

学号： 202210310142

# 上海海事大学

## 课 程 设 计 论 文



论文题目： 基于微信小程序的待做事项管理的设计与开发

学 院： 信息工程学院

专 业： 计算机科学与技术

班 级： 223

姓 名： 欧阳卓

指导老师： 刘昱昊

完成日期： 2024.11.27

## 摘 要

市面上的个人待做事项管理大多功能冗余,采用买断制或订阅制向我们收取费用,在这种场景下,想要拥有一款真正绿色免费的待做事项管理软件是比较困难的,我想做的就是设计出一款满足个人使用需求的,拥有独特交互体验的个人事项管理软件。

本文就是要利用已有的微信小程序开发框架和微信小程序开发平台,开发出一款能免费的个人事项管理小程序,主要完成了如下工作:

- (1) 实现个人待做事项的管理,例如增加,删除,查看
- (2) 以滑动操作为主,完成各种交互逻辑的实现
- (3) 对用户的个人事项完成情况进行记录,以日历的方式呈现

**关键词:** 微信小程序, 个人待做事项, 滑动交互

## Abstract

Most personal task management applications on the market tend to have redundant features and charge users through either a one-time purchase or a subscription model. In this context, finding a truly free and user-friendly task management application is challenging. This project aims to design a personal task management application that meets individual needs while offering a unique interaction experience.

This paper focuses on developing a free personal task management mini-program using the existing WeChat Mini Program development framework and platform. The main objectives achieved are as follows:

- (1) Implementation of personal task management features, such as adding, deleting, and viewing tasks.
- (2) Utilizing swipe gestures as the primary interaction method to achieve various functionalities.
- (3) Recording users' task completion statuses and presenting the data in a calendar view.

**Keywords:** WeChat Mini Program, personal task management, swipe interaction

目 录

1. 绪论 .....	2
1.1 研究背景 .....	2
1.2 要解决的问题 .....	2
2.概要设计 .....	3
2.1 系统框架 .....	3
2.2 模块划分 .....	4
3.关键技术 .....	5
3.1 待做事件类 .....	5
3.2 存储和加载本地数据 .....	5
3.3 重置数据 .....	6
3.4 滑动触发事件逻辑 .....	6
3.5 添加、删除、恢复待做事项的逻辑 .....	7
3.6 查看历史完成记录的逻辑 .....	7
4.系统运行结果 .....	8
4.1 历史完成情况查看界面 .....	8
4.2 操作选择界面 .....	8
4.3 新增待做事项界面 .....	9
4.4 查看待做事件完成情况界面 .....	9
4.5 删除和恢复事件界面 .....	9
主要参考文献 .....	10

# 1. 绪论

随着手机的流行，人们的生活重心也转移到了手机上，他不仅仅是一项娱乐项目，更是一种工具。而且现代人生活节奏快，待办事项管理（To-Do List）是提高个人时间管理效率的重要工具。微信小程序无需下载即可使用，用户体验轻量、便捷，非常适合作为待办事项管理工具的载体。

## 1.1 研究背景

基于微信小程序开发的个人待办事项管理如果开发成功，有以下几个好处：

- （1）基于微信的广泛使用，基于微信小程序的个人事项管理可以让人触手可及，降低了使用成本
- （2）将数据存储在手机中，可以随身携带，保护了数据的隐私性
- （3）借助于编程软件的实现，可以轻松对待办事项进行处理
- （4）软件实现可以轻松对待办事项的完成情况进行统计

所以，本系统旨在开发一个基于微信小程序开发的个人待办事项管理软件。

## 1.2 要解决的问题

本系统要提供以下几个功能：

- （1）友好的界面 ui：提供色彩丰富的 ui 界面，增加用户的体验感
- （2）独特的交互逻辑：使用滑动作为小程序的主要操作，与大部分软件的点击操作不同，可以提高用户的操作体验
- （3）个人待办事项的管理：实现个人待办事项的增加，删除和查看等操作
- （4）历史完成情况的记录，统计和可视化查看：记录每日待办事项的完成情况，在后台进行数据处理，随后以日历的形式展示出来
- （5）实现本地数据存储：将数据存在本地，避免退出小程序后个人待办事项消失

## 2.概要设计

### 2.1 系统框架

下面是使用流程图

- (1) 用户通过滑动以选择不同的操作
- (2) 如果是增加操作，则会弹出输入框，提示用户输入新的待做事项，随后后台将其添加到待做事项数组中，并由微信小程序框架渲染到前端页面中
- (3) 如果是删除操作，在用户选定的待做事项上实施删除操作后，该待做事项会被删除，从待做事项数据中删除，并由微信小程序框架渲染到前端页面中
- (4) 如果是查看操作，后台会查询历史记录数组，其中包含了每个待做事项的自添加以来的日完成情况，随后进行统计，前台渲染一个日历，并对完成的日作出特殊渲染，以告知用户该事项是否完成

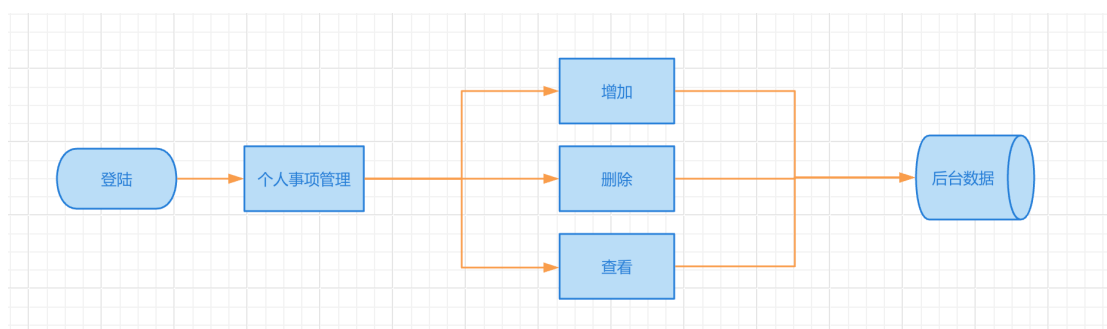


图 2.1 系统框架图

## 2.2 模块划分

针对上述框架，本系统包括一下五个模块：

- （1） 登陆管理模块：本模块主要负责用户的登陆，涉及获取用户头像和昵称，可用于小程序的个人信息展示；涉及查询用户是否首次登陆，若是，则对用户进行引导，否则，将加载用户的本地存储数据，为稍后的操作提供数据。
- （2） 操作管理模块：本模块主要实现对用户的操作进行逻辑上的响应，调用不同操作所对应入口函数，以实现软件上的逻辑交互。
- （3） 增加待做事项模块：本模块主要负责个人待做事项的添加，涉及获取用户的输入，对用户的输入进行判断，将合理的用户输入生成新的待做事项对象并添加到后台数据库中，并通知前台对新的数据进行渲染，实现视觉上的添加。
- （4） 删除待做事项模块：本模块主要涉及个人待做事项的删除，主要目标是将用户选中的待做事项进行删除，设计后台数据的删除和前台界面的渲染。
- （5） 查看待做事项模块：本模块主要涉及对待做事项历史完成情况的统计，涉及数据的处理（通过后台历史记录查看日完成情况），结果的展示。

## 3.关键技术

### 3.1 待做事件类

1. status 表示当前待做事项的状态，存在以下值：2（今日完成打卡），1（还需打卡），0（删除）
2. 在该类的构造函数中，对其 done 历史记录数组进行初始化时，通过 new Date().getDate()-1 获得今日在 done 数组中的下标值，以其为分界线赋初值 0 或 1

```
class todo{
  status = 1;
  scrollLeft = 0;
  constructor(text,id1){
    this.text = text;
    todo.count++;
    this.id = id1;
    this.done = new Array(31).fill(1)
    let today = new Date().getDate()-1
    console.log(today)
    for(let i=0;i<today;i++){
      this.done[i] = 0
    }
  }
}
```

### 3.2 存储和加载本地数据

在小程序初始化时通过 onLaunch()加载本地数据， onHide()时存储小程序中的数据

```
App({
  onLaunch() {
    ...
    var userInfo = wx.getStorageSync('userInfo')
    if(userInfo){
      this.globalData.userInfo = userInfo
      this.globalData.allTodo = wx.getStorageSync('todos')
    }else {...}
    ...
  },
  onHide(){
    ...
    wx.setStorageSync('todos', this.globalData.allTodo)
    wx.setStorageSync('lastDay', new Date().toDateString())
    wx.setStorageSync('userInfo', this.globalData.userInfo)
```



```
...
})
```

### 3.3 重置数据

待做事项中的状态应当在新的一天重置（复位为待做），这里会获取存在本地的“上次登录属性”，与现在的日期对比，判断是否为同一天，是则便利进行重置，否则不进行操作

```
App({
  onShow(){
    const lastDay = wx.getStorageSync('lastDay')
    if(lastDay){
      const currentDate = new Date().toDateString()
      if(currentDate!=lastDay){
        for(var item of this.globalData.allTodo){
          if(item.status==2){
            item.status=1
          }
        }
      }
    }
  }
})
```

### 3.4 滑动触发事件逻辑

在 page 对象中定义数据，记录手指按下时的 x，y 轴，函数中记录松开时的 x，y 轴，计算其中的差值做处理，如果差值满足某一条件，则调用目标函数

```
Page({
  data:{
    startX:0,
    startY:0,
  },

  onTouchStart(e){
    this.setData({
      startX:e.touches[0].pageX,
      startY:e.touches[0].pageY
    })
  },

  onTouchEnd(e){
    let moveX = e.changedTouches[0].pageX
```

```

        let moveY = e.changedTouches[0].pageY
        let delatX = moveX - this.data.startX
        let delatY = moveY - this.data.startY
        if(...) {...}
    })

```

### 3.5 添加、删除、恢复待做事项的逻辑

1. 添加：调用 `wx.showModel()` 函数显示输出框，输入完成确认后点击确认后，后台将获取输入的数据并生成对象插入数组

```
currentTodos.push(new todo(res.content, getApp().globalData.allTodo.length))
```

2. 删除：遍历待做事项数组，查找目标事件将其 `status` 状态设置为 0

```

for(var i=0;i<newTodos.length;i++){
    if (newTodos[i].id == id){
        newTodos[i].status = 0
    }
}

```

3. 恢复：遍历待做事项数组，查找目标事件将其 `status` 状态设置为设置为 1

```

for(var i=0;i<this.data.todos.length;i++){
    if(this.data.todos[i].id==id){
        currentTods[i].status=1
    }
}

```

### 3.6 查看历史完成记录的逻辑

1. javascript 逻辑页面中实现数据的初始化

2. wxml 中使用微信框架的 “`wx:if`”，“`wx:for`” 对目标数据进行渲染

```

<view id="itemall" wx:for="{{targets}}" wx:key="id" wx:if="{{item.status!=0}}">
    <text class="item1" wx:if="{{item.status==1}}">{{item.text}}</text>
    <text class="item2" wx:if="{{item.status==2}}">{{item.text}}</text>
</view>

```

## 4.系统运行结果

### 4.1 历史完成情况查看界面

图 5.1 是用户历史完成情况查看界面的展示，用户可点击图 5.1 中导航栏中的“ME”图标跳转至该界面；选中想查看的日期，随后会在下面更新以展示目标日期的完成情况展示



图 5.1 历史完成情况查看界面



图 5.2 操作选择界面

### 4.2 操作选择界面

图 5.2 是用户选择操作界面的截图

1. 用户可通过向左滑动个人待做事项块，选中该待做事项，随后会展示对对其实现的操作，例如恢复，查看和删除；
2. 用户可通过向右滑动实现所选事件块的完成；
3. 如需添加新的个人待做事项，在最下方“>>>添加新的 TODO 事件”处滑动即可；

### 4.3 新增待做事项界面

图 5.3 是新增待做事项界面的截图

- 1. 用户滑动待做事项最下方文字“>>>添加新的 TODO 事件”
- 2. 可在输入框中输入待做事项的内容，后台自动生成该对象并加入数据库中



图 5.3 新增待做事项界面



图 5.4 查看待做事件完成情况界面

### 4.4 查看待做事件完成情况界面

图 5.4 是查看待做事件的完成情况界面

- 1. 在对应的待做事项上向左滑动即可出现功能按键，点击查看后可展示该事件的月完成情况
- 2. 其会在完成时间节点用绿色背景标记

### 4.5 删除和恢复事件界面

图 5.5 和图 5.6 分别是删除和恢复时的界面展示，它们的操作逻辑类似： 在对应的待做事项上向左滑动即可出现功能按键，点击查看后可对其进行恢复或删除操作



图 5.5 恢复事件界面



图 5.5 删除事件界面

## 5. 结论和展望

本项目实现了待做事项的数字化、本地存储，可视化展示和管理。在存储方式上还可以进一步加强，例如添加服务器云端存储；在功能完备性上还能进一步加强，例如添加消息提醒功能；在某些界面的 ui 还需打磨；在交互方式上还存在一些不稳定性因素，待解决。

## 主要参考文献

- [1] 微信小程序开发官方文档,  
<https://developers.weixin.qq.com/miniprogram/dev/framework/>