迭代三报告书

课程名称： 智能移动开发

项　　目： 南开之家在线校园社区小程序

学　　号： 2014563

学生姓名： 白志炜

2022 年 6 月 14 日

目 录

第一章 概述

第一节 开发目的

大学生作为社交生活最活跃的群体之一，使用微信、QQ 等主流社交 APP 的频率很高。然而主流社交 APP 主攻的熟人社交却无法满足校园内陌生人社交的需求，这恰好是当今大学校园在线社交场景中的痛点。

本项目开发的南开之家小程序，是一款集表白墙、互助、跳蚤市场、树洞、失物招领等板块于一体的社交类小程序。依托微信小程序官方平台接口实现了免认证图文发帖、评论帖子、回复评论、小程序内实时接收消息、小程序外接收评论和回复消息等功能。本项目旨在为在校大学生提供更便捷的交流平台，让大学生校园生活更加丰富多彩。

第二节 开发环境

本项目的开发环境如下：

前端：微信小程序

后端：微信云开发

开发软件：微信开发者工具

操作系统：Windows10

第三节 开发技术

云数据库：是一个可以在小程序前端操作，也可以在云函数中操作的 json 类型数据库。

云存储：提供稳定、安全、低成本、简单易用的云端存储服务，支持任意数量和形式的非结构化数据存储，如图片、文档、音频、视频、文件等。

云函数：是一个在小程序端定义编写，编写完毕后部署到云服务器，开发者无需购买、搭建服务器，只需编写函数代码并部署到云端，在云服务器中运行的 nodejs 函数，同时云函数之间也可互相调用。

第二章 需求分析和系统设计

第一节 系统功能需求分析

2.1.1 浏览和发布内容

浏览：为了方便用户浏览，用户可以按照帖子的发布时间和热度进行排序。

板块：用户在发帖页面最上方可选择对应板块，以便论坛用户筛选查看。

发布图片内容：用户可以一次上传九张图片。

2.1.2 评论区

楼主：楼主评论回复，昵称后面带有特殊标识。

删除评论与回复：评论人或回复人可以长按自己的回复或评论进行删除，楼主及管理员可以删除所有评论。

收到消息：楼主可接收帖子内直接评论与自己评论被回复的消息通知，被回复者将收到回复消息。

2.1.3 接收消息

在线状态：即时接收评论与回复，收到消息有振动反馈，消息栏可查看消息。

离线状态：当开启接收消息的权限以后，可以在小程序外，即未开启小程序的情况下接收来自小程序的评论与回复消息。

2.1.4 推送文章

管理员可以在云开发后端发布推送文章，用户则可以在小程序的推荐板块看到所推送的内容。

2.1.5 “我的”页面

头像与昵称：用户可以授权使用微信头像与昵称。

删帖：在“我的发布”界面，用户可以直接按住左滑会看到删除选项。

评论过的帖子：用户可以可以查看评论过的帖子。

第二节 系统功能结构设计

该系统主要包括微信小程序的功能模块和管理后端的功能模块两部分。系统总体的功能结构如下图所示：

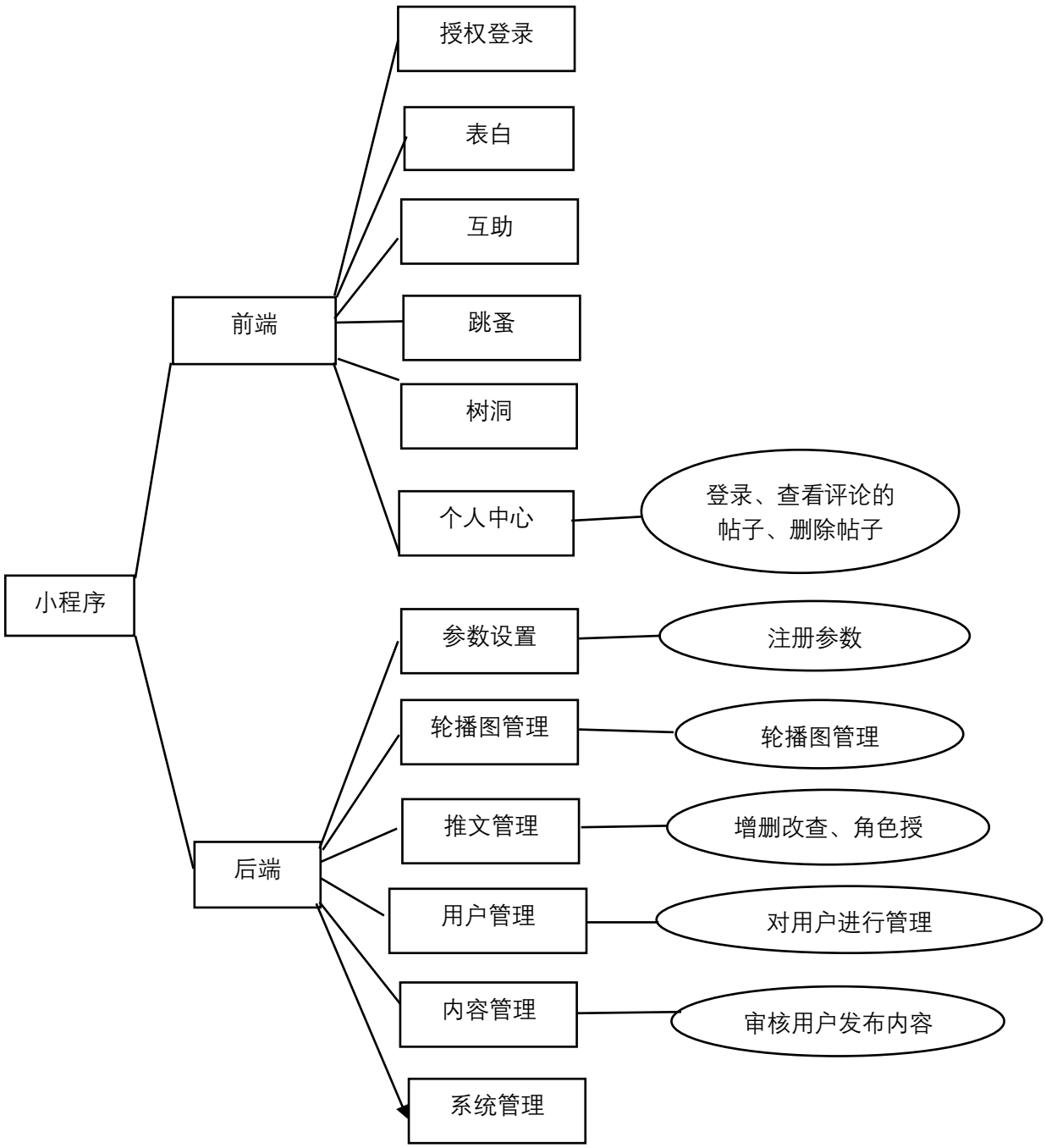

图 1 小程序总体功能结构图

第三章 系统功能实现

第一节 小程序前端实现

4.1.1 评论区功能实现

评论区需要实现三个功能。首先，在楼主的评论回复中，昵称后面将带有特殊标识。其次，删除评论与回复时也要部署确认功能，评论人或者回复人可以长按自己的回复或评论进行删除，楼主及管理员可以删除所有评论。最后，当收到消息时，楼主可接收帖子内直接评论与自己评论被回复的消息通知，被回复者将收到回复消息。评论区功能的实现如下

图 2 评论区功能界面示例

评论区功能实现的关键代码如下：

```
/*消息类型：
1.pinglun
2.huifu
*/
```

```
exports.main = async (event, context) => {
  console.log(event.pinglunnr)
  var pinglunnr=event.pinglunnr
  var pd=event.pd
  var liuyan=pinglunnr.liuyan
  var ku='ss'
  var id=event.pinglunnr.ssid
  var lzid=event.pinglunnr.lzid
  var plrid=event.pinglunnr.plrid
  const _ = cloud.database().command
  if(liuyan==true){
    ku='tj',
    event.pinglunnr.ywnr=event.pinglunnr.title
  }
  if(pd[1]!=""){
    //这说明是回复评论
    console.log("回复评论：",id,pd)
    return await cloud.database().collection(ku).where({
      "_id":id,
      "ss_xx.huifunr.plrid":pd[1],
      "ss_xx.huifunr.time":pd[2]
    }).update({
      data: {
        // 添加记录
        'ss_xx.huifunr.$.huifunb':_.inc(1),
        'ss_xx.huifunr.$.huifu':_.push(event.pinglunnr),
        'ss_xx.huifunb':_.inc(1),
      }
    }).then((res)=>{
      ////给自己加评论过记录
      var pinglunguode={
        id:event.pinglunnr.ssid,
        time:event.pinglunnr.time,
        nr:event.pinglunnr.ywnr,
        plnr:event.pinglunnr.wbnr,
      }
      ////给别人发送消息(被回复者)
  //额外加个判断是否是留言
      var newmessage={
        id:event.pinglunnr.ssid+event.pinglunnr.time,
        ssid:event.pinglunnr.ssid,
        type:"huifu",
        time:event.pinglunnr.time,
        bhfpl:event.pinglunnr.bhfpl,
```

```
      plnr:event.pinglunnr.wbnr,
      name:event.pinglunnr.name,
      photo:event.pinglunnr.photo
    }
    if(liuyan==true){
      newmessage.liuyan=true
    }else{
      newmessage.liuyan=false
    }
    //判断是否回复的自己
    if(event.pinglunnr.bhfid!=plrid){
      //不是回复的自己
      cloud.database().collection('users').doc(event.pinglunnr.bhfid).update({
        data: {
          message:_.push(newmessage)
        }
      }).then((res)=>{
        //console.log("!!!!",res)
        if(pd[0]!=true && liuyan==false){
          //首次评论家记录
          cloud.database().collection('users').doc(plrid).update({
            data:{
              pinglunguode:_.push(pinglunguode)
            }
          }).then((res)=>{
            console.log("成功")
            //前面重要的做完就在这里进行订阅消息的发送了          }}
        console.log("开始检测进行回复")
          //1.获取待操作用户的信息
          cloud.database().collection('users').doc(event.pinglunnr.bhfid).get().then((r
es)=>{
            //2.取到用户数据进行判断在线状态
            console.log("取到用户数据进行判断在线状态:",res.data.online)
            console.log("取到用户数据进行判断授权状态:",res.data.allow)
            console.log("取到用户数据进行判断次数剩余状态:",res.data.msgnb)
            var online=res.data.online
            var allow=res.data.allow
            var msgnb=res.data.msgnb
            var openid=res.data._openid
            if(allow){
              //开启了授权才可：
              console.log("开启了授权才可")
              if(!online){
                //不在线才可
```

```
console.log("不在线才可")
if(msgnb[1]>0)//可推送回复消息
 //消息数据格式化
 //name 10
 //thing 20
 var name=event.pinglunnr.yuanname
 if(name==undefined){
  name=event.pinglunnr.name
 }
 console.log("length:",name)
 if(name.length>20){
  name=name.substr(0,17)+"..."
 }
 if(newmessage.plnr.length>20){
  newmessage.plnr=newmessage.plnr.substr(0,17)+"..."
 }
 if(newmessage.bhfpl.length>20){
  newmessage.bhfpl=newmessage.bhfpl.substr(0,17)+"..."
 }
 console.log("推送回复消息")
 console.log("name:",name)
 console.log("plnr:",newmessage.plnr)
 console.log("date:",event.pinglunnr.riqi)
 cloud.openapi.subscribeMessage.send({
  touser: openid,
  page: 'pages/index/index',
  lang: 'zh_CN'
  data: {
   thing3: {
    //评论人
    value: name
   },
   thing2: {
    //评论内容
    value: newmessage.plnr
   },
   time4: {
    //评论时间
    value: event.pinglunnr.riqi
   },
   thing1: {
    //原评论
    value: newmessage.bhfpl
   }
```

```
                    },
.....
}
```

4.1.2 消息中心模块实现

用户出于在线状态时，可以即时接收评论与回复，且收到消息有振动反馈，并在消息栏可查看消息。

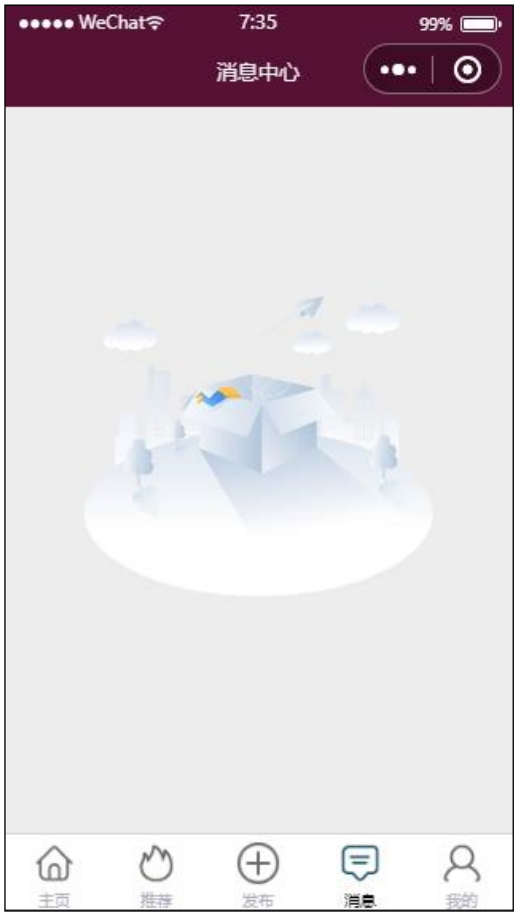

图 3 消息中心模块显示

消息中心模块实现的关键代码如下：

```
//主动授权
  allow(){
    const tmpllds=this.data.tmpllds
    var that=this
    wx.requestSubscribeMessage({
    tmpllds:tmpllds,
    success(res) {
```

```
console.log("订阅消息 API 调用成功：",res)

var diyi='lOXeoWHeYUbuM3TJKzyPM1XR-J7iX6HIc9-YWYcYny0'
var dier='9kAS4BdEjH46glaAr-wuo_qZndRNkp5Zqe3vWZbAab4'
var msgnb=that.data.msgnb
console.log(res[diyi],res[dier])
if(res[diyi]=='accept'){
  //第一个模板,评论
  msgnb[0]++
}else if(res[diyi]=='reject'){
  wx.showToast({
    title: '您拒绝了接收评论',
    icon:'none',
    duration: 1000
  })
}
if(res[dier]=='accept'){
  //第二个模板,回复
  msgnb[1]++
}else if(res[dier]=='reject'){
  wx.showToast({
    title: '您拒绝了接收回复',
    icon:'none',
    duration: 1000
  })
}
console.log(msgnb)
that.setData({
  msgnb:msgnb
})
app.userInfo.allow=true
////首次授权调用授权成功,下次直接进入剩余推送次数页面
//首次授权调用
that.setData({
  allow:'true',
  msgnb:msgnb
})
db.collection('users').doc(app.userInfo._id).update({
  data:{
    allow:true,
    msgnb:msgnb
  }
})
console.log('已经进行了第一次授权，不再出现此页面！')
```

```
    },
    fail(res) {

      console.log("订阅消息 API 调用失败：",res)

      var errCode=res.errCode
      if(errCode==20004){
        wx.showToast({
          title: '您拒绝接收消息',
          icon:'none'
        })
        this.turrenoff()
      }}
    })
},
//增加授权
allowup(e){
    console.log("e:",e.currentTarget.dataset.index)
    var index=e.currentTarget.dataset.index
    var diyi='lOXeoWHeYUbuM3TJKzyPM1XR-J7iX6Hlc9-YWYcYny0'
    var dier='9kAS4BdEjH46glaAr-wuo_qZndRNkp5Zqe3vWZbAab4'
    var tmpllds=this.data.tmpllds
    if(index==0){
      console.log("被评论")
      tmpllds=[diyi]
    }else{
      console.log("被回复")
      tmpllds=[dier]
    }
    var that=this
    wx.requestSubscribeMessage({
    tmpllds:tmpllds,
    success(res) {

      console.log("订阅消息 API 调用成功：",res,"up")


      var msgnb=that.data.msgnb
      console.log(res[diyi],res[dier])
      if(res[diyi]=='accept'){
        //第一个模板,评论
        msgnb[0]++
      }else if(res[diyi]=='reject'){
        wx.showToast({
          title: '您拒绝了接收评论',
```

```
                icon:'none',
                duration: 1000
            })
        }
        if(res[dier]=='accept'){
            //第二个模板,回复
            msgnb[1]++
        }else if(res[dier]=='reject'){
            wx.showToast({
                title: '您拒绝了接收回复',
                icon:'none',
                duration: 1000
            })
        }
        console.log(msgnb)
        that.setData({
            msgnb:msgnb
        })
    },
    fail(res) {

        console.log("订阅消息 API 调用失败：",res)

        var errCode=res.errCode
        if(errCode==20004){
            wx.showToast({
                title: '您拒绝接收消息',
                icon:'none'
            })
            this.turrenoff()
        }
    }
    })
},
onReady: function () {
},
onShow: function () {
    //var now=new Date()//.getTime()//现在的时间
    // var hour = now.getHours();
    // console.log("现在的小时：",hour)
},
onUnload: function () {
    //加到数据库
    console.log("加到数据库")
    var msgnb=this.data.msgnb
```

```
        var allow=app.userInfo.allow
        db.collection('users').doc(app.userInfo._id).update({
          data:{
            msgnb:msgnb,
            allow:allow
          }
        })
        console.log('增加了所有授权')
    },
})
```

第二节 云开发后端实现

本微信小程序的开发依赖于微信云开发。利用云数据库、云存储、云函数技术实现了对轮播图、文章推送、用户和系统的管理。

4.2.1 数据库

数据库主要有四个集合 ss、system、tj、user 分别代表帖子、系统配置、推荐文章、用户数据四张数据表，管理员可以在云数据对轮播图和推文等进行管理。如下图所示：

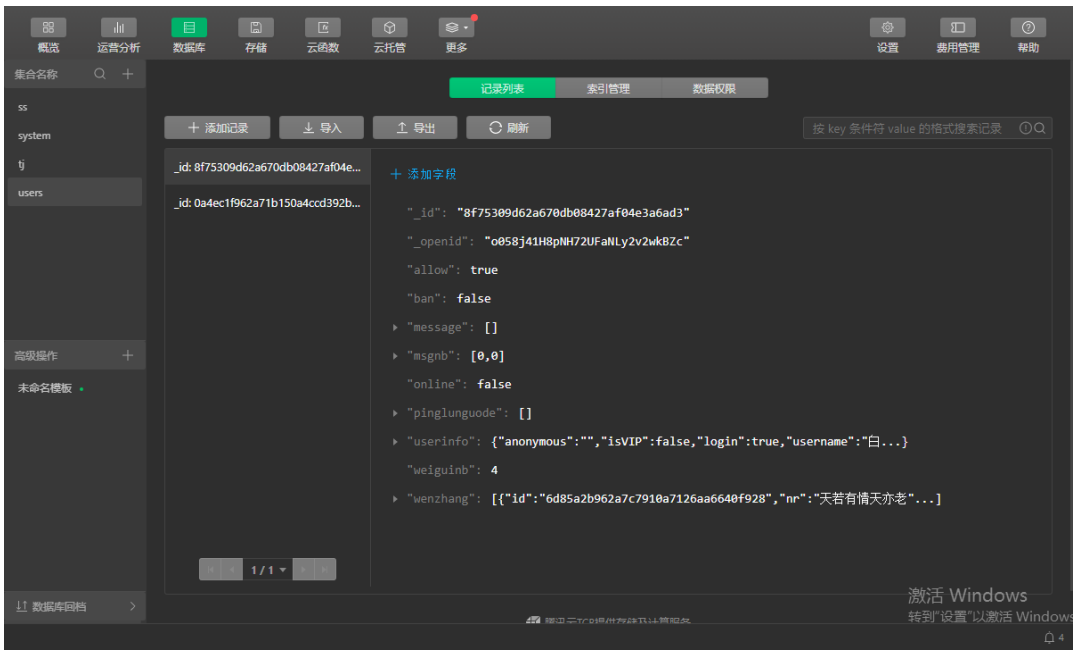

图 4 云数据库界面

4.2.2 云存储

云存储可以提供稳定、安全、低成本、简单易用的云端存储服务，支持任意数量和形式的非结构化数据存储，如图片、文档、音频、视频、文件等。本项目在云开发控制台建立了两个存储文件夹，其中 ss_img 用于存储帖子的图片，tj_img 用于存储主页轮播图的图片，如下图所示：

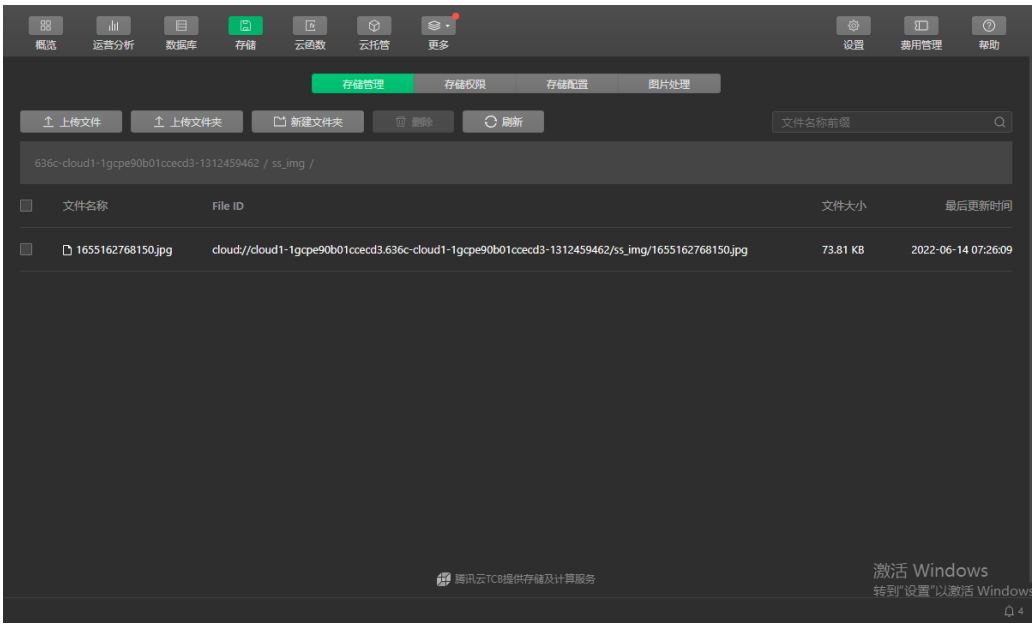

图 5 云存储界面

4.2.3 云函数

云函数是一个在小程序端定义编写，编写完毕后部署到云服务器，这使得开发者无需搭建服务器，只需编写函数代码并部署到云端，在云服务器中运行的 nodejs 函数，同时云函数之间也可互相调用。云函数的部署如下图所示：

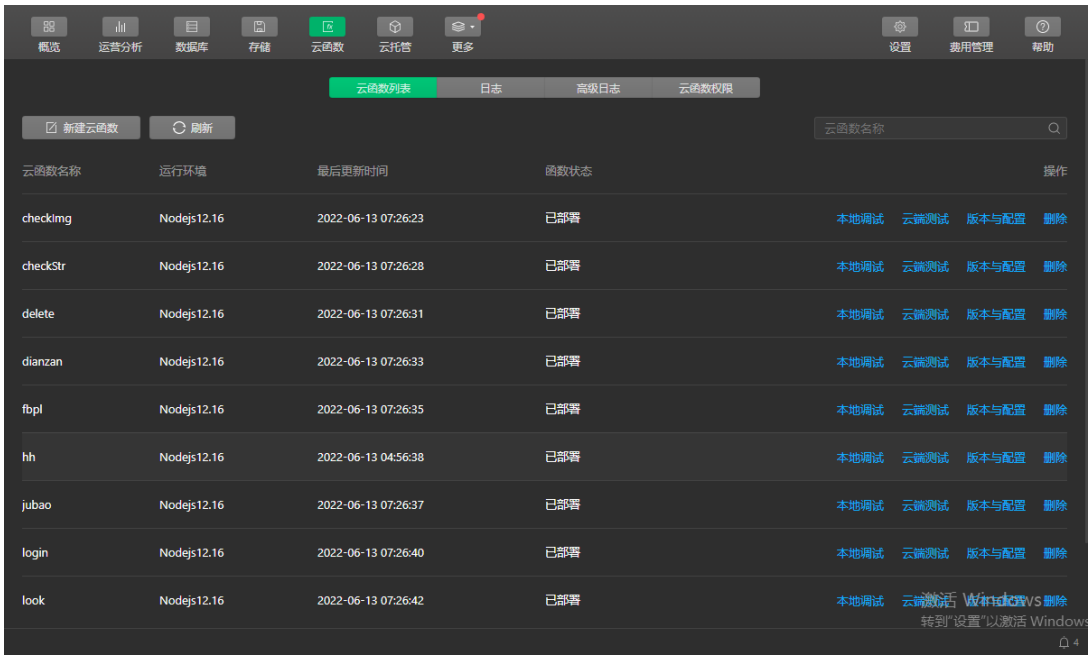
概览
运营分析
数据库
存储
云函数
云托管
更多
设置
费用管理
帮助
云函数列表
日志
高级日志
云函数权限
新建云函数
刷新
云函数名称
云函数名称 运行环境 最后更新时间 函数状态 操作
checkImg Nodejs12.16 2022-06-13 07:26:23 已部署 本地调试 云端测试 版本与配置 删除
checkStr Nodejs12.16 2022-06-13 07:26:28 已部署 本地调试 云端测试 版本与配置 删除
delete Nodejs12.16 2022-06-13 07:26:31 已部署 本地调试 云端测试 版本与配置 删除
dianzan Nodejs12.16 2022-06-13 07:26:33 已部署 本地调试 云端测试 版本与配置 删除
fbpl Nodejs12.16 2022-06-13 07:26:35 已部署 本地调试 云端测试 版本与配置 删除
hh Nodejs12.16 2022-06-13 04:56:38 已部署 本地调试 云端测试 版本与配置 删除
jubao Nodejs12.16 2022-06-13 07:26:37 已部署 本地调试 云端测试 版本与配置 删除
login Nodejs12.16 2022-06-13 07:26:40 已部署 本地调试 云端测试 版本与配置 删除
look Nodejs12.16 2022-06-13 07:26:42 已部署 本地调试 云端测试 版本与配置 删除
转到"设置"以激活 Windows

图 6 云函数部署界面