**editOL**

——多人实时协作云端办公工具

|  |  |  |
| --- | --- | --- |
| 文件状态：  　[ ]草稿  　[√]正式发布  　[　]正在修改 | 项目名称： | editOL |
| 当前版本： | 1.0 |
| 完成日期： | 2017-06-12 |
|  |  |
|  |  |

目录

[1. 项目介绍 2](#_Toc484981086)

[1.1 项目简介 2](#_Toc484981087)

[1.2 项目背景 2](#_Toc484981088)

[2. 页面 3](#_Toc484981089)

[2.1 登陆注册 3](#_Toc484981090)

[2.2 文件列表 3](#_Toc484981091)

[2.3 文档编辑 4](#_Toc484981092)

[3. 功能介绍 4](#_Toc484981093)

[3.1 登陆注册模块 4](#_Toc484981094)

[3.2 文件管理模块 4](#_Toc484981095)

[3.3 文本编辑管理模块 5](#_Toc484981096)

[3.4 群聊模块 5](#_Toc484981097)

[4. 总体设计 5](#_Toc484981098)

[4.1 前端 6](#_Toc484981099)

[4.1.1 运行环境 6](#_Toc484981100)

[4.1.2 开发环境 6](#_Toc484981101)

[4.1.3 技术路线 6](#_Toc484981102)

[4.2 服务端 6](#_Toc484981103)

[4.2.1 运行环境 6](#_Toc484981104)

[4.2.2 开发环境 6](#_Toc484981105)

[4.2.3 数据存储 6](#_Toc484981106)

[4.2.4 技术路线 6](#_Toc484981107)

[5. 技术难点 7](#_Toc484981108)

[5.1 实现群聊 7](#_Toc484981109)

[5.2 操作提取 7](#_Toc484981110)

[5.3 结果接收 7](#_Toc484981111)

# 项目介绍

## 项目简介

editOL是一款多人实时协作云端办公工具。顾名思义就是一个在线的记事本，特点是能进行多人实时编辑,从而完成、方案讨论、会议记录和 资料共享等工作。

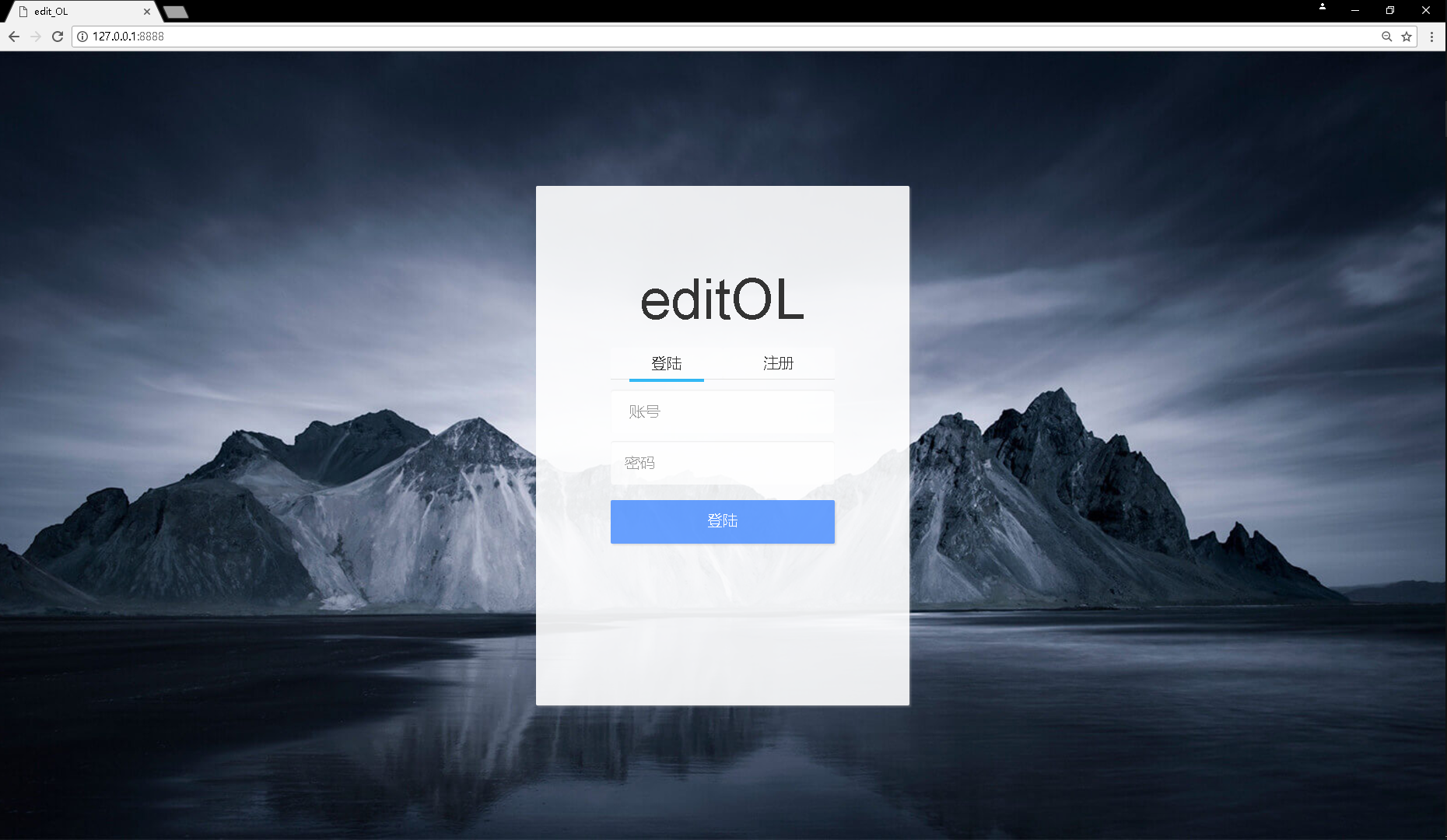
## 项目背景

思路来源于Google Docs， Google Docs 可以进行多人在线编辑，支持多种文本格式，具有还原历史版本等的功能。

平时在本地进行文档编辑或代码编写时，轮流编写会浪费时间，版本控制工具仍会产生修改冲突，因此想知道多人在线编辑的原理，有没有可能运用在版本控制器上，或者说让在线编辑支持更多的格式。

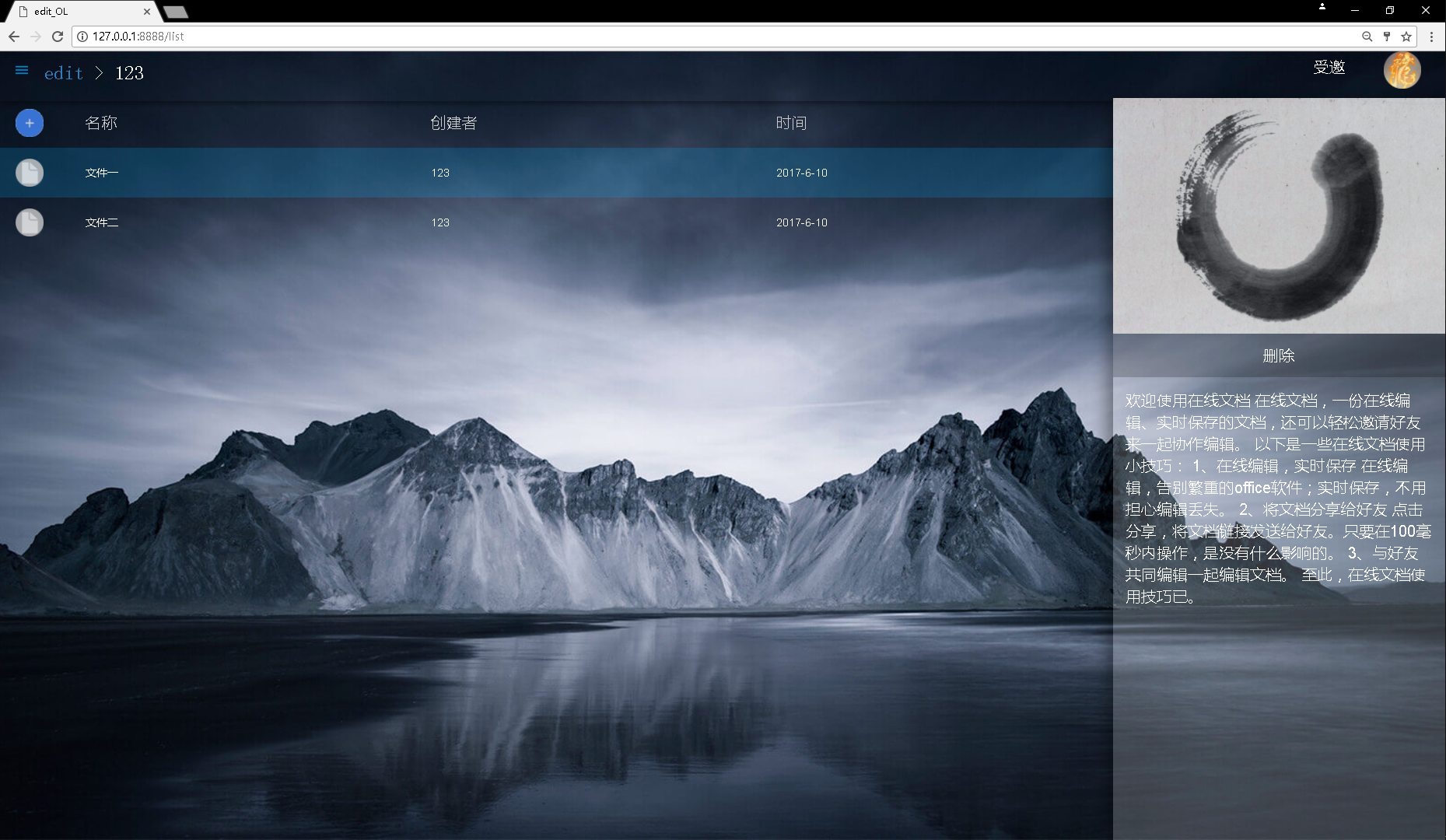
# 页面

## 登陆注册



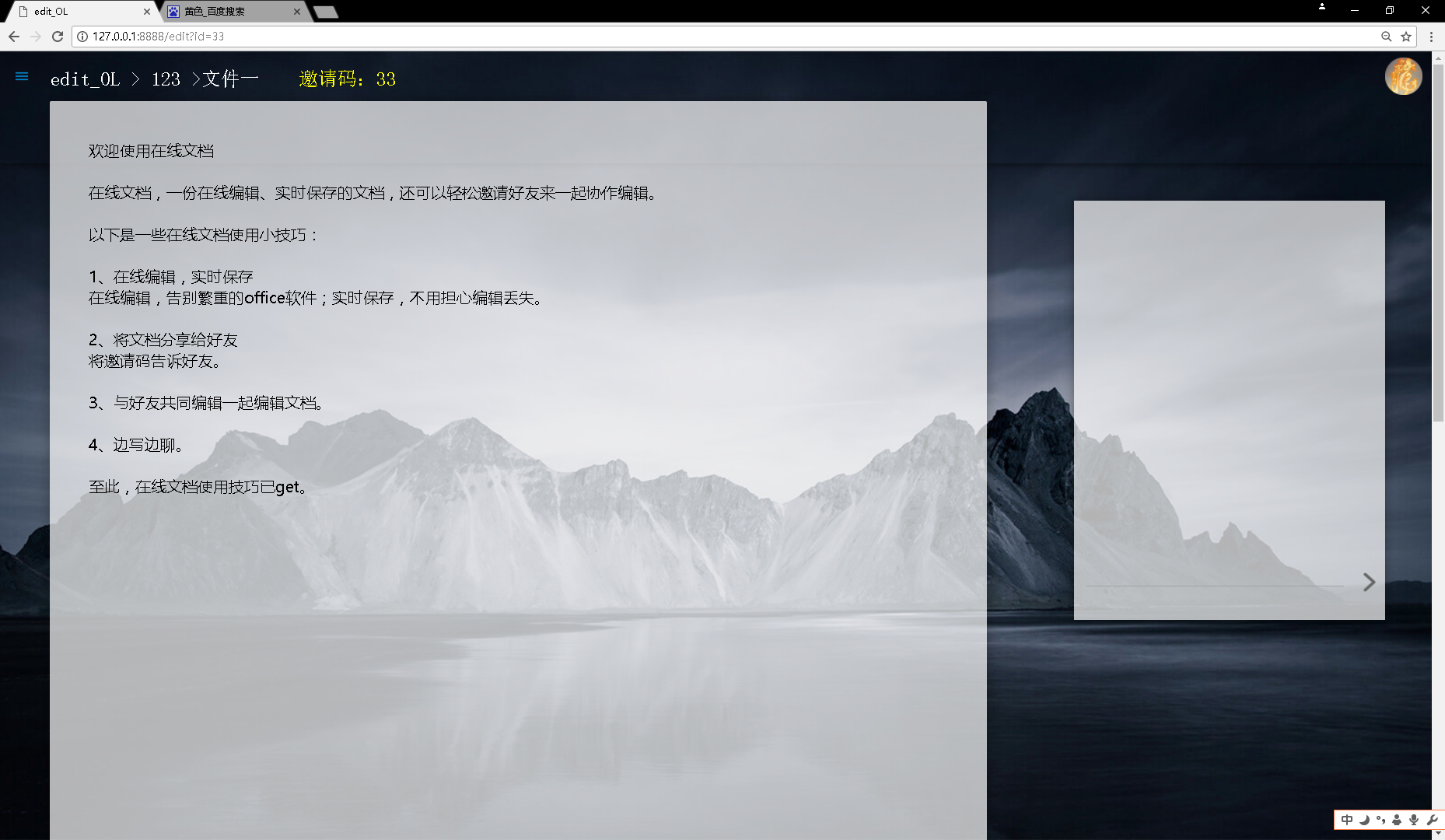
采用了html+js+css+bootstrap

## 文件列表



采用了html+js+css+bootstrap

## 文档编辑



采用了html+js+css

# 功能介绍

## 登陆注册模块

在注册页面，输入账号、密码、名字进行注册，注册信息不能为空，同一账号不能重复注册。

使用http协议post请求，将注册信息发送服务端进行检验注册，返回失败信息表示注册失败，否则注册成功。

注册成功后将自动登录。

在本平台后注册成功后，使用注册账号和正确的密码可进入访问网页，其他情况则登录失败。

使用http协议post请求，将账号密码发送服务端进行验证，返回该用户信息表示登录成功，否则返回失败信息。

## 文件管理模块

在文件管理界面，点击添加文件按钮，输入文件名，点击提交，确认添加。文件名不能为空。

使用http协议post请求，将文件信息发送服务端进行添加。

在添加文件后，文件显示在文件列表。

服务端发送该用户拥有的文件，客户端使用ajax接收数据并处理。

在文件选择列表选择文件，点击删除。

使用http协议post请求，将文件信息id发送至服务端删除。

## 文本编辑管理模块

实现了多人编辑互不影响，实时展现每个人的操作结果。

使用websocket实现实时传输消息，每当有用户连入时，服务器将保存该用户所在的文件id，用户id，会话凭据。根据文件id读取和保存文件内容。

服务器接收到信息时，修改保存的内容。遍历正在使用这个的文件的所有用户，转发消息。信息中包含了光标位置。

根据接收到的光标位置和现有光标位置以及文本内容是否增加来决定最终的光标位置。

每次按键弹起时，如果文本内容已发生变化，则发送数据。

传输结构：

{

msg :群聊信息或者文本内容;

type : 有连接、关闭、聊天、文本四种类型;

flname : 文件id;

ssname : 用户id;

guang:光标位置;

}

存储结构:

{

文件名1：{

用户id1:会话凭证;

用户id2:会话凭证;

…

}

文件名2：{

用户id1:会话凭证;

用户id2:会话凭证;

…

}

}

## 群聊模块

实现了实时群聊。

使用websocket实现实时传输消息，每当有用户连入时，服务器将保存该用户所在的文件id，用户id，会话凭据。服务器接收到信息时，遍历正在使用这个的文件的所有用户，转发消息。

# 总体设计

## 前端

### 运行环境

Chrome浏览器

IE浏览器

Firefox浏览器

### 开发环境

Vscode

### 技术路线

html+js+css+bootstrap

## 服务端

### 运行环境

Node.js

### 开发环境

VScode

### 数据存储

mysql

### 技术路线

Express

websocket

# 技术难点

## 实现群聊

实现群聊的方式也许有很多，但为了实现后续的多人共同编辑，就不能使用轮询，定时刷新等实时性不够的方案。

解决方案：使用websocket实现实时传输消息，每当有用户连入时，服务器将保存该用户所在的文件id，用户id，会话凭据。服务器接收到信息时，遍历正在使用这个的文件的所有用户，转发消息。

是否解决：是

## 操作提取

如何得知用户进行了什么操作，是进行文本改变前后的对比，还是分析键盘的操作？如何让每个用户的操作互不干扰。

目前方案：采用了全文传输+光标记录。

优化思路：通过按键按下时光标的位置和按键弹起时光标的位置，得到字符串改变的位置和数据，通知所有人进行此操作。

是否解决：暂时解决

## 结果接收

怎么让用户接收到最新的文档内容，通常服务器不能主动向客户端发送信息。是接收整个文件内容还是接收修改指令。在接收别人内容时如何防止干扰正在进行的操作。

解决方案：

使用websocket实现实时传输消息，每当有用户连入时，服务器将保存该用户所在的文件id，用户id，会话凭据。根据文件id读取和保存文件内容。

服务器接收到信息时，修改保存的内容。遍历正在使用这个的文件的所有用户，转发消息。信息中包含了光标位置。

根据接收到的光标位置和现有光标位置以及文本内容是否增加来决定最终的光标位置。

每次按键弹起时，如果文本内容已发生变化，则发送数据。

是否解决：是