



时间胶囊 HeyKong

1. 项目介绍

1.1 灵感来源

- 刘聪 Hey Kong 🧑
 - 这是刘聪的一首歌，讲的是过去的自己与现在的自己用写信对话的方式描述自己的成长经历
- 半年才能写一次的日记本 📔
 - 在家里的柜子有一个本子，然后只有每学期结束回去的时候才可以写一写，所以可以跟半年后的自己对话

1.2 面向用户及场景

2013年9月，由苹果公司创始人史蒂夫·乔布斯30年前埋入地下的“时间胶囊”被发掘。

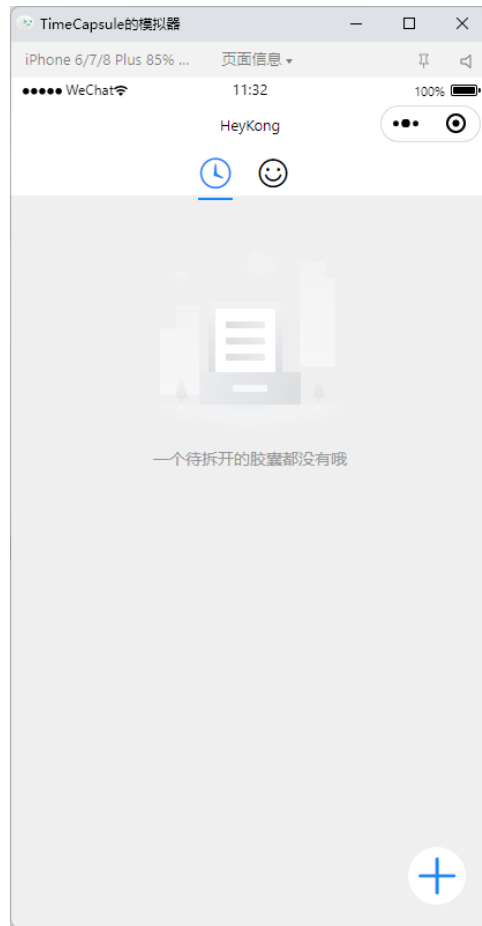
时间胶囊，或者说慢递，追求的是情感的传达并不是速度

时间胶囊的内容可以是对自己，朋友，爱人，家人，或者任何人想说的话，并且设置一个未来能够打开“时间胶囊”的时间，也就是未来收信时间，这样他们就能在未来的某天收到你的时间胶囊

2. 当前完成情况

已完成个人认为时间胶囊需要的功能，但后期需要继续完善，存在很多可以优化的部分

1. 刚进入的时候会显示空



2. 点击上方两个图标可以动态切换组件且有动画



3. 右滑滑块可以删除该条胶囊



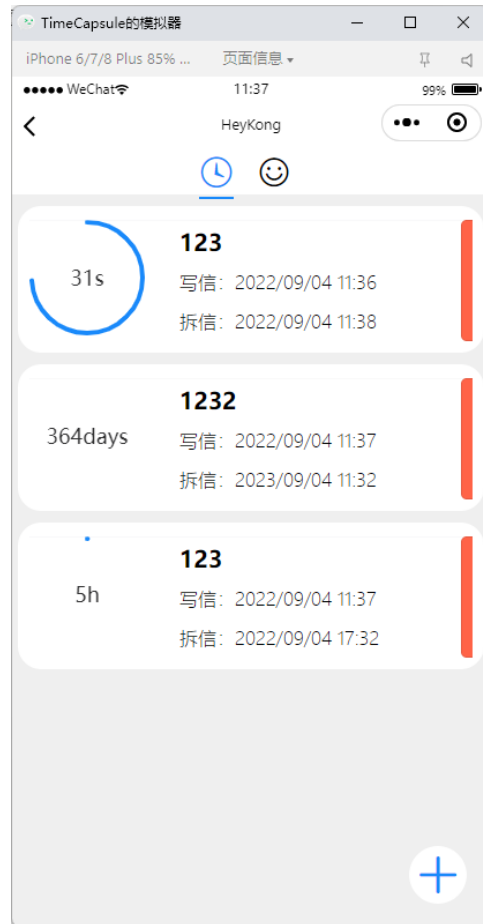
4. 点击右下方的+可以进入写信页，且点击会旋转，添加了动画，每个对应的信息错误会提供相应的反馈



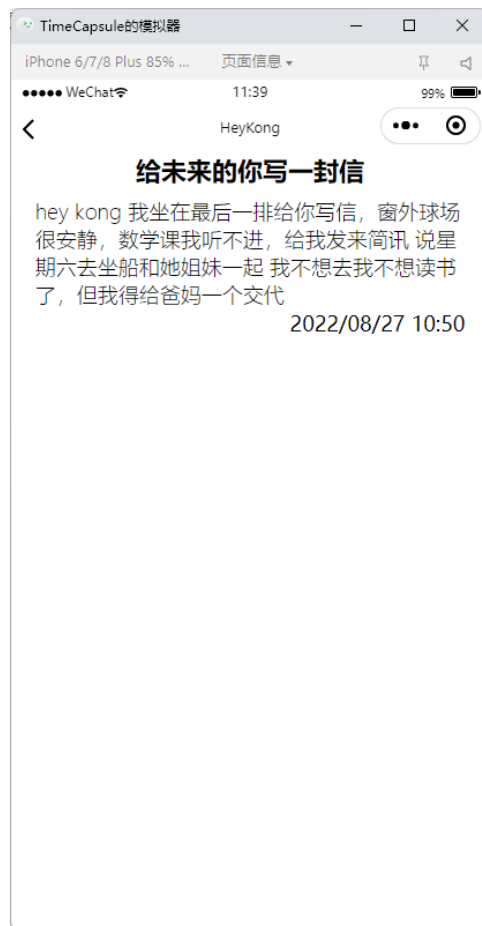
5. 当填写符合标准点击提交就可以发送



6. 主页未开启的胶囊时间修改，分别为days, h, m : s



7. open中查看已打开的界面，点击即可跳转



3. 页面、控件、技术

3.1 页面

- index(主页)
 - close(未开启的胶囊)
 - open(已开启的胶囊)
- write(写信页)
- info(胶囊具体内容)

3.2 控件

主页中的close、open为自定义的组件，此外还使用vant-weapp辅助开发


```

"usingComponents": {
  "vant-button": "@vant/weapp/button/index",
  "vant-icon": "@vant/weapp/icon/index",
  "van-empty": "@vant/weapp/empty/index",
  "van-circle": "@vant/weapp/circle/index",
  "van-datetime-picker": "@vant/weapp/datetime-picker/index",
  "van-action-sheet": "@vant/weapp/action-sheet/index",
  "van-cell-group": "@vant/weapp/cell-group/index",
  "van-field": "@vant/weapp/field/index",
  "van-uploader": "@vant/weapp/uploader/index",
  "van-notify": "/miniprogram_npm/@vant/weapp/notify/index",
  "van-swipe-cell": "@vant/weapp/swipe-cell/index",
  "van-cell": "@vant/weapp/cell/index"
}

```

3.3 技术

- 父子组件通信

```
<my_close wx:if="{{close_open}}" close="{{close}}"></my_close>
```

- 生命周期函数

onload()//页面加载时触发。一个页面只会调用一次，可以在 onLoad 的参数中获取打开当前页面路径中的参数。
onShow()//页面显示/切入前台时触发。

- JS时间戳的使用

- 获取时间差值
- 获取当前时间
- 根据时间显示不同部分

■

```

timedifference: function (faultDate, completeTime,id=0) {
  var stime = Date.parse(new Date(faultDate)); //获得开始时间的毫秒数
  var etime = Date.parse(new Date(completeTime)); //获得结束时间的毫秒数
  var usedTime = etime - stime; //两个时间戳相差的毫秒数
  if(id==1)return usedTime
  var days = Math.floor(usedTime / (24 * 3600 * 1000));
  if(days>0){

```

```

        return days+"days"
    }
    //计算出小时数
    var leave1 = usedTime % (24 * 3600 * 1000); //计算天数后剩余的毫秒数
    var hours = Math.floor(leave1 / (3600 * 1000)); //将剩余的毫秒数转化成小时数
    if(hours>0){
        return hours+'h'
    }
    //计算相差分钟数
    var leave2 = leave1 % (3600 * 1000); //计算小时数后剩余的毫秒数
    var minutes = Math.floor(leave2 / (60 * 1000)); //将剩余的毫秒数转化成分钟
    //计算相差秒数
    var leave3 = leave2 % (60 * 1000); //计算分钟数后剩余的毫秒数
    var seconds = Math.floor(leave3/1000); //将剩余的毫秒数转化成秒数

    var dayStr = days == 0 ? "" : days + "d:";
    var hoursStr = hours == 0 ? "" : hours + "h:";
    var minutesStr = minutes == 0 ? "" : minutes + "m:";
    var time = dayStr + hoursStr + minutesStr + seconds + "s";
    return time;
}, //计算两个时间之间的时间差

```

- 微信小程序本地存储

```

wx.setStorageSync('close', JSON.stringify(this.data.close))
let close1 = JSON.parse(wx.getStorageSync('close') || '[]')

```

- 组件动态切换及加载

```

<my_close wx:if="{{close_open}}" close="{{close}}"></my_close>
<my_open wx:else="{{close_open}}" close="{{openlist}}" empty="{{empty}}" ></my_open>

```

- CSS&&JS动画

- 人机交互的技术

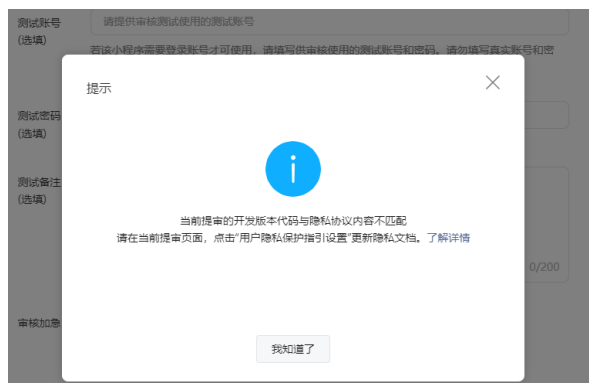
- 观感

- 整体采用蓝白的配色，干净简洁，每一步的操作都给用户反馈，包括动画、弹窗

4. 上线及测试情况

体验版本已提交

因开发版本代码与开发协议不符所以无法正式上线
已真机调试，暂时没有发现问题



5. 实际遇到的问题

- 微信开发者工具
 - 动态刷新偶尔卡壳
 - bindtap没有反应
- vant-weapp没有那么的好用
 - 样式冲突
 - 找不到对应的组件
- 数据动态刷新

6. 心得体会

- 对小程序的理解加深
- 熟悉vant-weapp的使用
- 浪漫
 - 这是一个与未来自己的浪漫的小程序，其实我们每天都在过着没法重复的生活，这一路上只有我们最理解自己，我们也一直在与过去的自己和解，就像上课卡的bug，现在运行却一点问题都没有，有一个彩蛋没给大家

