HERIOT WATT UNIVERSITY

Data Visualization and Analytics – F21 DV

Final Project

Issa Haddad

H00278537

Nov 17, 2017

Table of Contents

Introduction

The main objective of this application is to implement and adapt the D3 General update pattern, which is a method to bind data to DOM elements while simultaneously performing an enter, update, and an exit selection. This is useful to clearly visualize how data is being processed and to clearly track how data is being manipulated. Our aim is to use D3 to build a client side internet application that uses three specific types of layouts and to implement a rational interaction or behavior across all three layouts. Furthermore, we have an objective to build three distinct profiles in which different data is processed and different behaviors are implemented.

Out of the three layouts we must chose, one should be a hierarchical chart, and the other should be a map. The third layout could either be a scatter plot, d3 stack, or a pie chart. Data to be passed and rendered across all three layouts should come from a csv format file. These files compromise textual, relational, and quantitative data. The csv files contain publically made information about universities in the United Kingdom and other organizations.

In order to achieve consistency and coherence, we must use an architectural pattern along with design pattern to separate the major components of the data and create logical structure. Many techniques can be taken into consideration; such techniques include encapsulation. Our understanding of the overall d3 general update pattern functionality will play a huge role in allowing us to successfully complete and carry out a project of this type.

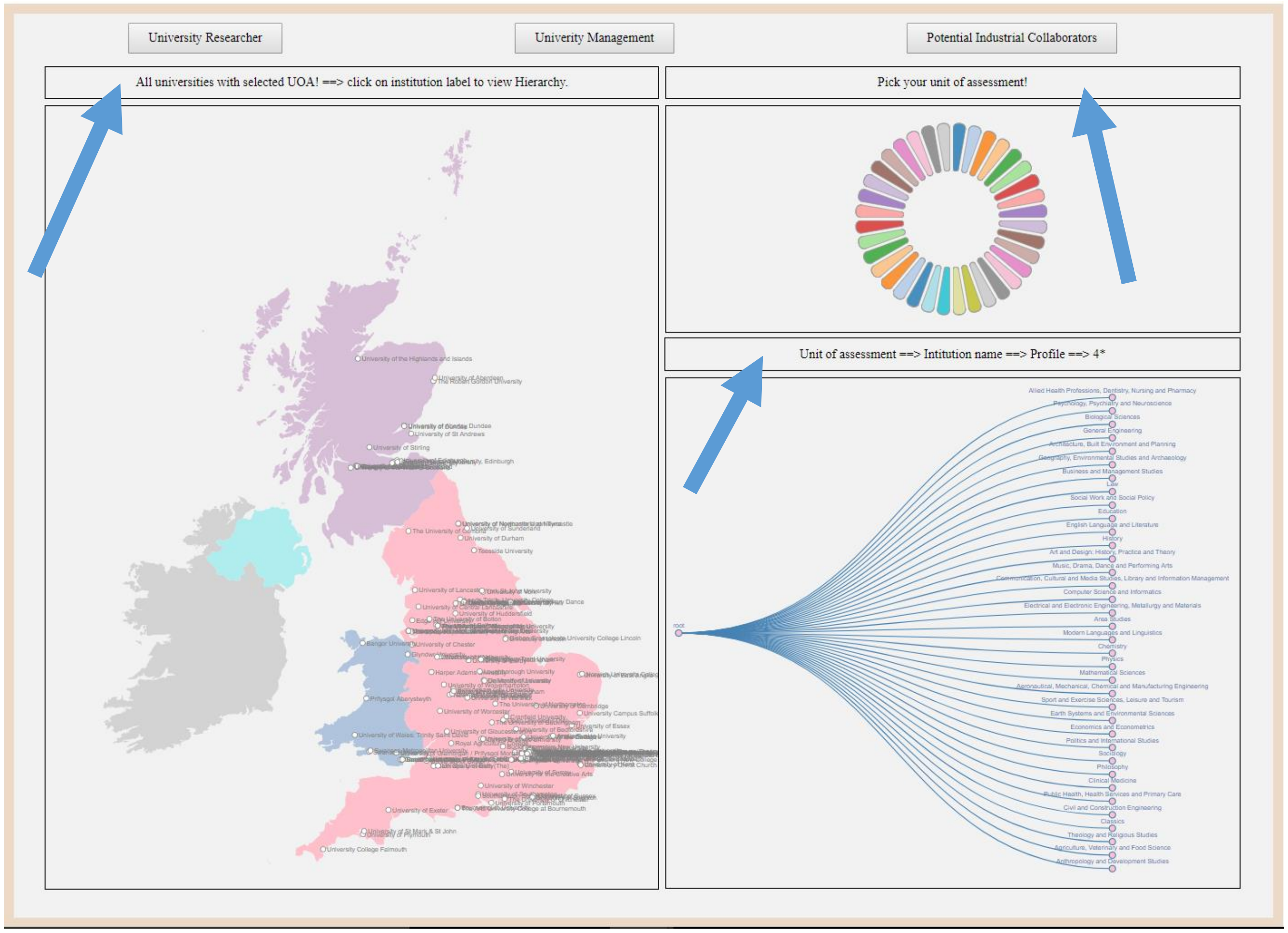
In the following paragraphs, I'm going to explain in details about the different aspects of this project and will give you a step by step walk through of my application and how I implemented it. I will also provide a graphical user manual on how my application runs.

Graphical User Manual

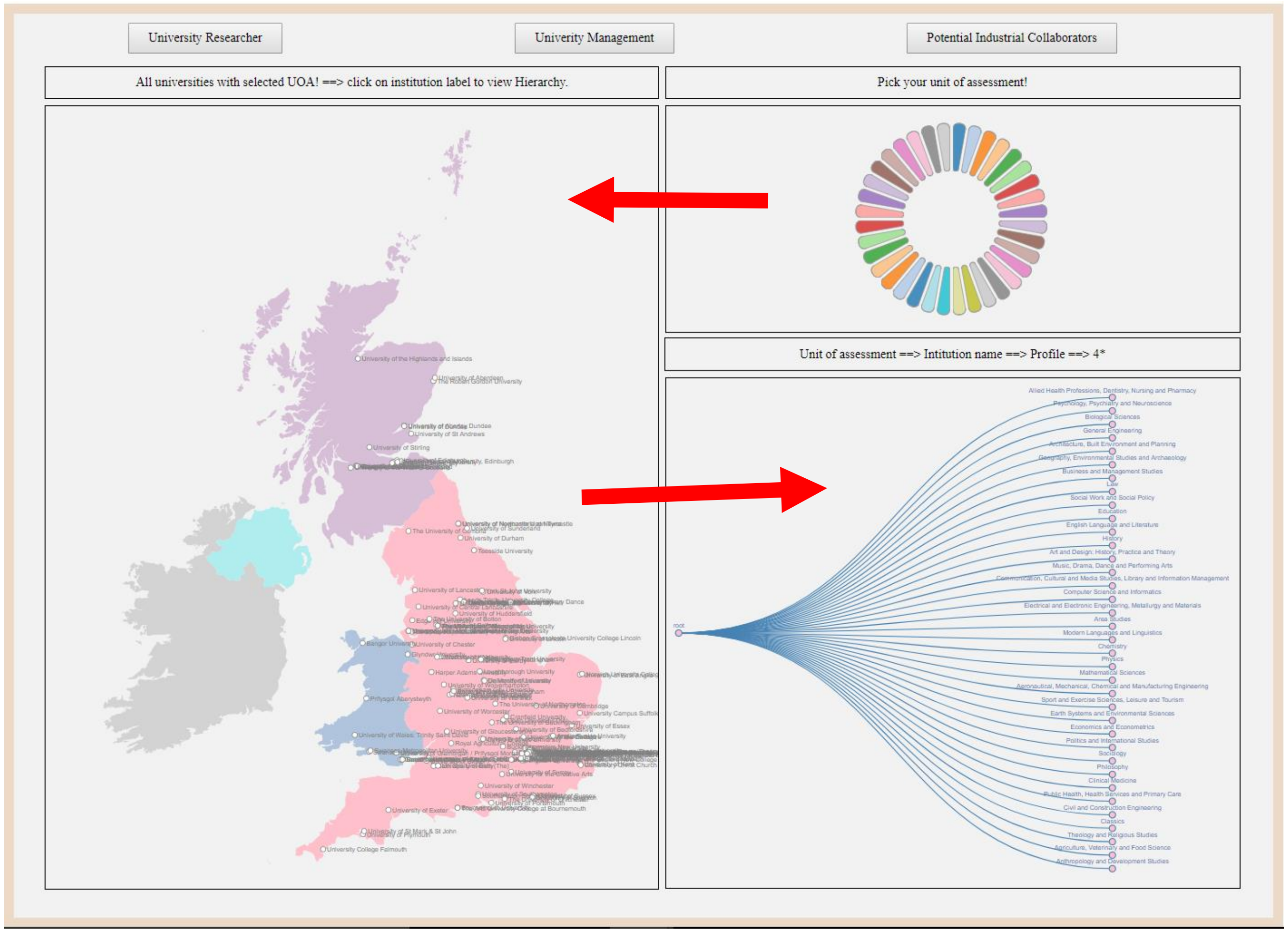
University Researcher

➢ This profile enables its user to choose a certain UOA from the pie chart. It then renders all universities that have that particular UOA on the map. It then enables its users to pick their desired institution from the map. It then provides useful information in the form of hierarchical graph to show the 4* rating for all profiles of the chosen UOA and institution.

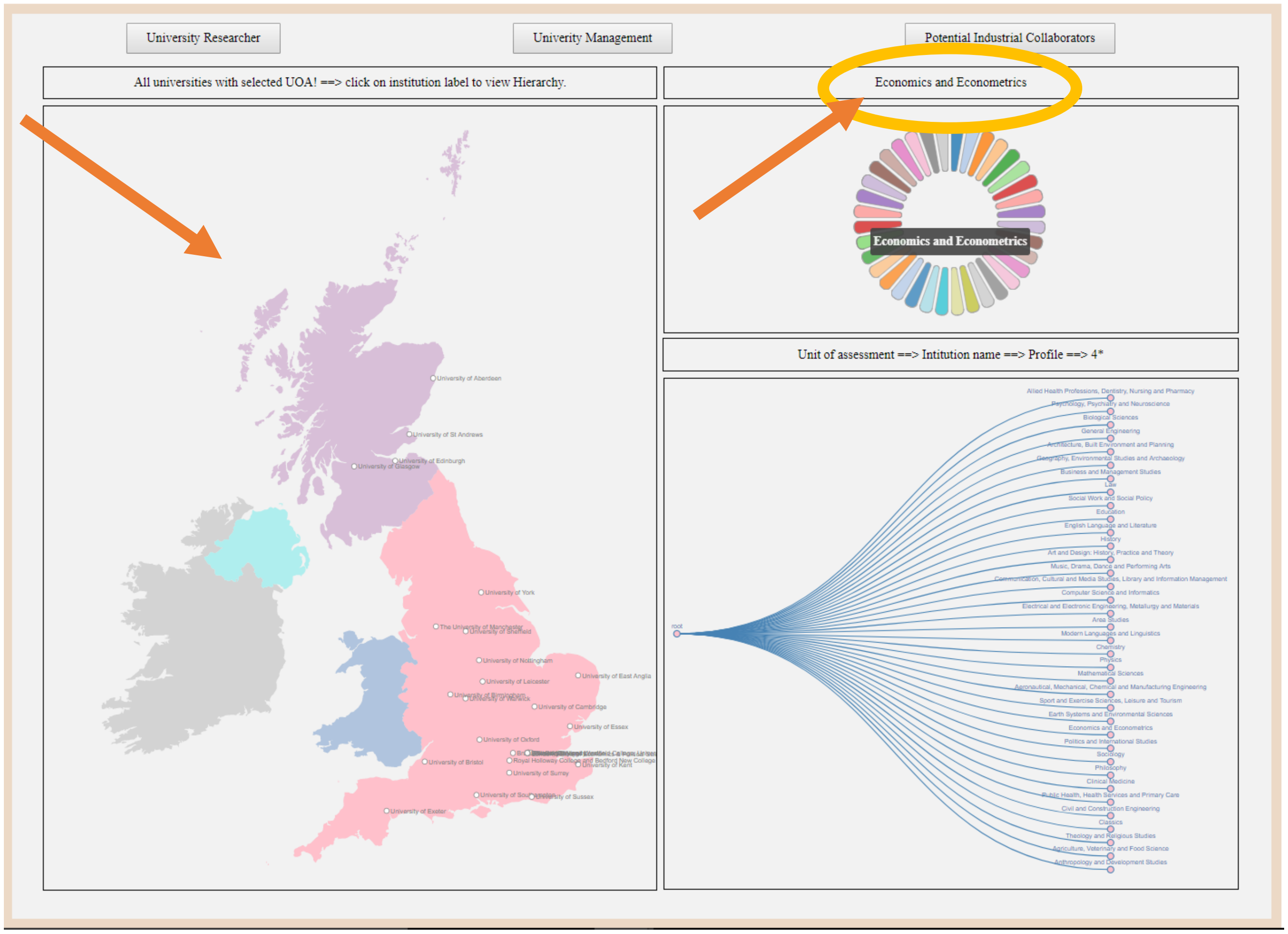
- Starting page.
- GUI tells user to pick UOA in pie chart.
- Labels are removed and replaced by tool tip to keep pie chart clean and simple instead of overcrowded.
- Text above each chart is a user guide, it tells him what to do, and also shows him his current standing.


University Researcher
Univerity Management
Potential Industrial Collaborators
All universities with selected UOA! ==> click on institution label to view Hierarchy.
Pick your unit of assessment!
Unit of assessment ==> Intitution name ==> Profile ==> 4*
root
Allied Health Professions, Dentistry, Nursing and Pharmacy
Psychology, Psychiatry and Neuroscience
Biological Sciences
General Engineering
Architecture, Built Environment and Planning
Geography, Environmental Studies and Archaeology
Business and Management Studies
Law
Social Work and Social Policy
Education
English Language and Literature
History
Art and Design: History, Practice and Theory
Music, Drama, Dance and Performing Arts
Communication, Cultural and Media Studies, Library and Information Management
Computer Science and Informatics
Electrical and Electronic Engineering, Metallurgy and Materials
Area Studies
Modern Languages and Linguistics
Chemistry
Physics
Mathematical Sciences
Aeronautical, Mechanical, Chemical and Manufacturing Engineering
Sport and Exercise Sciences, Leisure and Tourism
Earth Systems and Environmental Sciences
Economics and Econometrics
Politics and International Studies
Sociology
Philosophy
Clinical Medicine
Public Health, Health Services and Primary Care
Civil and Construction Engineering
Classics
Theology and Religious Studies
Agriculture, Veterinary and Food Science
Anthropology and Development Studies

- The flow of rendering and interactions is displayed as red arrows in the following graph.


University Researcher
Univerity Management
Potential Industrial Collaborators
All universities with selected UOA! ==> click on institution label to view Hierarchy.
Pick your unit of assessment!
Unit of assessment ==> Intitution name ==> Profile ==> 4*
root
Allied Health Professions, Dentistry, Nursing and Pharmacy
Psychology, Psychiatry and Neuroscience
Biological Sciences
General Engineering
Architecture, Built Environment and Planning
Geography, Environmental Studies and Archaeology
Business and Management Studies
Law
Social Work and Social Policy
Education
English Language and Literature
History
Art and Design: History, Practice and Theory
Music, Drama, Dance and Performing Arts
Communication, Cultural and Media Studies, Library and Information Management
Computer Science and Informatics
Electrical and Electronic Engineering, Metallurgy and Materials
Area Studies
Modern Languages and Linguistics
Chemistry
Physics
Mathematical Sciences
Aeronautical, Mechanical, Chemical and Manufacturing Engineering
Sport and Exercise Sciences, Leisure and Tourism
Earth Systems and Environmental Sciences
Economics and Econometrics
Politics and International Studies
Sociology
Philosophy
Clinical Medicine
Public Health, Health Services and Primary Care
Civil and Construction Engineering
Classics
Theology and Religious Studies
Agriculture, Veterinary and Food Science
Anthropology and Development Studies

- You can see here an illustration of how tooltip works with the pie chart.
- Clicking on a UOA would note down the selected UOA above the pie chart, this is to keep the user focused in case he forgot what he clicked.
- After clicking on UOA, the map will render all the universities that teach that particular UOA.


University Researcher
Univerity Management
Potential Industrial Collaborators

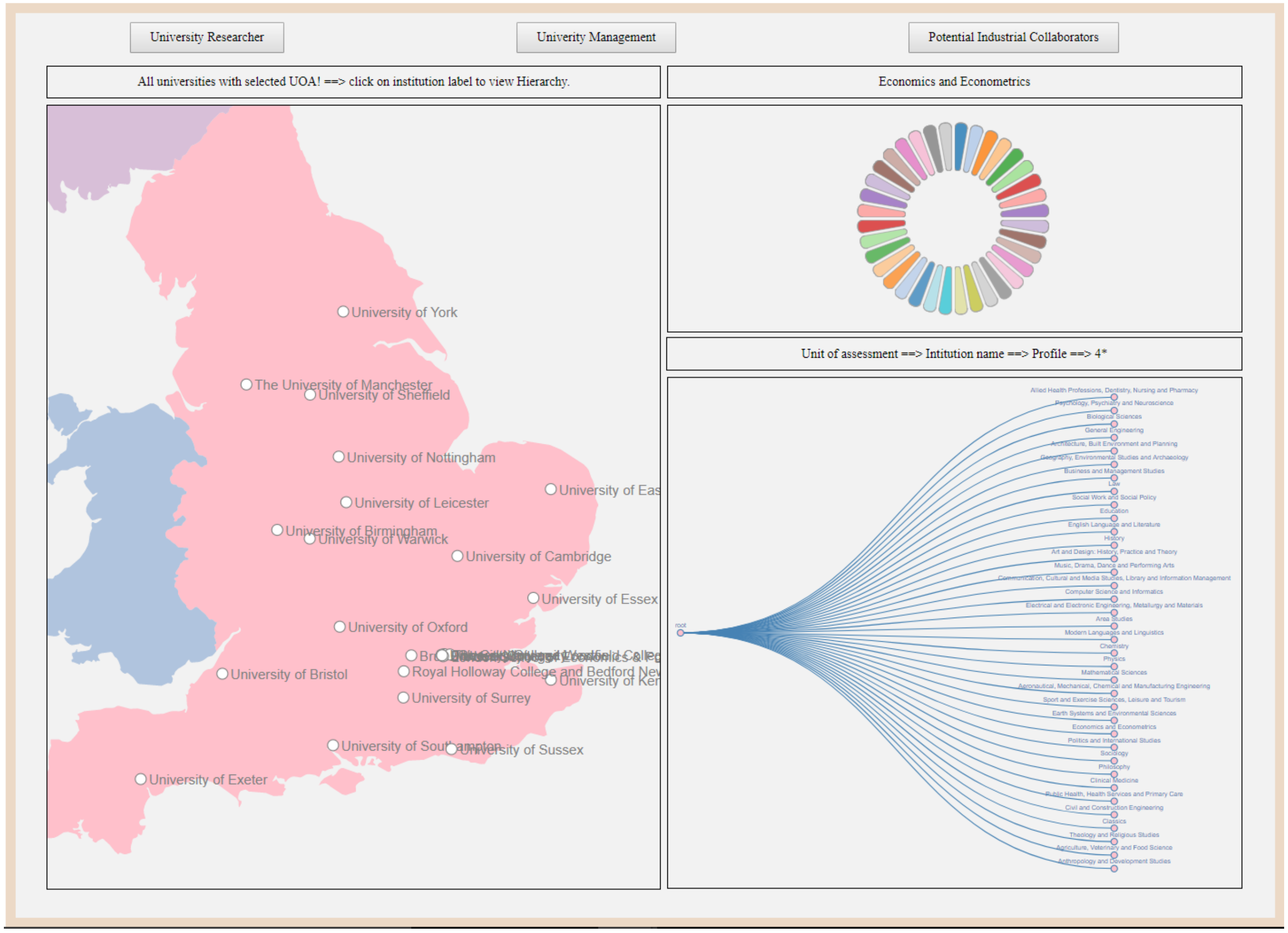
All universities with selected UOA! ==> click on institution label to view Hierarchy.

Economics and Econometrics

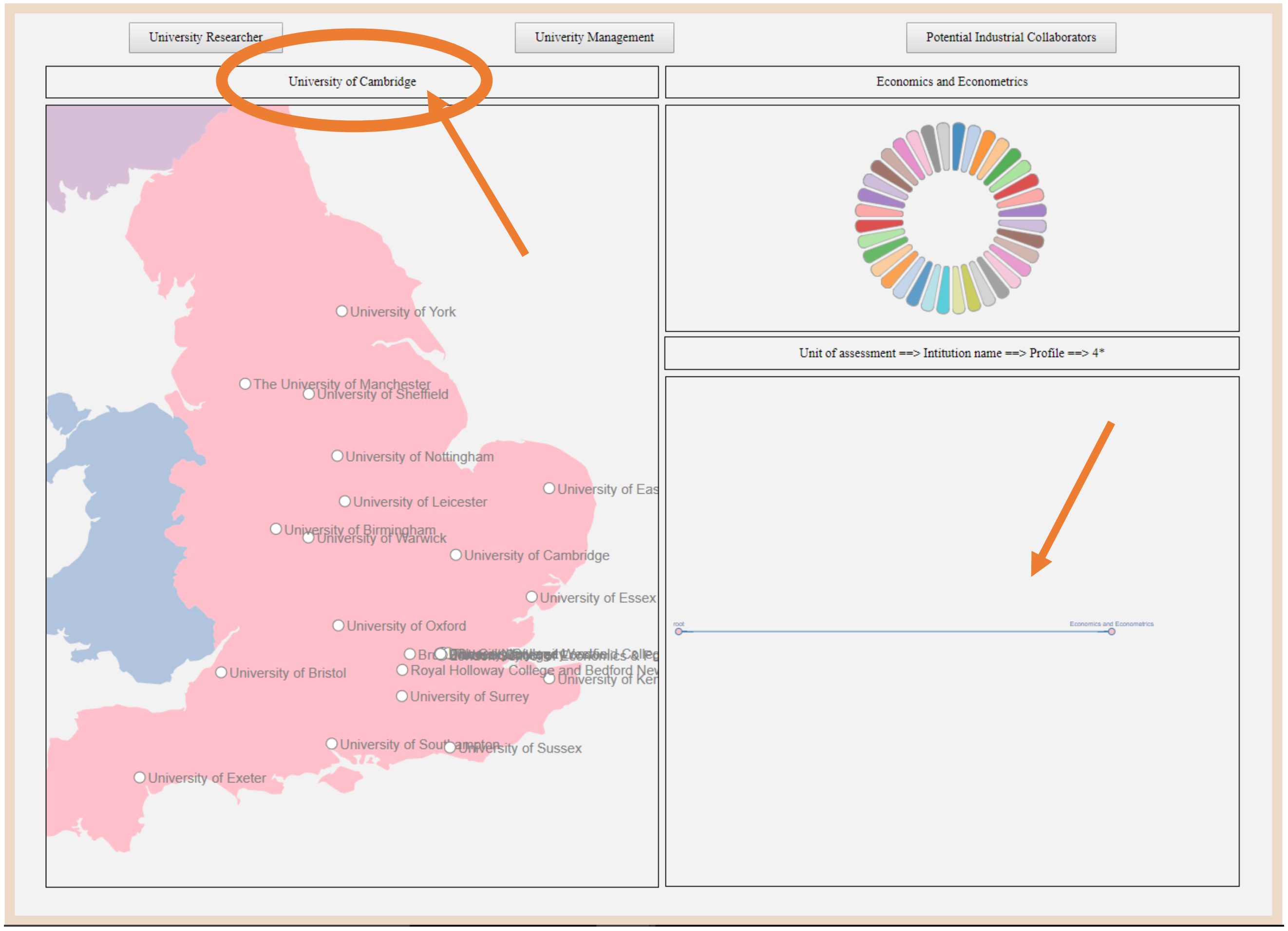
Unit of assessment ==> Intitution name ==> Profile ==> 4*

University of Aberdeen
University of St Andrews
University of Edinburgh
University of Glasgow
University of York
The University of Manchester
University of Sheffield
University of Nottingham
University of East Anglia
University of Leicester
University of Birmingham
University of Warwick
University of Cambridge
University of Essex
University of Oxford
University of Bristol
Royal Holloway College and Bedford New College
University of Kent
University of Surrey
University of Sussex
University of Exeter
Economics and Econometrics
root
Allied Health Professions, Dentistry, Nursing and Pharmacy
Psychology, Psychiatry and Neuroscience
Biological Sciences
General Engineering
Architecture, Built Environment and Planning
Geography, Environmental Studies and Archaeology
Business and Management Studies
Law
Social Work and Social Policy
Education
English Language and Literature
History
Art and Design: History, Practice and Theory
Music, Drama, Dance and Performing Arts
Communication, Cultural and Media Studies, Library and Information Management
Computer Science and Informatics
Electrical and Electronic Engineering, Metallurgy and Materials
Area Studies
Modern Languages and Linguistics
Chemistry
Physics
Mathematical Sciences
Aeronautical, Mechanical, Chemical and Manufacturing Engineering
Sport and Exercise Sciences, Leisure and Tourism
Earth Systems and Environmental Sciences
Economics and Econometrics
Politics and International Studies
Sociology
Philosophy
Clinical Medicine
Public Health, Health Services and Primary Care
Civil and Construction Engineering
Classics
Theology and Religious Studies
Agriculture, Veterinary and Food Science
Anthropology and Development Studies

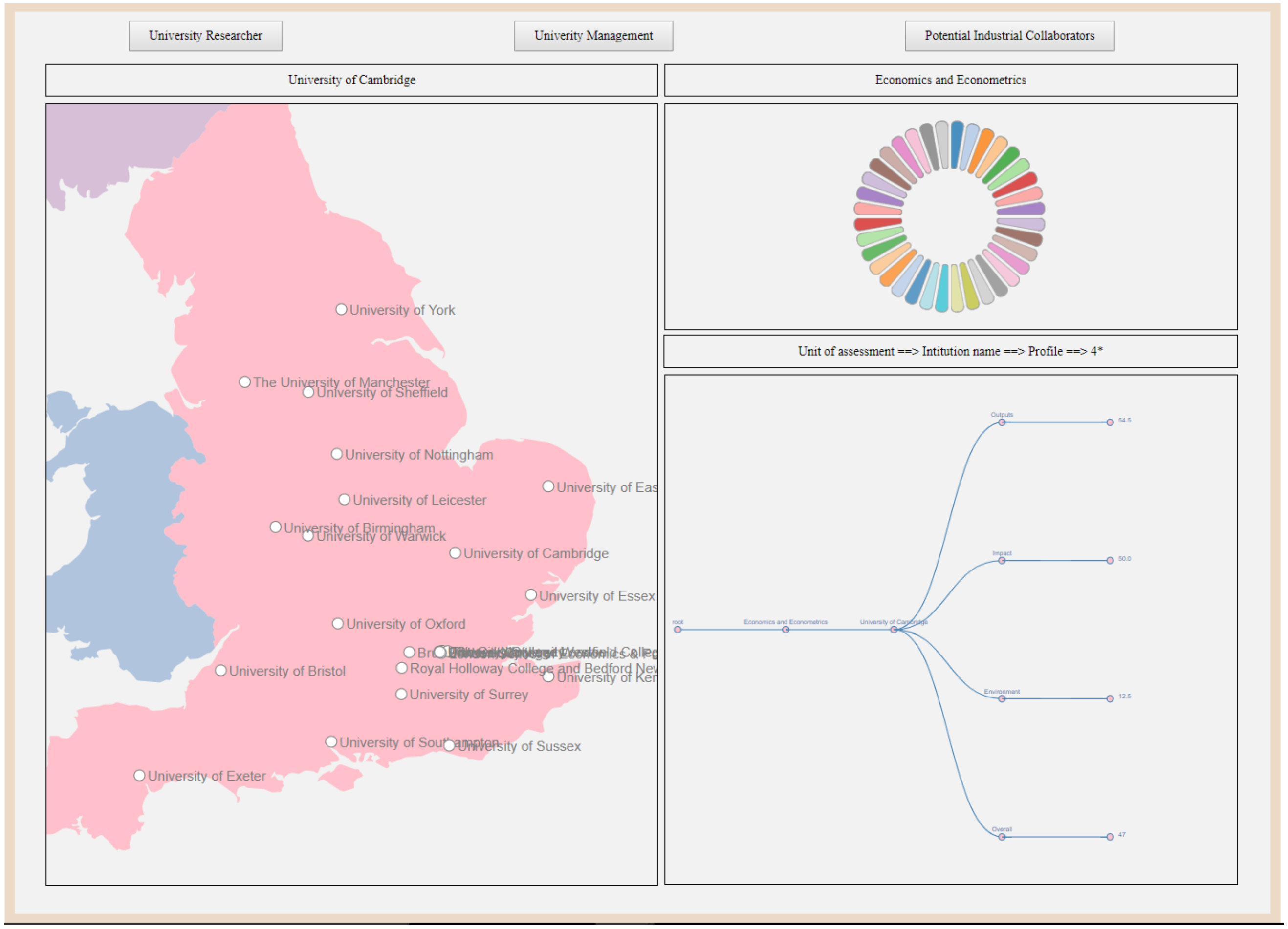
- Perhaps the institution name labels are too tiny to read.
- I implemented a zoom-able and movable map across all profiles, as shown below.


University Researcher
Univerity Management
Potential Industrial Collaborators
All universities with selected UOA! ==> click on institution label to view Hierarchy.
Economics and Econometrics
Unit of assessment ==> Intitution name ==> Profile ==> 4*
University of York
The University of Manchester
University of Sheffield
University of Nottingham
University of Eas
University of Leicester
University of Birmingham
University of Warwick
University of Cambridge
University of Essex
University of Oxford
University of Bristol
Royal Holloway College and Bedford Nev
University of Ker
University of Surrey
University of Southampton
University of Sussex
University of Exeter
root
Allied Health Professions, Dentistry, Nursing and Pharmacy
Psychology, Psychiatry and Neuroscience
Biological Sciences
General Engineering
Architecture, Built Environment and Planning
Geography, Environmental Studies and Archaeology
Business and Management Studies
Law
Social Work and Social Policy
Education
English Language and Literature
History
Art and Design: History, Practice and Theory
Music, Drama, Dance and Performing Arts
Communication, Cultural and Media Studies, Library and Information Management
Computer Science and Informatics
Electrical and Electronic Engineering, Metallurgy and Materials
Area Studies
Modern Languages and Linguistics
Chemistry
Physics
Mathematical Sciences
Aeronautical, Mechanical, Chemical and Manufacturing Engineering
Sport and Exercise Sciences, Leisure and Tourism
Earth Systems and Environmental Sciences
Economics and Econometrics
Politics and International Studies
Sociology
Philosophy
Clinical Medicine
Public Health, Health Services and Primary Care
Civil and Construction Engineering
Classics
Theology and Religious Studies
Agriculture, Veterinary and Food Science
Anthropology and Development Studies

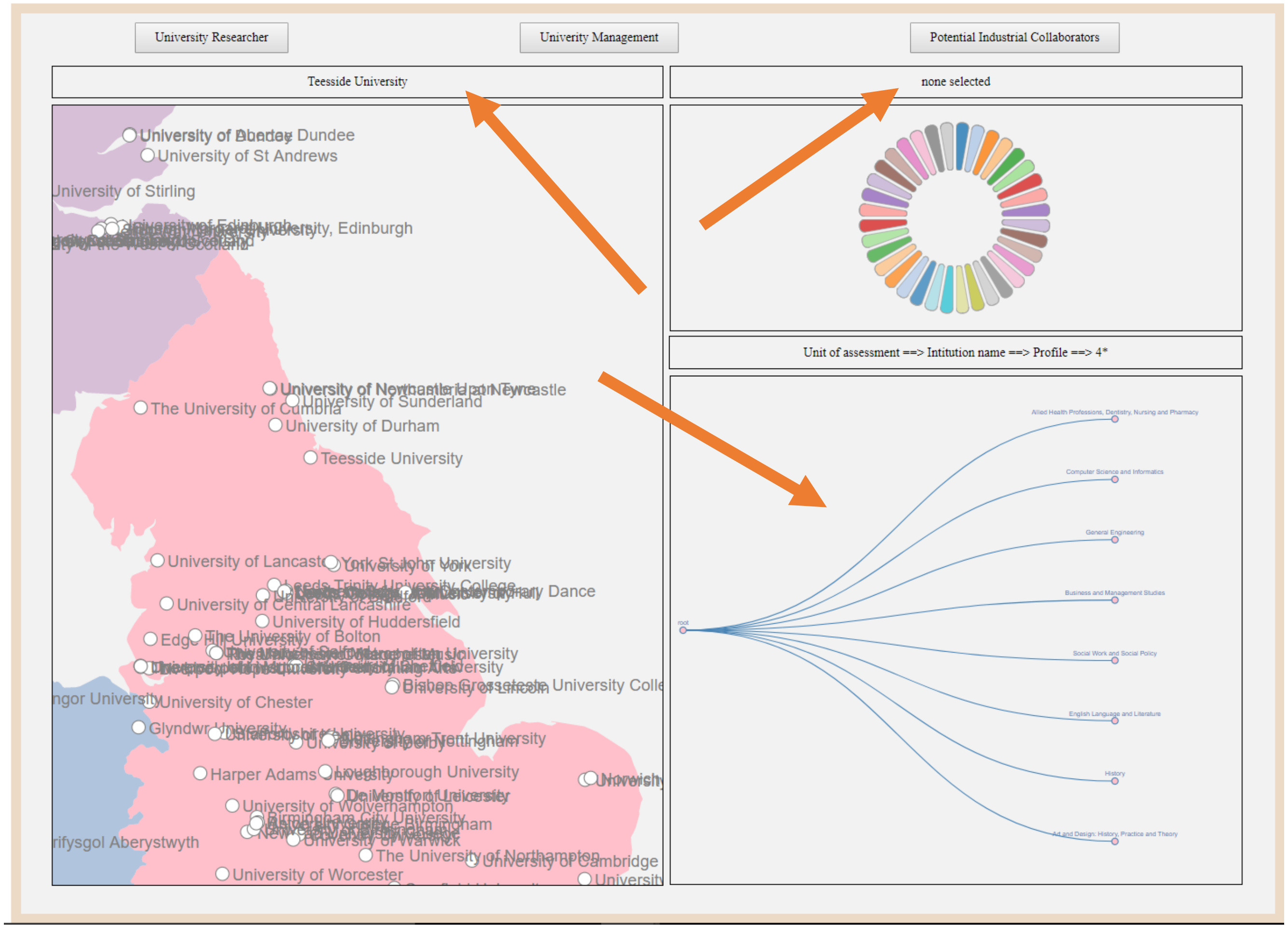
- After the map has rendered all the universities that have the selected UOA, it is time to chose your university of preference.
- The map has a click function on the labels or towns.
- Clicking on a university would note down the instution name above the map, and will also rereder the herarchical graph as shown below. The hierachy graphy is collapsable across all profiles.



- Expanding the tree would show you the 4* rating for all profiles of the selected UOA and the selected university.

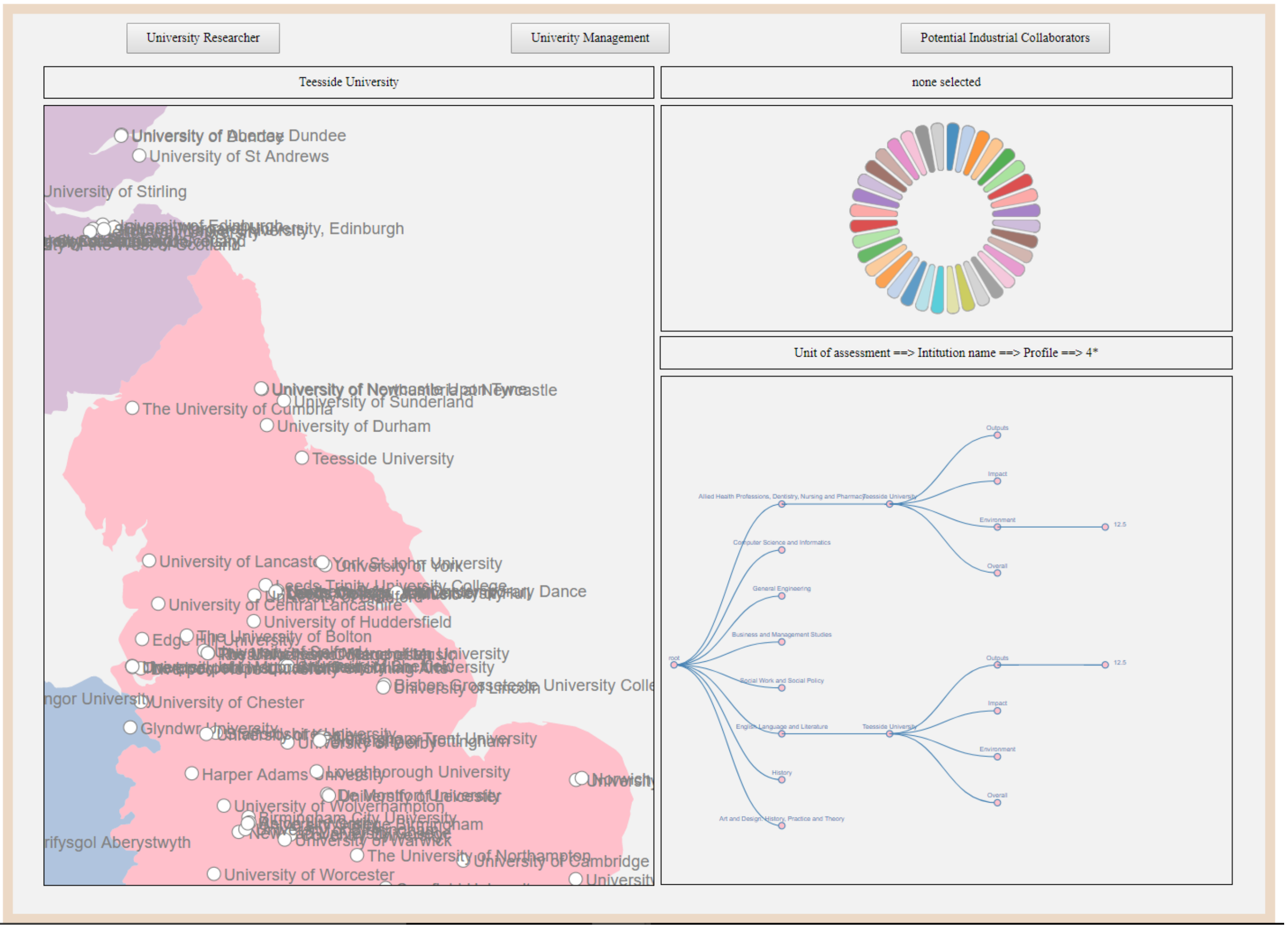


- Clicking on the same UOA would remove the selected UOA.
- This time I decide to click on another university called: Teesside University.
- This would render the hierarchy chart again. This time the hierarchy would display all the unit of assessments provided by the selected university on the first level of the hierarchy. Previously when the UOA was selected, the hierarchy would show only that particular UOA on the first level.



University Researcher
Univerity Management
Potential Industrial Collaborators
Teesside University
none selected
University of Dundee
University of St Andrews
University of Stirling
University of Cumbria
University of Sunderland
University of Durham
Teesside University
University of Lancaster
University of York
University of Central Lancashire
University of Huddersfield
University of Bolton
University of Chester
Loughborough University
University of Wolverhampton
Birmingham City University
University of Warwick
University of Worcester
University of Cambridge
Unit of assessment ==> Intitution name ==> Profile ==> 4*
root
Allied Health Professions, Dentistry, Nursing and Pharmacy
Computer Science and Informatics
General Engineering
Business and Management Studies
Social Work and Social Policy
English Language and Literature
History
Art and Design: History, Practice and Theory

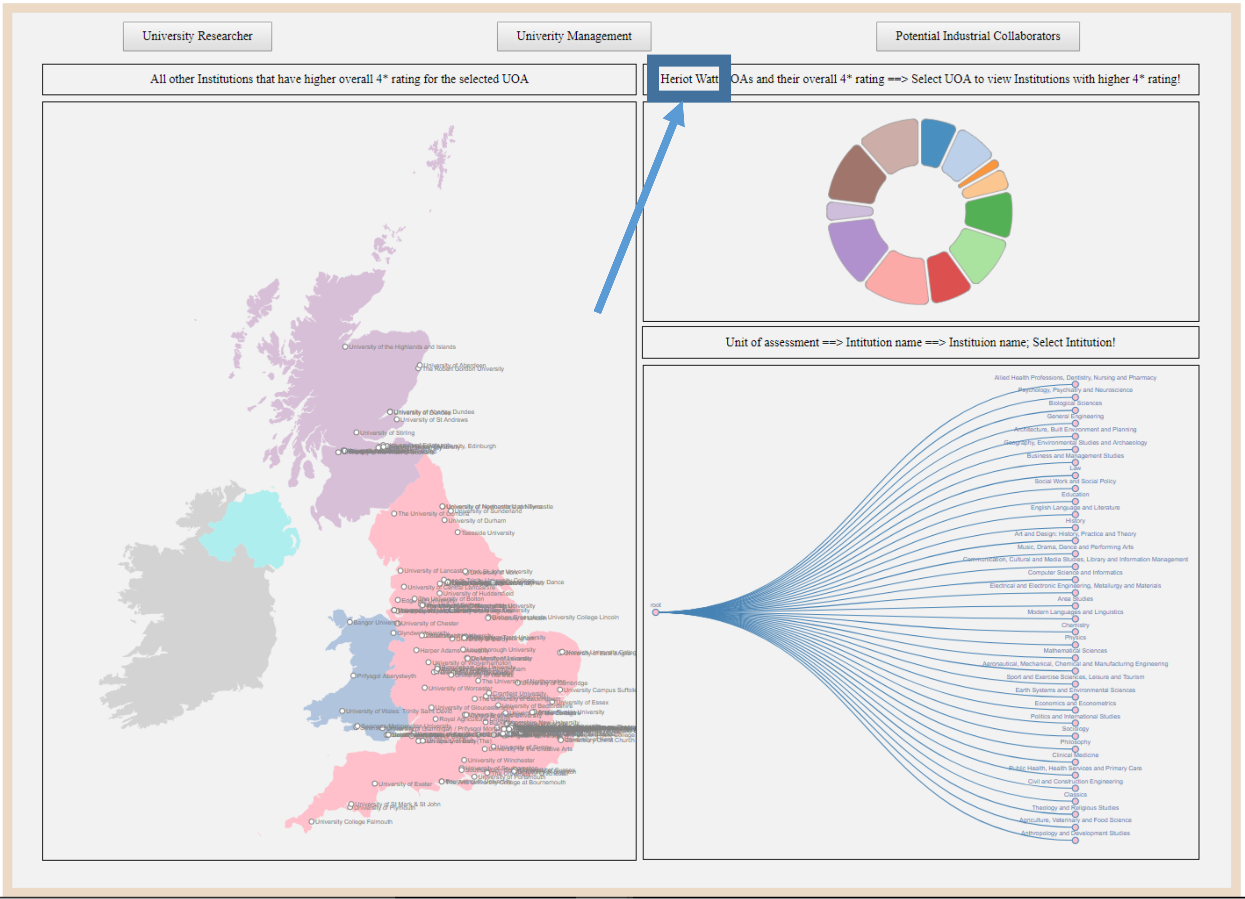
- Expanding the hierarchy would show the university name on the second level, and the profile on the third level.
- It will also show the 4* rating for each profile on the leaf node.



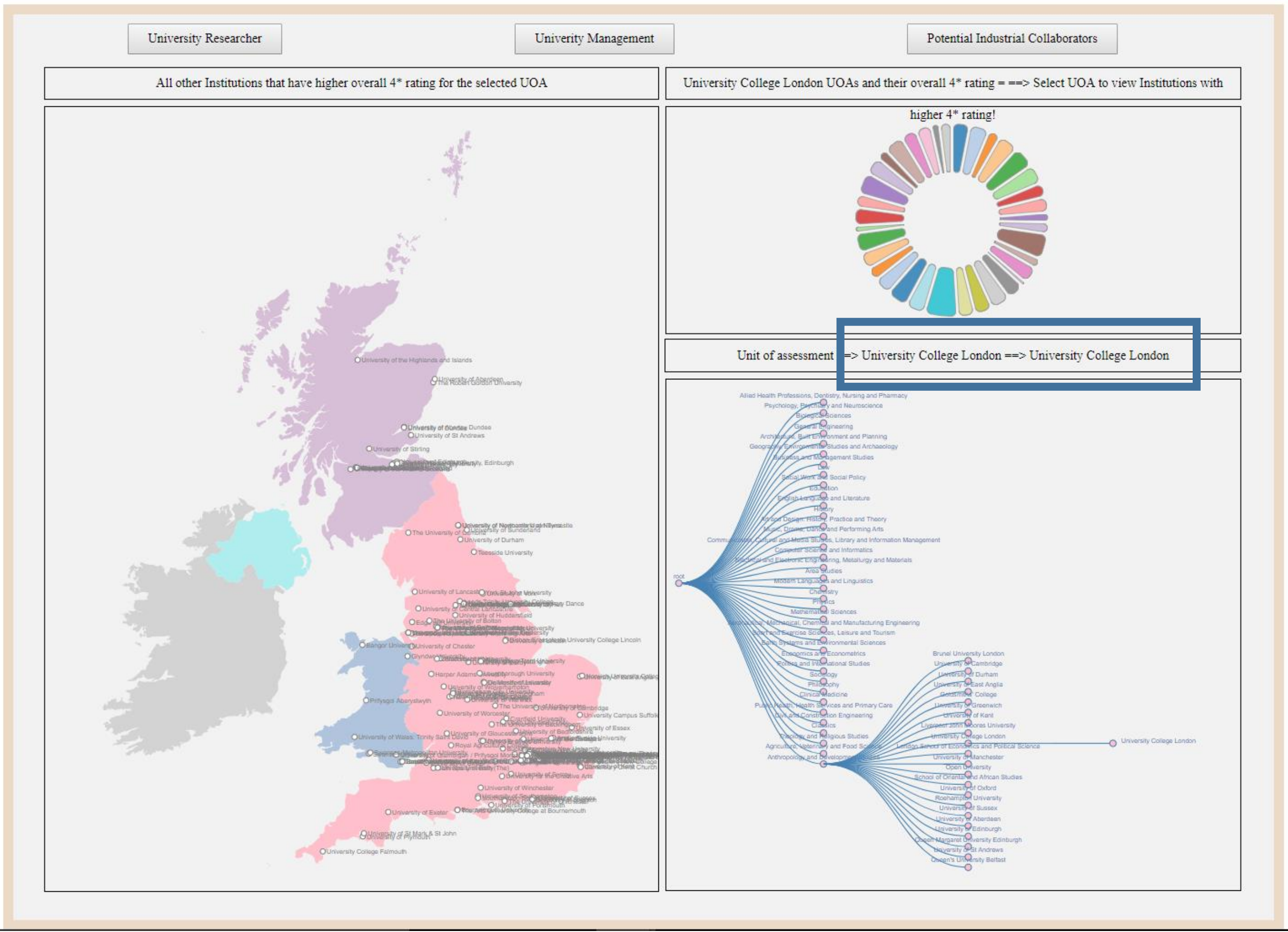
University Management

➢ This profile enables university management to pick their institution of choice via clicking on the hierarchy graph. It then provides useful information in the form of a pie chart that shows all the units of assessment in that particular institution and their corresponding overall 4* rating. The pie chart is weighted, that means if a particular UOA has a higher 4* rating than another UOA, the equivalent arc would be bigger. The profile doesn't stop there, clicking on a preferred UOA form the pie chart would display all the universities where that particular UOA has a higher overall 4* rating.

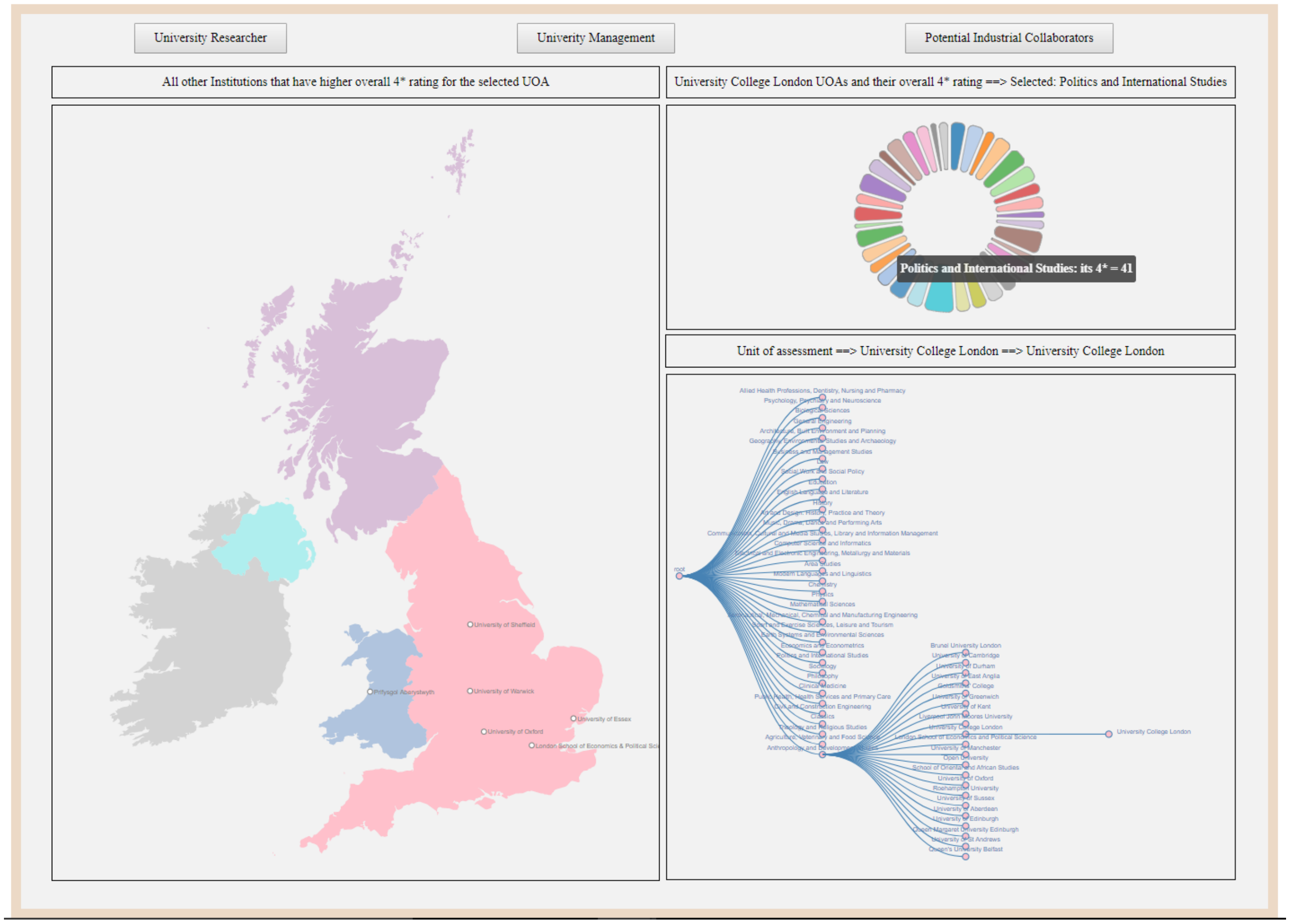
- Starting page.
- Pie chart is defaulted to show all UOA and their corresponding 4* rating for Heriot Watt university.
- GUI tells user to select instition from hierarchy chart. It also tells user to select UOA to view all institutions with higher overall 4*.
- The hierarchy shown has levels in the form of UOA → Insitution name → instituion name.
- The reason for that structure is due to the fact that viewing an instiution on the second level makes the chart less crowded than it would be if the institution was directly on the first level.
- The reason for another third institiion level is due to the fact that if a user presses on the wrong instiution name on the second level, he can undo and reshrink the graph to prevent the application from selecting a wrong instiution.


University Researcher
Univerity Management
Potential Industrial Collaborators
All other Institutions that have higher overall 4* rating for the selected UOA
Heriot Watt
OAs and their overall 4* rating ==> Select UOA to view Institutions with higher 4* rating!
Unit of assessment ==> Intitution name ==> Instituion name; Select Intitution!
root
Allied Health Professions, Dentistry, Nursing and Pharmacy
Psychology, Psychiatry and Neuroscience
Biological Sciences
General Engineering
Architecture, Built Environment and Planning
Geography, Environmental Studies and Archaeology
Business and Management Studies
Law
Social Work and Social Policy
Education
English Language and Literature
History
Art and Design: History, Practice and Theory
Music, Drama, Dance and Performing Arts
Communication, Cultural and Media Studies, Library and Information Management
Computer Science and Informatics
Electrical and Electronic Engineering, Metallurgy and Materials
Area Studies
Modern Languages and Linguistics
Chemistry
Physics
Mathematical Sciences
Aeronautical, Mechanical, Chemical and Manufacturing Engineering
Sport and Exercise Sciences, Leisure and Tourism
Earth Systems and Environmental Sciences
Economics and Econometrics
Politics and International Studies
Sociology
Philosophy
Clinical Medicine
Public Health, Health Services and Primary Care
Civil and Construction Engineering
Classics
Theology and Religious Studies
Agriculture, Veterinary and Food Science
Anthropology and Development Studies

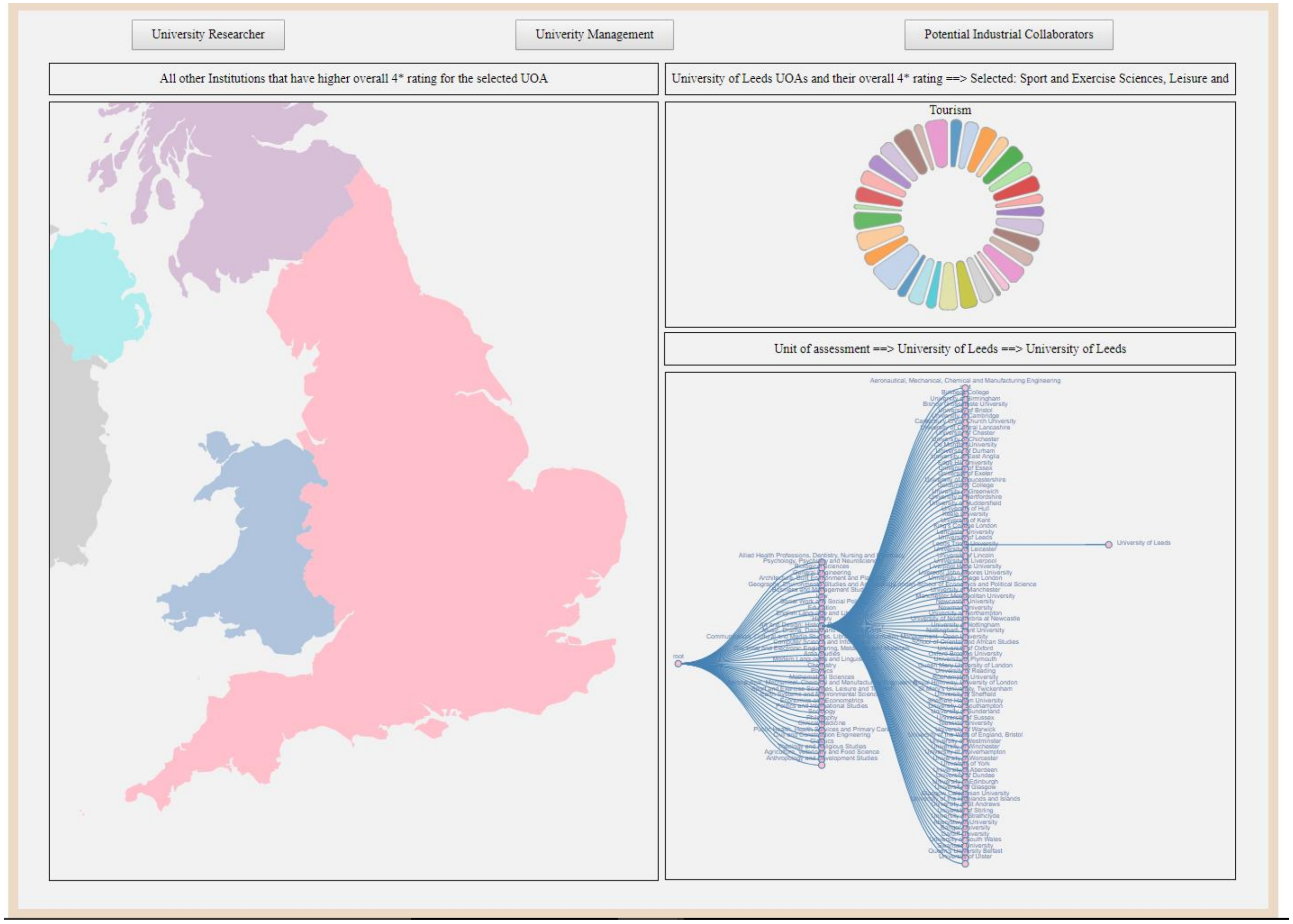
- An illustraion on how a user selects an instituion from the hierarchy is hown below.
- Text above the hierarchy also tells the user his position, and the fact that he selected that particular university. Eg: in the graph below its University of college london.
- The pie chart is also rendered at this stage to show all units of assessment in that university and their corrseponding overall 4* rating.


University Researcher
Univerity Management
Potential Industrial Collaborators
All other Institutions that have higher overall 4* rating for the selected UOA
University College London UOAs and their overall 4* rating = ==> Select UOA to view Institutions with higher 4* rating!
Unit of assessment ==> University College London ==> University College London
root
University College London

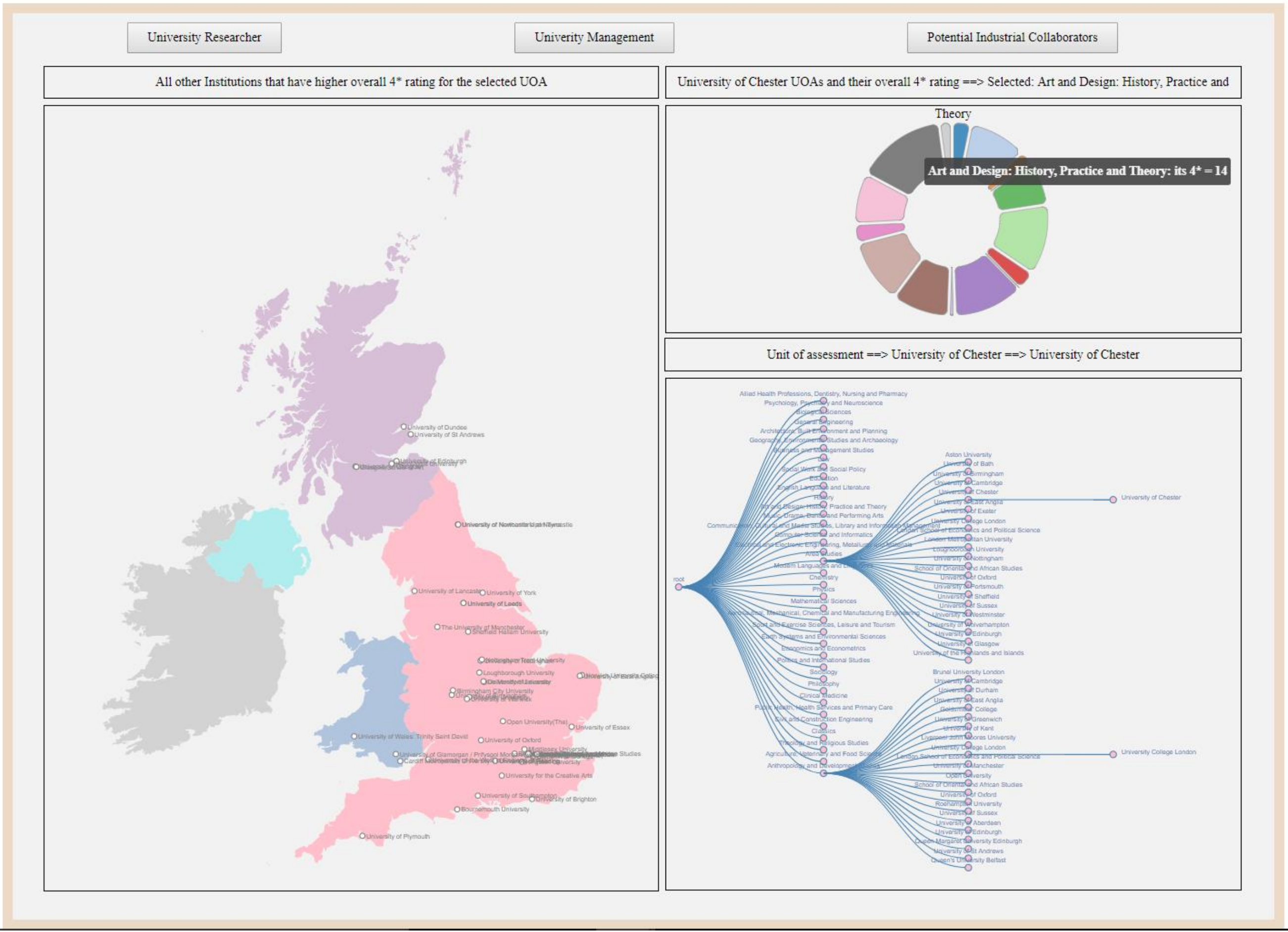
- You can see in the next graph how the tooltip also works in the pie chart. It displays the UOA and its corresponding overall 4* rating.
- Clicking on the particular UOA would display on the map only the instutions that have a higher 4* rating for that particular UOA.
- You can see that for Politics and international studies in University college london, it ranks 7th place in the UK with only 6 universities with a higher overall 4* rating.


University Researcher
Univerity Management
Potential Industrial Collaborators
All other Institutions that have higher overall 4* rating for the selected UOA
University College London UOAs and their overall 4* rating ==> Selected: Politics and International Studies
Politics and International Studies: its 4* = 41
Unit of assessment ==> University College London ==> University College London
University of Sheffield
Prifysgol Aberystwyth
University of Warwick
University of Essex
University of Oxford
London School of Economics & Political Sci
root
Allied Health Professions, Dentistry, Nursing and Pharmacy
Psychology, Psychiatry and Neuroscience
Biological Sciences
General Engineering
Architecture, Built Environment and Planning
Geography, Environmental Studies and Archaeology
Business and Management Studies
Law
Social Work and Social Policy
Education
English Language and Literature
History
Art and Design: History, Practice and Theory
Music, Drama, Dance and Performing Arts
Communication, Cultural and Media Studies, Library and Information Management
Computer Science and Informatics
Electrical and Electronic Engineering, Metallurgy and Materials
Area Studies
Modern Languages and Linguistics
Chemistry
Physics
Mathematical Sciences
Aeronautical, Mechanical, Chemical and Manufacturing Engineering
Sport and Exercise Sciences, Leisure and Tourism
Earth Systems and Environmental Sciences
Economics and Econometrics
Politics and International Studies
Sociology
Philosophy
Clinical Medicine
Public Health, Health Services and Primary Care
Civil and Construction Engineering
Classics
Theology and Religious Studies
Agriculture, Veterinary and Food Science
Anthropology and Development Studies
Brunel University London
University of Cambridge
University of Durham
University of East Anglia
Goldsmiths' College
University of Greenwich
University of Kent
Liverpool John Moores University
University College London
London School of Economics and Political Science
University of Manchester
Open University
School of Oriental and African Studies
University of Oxford
Roehampton University
University of Sussex
University of Aberdeen
University of Edinburgh
Queen Margaret University Edinburgh
University of St Andrews
Queen's University Belfast
University College London

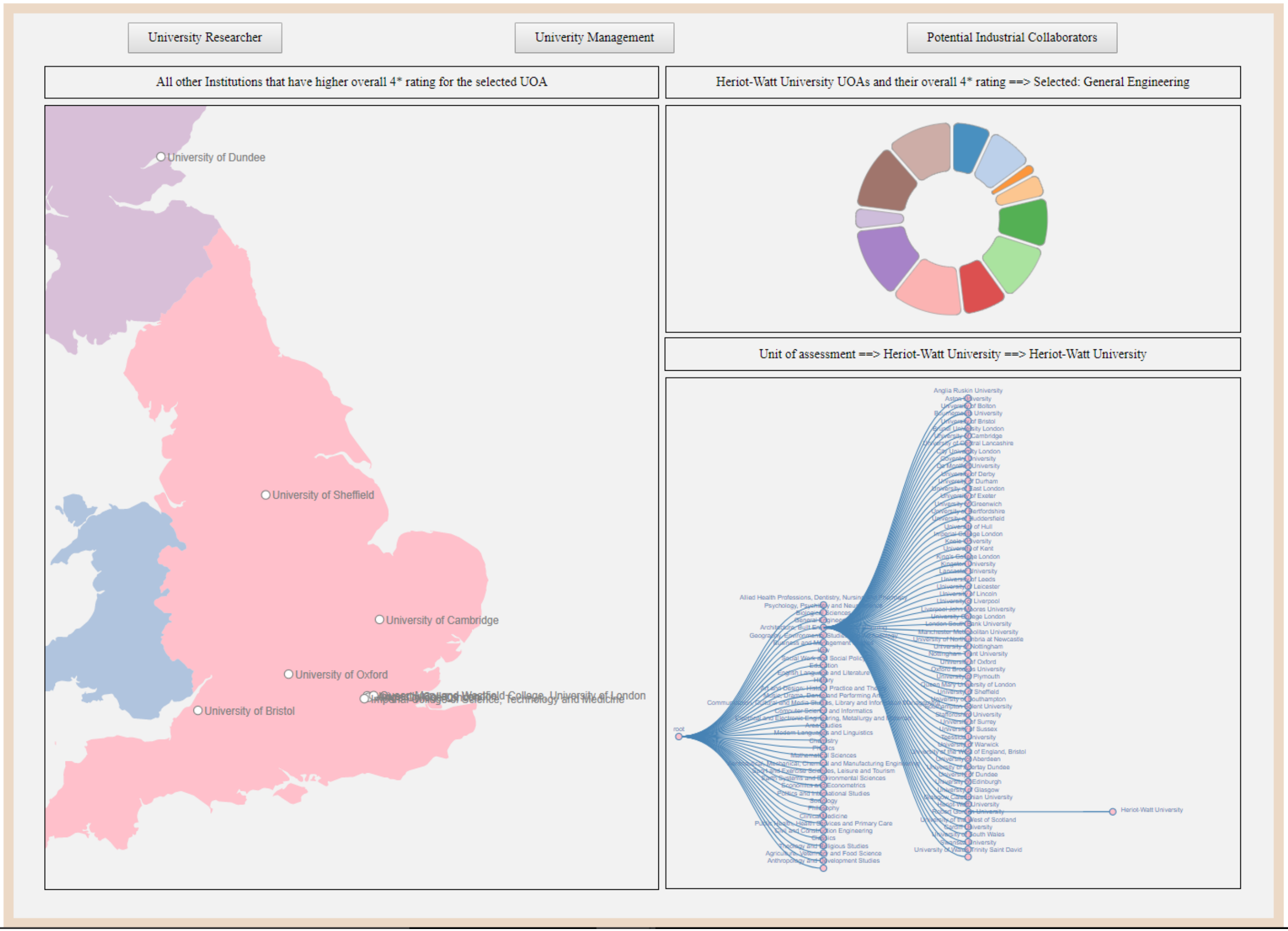
- Below is an example when the chosen UOA for a particular university is the best.
- The map disaplays no other other university, this is because the 4* rating for the current UOA and for that particular university is higher than all other universities in the UK.
- Example chosen below is when:
    - University = University of Leeds
    - UOA = Tourism


University Researcher
Univerity Management
Potential Industrial Collaborators
All other Institutions that have higher overall 4* rating for the selected UOA
University of Leeds UOAs and their overall 4* rating ==> Selected: Sport and Exercise Sciences, Leisure and
Tourism
Unit of assessment ==> University of Leeds ==> University of Leeds
root
University of Leeds

- Another example where institution is “University of Chester” & UOA is “Art and Design: History practice and Theory”.


University Researcher
Univerity Management
Potential Industrial Collaborators
All other Institutions that have higher overall 4* rating for the selected UOA
University of Chester UOAs and their overall 4* rating ==> Selected: Art and Design: History, Practice and Theory
Art and Design: History, Practice and Theory: its 4* = 14
Unit of assessment ==> University of Chester ==> University of Chester
root
University of Chester
University College London

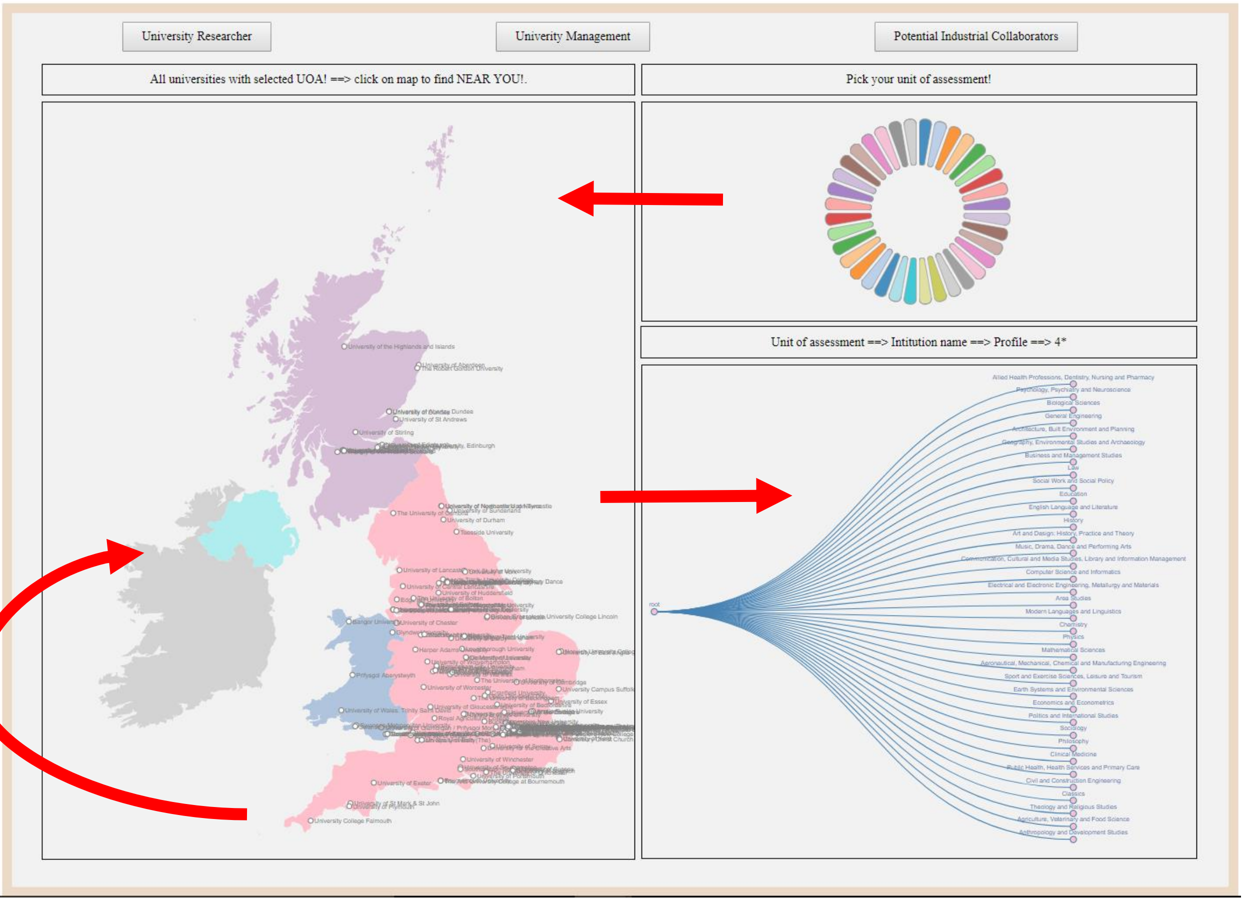
- Heriot watt university for instance is ranked 10th in the UK for General Engineering with only 9 universities with higher overall 4* ranking.


University Researcher
Univerity Management
Potential Industrial Collaborators
All other Institutions that have higher overall 4* rating for the selected UOA
Heriot-Watt University UOAs and their overall 4* rating ==> Selected: General Engineering
Unit of assessment ==> Heriot-Watt University ==> Heriot-Watt University
University of Dundee
University of Sheffield
University of Cambridge
University of Oxford
University of Bristol
Queen Mary and Westfield College, University of London
Imperial College of Science, Technology and Medicine
root
Heriot-Watt University

Potential industrial collaborators

- Even tho this profile may look the same as the others, it performs a completely different functionality. The profile enables the user to find information about universities with a particular UOA within 80 kilometer radius from his/her location.

- Starting Page.
- The flow of the rendering and interaction is displayed as red arrows in the following graph.
- GUI tells user to pick UOA in pie chart.

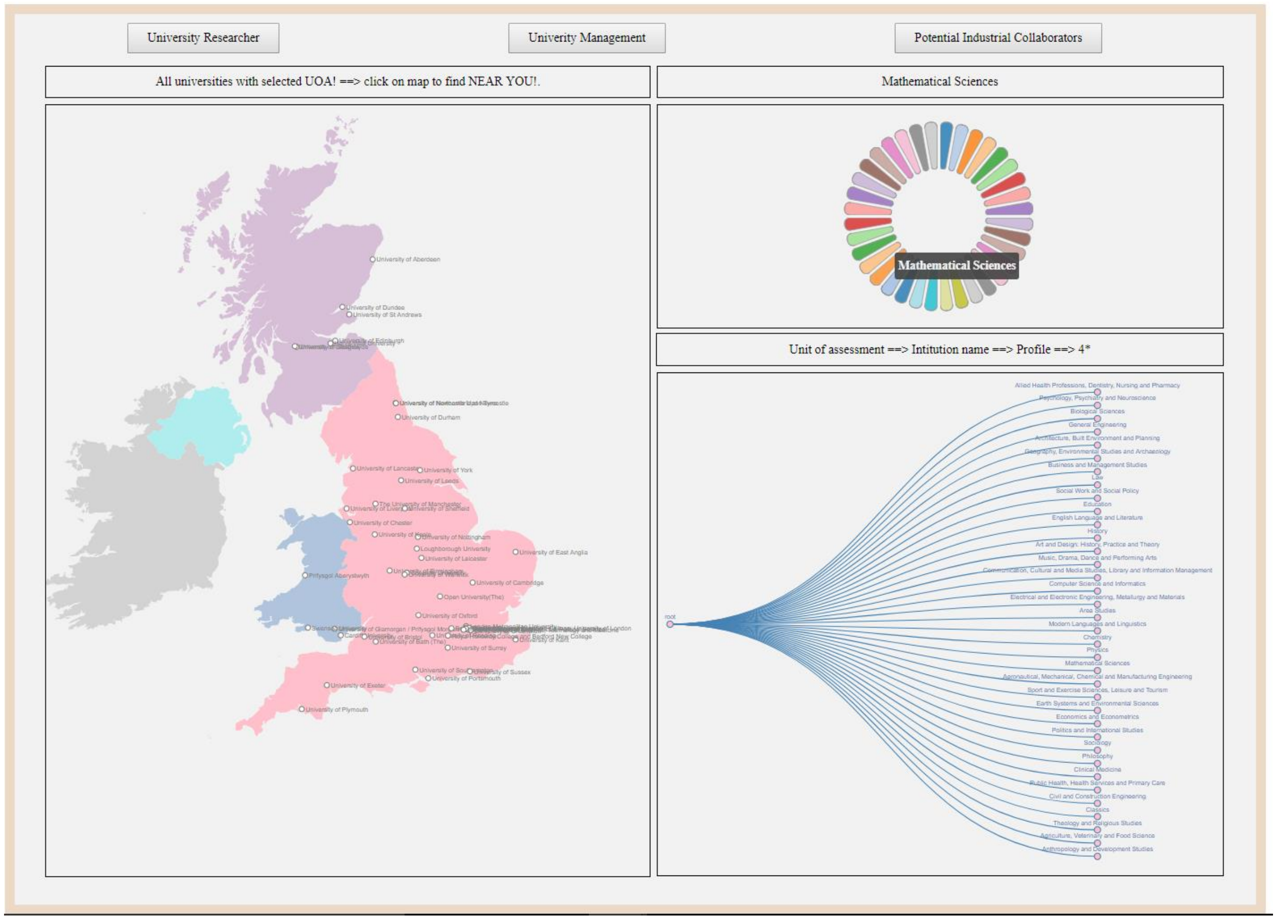

University Researcher
Univerity Management
Potential Industrial Collaborators
University of the Highlands and Islands
The Robert Gordon University
University of Dundee
University of St Andrews
University of Stirling
The University of Newcastle
University of Durham
Teesside University
University of Chester
Aberystwyth University
University of Worcester
University of Exeter
University College Falmouth
root
Allied Health Professions, Dentistry, Nursing and Pharmacy
Psychology, Psychiatry and Neuroscience
Biological Sciences
General Engineering
Architecture, Built Environment and Planning
Geography, Environmental Studies and Archaeology
Business and Management Studies
Law
Social Work and Social Policy
Education
English Language and Literature
History
Art and Design: History, Practice and Theory
Music, Drama, Dance and Performing Arts
Communication, Cultural and Media Studies, Library and Information Management
Computer Science and Informatics
Electrical and Electronic Engineering, Metallurgy and Materials
Area Studies
Modern Languages and Linguistics
Chemistry
Physics
Mathematical Sciences
Aeronautical, Mechanical, Chemical and Manufacturing Engineering
Sport and Exercise Sciences, Leisure and Tourism
Earth Systems and Environmental Sciences
Economics and Econometrics
Politics and International Studies
Sociology
Philosophy
Clinical Medicine
Public Health, Health Services and Primary Care
Civil and Construction Engineering
Classics
Theology and Religious Studies
Agriculture, Veterinary and Food Science
Anthropology and Development Studies

All universities with selected UOA! ==> click on map to find NEAR YOU!.

Pick your unit of assessment!

Unit of assessment ==> Intitution name ==> Profile ==> 4*

- User picks Mathematical Sciences.
- Text shows selected UOA.
- Map displays all universities that have the selected UOA.
- GUI tells user to click on map to pin point his exact location in order to find universities near him.

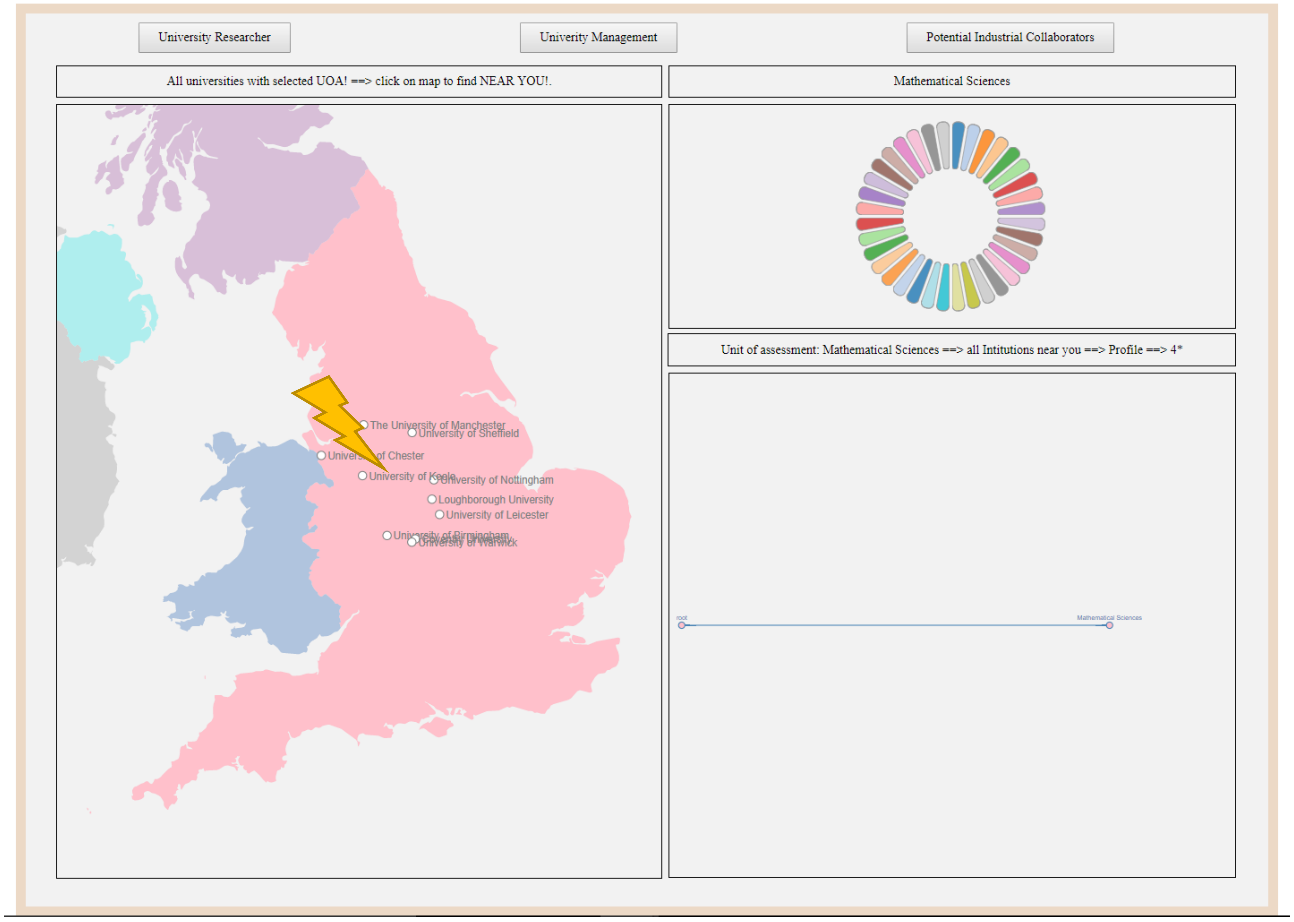

University Researcher
Univerity Management
Potential Industrial Collaborators
Mathematical Sciences
University of Aberdeen
University of Dundee
University of St Andrews
University of Edinburgh
University of Durham
University of York
University of Leeds
The University of Manchester
University of Sheffield
University of Chester
University of Nottingham
Loughborough University
University of Leicester
University of East Anglia
Prifysgol Aberystwyth
University of Cambridge
Open University(The)
University of Oxford
University of Bath (The)
University of Surrey
University of Sussex
University of Portsmouth
University of Exeter
University of Plymouth
root

All universities with selected UOA! ==> click on map to find NEAR YOU!.

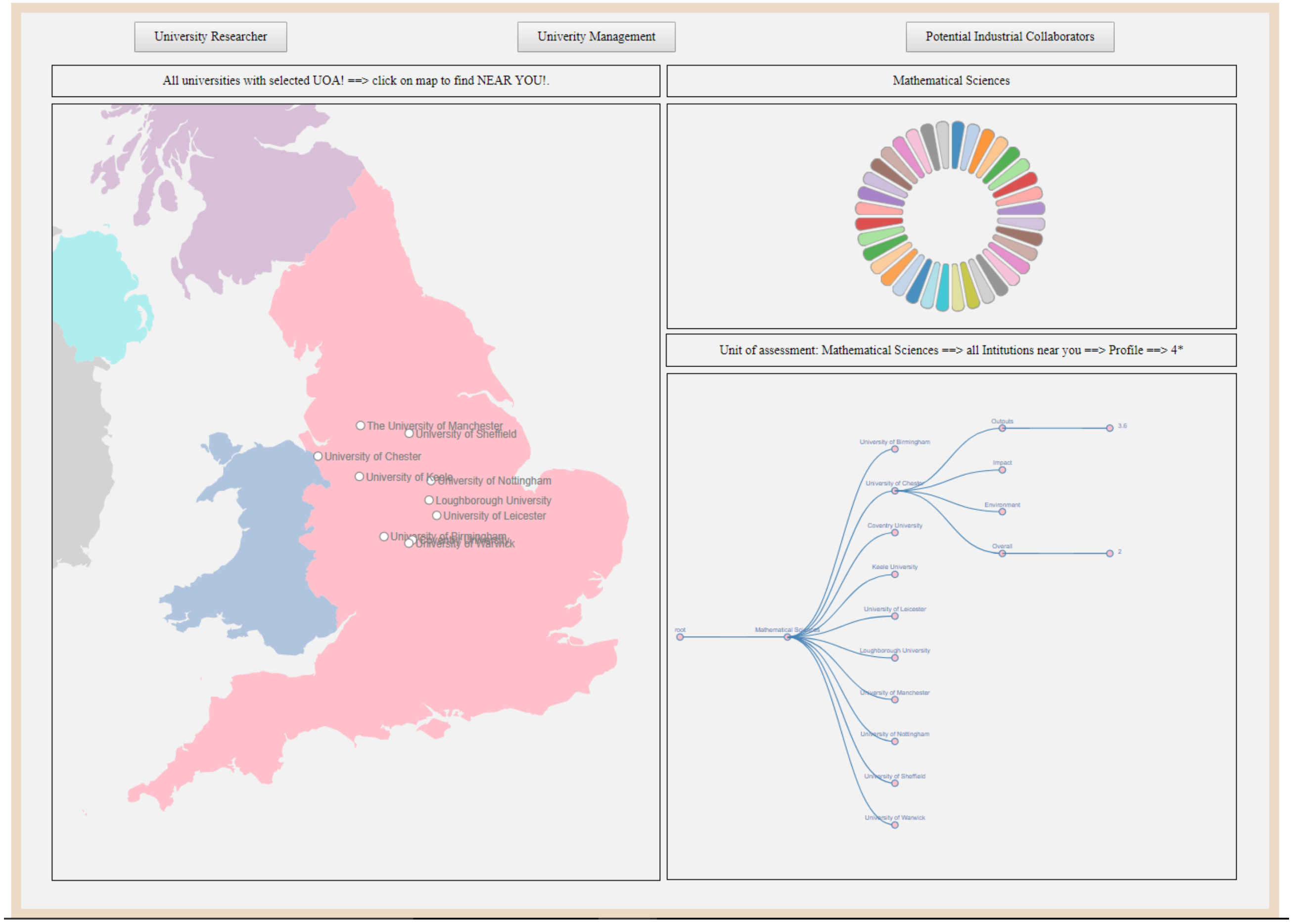
Unit of assessment ==> Intitution name ==> Profile ==> 4*

Allied Health Professions, Dentistry, Nursing and Pharmacy
Psychology, Psychiatry and Neuroscience
Biological Sciences
General Engineering
Architecture, Built Environment and Planning
Geography, Environmental Studies and Archaeology
Business and Management Studies
Law
Social Work and Social Policy
Education
English Language and Literature
History
Art and Design: History, Practice and Theory
Music, Drama, Dance and Performing Arts
Communication, Cultural and Media Studies, Library and Information Management
Computer Science and Informatics
Electrical and Electronic Engineering, Metallurgy and Materials
Area Studies
Modern Languages and Linguistics
Chemistry
Physics
Mathematical Sciences
Aeronautical, Mechanical, Chemical and Manufacturing Engineering
Sport and Exercise Sciences, Leisure and Tourism
Earth Systems and Environmental Sciences
Economics and Econometrics
Politics and International Studies
Sociology
Philosophy
Clinical Medicine
Public Health, Health Services and Primary Care
Civil and Construction Engineering
Classics
Theology and Religious Studies
Agriculture, Veterinary and Food Science
Anthropology and Development Studies

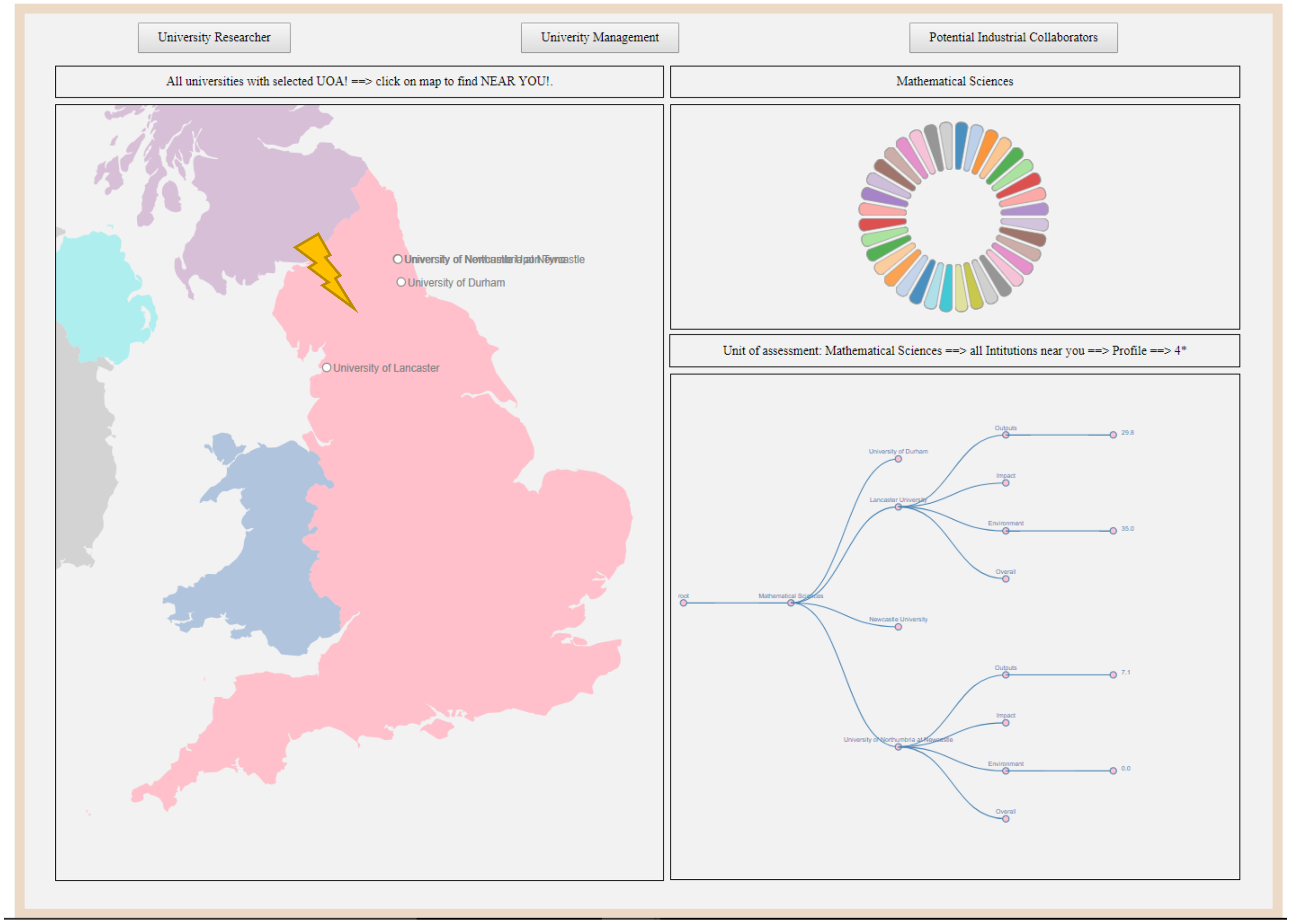
- User clicks on the part of the map resembled by a yellow lightning in the following graph
- Map renders universities within 80 kilometers of the exact point he clicks.



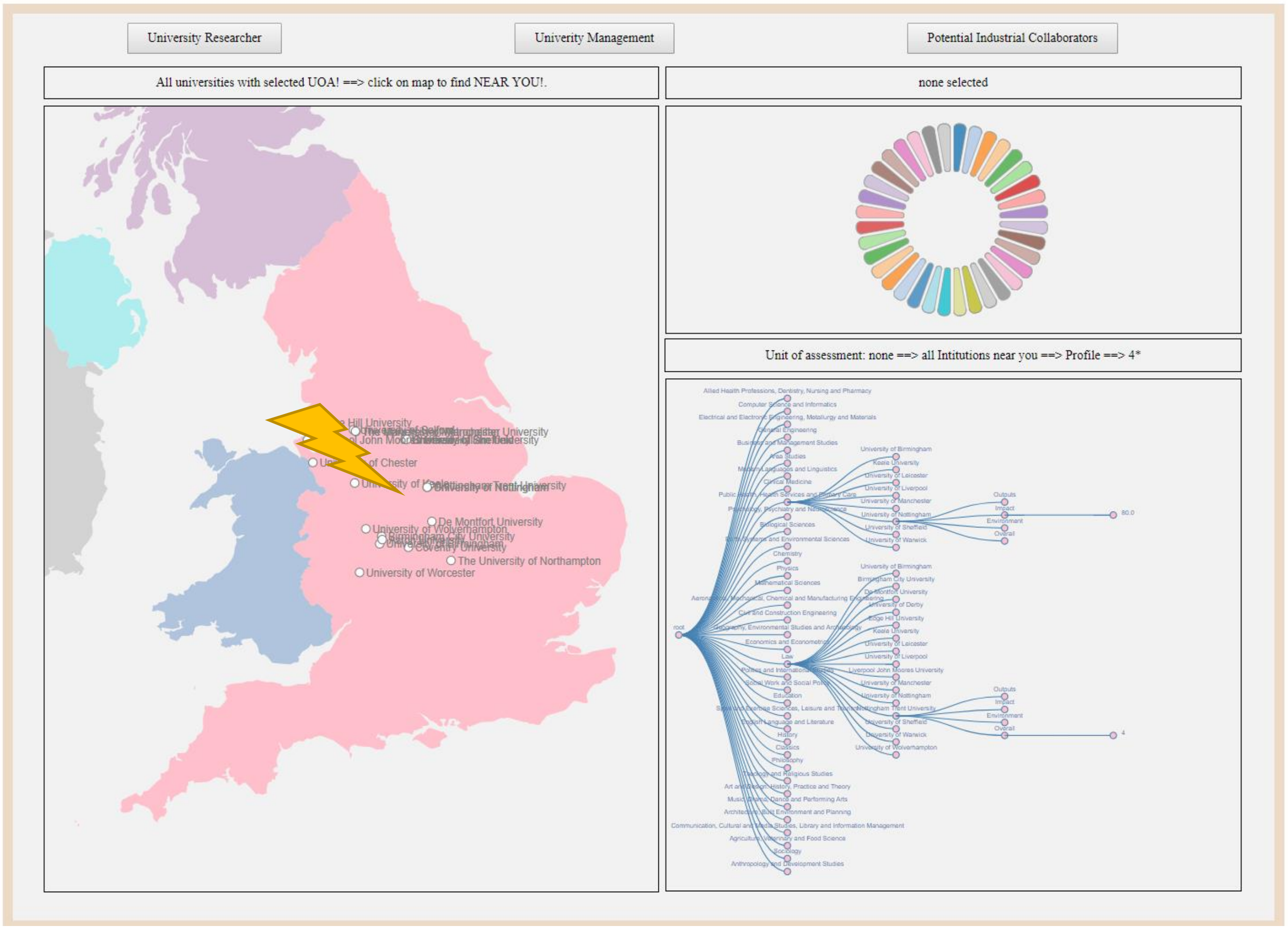
- Meanwhile, the hiearchy is also simulatnously rerendered to show information only about the universities near the user. This helps the user in visualizing and comparing which university is best suited among those near him for a particular UOA.
- The hiearachy shows the 4* rating for all profiles of the universities near the user



- Another example of reclicking on the map.
- Reclick has taken place where the yellow lightning icon strikes.
- Clicking on another point on the map will again rerender and show universities withing 80 kilometer radius from where he clicks. It also rerenders the hierarchy to show the corresponding information.


University Researcher
Univerity Management
Potential Industrial Collaborators
All universities with selected UOA! ==> click on map to find NEAR YOU!.
Mathematical Sciences
University of Northumbria at Newcastle
University of Durham
University of Lancaster
Unit of assessment: Mathematical Sciences ==> all Intitutions near you ==> Profile ==> 4*
root
Mathematical Sciences
University of Durham
Lancaster University
Outputs
29.8
Impact
Environment
35.0
Overall
Newcastle University
University of Northumbria at Newcastle
Outputs
7.1
Impact
Environment
0.0
Overall

- Clicking on the same UOA in the pie chart would remove the selected UOA. This is helpful in the case that the industrial collaborator is not interested in any particular UOA. He can simply find information about all universities around him and for all units of assessment.
- The hierarchy also provides the corresponding information.



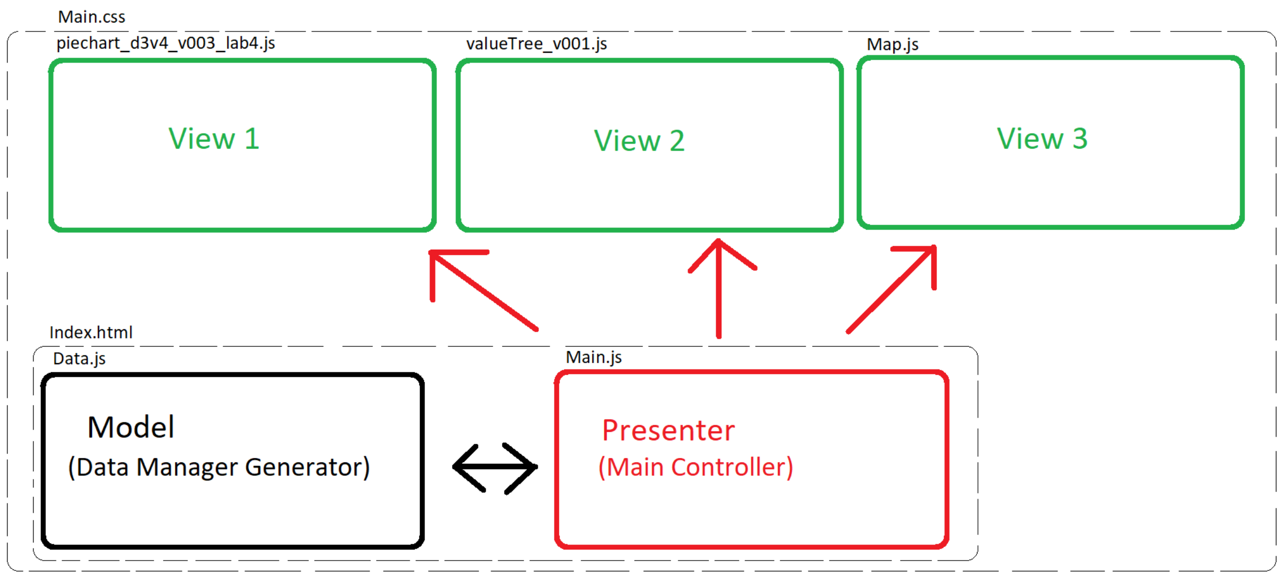
All universities with selected UOA! ==> click on map to find NEAR YOU!.

none selected

Unit of assessment: none ==> all Intitutions near you ==> Profile ==> 4*

Application Design

Design Overview



Reflection on the use the different Design Patterns that you have used

- I have used this design pattern to separate and encapsulate my views. This makes it easier to override functions for several instances of any particular view. This is a good practice to be able to apply different interactions and functions to many instances of one view. Encapsulating the data manager is also good to prevent data loss by accident manipulation or by accident referencing. This is especially useful in programming languages like JavaScript where data is constantly passed as reference, thus possessing a constant potential risk of data loss.

- The Design pattern I implemented does a brilliant job in organizing and separating my data into different classes. Data separation does not only help in organizing the code and increasing readability, it also increases reusability thus enhancing the overall scalability of the application. If I want to increase the number of views later in the future, I can simply reference the new view, and invoke it from my main class render. I won't have to worry about how to adapt the new view, it would be as easy as sticking pieces of LEGO together.

Use of interactions, and transitions

University researcher

Interactions:

1. Pie-chart click function → redraws map
2. Map label click function → redraws hierarchy
3. Hierarchy uses filter from both the pie chart and the map

Rationale:

- Using pie-chart with equal weights can efficiently display all units of assessment in clean and clear fashion. The map is perfect to display institution names for any given UOA, because a user can then use the location as another criterion to match his requirements. Since there are much more universities than units of assessment, the map can be used display more information than the pie chart. The hierarchy graph is ideal to compare between two or more data, simply by expanding all the corresponding nodes.

University Management

Interactions:

1. Hierarchy leaf node click function → redraws pie chart
2. Pie chart click function → redraws map
3. Map uses filters from both the pie chart and the hierarchy

Rationale:

- The Hierarchy is more suited than the pie chart to display institution names. And can give better information just by looking at it, because you can know the relation between institutions and units of assessment. The weighted pie-chart is the most idea of the three views to display all units of assessment for a given institution and their corresponding 4* rating. Final institution names should be eventually pin-pointed on map, because it gives the best visualization.

Potential Industrial Collaborators

Interactions:

1. Pie-chart click function → redraws map
2. Map Country click function → redraws map
3. Map Country click function → redraws hierarchy
4. Hierarchy uses filter from both the pie chart and the Map

Rationale:

- Getting a user's location via map is more efficient, easier, and more fun than allowing the user to enter his location via a text box. It is also more interactive, because I can eventually display universities that are near the user in an efficient manner.

Description of original features

Implement Graphical User Interface Design Principles. (novelty

1. Anticipation
    a. The code has been designed to handle all possible scenarios permitted by the interface.
2. Communication
    a. This is established through User Manual in the form of text fields displayed on the browser)
        i. The interface provides information on where the user stands.
        ii. The interface provides information on what the user can do.
        iii. The interface provides information on where the user was.
2. Consistency
    a. This is established through the use of aesthetics like color, shape, and layout
3. Focus
    a. Interface is focused on the user's task.
    b. Content is designed to fit the purpose of the user. E.g.:
        i. University researcher
        ii. University Management
        iii. Potential industrial collaborators

Using Zoom-able and movable map.

- Zoom-able and movable map has been adopted to comprehend the fact that many universities may be rendered at the same time on the map chart. This would overcrowd the screen with institution names
- Zoom-able and movable map has been adopted to try and represent a real life situation where an industrial collaborator cannot pin point his location from far, and would most certainly need to zoom in to the map in order find his location.

Get coordinates by clicking on the map.

- Using d3 projection invert on mouse-down to get the longitude and latitude by clicking on the map.

Design for three different user types

University researcher

- The University researcher is interested in a particular UOA. He picks it from the pie chart. He wants to find all universities that offer their chosen UOA. Map displays all universities. They would also like to view information about any of those universities in a clean and organized hierarchical fashion. They click on interested university from the map. Hierarchy provides information about the 4* rating for all profiles for their chosen UOA and university.

University Management

- The University Management is interested is knowing how their university is performing against other universities for any chosen UOA. The management first chose their university from a hierarchy and then chose a particular UOA from the pie chart. The map renders only the universities that have a higher overall 4* rating for the chosen UOA. The management gets a clear idea about which universities exactly and how many are performing better than them for any given UOA.

Potential Industrial Collaborators

- Potential collaborators are interned in finding a university with a particular UOA that is located near them. They chose their UOA from the pie chart. Map displays all universities which have that UOA. Collaborators then pin point their location by clicking anywhere on the map. The map displays all universities which have the interested UOA who are located within 80 km radius form their location. Hierarchy displays 4* rating information for all the universities that match their requirements.

Conclusions

We have integrated many different aspects of programming in order to put this project together. Other than using D3 General Update pattern, we have had to use a design pattern on top of an architectural pattern to create a code that is reusable, scalable. We have encapsulated our code on top of which we use the model view controller in order to effectively decouple the major components of our software.

The most interesting aspect of this project that I learned is the fact that Data Visualization is so interrelated with software engineering. It is not simply just how one can portray his data and ideas in a clear and systematic way to some audience. This project has allowed me to practice and implement not only Data Visualization principles and techniques, but also many different principles ranging from the back development to the front end development to testing and much more. It has allowed me to stand in the shoes of literally everyone and look at a scenario from all its possible edges and from all directions.

I am most proud of the novelty that I managed to put through. I am proud at how I managed to display relevant data and information to the user. I am proud at how I managed to fit the purpose of the user in my profiles, and how I managed to apply transitions and interactions efficiently throughout my code.

For my next project, I would explore more D3 layout options, I would also try and adapt different architectural patterns such as the model-view presenter. If given the opportunity, I also would try and make the code more interactive by using angular JavaScript. I can also upgrade this application to become an IoT application by using sensors that retrieve information and stores them in a database.

Source code file list

	% contribution by student	% taken from the course or d3 modules
Index.html	Not applicable	Not applicable
Main.js	95%	5%
Data.js	95%	5%
piechart_d3v4_v003_lab4.js	20%	80%
valueTree_v001.js	5%	95%
Map.js	60%	40%
Main.css	Not applicable	Not applicable

Source code listings

Index.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Issa Haddad Project- H00278537 </title>

    <!-- Stylesheets -->
    <link rel=stylesheet href="styles/main.css">

    <!-- Scripts -->
    <!-- libs -->
    <!--script src="libs/d3/d3.v4.min.js" type="text/javascript" charset="utf-8"></script--
>
    <script src="https://d3js.org/d3.v4.min.js"></script>
    <script src="http://d3js.org/topojson.v1.min.js"></script>
    <!-- controller -->
    <script src="scripts/main.js" type="text/javascript" charset="utf-8"></script>
    <!-- model -->
    <script src="scripts/model/data.js" type="text/javascript" charset="utf-8"></script>
    <!-- renderers -->
    <!--script src="scripts/view/barchart.js" type="text/javascript" charset="utf-
8"></script-->
    <script src="scripts/view/map.js" type="text/javascript" charset="utf-8"></script>
    <script type="text/javascript" src="scripts/view/piechart_d3v4_v003_lab4.js"></script>
    <script type="text/javascript" src="scripts/view/valueTree_v001.js"></script>
</head>
<body>
    <div id="dashboard">
    <button type="button" onclick="UniRes()">University Researcher</button>
    <button type="button" onclick="UniMan()">Univerity Management</button>
    <button type="button" onclick="PInCol()">Potential Industrial Collaborators</button>
        <div id="manual"></div>
        <div id="manual2"></div>
        <div id="wrapper">
            <svg id="map1svg">

                <rect id="map1rect"></rect>
                <g id="map1g"></g>
            </svg>
            <!--h3>My Pie</h3-->
            <svg id="pie1svg">
                <g id="pie1g"></g>
            </svg>
            <div id="pietool"></div>
            <div id="manual3"></div>
            <svg id="hi1svg">
                <rect id="hi1rect"></rect>
                <g id="hi1g"></g>
            </svg>
        </div>
    </div>

    <script>
        loadCSV1();
    </script>

</body>
</html>
```

Main.js

```
/*-------------------------------------------------------------------

   Module: MAIN CONTROLLER
   Author: Issa Haddad

   What it does:
    - Instantiates data manager, and makes it load data
    - Generates pie charts in dashboard
    - Generates maps in dashboard
    - Generates hierarchy charts in dashboard
    - Feeds data from data manager into charts
    - Provides click interactions across all charts

   Percentage of code written by Issa Haddad: 95%
   Percentage of code taken from course example: 5%

   Dependencies: (none)

   History: 17/11/2017

   References: (none)

-------------------------------------------------------------------- */


var dataManager = {};

//if no profile is clicked, the default one which is displayed is: (University researcher)
var button = 1;

// load University researcher if button is pressed
function UniRes()
{
    button = 1;
    loadCSV1();
}

// load University mmanagement if button is pressed
function UniMan()
{
    button = 2;
    loadCSV1();
}

// load industrial collaborator if button is pressed
function PInCol()
{
    button = 3;
    loadCSV1();
}


// load CSV: REF2014 and Learning providers - when ready (to invoke synchronous events)
function loadCSV1(){

    dataManager = dataManagerConstructor();

    dataManager.read1("resources/REF2014_Results.csv",
        function(error){ // to call in case of error
```

```
            console.log("Error");
            console.log(error);
        },
        function(){ // to call if everything ok
            loadCSV2();
        });
}

// load Json: UK - when ready (to invoke synchronous events)
function loadCSV2(){

    dataManager.loadDatasetJSON("resources/uk.json",
        function(error){ // to call in case of error
            console.log("Error");
            console.log(error);
        },
        function(){ // to call if everything ok
            makePage();
        });
}


function makePage(){

    // APPEND VISUALISATION
    var map1, map2, map3;
    var pie1, pie2, pie3;
    var tr1, tr2, tr3;

    // render views for university researcher
    if(button==1)
    {
        // set up map with relevant interactions
        map1 = map("#map1svg", "#map1rect", "#map1g");
        // pass learning providers to map
        map1.setlp(dataManager.getDataLP());
        // setupcallback overrides a function in .on(mousedown) on map, it should only be invoked for map3
        map1.setUpCallback(function() {
        })
        map1.loadAndRender(dataManager.getDataUK())
                // setmouseclickckcallback overrides a function in .on(click) on map labels; store UniName filter and update tree
                map1.setMouseclickCallback(function(d, i){
                dataManager.setUniNameFilter(d["VIEW_NAME"]);
                // redraw tree to show the 4* rating for all profiles of the selected UOA and the selected institution
                redrawtree();
                });
                //print user maual text to screen
                document.getElementById('manual2').innerHTML = '<p>Pick your unit of assessment!';
                document.getElementById('manual').innerHTML = '<p>All universities with selected UOA! ==> click on institution label to view Hierarchy.';
                document.getElementById('manual3').innerHTML = '<p>Unit of assessment ==> Intitution name ==> Profile ==> 4*';
                // set up pie with relevant interactions
                pie1 = piechart("#pie1svg", "#pie1g", "#pietool");
                // tooltip shows d.data.key (which is UoA)
                pie1.setTooltipText(function(d, i){
                    return "<b>"+d.data.key+"</b>";
                })
                // render all UOA's in piechart
                pie1.loadAndRenderDataset(dataManager.getHierarchyUOA().values);
                // update map when pie is clicked to show all universities that have the particular UOA that is clicked
                pie1.setMouseclickCallback(function(d, i){
                dataManager.setUOAFilter(d.data.key);
                // redraw map to show all univerities that have the particular filtered UOA
                redrawmap();
                })
                // set up tree with relevant interactions
                tr1 = tree("#hi1svg", "#hi1g");
                // leaf nodes should show the 4* rating for each university and its corresponding UOA
                tr1.leafLabelFn(function(d) {return d.data["4*"]});
                tr1.appendToClick(function(d){
                            //Your additional code here
                            //(e.g. in case you want interaction with other layouts)
                            if (d. height == 0) console.log("leaf node clicked, d=",d.data["4*"])
                        });
                // use hierarchy with keys: UOA - Institution name - Profile
                tr1.loadAndRenderDataset(dataManager.getHierarchy_UOA_Inst_Pro());
            }

            // render views for university management
            if(button ==2)
            {

                // set up tree with relevant interactions
                tr2 = tree("#hi1svg", "#hi1g");
                // leaf nodes should show institution name
                tr2.leafLabelFn(function(d) {return d.data["Institution name"]});
                // clicking on leaf node would show all UOA for that particular institution
                tr2.appendToClick(function(d){
                            //Your additional code here
                            //(e.g. in case you want interaction with other layouts)
                            if (d. height == 0)
                                {
                                    dataManager.setUniName2Filter(d.data["Institution name"]);

                                    // redraw pie to show all UOA for a particular filtered Institution with theri overall 4* proportionally
                                    redrawpie2();
                                }
                        });
                // use hierarchy with keys: UOA - Institution name
                tr2.loadAndRenderDataset(dataManager.getHierarchy4_UOA_Inst());

                //print user maual text to screen
                document.getElementById('manual2').innerHTML = '<p>Heriot Watt UOAs and their overall 4* rating ==> Select UOA to view Institutions with higher 4* rating!';
                document.getElementById('manual').innerHTML = '<p>All other Institutions that have higher overall 4* rating for the selected UOA';
                document.getElementById('manual3').innerHTML = '<p>Unit of assessment ==> Intitution name ==> Instituion name; Select Intitution!';

                // set up pie with relevant interactions
                pie2 = piechart("#pie1svg", "#pie1g", "#pietool");
                // override pie datafield to 4* to be used as arcgenerator value and override other values as well
                pie2.overrideDataFieldFunction(function(d){return d["4*"]});
                pie2.overrideDataKeyFunction(function (d){return d.data["Unit of assessment number"]});
                pie2.overrideDataTextFunction(function(d) { return d.data["Unit of assessment name"]});
                // show UOA and its overall 4* rating for a particular univeristy
                pie2.setTooltipText(function(d, i){
                    return "<b>"+d.data["Unit of assessment name"]+ ": its 4* = " + d.data["4*"] +"</b>";
                })
                // intially render all UOA with profile=overall for Heriot Watt university
                pie2.loadAndRenderDataset2(dataManager.getPieHeriotWatt_UOA_Overall());
                pie2.setMouseclickCallback(function(d, i){
                // use key (i.e. year) of element clicked on as filter
                dataManager.setUOA2Filter(d.data["Unit of assessment name"]);
                dataManager.setStar4Filter(d.data["4*"]);
                // redraw map to show all universities that have the filter UOA and which have a higher overall 4* rating
                redrawmap2();
                })

                // set up map with relevant interactions
                map2 = map("#map1svg", "#map1rect", "#map1g");
                map2.setlp(dataManager.getDataLP());
                // setupcallback ovrides a function in .on(mousedown) on map, it should only be invoked for map3
                map2.setUpCallback(function() {
                })
                map2.loadAndRender(dataManager.getDataUK())
            }

            // render views for industrial collaborator
            if (button ==3)
            {

                // set up map with relevant interactions
                map3 = map("#map1svg", "#map1rect", "#map1g");
                map3.setlp(dataManager.getDataLP());
                map3.loadAndRender(dataManager.getDataUK())
                // setupcallback ovrides a function in .on(mousedown) to update map3 and tr3 to show information about
                // .on(mousedown) --> records latitude and longitude when map is clicked, this is used to identify universities around the point which is clicked up to a certain radius
                map3.setUpCallback(function() {
                    // redraw map to show all near insitutions that have the filtered UOA
                    redrawmap3new();
                    // redraw tree to show ratings for all profiles for the instutions that appear on the map
                    redrawtree3();
                })
                map3.setMouseclickCallback2(function(d, i){
                });
                //invoke function that has mousedown behavior
                map3.loadInteraction2();
                //print user maual text to screen
                document.getElementById('manual2').innerHTML = '<p>Pick your unit of assessment!';
                document.getElementById('manual').innerHTML = '<p>All universities with selected UOA! ==> click on map to find NEAR YOU!.';
                document.getElementById('manual3').innerHTML = '<p>Unit of assessment ==> Intitution name ==> Profile ==> 4*';
                // set up pie with relevant interactions
```

```
                pie3 = piechart("#pie1svg", "#pie1g", "#pietool");
                pie3.setTooltipText(function(d, i){
                    return "<b>"+d.data.key+"</b>";
                })
                // render all UOA's in piechart
                pie3.loadAndRenderDataset(dataManager.getHierarchyUOA().values);
                pie3.setMouseclickCallback(function(d, i){
                dataManager.setUOA3Filter(d.data.key);
                // redraw map to show all universities that have the selected UOA
                redrawmap3();
                })

                // set up tree with relevant interactions
                tr3 = tree("#hi1svg", "#hi1g");
                // leaf node to show the 4* rating on each profile
                tr3.leafLabelFn(function(d) {return d.data["4*"]});
                tr3.appendToClick(function(d){
                            //Your additional code here
                            //(e.g. in case you want interaction with other layouts)
                            if (d. height == 0)
                                {
                                    console.log("leaf node clicked, d=",d.data["4*"])
                                }
                        });
                // use hierarchy with keys: UOA - Institution name - Profile
                tr3.loadAndRenderDataset(dataManager.getHierarchy2_UOA_Inst_Pro());
            }


            function redrawtree(){
                // use hierarchy with keys: UOA - Institution name - Profile
                tr1.loadAndRenderDataset(dataManager.getHierarchy_UOA_Inst_Pro());
            }
            function redrawmap(){
                map1.setlp(dataManager.getDataLP_filtered_UOA());
                map1.loadAndRender(dataManager.getDataUK());
            }

            function redrawmap2(){
                map2.setlp(dataManager.getDataLP_filtered_UOA_higher_4star());
                map2.loadAndRender(dataManager.getDataUK());
            }

            function redrawpie2()
            {
                pie2.setTooltipText(function(d, i){
                    return "<b>"+d.data["Unit of assessment name"]+ ": its 4* = " + d.data["4*
  "] +"</b>";
                })
                pie2.loadAndRenderDataset2(dataManager.getPieDatanew());
            }

            function redrawmap3(){
                map3.setlp(dataManager.getDataLP3_filtered_UOA());
                map3.loadAndRender(dataManager.getDataUK());
            }

            function redrawmap3new(){
                map3.setlp(dataManager.getDataLP_filtered_UOA_specific_Long_Lat());
                map3.loadAndRender(dataManager.getDataUK());
            }


            function redrawtree3(){
                tr3.loadAndRenderDataset(dataManager.getHierarchy_filtered_Long_Lat());
            }
        }
```

Data.js

```
/*---------------------------------------------------------------------

   Module: DATA MANAGER GENERATOR
   Author: Issa Haddad

   What it does:
    Generates a data manager (model), with API to load
    and filter given dataset.

   Percentage of code written by Issa Haddad: 95%
   Percentage of code taken from course example: 5%

   Dependencies
    D3.js v4

   History: 17/11/2017

   References: (none)

---------------------------------------------------------------------- */


function dataManagerConstructor(){

    // global variables
    var dmObj = {}; // main object
    var ref14data;
    var ref14datacopy;
    var refUniversitiesByUKPRN;
    var learningProviders;
    var learningProviderscopy;
    var heriotWattCS;
    var HeriotWatt_UOA_Overall;



    // returns REF14 filtered on Heriot Watt - OVerall
    dmObj.getPieHeriotWatt_UOA_Overall = function()
    {
        return HeriotWatt_UOA_Overall;
    }

    //if UniName2Filter is not set --> returns REF14 filtered on Heriot Watt - OVerall
    //if UniName2Filter is set --> rteurn REF14 filtered on UniName2Filter - OVerall
    dmObj.getPieDatanew = function()
    {
        var HeriotWatt_UOA_Overallnew = ref14data.filter(function(row)
        {
            if (UniName2Filter == null)
            {
                return (row["Institution code (UKPRN)"] == 10007764) && (row["Profile"] == "Overall")
            }
            else
            {
                return (row["Institution name"] == UniName2Filter) && (row["Profile"] == "Overall")
            }
        })
        return HeriotWatt_UOA_Overallnew;
    }

    // ---- PUBLIC API

    // function loading CSV data ("resources/REF2014_Results.csv" & "resources/learning-providers-plus.csv")
    // parameters:
    //      - url, url of data in string format
    //      - callbackError, callback in case of error
    //      - callbackSuccess, callback if data loads normally
    // returns:
    //      - Nothing
    dmObj.read1 = function (url, callbackError, callbackSuccess){
        d3.csv("resources/REF2014_Results.csv", function(error, csvData) {
        if(error){
                callbackError(error);
            } else {
                ref14data = csvData;
                ref14datacopy = csvData;
                console.log("ref14data= ", ref14data);
                d3.csv("resources/learning-providers-plus.csv", function(error, csvData) {
                    if(error){
                    callbackError(error);
                    } else {
                        learningProviders = csvData;
                        learningProviderscopy = csvData;
                        console.log("learningProviders = ", learningProviders);

                        //Check to see if we have any Universities in REF2014 that are not in the learning providers
                        findREFunisWithNoEntryInLearningProvidersPlus(ref14data, learningProviders);

                        //Add and duplicate learningProviders data into REF2014
                        //e.g. add the learningProviders HWU entry, to every HWU entry in REF 2014
                        combineCSVdata(ref14data, learningProviders);
                        console.log("\n\nref14data (after combo with learning provider data) = ", ref14data);

                        //Filter out HWU combined entries and show in html
                        displayHWCS(ref14data);

                        //Filter out HWU combined entries with overall
                        labFourOne(ref14data);

                               //Convert into Nested data, and show HWU object
                               displayOneUniversityAsNestedList(ref14data);
                               callbackSuccess();
                           }
                           })
                   }
               })
          }

          // function loading json data from url
          // parameters:
          //      - url, url of data in string format
          //      - callbackError, callback in case of error
          //      - callbackSuccess, callback if data loads normally
```

```javascript
        // returns:
        //      - Nothing
        dmObj.loadDatasetJSON = function (url, callbackError, callbackSuccess){
            d3.json(url, function(error, data){
                if(error){
                    callbackError(error);
                } else {
                    dataset = data;
                    console.log("Data loaded in dataManager");
                    console.log(dataset);
                    callbackSuccess();
                }
            })
        }


        // returns uk json file
        dmObj.getDataUK = function(){
            return dataset;
        }

        // reurns learning provider array
        dmObj.getDataLP = function(){
            return learningProviders;
        }

        // function to return array with all institutions that have a specific UOA
        // creates a higherarchy on REF14 data with keys: UOA - Profile
        // filters hierarchy to on row[key] == UOAFilter
        // if UOAFilter is null --> return all of learning providers
        // if UOAFilter is set --> return (lpnew2[0].values[3].values) containing filtered universities
        dmObj.getDataLP_filtered_UOA = function(){

            var hierarchyJSONnew = createJSONhierarchyNOTFIL(ref14data, "root", ["Unit of assessment name", "Profile"])
            lpnew2 = hierarchyJSONnew.values.filter(function(row){
                return (String(row["key"]) == UOAFilter || UOAFilter == null )
            })


            if(UOAFilter == null)
            {
                return learningProviders;
            }
            else{
                return lpnew2[0].values[3].values;
            }
        }

        // function to return array with all institutions that have a specific UOA
        // creates a higherarchy on REF14 data with keys: UOA - Profile
        // filters hierarchy to on row[key] == UOA3Filter
        // if UOA3Filter is null --> return all of learning providers
        // if UOA3Filter is set --> return (lpnew2[0].values[3].values) containing filtered universities
        dmObj.getDataLP3_filtered_UOA = function(){

            var hierarchyJSONnew = createJSONhierarchyNOTFIL(ref14data, "root", ["Unit of assessment name", "Profile"])
            lpnew2 = hierarchyJSONnew.values.filter(function(row){
                return (String(row["key"]) == UOA3Filter || UOA3Filter == null )
            })


            if(UOA3Filter == null)
            {
                return learningProviders;
            }
            else{
                return lpnew2[0].values[3].values;
            }
        }


        // function to return array with all institutions that fall withing a certain range of LatFilter & LongGilter
        // function filters REF14 data: gets all records with (lat lower than max radius range & lat greater than min radius range & long lower than max radius range & long greater than min radius range)
        // creates a higherarchy on filtered data with keys: UOA - Profile
        // filters hierarchy to on row[key] == UOA3Filter
        // if LatFilter is null --> return all of learning providers
        // if LAtFilter is set , but no elements exists --> return empty object
        // if LAtFilter is set --> return (lpnew2[0].values[3].values) containing filtered universities
        dmObj.getDataLP_filtered_UOA_specific_Long_Lat= function(){
            var kmInLongitudeDegree = 111.320 * Math.cos( LatFilter / 180.0 * Math.PI);
            // specify radius in KM = 80
            var deltaLat = 80 / 111.1;
            var deltaLong = 80 / kmInLongitudeDegree;
            console.log("delta lat: ", deltaLat);
            console.log("delta long: ", deltaLong);

            var temp11 = ref14data.filter(function(row){
                return ((Number(row["LONGITUDE"])) > (Number(LongFilter)-deltaLong) && (Number(row["LONGITUDE"])) < (Number(LongFilter)+deltaLong) && (Number(row["LATITUDE"])) < (Number(LatFilter)+deltaLat) && (Number(row["LATITUDE"])) > (Number(LatFilter)-deltaLat))
            })

            var hierarchyJSONnew = createJSONhierarchyNOTFIL(temp11, "root", ["Unit of assessment name", "Profile"])
            lpnew2 = hierarchyJSONnew.values.filter(function(row){
                return (String(row["key"]) == UOA3Filter || UOA3Filter == null )
            })


            if(LatFilter == null)
            {
                return learningProviders;
            }
            else{
                if(lpnew2[0] == null)
                {
                    return {} ;
                }
                else
                {
                return lpnew2[0].values[3].values;
                }
            }
        }

```

```
        // function to return array with all institutions that have a particular UOA and whos 4* overall rating is greater than Star4Filter
        // function filters REF14 data: gets all records with profile = overall and 4* rating higher than Star4Filter
        // creates a higherarchy on filtered data with keys: UOA - Profile
        // filters hierarchy to on row[key] == UOA2Filter
        // if UOA2Filter is null --> return all of learning providers
        // if UOA2Filter is set , but no elements exists --> return empty object
        // if UOA2Filter is set --> return (lpnew2[0].values[0].values) containing filtered universities
        dmObj.getDataLP_filtered_UOA_higher_4star = function(){

            var higher4 = ref14data.filter(function(row){
                return (Number(row["4*"]) > Star4Filter || Star4Filter == null) && (row["Profile"] == "Overall")
            })

            var hierarchyJSONnew = createJSONhierarchyNOTFIL(higher4, "root", ["Unit of assessment name", "Profile"])
            lpnew2 = hierarchyJSONnew.values.filter(function(row){
                return (String(row["key"]) == UOA2Filter || UOA2Filter == null )
            })

            if(UOA2Filter == null)
            {
                return learningProviders;
            }
            else{
                if(lpnew2[0] == null)
                {
                    return {} ;
                }
                else
                {
                return lpnew2[0].values[0].values;
                }
            }
        }

        // returns a higherarchy on data and keys passed from parameter
        function createJSONhierarchyNOTFIL(flatDataset, rootKey, keys){
            var hierarchy = d3.nest();
            keys.forEach(applyKey);

            function applyKey(key, i){
                hierarchy = hierarchy
                    .key(function (d) {
                        return d[key];
                    });
            }

            //Uncomment to see effect of rollup method

            hierarchy = hierarchy.entries(flatDataset);
            //Return single top node called the value of rootKey
            return {"key":rootKey, "values": hierarchy}
        }

        // returns a higherarchy on filtered data (UniNameFilter & UOAFilter) with multiple keys passed from parameter
        function createJSONhierarchy(flatDataset, rootKey, keys){
            var hierarchy = d3.nest();
```

```
                keys.forEach(applyKey);

                function applyKey(key, i){
                    hierarchy = hierarchy
                        .key(function (d) {
                            return d[key];
                        });
                }

                //Uncomment to see effect of rollup method

                hierarchy = hierarchy.entries(filteredData(flatDataset));
                //Return single top node called the value of rootKey
                return {"key":rootKey, "values": hierarchy}
            }

            // returns a higherarchy on data and keys passed from parameter
            function createJSONhierarchy2(flatDataset, rootKey, keys){
                var hierarchy = d3.nest();
                keys.forEach(applyKey);

                function applyKey(key, i){
                    hierarchy = hierarchy
                        .key(function (d) {
                            return d[key];
                        });
                }

                //Uncomment to see effect of rollup method

                hierarchy = hierarchy.entries(flatDataset);
                //Return single top node called the value of rootKey
                return {"key":rootKey, "values": hierarchy}
            }

            // returns a higherarchy on filtered data (UOA3Filter) with keys passed from parameter
            function createJSONhierarchy3(flatDataset, rootKey, keys){
                var hierarchy = d3.nest();
                keys.forEach(applyKey);

                function applyKey(key, i){
                    hierarchy = hierarchy
                        .key(function (d) {
                            return d[key];
                        });
                }

                //Uncomment to see effect of rollup method

                hierarchy = hierarchy.entries(filteredData3(flatDataset));
                //Return single top node called the value of rootKey
                return {"key":rootKey, "values": hierarchy}
            }

            // returns a higherarchy on filtered data (UniNameFilter & UOAFilter) with 1 key passed from parameter
            function createJSONhierarchyOne(flatDataset, rootKey, key){
                var hierarchy = d3.nest();
                keys.key(function(d) { return d[key]; })

                //Uncomment to see effect of rollup method

                hierarchy = hierarchy.entries(filteredData(flatDataset));
                //Return single top node called the value of rootKey
                return {"key":rootKey, "values": hierarchy}
            }


            // returns a higherarchy on filtered data with keys: UOA - Institution name
            dmObj.getHierarchy = function(){
                var hierarchyJSON = createJSONhierarchy(ref14data, "root", ["Profile", "Main panel", "Unit of assessment name"])
                return hierarchyJSON;
            }

            // return heirarchy with key = UOA
            // append an equal number with all keys, to render pie archs of equal size
            dmObj.getHierarchyUOA = function(){
                var hierarchyJSON2 = createJSONhierarchy(ref14data, "root", ["Unit of assessment name"])
                hierarchyJSON2.values.forEach(function(ref14entry){
                            ref14entry.num = "2";
                    })
                return hierarchyJSON2;
            }

            // returns a higherarchy on REF14 data with keys: UOA - Institution name - Profile

            dmObj.getHierarchy_UOA_Inst_Pro = function(){
                // call JSONhierarchy function that correclty identiefies related filters
                var hierarchyJSON2 = createJSONhierarchy(ref14data, "root", ["Unit of assessment name","Institution name", "Profile"])
                //var hierarchyJSON2 = createJSONhierarchyOne(ref14data, "root", "Institution name")
                return hierarchyJSON2;
            }

            // function filters REF14 data: gets all records with profile = overall
            // returns a higherarchy on filtered data with keys: UOA - Institution name
            dmObj.getHierarchy4_UOA_Inst = function(){

                var temp = ref14data.filter(function(row){
                    return (row["Profile"] == "Overall")
                })

                var hierarchyJSON2 = createJSONhierarchy2(temp, "root", ["Unit of assessment name","Institution name"])
                //var hierarchyJSON2 = createJSONhierarchyOne(ref14data, "root", "Institution name")
                return hierarchyJSON2;
            }

            // returns a higherarchy on REF14 data with keys: UOA - Institution name - Profile

            dmObj.getHierarchy2_UOA_Inst_Pro = function(){

                // call JSONhierarchy function that correclty identiefies related filters
                var hierarchyJSON2 = createJSONhierarchy3(ref14data, "root", ["Unit of assessment name","Institution name", "Profile"])
                //var hierarchyJSON2 = createJSONhierarchyOne(ref14data, "root", "Institution name")
                return hierarchyJSON2;
            }

            // function filters REF14 data: gets all records within a radius of LatFilter & LongFilter
            // returns a higherarchy on filtered data with keys: UOA - Institution name - Profile
            dmObj.getHierarchy_filtered_Long_Lat = function(){
                var kmInLongitudeDegree = 111.320 * Math.cos( LatFilter / 180.0 * Math.PI);
                // specify radius in KM = 80
                var deltaLat = 80 / 111.1;
                var deltaLong = 80 / kmInLongitudeDegree;
                console.log("delta lat: ", deltaLat);
                console.log("delta long: ", deltaLong);

                var temp11 = ref14data.filter(function(row){
                    return ((Number(row["LONGITUDE"])) > (Number(LongFilter)-deltaLong) && (Number(row["LONGITUDE"])) < (Number(LongFilter)+deltaLong) && (Number(row["LATITUDE"])) < (Number(LatFilter)+deltaLat) && (Number(row["LATITUDE"])) > (Number(LatFilter)-deltaLat))
                })

                var hierarchyJSONnew = createJSONhierarchy3(temp11, "root", ["Unit of assessment name","Institution name", "Profile"])

                return hierarchyJSONnew;
            }


            // print filter for debugging
            dmObj.printUniName = function(){
                console.log("UniNameFilter = ", UniNameFilter);
            }

            // print filter for debugging
            dmObj.printUniName2 = function(){
                console.log("UniName2Filter = ", UniName2Filter);
            }

            // print filter for debugging
            dmObj.printUOA3Filter = function(){
                console.log("UOA3Filter = ", UOA3Filter);
            }

            // print filter for debugging
            dmObj.printlongFilter = function(){
                console.log("LongFilter = ", LongFilter);
            }

            // print filter for debugging
            dmObj.printlatFilter = function(){
                console.log("LatFilter = ", LatFilter);
            }

            // set Longitude filter
            dmObj.setlongFilter = function(u){
                LongFilter = u;
                if (UOA3Filter ==null)
                {
                    document.getElementById('manual3').innerHTML = '<p>Unit of assessment: none ==> all Intitutions near you ==> Profile ==> 4*';
                }
                else
                {
```

```
                document.getElementById('manual3').innerHTML = '<p>Unit of assessment: '+ UOA3Filter+ ' ==> all Intitutions near you ==> Profile ==> 4*';
            }
        }

        // set latitude filter
        dmObj.setlatFilter = function(u){
            LatFilter = u;
        }

        // function setting UOA filter and updating user manual text on screen
        // parameters:
        //      - u, filter to set, if equal to current filter, set to null
        // returns:
        //      - nothing
        dmObj.setUOAFilter = function(u){
            if(UOAFilter == u){
                UOAFilter = null;
                document.getElementById('manual2').innerHTML = '<p>' + "none selected";
            } else {
                UOAFilter = u;
                document.getElementById('manual2').innerHTML = '<p>' + u;
            }
        }

        // function setting UOA3 filter and updating user manual text on screen
        // parameters:
        //      - u, filter to set, if equal to current filter, set to null
        // returns:
        //      - nothing
        dmObj.setUOA3Filter = function(u){
            if(UOA3Filter == u){
                UOA3Filter = null;
                document.getElementById('manual2').innerHTML = '<p>' + "none selected";
            } else {
                UOA3Filter = u;
                document.getElementById('manual2').innerHTML = '<p>' + u;
            }
        }

        // function setting UniName filter and updating user manual text on screen
        // parameters:
        //      - u, filter to set, if equal to current filter, set to null
        // returns:
        //      - nothing
        dmObj.setUniNameFilter = function(u){
            if(UniNameFilter == u){
                UniNameFilter = null;
                document.getElementById('manual').innerHTML = '<p>' + "none selected";
            } else {
                UniNameFilter = u;
                document.getElementById('manual').innerHTML = '<p>' + u;
            }
        }

        // function setting UniName2 filter and updating user manual text on screen
        // set old and new values of UniName2 to Uniarray[0] & Uniarray[1] respectively
        // this is imporatant for UAO2 filter to not be reset to null if the UniName2 changes from 1 value to other
        // parameters:
        //      - u, filter to set, if equal to current filter, set to null
        // returns:
        //      - nothing

        dmObj.setUniName2Filter = function(u){
            if(UniName2Filter == u){
                    Uniarray[0] = UniName2Filter;
                UniName2Filter = null;
                document.getElementById('manual3').innerHTML = '<p>Unit of assessment ==> Intitution name ==> Institiuon name; Select Intitution!';
                document.getElementById('manual2').innerHTML = '<p>Heriot Watt UOAs and their overall 4* rating ==> Select UOA to view Institutions with higher 4* rating!';
                Uniarray[1] = u;
            } else {
                if (UniName2Filter == null)
                {
                Uniarray[0] = "null";
                }
                else
                {
                    Uniarray[0] = UniName2Filter;
                }
                UniName2Filter = u;
                document.getElementById('manual3').innerHTML = '<p>Unit of assessment ==> '+ u +' ==> ' + u;
                document.getElementById('manual2').innerHTML = '<p>'+ u +' UOAs and their overall 4* rating = ==> Select UOA to view Institutions with higher 4* rating!';
                Uniarray[1] = u;
            }
        }


        // function setting UOA2 filter and updating user manual text on screen
        // parameters:
        //      note: you will not be able to set filter to null if UniName2 filter changes from 1 value to other, becuase if i click Computer science for Univeristy of cambridge and the I click computer science for Heriot watt, the filter should not be set to null
        //      - u, filter to set, if equal to current filter, set to null
        // returns:
        //      - nothing
        dmObj.setUOA2Filter = function(u){
            if (Uniarray[0] != Uniarray[1]) {
                document.getElementById('manual2').innerHTML = '<p>'+ Uniarray[1]+' UOAs and their overall 4* rating ==> Selected: '+ u;
                UOA2Filter = u;
                Uniarray[0] = Uniarray[1];
            }
            else if(UOA2Filter == u){
                if(UniName2Filter ==null)
                {
                    document.getElementById('manual2').innerHTML = '<p>Heriot Watt UOAs and their overall 4* rating ==> Selected: none';
                }
                else{
                    document.getElementById('manual2').innerHTML = '<p>'+ UniName2Filter + ' UOAs and their overall 4* rating ==> Selected: none';
                }
                UOA2Filter = null;
            } else {
                if (UniName2Filter ==null)
                {
                document.getElementById('manual2').innerHTML = '<p>Heriot Watt UOAs and their overall 4* rating ==> Selected: ' + u;
                }
                else
                {
                    document.getElementById('manual2').innerHTML = '<p>'+ Uniarray[1]+' UOAs and their overall 4* rating ==> Selected: '+ u;
                }
                UOA2Filter = u;
            }
        }

        // print filter for debugging
        dmObj.printUOA2Filter = function(){
            console.log("UOA2Filter = ", UOA2Filter);
        }

        // function setting 4* filter
        // parameters:
        //      - u, filter to set, if equal to current filter, set to null
        // returns:
        //      - nothing
        dmObj.setStar4Filter = function(u){
            if(Star4Filter == u){
                Star4Filter = null;
            } else {
                Star4Filter = u;
            }
        }

        // print filter for debugging
        dmObj.printStar4 = function(){
            console.log("Star4Filter = ", Star4Filter);
        }

        // ---- PRIVATE VARIABLES

        var dataset = []; // dataset array

        // filters
            var UOAFilter = null;
            var UniNameFilter = null;
            var UniName2Filter = null;
            var UOA2Filter = null;
            var UOA3Filter = null;
            var Star4Filter = null;
            var LongFilter = null;
            var LatFilter = null;

        // record changing values for UOA2Filter

            var Uniarray = [];

        // ---- PRIVATE FUNCTIONS

        // function to get data using filters


        function combineCSVdata(ref14data, learningProviders){
            console.log("\n\nFUNCTION: findREFunisWithNoEntryInLearningProvidersPlus\n")

            // For each learning provider university - add learning provider entry as field
            // 'lp' in relevant REF14 table entry
            learningProviders.forEach(processUniversity);
```

```javascript

                function processUniversity(learningProvider){
                    ref14data.forEach(function(ref14entry){
                        if (ref14entry["Institution code (UKPRN)"] == learningProvider.UKPRN){
                            ref14entry.LONGITUDE = learningProvider["LONGITUDE"];
                            ref14entry.LATITUDE = learningProvider["LATITUDE"];
                            ref14entry.VIEW_NAME = learningProvider["VIEW_NAME"];
                            ref14entry.lp = learningProvider;
                        }

                    })
                }
            }

            function findREFunisWithNoEntryInLearningProvidersPlus(ref14data, learningProviders){
                //Find REF universities with no entry in learning-providers-plus.csv
                var listOfUniWithNoEntryInLearningProviders = "";

                console.log("\n\nFUNCTION: findREFunisWithNoEntryInLearningProvidersPlus\n")

                //Get list (array) of REF universities
                refUniversitiesByUKPRN = d3.nest()
                    .key(function(d) { return d["Institution code (UKPRN)"]; })
                    .entries(ref14data);

                // For each REF14 university - see if its PRN is in the Learning Providers' list
                // If not add it to listOfUniWithNoEntryInLearningProviders
                refUniversitiesByUKPRN.forEach(processUniversity);

                function processUniversity(ref14university){
                    learningProviderUni=learningProviders.filter(function(uni){return uni.UKPRN == ref14university.key})
                    if (!learningProviderUni[0]) {
                        //If no entry, accumulate
                        listOfUniWithNoEntryInLearningProviders
                            += "<p>"+ref14university.key
                            + ": "
                            + ref14university.values[0]["Institution name"]
                            +"</p>";
                        console.log("PRN = ", ref14university.key)
                        console.log("ref14university = ", ref14university)
                        console.log("Name = ", ref14university.values[0]["Institution name"] )
                    }
                }
            }

            function displayHWCS(ref14data){
                console.log("\n\nFUNCTION: displayHWCS\n")

                heriotWattCS = ref14data.filter(function(row){
                    return (row["Institution code (UKPRN)"] == 10007764)
                    && (row["Unit of assessment name"] == "Computer Science and Informatics")
                })
                console.log("heriotWattCS = ", heriotWattCS)
            }

            //saves REF14 filter on (heriot watt & overall) to global variable
            function labFourOne(ref14data){
                console.log("\n\nFUNCTION: labFourOne\n")

                HeriotWatt_UOA_Overall = ref14data.filter(function(row){
                    return (row["Institution code (UKPRN)"] == 10007764)
                    && (row["Profile"] == "Overall")
                })

                console.log("HeriotWatt_UOA_Overall = ", HeriotWatt_UOA_Overall)
            }


            function displayOneUniversityAsNestedList(ref14data){
                console.log("\n\nFUNCTION: displayOneUniversityAsNestedList\n")

                var refUniversitiesByName = d3.nest()
                    .key(function(d) { return d["Institution name"]; })
                    .key(function(d) { return d["Unit of assessment name"]; })
                    .key(function(d) { return d["Profile"]; })
                    .entries(ref14data);
                console.log("refUniversitiesByName-UOA-Profile= ", refUniversitiesByName);

                var heriotWatt = refUniversitiesByName.filter(function(uni){
                    return (uni.key == "Heriot-Watt University")})
                console.log("heriotWatt= ",heriotWatt)
            }

            //filter data on UniNameFilter and UOAFilter
            function filteredData(ds){
                return ds.filter(function(entry){
                        return (UniNameFilter === null || UniNameFilter === String(entry["VIEW_NAME"]))
                            && (UOAFilter === null || UOAFilter === String(entry["Unit of assessment name"])); // keep if no year filter or year filter set to year being read,
                    })
            }

            //filter data on UOA3Filter
            function filteredData3(ds){
                return ds.filter(function(entry){
                        return (UOA3Filter === null || UOA3Filter === String(entry["Unit of assessment name"])); // keep if no year filter or year filter set to year being read,
                    })
            }


            return dmObj; // returning the main object
        }
```

piechart_d3v4_v003_lab4.js

```
/*-------------------------------------------------------------------

   Module: piechart class implemented in Bostock's functional style
   Author: Issa Haddad

   What it does:
    Renders a pie chart using the GUP

   Percentage of code written by Issa Haddad: 20%
   Percentage of code taken from course example: 80%

   Dependencies
    D3.js v4

   History: 17/11/2017

   References: (none)

------------------------------------------------------------------- */

function piechart(targetDOMelement, targetDOMelement2, targetDOMelement3) {
    //Here we use a function declaration to imitate a 'class' definition
    //
    //Invoking the function will return an object (piechartObject)
    //    e.g. piechart_instance = piechart(target)
    //    This also has the 'side effect' of appending an svg to the target element
    //
    //The returned object has attached public and private methods (functions in JavaScript)
    //For instance calling method 'updateAndRenderData()' on the returned object
    //(e.g. piechart_instance) will render a piechart to the svg


    //Delare the main object that will be returned to caller
    var piechartObject = {};

    //=================== PUBLIC FUNCTIONS =========================
    //
    piechartObject.overrideDataFieldFunction = function (dataFieldFunction) {
        dataField = dataFieldFunction;
        return piechartObject;
    }

    piechartObject.overrideDataKeyFunction = function (dataKeyFunction) {
        dataKey = dataKeyFunction;
        return piechartObject;
    }

    piechartObject.overrideDataTextFunction = function (dataTextFunction) {
        dataText = dataTextFunction;
        return piechartObject;
    }

    piechartObject.overrideMouseOverFunction = function (callbackFunction) {
        mouseOverFunction = callbackFunction;
        layoutAndRender();
        return piechartObject;
    }

    piechartObject.overrideMouseOutFunction = function (callbackFunction) {
        mouseOutFunction = callbackFunction;
        layoutAndRender();
        return piechartObject;
    }

    piechartObject.render = function (callbackFunction) {
        layoutAndRender();
        return piechartObject;
    }

    piechartObject.loadAndRenderDataset = function (data) {
        dataset=data;
        layoutAndRender();
        updateInteractions();
        return piechartObject;
    }

    piechartObject.loadAndRenderDataset2 = function (data) {
        dataset=data;
        layoutAndRender2();
        updateInteractions();
        return piechartObject;
    }

    piechartObject.sort = function () {
        dataset.sort(function (a,b){
            return dataField(a) - dataField(b)
        })
        layoutAndRender();
        return piechartObject;
    }

    piechartObject.sortR = function () {
        dataset.sort(function (a,b){return dataField(b) - dataField(a)})
        layoutAndRender();
        return piechartObject;
    }

    piechartObject.setMouseclickCallback = function(f){
        mouseclickCallback = f;
                updateInteractions();
                return piechartObject;
            }

            piechartObject.sortKey = function () {
                dataset.sort(function (a,b){
                    if (a.keyField < b.keyField) return -1;
                    if (a.keyField > b.keyField) return  1;
                    return 0;
                });
                layoutAndRender();
                return piechartObject;
            }

            piechartObject.height = function (hei) {
                svgHeight = hei
                arcShapeGenerator
                        .outerRadius(svgHeight/2)
                        .innerRadius(svgHeight/4)
                layoutAndRender();
                return piechartObject;
            }
```

```

           piechartObject.width = function (wid) {
               svgWidth = wid
               layoutAndRender();
               return piechartObject;
           }

           piechartObject.setTooltipText = function(f){
               tooltipText = f;
               updateInteractions();
               return piechartObject;
           }

           //=================== PRIVATE VARIABLES ====================================
           //Width and height of svg canvas
           var svgWidth = 300;
           var svgHeight = 300;
           var pieColour = "steelBlue"
           var dataset = [];
           var mouseclickCallback = function(d, i){ console.log(d, i) };
           var keySelected = null;
           var tooltipText = function(d, i){return "tooltip over element "+i;}
           var mouseoverCallback = function(d, i){ };
           var mouseoutCallback = function(d, i){ };

           var color_scale = d3.scaleOrdinal(d3.schemeCategory20);

           //Declare and append SVG element
           var svg = d3.select(targetDOMelement)
                       .attr("class", "framed2")
                       .attr("width", svgWidth)
                       .attr("height", svgHeight);


           //Declare and append group that we will use tp center the piechart within the svg

           var grp = d3.select(targetDOMelement2);

           var tooltip = d3.select(targetDOMelement3)
               .classed("tooltip", true);

           //=================== PRIVATE FUNCTIONS ====================================

           //var dataField = function(d){return d["FTE Category A staff submitted"]}
           var dataField = function(d){return d["num"]}
           var dataKey = function (d){return d.data["key"]}
           var dataText = function(d) { return d.data["key"]}

           //Set up shape generator
           var arcShapeGenerator = d3.arc()
               .outerRadius(svgHeight/2)
               .innerRadius(svgHeight/4)
               .padAngle(0.03)
               .cornerRadius(8);

           function layoutAndRender(){
               //Taken and addapted from https://github.com/d3/d3-
shape/blob/master/README.md#pie

               //Generate the layout
               var arcsLayout = d3.pie()
                   .value(dataField)
```

```
                    .sort(null)
                    (dataset);

                console.log("Layout=", arcsLayout)


                //center the group within the svg
                grp.attr("transform", "translate("+[svgWidth/2, svgHeight/2]+")")

                //Now call the GUP
                GUP_pies(arcsLayout, arcShapeGenerator);
                //GUP_labels(arcsLayout, arcShapeGenerator);

            }


            function layoutAndRender2(){
                //Taken and addapted from https://github.com/d3/d3-shape/blob/master/README.md#pie

                //Generate the layout
                var arcsLayout = d3.pie()
                    .value(dataField)
                    .sort(null)
                    (dataset);

                console.log("Layout=", arcsLayout)


                //center the group within the svg
                grp.attr("transform", "translate("+[svgWidth/2, svgHeight/2]+")")

                //Now call the GUP
                GUP_pies(arcsLayout, arcShapeGenerator);
                //GUP_labels(arcsLayout, arcShapeGenerator);

            }

            function GUP_labels(arcsLayout, arcShapeGenerator){


                //Bind data
                var labels = grp.selectAll("text")
                    .data(arcsLayout/*, dataKey*/)
                //add text elements for new labels

                var labels_enter = labels.enter().append("text")

                //Define content and location in Update + enter selection
                var merged_labels = labels_enter.merge(labels)
                    //Translate label to center of arc
                merged_labels.attr("transform", function(d) {
                    return "translate(" + arcShapeGenerator.centroid(d) + ")";
                })
                    .attr("text-anchor", "middle")
                    .text(dataText);
                //Remove old labels
                labels.exit().remove();
            };

            function GUP_pies(arcsLayout, arcShapeGenerator){

                //GUP = General Update Pattern to render pies

                //GUP: BIND DATA to DOM placeholders
                var selection = grp.selectAll("path")
                    .data(arcsLayout/*, dataKey*/)
                //GUP: ENTER SELECTION
                var enterSel = selection
                    .enter()
                    .append("path")
                    .each(function(d) { this.dPrevious = d; }); // store d for use in tweening

                //GUP ENTER AND UPDATE selection
                var mergedSel = enterSel.merge(selection)

                mergedSel
                    .style("stroke", "gray")
                    .style("opacity", 0.8)
                    .style("fill",function(d,i){return color_scale(i);})

                mergedSel
                    .transition()
                    .duration(750)
                    .attrTween("d", arcTween); //Use custom tween to draw arcs

                //GUP EXIT selection
                selection.exit()
                    .remove()
            };


            //Ignore this function unless you really want to know how interpolators work
            function arcTween(dNew) {
                //Create the linear interpolator function
                //this provides a linear interpolation of the start and end angles
                //stored 'd' (starting at the previous values in 'd' and ending at the new values in 'd')
                var interpolateAngles = d3.interpolate(this.dPrevious, dNew);
                //Now store new d for next interpoloation
                this.dPrevious = dNew;
                //Return shape (path for the arc) for time t (t goes from 0 ... 1)
                return function(t) {return arcShapeGenerator(interpolateAngles(t)) };
            }


            function updateInteractions(){
                // separate function for interactions,
                // this way there is no need to do GUP when just changing callbacks
                grp.selectAll("path")
                    .on("mouseover", function(d, i){
                        d3.select(this).style("opacity", 1);
                        console.log(d.data["Unit of assessment name"], " && ", d.data["4*"]);

                        tooltip.html(tooltipText(d,i))
                            .style("left", (d3.event.pageX) + "px")
                            .style("top", (d3.event.pageY - 28) + "px")
                            .style("opacity", 0.9);

                        mouseoverCallback(d, i);
                    })
                    .on("mouseout", function(d, i){
                        d3.select(this).style("opacity", 0.7);

                        tooltip.style("opacity", 0);

                        mouseoutCallback(d, i);

                    })
                    .on("click", function(d, i){

                        console.log("keySelected before ");
                        console.log(keySelected);
                        keySelected = (keySelected == d.data.key) ? null : d.data.key;
                        console.log("keySelected after ");
                        console.log(keySelected);
                        mouseclickCallback(d, i);
                    })
            }


            //================== IMPORTANT do not delete ==================================
            return piechartObject; // return the main object to the caller to create an instance of the 'class'

        } //End of piechart() declaration
```

valueTree_v001.js

```
/*-----------------------------------------------------------------------

   Module: tree class implemented in Bostock's functional style
   Author: Issa Haddad

   What it does:
    Renders a tree hierarchy using the GUP
    Assumes input is JSON hierarcy with
    1. 'children' denoted by a 'values[]' array
    2. a single root node

   Percentage of code written by Issa Haddad: 5%
   Percentage of code taken from course example: 95%

   Dependencies
    D3.js v4

   History: 17/11/2017

   References: (none)

------------------------------------------------------------------------ */

function tree(targetDOMelement, targetDOMelement2) {
    //Here we use a function declaration to imitate a 'class' definition
    //
    //Invoking the function will return an object (treeObject)
    //    e.g. tree_instance = tree(target)
    //    This also has the 'side effect' of appending an svg to the target element
    //
    //The returned object has attached public and private methods (functions in JavaScript)
    //For instance calling method 'updateAndRenderData()' on the returned object
    //(e.g. tree_instance) will render a tree to the svg


    //Delare the main object that will be returned to caller
    var treeObject = {};

    //=================== PUBLIC FUNCTIONS =========================
    //


    treeObject.loadAndRenderDataset = function (data) {
        jsonTreeData=data;
        layoutAndRender();
        return treeObject;
    }

    treeObject.leafLabelFn = function (fn) {
        leafLabel=fn;
        return treeObject;
    }

    treeObject.appendToClick = function (fn) {
        appendedClickFunction=fn;
        return treeObject;
    }


    //=================== PRIVATE VARIABLES ====================================

    //Declare and append SVG element
    var margin = {top: 20, right: 200, bottom: 20, left: 20},
    width = 900 - margin.right - margin.left,
    height = 800 - margin.top - margin.bottom;

    //Set up SVG and append group to act as container for tree graph

    var svg = d3.select(targetDOMelement)
        .attr("class", "framed3")
        .attr("width", width + margin.right + margin.left)
        .attr("height", height + margin.top + margin.bottom);

    var grp = d3.select(targetDOMelement2)
        .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

    var hierarchyGraph,
        sourceNode,
        listOfLinksByDescendants,
        listOfNodes;


    //=================== PRIVATE FUNCTIONS ====================================

    function layoutAndRender(){
        //GENERATE THE DOUBLY LINKED HIERARCHY GRAPH
        hierarchyGraph = d3.hierarchy(jsonTreeData, function(d){return d.values});
        //Hide all grandchildren nodes at start to produce compact layout
        //BETTER TO DO THIS WITH A RECURSIVE FUNCTION!!!!
        hierarchyGraph.children.forEach(function (child){
            child.children.forEach(function(grandchild){
                grandchild.children.forEach(hideUnhideChildren)
                hideUnhideChildren(grandchild)
            });
            hideUnhideChildren(child)
        });
        //hierarchyGraph.children.forEach(hideUnhideChildren);
        //Set source node to root of hierarchy to start
        sourceNode = hierarchyGraph;
                //And set it's 'old' position to (0,0) as we'll
                //start drawing the tree from here
                sourceNode.xOld=sourceNode.yOld=0;
                //Add (x,y) positions and render
                calculateXYpositionsAndRender(sourceNode);
            }


            function calculateXYpositionsAndRender(sourceNode){
                //Note that the 'sourceNode' is the clicked node in a collapse or
                //uncollapse animation

                //get and setup the tree layout generator and generate (x,y,) data
                var myTreeLayoutGenerator = d3.tree().size([height, width]);
                var treeLayout = myTreeLayoutGenerator(hierarchyGraph);

                //Get lists of nodes and links
                listOfLinksByDescendants = treeLayout.descendants().slice(1);
                listOfNodes = treeLayout.descendants();

                //RENDER
                renderLinks(listOfLinksByDescendants, sourceNode);
                renderNodes(listOfNodes, sourceNode);

                // Store the old positions for collapsable transition.
                listOfNodes.forEach(function(d){
                    d.xOld = d.x;
                    d.yOld = d.y;
                });
            }

            function renderLinksAndNodes(){
                renderLinks(listOfLinksByDescendants, sourceNode);
                renderNodes(listOfNodes, sourceNode);
            }

            function renderNodes(listOfNodes, sourceNode){

                //DATA BIND
                var selection = grp
                    .selectAll("g.classNode")
                    .data(listOfNodes, generateUniqueKey);

                //ENTER
                var enterSelection = selection.enter()
                    .append("g")
                    .on("click", onClick)
                    .classed("classNode", true);

                //transitions
                enterSelection
                    .attr("transform", function(d) {
                        return "translate(" + sourceNode.yOld + "," + sourceNode.xOld + ")";

                    })
                    //Transition to final entry positions
                    .transition()
                    .duration(2000)
                    .attr("transform", function(d) {
                        return "translate(" + d.y + "," + d.x + ")";
                    });

                //Append nodes
                enterSelection.append("circle")
                    .attr("r", 5);

                //Append associated node labels
                enterSelection
                    .filter(function(d){return !(d.height == 0)})
                    .append("text")
                    .attr("y", -8)
                    .attr("text-anchor", "middle")
                    .text(function(d) {return d.data.key;});

                //Append associated leaf node labels
                enterSelection
                    .filter(function(d){return (d.height == 0)})
                    .append("text")
                    .attr("x", 13)
                    .attr("text-anchor", "start")
                    .text(leafLabel);

                //UPDATE
                selection
```

```
                    .on("click", onClick)
                    .transition()
                    .duration(2000)
                    .attr("transform", function(d) {
                        return "translate(" + d.y + "," + d.x + ")";
                    });

                // EXIT
                selection
                    .exit()
                    .transition()
                    .duration(2000)
                    .attr("transform", function(d) {
                        return "translate(" + sourceNode.y + "," + sourceNode.x + ")";
                    })
                    .remove();
            }

            function renderLinks(listOfLinksByDescendants, sourceNode){

                // DATA JOIN
                var selection = grp
                    .selectAll("path.classLink")
                    .data(listOfLinksByDescendants, generateUniqueKey);

                //ENTER
                var enterSel = selection
                    .enter()
                    .append('path')
                    .classed("classLink", true);

                enterSel
                    .attr('d', function(d){
                        var o = {x:sourceNode.xOld, y:sourceNode.yOld}
                        return diagonalShapeGenerator2(o, o);
                    })
                    .transition()
                    .duration(2000)
                    .attr("d", diagonalShapeGenerator1);

                // UPDATE
                selection
                    .transition()
                    .duration(2000)
                    .attr("d", diagonalShapeGenerator1);

                // EXIT
                selection
                    .exit()
                    .transition()
                    .duration(2000)
                    .attr('d', function(d){
                        return diagonalShapeGenerator2(sourceNode, sourceNode);
                    })
                    .remove();

            }

            //Click callback
            var onClick = function (d,i){
                hideUnhideChildren(d);
                calculateXYpositionsAndRender(d);
                appendedClickFunction(d);
            }

            //Additional click callback functionality
            var appendedClickFunction = function(){
                //it's intended that this is defined by
            };

            var leafLabel = function(d) {return "leafLabel"}

            // Toggle children on click.
            function hideUnhideChildren(d) {
                if (d.children) {
                    d._children = d.children;
                    d.children = null;
                } else {
                    d.children = d._children;
                    d._children = null;
                }
            }

            //Diagonal path shape generator function
            function diagonalShapeGenerator1(d){
                source = d.parent;
                descendant = d;
                return diagonalShapeGenerator2(source, descendant);
            }

            //Diagonal path shape generator function
            function diagonalShapeGenerator2(source, descendant){
                return "M" + source.y + "," + source.x
                    + "C" + (source.y + descendant.y) / 2 + "," + source.x
                    + " " + (source.y + descendant.y) / 2 + "," + descendant.x
                    + " " + descendant.y + "," + descendant.x;
            }

            //Define key generator
            var lastKey=0; // for keeping track of unique key
            function generateUniqueKey(d) {
                if(!d.hasOwnProperty("key")) d.key = ++lastKey;
                return d.key;
            }

            //================== IMPORTANT do not delete ==================================
            return treeObject; // return the main object to the caller to create an instance of the 'class'

        } //End of tree() declaration
```

Map.js

```
/*-------------------------------------------------------------------

   Module: map class implemented in Bostock's functional style
   Author: Issa Haddad

   What it does:
    Renders a zoomable uk map using the GUP
    Assumes input is UK json file

   Percentage of code written by Issa Haddad: 60%
   Percentage of code taken from course example: 40%

   Dependencies
    D3.js v4
    topojson.v1.min.js V1

   History: 17/11/2017

   References:
    -https://bl.ocks.org/iamkevinv/0a24e9126cd2fa6b283c6f2d774b69a2

---------------------------------------------------------------------- */
function map(DOMElement, DOMElement2, DOMElement3){


    var Learning;
    var mapObj = {}; // main object
    var Long;
    var Lat;
    // function loading data and rendering it in bar chart
    // parameters:
    //      - dataset in following format:
    //          [{"key": year, "value": money}, {...}, ...]
    // returns:
    //      - barObj for chaining
    mapObj.loadAndRender = function(uk){
        var countries = topojson.feature(uk, uk.objects.subunits).features;
        //Ditto for places (towns and cities)
        var towns = topojson.feature(uk, uk.objects.places).features;

        console.log("uk = ", uk); console.log("countries = ", countries); console.log("towns = ", towns);
        GUP_countries(svg, countries);
        GUP_towns(svg, towns);
        updateInteractions();
        return mapObj;
    }

    mapObj.loadInteraction2 = function()
    {
        updateInteractions2();
    }

    mapObj.setlp = function(l){
        Learning = l;
        return mapObj;
    }

    mapObj.getlp = function(l){
        return Learning;
    }

    mapObj.setMouseclickCallback = function(f){
        mouseclickCallback = f;
        updateInteractions();
        return mapObj;
    }

    mapObj.setMouseclickCallback2 = function(f){
        mouseclickCallback = f;
        updateInteractions();
        updateInteractions2();
        return mapObj;
    }

    // modifies funcion in .on(mousedown)
    mapObj.setUpCallback = function(f){
        func = f;
        updateInteractions();
        updateInteractions2();
        return mapObj;
    }



    // ---- PRIVATE VARIABLES

    // sizing vars
    var width = 960,
        height = 1227;
        active = d3.select(null);

    // dom elements
    var svg = d3.select(DOMElement) //.append("svg")
        .attr("class", "framed")
        .style("float", "left")
        .attr("width", width)
        .attr("height", height)
        .on("click", stopped, true);


           var rec = d3.select(DOMElement2)
               .attr("class", "background")
               .attr("width", width)
               .attr("height", height)
               .on("click", reset);


           //define projection of spherical coordinates to the Cartesian plane
           var projection = d3.geoAlbers()
           .center([0, 55.4])
           .rotate([4.4, 0])
           .parallels([50, 60])
           .scale(1200 * 5)
           .translate([width / 2, height / 2]);

           //Define path generator (takes projected 2D geometry and formats for SVG)
           var pathGen = d3
           .geoPath()
           .projection(projection)
           .pointRadius(2);
```

```


            //zoom
            var zoom = d3.zoom()
            .scaleExtent([1, 8])
            .on("zoom", zoomed);

            var g = d3.select(DOMElement3);

            svg.call(zoom);

            // data
            var dataset = [];
            var keySelected = null;
            var mouseclickCallback = function(d, i){ console.log(d, i) };
            var func = function(){ };
            // ---- PRIVATE FUNCTIONS

            function GUP_countries(svg, countries){
            //Draw the five unit outlines (ENG, IRL, NIR, SCT, WLS)

            //DATA BIND
            var selection = g
                .selectAll(".classCountry")
                .data(countries);

            //ENTER
            var enterSel = selection
                .enter()
                .append("path")
                .attr("class", function(d) { return d.id; })
                .classed("classCountry", true)
                .attr("d", pathGen)
                .on("click", clicked)

            //UPDATE
            //Nothing for now

            //EXIT
            //Nothing for now
            }

            function GUP_towns(svg, towns){

            //DATA BIND
            var selection = g
                .selectAll("g.classTown")
                .data(Learning, getKey);

            //ENTER
            var enterSelection = selection.enter()
                .append("g")
                .classed("classTown", true)
                .attr("transform", function(d) {
                    return "translate(" + projection([d.LONGITUDE,d.LATITUDE]) + ")";
                });

            //Append circles
            enterSelection.append("circle")
                .attr("r", 4);

            //Append labels
            enterSelection.append("text")
                .attr("x", 6)
                .attr("y", 4)
                .attr("text-anchor", "start")
                .style("font-size", "10px")
                .text(function(d) {return d.VIEW_NAME;});


            //UPDATE
            //Nothing for now

            selection.exit()
                    .remove()

            }

            function getKey(d){
            return d.VIEW_NAME;
        }

            function clicked(d) {
                if (active.node() === this) return reset();
                 active.classed("active", false);
                 active = d3.select(this).classed("active", true);

                 var bounds = pathGen.bounds(d),
                   dx = bounds[1][0] - bounds[0][0],
                   dy = bounds[1][1] - bounds[0][1],
                   x = (bounds[0][0] + bounds[1][0]) / 2,
                   y = (bounds[0][1] + bounds[1][1]) / 2,
                   scale = Math.max(1, Math.min(8, 0.9 / Math.max(dx / width, dy / height))),

                   translate = [width / 2 - scale * x, height / 2 - scale * y];

                svg.transition()
                   .duration(750)
                      // .call(zoom.translate(translate).scale(scale).event); // not in d3 v4

                     .call( zoom.transform, d3.zoomIdentity.translate(translate[0],translate[1]).scale(scale) ); // updated for d3 v4
        }

            function reset() {
                active.classed("active", false);
                active = d3.select(null);

                svg.transition()
                .duration(750)
                // .call( zoom.transform, d3.zoomIdentity.translate(0, 0).scale(1) ); // not in d3 v4
                .call( zoom.transform, d3.zoomIdentity ); // updated for d3 v4
            }

            function zoomed() {
                g.style("stroke-width", 1.5 / d3.event.transform.k + "px");
                // g.attr("transform", "translate(" + d3.event.translate + ")scale(" + d3.event.scale + ")"); // not in d3 v4
                g.attr("transform", d3.event.transform); // updated for d3 v4
            }

            function stopped() {
                if (d3.event.defaultPrevented) d3.event.stopPropagation();
            }

            function updateScales(){
                xScale.domain(dataset.map(function(d){return d.key;}))
                    .paddingInner(0.1)
                    .range([svgPadding, svgWidth-svgPadding]);

                yScale.domain([0, d3.max(dataset, function(d){return d.value;})])
                    .range([0, svgHeight-svgPadding*2]);
            }

            function updateInteractions2()
            {
                g.selectAll(".classCountry")
                    .on("mousedown.log", function() {
                    var Long = projection.invert(d3.mouse(this))[0];
                    var Lat = projection.invert(d3.mouse(this))[1];
                    console.log("long: ", Long);
                    console.log("lat: ", Lat);
                    dataManager.printlongFilter();
                    dataManager.setlongFilter(Long);
                    dataManager.printlongFilter();
                    dataManager.printlatFilter();
                    dataManager.setlatFilter(Lat);
                    dataManager.printlatFilter();
                    func();
                    });

                }

            function updateInteractions(){
                // separate function for interactions,
                // this way there is no need to do GUP when just changing callbacks


                g.selectAll("g.classTown")
                    .on("click", function(d, i){
                        console.log("bef");
                        console.log(keySelected);
                        console.log("d:",d);
                        keySelected = (keySelected == d["VIEW_NAME"]) ? null : d["VIEW_NAME"];

                        console.log("bef");
                        console.log(keySelected);
                        mouseclickCallback(d, i);
                    })
            }

            return mapObj; // returning the main object
        }
```

Main.css

```css
* {
    font-family: 'Open Sans';
    margin: 0;
    padding: 0;
}

#dashboard {
  width:1975px;
  height: 1420px;
  clear:both;
  background-color: #f2f2f2;
  border: 15px solid #ecd9c6;
  margin-left: 200px;
  margin-top: 200px;
}

#manual {
  width:960px;
  height: 50px;
  border: 1px solid black;
  margin-left: 50px;
  margin-right: 10px;
  float: left;
  font-size: 20px;
  text-align: center;
  line-height: 50px;
}

#manual2 {
  width:899px;
  height: 50px;
  border: 1px solid black;
  margin-right: 52px;
  float: right;
  font-size: 20px;
  text-align: center;
  line-height: 50px;
}

#manual3 {
  width:901px;
  height: 50px;
  border: 1px solid black;
  margin-top: 5px;
  margin-right: 80px;
  margin-left: 1021px;
  margin-bottom: 10px;
  font-size: 20px;
  text-align: center;
  line-height: 50px;
}

#wrapper {
  width:2000px;
  clear:both;
}

button{
    margin: 15px 180px 20px 180px;
    padding: 10px 30px 10px 30px;
    font-size: 20px;
}

h1 {
    text-align: center;
    padding: 10px 0;
}

div.barchart{
    text-align: center;
}

div.barchart > svg{
    margin: auto;
}

div.tooltip{
    position: absolute;
    color: #fafafa;
    background-color: #424242;
    padding: 10px 5px;
    border-radius: 4px;
    pointer-events: none;
    opacity: 0;
    font-size: 1.2em;
}

text {
      font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
      font-size: 15px;
      //pointer-events: none;
    }
    .classCountry.SCT { fill: thistle; }
    .classCountry.WLS { fill: lightsteelblue; }
    .classCountry.NIR { fill: paleturquoise; }
    .classCountry.ENG { fill: pink; }
    .classCountry.IRL { fill: lightgrey; }

    .classTown text {
                fill: grey;
            }
            .classTown circle {
                stroke: grey;
                fill: white;
            }
            .framed {
             border: 1px solid black;
             margin: 10px 10px 50px 50px;
            }

            .framed2 {
             border: 1px solid black;
             margin: 10px 15px 0px 0px;
             padding: 27px 302px 27px 297px;
            }
            .framed3 {
             border: 1px solid black;
            }


        .background {
          fill: none;
          pointer-events: all;
        }

        .classLink {
                        fill: none;
                        stroke: steelblue;
                        stroke-width: 1.5px;
        }
        .classNode text {
                        fill: steelblue;
                        font: 10px sans-serif;
        }
        .classNode circle {
                        stroke: steelblue;
                        fill: pink;
                        stroke-width: 1.5px;
        }
        .classNode {
                        cursor: pointer;
        }
```