**系统设计说明书**

胡剑桥 201830660420

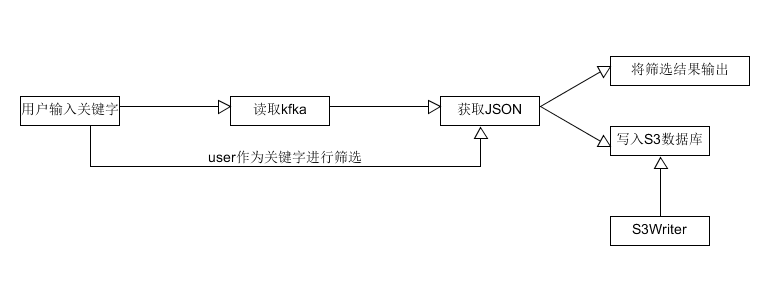
1. 系统概述

使用了flink本地化的功能，连接了socket，可以在本地获取到远程服务器的kafka数据流，获取数据。本系统实现的功能还包括使用kafka连接S3，将数据持久化保存在S3数据库里面。另外，对于数据流的处理，可以根据其中的关键字进行分类，然后将分类筛选的结果持久化保存在S3数据库中。

1. 系统运行流程

在运行系统之后，允许用户输入一个名字表示关键字，在此之后，系统会依照此关键字进行kafka数据流中JSON格式的数据匹配，并且在进行了筛选之后，将筛选之后的数据保存在S3数据库中进行持久化保存。保存的格式是txt格式的文本。

1. 系统流程图

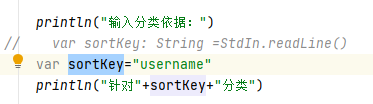


（该图式源文件为根目录下的图例1.uxf）

用户输入的关键字为username,buy\_address,buy\_time,origin等等，用于作为分组并且保存在S3数据库上的依据。

在代码中使用sortKey表示

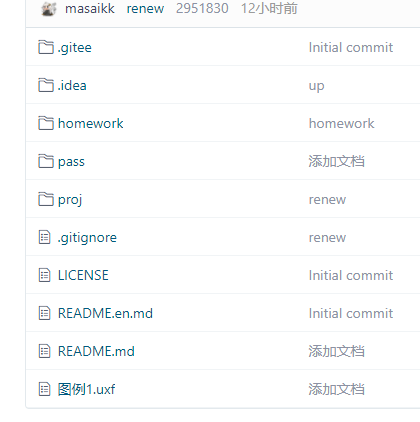
具体如下所示：



在测试的时候使用username作为分类的字段

1. 代码结构

本项目中，文件夹及代码结构如下所示，使用码云作为版本管理器：



在homework文件夹中是拓展题的代码

在pass文件夹中是文档

在proj文件夹中是项目的代码，主要的入口是Main.scala文件。

项目的根目录下还包括了.gitignore文件等，项目的示例图也放在根目录下。

1. 系统主要功能概述
2. 将数据写入S3数据库中

定义了一个读写S3数据库的类，用于将流式数据写入到指定的S3数据库中，该类的名字为S3Writer，在允许它的时候，将它作为参数传入。如下图所示：



其中S3Writer对象在初始化之前的数据配置已经给定，用于连接S3。

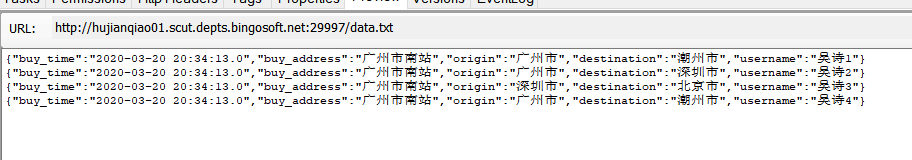
代码实现的部分截图如下所示：



1. 读取数据

可以使用fileName值来移动数据，在本系统中暂时使用了val *fileName* = "data.txt"

来定义，它应该预先放在S3中，它的数据如下所示：



值得注意的是，在这个文本文件中，虽然使用了吴诗的名字，而且购买时间是一样的，但是在名字后加入了标识，所以在程序运行之后，这四条数据依然会被分成不同的结果。