



一、项目分工

学号	名字	角色	班级	职责	贡献
16340025	陈慕远	组长	下午班	负责项目 UI 界面的设计，以及对前端显示效果的优化，对项目功能的测试。	28%
16340023	陈明亮	组员	下午班	负责项目后端代码的编写，以及服务器 API 的文档编辑，对项目的功能测试。	28%
16340024	陈铭涛	组员	下午班	负责项目前端调用服务器 API 代码的编写，以及对项目服务器的提供和维护。	28%
16340007	蔡湘国	组员	下午班	负责项目 UI 的建议，实验报告的编写。	16%

二、开发环境

Windows10,visual studio2017

三、项目阐述

项目名称：**Know-It**

项目简介：**What is Know-it?**

1. 该 app 旨在让用户记录下他们想要和别人分享或讨论的有趣的事。在 Know It 这个 app 里面，我们可以注册账号，然后登陆，然后便可以发布一些你觉得有趣的事物，文字，图片，gif，以及视频，让你发现不一样的世界。

2. 该 app 设计中，我们有 5 个页面。首先，用户可以通过注册界面，填写账户的相应信息，然后成功注册用户。登录界面则允许用户通过用户名和密码登入应用的主页面。主页上有其他用户上传的所有帖子，我们可以浏览，查看其具体的内容，同时也可以点赞，也可以在帖子下面留言评论，所有点赞和评论的动作都会实时更新到帖子详情界面中。显然，我们在应用的主界面，以及帖子的详情界面中都使用了数据绑定的技术，采用与网络连接服务端的方法，将数据库中的内容实时更新到页面上。

3. 对于自适应 UI，用户可以通过扩展 app 窗口的宽度来探索聊天室页面。与此同时，当他们发现一些很棒的东西时，他们可以通过电子邮件与他人分享，从而创建应用到应用的交流。



4. 关于帖子内容，用户可以上传文字、图片和视频来组成他们的文章，所以这个 app 也有 MediaElement tech 和 File management。此外，第二页是聊天室页面，它使用网络访问技术，使用户可以从任何地方发送和接收消息。

5. 最后一个页面是用户信息页面，您可以更改帐户的名称、性别、头像和其他信息。最后点击确定修改按钮，上传修改后的个人账户信息。

知识点涉及：

- (1) Adaptive UI
- (2) Data Binding
- (3) Database
- (4) Network accessing
- (5) File management
- (6) MediaElement Tech
- (7) Live Tiles
- (8) App to app communication.

项目功能：

- ①网络型应用，可进行用户注册和登陆，支持用户信息修改，更改帐户的名称、头像和其他信息，信息传到数据库，通过网络访问。
- ②论坛型应用，可发帖子，内容可为文字，图片和视频，内容存储于数据库。
- ③可浏览主页上其他用户的帖子，观看视频，给心仪的帖子点赞，取消赞，评论等。
- ④自适应 UI 界面，不同用户在不同 PC 端同时在线后可在聊天室进行聊天。
- ⑤可以通过电子邮件与他人分享，达到 app 到 app 之间的交流。同时也能使用磁贴功能，方便地看到当前的所有帖子内容和更新动态。

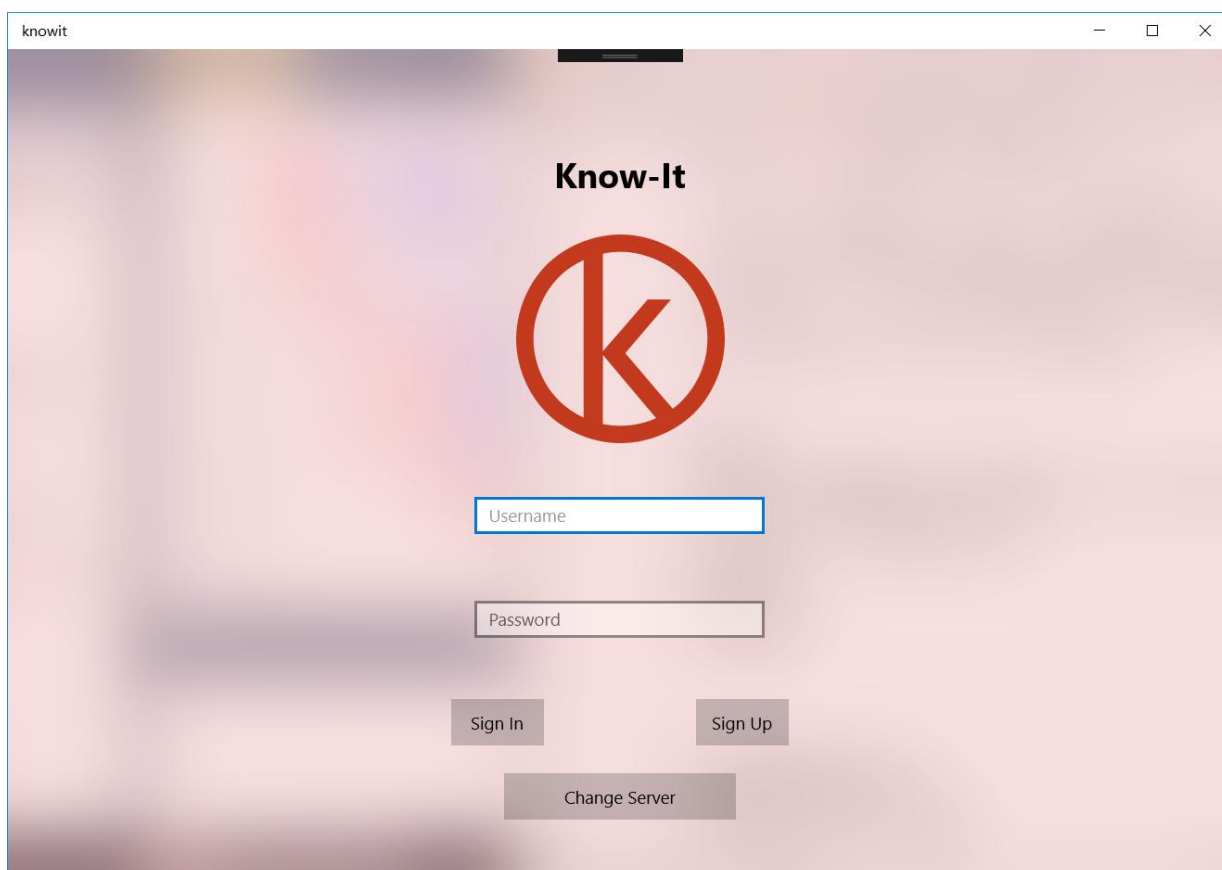


项目亮点：

- ① 网络型应用，只要有网，登陆后便能同步以往动态，包括各种信息。
- ② 聊天室的设计，让不同用户能实时交流。
- ③ 帖子源的多样性。支持文字图片，gif 图和视频加载。
- ④ 用户间的互动，点赞评论等。

四、项目展示

主界面注册登陆界面：



点击 Sign Up 跳转到用户注册页面



knowit

Sign Up

Avatar

Username

Password

Confirm Password

Submit Cancel

此处可选择头像，不选择为默认头像。用户名第一位必须为字母，后面必须为 5-17 位字母或数字；密码必须为 6 到 12 位数字或字母或下划线；邮箱格式不解释，手机大于 11 位数字。点击 submit 创建账号，cancel 取消。用户名不能重复，会弹出信息框提醒，填写信息不符格式也会有对应错误信息。

Sign Up

Avatar

Username

sd

Username Invalid!

Password

..

Password Invalid!

Confirm Password

..

Password Invalid!

Phone Number

s

Phone Number Invalid!

Email Address

×

Email Address Invalid!

Submit Cancel



knowit

Sign Up

Username

OK!

Password

OK!

Comfirm Password

OK!

Phone Number

OK!

Email Address

OK!

knowit

Sign Up

Username

OK!

Password

OK!

Success!

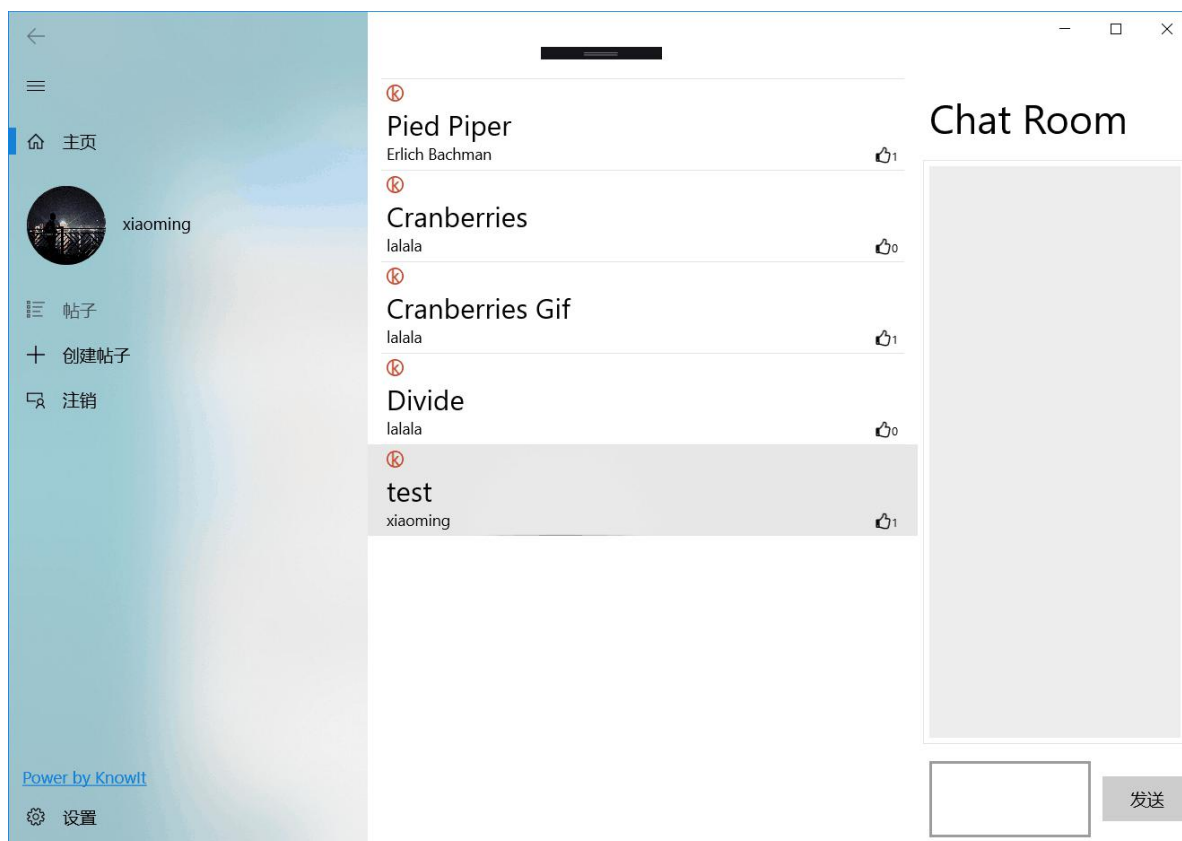
Phone Number

OK!

Email Address

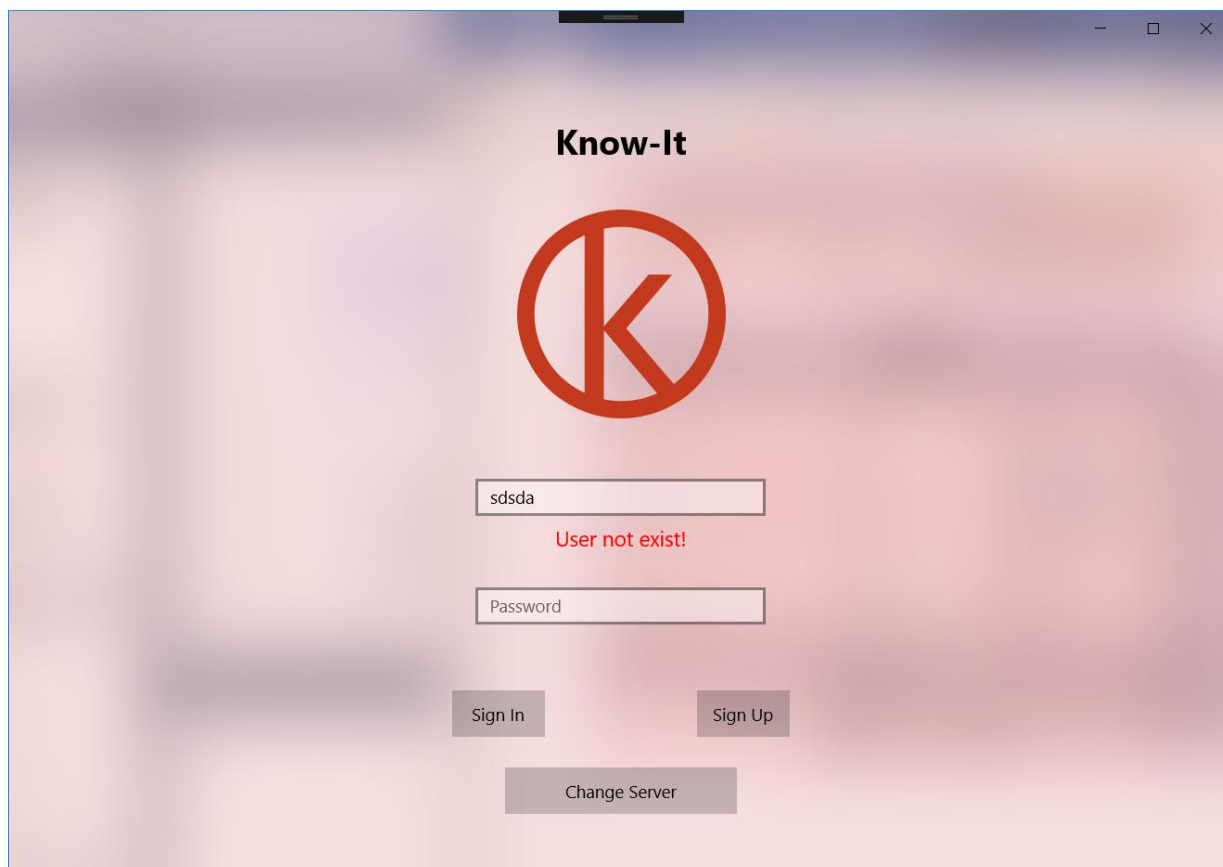
OK!

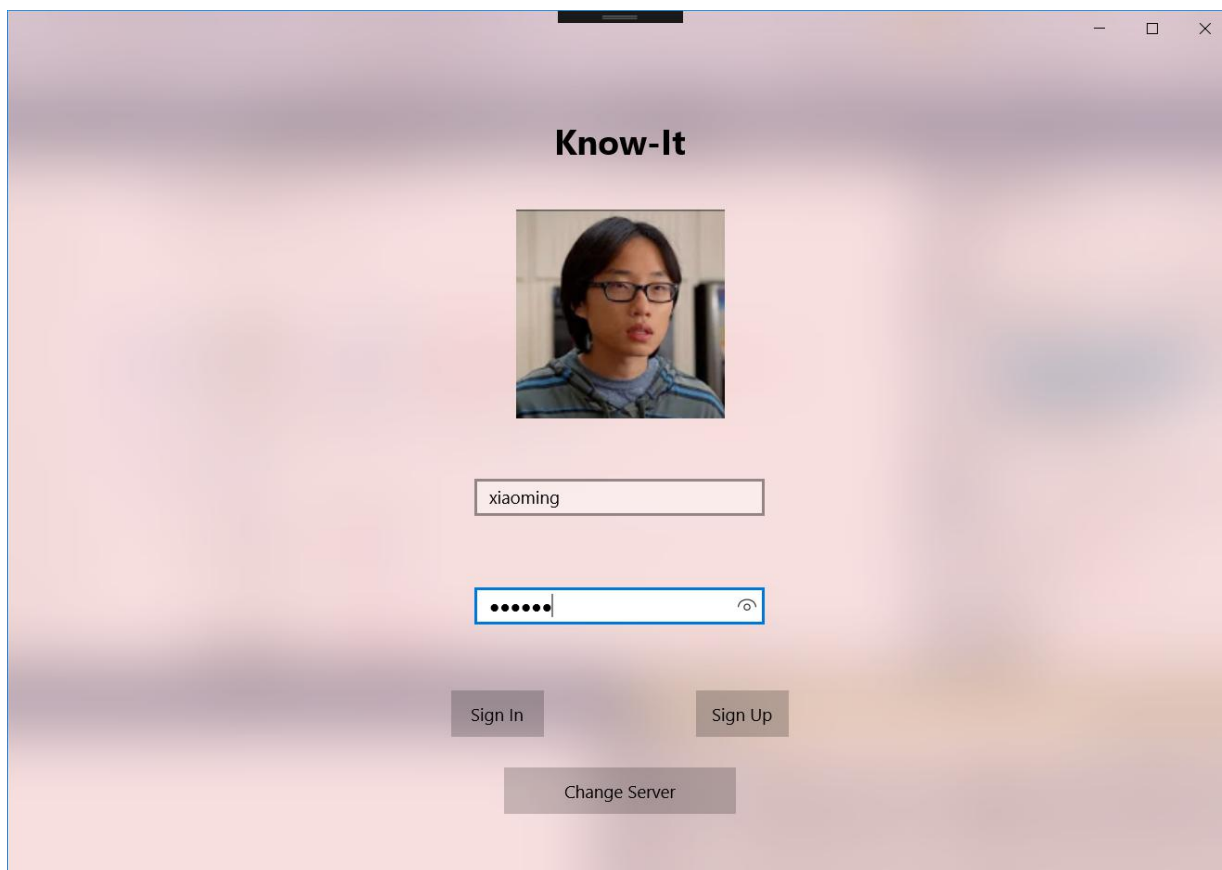
点击关闭后自动登陆。



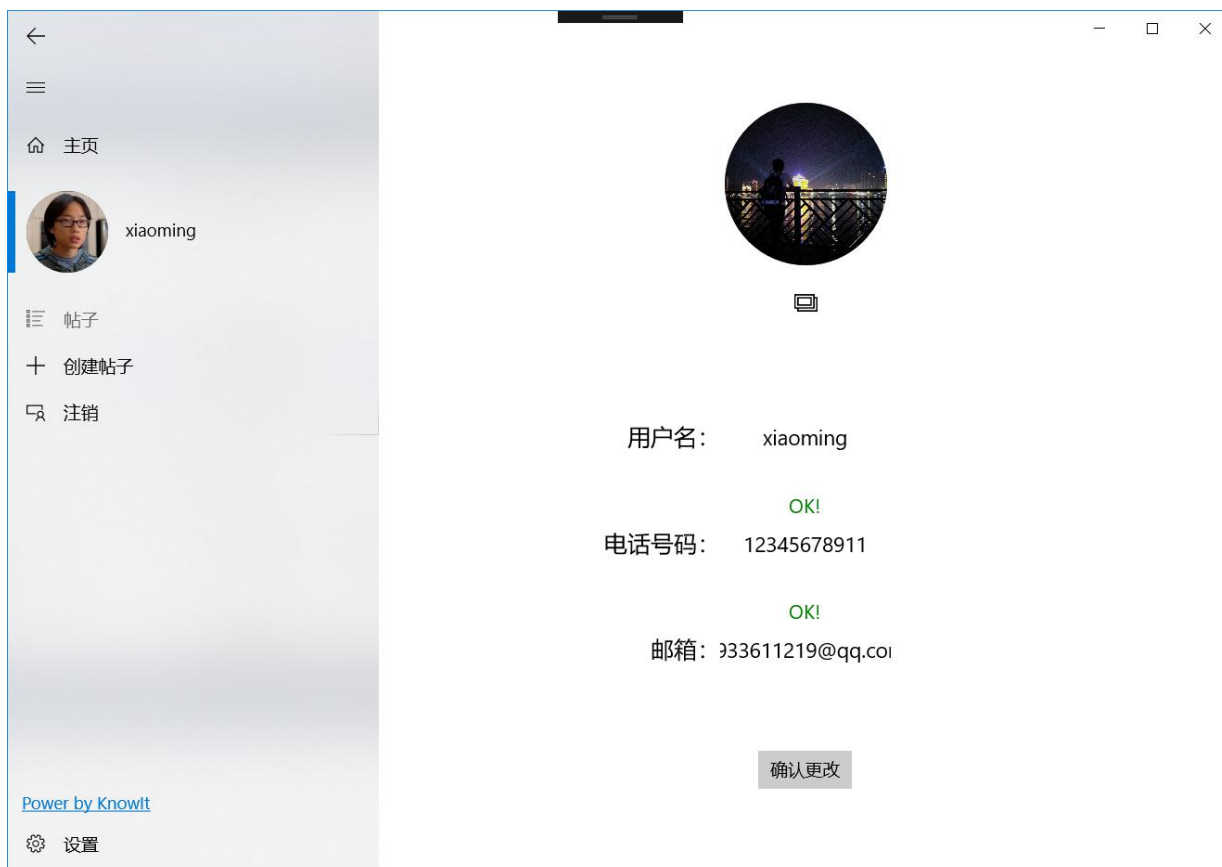
退出后重新登陆界面，点击 Sign in 进入上图主页，点击 Change Server 可切换服务器。

没有该用户的时候显示错误信息，查找到该用户瞬间加载其头像。





登陆成功后进入主界面，点击头像对应那栏可进行用户信息修改（点击对应项更改对应头像，电话号码，邮箱）

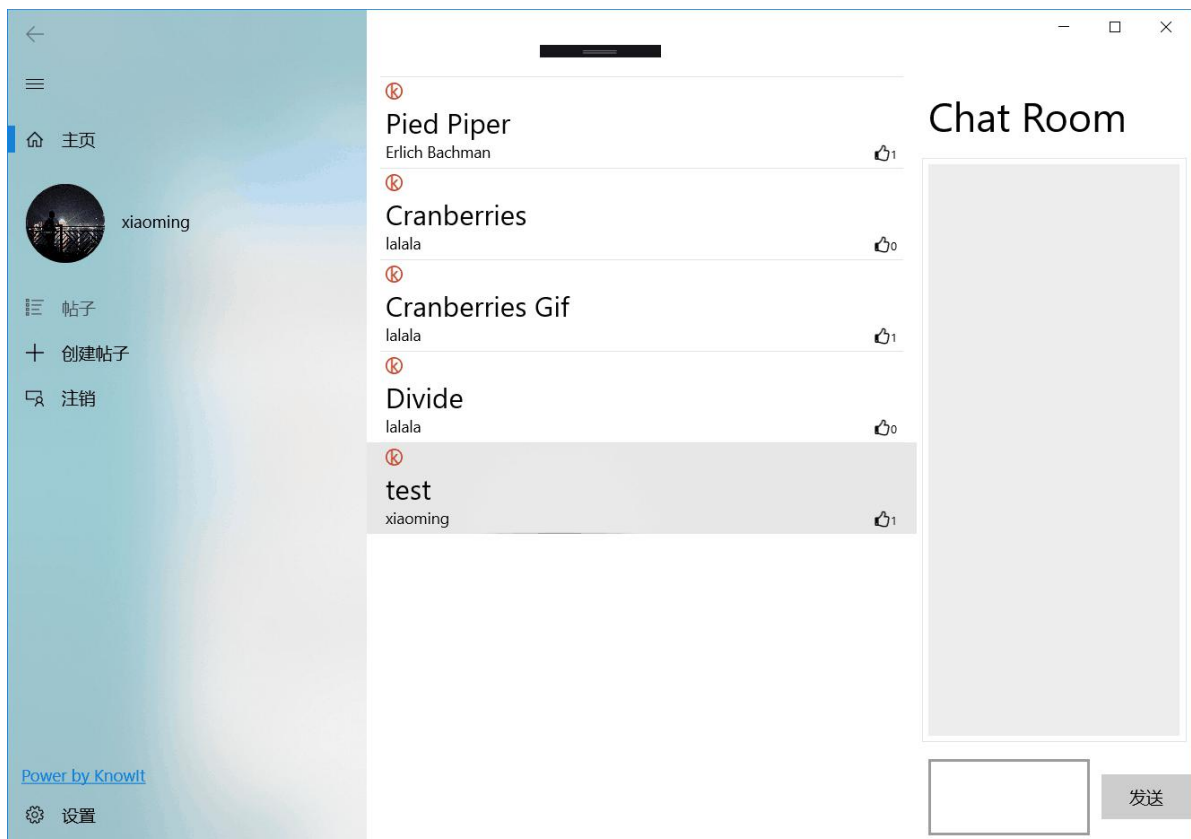
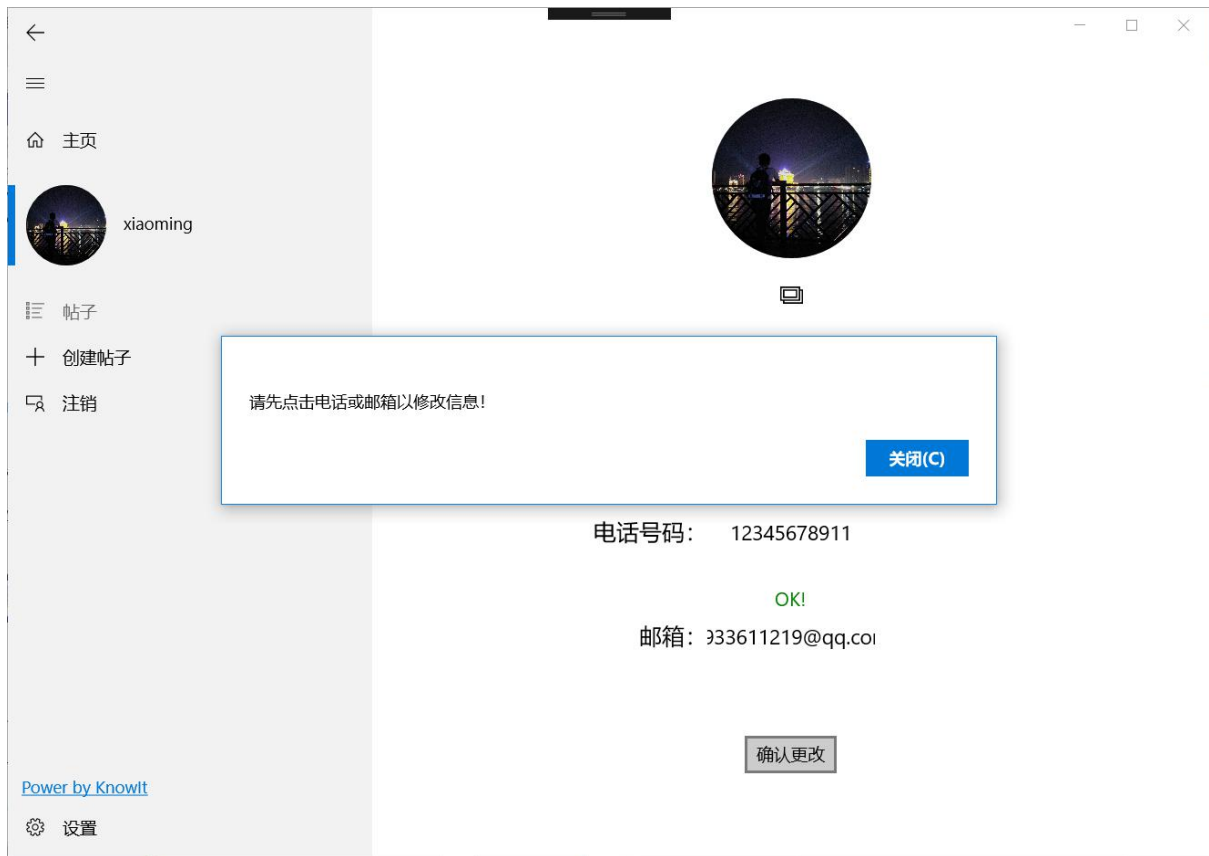


正确修改并点击确认修改后弹出对话框修改成功，然后跳回到主界面，同时更新修改信



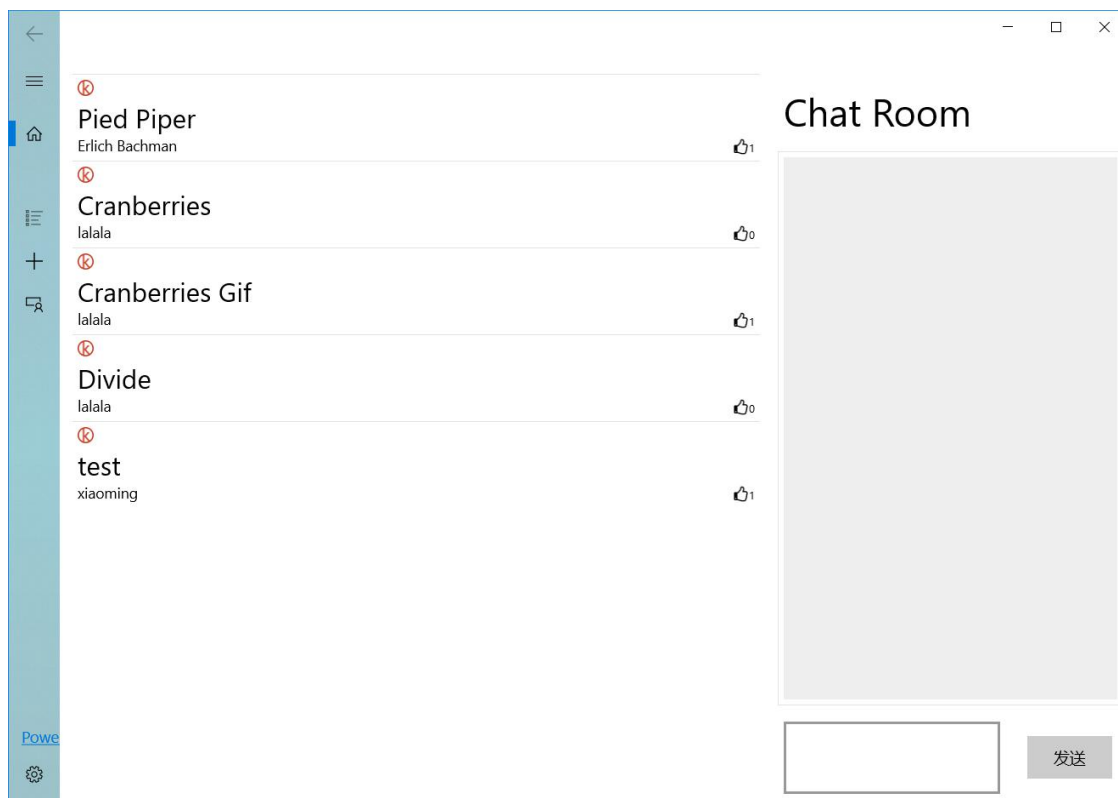
息。

如果未修改就点击确认修改按钮，弹出对话框提示。

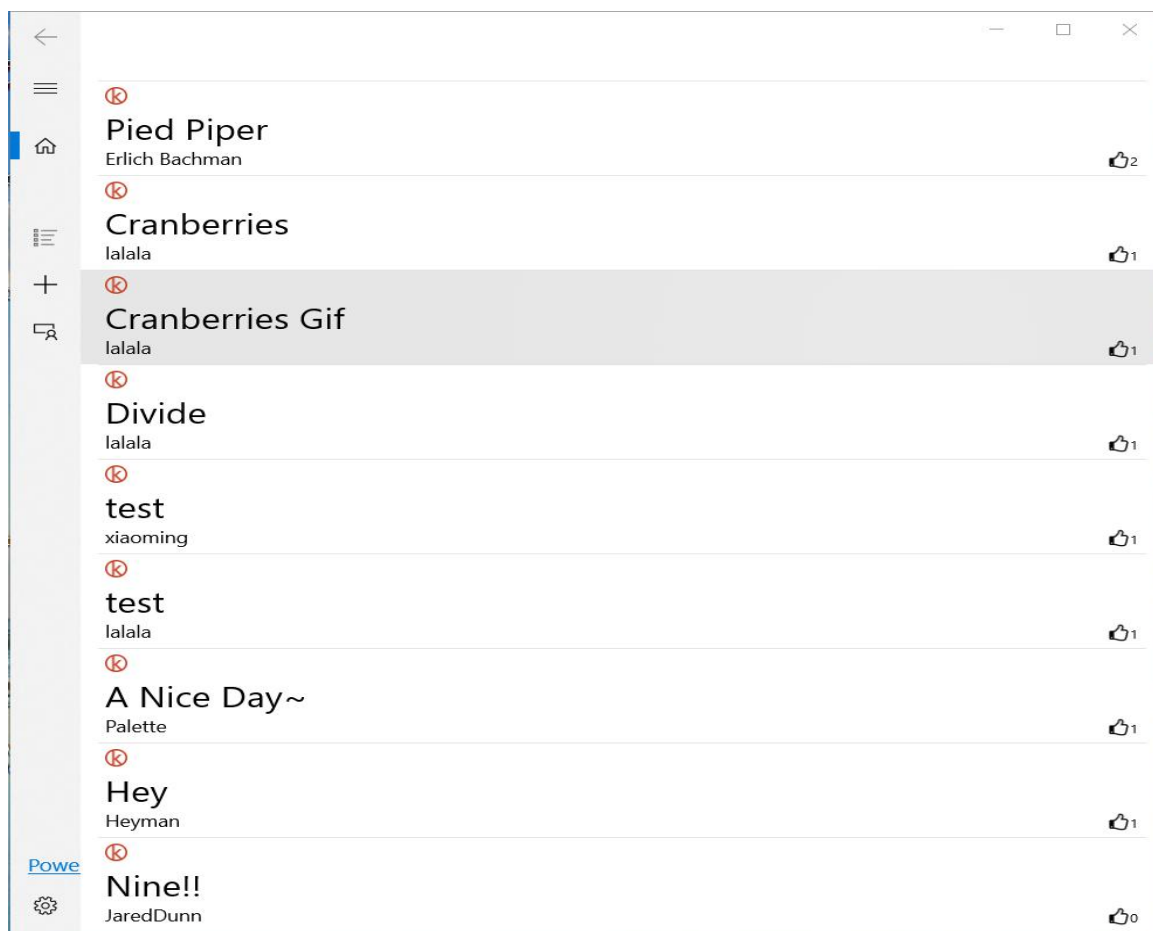




点击左上角菜单（关闭导航）

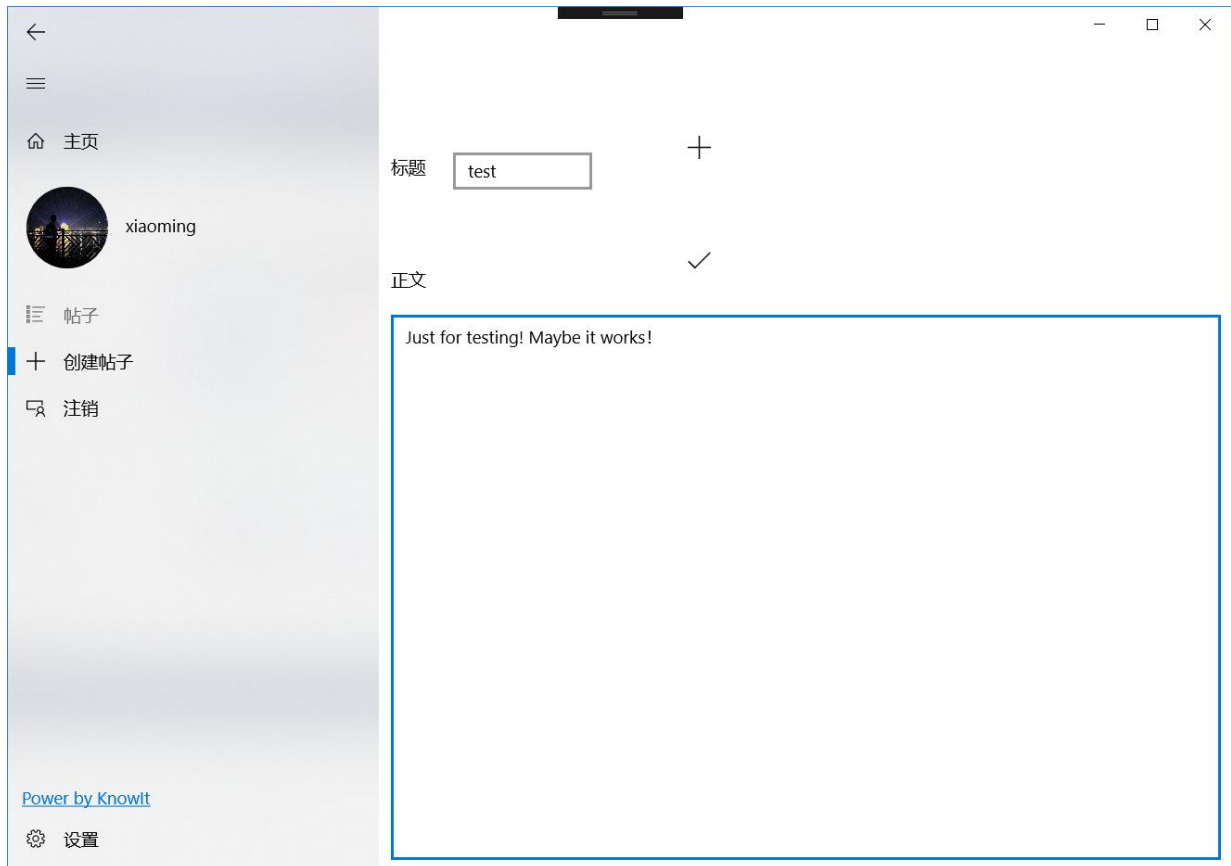


应用主界面的自适应功能，能够在缩小的时候折叠聊天室界面，以及左边的功能菜单栏。

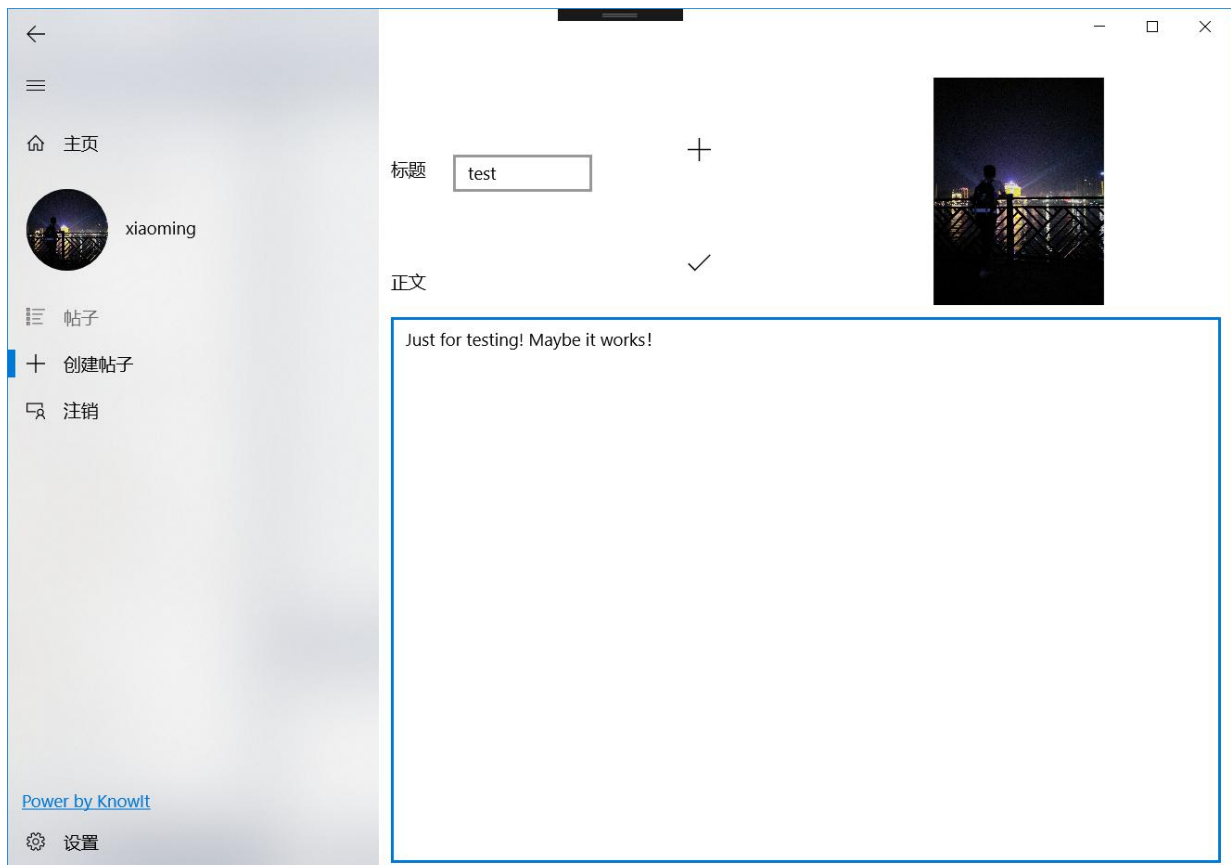




在上图界面或主界面点击+创建帖子。跳转到创建界面



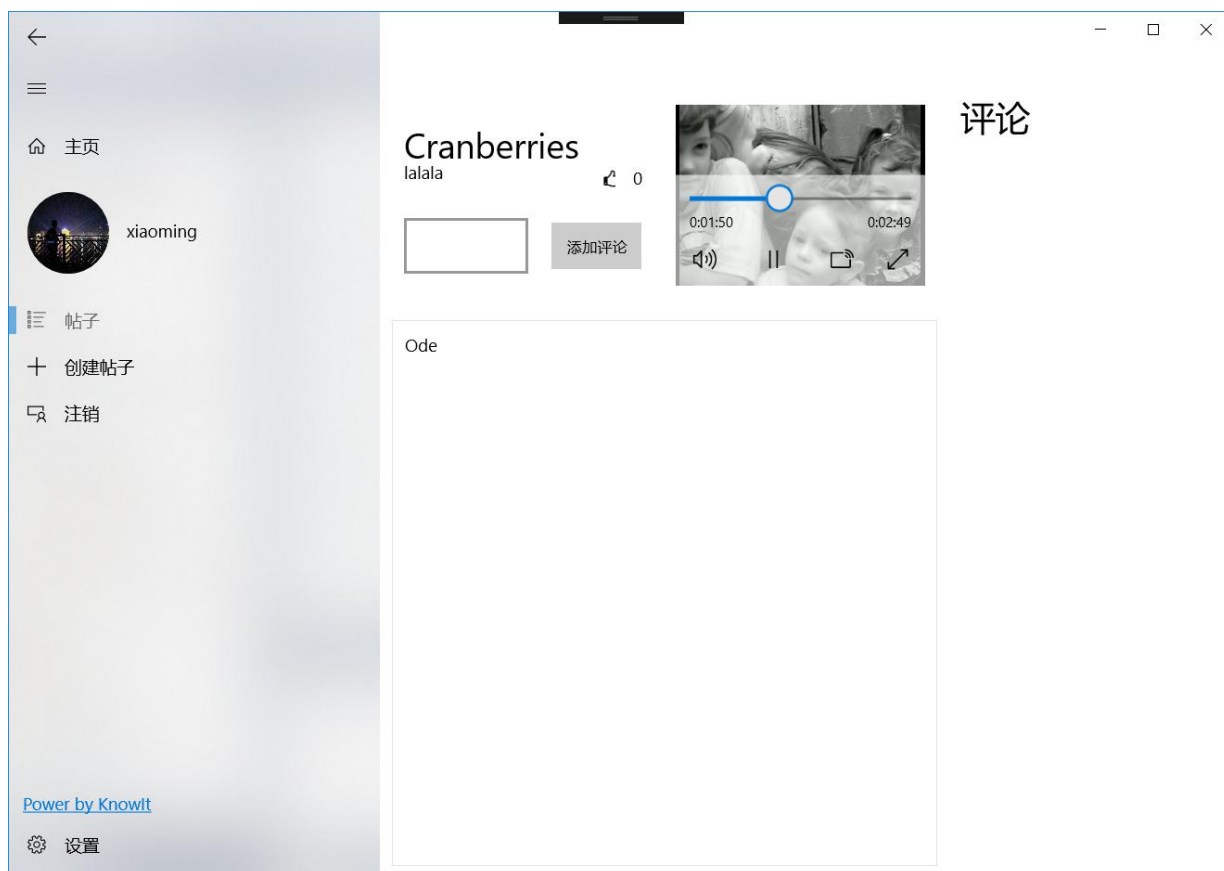
点击右上方+号可添加图片，gif 图和视频。



点击√创建，弹出创建成功对话框，点击确定之后跳转回主页面。



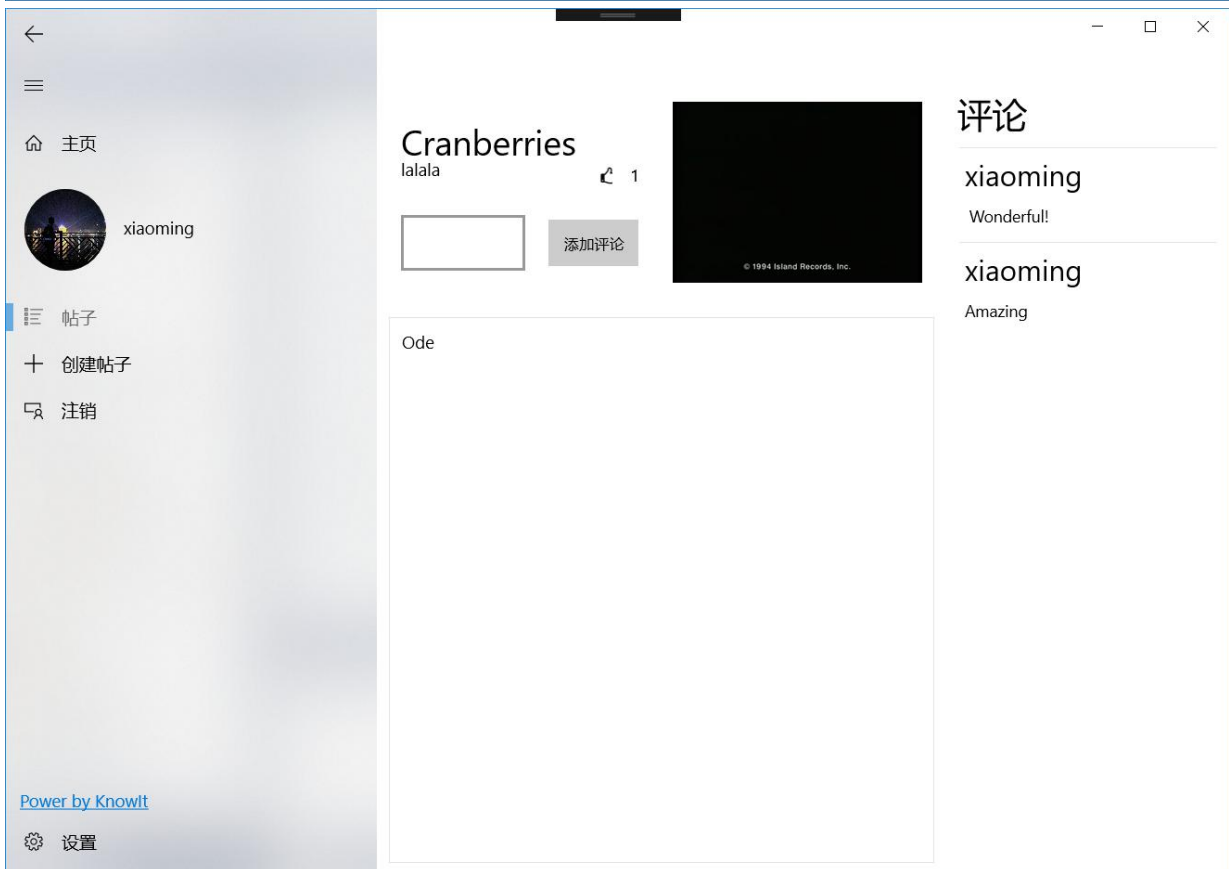
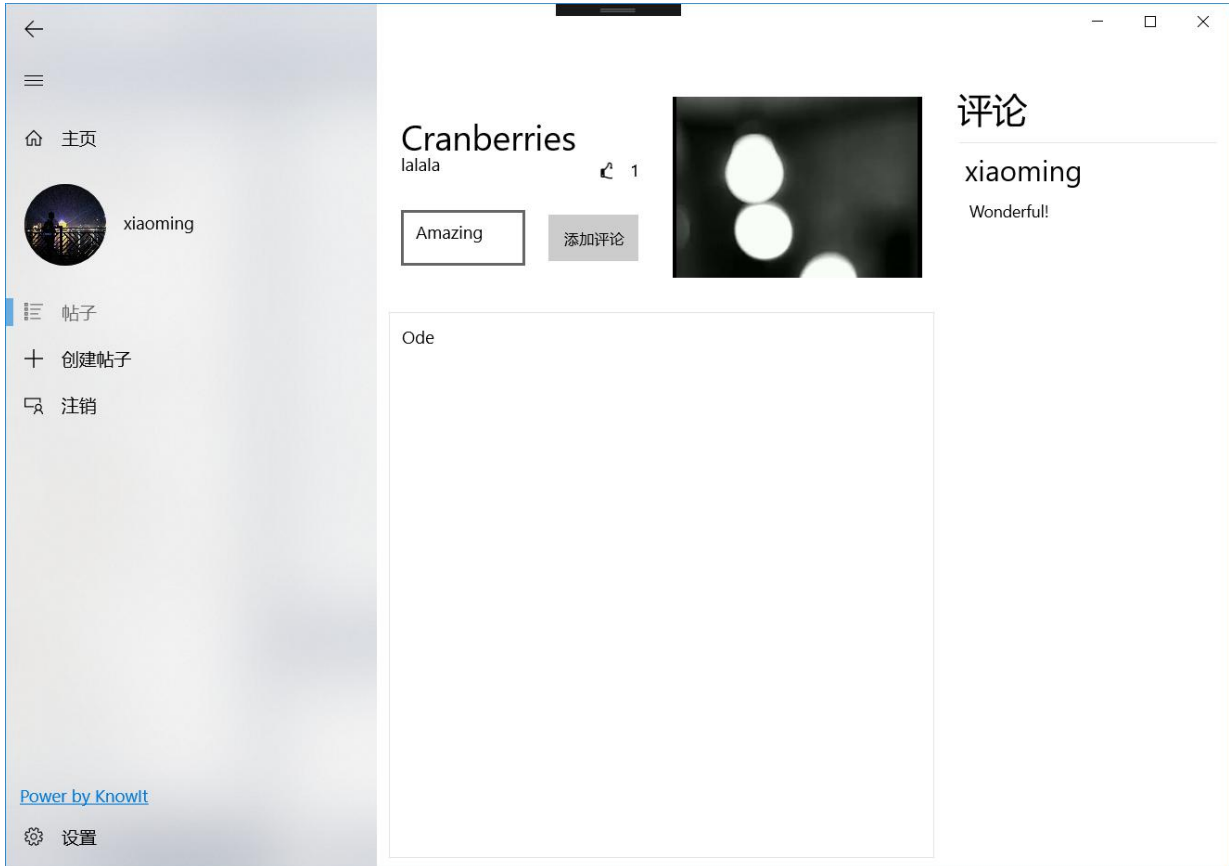
点击主界面的某个帖子，跳转到详情页。下图为带视频的帖子，视频支持拖动进度条同步，调节声音，暂停播放，全屏等操作。



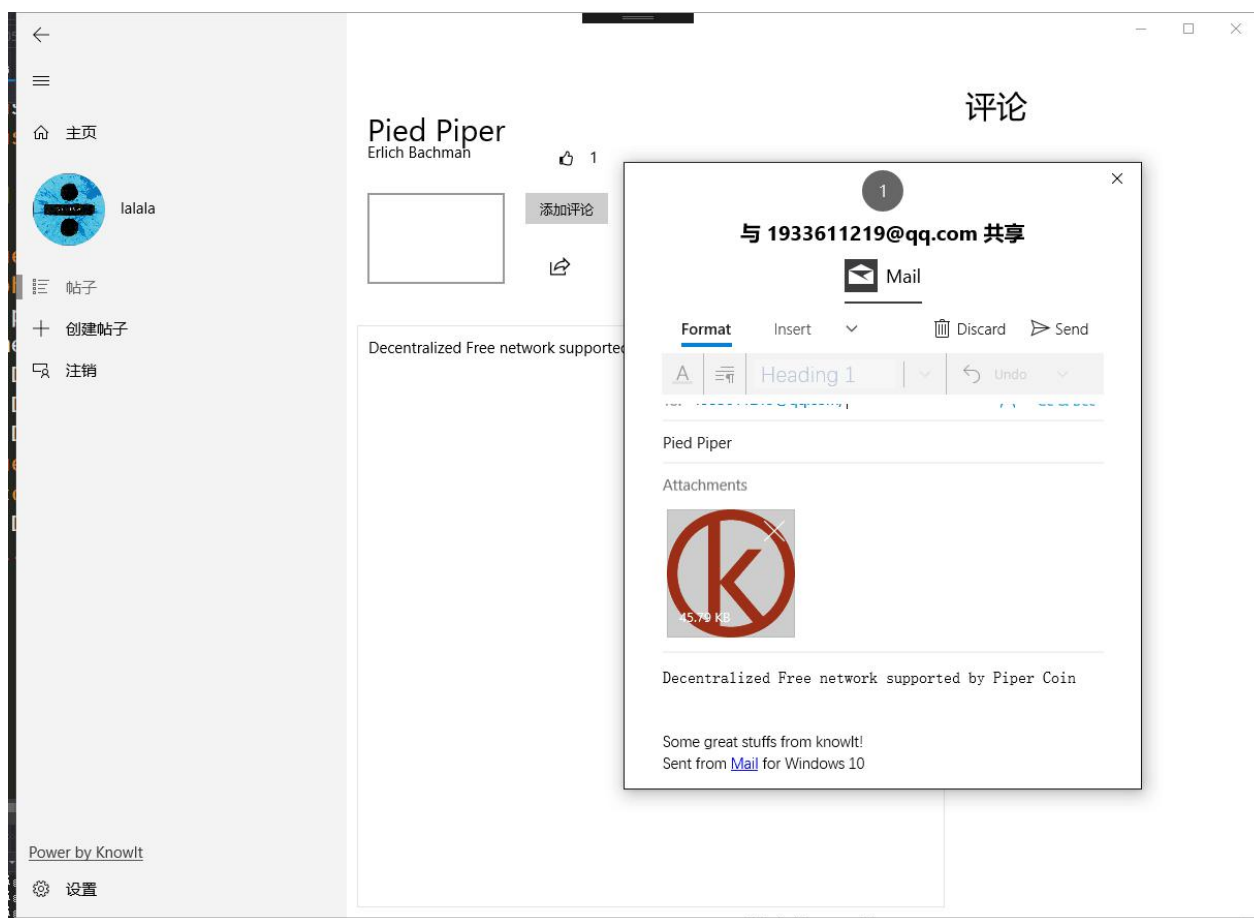
点击全屏：



可点击点赞按钮，点赞数加一，再次点击取消点赞。文本框内输入文字，点击添加评论，添加到右边评论界面。

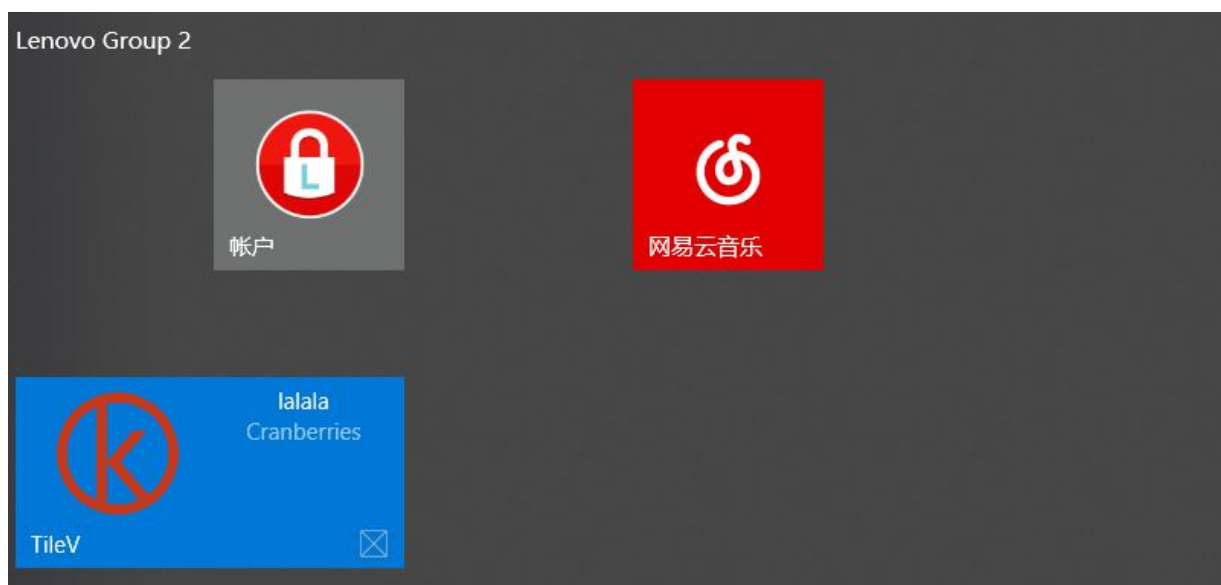


点击分享图标，可以发邮件方式实现 app 与 app 之间的交流。



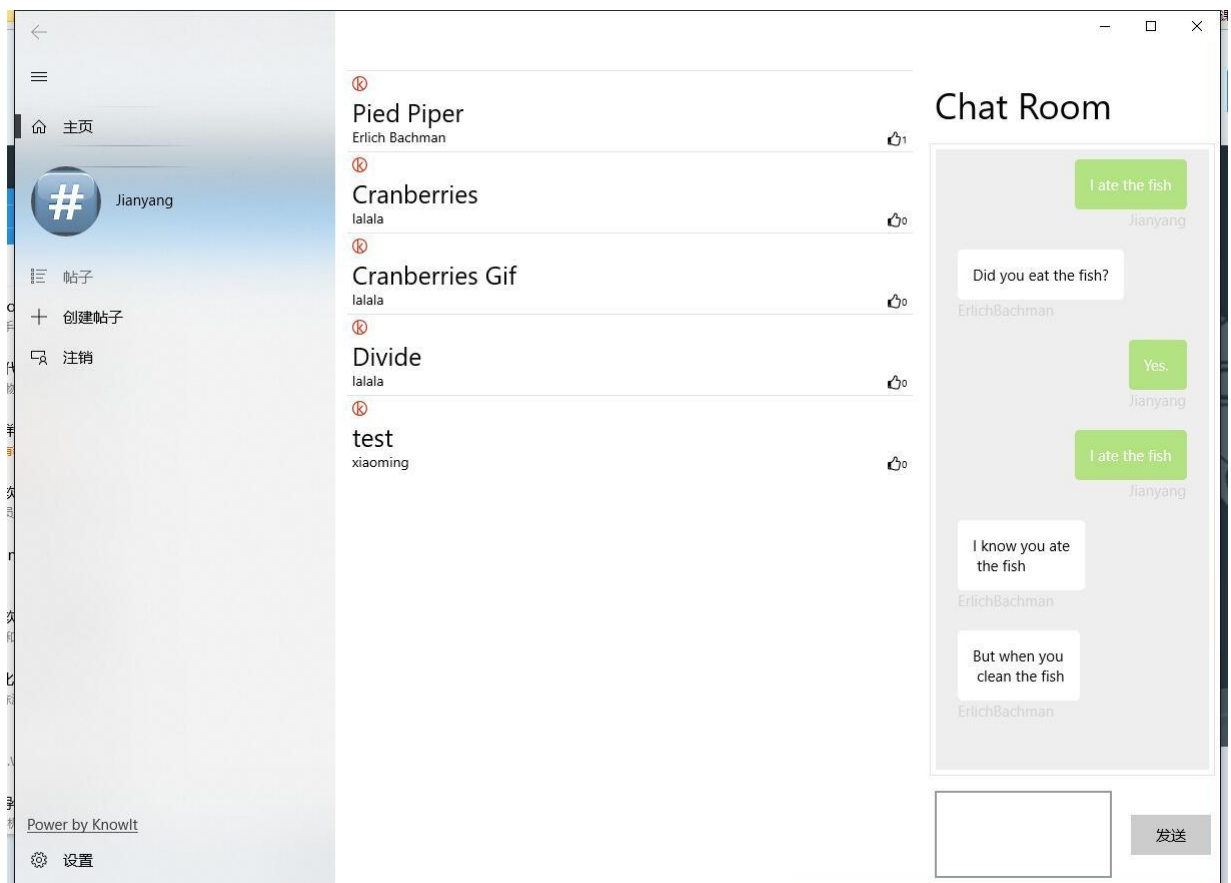
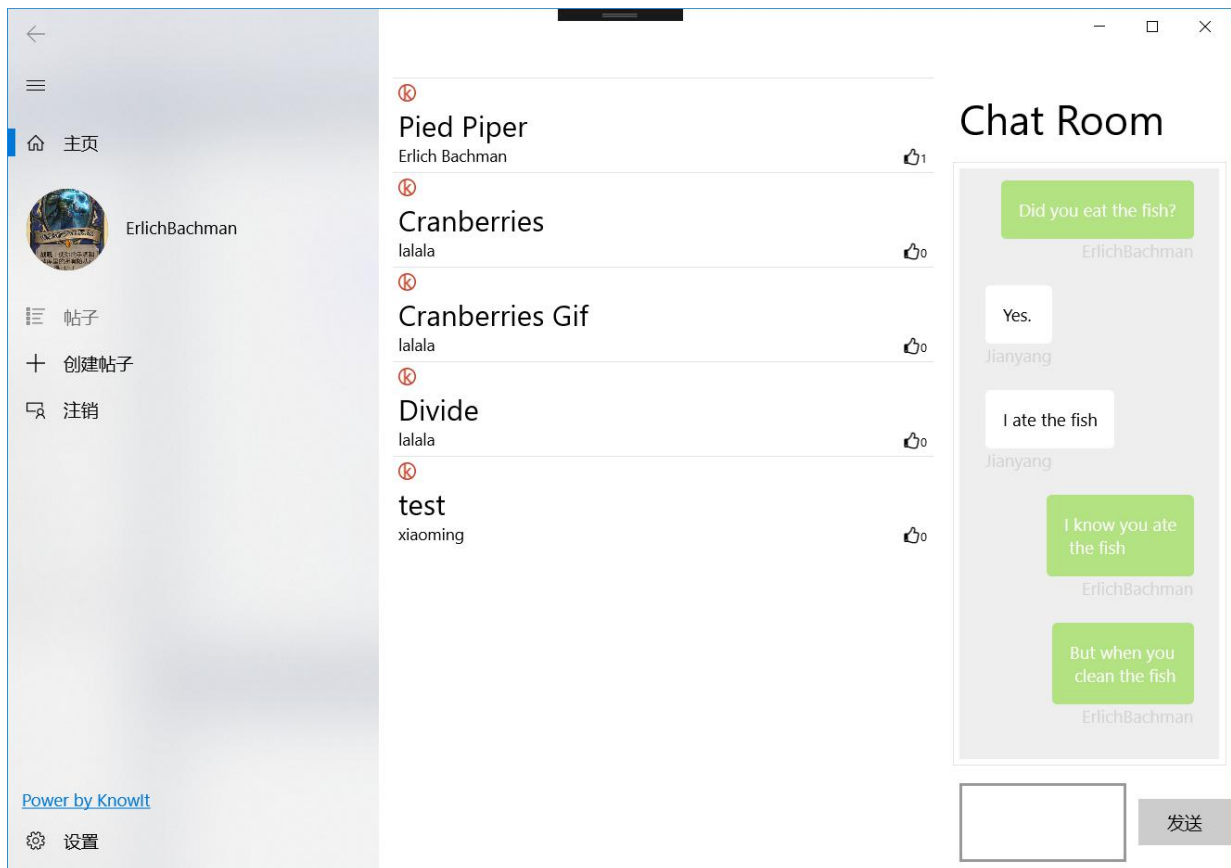
同时也可以通过点击主界面注销按钮，退出登陆，可以跳转回登陆注册界面。

下面是活动磁贴的部分截图，左下角即为该应用的磁贴，可以看到在实时地显示着帖子的主题和编辑者。





在主界面的聊天室下方文本框输入文字发送进行聊天，此功能需要两台机器进行测试。





五、项目难点及解决方案

1. 服务端与客户端之间的文件传输难题

问题描述：由于本次应用的实现，服务端采用了 NodeJS 语言，客户端采用了 C# 语言，关于这两种类型之间的数据传输毕竟是第一次接触，不免遇到来了较多的困难。其中最让人头疼的便是客户端程序发送图像，视频文件到服务端的问题。客户端每次传输媒体文件给服务端处理，并存储到对应服务器上时，总是由于编码问题，无法在服务端还原出原始正确文件。

解决方法：客户端 C# 程序在上传对应的媒体文件时，先将文件转换为二进制流文件再设置 HttpClient 的请求参数的头部类型，将其设置为“application/octet-stream”，连接服务端并将该数据传输过去。同时，服务端要相应地加入读取二进制流的模块，此处我们采用了 raw-body 模块解析二进制流文件，最终成功利用 NodeJS 特有的 Buffer 类新建了对应的媒体文件。

2. 服务器上的文件访问

问题描述：对于服务端的数据库存储原理，我们采用的是：数据库表上存储服务器文件地址，而将真正的文件放在服务器的对应目录路径下。客户端需要拿相应的媒体文件时，我们传输对应的存储路径，通过网络 Uri 访问对应文件。但是，在应用的编写过程中，有时候还是存在无法正确访问对应文件的时候。

解决方案：我们新建对应的存储文件夹，利用 Nginx 的文件管理系统，对文件的访问申请进行检查和分配，排除非法的文件申请，最终允许对应的媒体文件正确访问。

3. 聊天室发出消息后，另一端消息不同步。

问题描述：本应用的聊天室采用 WebSocket 技术实时发送对应的用户聊天信息，但



在最初的实现中会出现两个用户无法互相发送消息的情况。

解决方案：该问题原因是服务端没有保存所有连接，只保存了最新登陆的用户连接，所以只需在服务端将现存的未关闭的 WebSocket 连接放到一个数组中保存起来，给所有的连接都发送最新接收到的消息即可。

4. 观看视频后退回到主界面，视频在返回后仍在播放。

问题描述：在查看具有视频文件的帖子详情页面之后，发现返回主界面时，该媒体文件还在继续地播放着。

解决方案：对于这种情况，我们只需要对播放媒体源进行处理即可。实际上既可以把对于的播放器 Source 设为 NULL，也可以显式地调用 C# 的析构函数，对当前占用内存，还在播放的媒体进行回收。

六、项目总结

这次项目，从构思，分工，到实现，整合，测试，修复 bug，再测试，继续修复 bug 的循环中暂时告一段落，项目完成过程中有遇到各种各样的问题，前端和后端各自出现的 Bug，在各自完成对应部分，相互协助修复后进行合并，又会出现其他新的问题。开始测试，断点调试，确保自己负责部分没问题之后定位错误在谁负责的哪个位置发生，查找相应资料进行解决，主要是发生在前端 UI 和后端代码的衔接，客户端与服务端之间的问题。但总算是都逐一解决了。

构思部分：我们要做一个怎样的应用，要有什么样的功能，满足怎样的需求？生活还有诗和远方，要做一个有趣的程序员，所以之后决定做简易论坛，能够分享生活中有趣的事物，无论是文字，图片还是视频，在这个平台上你都能和其他人分享。说不定就遇上有趣的灵魂了。同时还能和其他人积极互动，评论点赞，不局限于自己生活中



狭窄的圈子。

实现部分：有了 idea，要怎样去实现？要考虑服务端和客户端之间同步，前端和后端的实现与连接问题。前端后端分工，到实现，整合，测试，修复 bug，再测试，继续修复 bug，遇到问题寻找问题所在，查找对应的解决方法，完成这个项目循环了好多次。时常的对话就是，“我这边发了个帖子，你数据库那边有记录没有？”，“聊天室我这边发了个消息，你那边有显示没有”，“嗯，没有”，断点调试，我这边发出去了，是服务端的问题，然后找问题修改再循环。直至项目完成。

后期完善和修改：做出大致功能后要考虑对应的细节，比如用户名电话号码等的格式，比如界面的美观性问题，要在原来的基础上进行一定的调整，同时进行不同的测试，不同人测试的结果往往不一样，所以往往会发现不同的问题，再进一步完善。

总而言之，这个项目是团队齐心协力的结果。