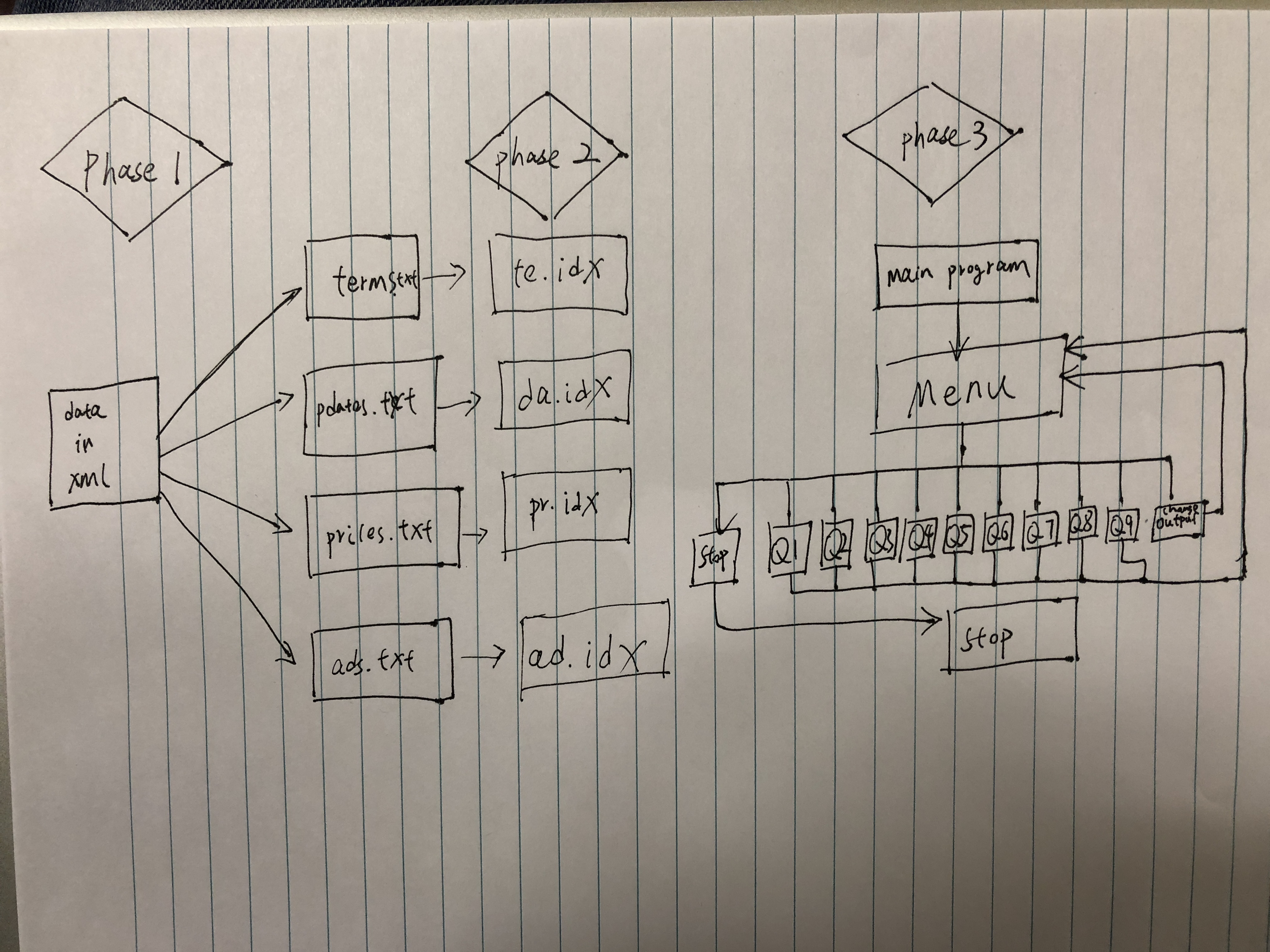
Design Document

General overview of the system / User Guide

The application contains three phases. The first phase will take an xml file as input and it will generate four txt files. The second phase will sort the files that generated in phase 1 and it will also create 4 index files. The third phase will allow users to do some queries. The application is mostly programed in python, and uses its bsddb3 library in order to store data in Berkeley DB. A Linux command script is also included.



Software Design Components

PHASE 1:

Main Function:

The main function will take an xml file as input and generate four txt files (terms.txt,pdates.txt, prices.txt, ads.txt)

Get\_content:

This function will parse the xml file. This function will take three arguments as input: the file name, the tag, and /tag. The content in the tag will be returned.

Write\_terms:

This function will take the xml file name as input. It will generate a file called terms.txt. The content of terms.txt is the content from tag <ti> and <desc>. All the strange characters and the words shorter than 3 characters will be removed.

Write\_pdates:

This function will take the xml file name as input. It will generate a file called pdates.txt. The content of pdates.txt is the content from tag <aid> <date><cat><loc>.

Write\_prices:

This function will take the xml file name as input. It will generate a file called prices.txt. The content of prices.txt is the content from tag <aid> <price><cat><loc>.

Write\_ads:

This function will take the xml file name as input. It will generate a file called ads.txt. The content of ads.txt contains one line for each ad.

PHASE 2:

sort.sh:

This is a Linux shell command program. There is also a Perl script in side sort.sh, which is used for formatting the txt files. This shell command will use Linux command and db\_load function to generate 4 .idx indexes. ad.idx in created in hash and other indexes are created in b\_tree.

PHASE3:

We did not fish PHASE3

Testing Strategy

Group Work Breakdown Strategy