《开源软件基础》课程大作业

题目 银行叫号系统

姓名 刘潇

班级 软1514

学号 201592079

时间 2017年 12 月 28 日

**本项目github地址：**

**web端及服务器端：**

[**https://github.com/Nepgear-mpbl/counter-system**](https://github.com/Nepgear-mpbl/counter-system)

**Android端及大作业电子档：**

[**https://github.com/mcat1997/BankQueue**](https://github.com/mcat1997/BankQueue)

大连理工大学软件学院

School of Software, Dalian University of Technology

目录

[1.简介 4](#_Toc503017094)

[1.1.背景 4](#_Toc503017095)

[1.2.项目介绍 4](#_Toc503017096)

[2.系统总体概述 5](#_Toc503017097)

[2.1.目标 5](#_Toc503017098)

[2.2.用户类和特性 5](#_Toc503017099)

[2.3.运行环境 5](#_Toc503017100)

[2.3.1.硬件环境 5](#_Toc503017101)

[2.3.2.软件环境 6](#_Toc503017102)

[3.系统特性 7](#_Toc503017103)

[3.1说明和优先级 7](#_Toc503017104)

[3.2功能需求 7](#_Toc503017105)

[3.2.1移动APP端功能 7](#_Toc503017106)

[3.2.2柜台Web端功能 8](#_Toc503017107)

[3.2.3服务器端功能 8](#_Toc503017108)

[3.3功能详述 9](#_Toc503017109)

[4.概要设计 10](#_Toc503017110)

[4.1总体设计 10](#_Toc503017111)

[4.1.1基础设计概念和处理流程 10](#_Toc503017112)

[4.1.2结构 10](#_Toc503017113)

[4.1.3功能分配 11](#_Toc503017114)

[4.2接口设计 11](#_Toc503017115)

[4.2.1用户接口 11](#_Toc503017116)

[4.2.2外部接口 11](#_Toc503017117)

[4.2.3内部接口 12](#_Toc503017118)

[4.3运行设计 12](#_Toc503017119)

[4.3.1数据库设计 12](#_Toc503017120)

[4.3.2出错信息 12](#_Toc503017121)

[5.详细设计 13](#_Toc503017122)

[5.1Model层（数据库结构设计） 13](#_Toc503017123)

[5.2.View层 14](#_Toc503017124)

[5.2.1Web端 14](#_Toc503017125)

[5.2.2.APP端 15](#_Toc503017126)

[6.测试 16](#_Toc503017127)

[6.1Web端运行截图 16](#_Toc503017128)

[6.2.Android端运行截图 18](#_Toc503017129)

[7.GitHub 21](#_Toc503017130)

# 1.简介

## 1.1.背景

随着人们生活节奏的加快，银行的手工叫号系统越来越不适合社会的发展。每次到银行进行业务处理时，都要在银行大厅内浪费一个小时甚至数个小时的时间等待叫号，这是非常不合理的一种现象。本项目组在生活中观察到了这个问题，为了解决这个问题，我们研发了一款新型的银行叫号系统。

## 1.2.项目介绍

本项目由本项目组开发，分为移动端（客户端），服务器端以及Web端（柜员端）。操作流程大概为客户端进行叫号，服务器端返回等待信息，并实时刷新；柜员端进行接号，服务器端处理请求并推送给客户端。本系统封装良好，APP及Web简单易用，实现了傻瓜式操作。

# 2.系统总体概述

## 2.1.目标

本系统的主要目标是一款基于移动端和服务器端和Web端的银行叫号系统。可以用于银行及其他公共服务的叫号服务。柜台端使用Web页面处理叫号请求，用户通过移动端进行叫号，服务器对叫号数据及逻辑进行处理。实现有序叫号，最短分配，使叫号这一原本比较费时费力的活动，变得省时高效自动化。

根据项目的计划，项目目标主要有以下几个方面：

1. 移动APP
2. 柜台Web管理页面
3. 服务器数据处理控制台

## 2.2.用户类和特性

主要用户是银行柜员和办业务的客户。界面需要友好，客户及会员实现傻瓜式操作。

本系统预期的使用频率为：8\*5小时

## 2.3.运行环境

### 2.3.1.硬件环境

移动端支持安卓手机，Web端支持IE9及其以上浏览器，服务器端需要服务器，对硬件网络无实际特殊要求。

### 2.3.2.软件环境

1用Mysql作为数据存储数据库；

2采用uWSGI服务器部署

3服务器端使用python环境

# 3.系统特性

## 3.1说明和优先级

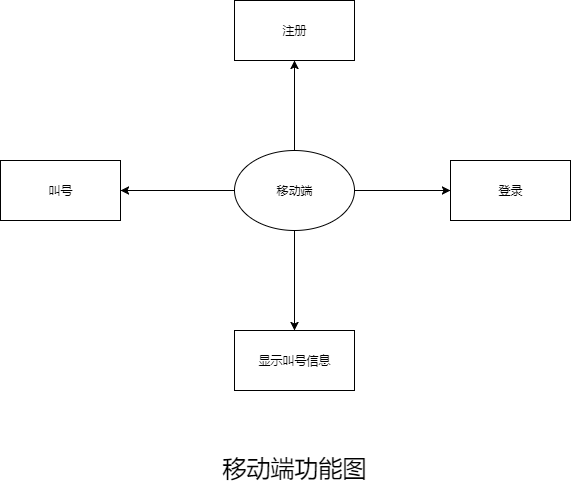
本系统根据最终用户所具有的不同功能将用户分为以下两类：

1. 客户：需要叫号服务以及服务具体信息
2. 柜台：需要接号服务

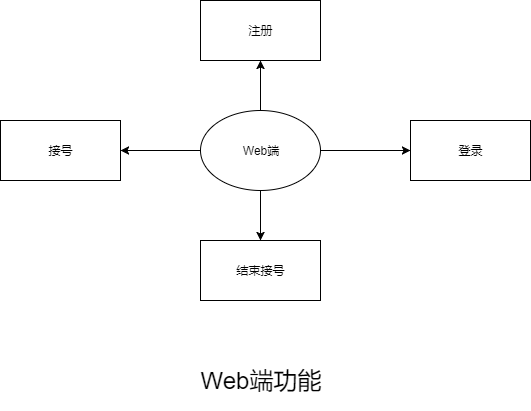
产品在做最基础的架构之后，应该优先做出手机APP的模块，然后才是柜员和服务器端的功能

## 3.2功能需求

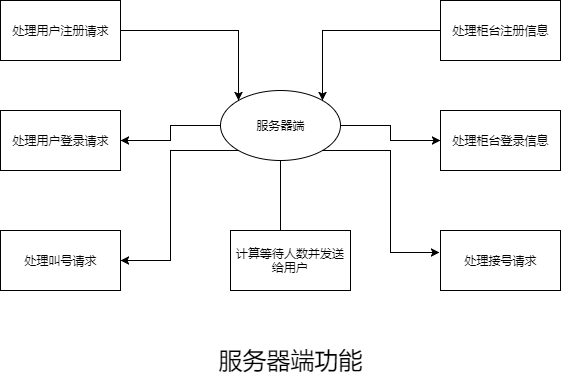
### 3.2.1移动APP端功能

****

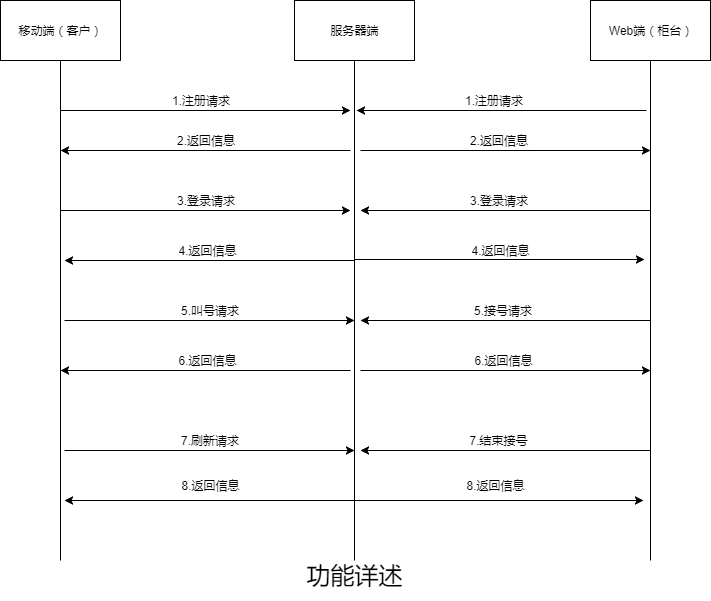
### 3.2.2柜台Web端功能

****

### 3.2.3服务器端功能

****

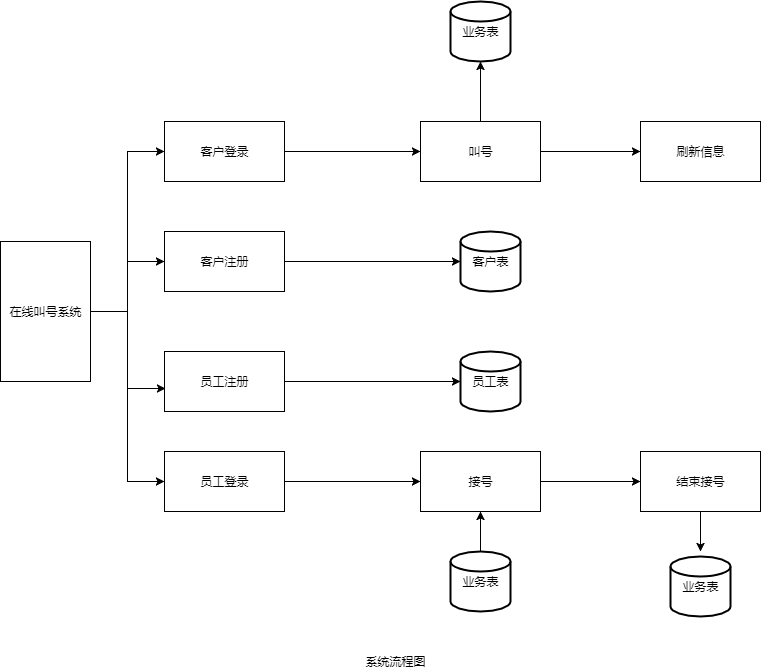
## 3.3功能详述

****

# 4.概要设计

## 4.1总体设计

### 4.1.1基础设计概念和处理流程



### 4.1.2结构

该系统设计采用传统MVC架构

Model层：数据库表模型实体。

View层：用户视图，分为web端和APP端。

Controller层：逻辑控制，对数据库及请求处理并回复。

### 4.1.3功能分配

|  |  |  |
| --- | --- | --- |
|  | 客户 | 员工 |
| 叫号 | √ |  |
| 接号 |  | √ |
| 登录 | √ | √ |
| 注册 | √ | √ |
| 获取排队信息 | √ | √ |
| 结束接号 |  | √ |

## 4.2接口设计

### 4.2.1用户接口

在用户界面部分，根据需求分析的结果，用户需要一个友善使用户界面。在界面设计上，应做到简单明了，易于操作。使用Andriod Studio工具拖拽出美观的界面。

### 4.2.2外部接口

服务器端程序可DAO（数据访问对象）应用程序编程接口（API），对MySql数据库进行所有的访问。服务器程序上可使用MySql的对数据库的备分命令，以做到对数据的保存。在网络软件接口方面，使用Http传输协议,对数据进行网络传输及接收

### 4.2.3内部接口

获取当前正在等待的用户队列

获取用户队列的前方人数

## 4.3运行设计

服务器程序在有请求时启动逻辑处理模块，通过各模块之间的调用，查询数据并回复。调用网络传输模块，将回复传给客户端，服务器程序的接收网络数据模块必须始终处于活动状态。接收到数据后，调用数据处理/查询模块对数据库进行访问，完成后调用网络发送模块，将信息返回客户端。

### 4.3.1数据库设计

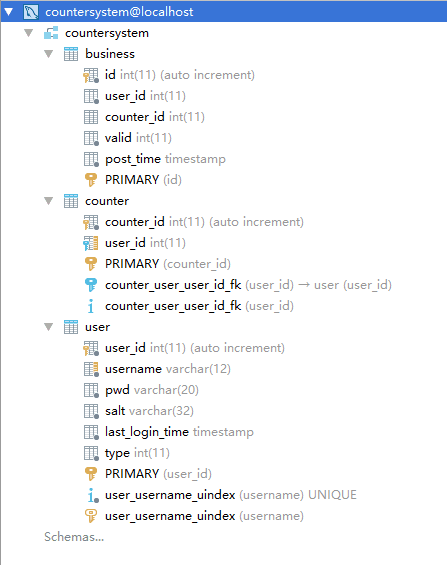
见详细设计

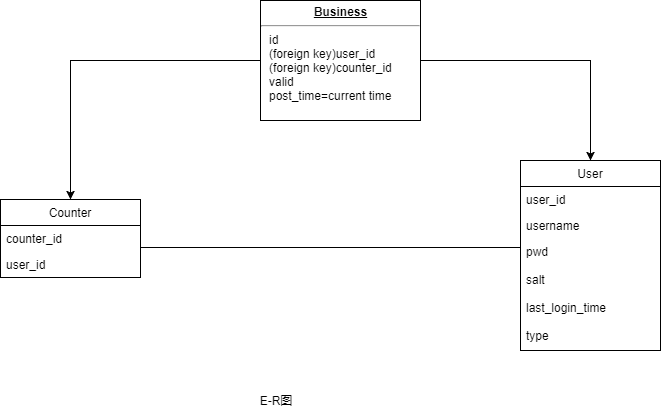
### 4.3.2出错信息

|  |  |  |
| --- | --- | --- |
| 出错名称 | 系统输出信息 | 处理方法 |
| 用户名输入错误 | “该用户不存在！” | 进入登录页面 |
| 密码输入错误 | “您的密码错误，请重新输入！” | 进入登录页面 |

# 5.详细设计

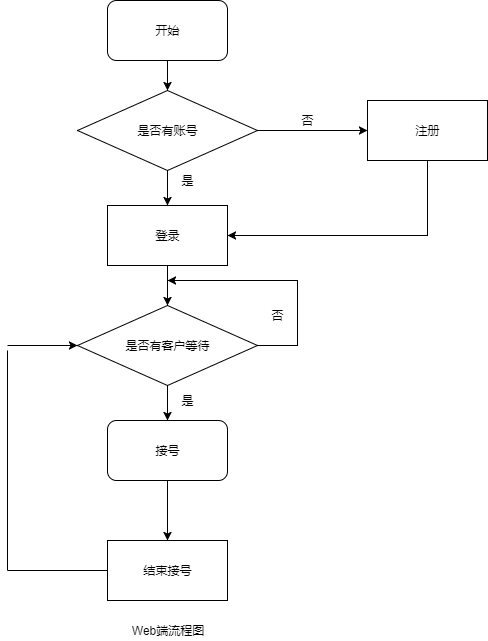
## 5.1Model层（数据库结构设计）



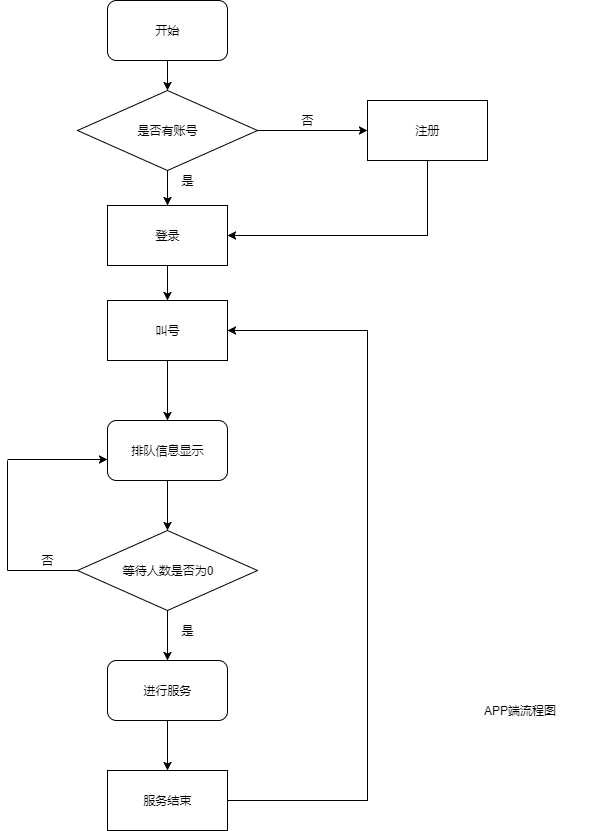


## 5.2.View层

### 5.2.1Web端



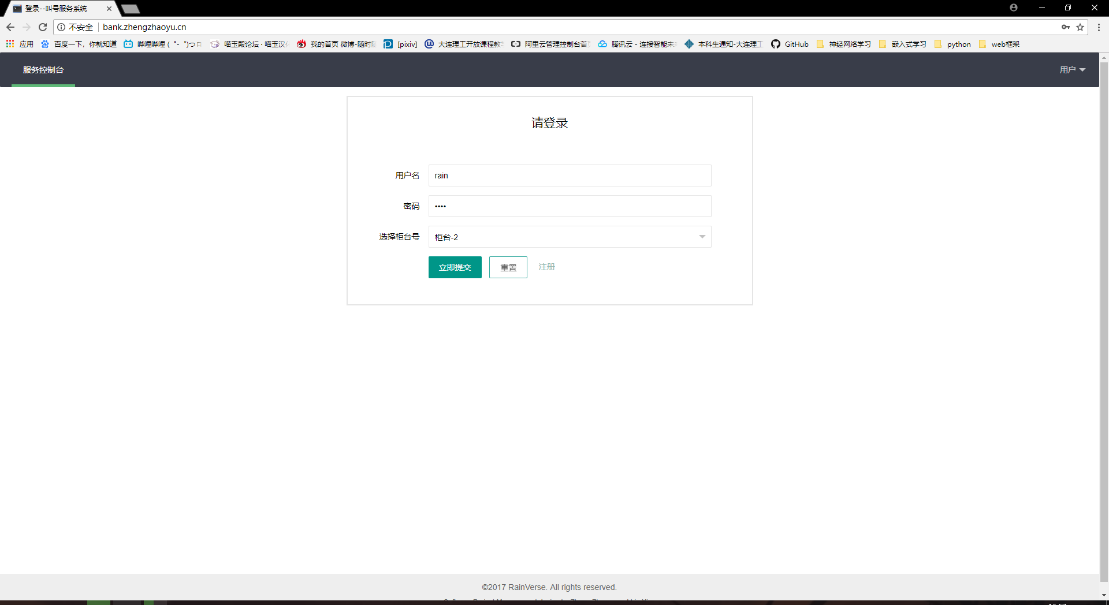
### 5.2.2.APP端



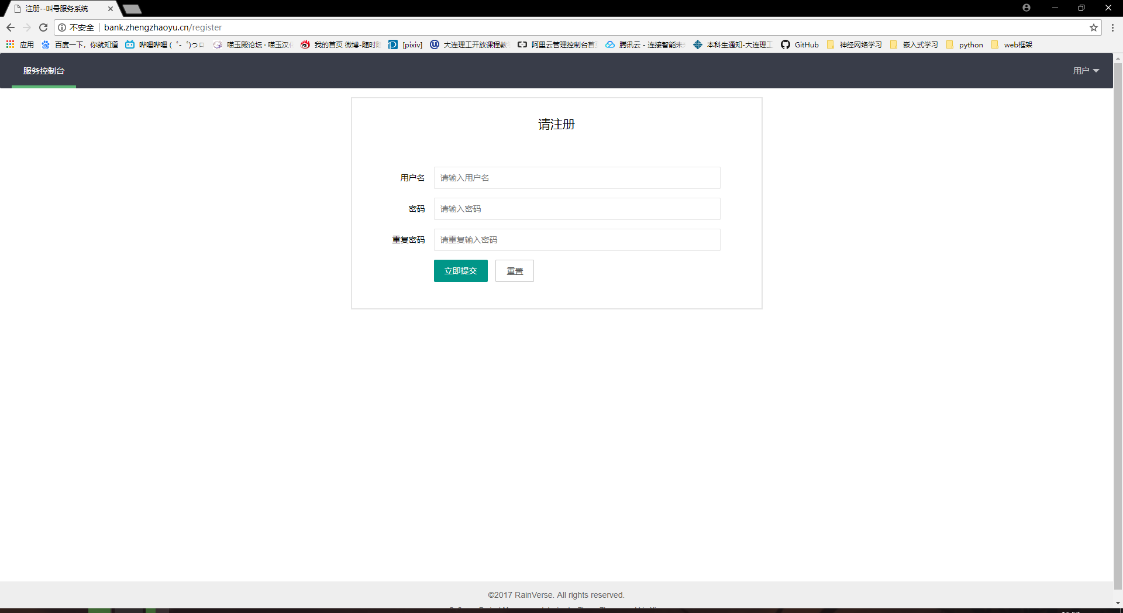
# 6.测试

## 6.1Web端运行截图

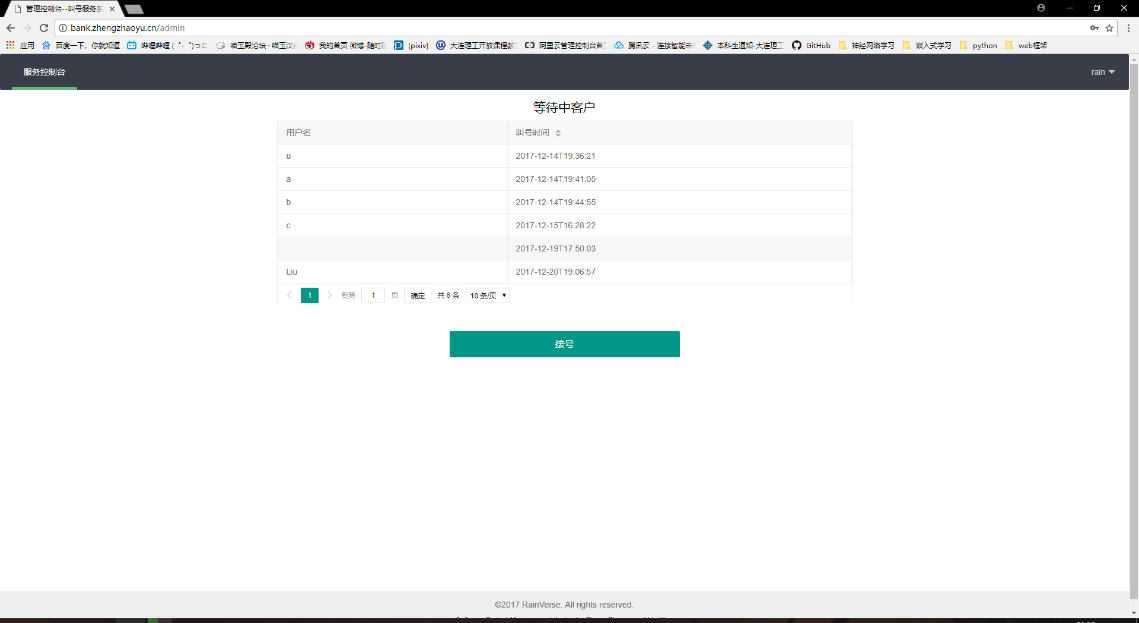
登录



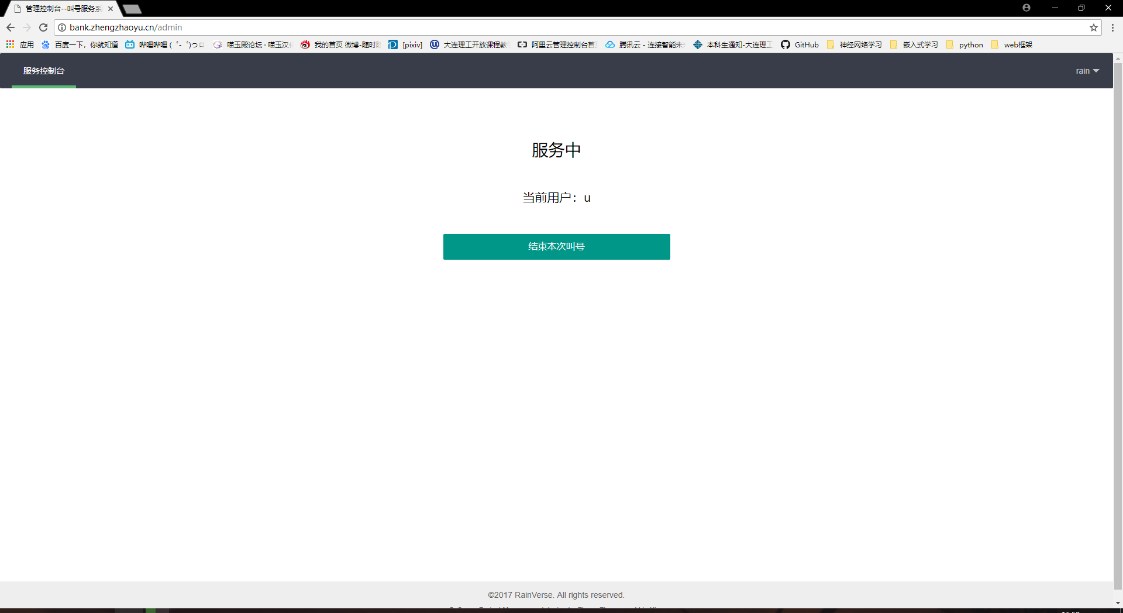
注册



接号及显示客户信息

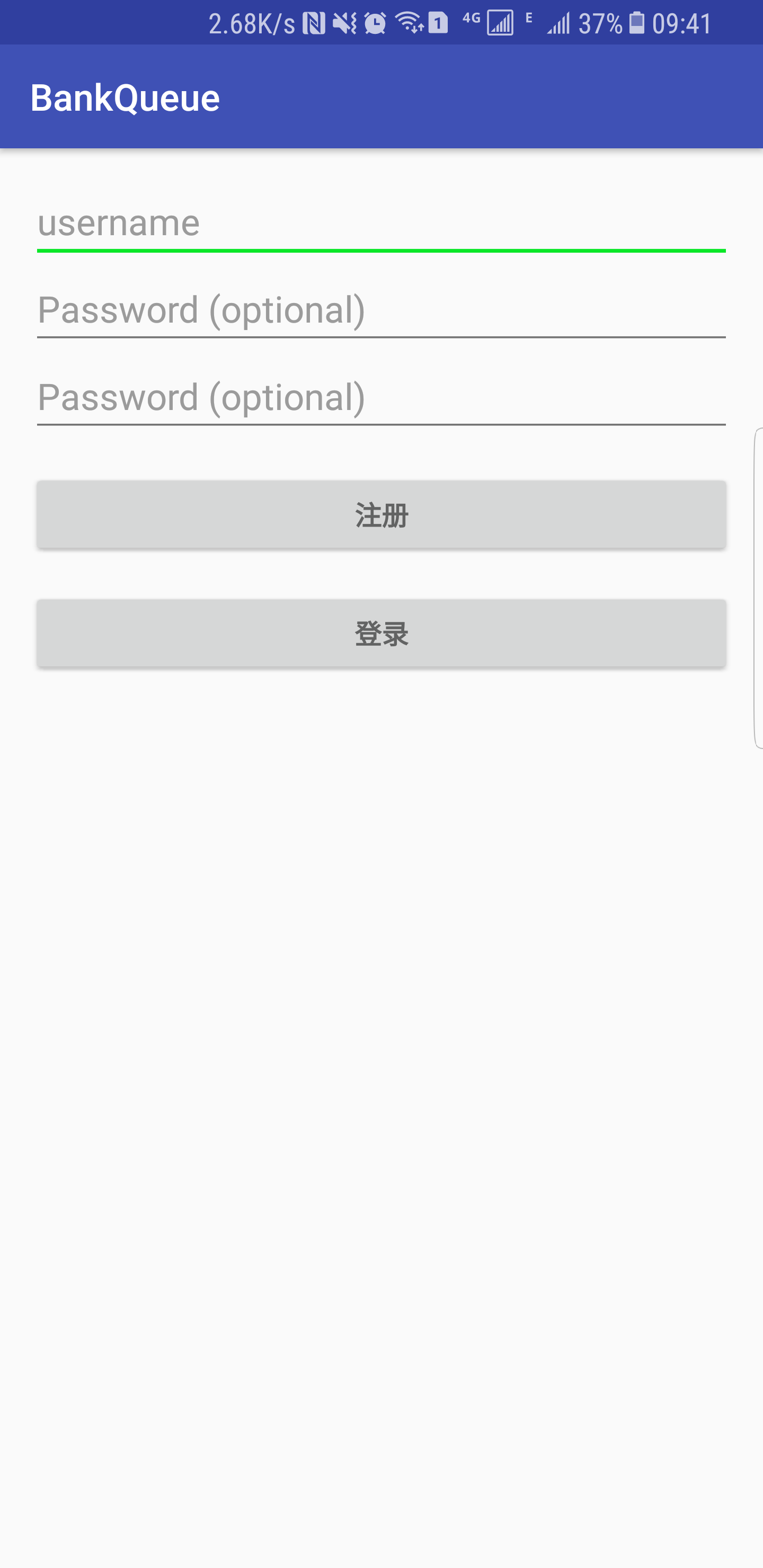
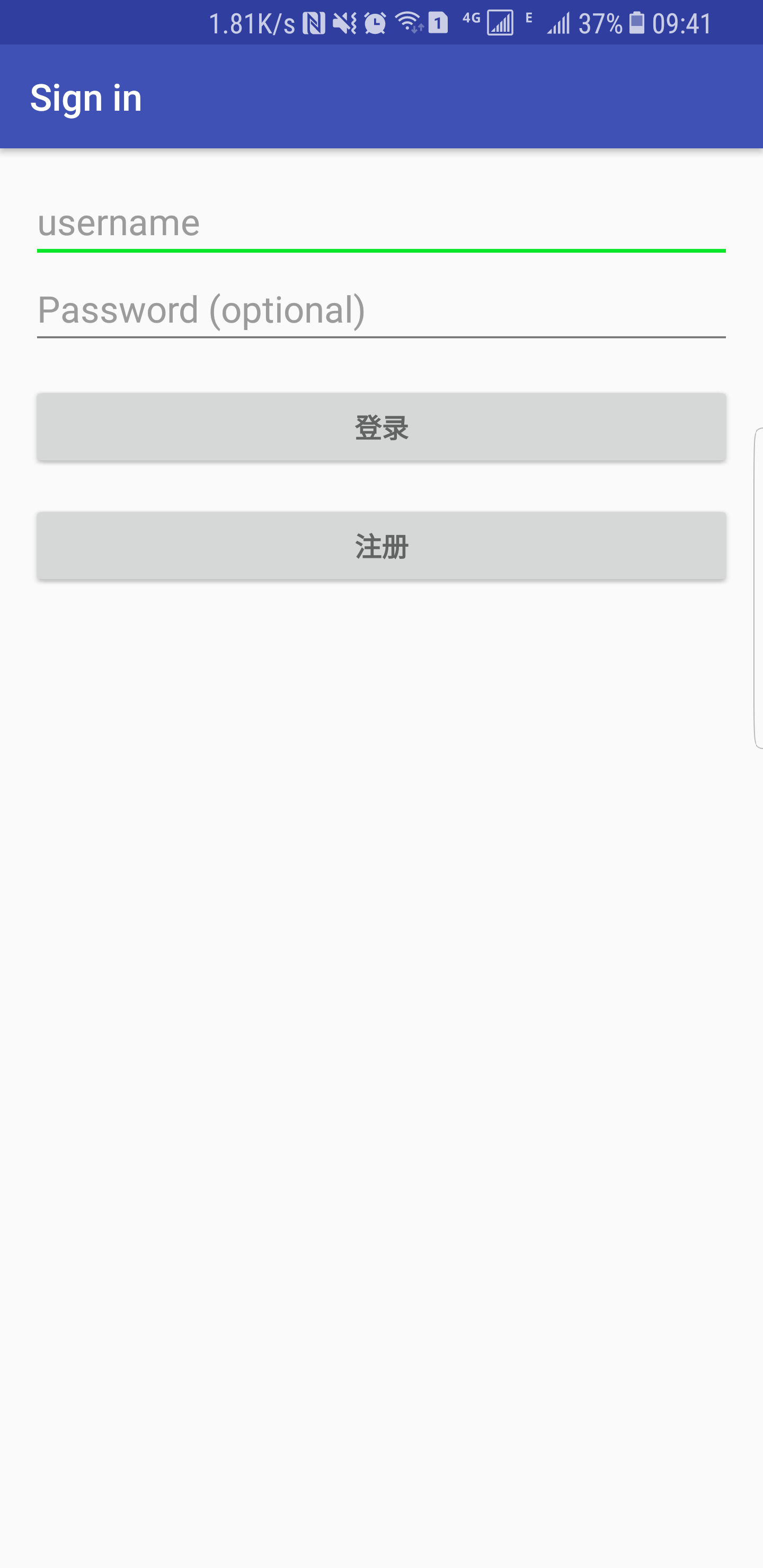


当前信息及结束接号

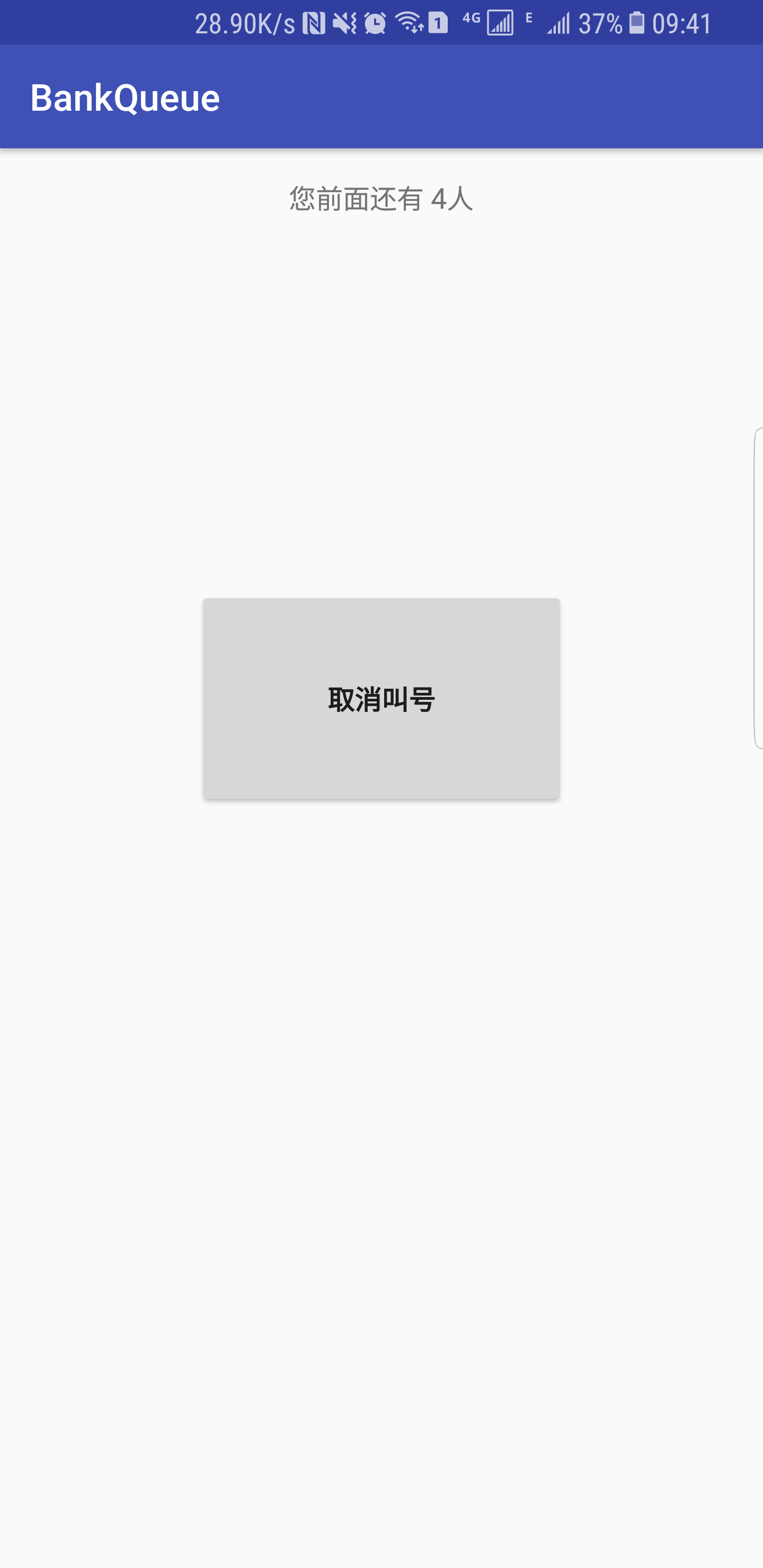
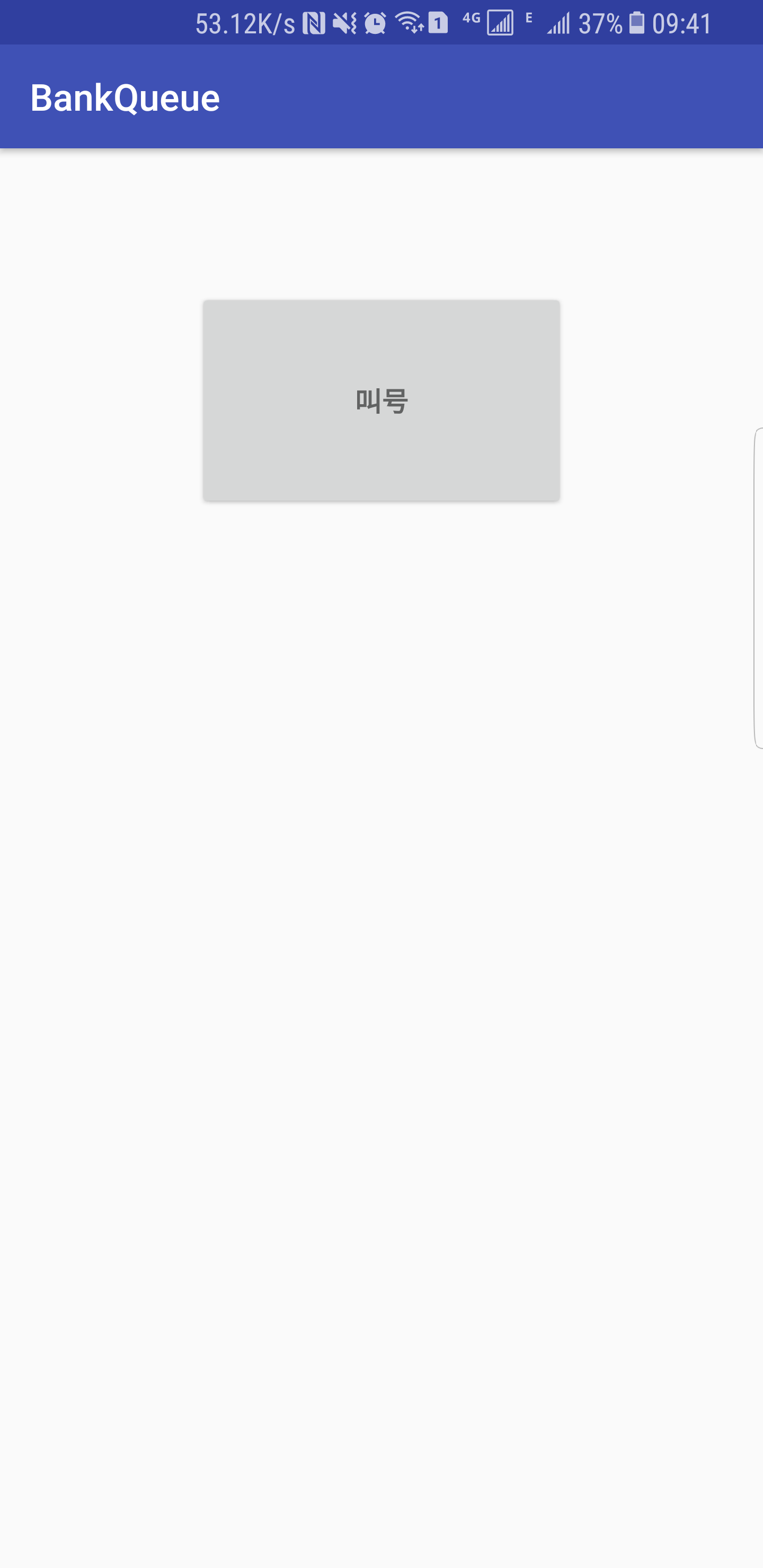


## 6.2.Android端运行截图

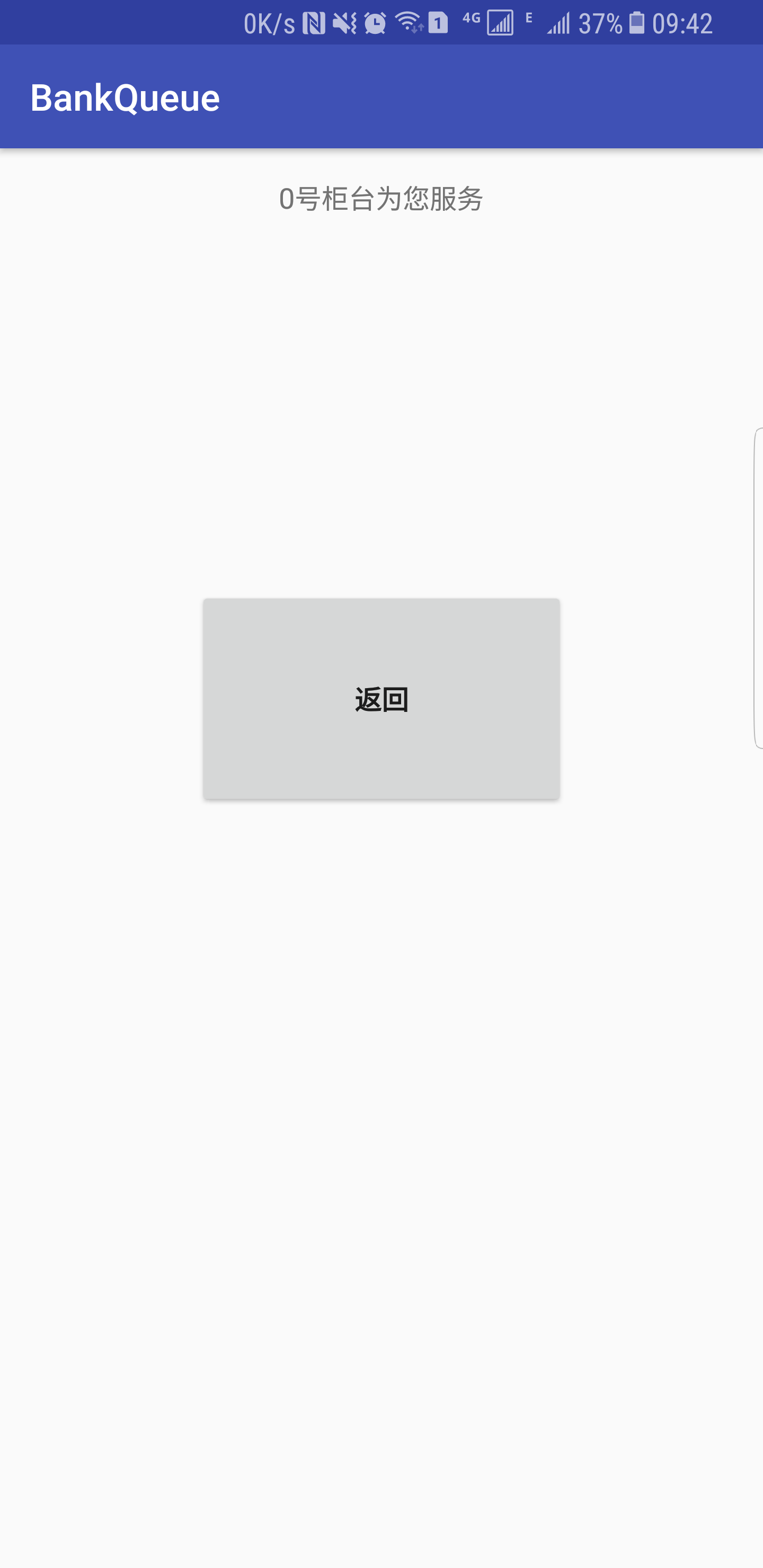
登录 注册



叫号 等待中



服务中



# 7.GitHub

本项目github地址：

web端及服务器端：

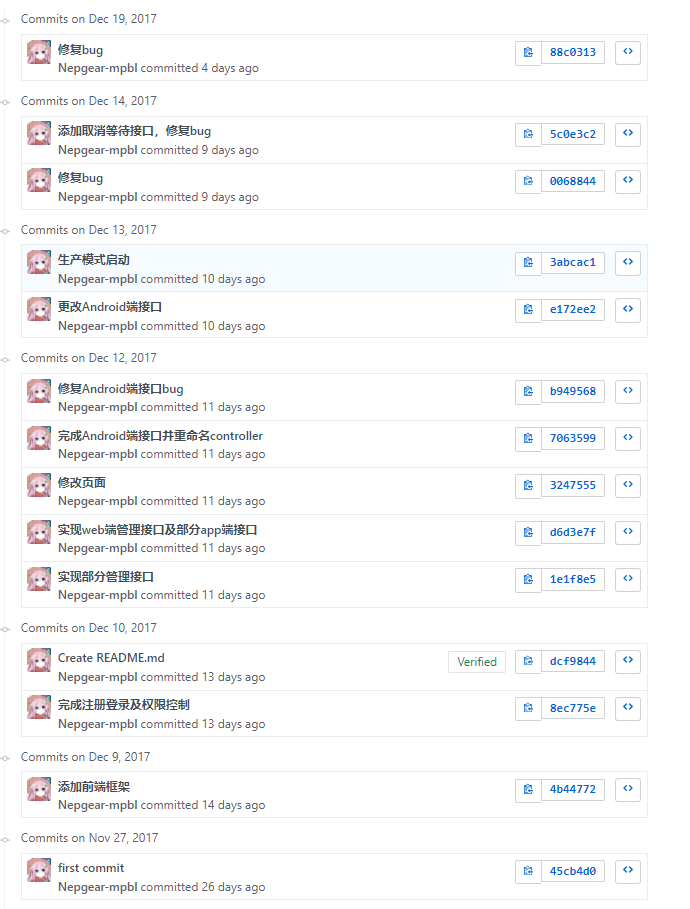
<https://github.com/Nepgear-mpbl/counter-system>

Android端：

<https://github.com/mcat1997/BankQueue>

本项目github均已开源，可随时查看源代码。

本项目commit记录截图见下图：Web端及服务器端



Android端见下图：

Android端

