Yongxing Jiang. Build a Web Database Platform to Analyze Health Status, Health-Related Quality of Life (HRQOL), and Health Economic Outcomes (HEO) of Cancer Patients and Their Caregivers. A Master's Paper for the M.S. in I.S. degree. April, 2018. 36 pages. Advisor: Fei Yu

The goal of this exploratory study is to build a web database platform to facilitate the analysis of the relationship between health status, HRQOL, and HEO in family caregivers and cancer patients. This paper describes the whole design process of the Web database, query and data visualization platform including (1) target users' requirement (scientific research Needs); (2) medical expenditure panel survey (MEPS) data cleaning and processing; (3) methods and technical details of developing web databases (server-side) and the front-end user interfaces. This web database platform will help researchers achieve efficient querying and visualization of big medical dataset.

Headings:

Data processing

Web design

Web database

Web development

BUILD A WEB DATABASE PLATFORM TO ANALYZE HEALTH STATUS, HEALTH-RELATED QUALITY OF LIFE (HRQOL), AND HEALTH ECONOMIC OUTCOMES (HEO) OF CANCER PATIENTS AND THEIR CAREGIVERS

by
Yongxing Jiang

A Master's paper submitted to the faculty
of the School of Information and Library Science
of the University of North Carolina at Chapel Hill
in partial fulfillment of the requirements
for the degree of Master of Science in
Information Science.

Chapel Hill, North Carolina

April 2018

Approved by

____

Fei Yu

Table of Contents

Introduction

Family caregivers provide a significant portion of outpatient care for cancer patients. Yet cancer-related stress and the demands of caregiving have a profound negative effect on caregivers' health status, and subsequently on caregivers' health-related quality of life (HRQOL) Health status in caregivers can directly and indirectly impact both caregivers' and cancer patients' health economic outcomes (HEO), including health care resources use and costs. Caregivers may need to seek care for their own health and illness that are associated with the stress related to the patients' cancer and caregiving for the patients. Meanwhile patients may need additional services because caregivers are less capable of caring for the patient. However, little research has examined the interrelationship between health status, HRQOL, and HEO in patients and caregivers. Such information is important to help identify and develop effective and affordable strategies to reduce the burden of cancer. The goal of this exploratory study is to build a web database platform to facilitate the analysis of the relationships between health status, HRQOL, and HEO in family caregivers (caregiver) and cancer patients (patient). This web database platform will serve the following two aims:

Aim 1: To describe health status, HRQOL, and HEO in caregivers and patients in relation to caregivers' and patients' characteristics. This will be done first for patients and caregivers separately and then jointly.

Aim 2: To explore the associations between health status, HRQOL, and HEO in caregivers and patients, controlling for patient and caregiver characteristics.

The dataset for this project was collected from the Medical Expenditure Panel Survey (MEPS), which the Agency for Healthcare Research and Quality (AHRQ) (www.ahrq.gov) provides public access to. MEPS is a family of surveys intended to provide nationally-representative estimates of health expenditure, utilization, payment sources, health status, and health insurance coverage among the noninstitutionalized, nonmilitary population of the United States. This series of government-produced datasets can be used to examine how individuals interact with the medical care system in the United States, including the two aims that this project is trying to address.

Although over 10 years MEPS datasets are available on the AHRQ website, the household dataset for the single year 2015 was selected for the prototype of this project. I developed a web database scheme on the server to host the selected MEPS dataset and then designed a user interface connecting with the back-end server enabling user to directly query and access the data stored in the web database.

The web database platform will not only provide access to the stored health dataset, but also facilitate the analysis of the relationship between complex health variables. For example, soon after a user enters a key word to retrieve health data on the platform, the results and visualized statistical charts will automatically be displayed on the webpage.

My contributions to this project are (1) transforming the raw MEPS data into a human readable format by applying R and SAS programming skills; (2) Creating a human-computer interactive platform providing data retrieval and visualization. Although the raw data on AHRQ website is publicly available, due to its own unique coding format and data type, they are not human readable and requires advanced R and SAS programming to understand the data content. Therefore, my work first involved data type and format transformation and then creating a Web platform to facilitate researchers in data searching, extracting, and analyzing.

1 Components

1.1 Web database

The web database refers to embedding the database into the computer network system by the help of network technology (Zhi, c 2015). A web database enables a large amount of information being stored in the database. Web database enables a real-time dynamic data exchange between front-end user interface and back-end server.

1.2 User interface design

User Interface (UI), is a medium for the interaction and information exchange between systems and users. The user interface facilitates the interactive communication between the user and the hardware, software design so that the user can easily and effectively to operate hardware to achieve two-way interaction. User interface is usually defined broadly, including human-computer interaction and graphical user interface.

1.3 Visual analytics

Data visualization is a visual representation of data. The visual expression form of data should include all attributes and variables of corresponding information units.

2 Method

2.1 System design:

The designed system includes a front-end user interface, which enables users to enter their queries and analyze the search result. A back-end database server hosts the treated MEPS dataset downloaded from the AHRQ website.

2.1.1 Target users and basic requirement

In the software design process, the very beginning step is to determine the target users of the software and obtains the needs of them. In addition, the future maintenance and revision of the system should also be taken into account.

The identified target users of the design system are researchers who are interested in investigating all health sciences related issues using MEPS data.

2.1.2 Collecting the users' requirement

I communicated with two primary target users (i.e., Dr. Lixin Song at the UNC School of Nursing and Dr. Fei Yu at the UNC Health Sciences Library) at several meetings to fully understood their needs and expectations. Since the targeted users of this system are mainly researchers in academic institutions, the system design should be rigorous, accurate, simple, and direct. In addition, according to the requirement of the primary target users, the database schema and the code of user interfaces need to be reusable and expandable for ongoing research interests.

The basic requirement of the target users is a web database system storing multiple-year MEPS datasets. This web database should provide keyword search functions. For example, uses should be able to type keywords into a search box or select items from a drop-down list, then the website will display the search results, perform statistic data analysis, and visualize the data via graphs.

For user interaction results and feedback, results and system feedback information should be well organized, and the user is guided to perform the next operation required by themselves.

I conducted iterative interviews to understand user needs. The goal is obtaining and realizing the users' entire requirements. In each interview, I tried to understand the user's web database use behavior, discuss the functions of the expected system and the design goals of the user interface, and confirm the collected information. In addition, I observed the user-database interaction behavior in order to design a user-friendly system that is relevant to users' mental model and existing behavior.

2.2 Prototype of User Interface

The prototype of the user interface provides the following functions:

- Two query input boxes, which enable users to enter query word to look for matched data set in the back-end database according to the needs of his/her work and research. The query word can be specific disease name, cancer type, or both. In addition, this part of the function and layout was designed to be flexible because the input box can be further customized to suite target users' evolving requirement. Such design aims to achieve a more expandable open source system in the future.
- Eight parameters with drop-down menus (i.e., age, sex, race, marital status, health status, mental status, high blood pressure, and cancer) were provided as additional data search filters. For example, upon clicking the drop-down menu of age, users will be provided with a drop-down menu listing all age intervals they would like to view and analyze. The type of drop-down menu and the content and arrangement on the menu can also be easily customized according to the needs of the user.
- "result level display" selection bar, which controls the detail of the information disclosure is presented in the form of a pull-down menu. The returned results can be displayed at three levels on demand and each level presents different granularity of data details. The first level shows only the basic information of eligible patients; the second level shows detailed information about their physical condition beyond the first level; the third level shows all information in each record, including insurance and treatment costs.
- The result summary by selected parameter filters and visualization are provided as search output. If a user used parameter filters to search, the retrieved data will be

summarized and displayed in a table for each parameter filter that the user adopted. To visualize the data based on the result summary, statistical graphs (bar charts, line charts, and pie charts) are automatically generated.

- The retrieved data records with specified level of display are presented at the bottom of the web page. The data can be directly exported to excel spreadsheet through copying and pasting functions.

2.3 Iterative development

In an iterative development approach, the entire development effort is organized as a series of small, fixed-length (e.g., 3 weeks) small projects called a series of iterations. Each iteration includes definitions, requirements analysis, design, implementation, and testing. With this approach, development work can be started before requirements are fully identified and development of a part of the system's functionality or business logic can be completed in one iteration. Then through customer feedback to refine the requirements and start a new round of iterations.

In this project, I used iterative development due to its small scale and less complex system design. The whole design process consisting of identifying user needs, specifying user requirements, creating prototypes, and testing with two primary target users has been iterated until the prototype met the target users' needs.

I used the Axure (https://www.axure.com/) interface to design and build prototypes of user interfaces and used tentative prototypes to communicate with target users to identify problems and solve problems.

Axure RP, the flagship product of Axure Software Solution, Inc., is a professional rapid prototyping tool that enables experts to define requirements and specifications, design features and interfaces, and quickly create wireframes, flow diagrams, or prototypes.

2.4 Data source

According to the needs of the target users, I downloaded the MEPS data set of 2015 from the AHRQ website.

MEPS has three components: the core Household Component, the Insurance/Employer Component, and the Medical Provider Component. Only the core Household Component is currently available for free downloading. The three components provide comprehensive national estimates of health care use and payment by individuals, families, and any other demographic group of interest. We downloaded the dataset for 2015 whole year for our prototype with more than 30,000 data records and over 2,000 data fields included.

The raw MEPS data set directly downloaded from the AHRQ website is in the SAS format. Which can only be handled by SAS programming. In addition, the data content is coded by the MEPS statistical rules, in which the attributes and values are not human readable.

2.4.1 Dataset format

The original downloaded MEPS data is in SAS format. SAS (Statistical Analysis System) is a modular, integrated large-scale application software system. It consists of dozens of dedicated modules, including data access, data storage and management, application development, graphics processing, data analysis, report preparation, operations research methods, econometrics, and forecasting. Although the SAS format data has many conveniences and advantages in statistics, it is not a good format for data analysis compared with R language. Data in SAS format can’t be directly processed and imported to a database.

Therefore, for this project, I used R to transform the SAS format to a CSV format for the sake of database contents.

2.4.2 Dataset format transition

Comma-Separated Values (CSV), sometimes referred to as character-separated values, store tabular data (numbers and text) in plain text. Plain text means that the file is a sequence of characters and does not contain data that must be interpreted as binary digits. A CSV file can store any number of records separated by line breaks and each record Consists of fields whose separators are other characters or strings, the most common of which are commas or tabs.

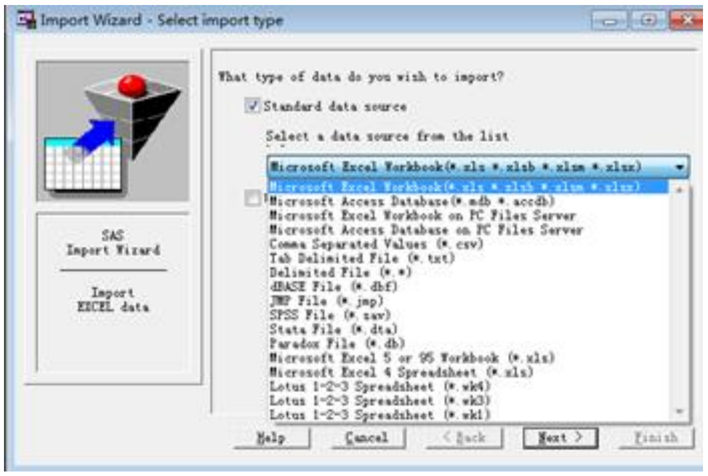
There is no universal standard for the CSV file format, but there is a basic description in RFC 4180. For character encoding, CSV is flexible and have not designate a specific encoding format, but 7-bit ASCII is the most common one.

I used the wizard software to convert the SAS formatted dataset to CSV and view it using an Excel document.

To set up wizard we can use a sequence of dialog boxes that lead the user through a series of well-defined steps.

1) import data

2) select a data source format from list



What type of data do you wish to import?

☑ Standard data source

Select a data source from the list

Microsoft Excel Workbook(*.xls *.xlsb *.xlsm *.xlsx)

```
Microsoft Excel Workbook(*.xls *.xlsb *.xlsm *.xlsx)
Microsoft Access Database(*.mdb *.accdb)
Microsoft Excel Workbook on PC Files Server
Microsoft Access Database on PC Files Server
Comma Separated Values (*.csv)
Tab Delimited File (*.txt)
Delimited File (*.*)
dBASE File (*.dbf)
JMP File (*.jmp)
SPSS File (*.sav)
Stata File (*.dta)
Paradox File (*.db)
Microsoft Excel 5 or 95 Workbook (*.xls)
Microsoft Excel 4 Spreadsheet (*.xls)
Lotus 1-2-3 Spreadsheet (*.wk4)
Lotus 1-2-3 Spreadsheet (*.wk3)
Lotus 1-2-3 Spreadsheet (*.wk1)
```

Figure 1 Select data source on wizard

3) select the data to be imported

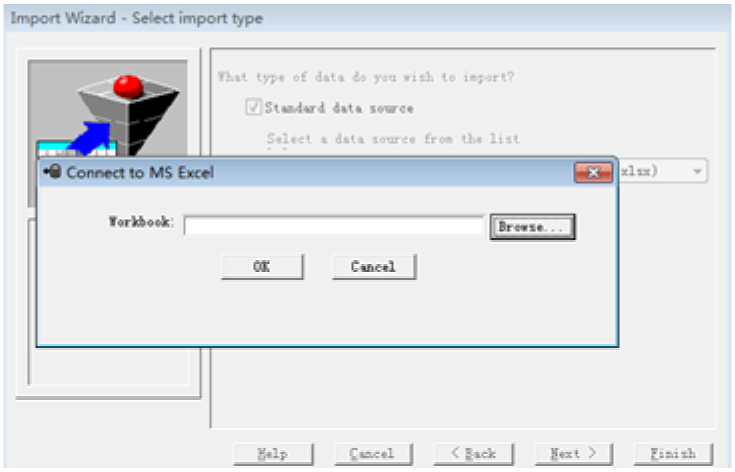
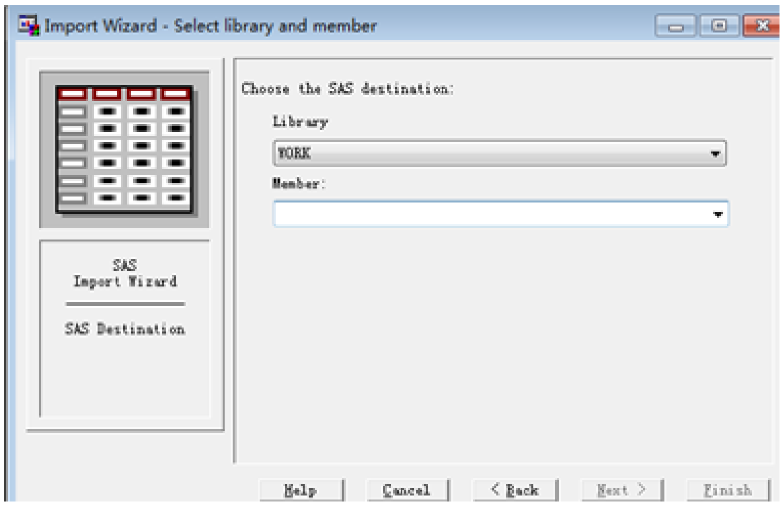

Import Wizard - Select import type
What type of data do you wish to import?
Standard data source
Select a data source from the list
xlsx)
Connect to MS Excel
Workbook:
Browse...
OK
Cancel
Help
Cancel
< Back
Next >
Finish

Figure 2 Select data set

4) Click Ok to enter the library list then select logic library required.



Choose the SAS destination:
Library
WORK
Member:

Figure 3 Enter target version and file name

5) finish.

2.5 Data cleaning

After converting the dataset format from SAS to CSV, the readability of the data was enhanced significantly. The next step is to read the dataset description document such as the MEPS HC-181 2015 Full Year Consolidated Data File and understand each element of each row and column in the dataset for data cleaning.

The data document is about the background of the dataset, which revealed the house hold component, Medical Provider Component, Survey Management and Data Collection. The Technical and Programming Information includes the General Information, Data File Information, which enabled me to better understand the dataset. Based on the understanding of the data document, I can read codebook to revise and clean the dataset.

After having a detailed understanding of the data set, I used the Python language to clean and complete the data set.

After the dataset was cleaned, I tagged and translated the data against the codebook from MEPS official code form to meaningful human-readable form. The next step was data selection, such as the selection of the corresponding columns. According to user requirements, key columns were selected and dataset were imported to the web database.

2.6 Local server setup:

To create a prototype of the web database and user interface, I used phpstudy to build local servers and back up the back-end database on a local server. Follow-up. The process of using phpstudy to build a local server is below:

1. Download phpstudy and configure the environment.

2. Follow the steps to install phpstudy.

3. Create the website locally, set the site's name to mepsdb, and then specify mepsdb.com as the site's directory on the desktop.

4. Reserve corresponding folders and space for existing and upcoming src programs.

5. Open phpstudy, click the "Other Options" menu, enter the website domain management.

6. Set up our site information, save the settings, and generate a configuration file.

7. Open phpstudy, click on the "Other Options" menu, change the "hosts" setting, and visit the website on the browser to check the redirection. If the domain name is redirected, the host’s configuration is successful.

8. Create a database.

2.6.1 Create database on the server-side

1 select SQL Server 2008 Microsoft SQL Server 2008 Management Studio to create the server-side database.

2 Select the server name of the machine in the server name. Then click the "Link" button. Enter the Microsoft SQL Server 2008 management Studio.

3 right-click "database", select "New Database", there will be a window, in this window you can enter the name of the database you want to build, but also can modify the file type of the database.

4 After the database name and type been set, click the "OK" button below. The database is established.

2.6.2 Import data into the database

1 CSV file preparation, save the CSV file to the appropriate path, and check the file encoding format to ensure that the data encoding format is ANSI format.

2 Check the existing database with the command "show database".

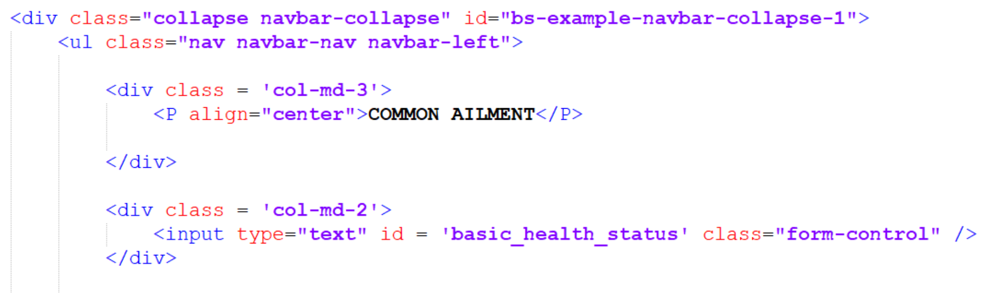
3 Use the "load data" statement to import data into a database created on the server side.

4 Use the "select * from" statement to check the import result.

2.7 Front-end user interface development based on bootstrap frame

Bootstrap is based on HTML, CSS, and JavaScript. It is simple and flexible, making Web development faster. It was developed by Twitter designer Mark Otto and Jacob Thornton and is a CSS/HTML framework. Bootstrap provides elegant HTML and CSS specifications, which is written by the dynamic CSS language Less.

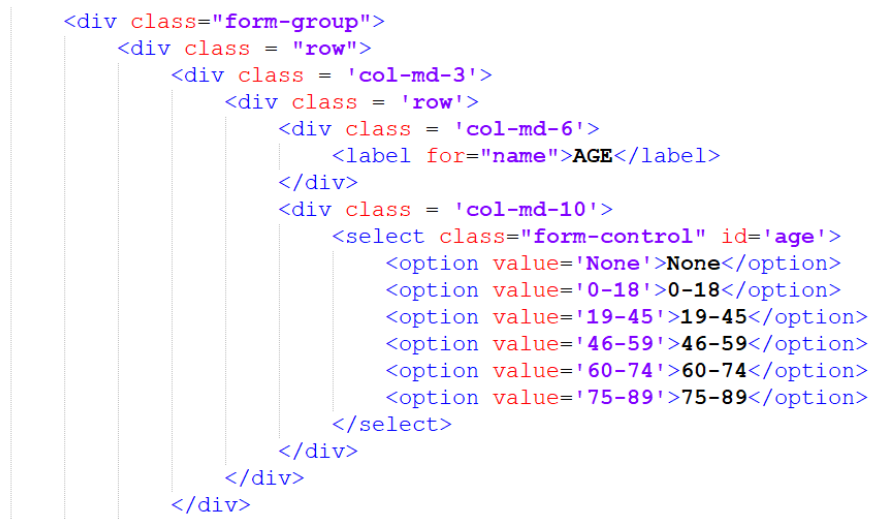
Figure 4 shows the key code to realize main function and elements for the user interface.



```
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
    <ul class="nav navbar-nav navbar-left">

        <div class = 'col-md-3'>
            <P align="center">COMMON AILMENT</P>

        </div>

        <div class = 'col-md-2'>
            <input type="text" id = 'basic_health_status' class="form-control" />
        </div>
```

Figure 4 Code for enter text

2.7.1 Embed JavaScript into the HTML to enable the user input box.



```
<div class="form-group">
    <div class = "row">
        <div class = 'col-md-3'>
            <div class = 'row'>
                <div class = 'col-md-6'>
                    <label for="name">AGE</label>
                </div>
                <div class = 'col-md-10'>
                    <select class="form-control" id='age'>
                        <option value='None'>None</option>
                        <option value='0-18'>0-18</option>
                        <option value='19-45'>19-45</option>
                        <option value='46-59'>46-59</option>
                        <option value='60-74'>60-74</option>
                        <option value='75-89'>75-89</option>
                    </select>
                </div>
            </div>
        </div>
```

Figure 5 Code for drop down menu

2.7.2 Embed JavaScript into the HTML to realize the drop down menu for users.

Bootstrap frame provides a flexible website design and development tools. For example, containers enable developers to add or remove the elements on the website. I added my formatted layout and then embedded drop-down menus, input boxes, pictures, query results display area, page number navigation, are appropriately embedded according to our design.

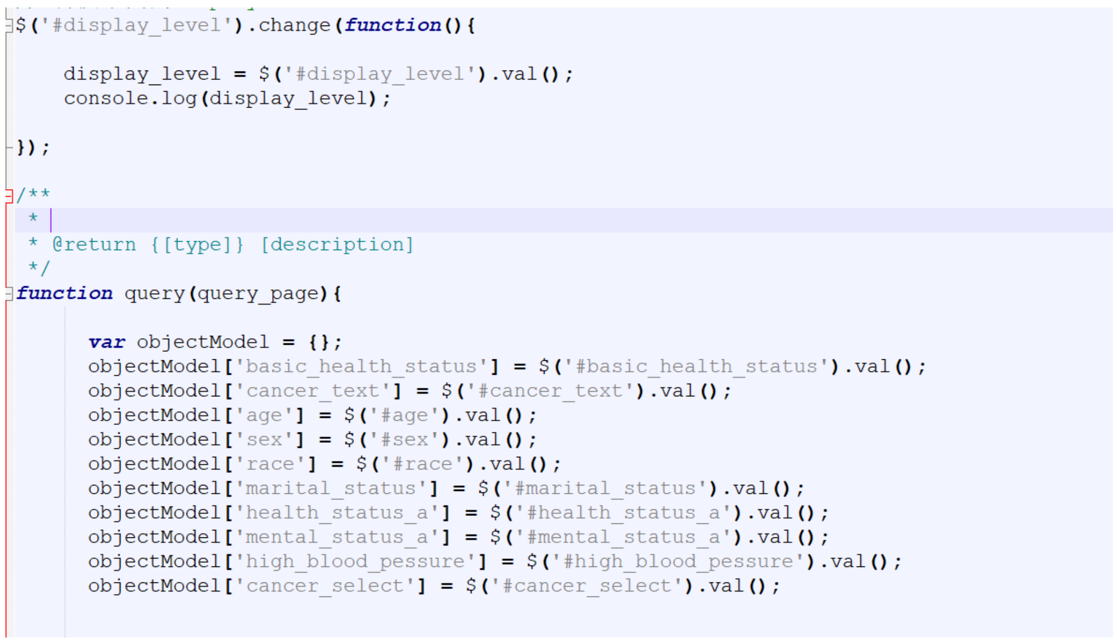

```
$('#display_level').change(function(){

    display_level = $('#display_level').val();
    console.log(display_level);

});

/**
 * 
 * @return {[type]} [description]
 */
function query(query_page){

      var objectModel = {};
      objectModel['basic_health_status'] = $('#basic_health_status').val();
      objectModel['cancer_text'] = $('#cancer_text').val();
      objectModel['age'] = $('#age').val();
      objectModel['sex'] = $('#sex').val();
      objectModel['race'] = $('#race').val();
      objectModel['marital_status'] = $('#marital_status').val();
      objectModel['health_status_a'] = $('#health_status_a').val();
      objectModel['mental_status_a'] = $('#mental_status_a').val();
      objectModel['high_blood_pessure'] = $('#high_blood_pessure').val();
      objectModel['cancer_select'] = $('#cancer_select').val();
```

Figure 6 Code for the query entered by user

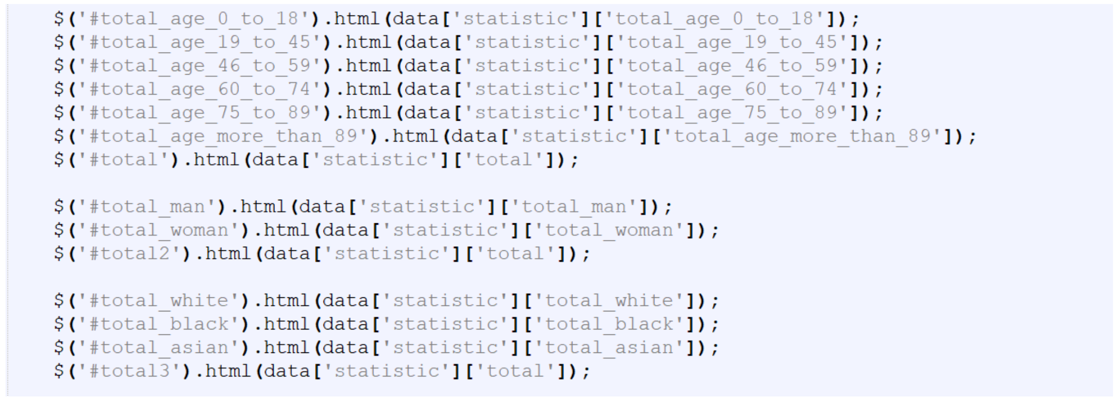

```
   $('#total_age_0_to_18').html(data['statistic']['total_age_0_to_18']);
   $('#total_age_19_to_45').html(data['statistic']['total_age_19_to_45']);
   $('#total_age_46_to_59').html(data['statistic']['total_age_46_to_59']);
   $('#total_age_60_to_74').html(data['statistic']['total_age_60_to_74']);
   $('#total_age_75_to_89').html(data['statistic']['total_age_75_to_89']);
   $('#total_age_more_than_89').html(data['statistic']['total_age_more_than_89']);
   $('#total').html(data['statistic']['total']);

   $('#total_man').html(data['statistic']['total_man']);
   $('#total_woman').html(data['statistic']['total_woman']);
   $('#total2').html(data['statistic']['total']);

   $('#total_white').html(data['statistic']['total_white']);
   $('#total_black').html(data['statistic']['total_black']);
   $('#total_asian').html(data['statistic']['total_asian']);
   $('#total3').html(data['statistic']['total']);
```

Figure 7 Code for access feedback data

2.7.3 Interaction between the front-end and the back-end

The designed system used jQuery and ajax to enable the communication between the front-end and the back-end by generating sets of query condition gathered from the user input and then sending them to back-end. After the result data is sent back, the front-end program will match each attribute by different variables, and then render them into where it should be displayed.

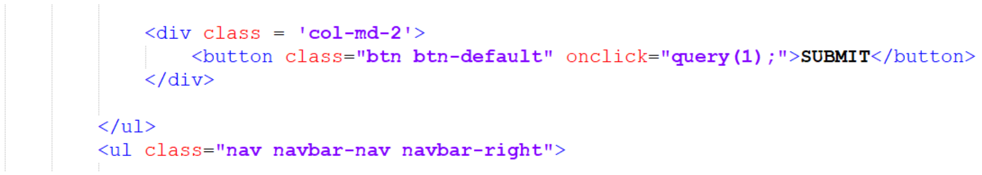
2.7.4 Web page design

The global CSS library in the bootstrap was used to adjust the color and structure ratio of the entire web page. The JavaScript library in the bootstrap and the jQuery library were adopted to generate the dynamic response of the web page, such as generating query apply to the server. Then access the data sent back from server and display it on the table written by JavaScript. I developed the data visualization section by using the plotly library within JavaScript.

2.8 Back-end network database development based on thinkPHP framework

The back-end database was developed by thinkPHP framework. ThinkPHP is a fast, compatible and simple lightweight domestic PHP development framework. It was released under the Apache2 open source protocol, and was transplanted from the Struts structure and further improved.

Figure 8 shows the main method and function at the back-end to realize the interaction between users' input in the user interface and the database on the server, which includes how to manipulate query, how to get data and how to generate a data stream then send back to front-end.

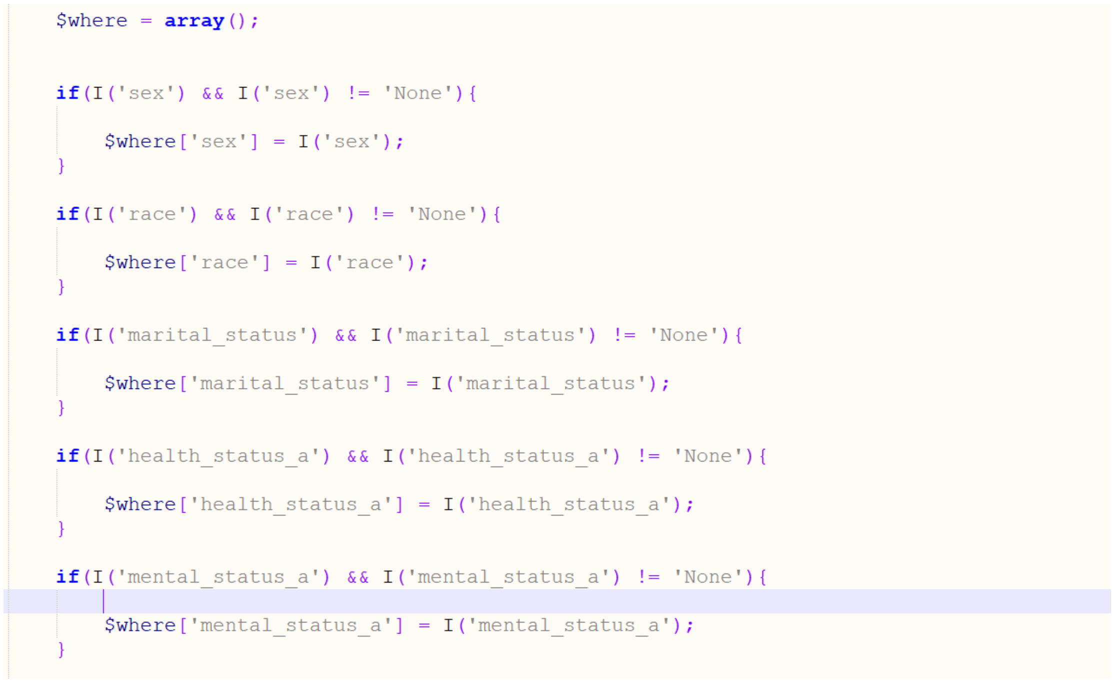


```
        <div class = 'col-md-2'>
            <button class="btn btn-default" onclick="query(1);">SUBMIT</button>
        </div>

    </ul>
    <ul class="nav navbar-nav navbar-right">
```

Figure 8 Code for trigger query submission

2.8.1 Submit button and its trigger function

The front-end user interface triggers the submission of the form each time when the user clicks on the "submit" button. The content includes all user-entered queries and display conditions. The form will be sent to the thinkPHP module based on the back-end. After the background program receives the form sent from the front end, the query conditions contained in the form will be processed separately for classification and different attributes will be treated differently. For example, the age attribute can be treated as interval; user input in the text box can be treated as string, and then the system can do the approximate string matching.

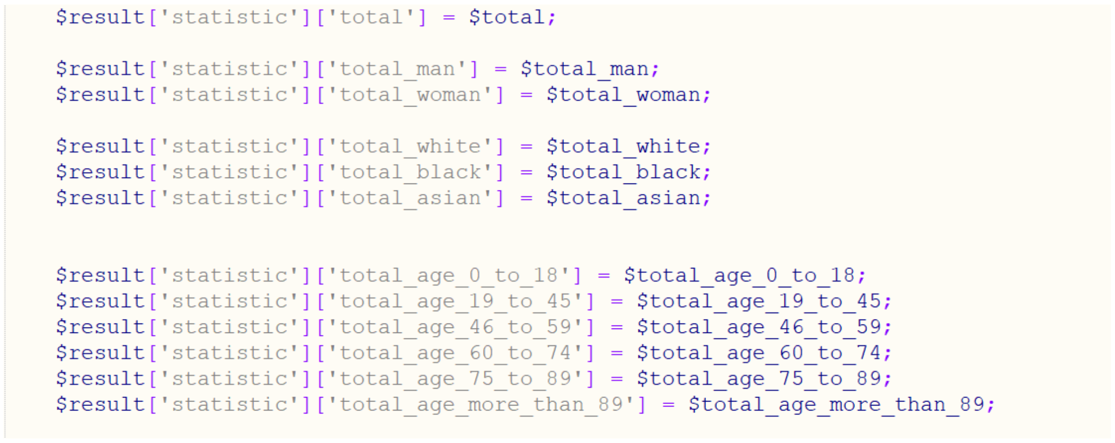


```
$where = array();

if(I('sex') && I('sex') != 'None'){
    $where['sex'] = I('sex');
}

if(I('race') && I('race') != 'None'){
    $where['race'] = I('race');
}

if(I('marital_status') && I('marital_status') != 'None'){
    $where['marital_status'] = I('marital_status');
}

if(I('health_status_a') && I('health_status_a') != 'None'){
    $where['health_status_a'] = I('health_status_a');
}

if(I('mental_status_a') && I('mental_status_a') != 'None'){

    $where['mental_status_a'] = I('mental_status_a');
}
```

Figure 9 Code for generate query array

2.8.2 Query (sorted in an array) on the server side.

All query conditions are integrated into the same query array, which is the advantage of thinkPHP frame. ThinkPHP helps convert MySQL query language into sorted arrays and then query conditions included in such arrays will be sent into back-end database and retrieve matched data records.



```
    $result['statistic']['total'] = $total;

    $result['statistic']['total_man'] = $total_man;
    $result['statistic']['total_woman'] = $total_woman;

    $result['statistic']['total_white'] = $total_white;
    $result['statistic']['total_black'] = $total_black;
    $result['statistic']['total_asian'] = $total_asian;


    $result['statistic']['total_age_0_to_18'] = $total_age_0_to_18;
    $result['statistic']['total_age_19_to_45'] = $total_age_19_to_45;
    $result['statistic']['total_age_46_to_59'] = $total_age_46_to_59;
    $result['statistic']['total_age_60_to_74'] = $total_age_60_to_74;
    $result['statistic']['total_age_75_to_89'] = $total_age_75_to_89;
    $result['statistic']['total_age_more_than_89'] = $total_age_more_than_89;
```

Figure 10 Query result from database

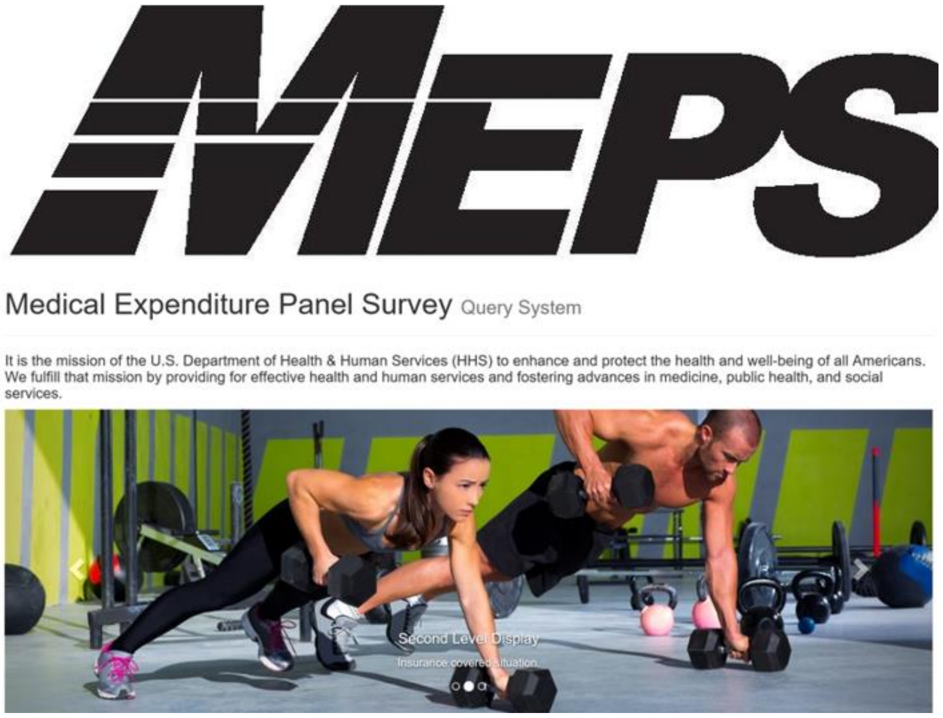
2.8.3 Back-end query result

The returned results are divided into complete data reports and associated statistics, such as the number of data used for paging and the statistical results for data visualization. The backend program on the server side will integrate all the query results and return them to the front page for display.

3 User Interface Prototype

The user interface prototype consists of four major areas, which are introduction area, search area, result summary and visualization area, and result data display area.

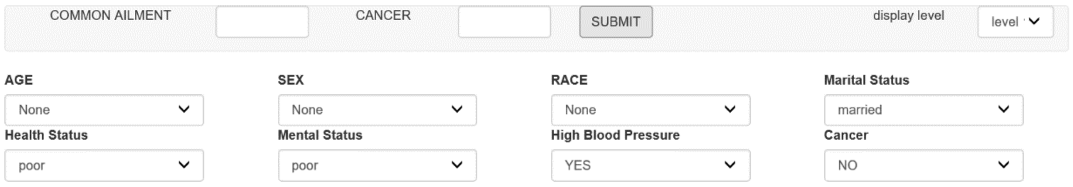
When a user logs into the website, the top area of the webpage is an introduction to the website regarding the data source (Figure 11).


MEPS
Medical Expenditure Panel Survey Query System
Second Level Display
Insurance covered situation.

It is the mission of the U.S. Department of Health & Human Services (HHS) to enhance and protect the health and well-being of all Americans. We fulfill that mission by providing for effective health and human services and fostering advances in medicine, public health, and social services.

Figure 11 Introduction area

Followed by a description of this information display canvas is a slideshow. As a prototype, the current information displaying is the introduction of hierarchical display function of the website. However, this introduction area is prepared for the future maintainer of this system. Both the background and information can be changed and revised.



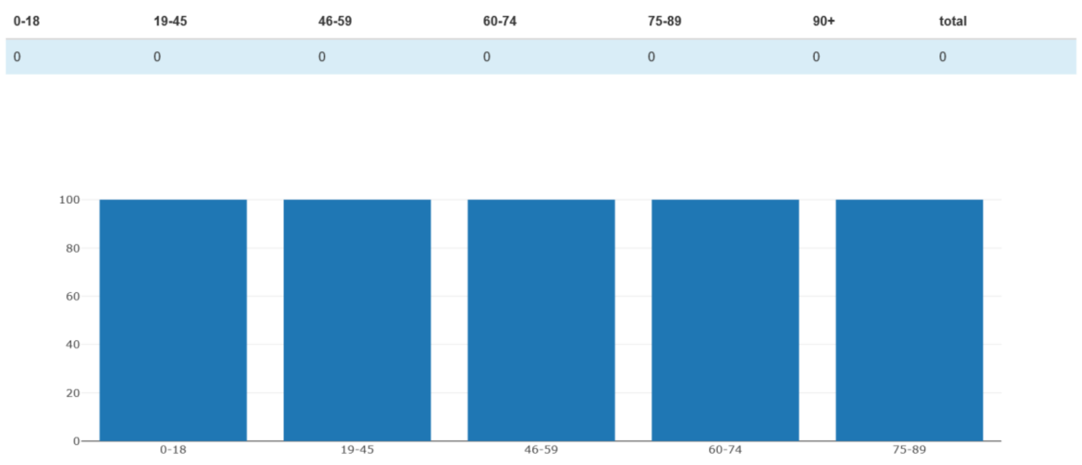
COMMON AILMENT CANCER SUBMIT display level level

AGE	SEX	RACE	Marital Status
None	None	None	married
Health Status	Mental Status	High Blood Pressure	Cancer
poor	poor	YES	NO

Figure 12 User input area

Figure 12 shows the search area on the user interface. Two search boxes are provided which enable users to specify which "common aliment" or "cancer" they would like to search for. In addition, there are eight parameters that users can further use to filter the search results, which are age, sex, race, marital status, health status, mental status, high blood pressure, and cancer. The drop-down choices for each parameter are selected from the MEPS dataset. The "display level" is an option for hierarchical display of data on demand. Since MEPS data include more than 4000 attributes for each record, the user interface offers 3 levels to display returned results. The first level is to display the very basic information of each patient, such as name, sex fields. The second level is to display the physical state of each patient, such as motion state, physical state fields, the third level is to display the financial coverage for each patient such as insurances.

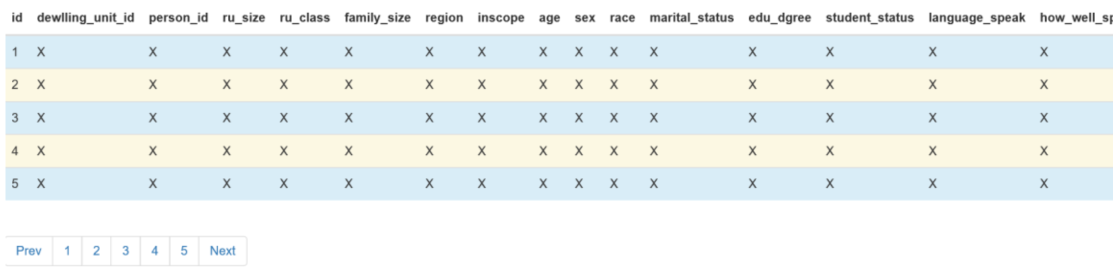
After users complete keyword input and parameter selection and click the "SUBMIT" button, the front web page is trigged to send a query request to the backend database. Then, the result summary and visualization area are updated (Figure 13).



0-18	19-45	46-59	60-74	75-89	90+	total
0	0	0	0	0	0	0

Figure 13 Visual analytics area

Results data display area is at the bottom of the web page. Before the user input query conditions, this area does not show anything. However, the query conditions are executed and results are returned, this area will be updated by providing all data fields related to query conditions.



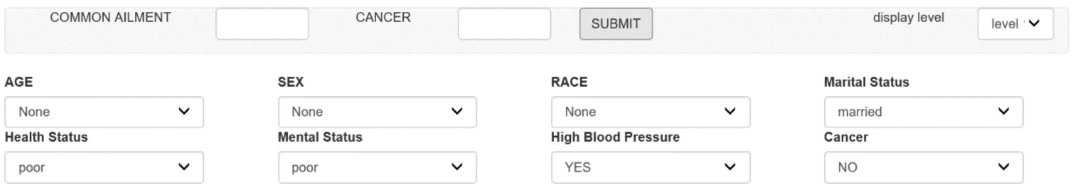
id	dewlling_unit_id	person_id	ru_size	ru_class	family_size	region	inscope	age	sex	race	marital_status	edu_dgree	student_status	language_speak	how_well_sp
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Figure 14 Result data display area

The following are two use case scenarios for this user interface prototype.

Use Case 1

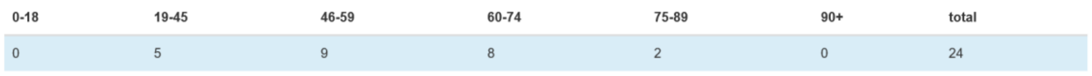
Use Case 1 was to search MEPS data for all married patients who have poor health status, poor mental status, and high blood pressure.



COMMON AILMENT		CANCER		SUBMIT	display level	level

AGE	SEX	RACE	Marital Status
None	None	None	married
Health Status	**Mental Status**	**High Blood Pressure**	**Cancer**
poor	poor	YES	NO

Figure 15 Case 1 user input



0-18	19-45	46-59	60-74	75-89	90+	total
0	5	9	8	2	0	24

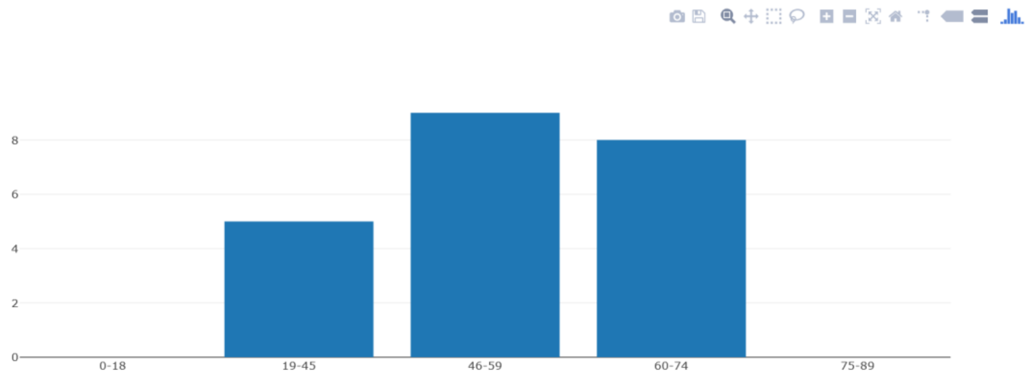
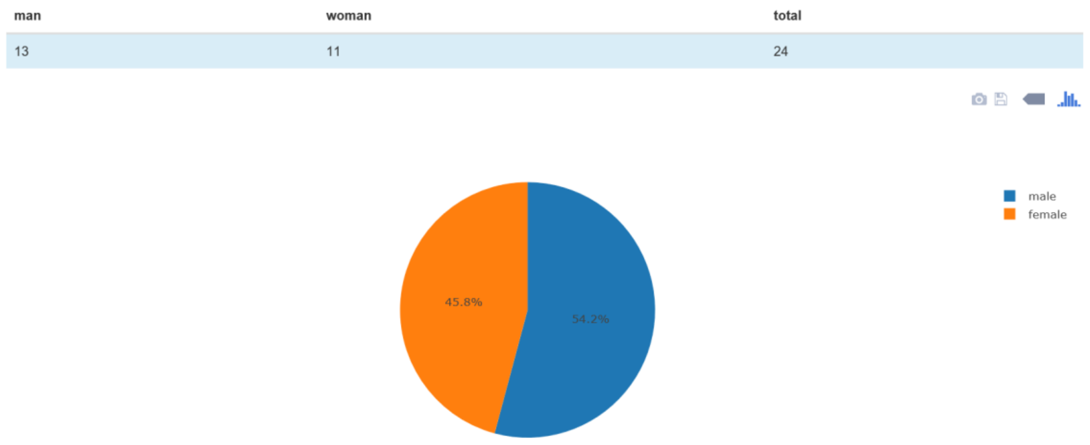


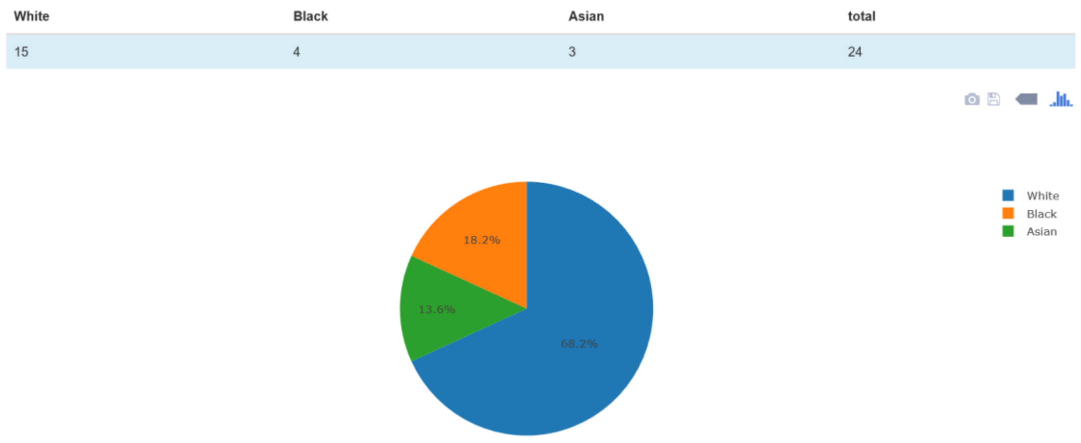
Figure 16 Case 1 result data visualization

According to the input of “high blood pressure” and parameter selections, the result summaries and data visualization diagrams were generated. For example, the total numbers of patients matching the search criteria were visualized across different age-group ranges (Figure 17) and race (Figure 18). The user interface also provides users with zooming and downloading functions for the data visualization diagrams. In addition, the summarized results can also be downloaded as a data table.



man	woman	total
13	11	24

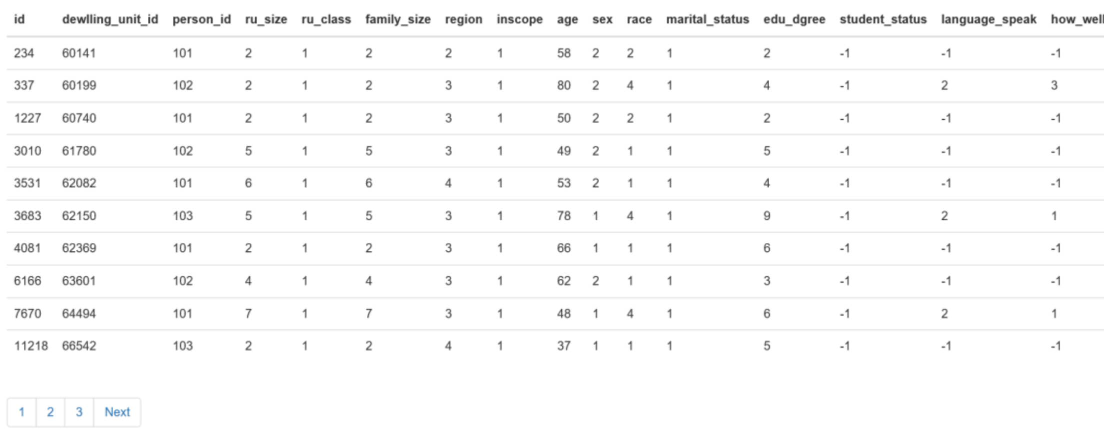
Figure 17 Case 1 result data visualization



White	Black	Asian	total
15	4	3	24

Figure 18 Case 1 result data visualization

Last, the retrieved records were displayed at the bottom with all available data fields (Figure 19).

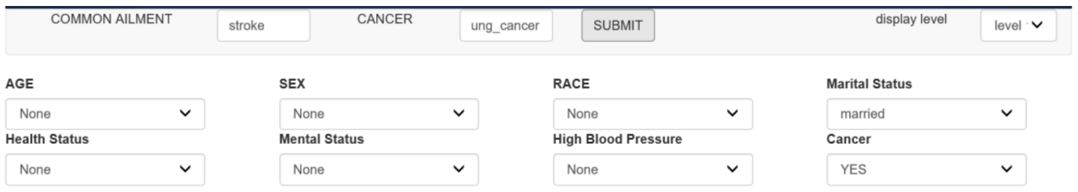


id	dewlling_unit_id	person_id	ru_size	ru_class	family_size	region	inscope	age	sex	race	marital_status	edu_dgree	student_status	language_speak	how_well
234	60141	101	2	1	2	2	1	58	2	2	1	2	-1	-1	-1
337	60199	102	2	1	2	3	1	80	2	4	1	4	-1	2	3
1227	60740	101	2	1	2	3	1	50	2	2	1	2	-1	-1	-1
3010	61780	102	5	1	5	3	1	49	2	1	1	5	-1	-1	-1
3531	62082	101	6	1	6	4	1	53	2	1	1	4	-1	-1	-1
3683	62150	103	5	1	5	3	1	78	1	4	1	9	-1	2	1
4081	62369	101	2	1	2	3	1	66	1	1	1	6	-1	-1	-1
6166	63601	102	4	1	4	3	1	62	2	1	1	3	-1	-1	-1
7670	64494	101	7	1	7	3	1	48	1	4	1	6	-1	2	1
11218	66542	103	2	1	2	4	1	37	1	1	1	5	-1	-1	-1

Figure 19 Case 1 result data display

Use Case 2

Use Case 2 was to search MEPS data for all married patients who have stroke and lung cancer. Figure 20 shows that "stroke" was entered into the search box of COMMON AILMENT and "lung cancer" was entered into the search box of CANCER. The marital status was selected as married.

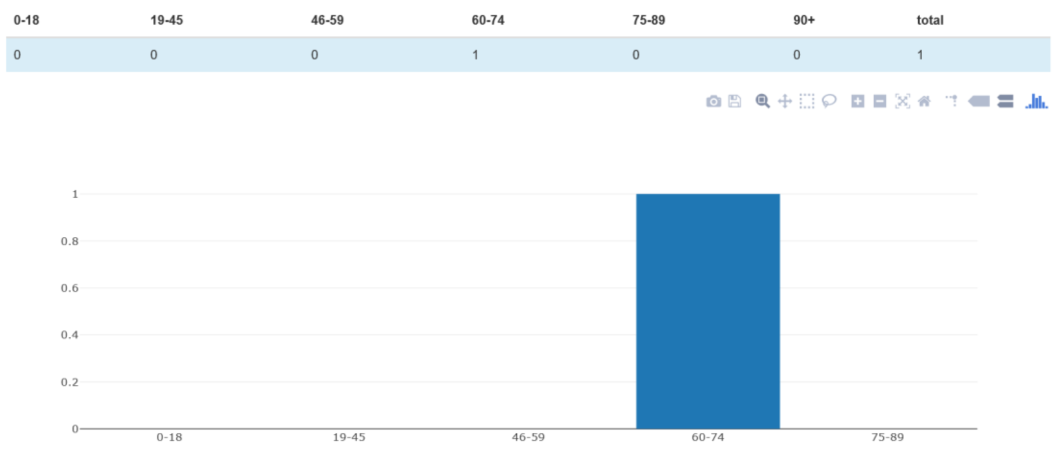


COMMON AILMENT: stroke
CANCER: ung_cancer

display level: level

AGE	SEX	RACE	Marital Status
None	None	None	married
Health Status	Mental Status	High Blood Pressure	Cancer
None	None	None	YES

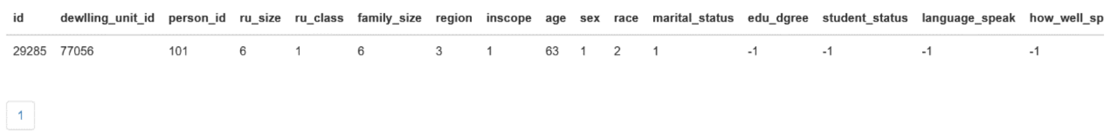
Figure 20 Case 2 user input



0-18	19-45	46-59	60-74	75-89	90+	total
0	0	0	1	0	0	1

Figure 21 Case 2 result data visualization

Due to the small number of patients who met the search criteria, the result summary and visualization shows only one patient in Figure 21.



id	dewlling_unit_id	person_id	ru_size	ru_class	family_size	region	inscope	age	sex	race	marital_status	edu_dgree	student_status	language_speak	how_well_sp
29285	77056	101	6	1	6	3	1	63	1	2	1	-1	-1	-1	-1

1

Figure 22 Case 2 result data display

The only one retrieved record was displayed with all available data fields at the bottom of the webpage (Figure 22).

4 Evaluation

4.1 Usability principle

Although there is no formal usability evaluation conducted to evaluate the user interface prototype, the following usability principles (Bolchini, D., 2007) were applied to the system design.

4.2 Understandable

Since software is used by the user, and the user must be able to understand the corresponding functions of the various elements of the software. If it cannot be understood by the user, then the software needs to be improved or redesigned to match the user's mental model.

4.3 Achievable

The user is the center of interaction design and the interactive elements correspond to the functions that the user needs. Therefore, interactive elements must be user controllable.

For the design of this user interface prototype, all of the user interactive elements are easy to control by users.

4.4 Interoperable.

In order not to make users feel confused and overwhelmed, I specifically designed a hierarchical display of data records by applying this principle and provided users with a drop-down menu for different display level selection.

4.5 Controllable

The software interaction process can be controlled by the user. For example: The user can control the query flow and display mode through all drop-down menus and input boxes. The function execution also can be controlled by the user. For example, the user can review and check the input of the data query conditions on the screen to make sure they are correct.

5 Conclusion

This project overcame several hurdles in using MEPS data for the study of health status, HRQOL, and HEO of cancer patients and their caregivers. First, the raw MEPS data format were converted to CSV format to facilitate data cleaning, sorting, and analyzing. Second, the coded data content was translated to be human readable and used by healthcare researchers. Third, according to the user's needs, the dataset was reorganized and the most important attributes and data elements were identified for a network database. Fourth, a web database schema was created to host the processed and cleaned MEPS dataset. Fifth, a user interface prototype was developed to facilitate users' data retrieval and analysis. The thinkPHP framework was used to connect the front and back end of the entire system.

After presenting the user interface prototype to the two primary users, this system received positive comments in terms of effectiveness and efficiency of data retrieval. Users requirements were met. However, there are still areas that can be improved. For example: The database system I used was MySQL, which makes the back-end database capacity have a lower upper limit. When users demand larger amount of data, the efficiency of our system could be reduced, and the stability would be affected consequently. One of the solution would be to introduce a non-relational database system, which have potential to improve the performance of the entire system

Bibliography

1. Bolchini, D., & Garzotto, F. (2007). Quality of Web Usability Evaluation Methods : An Empirical Study on MiLE +. *WISE 2007 Workshops*, *LNCS*(4832), 481–492. https://doi.org/10.1007/978-3-540-77010-7_47
2. http://www.thinkphp.cn/
3. http://www.phpstudy.net/
4. https://www.wizardmac.com/
5. https://www.navicat.com/en/download/navicat-premium?gclid=CjwKCAjw-6bWBRBiEiwA_K1ZDeUbKkNo22Q2hpdU6h3KenOZXQDM8t855vTLkT65fp_-J33iDoq1gRoCAUwQAvD_BwE
6. Zhi, C. (2015). The Web Database Application System Optimization Research. 2015 Seventh International Conference on Measuring Technology and Mechatronics Automation, 1329–1332. https://doi.org/10.1109/ICMTMA.2015.325
7. Yasmin, F., Rahman, A., & Dzulkifli, M. R. (2016). Development of FKE UiTM Kampus Pasir Gudang Lab Equipment Web Database, (August), 57–61.

8. Blochwitz, C., Joseph, J. M., Backasch, R., Pionteck, T., Werner, S., Heinrich, D., & Groppe, S. (2016). An optimized radix-tree for hardware-accelerated dictionary generation for semantic web databases. 2015 International Conference on ReConFigurable Computing and FPGAs, ReConFig 2015. https://doi.org/10.1109/ReConFig.2015.7393291
9. Sabitha, V. (2017). A Novel Approach for Finding Optimal Search Results from Web Database Using Hybrid Clustering Algorithm, (Icices), 1–4.
10. Zhou, Y., Wu, J., Yu, L., Yu, H., & Tang, Z. (2016). A geohydrologie data visualization framework with an extendable user interface design. 2016 IEEE International Conference on Big Data (Big Data), 2322–2331. https://doi.org/10.1109/BigData.2016.7840865
11. Mulimani, D., Seeri, S. V., Patil, P., & Kulkarni, S. (2016). Experiential Learning Enhancing User Interface Design Skills through Cognitive Action. 2016 IEEE 4th International Conference on MOOCs, Innovation and Technology in Education (MITE), 362–365. https://doi.org/10.1109/MITE.2016.077
12. Pratama, M., Setiawan, N. A., & Wibirama, S. (2017). User Interface Design for Android-based Family Genealogy Social Media.
13. Wardhana, S., Sabariah, M. K., Effendy, V., & St, D. S. K. (2017). User Interface Design Model For Parental Control Application On Mobile Smartphone Using User Centered Design Method, 0(c).Outlook, C. F. (n.d.). with Combinatorial
14. Liu, J., Dwyer, T., Marriott, K., Millar, J., & Haworth, A. (2018). Understanding the Relationship between Interactive Optimisation and Visual Analytics in the Context of Prostate Brachytherapy, 24(1), 319–329.

15. Mehrotra, C. (2017). Scope and Challenges of Visual Analytics : A Survey, 1229 – 1234.
16. Osen, O. L., Vagale, A., & Wang, H. (2017). Data Dimension Reduction for Visual Analytics : A Case Study of Oil-in-Water Detection.
17. Poulymenopoulou, M., & Tsois, A. (2017). Customs Risk Analysis through the ConTraffic Visual Analytics Tool. 2017 European Intelligence and Security Informatics Conference (EISIC), 107–114. https://doi.org/10.1109/EISIC.2017.22
18. Bedek, M. A., Nussbaumer, A., Hillemann, E.-C., & Albert, D. (2017). A Framework for Measuring Imagination in Visual Analytics Systems. 2017 European Intelligence and Security Informatics Conference (EISIC), 151–154. https://doi.org/10.1109/EISIC.2017.31