Final Report

[redacted]

Abstract

Our project aiming for providing a light wight solution to provide automatic generation and update of group publication lists. There are several publications list management systems such as DBLP and Google Scholar, but these systems only provide bibliographies for individuals. The requirements of publication lists are wildly considered by many research teams, and some of them have built their own systems, but most of them need to be manually updated by researchers. Our project can acquire data from DBLP which is a regularly updated data base for publications in computer science and generate webpages. The whole process can be easily achieved by finishing a few operations on the front end, and the programs will automatically finish the works.

1 Introduction

Publication list is needed in varies scenarios, and there has some websites provide bibliography management services for individuals. Most of the publications list management providers can provide a stable and convenience service for individuals, but barely have one of them have special focus on managing researchers as research groups, or provide service for groups. Most of the universities or research facilities have one or more research groups which contain most of the researchers of them and focus on special areas, which means that the publications in a research group are mostly focus on the same area and have certain authority in corresponding areas. It is very useful for researchers who looking for more publications in the area if there is a website that unite all publications for groups of researchers. On the other hand, researchers in research facilities normally need to update their publications list manually, which often took lots of trouble and effort while these works already have some third party doing them, for example, DBLP has a regularly updating database. Therefore, the aim of our project is to combine the automatic update and group the researchers.

Our project shell acquires the publication list of required researchers from DBLP open database and generate the webpages to display all data, and the whole process shell completed automatically. Therefore, by using our product which combined the features acquired from DBLP and university research group websites, can improve the experience of both researchers and which looking for publications in specific areas.

2 Project Aims

Since DBLP has an open database which contains BibTxt files which are a greatly formatted bibliography type file for researchers, and they update their database regularly, we planned to use a program to automatically acquire these files for researchers in the researcher list that user inputted. The object of the generation is the website thus the second main target is automatic generation of the webpages contain the information we acquired from DBLP. Moreover, final product shell be easy for visitors to find the information they need and convivence for team manager to manage researcher list.

3 Approach

After first two client meetings, we decide to use python to build back-end server part and BibTxt download part, use C++ to build a parser to parse the downloaded BibTxt, use HTML, CSS and JavaScript to build the front-end.

Python is a commonly used programming language for the server, it is easily to use, can provide lots of packets to provide needed functions and it can provide sufficient performance for our project due to the files we need to transmit. HTML is a wildly used webpage programming language, due to the purpose of the website, display a publication list which contains a few required information, HTML is a suitable choice which can be easily generated by a C++ program. On the other hand, HTML can work very well with CSS and JavaScript because it can take CSS files and JavaScript files as extensions and share the same files with other HTML files. Therefore, our team can work on their part simultaneously and assemble and test the product after related team member finished their parts of job.

My main job of the whole project is writing a parser to convert BibTxt files into HTML files which are organised webpages contain all required information from BibTxt files. I changed the approach twice in milestone 1 and change the program structure once again before milestone 2. I chose a simple approach at first because there has not much work been done at the first few weeks, which means that I did not need to cooperate with other parts of our project. I finished the first edition as soon as possible to generate HTML files as my team member required and in the format they ordered. The second time I changed the structure in milestone 1 is for making my parser can generate two different versions of HTML files, one for the project and another for demonstration. The last big change happened in milestone 2, I removed all functions to output the second version of HTML files and modalised the whole program to make it able to extend its function much easier. I did a few changes after that change of structure to improve the functionality and add a few extensions which are required by our clients. This approach has met the requirements of generation and provide the extendibility as planned, which means this will be the final approach of the parser.

4 Results

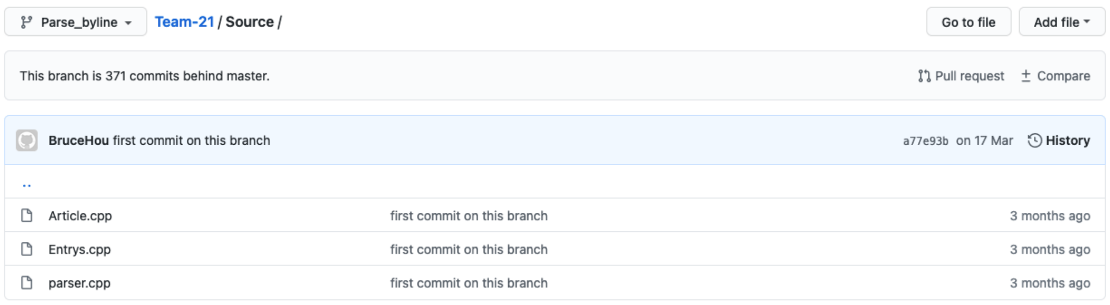
The front-end is the user interface, and its useability is one of the important parts that our client will concern. We have implemented some functions such as filter and navigation bar to improve the user experience. However, based on our test, the filter will work very slow while there has more than a thousand publications in the page. We have tried to improve the performance of our algorism but due to the limitation of the filtering process, we have not been able to make much improvidence. As the process goes, our client came up with new function every week, such as customise fields, add local fields, hide publications and URLs of publications. Once I need to add an extension of my parser, I only need to write a new file that contains all additional functions and add some new lines in the main function, which is benefited from OOP and my program structure. We also tested various wrong inputs to make our program able to handle them and provide some information for our user, but there often still have some unexpected wrong operation that our client did while they play around our website that we have to fix them after we revived feedbacks from our clients.

5 Conclusion

Our team have experienced the benefit of agile development. We provide our solution or subject to our clients and they give us reflections which can greatly prevent us from waste our time and effort on doing the wrong things. During the development, we discovered that the communication was vitally important for a team. We have wasted some time and effort due to the bad communication, for example, we have done some duplicate work because we could not contact with one of our team members, another team member did the work that needed for the demonstration, but the team member who cannot be reached was doing the same job. However, after a few weeks of working together, we were able to establish a fine cooperative relationship that can make us work efficiently as a team and finish the requests of our client.

6 Appendix

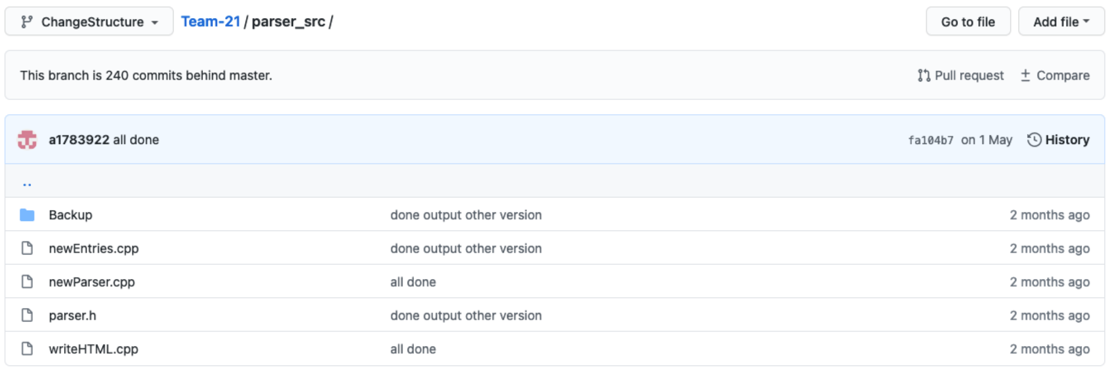
My iteration:



Parse_byline | Team-21 / Source / | Go to file | Add file

This branch is 371 commits behind master. | Pull request | Compare

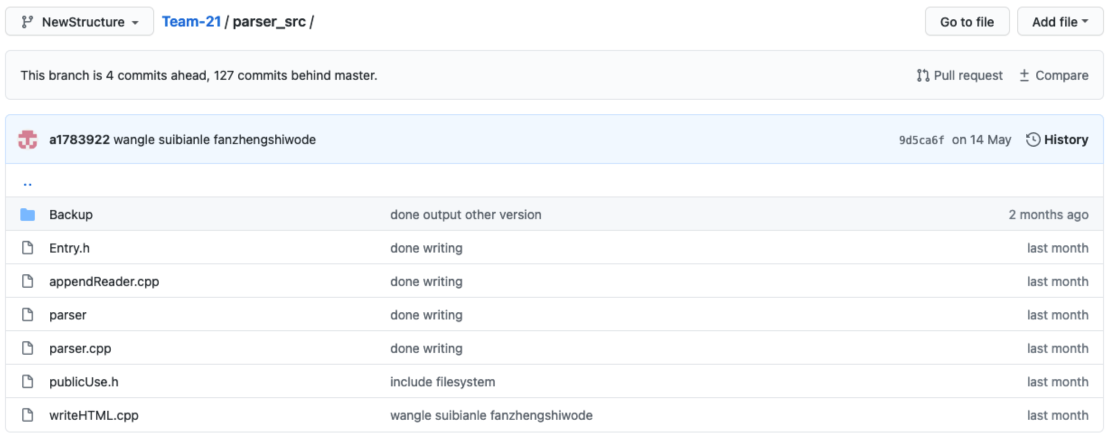
BruceHou first commit on this branch	a77e93b on 17 Mar	History
..		
Article.cpp	first commit on this branch	3 months ago
Entrys.cpp	first commit on this branch	3 months ago
parser.cpp	first commit on this branch	3 months ago



ChangeStructure | Team-21 / parser_src / | Go to file | Add file

This branch is 240 commits behind master. | Pull request | Compare

a1783922 all done	fa104b7 on 1 May	History
..		
Backup	done output other version	2 months ago
newEntries.cpp	done output other version	2 months ago
newParser.cpp	all done	2 months ago
parser.h	done output other version	2 months ago
writeHTML.cpp	all done	2 months ago



NewStructure | Team-21 / parser_src / | Go to file | Add file

This branch is 4 commits ahead, 127 commits behind master. | Pull request | Compare

a1783922 wangle suibianle fanzhengshiwode	9d5ca6f on 14 May	History
..		
Backup	done output other version	2 months ago
Entry.h	done writing	last month
appendReader.cpp	done writing	last month
parser	done writing	last month
parser.cpp	done writing	last month
publicUse.h	include filesystem	last month
writeHTML.cpp	wangle suibianle fanzhengshiwode	last month