# 移动开发技术期末大作业报告

## 项目介绍

本项目名为HelloWorld，是一个基于uni-app实现的日记小程序。

本小程序可以：

①借助云函数、云空间，可以读取与写入云数据库中的信息实现用户的注册与登录。

②在主页的tabbar中存在三个栏目：

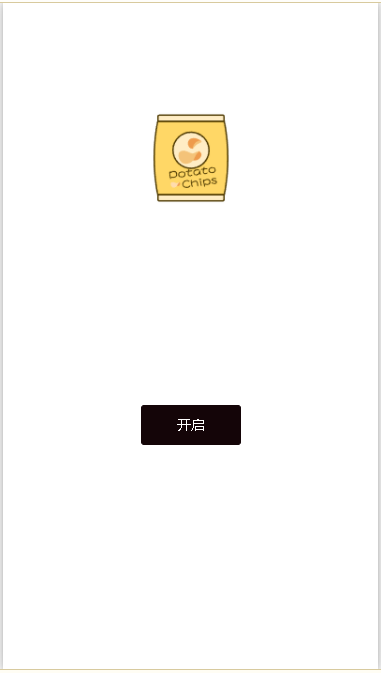
1）时之钟：本页包含了时钟功能，及文字展示。

2）写什么：这个页面包括两个page：“日记”和“Todo”（备忘录)，在这个页面可以进行对日记和todo条目的增删查改。

3）这是我：个人信息页，在此页面可以获取一些个人信息、联系开发人员。

## 项目页面及功能展示

### ①初始页面：（路由：pages/index/index)



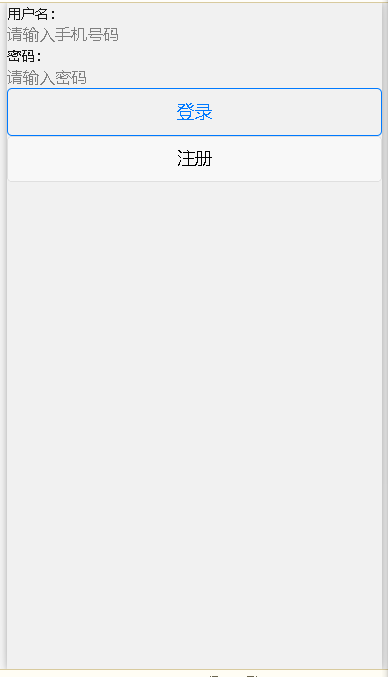
本页面用于点开过渡。点击开始，进入登录/注册页面。

在本页面中，为了提高趣味性，实现了一个动画效果：通过改变图片“potatoChips”的偏移位置与速度，实现图片的晃动。

### ②点击初始页面中的“开启”按钮，进入登陆/注册页：（路由：pages/login/login和pages/register/register)

###### 登录页：（pages/login/login）

1.输入规范：



本页面中对用户登录格式进行了规范；

用户名即为手机号码，此处对手机号码的正确性进行了正则判断：



当鼠标移开时进行判断；如果输入字符串不符合要求，进行提示：



如果格式正确，则提示：



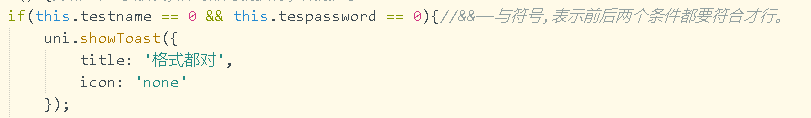
代码实现：

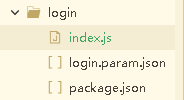


密码输入规范同上：



当输入密码时不进行提示；

点击“登录”按钮，当密码、用户名格式均正确时，会提示：“格式都对”

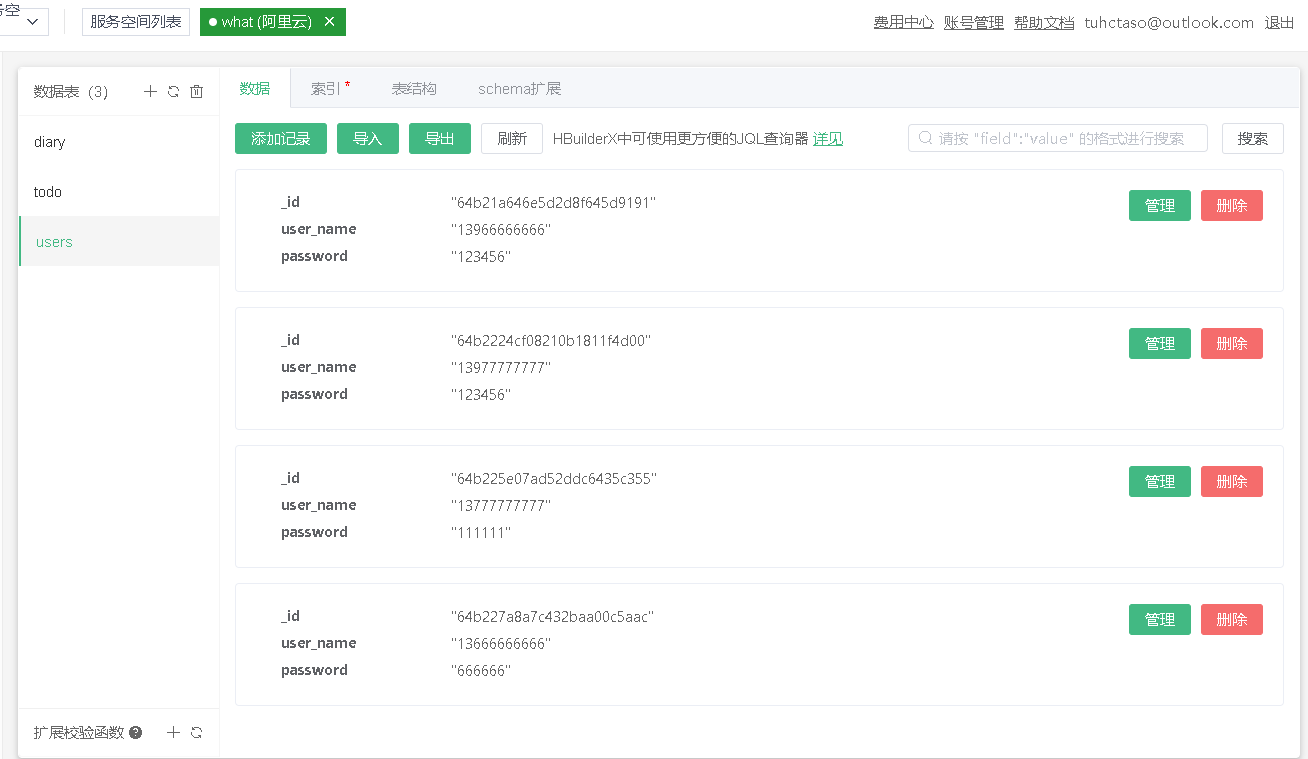
1. 登录云函数的设计：

本项目中，通过与云空间进行关联实现用户信息的调用；

为了实现在云空间中对数据库中信息的查询与匹配，设计了云函数”login”:



云空间中与用户信息相关联的数据表“users”：



其中包含了云空间自动生成的“\_id”,以及自定义的信息“user\_name”和 “password”。



1. 登录函数的设计：

点击登录页的“登录”按键，将会触发点击事件“login”,它将先对云空间中的数据库发出请求，进行数据匹配：



\*返回结果 {“status”:1} 说明 用户不存在，这是在云函数login中进行了规定的。

若用户数据不存在，则提示”用户名不存在“：



当用户名匹配成功，但密码错误时，提示：





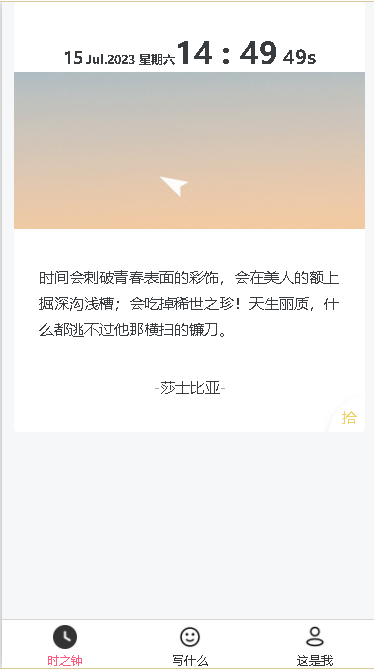
\*返回结果 {“status”:2} 说明 用户名存在，但是密码错误。

当用户名与密码均匹配成功时，登陆成功，



\*返回结果 {“status”:0} 说明 用户名存在，且密码正确。

于是进入主页面（路由：”pages/clock/clock”)



**代码：**

login() {

if(this.testname == 0 && this.tespassword == 0) uni.showToast({

title: '格式都对',

icon: 'none'

});

uniCloud.callFunction({

name: "login",

data: {

"user\_name": this.user\_name,

"password": this.password,

},

success: (res) => {

console.log("对login函数发送请求成功",res)

var status = res.result.status

if(status == 0){

uni.showToast({

title: '登录成功',

icon: 'none'

});

uni.switchTab({url:'/pages/clock/clock'});

}if(status == 1){

uni.showToast({

title: '用户名不存在',

icon: 'none'

});

}if(status == 2){

uni.showToast({

title: '密码错误',

icon: 'none'

});

}

},

fail: (err) => {

console.log("发送请求失败,错误信息如下",res)

}

})

}else{

uni.showToast({

title: '错了,重来',

icon: 'none'

});

}

},

register() {//点击注册按钮跳转注册页面

uni.redirectTo({

url: '/pages/register/register'

});

}

4.点击“注册”按钮，进入注册页面。

###### 注册页：(pages/register/register)



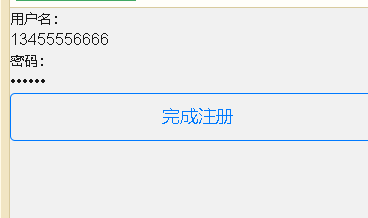
注册页的输入规范和提示功能与登录页是一样的，此处不再进行赘述。

同登录页，为了实现对用户信息的读取、匹配，注册功能还要实现信息的插入；

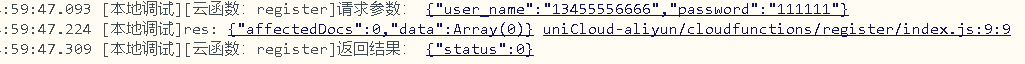
设计云函数“register”：



进行注册：



注册成功：



返回登录页。

### ③主页布局与功能介绍：

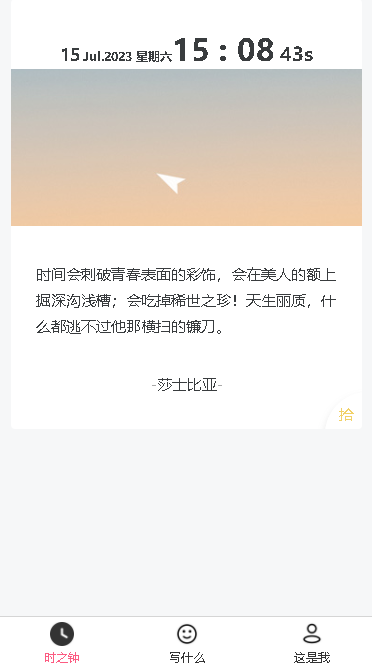
1. 布局介绍：

在主页部分存在三个子页面，可在tabbar中进行切换。

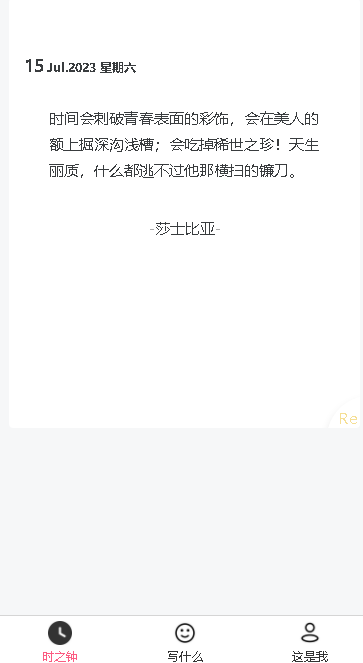


1. 子页面“时之钟”：

1）页面展示：



在本页面中，实现了一个翻页卡片的功能。通过点击卡片右下角的“拾”可进行翻页：



2）视图安排

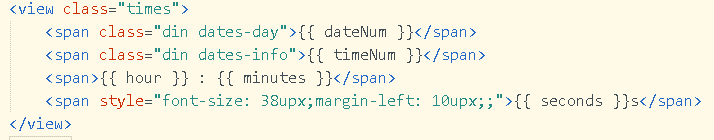
将所有的元素均封装在“box”中，实现所有内容的共同翻转。

3）功能介绍

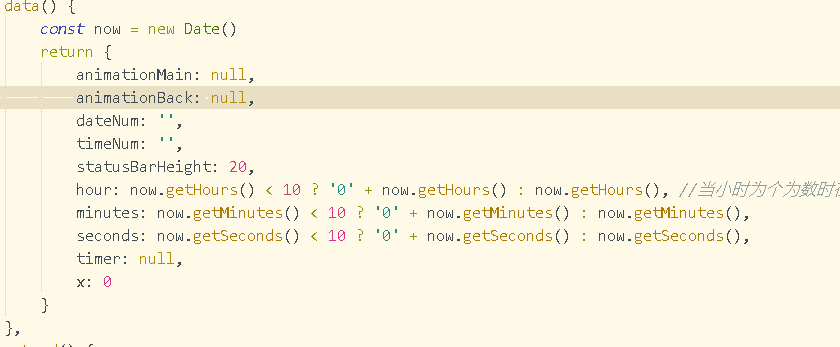
Ⅰ.时间显示：

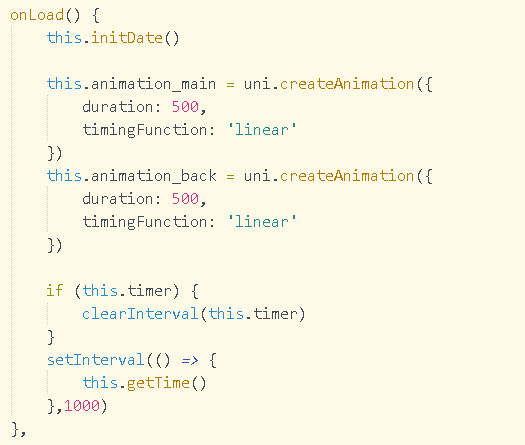
通过对时间先进行初始化，再拉去当前时间；通过将不同单位的时间各自不同的表现格式进行组合，实现时间显示：

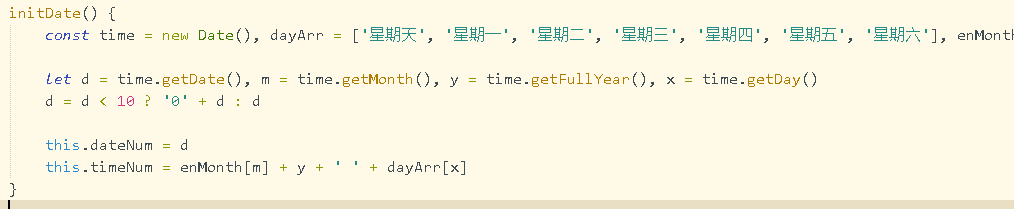
视图：



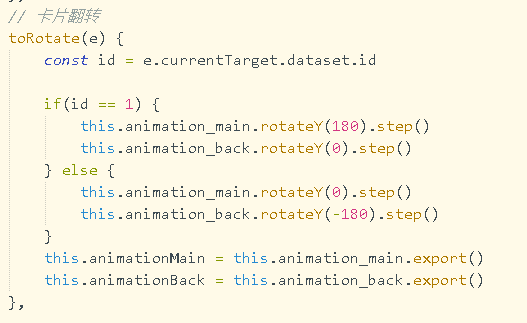
功能（日期初始化与获取）：



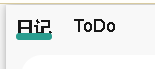




Ⅱ.翻页效果的实现：



1. **子页面“写什么”：（路由：”pages/add/add”)**



通过点击实现副页切换。



Ⅰ.功能介绍：

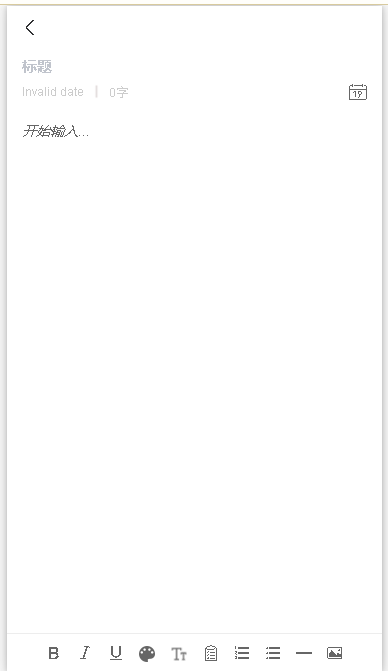
在本页中，可以进行对日记与TODO的增删查改操作：

1. 日记副页：

**增**：点击字符“+”：



进入页面 ”pages/addNote/addNote”:



内容为空时，只显示左上标退出。

在本页中，为了实现多样的输入格式，引入了格式：

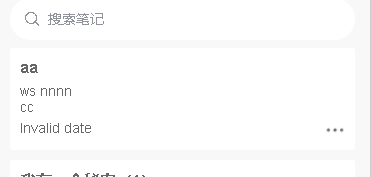
通过对它们的调用，实现了图片插入、字号大小、字体选择等功能。

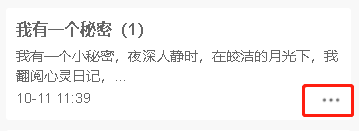


点击右上角的✅：



完成对日记的保存：



**删：**点击：



点击删除：



成功删除。

**查：**

内容查询：

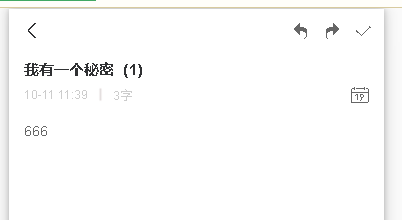


标题查询：



**改：**

点击进入页面，直接对内容进行修改：

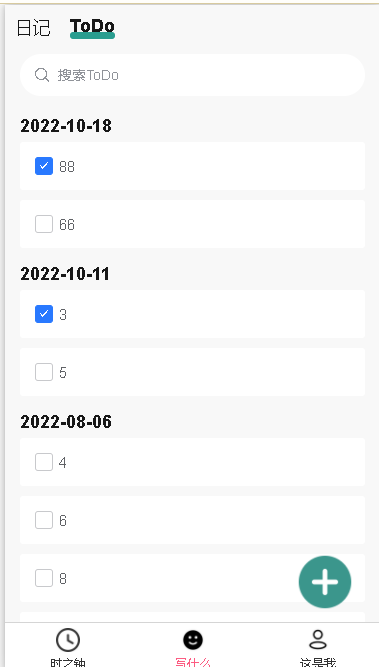


点勾保存：

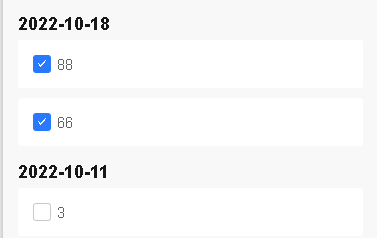


修改成功。

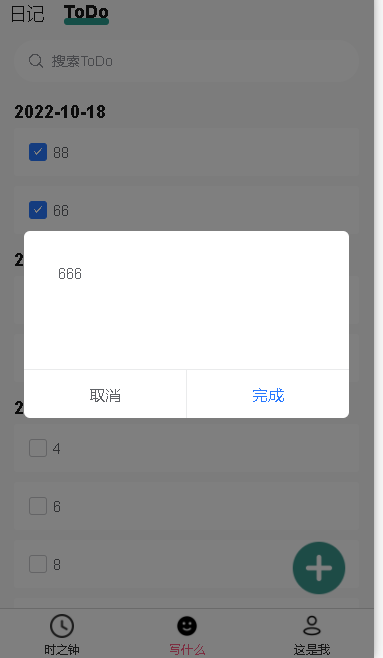
1. ToDo副页：



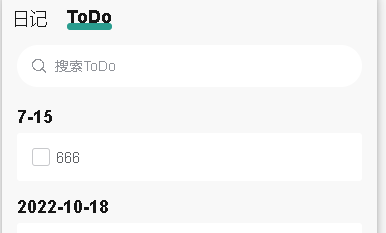
1）实现了方框点击对勾：



2）调取时间新建：



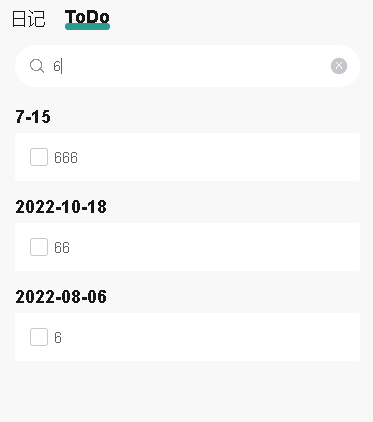
点击完成：



当内容为空时点击确定：



1. 根据内容查询条目：

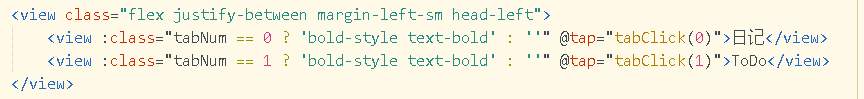


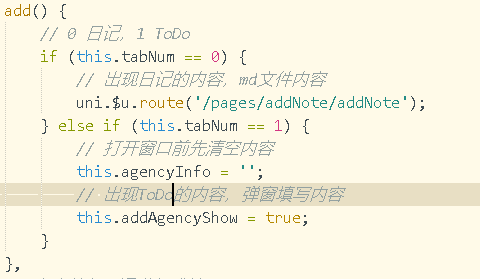
Ⅱ.功能函数

1)通过过滤函数实现查询功能：



1. 在不同的副页中，加号➕有不同的标志；用于对应增加页面的调取：





ToDo的添加函数：

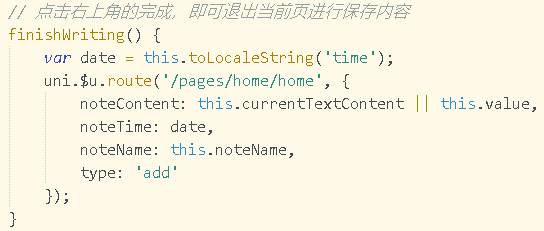


日记的添加与保存在”pages/addNote/addNote”中完成：

当日记页面中内容不为空时，可以选择点击保存：



点击右上角的勾进行保存：



## 返回内容。

4）点击跳转：到对应日记页面



1. **子页面“这是我”：(路由：“pages/my/my”）**



本页面尚未完善；仅选项“退出登录”有实际路由跳转；点击后跳转至“pages/index/index”页面实现用户退出与重新登陆功能。



## 项目心得

在本次大作业项目的设计过程中，我遇到了一些困难。例如：在uni-app中，路由的跳转函数是多样的；且不同的函数拥有着不同的功能与使用条件。就像uni.switchTab里面的url路径必须为tabtar里面的文件，才能进行跳转；

也因为日期格式的差异，进行了多次修改才实现了“写什么”页面中ToDo副页日期的正确调取与显示。

对于云空间的接触，也让我得以通过云数据库实现用户的登录与注册功能；在对云函数进行设计、参数的设计与传递的过程中，我更加了解了改如何去正确地连接云空间的数据库、调用其中的数据。

在进行本次项目的过程中，我加深了对各种uni函数的了解与掌握，熟悉了功能的设计与布局。我还熟悉了许多uni系的插件、外库，通过它们，可以实现更为多样的功能。在这个过程中，我收获了许多新的知识与体悟。