Final Project: Search Author Online

Zhan Xinyu, 517030910358
Wang Zhongye,
Xie Yichen,
Yue Ye

June 17, 2018

Contents

1 Introduction

1.1 Report Overview

1.2 Special Specification

2 Project Overview

2.1 Website Overview

2.2 Project Structure

2.2.1 Website Structure

Pages Our website mainly consist of 4 types of pages: *Home Page*, *Result Page*, *Information Pages* and *Stats Pages*.

Home Page The entrance of the website. Users can launch queries from here.

Result Page Presents the result of the query, either from the *Home Page* or from the search box in the navigation bar.

Information Page Displays the basic information of one entity (publications, coauthors, affiliations, conferences, etc).

Stats Page Contains the visualization graphs and charts, and recommendations on *Paper Stats*.

These pages are organized into the structure showed in the following chart:

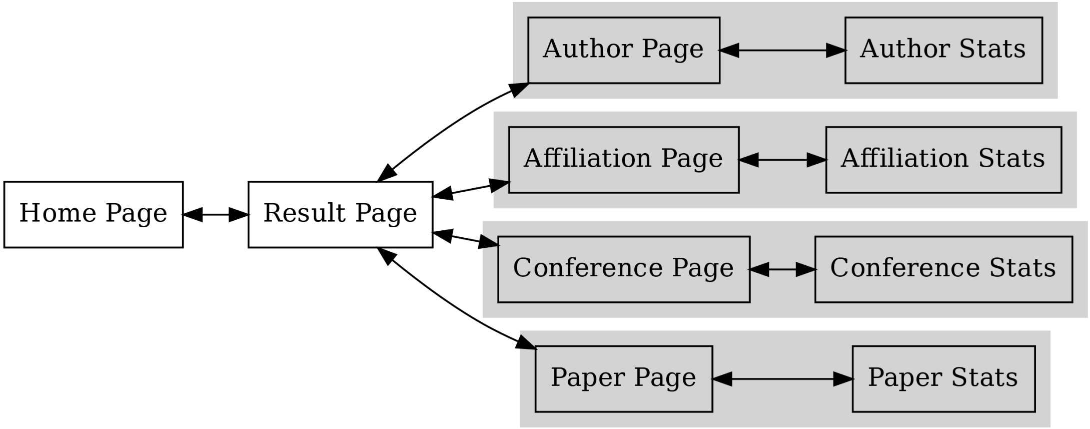


Figure 1: The structure of the website

As the chart shows, users will first visit our *Home Page*, input some keywords, and press Enter. The page will redirect to *Result Page*, and after user click one of the links present there, it will jump to the *Information Pages*, which includes *Author Page*, *Affiliation Page*, *Conference Page* and *Paper Page*. From the navigation bar on the top of the *Information Pages*, users can direct to the *Stats Pages* corresponding to the *Information Pages*. All pages can return to the *Result Page* of the most recent query, and the *Home Page*.

2.2.2 Directory Structure

In this project we use Code Igniter framework. The framework features MVC design pattern and a bunch of utility and helper funcitons. As a result, we mainly write codes and The directory structure of out project is like following:

application This directory contains the php scripts that runs on the server and fulfill different jobs. These files mainly fall into three category: controllers, models and views, which follows the MVC design pattern.

controllers

1. Lab.php
   Static Content Controller for all pages. Also provides dynamic content for text queries.

2. Autocomplete.php
   Dynamic Content Controller for all input box's auto completion.

3. Label.php
   Dynamic Content Controller for label extraction and paper recommendation.

4. Visual.php
   Dynamic Content Controller for visualization graphs and charts.

5. Pagin.php
   Dynamic Content Controller for pagination bar and corresponding js code.

models

1. Lab_model.php
   Data Provider for all text queries.

2. Visual_model.php
   Data Provider for queries from visualization graphs and charts.

3. Label_model.php
   Data Provider for label extraction and paper recommendation.

views The directory contains multiple sub-directories.

assets This directory contains the css files and js scripts that will be sent to the client side and processed by the browser.

css

js
Overview:

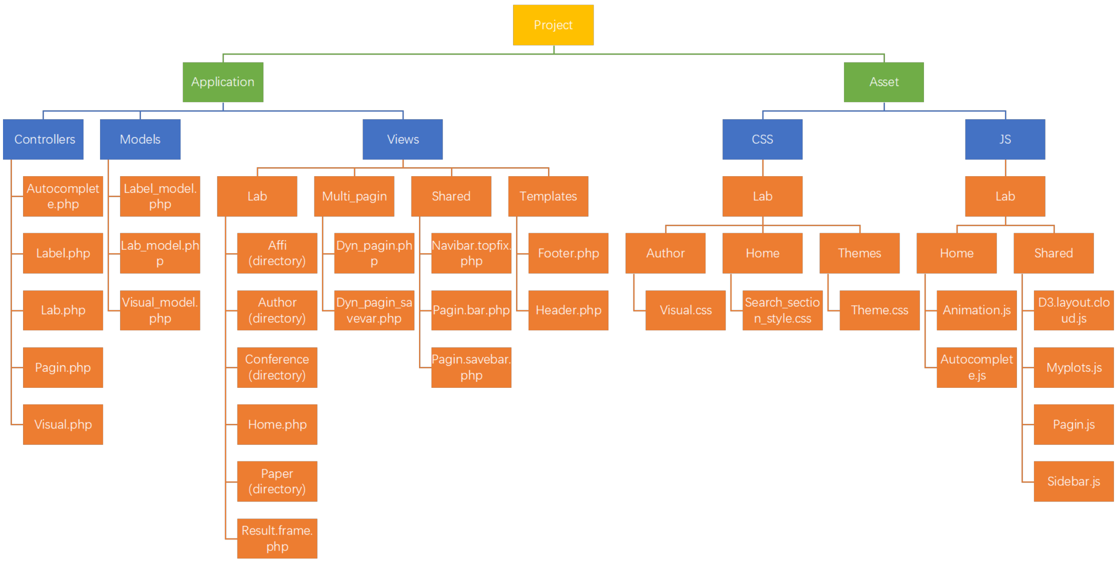
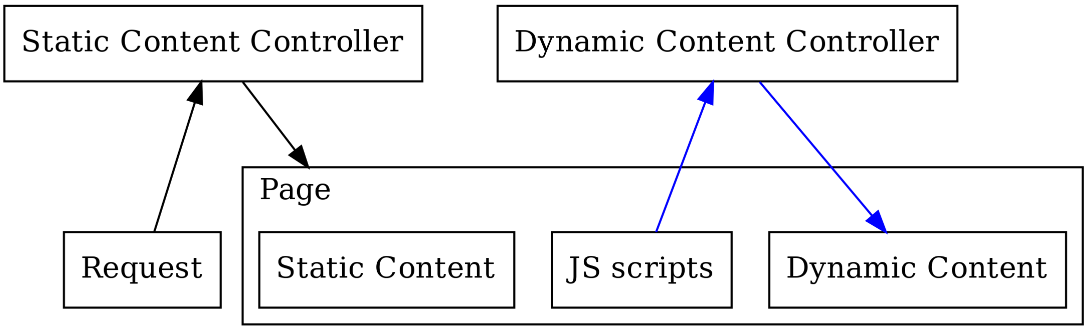


Figure 2: Directory Structure

2.2.3 Request Process Procedure

The *Result Page*, *Information Pages* and *Stats Pages* all involves many dynamic content loaded by js scripts from client side. To keep things simple, we design a standard and uniform request process procedure, and write some utilitiy functions for it. The procedure is showed as follows:

Figure 3: Request Process Procedure



2.2.4 Thirdparty packages and libraries

2.3 Project Organization

2.3.1 Workload Division

Zhan Xinyu Project structure; Website backend; Utility functions.

Wang Zhongye Webpage Design; Website frontend, including css and js; Visualization charts.

Xie Yichen Label Extraction; Page Recommendation.

Yue Ye Visualization backend; Slides.

2.3.2 Collaboration with Git and Github

To share code and synchronize

2.3.3 Website Deploy

3 Frontend Implementation

3.1 Overview

3.2 UI Design

3.3 Data Visualization

4 Functionality Implementation

4.1 Backend Implementation

4.2 Visualization Implementation

4.3 Label Extraction

4.4 Paper Recommendation

5 Future Work

6 Conclusion