Column() {
        Text("取消")
            .fontSize(15)
            .fontColor(Color.Gray)
    }
    .onClick() => {
        router.back()
        // ({
        //   url: 'pages/Index'
        // })
    })
    .position({ x: 320, y: 30 })
    }
    }
}

class ItemData {
    index: number;
    title: string;
    img: Resource;

    constructor(index: number, title: string, img: Resource) {
        this.index = index;
        this.title = title;
        this.img = img;
    }
}
```



```
class ItemData2 {
    index: number;
    title: string;
    img: Resource;

    constructor(index: number, title: string, img: Resource) {
        this.index = index;
        this.title = title;
        this.img = img;
    }
}

@CustomDialog
struct CustomDialogWidget {
    private controller: CustomDialogController;
    @State turn: number = 0;
    @State private gridData: ItemData[] = [
        new InItemData(1, "7"),
        new InItemData(2, "8"),
        new InItemData(3, "9"),
        new InItemData(4, "今天", $r("app.media.RiLi")),
        new InItemData(5, "4"),
        new InItemData(6, "5"),
        new InItemData(7, "6"),
        new InItemData(8, "清空"),
        new InItemData(9, "1"),
        new InItemData(10, "2"),
        new InItemData(11, "3"),
        new InItemData(12, "取消"),
        new InItemData(13, "."),
        new InItemData(14, "0"),
        new InItemData(15, "x", $r("app.media.delete")),
        new InItemData(16, "完成"),
    ];
    selectedDate: Date = new Date("2023-11-29")
    @State private today: boolean = true;
    @State private val: string = "0";
    // @State accounts: Array<AccountData> = [];

    build() {
        Column({ space: 20 }) {
            Text(this.val)
                .fontSize(30)
        }
    }
}
```

```
.width("90%")
.textAlign(TextAlign.End)
TextInput({
  placeholder: "请输入备注"
})
.fontSize(20)

Grid() {
  ForEach(this.gridData, (item: InItemData) => {
    GridItem() {
      Row({ space: 3 }) {
        if (item.index == 4) {
          if (this.today) {
            Image(item.img)
              .width(25)
              .height(25)
            Text(item.title)
              .fontSize(16)
              .fontColor(Color.Black)
          } else {
            Text("11-28")
              .fontSize(15)
              .fontColor(Color.Black)
          }
        } else if (item.index == 15) {
          Image(item.img)
            .width(30)
            .height(30)
        } else {
          Text(item.title)
            .fontSize(16)
            .fontColor(Color.Black)
            .textAlign(TextAlign.Center)
        }
      }
    }
  })
  .border({
    color: Color.Gray,
    width: 2,
    radius: 0,
    style: BorderStyle.Solid
  })
  .width('100%')
  .height('100%')
```

```
.onClick() => {
  if (item.index == 4) {
    DatePickerDialog.show({
      start: new Date("2000-1-1"), // 设置选择器的起始日期
      end: new Date("2023-12-31"), // 设置选择器的结束日期
      selected: this.selectedDate, // 设置当前选中的日期
      lunar: false,
      onAccept: (value: DatePickerResult) => { // 点击弹窗中的“确定”
        按钮时触发该回调
        // 通过 Date 的 setFullYear 方法设置按下确定按钮时的日期，这样
        当弹窗再次弹出时显示选中的是上一次确定的日期
        this.selectedDate.setFullYear(value.year, value.month,
        value.day)
        console.info("DatePickerDialog:onAccept()" +
        JSON.stringify(value))
        this.today = !this.today
      },
      onCancel: () => { // 点击弹窗中的“取消”按钮时触发该回调
        console.info("DatePickerDialog:onCancel()")
      },
      onChange: (value: DatePickerResult) => { // 滑动弹窗中的滑动选
        择器使当前选中项改变时触发该回调
        console.info("DatePickerDialog:onChange()" +
        JSON.stringify(value))
      }
    })
  }
  else if (item.index == 8) {
    this.val = "0";
  }
  else if (item.index == 12) {
    router.replaceUrl({
      url: 'pages/Index'
    })
  }
  else if (item.index == 15) {
    this.val = this.val.slice(0, -1)
    if (this.val.length == 0) {
      this.val = "0";
    }
  }
  else if (item.index == 13) {
    this.val = this.val + item.title;
  }
  else if (item.index == 16) { // 存储用户记录
    router.back({
```

```
        url: 'pages/Index'
    })
}
else {
    if (this.val == "0") {
        this.val = "";
    }
    this.val = this.val + item.title;
}
}))

    }, item => JSON.stringify(item))
}
.columnsTemplate('1fr 1fr 1fr 1fr')
.rowsTemplate('1fr 1fr 1fr 1fr')
.columnsGap(5)
.rowsGap(5)
.height(250)
}
.backgroundColor(Color.White)
.padding(10)
.height(400)
.width("90%")
.borderRadius(20)
}
}

class InItemData {
    index: number;
    title: string;
    img: Resource = null;

    constructor(index: number, title: string, img?: Resource) {
        this.index = index;
        this.title = title;
        this.img = img;
    }
}

function newAccount(newAccount: any, arg1: (id: number) => void) {
    throw new Error('Function not implemented.');
```

6.3 月账单/年账单界面代码

```
import router from '@ohos.router'

@Entry
@Component
struct PageAccount {
  @State year: string = '2023'
  @State message: object = router.getParams()
  private controller: TabsController = new TabsController();
  years: string[] = ['2022', '2023', '2024', '2015', '2016', '2017', '2018',
    '2019', '2020', '2021']
  @State private month_list: MonthCount[] = [
    new MonthCount(1, 4000.00, 2800.5),
    new MonthCount(2, 3900, 2750),
    new MonthCount(3, 3950, 2600),
    new MonthCount(4, 4200, 3100),
    new MonthCount(5, 3880, 2550),
    new MonthCount(6, 4330, 2210),
    new MonthCount(7, 5050, 4000),
    new MonthCount(8, 4500, 2100),
    new MonthCount(9, 4000, 2900),
    new MonthCount(10, 4100, 2850),
    new MonthCount(11, 3000, 1600),
    new MonthCount(12, 0, 0),
  ]
  @State Nianjieyu: string = "12449.5";
  @State Nianshouru: string = "44910.00";
  @State Nianzhichu: string = "29460.5";

  build() {
    Column() {
      Tabs({ // Tabs 不设置 vertical 属性时，默认上下排版
        barPosition: BarPosition.Start,
        controller: this.controller,
        index: this.message['id']
      }) {
        TabContent() {
          Column() {
            Row() {
              Column({ space: 10 }) {
                Row() {
                  Text(this.year + "年▼")
                    .fontSize(15)
                }
              }
              .width("80%")
            }
          }
        }
      }
    }
  }
}
```

```
.justifyContent(FlexAlign.Start)
.onClick(() => {
    TextPickerDialog.show({
        range: this.years,
        selected: 13,
        onAccept: (value: TextPickerResult) => {
            this.year = value.value;
        }
    })
})

Row({ space: 20 }) {
    Text("年结余")
        .fontSize(25)
    Text(this.Nianjieyu)
        .fontSize(35)
}
.width("100%")
.justifyContent(FlexAlign.Center)

Row() {
    Row({ space: 10 }) {
        Text("年收入")
        Text(this.Nianshouru)
            .fontSize(20)
    }
    .width("50%")
    .justifyContent(FlexAlign.Center)

    Row({ space: 10 }) {
        Text("年支出")
        Text(this.Nianzhichu)
            .fontSize(20)
    }
    .width("50%")
    .justifyContent(FlexAlign.Center)
}
.width("100%")
}

}
.borderRadius(15)
.backgroundColor(Color.Orange)
.width("100%")
.height("20%")
```

```
Column({ space: 10 }) {  
  Row() {  
    Text("月份")  
      .fontSize(16)  
      .textAlign(TextAlign.Center)  
      .width("22%")  
    Text("月收入")  
      .textAlign(TextAlign.Center)  
      .width("22%")  
      .fontSize(16)  
    Text("月支出")  
      .textAlign(TextAlign.Center)  
      .width("22%")  
      .fontSize(16)  
    Text("月结余")  
      .textAlign(TextAlign.Center)  
      .width("22%")  
      .fontSize(16)  
    Text("明细")  
      .textAlign(TextAlign.Center)  
      .width("12%")  
      .fontSize(16)  
  }  
  .width("100%")  
  .margin(10)  
  .justifyContent(FlexAlign.SpaceAround)  
  
  ForEach(this.month_list, (item: MonthCount) => {  
    Row() {  
      Text(item.index.toString())  
        .width("20%")  
        .textAlign(TextAlign.Center)  
      Text(item.In.toString())  
        .textAlign(TextAlign.Center)  
        .width("20%")  
      Text(item.Out.toString())  
        .textAlign(TextAlign.Center)  
        .width("20%")  
      Text(item.leave.toString())  
        .textAlign(TextAlign.Center)  
        .width("20%")  
      Image($r("app.media.JianTou"))  
    }  
  })  
}
```

```
        .width(20)
      }
      .width("100%")
      .borderRadius(15)
      .height(32)
      .justifyContent(FlexAlign.SpaceAround)
      .backgroundColor(Color.White)
      .onClick(() => {
        router.pushUrl({
          url: 'pages/PageMonthCount',
          params: {
            name: item.index.toString(),
            value: this.year.toString()
          }
        })
      })
    }, item => JSON.stringify(item))

  }
  // .backgroundColor(Color.Red)
  .width("100%")
  .height("80%")

}
.width('100%')
.height('100%')
.backgroundColor("#bbccaa")
.padding(8)
}
.size({ width: "100%", height: "100%" })
.tabBar("月账单")

TabContent() {
  Scroll() {
    Column() {
      Column() {
        Row() {
          Column({ space: 10 }) {
            Row({ space: 20 }) {
              Text("总结余")
                .fontSize(25)
              Text("12449.5")
                .fontSize(35)
            }
          }
        }
      }
    }
  }
}
```



```
    }
    .width("100%")
    .justifyContent(FlexAlign.Center)

    Row() {
      Row({ space: 10 }) {
        Text("总收入")
        Text("44910.00")
        .fontSize(20)
      }
      .width("50%")
      .justifyContent(FlexAlign.Center)

      Row({ space: 10 }) {
        Text("总支出")
        Text("29460.50")
        .fontSize(20)
      }
      .width("50%")
      .justifyContent(FlexAlign.Center)
    }
    .width("100%")
  }
}
.backgroundColor("#FFA07A")
.borderRadius(15)
.width("100%")
.height(150)

Row()
  .height(20)

Column({ space: 15 }) {
  Column() {
    Row() {
      Text("总支出趋势图")
      .fontSize(20)
      .fontWeight(FontWeight.Bold)
    }
    .width("100%")
    .justifyContent(FlexAlign.Start)
    .padding(15)

    Row() {
```

```
        Image($r('app.media.Line2'))
        // .width("90%")
        .height(190)
    }
}
.width("100%")
.height("280")
.borderRadius(15)
.backgroundColor(Color.White)

Column() {
    Row() {
        Text("总收入趋势图")
        .fontSize(20)
        .fontWeight(FontWeight.Bold)
    }
    .width("100%")
    .justifyContent(FlexAlign.Start)
    .padding(15)

    Row() {
        Image($r('app.media.Line1'))
        // .width("90%")
        .height(190)
    }
}
.width("100%")
.height("280")
.borderRadius(15)
.backgroundColor(Color.White)
}
}
.width('100%')
.height('100%')
.backgroundColor("#E6E6FA")
.padding(8)
}
.width("100%")
.height("850")
.scrollBar(BarState.Off)

}

.size({ width: "100%", height: "100%" })
```

```
        .tabBar("年账单")
    }
    .size({ width: "100%", height: "100%" })
    .barWidth("50%")

    Column() {
        Text("取消")
            .fontSize(15)
            .fontColor(Color.Gray)
    }
    .onClick(() => {
        router.back()
        // ({
        //     url: 'pages/Index'
        // })
    })
    .position({ x: 320, y: 20 })

}
.width('100%')
.height('100%')
.padding(10)

}
}

class MonthCount {
    index: number;
    In: number;
    Out: number;
    leave: number;

    constructor(index: number, In: number, Out: number) {
        this.index = index;
        this.In = In;
        this.Out = Out;
        this.leave = In - Out;
    }
}
```

6.4 月账单明细界面代码

```
import router from '@ohos.router'
import media from '@ohos.multimedia.media';
@Entry
```

```
@Component
struct MonthCount {
  @State message: object = router.getParams()
  @State Yuejieyu: string = "1400.00";
  @State Shangyuejieyu: string = "11049.5";
  @State Zhichu: string = "1600.00"
  @State Shouru: string = "3000.00"
  @State Tianshu: string = "25"
  @State Bishu: string = "45"
  build() {
    Scroll() {
      Column() {
        Row({ space: 60 }) {
          Image($r('app.media.FanHui'))
            .width(30)
            .onClick() => {
              router.back()
            })
          Text("2023" + "年" + "11" + "月账单")
            .fontSize(25)
            .fontWeight(FontWeight.Bold)
        }
        .width("100%")
        .height("115")
        .backgroundColor(Color.Orange)
        .justifyContent(FlexAlign.Center)
        Column({ space: 10 }) {
          Row() {
            Column({ space: 15 }) {
              Text("本月结余")
                .fontSize(20)
                .fontColor(Color.Gray)
              Text(this.Yuejieyu)
                .fontSize(20)
                .fontWeight(FontWeight.Bold)
            }
            .width("50%")
            Column({ space: 15 }) {
              Text("上月结余")
                .fontColor(Color.Gray)
                .fontSize(20)
              Text(this.Shangyuejieyu)
                .fontSize(20)
            }
          }
        }
      }
    }
  }
}
```

```
.width("50%")
}
.width("100%")
.height("115")
.borderRadius(15)
.backgroundColor(Color.White)
Row() {
  Column({ space: 15 }) {
    Text("支出")
      .fontColor(Color.Gray)
      .fontSize(20)
    Text(this.Zhichu)
      .fontSize(20)
  }
  .width("25%")

  Column({ space: 15 }) {
    Text("收入")
      .fontColor(Color.Gray)
      .fontSize(20)
    Text(this.Shouru)
      .fontSize(20)
  }
  .width("25%")

  Column({ space: 15 }) {
    Text("记账天数")
      .fontColor(Color.Gray)
      .fontSize(20)
    Text(this.Tianshu)
      .fontSize(20)
  }
  .width("25%")

  Column({ space: 15 }) {
    Text("记账笔数")
      .fontColor(Color.Gray)
      .fontSize(20)
    Text(this.Bishu)
      .fontSize(20)
  }
  .width("25%")
}
.width("100%")
```

```
.height("115")
.borderRadius(15)
.backgroundColor(Color.White)

Column() {
  Row() {
    Column() {
      Text("支出类别")
      .fontSize(20)
      .fontWeight(FontWeight.Bold)
    }
  }
  .width("100%")
  .justifyContent(FlexAlign.Start)
  .padding(15)

  Row() {
    Image($r('app.media.Piel'))
    // .width("100%")
    .height(290)
  }
  .width("100%")
}
.width("100%")
.height("380")
.borderRadius(15)
.backgroundColor(Color.White)

Column() {
  Row() {
    Text("月支出对比")
    .fontSize(20)
    .fontWeight(FontWeight.Bold)
  }
  .width("100%")
  .justifyContent(FlexAlign.Start)
  .padding(15)

  Row() {
    Image($r('app.media.Column1'))
    // .width("90%")
    .height(250)
  }
}
```

```
}  
.width("100%")  
.height("320")  
.borderRadius(15)  
.backgroundColor(Color.White)  
  
Column({ space: 5 }) {  
  Row() {  
    Text("收入")  
      .fontSize(20)  
      .fontWeight(FontWeight.Bold)  
  }  
  .width("100%")  
  .justifyContent(FlexAlign.Start)  
  .padding(15)  
  
  Column() {  
    Text("总收入")  
      .fontColor(Color.Gray)  
    Text(this.Shouru)  
      .fontSize(23)  
      .fontWeight(FontWeight.Bold)  
    Row() {  
      Image($r('app.media.Pie2'))  
        // .width("100%")  
        .height(230)  
    }  
    .width("100%")  
    Row()  
      .height(20)  
  
    Text("月收入对比")  
      .fontColor(Color.Gray)  
    Row()  
      .height(10)  
    Row() {  
      Image($r('app.media.Column2'))  
        // .width("90%")  
        .height(200)  
    }  
  }  
}  
  
.alignItems(HorizontalAlign.Start)  
.width("90%")
```

```
    }  
    .width("100%")  
    .height("620")  
    .borderRadius(15)  
    .backgroundColor(Color.White)  
  }  
  .height("1730")  
  .padding(8)  
  .backgroundColor("#aabbcc")  
}  
.width('100%')  
// .height("100%")  
}  
.width("100%")  
.height("1500")  
.scrollBar(BarState.Off)  
}  
}
```

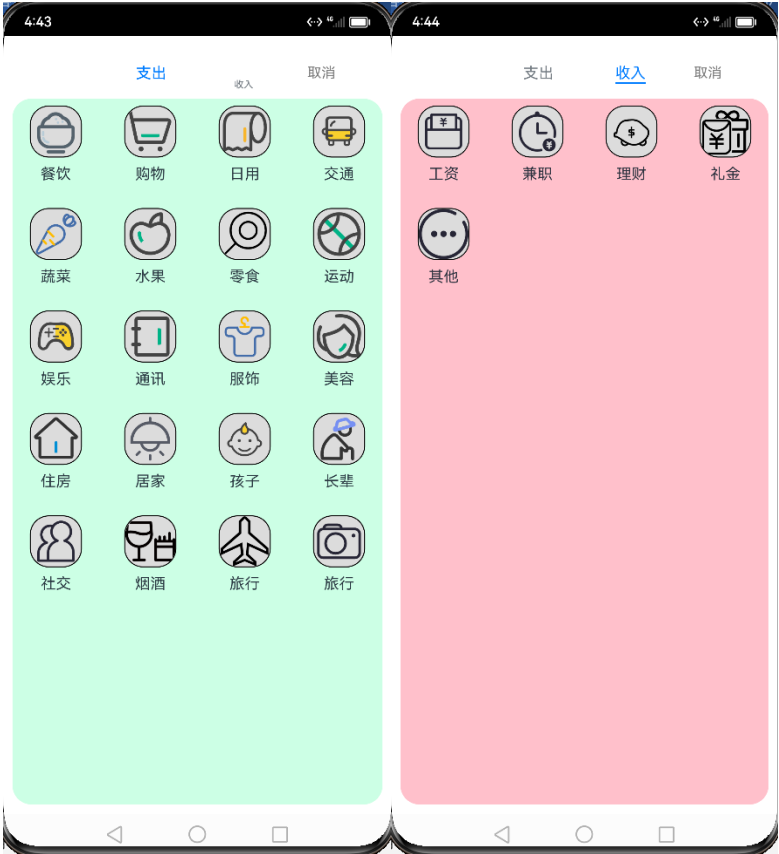
7. 运行截图

7.1 记账本首页





7.2 收入/支出界面





7.3 月账单/年账单界面



7.3 月账单明细



四、实验结果及分析和（或）源程序调试过程（界面截图和文字）、实验总结与体会

1. 实验亮点

- （1）能够根据日期、月份和年份查询具体账单明细
- （2）月账单和年账单统计包括条形图、饼图和折线图
- （3）提供多种支出和收入类型选择，提供自定义弹窗用于输入金额、备注和日期选择功能
- （4）整体页面布局美观，使用很多的图标和文字搭配

2. 实验结果分析

本次实验使用 DecEco Studio 结合 arkUI 语言进行收支记账 APP 的开发。分别实现了分类记录日常收入和支出数据，包括金额、用途、日期、分类等信息；能够增加、删除等，可以支持侧滑删除；显示月、日收支流水记录，显示当月、当日收支总额；按周、月、年进行汇总形成结果；具备友好的用户数据输入界面和查询界面的功能，并额外实现了能够根据日期查询账单；界面美观漂亮；使用了不同的统计图表的功能。完成预期的实验目标。

3. 实验总结与体会

这项实验是基于 HarmonyOS 平台，使用“声明式开发范式”和 ArkTS 语言来开发一个日常收支记账 app。首页中通过 Row-Column 布局，首先定义了表头的“记账本”标题名称，当日收入和当日支出的文本框，其中通过@State 方式绑定了当日的收入和支出金额，通过 if-else 的动态渲染方式，一旦发生金额变化，就会同步发生改

变。具体的，首先在生命周期 `aboutToAppear` 函数中初始化数据库 `AccountTable` 表，并通过 `query` 函数查询数据库具体的每一笔收支信息，再将对应的值渲染到程序上。同时使用 `DatePickerDialog` 配合文本框的点击事件，实现了更换日期的功能。下方的功能选项布局通过调整 `margin` 和 `borderRadius` 实现了悬浮在标题下方的效果，其中通过 `SpaceEvenly` 布局使每个功能图标布局更加美观，不同的图标绑定了对应的 `router` 路由，跳转到不同页面。下方当日记账的明细，通过 `ForEach` 函数，为数据库中取得的每个 `AccountData` 项动态渲染在页面上，并通过 `if-else` 结合 `@State` 类型的变量，实现了有无备注的明细不同的显示方式。每一个明细通过 `Scroll` 和 `scrollable` (`Scroll Direction.Horizontal`)相结合的方式，实现了右滑删除的功能。点击删除按钮会弹出提示框提醒是否删除，点击删除后对应的判断变量会置否，从而达到不显示在页面上的效果。

通过 `TabsController` 和 `Tabs` 组件，让菜单栏置于页面顶端，滑动菜单栏或者页面，可以实现页面翻转。页面中每一个支出（收入）类型的标签首先封装在 `ItemData` 类中，再使用 `gridData` 存放所有标签的 `id`、图片和文字信息，使用 `grid` 布局，将标签按照每行 4 个的方式排列在页面中。点击对应的标签，通过 `onClick` 绑定点击事件，标签的背景颜色会变成黄色，同时弹出自定义弹窗，输入对应的金额和选择日期。自定义弹窗用 `CustomDialog` 类进行复写，通过 `grid` 布局实现 4*4 的输入框布局，通过 `if-else` 绑定不同按钮的点击事件，实现删除、清空和选择日期的功能。若点击页面蒙层，使用回调函数将选择的标签背景变回原来的颜色。若输入正确的金额并点击确定后，填写的参数会通过 `@Link` 类型的变量与父组件交互，同时通过 `insert` 方式存储到数据库中，页面跳转到首页并将记账明细重新渲染在页面上。

在月账单跳转到月明细页面时，`router` 中 `params` 参数传递对应的月份以及支出、收入、结余、记账天数和记账笔数信息，在明细页面中通过 `router.getParams` 函数接收对应的参数，渲染在明细页面中对应的位置。在通过网页加载的方式，传递月份的收入和支出排名前五的明细内容到 `Web` 组件中，`Web` 组件通过调用统计图表 `api`，将返回的绘制的饼图、条形图和折线图通过调整 `Web` 组件的比例渲染在页面上。整个页面通过 `Scroll` 结合页面高度的调整，实现滑动效果。

在 `app` 开发的过程中，我也遇到了一些困难，比如如何有效地组织和展示大量的数据，以及如何设计一个既美观又实用的用户界面。通过不断尝试和优化，我最终实现了一个功能齐全、用户友好的记账 APP。总体来说，这次实验不仅提高了我在使用 `HarmonyOS` 平台和 `ArkTS` 语言开发 APP 的技能，也增强了我在数据库管理和界面设计方面的能力。通过这次实践，我更深刻地理解了 APP 开发的复杂性和细节要求，同时也激发了我对移动应用开发的兴趣。