

# Design Documentation - Job Search Web Scraper

## **Documentation Information:**

- Documentation version 1.2
  - Date of the last update: 26/01/2023
  - **Author:** EL SINGABY Waël S22005439
- 

## **Table of contents:**

### **1 - Introduction:**

### **2 - User Interface:**

#### 2.1 - User Interface Context:

- 2.1.1 - Purpose:
- 2.1.2 - Target Users / Audience:
- 2.1.3 - Interaction Design:

#### 2.2 - Wireframe Low-Fidelity:

- 2.2.1 - Find job form:
- 2.2.2 - Result of the search form:

### **3 - Requirements / System Overview:**

#### 3.1 - Functional requirements:

- 3.1.1 - Data Source and Scraped Information:
- 3.1.2 - Job Search Functionality
- 3.1.3 - Display Job Data Functionality
- 3.1.3 - Recent searches Functionality
- 3.1.4 - Wishlist Functionality

#### 3.2 - Technical detail:

- 3.2.1 - Technologies used:
- 3.2.2 - Libraries used:
- 3.2.3 - Web scraper interaction:
- 3.2.4 - Data Extraction:
- 3.2.5 - UML Class:

### **4 - Software Testing:**

#### 4.1 - Web Scraper Testing:

#### 4.2 - User Input Testing:

- 4.2.1 - Random User Input Testing:
- 4.2.2 - Salary, Job Type, Work hour Input Testing:

#### 4.3 - Performance Software:

##### 4.3.1 - Request Testing:

#### **5 - Conclusion:**

---

## **1 - Introduction:**

This design document puts into perspective the development of a **Web Scraper in C#** which aims to **scrape specific data** coming from a **website making job offers**. The purpose of this Web Scraper is to **scrape some data (job offer) of which the user is interested** in order to find a job which would be profitable to him and all this in a simple and effective way. This documentation will be divided into several parts where we will talk about the user interface, the requirement, technical detail, and testing of the software.

---

## **2 - User Interface:**

### **2.1 - User Interface Context:**

#### **2.1.1 - Purpose:**

The **primary purpose of this web scraper** is to **scrape the data that the user wants**, here in our case these are **job offers**. For example the user can put several search criteria two are mandatory: the location of the job and as well as the title of the job (eg Web designer, Full Stack and more...). Other options are highlighted by the software which allows further research such as the starting salary, work hour (part time, full time) and the type of job (Contract, Permanent and Temporary), This will allow the user to **find a job offer that suits him better**.

Several information will be displayed to the user when the search has been made, such as the **title of the job, its location, the salary, a brief description, and the type of the job**.

The searches made by the user will then be **stored using a convenient feature that allows you to keep track of your search (history)**. Additionally the user can put the job offers that sees in front of him in a "**Wish list**" which allows the user to put job offers that interest him in order to come back later for example and apply.

### **2.1.2 - Target Users / Audience:**

First of all, the target audience is **mainly people looking for a job** in order to get into or get back into active life. So all job seekers between the ages of 18 and 35 will be the most targeted because they have not necessarily entered the world of work yet.

As for the user of the software, **there are no prerequisites**, you just have to know how to use a browser because the software acts as a browser in a way.

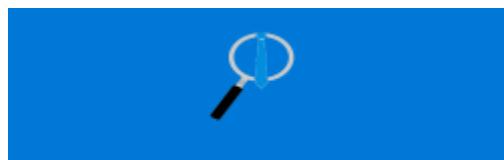
If you need a job and want to have a search personalized to your needs then this software is for you. You will find the **opportunity to have a job offer that suits you**. You don't need to interact a lot with the software because it simply does the job for you, all this to avoid consuming your time.

### **2.1.3 - Interaction Design:**

This section will **highlight the design choices** to deliver user friendly software and create an intuitive user interface. Everything has been done so that the user can have a **good time and so that he can get the information he needs**.

#### **Layout:**

*Layout Find Job Form*



*Logo of the Web Scraper Job Search*

When the **logo** was created I wanted it to be **simple** and **effective** and above all **recognizable** so that the user could remember it and therefore put a link between the **logo and the software**. It features a magnifying glass on a blue background, with the magnifying glass wearing a small tie. The **color blue** corresponds to **the idea of searching and finding something that the user needs**.

I put the logo at the top right for a very good reason because **it's simply the first thing the user will see because it's innate**. We have looked from left to right since we were little .

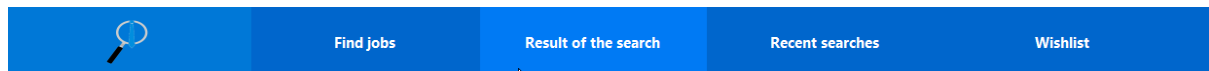
The layout of every form has been designed so that it is as **intuitive as possible for the user and thus more easily navigable**.

On each form a **navigation menu** is put so that the user can go on each of the forms of the software.

*Layout Navigation Bar (Every Form)*

Several forms can be accessed such as "**Find Job**", "**Result of the search**", "**Recent Search**", "**Wishlist**". The placement of this navigation bar aims to be the **most logical** so that it is **closest to a website** (like a navbar in the end). This allows the user to find the information and access the features they need.

The **"Find jobs"** button provides access to the page allowing you to **search for a job offer according to several criteria** set out above and below[1]. The **"Result of the search"** page takes the user to the results of his search, he **sees the job offers corresponding to his search**. The **"Recent Search"** (History) button provides access to the user **history** of past searches. The **"Wishlist"** button allows the user to access the **job offers that he has put in favorites in the past**. Overall the menu navigation has been designed to be most easily used by users.



*Layout Navigation Bar when we place our mouse over an element.*

Furthermore, the navigation menu has a **hover effect when the user hovers his mouse over an element**, which lets him know that this button is well made to be interacted with and thus know where to click.

Now let's talk about the layout of the **first page which is the most important for doing the job search**. This is the default page, it is the page the user accesses first when launching the software. He can also subsequently access it using the navigation menu using the **"Find jobs"** button.

*Layout Find Job Form*

So here is the main section which allows the user to do a **simple or advanced search (Advanced Option)** in order to **find job offers that correspond to him**.

One of the main goals of this form is that it should be very intuitive for the user. On the left we can **see the advanced options which are not mandatory**, that is to say that the user can simply not use them and in **the middle the so-called essential information** that the

user must put under penalty of error. The necessary information is the location of the job and the title of the job he wants. This may very well be a general area such as cybersecurity or more specific such as "Waiter in a restaurant".


As you can see the design of this **form looks very similar to several sites such as LinkedIn Indeed, Totaljob** and more... This decision is a logical decision because as soon as the user comes to the software **it will be directly that this one is done to search for job offers** and he will even be more apt to take control of the software's functionalities.





*Little icons in the Job Form*

So that the user can find himself I put **small icons corresponding to the input that the user must put**. Here there is a "search" icon for the desired job and a location icon for the city where the job offer should be located.



**Advanced Job Search**

**Job Type**

☒ Contract  
☒ Permanent  
☐ Temporary

**Work Hour**

☐ Part Time  
☒ Full Time

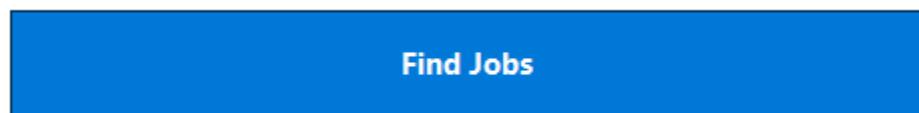
**Salary**

*Advanced Option (Job Find Form)*

To continue if the user uses the advanced options this has a fairly specific design where the **user sees the confirmations of the choices he has made**. For example if he clicked on a checkbox then a small icon inside it will show him that he has confirmed his choice and will be used in his next search. This can of course be unchecked by the user if he wishes to change his search. This **lets the user know what their query will look like and what the results of it will be**.

As you can see there are **several options and several categories** such as the **type of job** (Contract, Permanent, Temporary), the **hours of work** (part time, full time) and a **starting salary**.

All the options are categorized allowing the user to choose which options according to his criteria. The placement of its categories has been designed logically and consistently. The **white color is highlighted on the blue in order to show the user the options and category that there is**. The color of the text of the categories is also put in white in order to bring out this one.



*Find job button (Find Job Form)*

The Find Jobs button was put in the middle for a very good reason because it's the **central piece of our software, looking for job offers**. I made the **button have a blue background** because it indicates that the **user wants to search for something**. In addition, the **color blue** corresponds to the **reliability of the software**.

After the user **clicks on the "Find Jobs" button it will be redirected to another Form "Result of the Search"** which will show the results according to the criteria that the user has chosen. The predisposition of the offers is made to be in the most organized way.

Find jobs Result of the search Recent searches Wishlist

**Senior Design Engineer**  
Reading, Oxfordshire  
£42,650 to £65,000

Are you an experienced Design Engineer interested in delivering the Engineering content across multiple Facilities within the Conventional and Nuclear Projects#39; portfolio? If so, this could be the role for you! This is a great opportunity to work with a...

**Mechanical Design Engineers**  
Reading, Oxfordshire  
£33,850 to £50,800

**Senior Design Engineer**  
Reading, Oxfordshire  
[Apply Now](#) [Wish](#)

**Job details**  
£42,650 to £65,000

**Job Overview**  
Are you an experienced Design Engineer interested in delivering the Engineering content across multiple Facilities within the Conventional and Nuclear Projects#39; portfolio? If so, this could be the role for you! This is a great opportunity to work with a...

AWE is looking for Mechanical Design

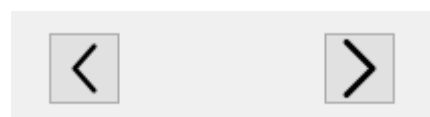
*Result of the search Form*

As you can see here it finds **job offers related to the user search**. The page makes sure to display at least **three jobs that the user can see by scrolling down with their mouse wheel**.

Here the research was done **using two pieces of information**:

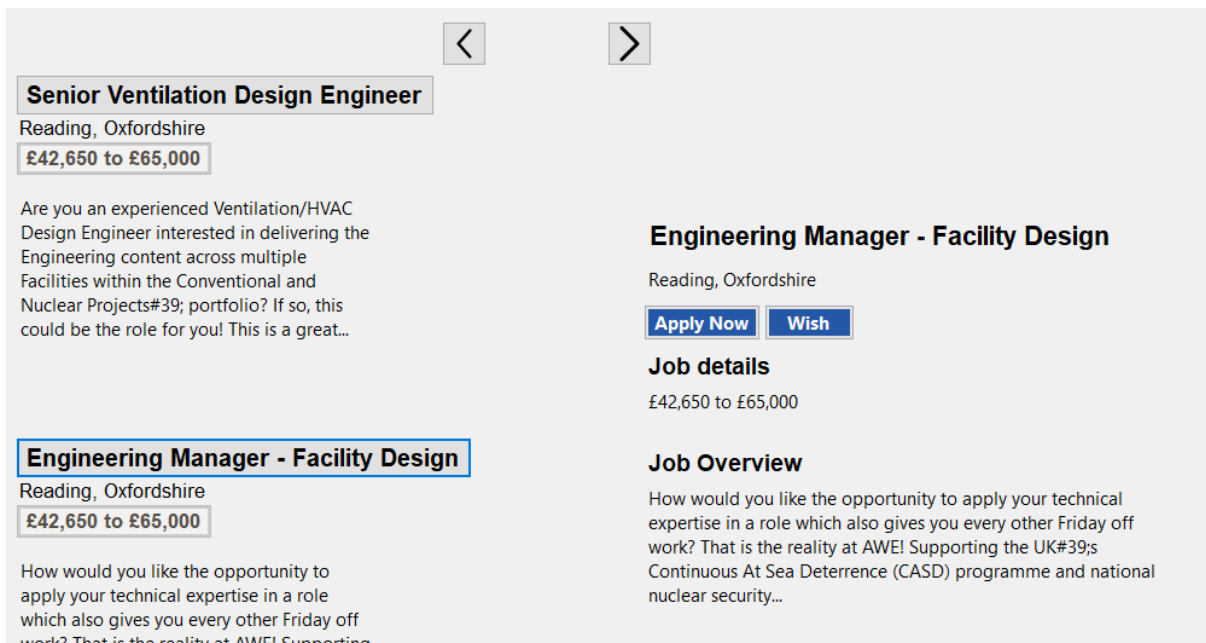
- **Job title:** Design engineer
- **Job location:** London

If the user wishes to see other jobs in addition to the three in front of him, he **can interact with two buttons** which allow him to go to the next job offered by our scraper software. The two buttons are simply two icons: **an arrow that goes to the right and another to the left**. The one on the right allows you to see the next three job offers and the arrow on the left the 3 previous ones.



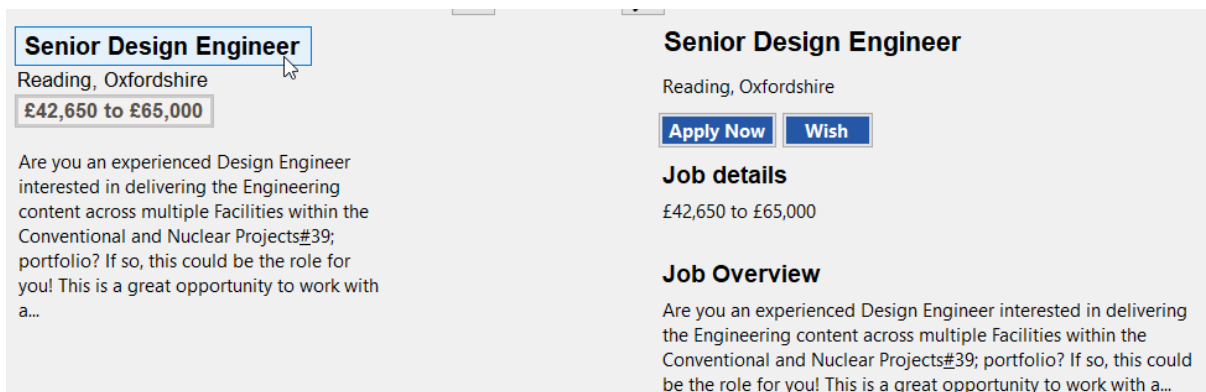
*Arrow that allows you to navigate between job offers.*





*Other job offers if you click on the following button (right arrow).*

Each job offer has been organized in a clear and concise manner so that the user can find their way around. On the **left we can each job offer with a title, salary, location and a brief description**. As for the one **on the right**, it is when the user clicks on one of the job offer titles that even **more information organized according to the job will be displayed**.



*Button hover on title of the job offer*

In addition I made sure that when the **user passes his mouse over the title of the job offered it changes color** in order to make it known that the user can click here to have more information on the job and have access to its options.

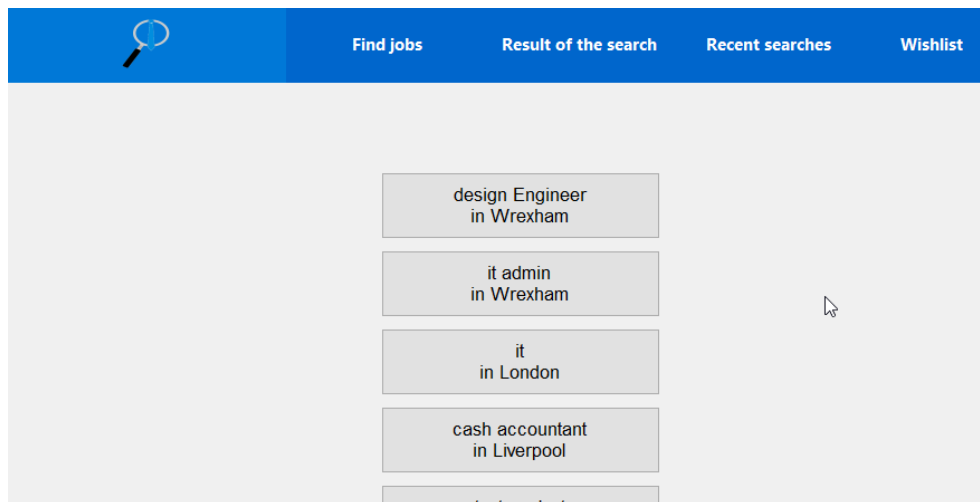
In the more detailed job offer descriptions there are **two buttons that appear in blue**:



*Interactive Button for the user*

- **Apply Now** which is a button that will allow the user to be **redirected to the site** that made the **job offer** and therefore to send his CV there.
- **Wish button** which is a button that allows the user to **add the job offer** in his **wish list will be stored in the software**.

Following the user's search, he can, if he wishes, **access the history of his searches**, which allows him to have an **overview of what he has been able to search for in the past**.

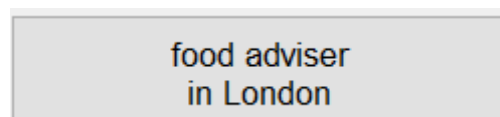


*History Form (Recent searches)*

This form can be accessed at any time by the user by clicking on the **"Recent searches"** button on the **navigation menu**.

The organization of this page has been made so as not to be too **cumbersome for the user**, we provide him with simple things so that he can easily find his way around and make him want to review the job offers in relation to his history. The history **buttons are arranged from the most recent search to the oldest** and there cannot be more than 20 in the list because after 20 this means that the user is not interested enough in relation to the offers he has seen in the past.

If the user clicks on one of the buttons such as **"Food Adviser in London"** it will redirect him in the **"Result of the search"** form which will display the job offers linked to the button. Here will be jobs related to the "Food Adviser" profession and which is located in London or its surroundings.



*Button Food Adviser in the History Form (Recent searches)*

Here are the results linked to the button coming from the history that we have chosen.

[Find jobs](#)
[Result of the search](#)
[Recent searches](#)
[Wishlist](#)

<

>

**Self Employed Mortgage and Protection Adviser**

Earley, Oxfordshire

Just Mortgages is part of Spicerhaart: the UK's largest Independent Property Group. With its rapidly growing financial services business, Just Mortgages, our reputation has grown to be one of the leaders in both innovation and technology in its industry....

**Food and Beverage Supervisor**

Henley-on-thames, Oxfordshire

£24,500

Job Title Food and Beverage Supervisor

**Food and Beverage Supervisor**

Henley-on-thames, Oxfordshire

£24,500

Job Title Food and Beverage Supervisor

**Food and Beverage Supervisor**

Henley-on-thames, Oxfordshire

£24,500

Job Title Food and Beverage Supervisor

**Food and Beverage Supervisor**

Henley-on-thames, Oxfordshire

£24,500

Job Title Food and Beverage Supervisor

Result when we click on the button

Finally we will talk about the last page (form) which allows the **user to save the job offers that interest him**. Each job offer is added if the user clicks on the **"Wish"** button. This list can be accessed at any time by pressing the **"Wishlist"** button **on the navigation menu**.

[Find jobs](#)
[Result of the search](#)
[Recent searches](#)
[Wishlist](#)

	Job title	Job location	Job salary	Apply Link	link
▶	IT Systems Sup...	Wrexham, Clwyd	£27,000 to £30,...	https://starjobs...	<a href="#">Apply Now</a>
	Sales Administr...	Thame, Oxfords...	£24,000 to £28,...	https://starjobs...	<a href="#">Apply Now</a>
	Desktop Suppo...	Wrexham, Clwyd	£180 to £210 Pe...	https://starjobs...	<a href="#">Apply Now</a>
*					<a href="#">Apply Now</a>

Clear

Wishlist Form

The wishlist form aims to give an **overview of the job offers that interest him**. At its center there is a datagrid allowing to find the necessary information related to the job offers that the user has registered. Each line of the **datagrid contains information such as job title**,

**salary, location and a button to access the advertiser link.** The datagrid allows you to have an organized overview instead of putting labels all the time.

The form also has a **clear button** with a blue background **that allows the user to remove a line from the datagrid that he has previously selected.**

This allows the user to be able to **organize themselves** if they are no longer interested in the offer in question.

## **2.2 - Wireframe Low-Fidelity:**

To provide an overview of the design process, a **wireframe low-fidelity was created before the final design was implemented.** This wireframe allowed me to help me on the final layout. He was a reference and I was able to improve certain things on top of that.

### **2.2.1 - Find job form:**

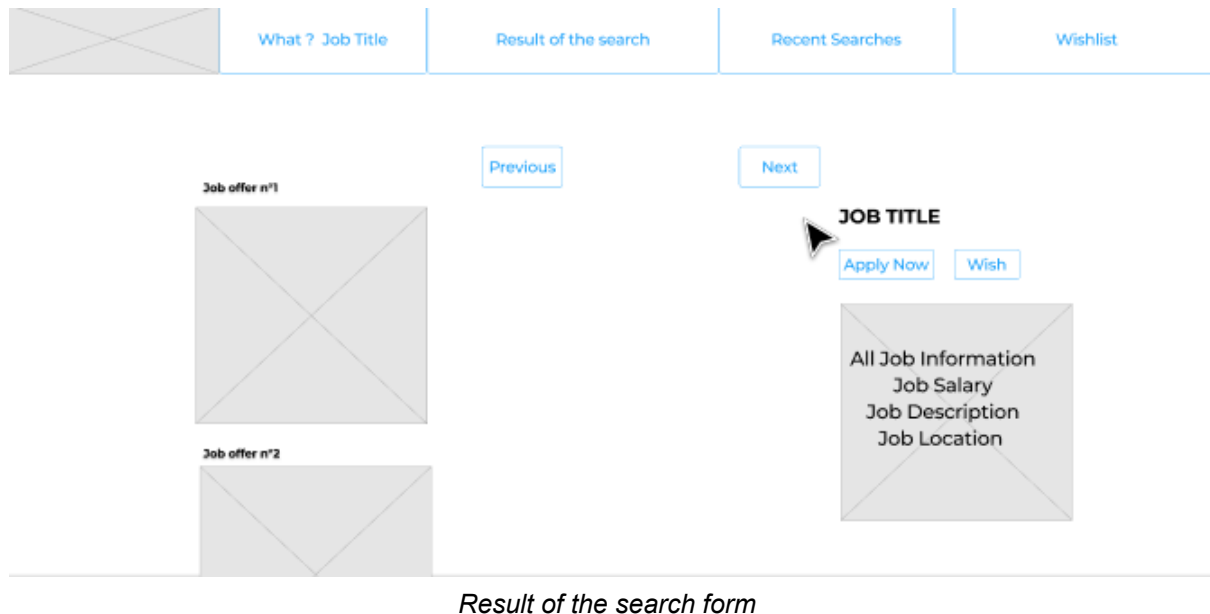
The wireframe shows a navigation bar at the top with four tabs: 'What ? Job Title', 'Result of the search', 'Recent Searches', and 'Wishlist'. Below the navigation bar is a section titled 'Advanced Job Search'. On the left side of this section are four filter categories: 'Job Type', 'Work Hour', and 'Salary', each with a placeholder box. To the right of these filters are two input fields: 'What ? Job Title' and 'Where ? Job Location', each followed by a small square icon. At the bottom center of the search area is a large blue button labeled 'Find Jobs'.

*Wireframe low-fidelity of the "Find job" form*

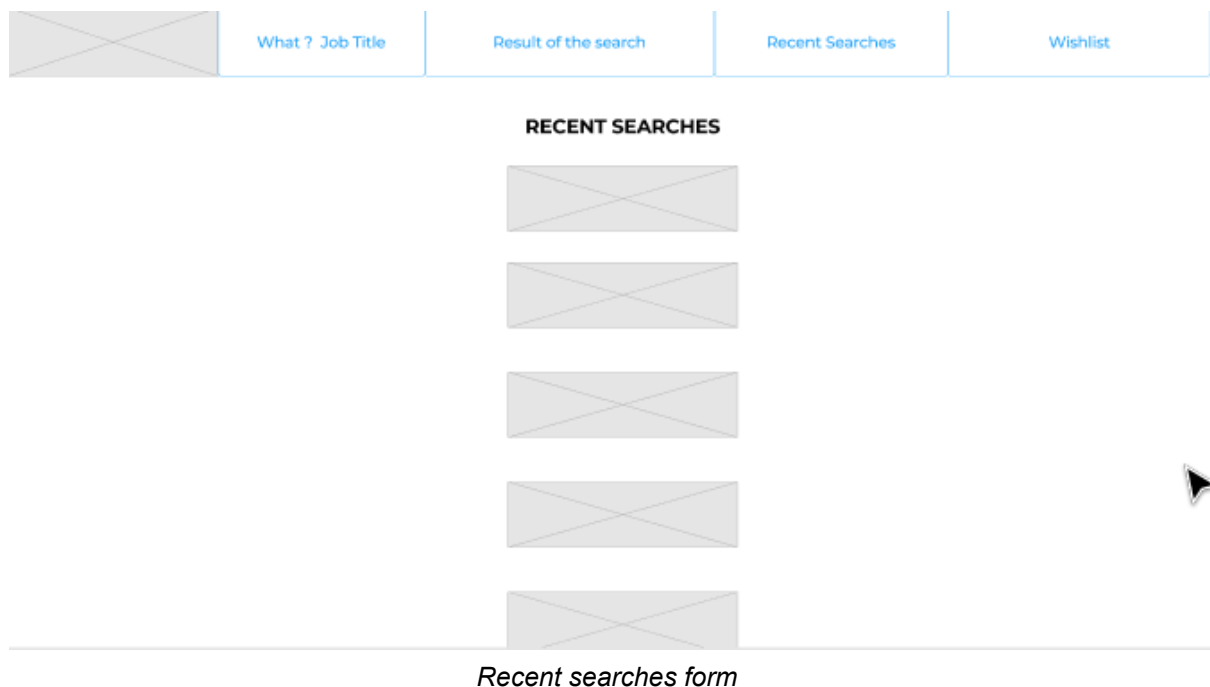
This wireframe **includes the main input fields of the job search**, including the job title and the job location. We can also see **on the left the advanced options** which are not there yet because I did not know at the time the options it would have but I knew there would be that's why there are squares predisposed to this function.

We can see the layout of the navigation **menu at the top of the page** and as well as the central button "**Find jobs**" which allows you to **perform the search**.

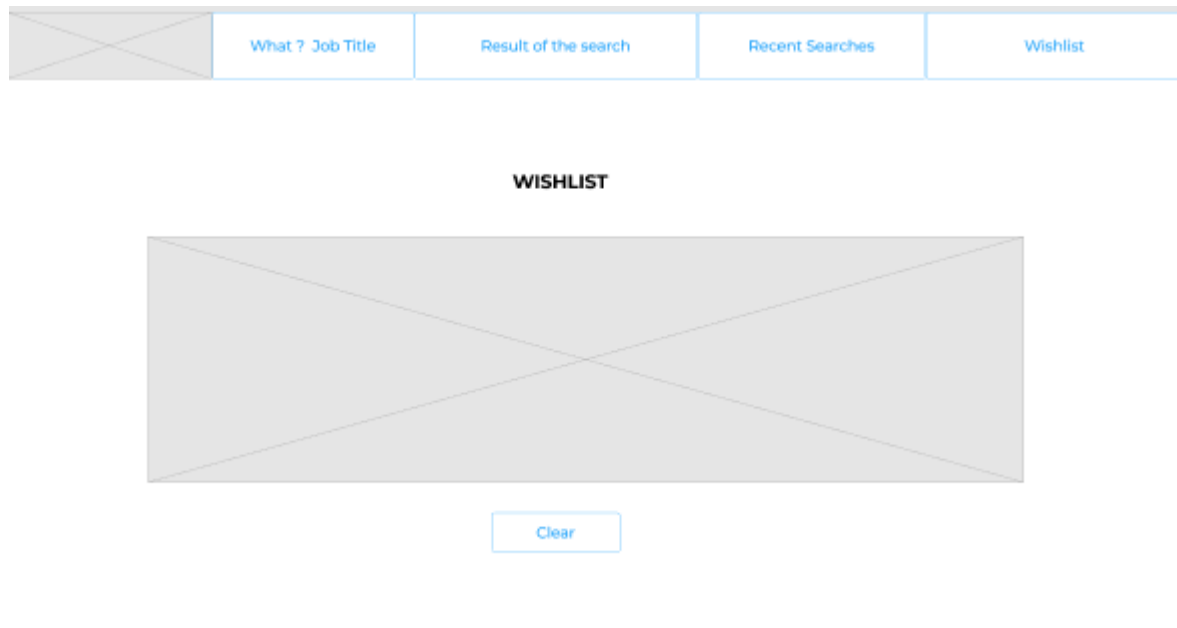
### 2.2.2 - Result of the search form:



The results wireframe that displays the job offers **includes the list of job offers containing the title, salary, location and a brief description of the job**. The wireframe also shows the predisposition of the **"Apply Now"** and **"Wish"** buttons that we talked about above.



Then let's talk about the wireframe that shows **the user search history**. It is really very simplistic and allows the user to find his way around without having to think. **Each square in the center represents a query that the user has made**, it contains the title and location of the search.



*Wishlist Wireframe Low-fidelity*

When I created the wireframe for the wishlist I already knew how the form would be laid out, I knew that **there would be a datagrid as well as a "Clear" button which would simply allow you to delete a line from the datagrid**, however I didn't know the information I was going to put inside of it.

After talking about the design we have to talk about the requirements in order to deepen the **functionalities of the software and it is the technical aspects that allowed the software to come to life as well as the design.**

---

## 3 - Requirements / System Overview:

In this section we will provide you **necessary information about the technical requirements, software features and database interactions.**

### 3.1 - Functional requirements:

The objective of this section is to show the **necessary functionalities** to accomplish the **tasks that the user wants** in order to achieve an adequate result.

### 3.1.1 - Data Source and Scraped Information:

As said above, the **purpose of the software** is to **scrape data from a website offering job offers**. This is why I decided to focus on only one site because quite simply it was more than enough given the number of job offers there were (more than 22,594). I also decided to use this website simply because it is **popular and is often used by advertisers and is therefore frequently updated**. In addition, the site offers a **lot of parameters in order to do very extensive research** and is very **customizable** for the user. This allowed me to create software that allows the user to have job offers that best meet their requirements.

Here is the site in question:

- <https://starjobsearch.co.uk/>

Of course I do not scrap from this **page but rather on this one**:

<https://starjobsearch.co.uk/jobs/job-search-results/lc-{location}/kw-{job-title}>

As you can see in the link I put **two parameters in brackets** on purpose in the query "**location**" and "**job-title**" these are **two parameters that are fundamental to simply allow me to make the request that the user wants** (standard search without advanced search).

For example here is the link that is made when I search for "**Design Engineer**" in the job title and "**London**" in the city.

<https://starjobsearch.co.uk/jobs/job-search-results/lc-London/lcr-50/kw-design-engineer>

and here are the corresponding results (one of them):

Your search returned **27** results Receive jobs like these by: [Email](#) | [RSS](#)

---

Results **1** to **10** Order list by: Relevance ▼ Results per page: 10 ▼

---

◀ Page: 1 2 3 ▶

### Senior Design Engineer

READING, OXFORDSHIRE | £42,650 TO £65,000 | PERMANENT | POSTED 10 JAN 2023

Are you an experienced Design Engineer interested in delivering the Engineering content across multiple Facilities within the Conventional and Nuclear Projects' portfolio? If so, this could be the role for you! This is a great opportunity to work with a...

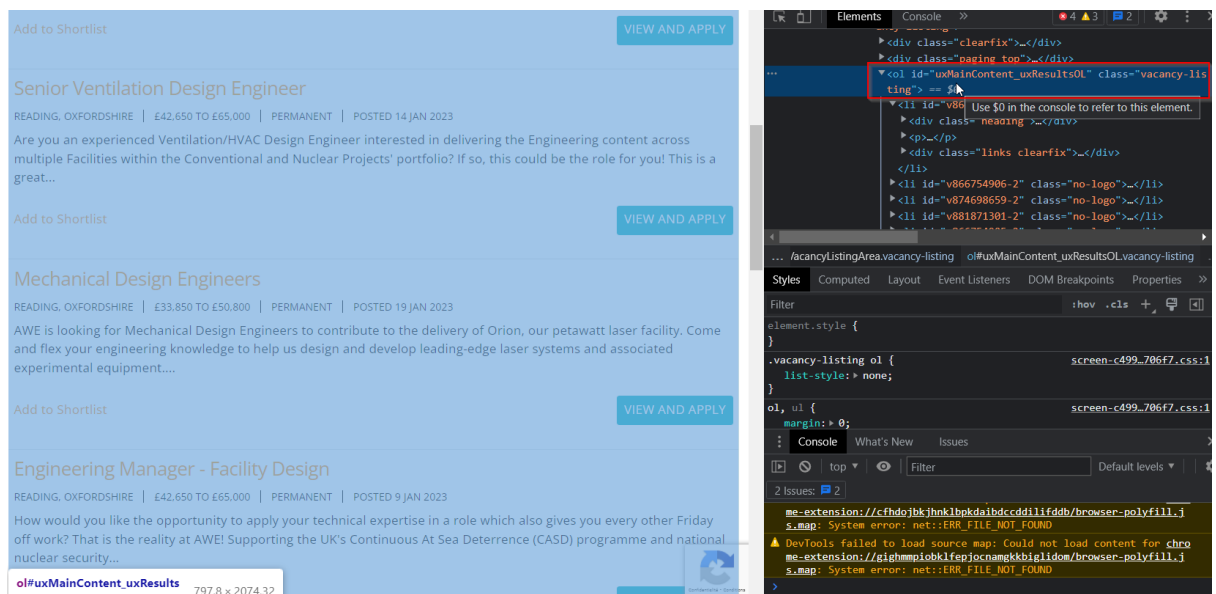
[Add to Shortlist](#)VIEW AND APPLY

#### *Result of the research*

The information to be scraped is relatively simple to perform because the **HTML code is easily exploitable** and **I thank the developers for it**.

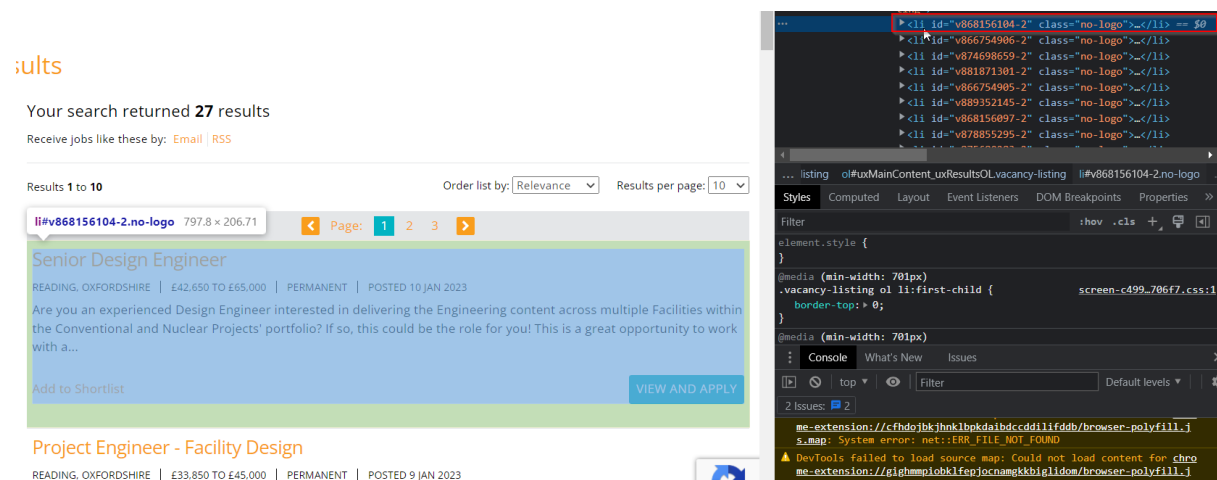


For example, to exploit the data I just had to **locate the main node which contains a collection of nodes containing the information that will be useful to me**. Here the node is an "ol" element



*Main node for the Web Scraper*

As you can see below this ol element is a "li" list which is a collection of nodes and all of it is li represents a job offer allowing me to wander around each one by retrieving the information that I need such as **title, location, salary, description, type of job** and more...

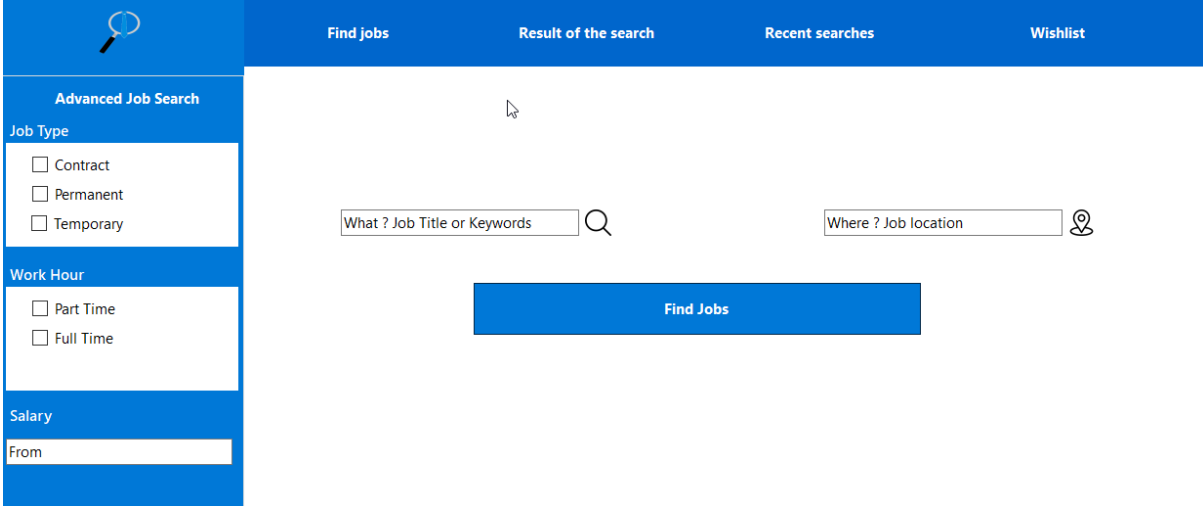


*Collection of node for the Web Scraper*

These data are then displayed to the user in an organized way and all this respecting these criteria.

### 3.1.2 - Job Search Functionality

We are now going to talk about what **happens when the user sends the parameters needed to search for a job offer.**

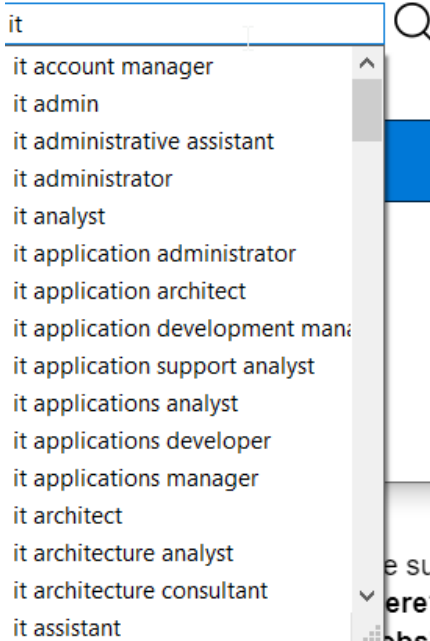


The image shows a web interface for job searching. On the left is a sidebar with filters: 'Job Type' (Contract, Permanent, Temporary), 'Work Hour' (Part Time, Full Time), and 'Salary' (From). The main area has a top navigation bar with 'Find jobs', 'Result of the search', 'Recent searches', and 'Wishlist'. Below this is a search form with two input fields: 'What ? Job Title or Keywords' and 'Where ? Job location', each with a magnifying glass icon. A blue 'Find Jobs' button is centered below the inputs.

*Job Search Form*

Let's first talk about the **essential parameters** that the user must put in order to access the results of his search.

First of all the user must make sure to put a type of job in the input "**What? Job Title or Keywords**" and a city in "**Where? Job Location**" in effect so that functionality which is located in the button "**Find Jobs**" to work the user must put at least its two parameters.



The image shows a dropdown suggestion list for the 'Job Title' input field. The input field contains the text 'it'. The list contains the following suggestions: 'it account manager', 'it admin', 'it administrative assistant', 'it administrator', 'it analyst', 'it application administrator', 'it application architect', 'it application development man', 'it application support analyst', 'it applications analyst', 'it applications developer', 'it applications manager', 'it architect', 'it architecture analyst', 'it architecture consultant', and 'it assistant'. The list is scrollable and has a blue bar on the right side.

*Suggestion List for the input "Job Title" if the user type "IT"*

When the user puts the necessary inputs, for example in the input where it is necessary to put the type of job desired, **a list of suggestions appears according to what the user types**, for example if the user writes "it" suggestions linked to it will appear. This data is retrieved from a database table "research\_information\_jobcity" and "research\_information\_jobtitle".

```
jobSearch.PutInformationCollectionJobTitle("SELECT jobtitle  
FROM research_information_jobtitle");  
customSourceJobTitle = jobSearch.jobTitle;
```

**All interactions with the database** are explained in the section "**Database Integration**"[2].



*Parameters here are "Architect and "Liverpool"*

Imagine that **the user clicks on the "Find jobs" button** with the required parameters, the software will make sure to **create an HTTP Client in order to simply behave like a client** and make the necessary request containing the parameters that the user put before.

Here is the **query made by the software**:

<https://starjobsearch.co.uk/jobs/job-search-results/lc-Liverpool/lcr-50/kw-architect/co-225/>

When the query is **sent to the web site (HTTP Request Method GET)**, the software will make sure to retrieve the job offers that there are some data **such as the location of the job, the title, the description, the salary, the type of job, the link to go to the advertiser site**. All this data is first **stored in a database table (joboffer)** then highlighted on a new form called "**Result of the search**" where the user can interact and see the job offers that correspond to him. I specify that everything that happens does not happen live in front of the user but in the background.

Here we can see the result of the query:

The screenshot shows a job search interface with a blue header bar containing a magnifying glass icon and four tabs: 'Find jobs', 'Result of the search', 'Recent searches', and 'Wishlist'. The 'Result of the search' tab is active. Below the header, there are navigation arrows and a job listing for 'SAP Technical Architect' in 'Stoke-on-trent, Staffordshire' with a salary range of '£625 to £675 Per Day'. The job description on the left mentions 'Contract (outside IR35) SAP | HANA | ECC (6.0) | ERP Day rate: Negotiable Location: Staffordshire Xpertise'. On the right, there are 'Apply Now' and 'Wish' buttons, followed by 'Job details' and 'Job Overview' sections, both containing the same job description text.

*Result of the search*

All the information you see on the screen was **also sent directly to a table (joboffer) and in the history** for the history that several information are retained such as the place of the job, the title and the advanced options.

But what happens if you do an **advanced search**? The **only thing that changes is the query, more parameters will be put inside it**, here is an example.

The screenshot shows an 'Advanced Job Search' form. On the left, there are three sections: 'Job Type' with checkboxes for 'Contract', 'Permanent' (checked), and 'Temporary'; 'Work Hour' with checkboxes for 'Part Time' and 'Full Time' (checked); and 'Salary' with a text input field containing '35000'. The main search area has two input fields: 'cyber analyst' and 'Liverpool', each followed by a magnifying glass icon. Below these fields is a large blue 'Find Jobs' button.

Here we choose to have a **job type permanent**, a **full time work hour** and a **salary start of at least 35000 £**.

Here is the request to the website :

<https://starjobsearch.co.uk/jobs/job-search-results/lc-Liverpool/lcr-50/kw-cyberanalyst/co-225/sa-35000~0~5/wh-3/jt-2/>

Since the request is **quite complicated**, we will **cut it into several pieces in order to explain it**:

- **lc** : location
- **kw**: job title
- **co**: default value
- **sa**: start salary
- **wh**: work hour option 3 (here there is two option 2: Part Time option 3: Full Time)
- **jt**: job type (1 for contract, 2 for permanent, 3 for temporary)
- **~0**: max salary (we don't use it and the default value is 0=
- **~5**: default value.

Here is the result:

The screenshot displays two job listings on a search engine. The first listing is for a 'Consumer Card Pricing and Performance Senior Analyst' in Chester, Cheshire, with a salary range of £44,000 to £69,000. The second listing is for a 'UX/UI Designer - Cyber Security' in Knutsford, Cheshire, with a salary range of £85,000 to £100,000. The second listing is highlighted with a blue border and a mouse cursor. The job details for the second listing are expanded, showing 'Job details' (£85,000 to £100,000) and 'Job Overview' (UX/UI Designer (Figma Sketch) Cheshire to #163;100k Are you a tech savvy UX/UI Designer who would enjoy being part of a hugely ambitious cyber security company where you will help deliver amazing products and services? Do you want to be part of a fantastic...).

**Consumer Card Pricing and Performance Senior Analyst**  
Chester, Cheshire  
£44,000 to £69,000

At Lloyds Banking Group we have a clear purpose; to help Britain prosper and to become the best bank for our customers. What we do makes a genuine difference to families, businesses and communities, and we're playing a central role in building a growing...

**UX/UI Designer - Cyber Security**  
Knutsford, Cheshire  
£85,000 to £100,000

UX/UI Designer (Figma Sketch) Cheshire to #163;100k Are you a tech savvy UX/UI Designer who would enjoy being part of a hugely ambitious cyber security company where you will help deliver amazing products and services? Do you want to be part of a fantastic...

**UX/UI Designer - Cyber Security**  
Knutsford, Cheshire  
£85,000 to £100,000

**Job details**  
£85,000 to £100,000

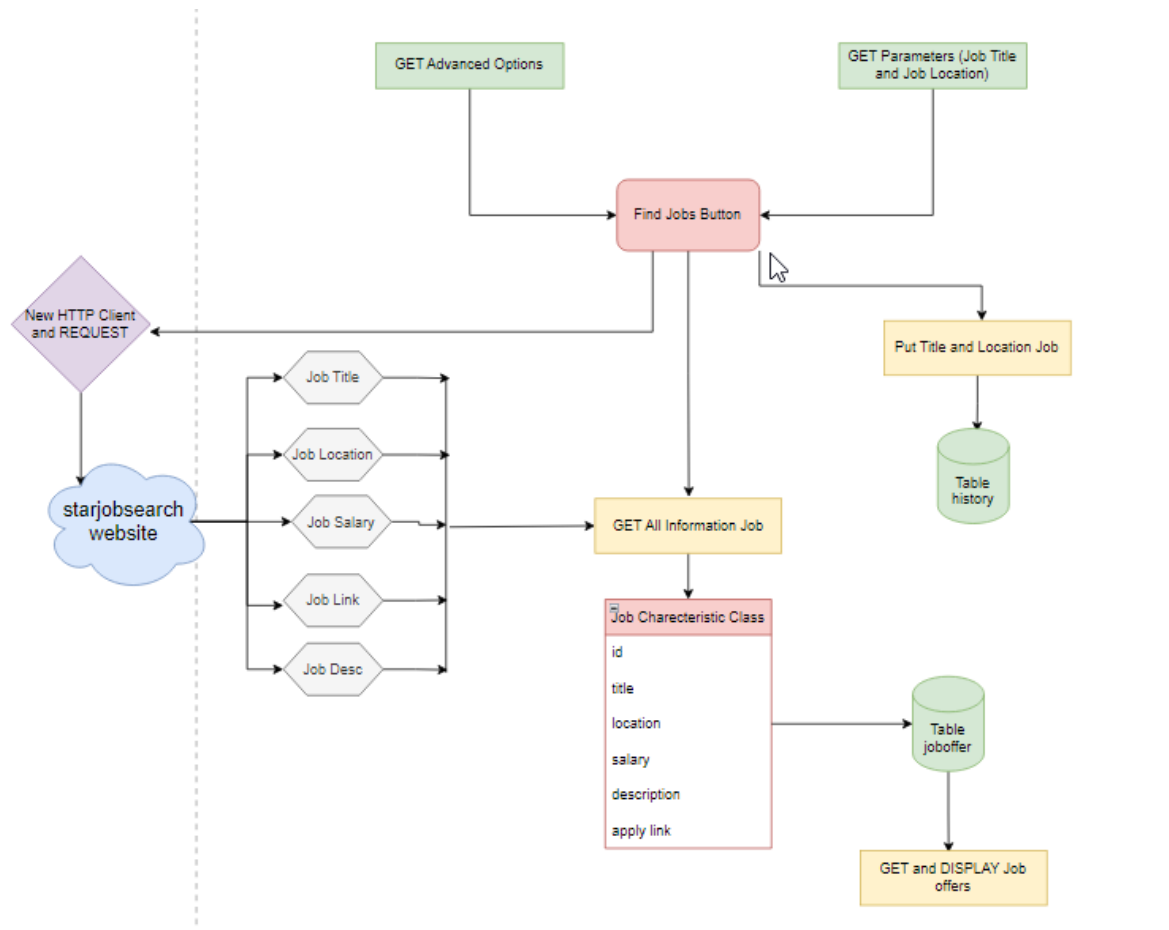
**Job Overview**  
UX/UI Designer (Figma Sketch) Cheshire to #163;100k Are you a tech savvy UX/UI Designer who would enjoy being part of a hugely ambitious cyber security company where you will help deliver amazing products and services? Do you want to be part of a fantastic...

*Result of the search*

Each profession **has a salary higher than 35000 and are full time**.

The flow chart presented below shows **what happens when the user presses the "Job Finds" button**. All steps are taken from creating the request to getting the data and displaying it.

**The query is retrieved and created according to the parameters issued by the user.** As we can see the information of the job offer is stored in a "characteristic" class through a **GetAllInformationJob** function in order to be subsequently sent to the database.



*Flow chart when the use click on the "Find Jobs" button*

We can see that the **web scraper retrieves information from the Star Job Search website which is the site that we scrap**. The information display is each job offer that contains a job title, job location, job salary, job link, and a job description.

As we can see in this flow chart all the recovered data is stored in a table named **"joboffer"** this **will allow us to display it later**. To give you an overview here is how the data is put in the joboffer table:

	id	location	title	salary	img	description	apply_link
▶	1	Chester, Cheshire	Consumer Card Pricing a...	£44,000 to £69,000	NULL	At Lloyds Banking Group ...	https://starjobsearch.co.uk/jobs/apply/ats-red...
	2	Knutsford, Cheshire	UX/UI Designer - Cyber ...	£85,000 to £100,000	NULL	UX/UI Designer (Figma S...	https://starjobsearch.co.uk/jobs/apply/?id=86...
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

*Table joboffer (temporary table)*

As you can see it is the **last research that we have done when I explained the advanced search option**. It is because **we only use a temporary table to store data to collect job offers**.

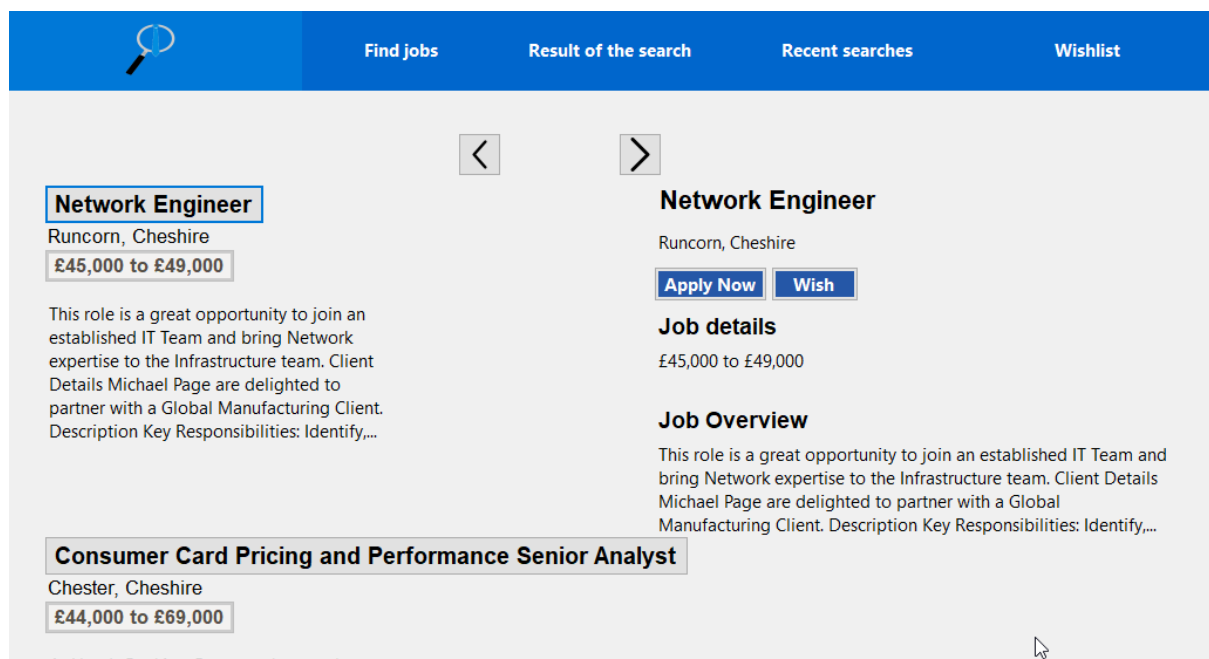
In addition we can see that the **parameters such as the job title and the job location as well as the advanced options are put in the history table**, this history will then be displayed in the form "Recent searches"

After obtaining the data, **it must be displayed** and that is why we are going to talk about the functionality that allows job offers to be displayed.

### 3.1.3 - Display Job Data Functionality

Now that we have the necessary data we must **make sure to display them in such a way that it is as readable as possible**.

As soon as the data is retrieved, it is displayed to the user. The data is presented to him in a logically organized way: **a title, a salary, a location, a description**. The user **can click on one of the titles in order to have more information on the job offer** in question. He will then have access to two new buttons (**Apply Now** and **Wish**) and a much more organized job offer.



*Image that show how is display the data in the form (Result of the search)*

The **Apply Now** button **provides access to the advertiser site in order to apply for this job offer**. For example if I click on one of the Apply Now buttons of the **Network Engineer** Job this is what happens:

### Application to Network Engineer, Runcorn, Cheshire

To register with Star Job Search and apply to the **Network Engineer** role complete the form below. All fields are required unless otherwise indicated.

First Name

Last Name

Email

Confirm Email

Password

Password must contain 8 characters or more and contain at least 1 uppercase letter, 1 lowercase letter, 1 number and 1 special character (for example: @ or !)

Attach CV

☒ Upload a CV (max size 2MB)

Aucun fichier choisi

☐ Build a CV using our CV Builder

*Apply to Network Engineer*

**This opens the default browser** and this is where I end up. I can now apply for this job offer, simple and effective.

As for the **"Wish"** button, this will simply add my job offer to my table which **contains my wishes and display all of this in the "Wishlist" form**.

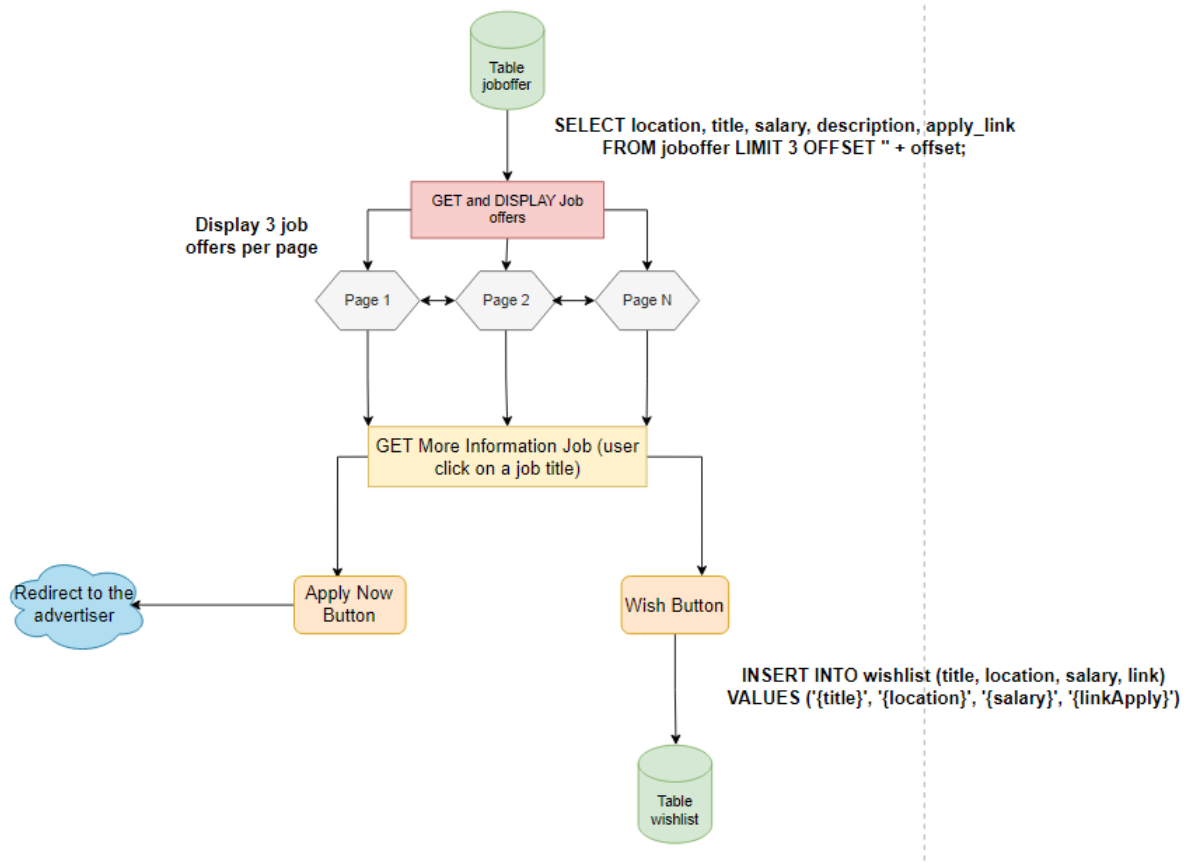
For example, if I add the **Network Engineer** job to my wish list, this is **what was added to the "wishlist" table**.

	id	title	location	salary	link
▶	6	IT Systems Support Administrator	Wrexham, Clwyd	£27,000 to £30,000	<a href="https://starjobsearch.co.uk/jobs/apply/ats-red...">https://starjobsearch.co.uk/jobs/apply/ats-red...</a>
	9	Sales Administrator	Thame, Oxfordshire	£24,000 to £28,000	<a href="https://starjobsearch.co.uk/jobs/apply/ats-red...">https://starjobsearch.co.uk/jobs/apply/ats-red...</a>
	10	Desktop Support Engineer/IT Technician	Wrexham, Clwyd	£180 to £210 Per Day	<a href="https://starjobsearch.co.uk/jobs/apply/?id=88...">https://starjobsearch.co.uk/jobs/apply/?id=88...</a>
	11	Network Engineer	Runcorn, Cheshire	£45,000 to £49,000	<a href="https://starjobsearch.co.uk/jobs/apply/?id=88...">https://starjobsearch.co.uk/jobs/apply/?id=88...</a>
*					

*Wishlist Network Engineer*



In order to **better understand the process for displaying the job offers** I made sure to make a **flowchart** allowing me to visualize the interactions with the functions and the **database** to display the data necessary for the user.



*Flow chart when the data is displayed*

The goal is to retrieve data that **has been stored previously using the GetAllInformationJob function**. This is displayed by the **ShowJobInformationData function (GET and DISPLAY job offer step)**. Each page will contain 3 job offers and the user will be able to move between these pages using the right arrow and left arrow buttons.

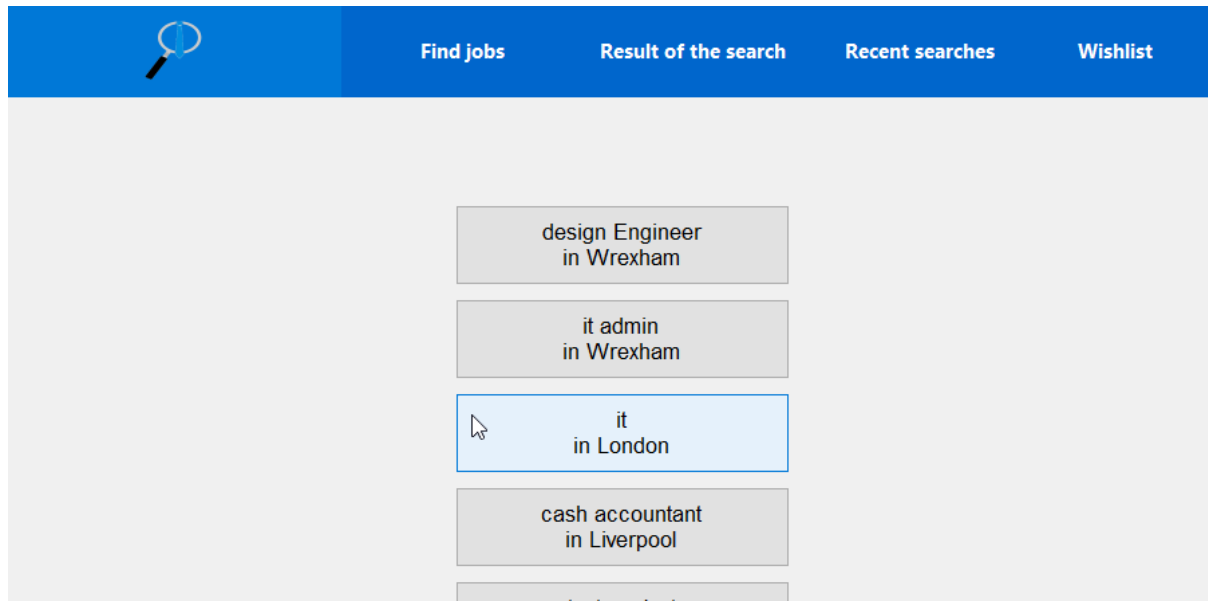
If the user **wishes to have more information on one of the jobs offered**, he just has to click on one of the titles (**ShowMoreInformationJob -> GET More Information Job (user click on a job title)**). A more organized job offer and two buttons in it will appear.

The **apply button** allows you to go to the advertiser site in order to apply and the **wish button** allows you to add the job offer to your wish list. This is then stored in the database with several information such as title, salary, location and link to apply to the job offer (PutInformationJobWish).

When the data was displayed, the **query that was passed to get all its information was stored in the history (Recent searches)**.

### 3.1.3 - Recent searches Functionality

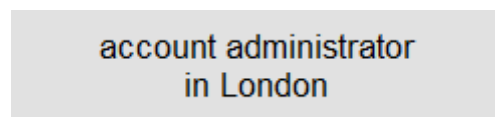
The history allows you to see the **old searches carried out by the user**.



*History Form (Recent searches)*

This feature **allows the user not to have to re-enter the data necessary to find job offers**. The jobs have been put in chronological order so that the user can better find his way around. This also allows the **user to find job offers that he could have missed and therefore to find them very easily**.

Imagine **that the user want to click on one of the buttons so that the user can remember job offers**.



*Button in the history form*

When the user clicks on one of the buttons, **it is exactly like clicking on the "Find Jobs" button, but without the user needing to set the parameters..**

Here is the result of **what happens when the button is clicked**. Here we can find some job offers (I went to the second page). I specify that when a salary button is empty it means that the advertiser has not specified a salary but I put an empty button so that the user knows that the advertiser did not wish to communicate the salary.

The screenshot displays a job search interface with two job listings. The first listing, 'Territory Account Manager - Commercial', is located in Reading, Oxfordshire, and includes a description of the role at Nutanix. The second listing, 'Production Manager', is located in Henley-on-thames, Oxfordshire, and includes a description of the role at a creative agency. Both listings have a salary range of £30,000 to £40,000. The 'Production Manager' listing also features 'Apply Now' and 'Wish' buttons.

**Territory Account Manager - Commercial**  
Reading, Oxfordshire

Nutanix is looking for a Territory Account Manager for the Commercial segment in the London region in the United Kingdom. As a Territory Account Manager, you are responsible for selling Nutanix's solutions to customers and partnering with our extensive...

**Production Manager**  
Henley-on-thames, Oxfordshire  
£30,000 to £40,000

I am currently recruiting for an Events/Experiential Production Manager to join a growing creative agency based near Twyford. Position overview: Manage, from concept to completion, the production, logistics and installation of projects delivered by the...

**Production Manager**  
Henley-on-thames, Oxfordshire  
£30,000 to £40,000

**Job details**  
£30,000 to £40,000

**Job Overview**  
I am currently recruiting for an Events/Experiential Production Manager to join a growing creative agency based near Twyford. Position overview: Manage, from concept to completion, the production, logistics and installation of projects delivered by the...

*Result of the search*

In order to **better understand the functionality** I made a **flow chart**. This flowchart **aims to show the functionality when clicking on one of the history buttons**.



	id	title_job	job_localization_id
▶	1	design Engineer	6285
	2	Design Engineer	6285
	3	it admin	6285
	4	it	1530
	5	cash accountant	5707
	6	test analyst	5347
	7	admin assistant	1530
	8	account administrator	1530
	9	food analyst	1530
	10	food and beverage analyst	1530

*jobhistory table*

Here is the data that we retrieve each time in the history table, the **job\_localization\_id** is a **Foreign Key** that is **linked to another table called "research\_information\_jobcity"**. I wanted to complicate things in order to give myself a certain challenge by handling foreign keys. Here is what I do to find the name of the city linked to the id location\_id.

```
SELECT DISTINCT title_job, city FROM joboffer.job_titles_history JOIN
joboffer.research_information_jobcity ON
joboffer.job_titles_history.job_localization_id =
joboffer.research_information_jobcity.id;
```

I select the id (**job\_localisation\_id**) and link it to the **city id** coming from the **"research\_information\_jobcity"** table If the id is the same I **select the city coming from the "research\_information\_jobcity" table which has the same id** (it's like a WHERE id = \$id)

To give you an idea of the research\_information\_jobcity table, here it is:

	id	city
▶	1	Kabul
	2	Luanda
	3	The Valley
	4	Abu Dhabi
	5	Ajman
	6	Al Ain
	7	Dubai
	8	Ras Al-Khaimah
	9	Salta
	10	Avellaneda
	11	Bahia Blanca

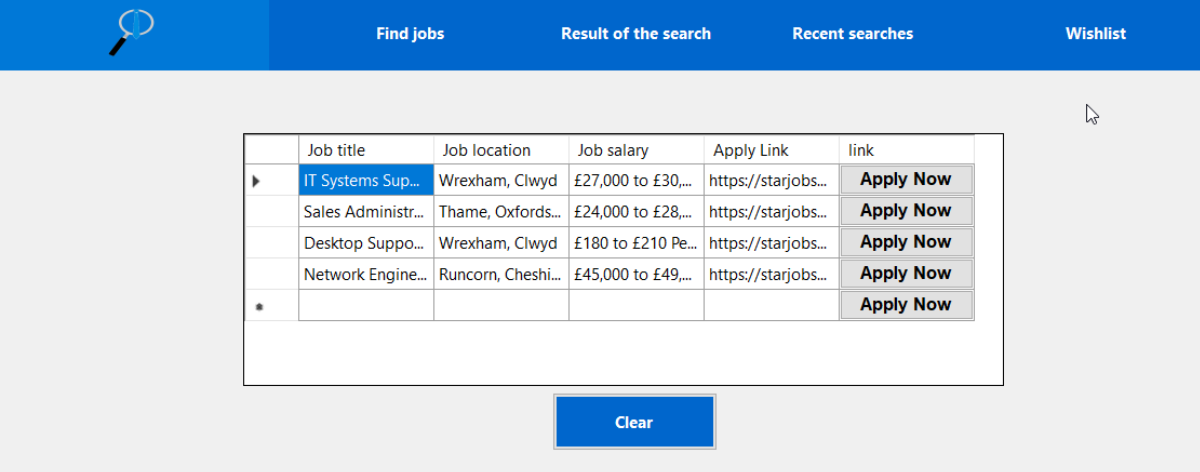
*research\_information\_jobcity table*

There are more than 70000 records inside, it is the **same table used for the suggestions when the user types the parameters** necessary for the job offer search.

Moving on, let's discuss the **wishlist** feature of our job searching software.

### 3.1.4 - Wishlist Functionality

This feature allows the user to **bookmark or save the job offer that interests him**, this allows him to somehow sort out what he likes and what he does not like.

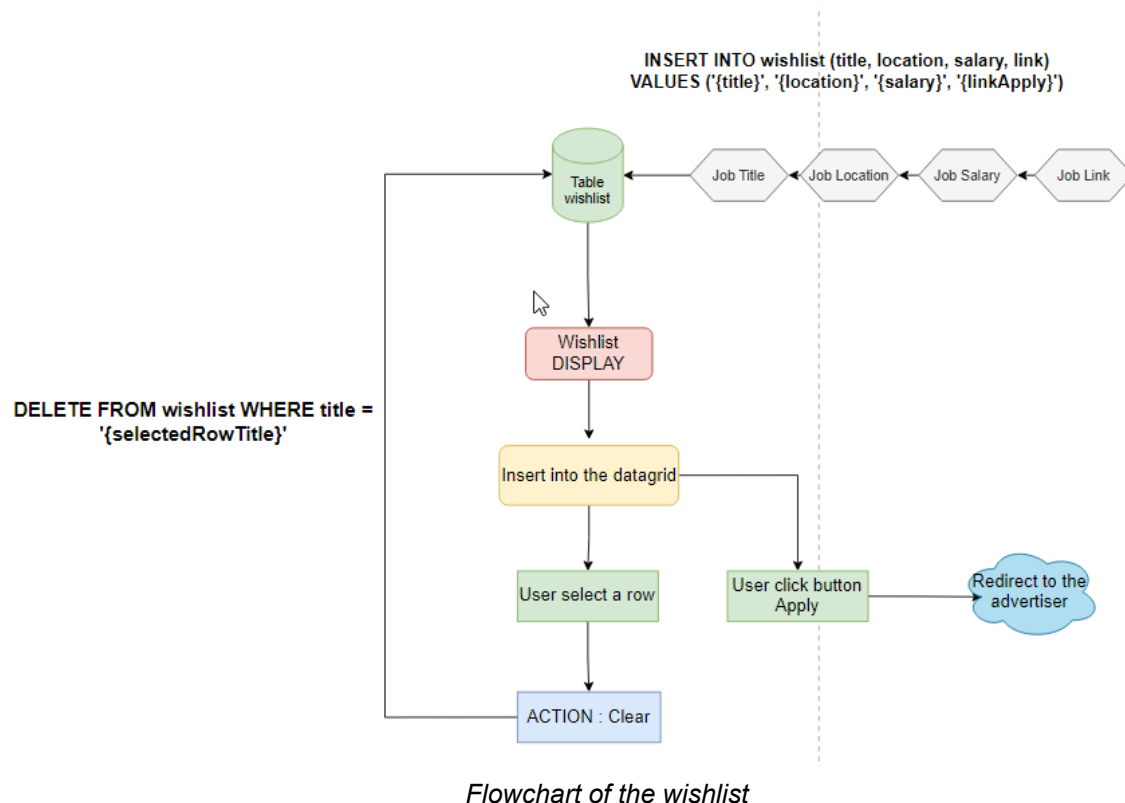


	Job title	Job location	Job salary	Apply Link	link
▶	IT Systems Sup...	Wrexham, Clwyd	£27,000 to £30,...	https://starjobs...	<a href="#">Apply Now</a>
	Sales Administr...	Thame, Oxfords...	£24,000 to £28,...	https://starjobs...	<a href="#">Apply Now</a>
	Desktop Suppo...	Wrexham, Clwyd	£180 to £210 Pe...	https://starjobs...	<a href="#">Apply Now</a>
	Network Engine...	Runcorn, Cheshi...	£45,000 to £49,...	https://starjobs...	<a href="#">Apply Now</a>
•					<a href="#">Apply Now</a>

[Clear](#)

*Wishlist Form*

Each time the user **was able to click on the "Wish" button, the job offer was inserted into this list**. This is displayed using a **datagrid** which allows the user to manage what is inside, for example if one of the wishes **no longer interests him then he can select a row and delete it with the "Clear" button**. How it can be seen several pieces of information are displayed such as the **job title, salary, location and the link to go to the advertiser site**. In order to better understand how this feature works, a **flowchart** that you will find below has been put. This explains **how this feature works and how the user can interact with the wishlist**.



Here is a **flowchart showing how the wishlist works**. First of all, in order to retrieve the user's wishes, they are **extracted from the table (wishlist)** where the wishes are saved. Data is **injected into the datagrid** using a `getAllInformationWishListDatabase` function.

If a **row is selected** at the same time and then the **clear button** is executed then a **DELETE** order to the table wishlist is launched in order to remove the wish in question. The user can also very well press one of the **"Apply Now"** buttons which will allow him to redirect to the page of the advertiser of the job offer.

	id	title	location	salary	link
▶	6	IT Systems Support Administrator	Wrexham, Clwyd	£27,000 to £30,000	<a href="https://starjobsearch.co.uk/jobs/apply/ats-red...">https://starjobsearch.co.uk/jobs/apply/ats-red...</a>
	9	Sales Administrator	Thame, Oxfordshire	£24,000 to £28,000	<a href="https://starjobsearch.co.uk/jobs/apply/ats-red...">https://starjobsearch.co.uk/jobs/apply/ats-red...</a>
	10	Desktop Support Engineer/IT Technician	Wrexham, Clwyd	£180 to £210 Per Day	<a href="https://starjobsearch.co.uk/jobs/apply/?id=88...">https://starjobsearch.co.uk/jobs/apply/?id=88...</a>
	11	Network Engineer	Runcorn, Cheshire	£45,000 to £49,000	<a href="https://starjobsearch.co.uk/jobs/apply/?id=88...">https://starjobsearch.co.uk/jobs/apply/?id=88...</a>
*	NULL	NULL	NULL	NULL	NULL

Table wishlist

### 3.2 - Technical detail:

In this section we will talk about the **technologies** used and the **library** that has been used to run this software.

### 3.2.1 - Technologies used:

Several technologies have been used to maintain and create the software. The programming language **used is C#** which allows the development of software on Windows. As for the graphic side, we used the **NET Framework which allows us to make Windows Applications**.

### 3.2.2 - Libraries used:

In order for the web scraper to work, we needed to **use two libraries**, one of which allows **data to be scrapped, which is called HTML Agility Pack**, allowing the extraction of **data and resources from an HTML page coming from a website**, then subsequently the data could be stored using another library which **allows the join between the software and the MySQL Connector database**.

### 3.2.3 - Web scraper interaction:

The **web scraper is an essential component of the software** that allows the extraction of data from a website, all using the **HTML Agility Pack library**. Only the necessary information coming from the main node which contains a collection of nodes which each have job offers are recovered. The information is then sorted and only the title, salary, location, description and link of the offer are retained. **Each offer is then put in a temporary table in the database**.

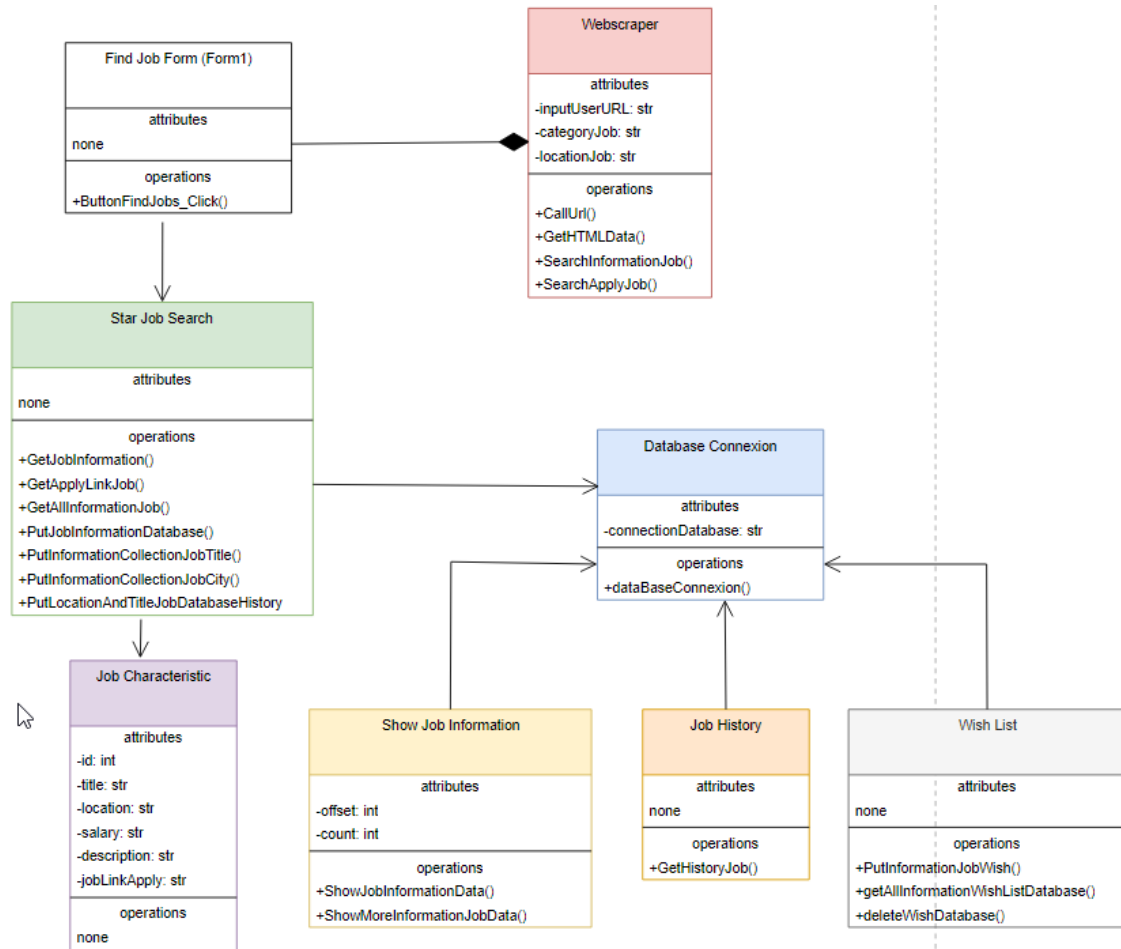
### 3.2.4 - Data Extraction:

When we receive the data from the website it is **only a piece of code of the whole page** and the goal and to sort all this thanks to the HTML Agility Pack component called **XPath which allows you to walk around one thing leading to another in the HTML code**. When the necessary data has been extracted, it is put in a table.



### 3.2.5 - UML Class:

The following section details the **class structure and relationships of the web scraper through the use of UML diagrams.**



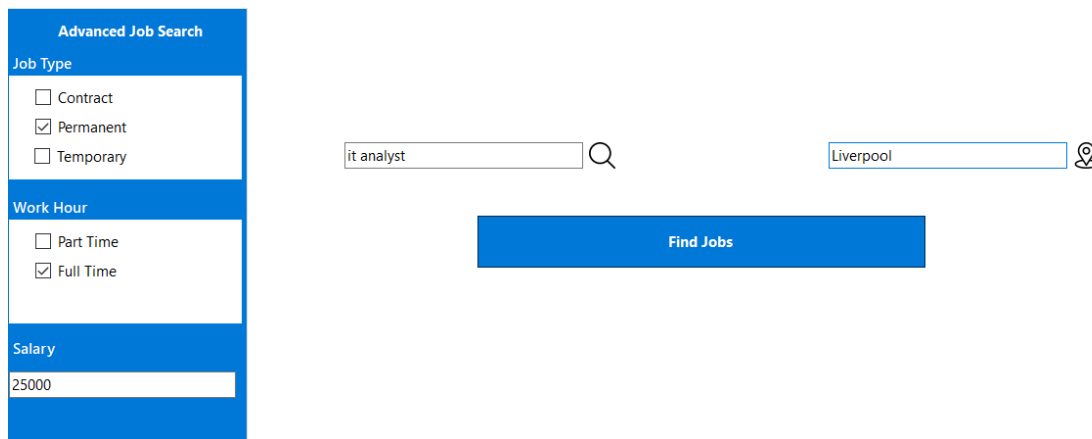
*UML Class of the software*

## 4 - Software Testing:

This section aims to see **if the software meets the requirements and functions as intended**.

### 4.1 - Web Scraper Testing:

This section aims to **compare the results of the website to those of the web scraper**, indeed the data must match in order to have the same result as on the site using of course the **same parameters**.



The image shows a job search interface. On the left is a sidebar titled 'Advanced Job Search' with three sections: 'Job Type' (with checkboxes for Contract, Permanent (checked), and Temporary), 'Work Hour' (with checkboxes for Part Time and Full Time (checked)), and 'Salary' (with a text input containing '25000'). To the right of the sidebar are two search input fields: 'it analyst' with a magnifying glass icon, and 'Liverpool' with a location pin icon. Below these fields is a blue button labeled 'Find Jobs'.

*Test Parameters*

Let's imagine that these parameters have been set, here is their list:

- **Job Type:** Permanent
- **Work Hour:** Full Time
- **Salary start:** 25 000 £
- **Job title:** IT Analyst
- **Location:** Liverpool

Here are the result:

<

>

**Consumer Card Pricing and Performance Senior Analyst**

Chester, Cheshire

£44,000 to £69,000

At Lloyds Banking Group we have a clear purpose; to help Britain prosper and to become the best bank for our customers. What we do makes a genuine difference to families, businesses and communities, and we're playing a central role in building a growing...

**Development Manager (Grants, Funding, Partnership Development)**

Widnes, Cheshire

£35,000 to £40,000

Do you have the passion and skills to help

*Result of the search*

If we compare with the website by putting the results, we can see that the site has the same results as us as seen below.

**Refine Search**

IT Analyst

United Kingdom

Liverpool

50 miles

**Sector**

☐ Accounting

☐ Administration & Secretarial

☐ Advert / Media / Entertainment

☐ Agricultural

**Job Type**

☐ Contract

☒ Permanent

☐ Temporary

**Work Hours**

☐ Part Time

☒ Full Time

**Salary**

25000

To

Your search returned **23** results

Receive jobs like these by: [Email](#) | [RSS](#)

Results 1 to 10


Order list by: [Relevance](#)

Results per page: [10](#)

Page: [1](#) [2](#) [3](#)

**Consumer Card Pricing and Performance Senior Analyst**

CHESTER, CHESHIRE | £44,000 TO £69,000 | PERMANENT | POSTED 16 JAN 2023




At Lloyds Banking Group we have a clear purpose; to help Britain prosper and to become the best bank for our customers. What we do makes a genuine difference to families, businesses and communities, and we're playing a central role in building a growing...

[Add to Shortlist](#)

[VIEW AND APPLY](#)

**Development Manager (Grants, Funding, Partnership Development)**

WIDNES, CHESHIRE | £35,000 TO £40,000 | PERMANENT | POSTED 22 JAN 2023



Do you have the passion and skills to help one of Britain's biggest charities to secure new partnerships and funding for programmes that can change lives at scale? Community Integrated Care is seeking a Development Manager to help grow our...

[Add to Shortlist](#)

[VIEW AND APPLY](#)

**Care Coordinator**

WIDNES, CHESHIRE | £25,000 | PERMANENT | POSTED 24 JAN 2023

*Result on the website*

As we can see, the results match and we can conclude that the web scraper is functioning correctly and accurately extracting the desired information from the website.

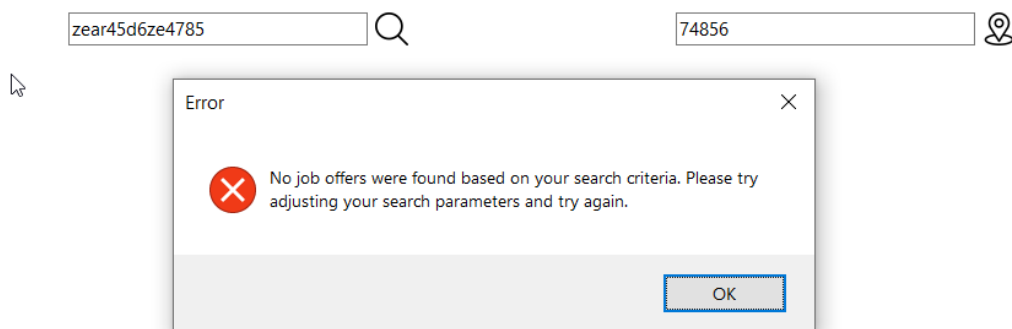
## 4.2 - User Input Testing:

The purpose of this section will allow us to **know if all user input is processed by the software whether it is an error or there are results.**

### 4.2.1 - Random User Input Testing:

In this scenario we will make sure to **put completely random parameters** such as "rezrzerzerze" for the title of the job and "ezrrrezrez" for the location of the job.

**If we do this an error will appear, we say that the web scraper did not find a result.** It's simply because the site has no results for this search.



*Result of the Random User Input Testing*

### 4.2.2 - Salary, Job Type, Work hour Input Testing:

This scenario **has already been done indirectly in the Web Scraper Testing section**, we were able to compare the result with the software and the site and everything was fine.

### 4.3 - Performance Software:

In this section, we will **evaluate the performance of the software** by testing the **maximum number of requests** it can handle and the **speed at which the requests are processed**. This will allow you to know if the software or website supports a lot of requests and if it is fast, all this to prevent the user from being faced with software that is too slow.

#### 4.3.1 - Request Testing:

In order to know if the software supports requests well, **I made sure to do tests whose goal is to make a multitude of requests here 20 in a row for at least 1 minute**. Everything went well, all the requests could be processed and the results were consistent with the parameters issued. The only problem that was encountered was coming from the site **which thought that I was a robot and had to slow down some of my requests** but I cannot do much about it unfortunately.

**The time of a request is not even 2 seconds** the time that the software receives the information from the website which is a normal time and **it also depends on the connection of the user** but it will not exceed never longer than 5 seconds.

## 5 - Conclusion:

This software therefore makes it possible to **retrieve job offers and display them to the user according to the criteria desired by the user**. The design of this software is done in a way to be user friendly. Several technologies like **HTML Agility Pack** and **C#** are used to bring the software to life.

Finally, this software will be accessible to everyone and especially to those **who wish to search for job offers in a simple and efficient way while giving reliable results**.