**实验一：需求获取**

小组成员：171860566 王盼盼

171860565 刘振国

成绩比例1:1

**实验目标**

软件需求的抽取与分类。

**实验基本思路**

选定一个开源IDE项目，确定可能的信息来源，获取有效信息，使用特定技术获取潜在需求，并进行分类。

**实验步骤**

1. **确定IDE项目**

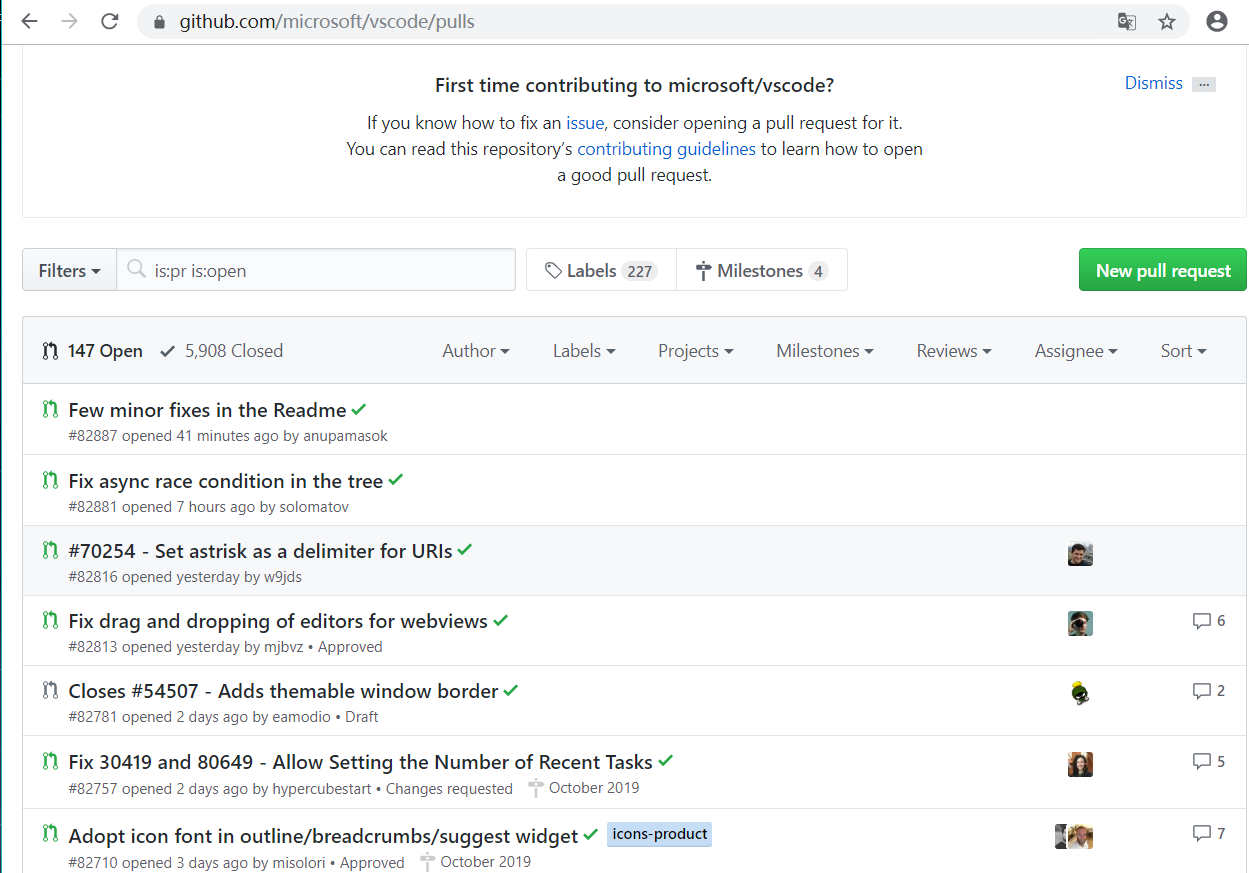
本次实验我们选取Visual Studio Code作为IDE的研究项目。

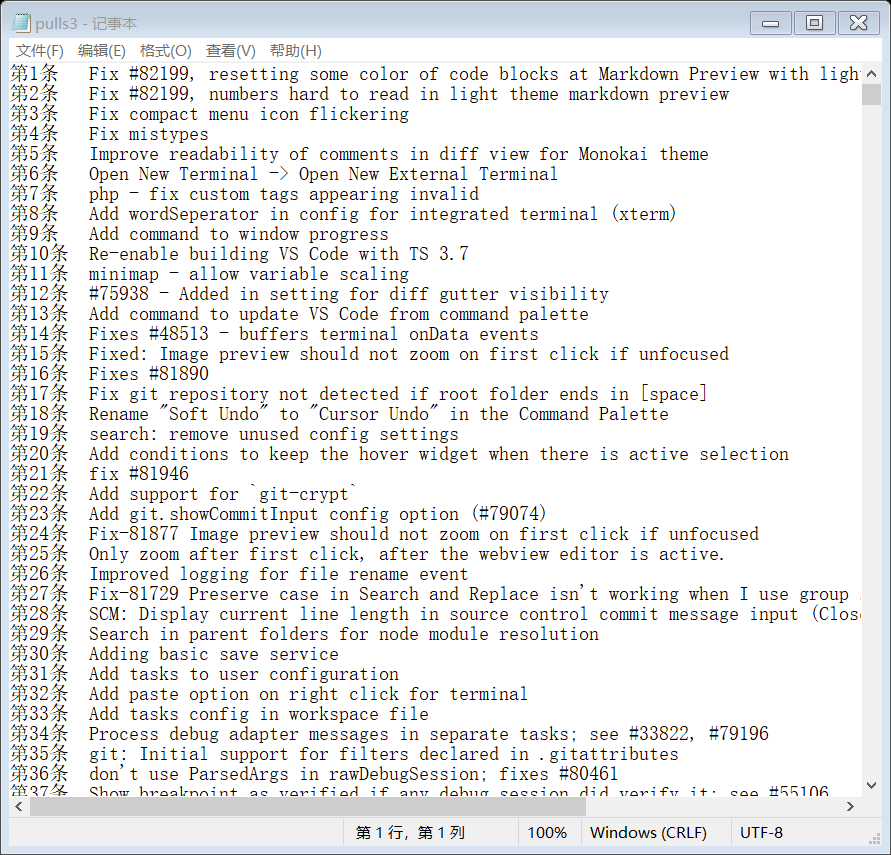
1. **明确信息源**

根据实验指导中思路一的提示，我们明确vscode托管在在GitHub上的地址作为信息来源，实验的思路是对该项目中的Pull Request（地址：https://github.com/microsoft/vscode/pulls）作为一种需求，获取相关信息。

1. **获取数据**

在明确信息源之后，我们在相应地址找到了PR的信息。



我们发现这些PR的标题可以概括相应的需求信息，为了获得众多的标题信息，我们使用Python语言写代码（文件：vs\_pulls\_get.py）来抓取这些信息并存入文件（pulls3.txt）。

1. **需求分析**

通过用户在使用vs的不同场景分析，可将需求大致分为下面几类：

1编辑

2编译

3调试

4资源管理

5调试

6其他

1. **需求分类**

在获得这些信息之后，为了分析不同的信息。我们首先对获得信息进行了词频分析（代码实现文件：1.cpp）获得了从高到低的出现的频率词汇。之后，在这些高频词汇中选取有意义的具有代表性的词汇，将这些词汇出现过的那些PR标题信息单独提取出来（代码实现文件：2.cpp），完成初步分类，再通过对词意相近或相关的分组进行语义分析，完成二次分类：

1.编辑

查找指定内容

查找并替换指定内容

支持多种字体

显示行号、列号

关键字联想

自动对齐

多行注释功能

自定义快捷键

支持文件浏览器水平滚动

将VSCode设置为git编辑器

网格编辑器布局功能

在编辑器状态栏中显示所选行数

实施最近关闭的编辑器，使其在与关闭时相同的位置打开。

当检测到文件编码为非UTF时提示

2.编译

支持多种语言的语法

语法检查，提示出错位置和出错类型

语义检查，提示出错位置和出错类型

单词拼写检查功能，突出显示错误

3.调试

断点调试

带注释的反汇编

显示栈状态

跨工作区文件夹的复合调试

选择和启动调试配置

多个调试会话

4.用户界面

支持多种语言

查找所有按钮

按钮功能

翻译的快捷键

主题的颜色自定义

自定义关键字颜色

用户自定义新的子菜单

窗口（自动适应）、拆分窗口

5.资源管理

文件路径（在窗口标题中显示文件所在的文件夹路径）

从声明处转到定义的功能

为当前项目提供清晰的类视图

暂存新文件

文件流

根据文件内容建议文件名

无标题文件第一行文件名功能

预览文件

允许写入大文件（>256M）

如果项目中有未保存的文件，提交时显示警告

保留大小写但不区分大小写的文件处理

允许在工作区编辑中重命名/创建/删除文件

6.外部扩展

支持当前用户安装Git Bash

7.版本控制系统

支持git

**仓库文件说明**

***data***

word\_frequency.txt 词频统计结果

pulls3.txt PR爬取结果

pullswithlink\_2.txt 附链接爬取结果

其余.txt文件 按关键词提取爬取的信息

***code***

Word\_frequency.cpp 词频统计程序

select\_keyword.cpp 按关键词分类程序

vs\_pulls\_get.py vscode的PR标题爬取程序

vs\_pulls\_get\_with\_site.py 附带链接的vscode的PR标题爬取程序