**18:00, Friday 24 Mar 2023 code**

**18:00, Friday 21 Apr 2023 Final Report & Screen cast**

**Automatically accessibility evaluation and analysis of NHS General Practices’ websites**

The World Wide Web has improved the lives of many. People with disabilities belong to the users that can benefit most from it. However, to a large extent, the Web is not accessible. The W3C consortium has proposed a set of guidelines that web pages should satisfy to make the websites accessible, i.e. the Web Content Accessibility Guidelines.

While access to health should be universal, if websites containing critical information about the services offered by the National Health Service are not accessible, many citizens get excluded from it.

The purpose of this project is to establish a software system to automatically evaluate the websites of General Practices (primary care). The following activities should be conducted to complete the project successfully.

- Analysis of existing automatic evaluation tools to choose those that can be embedded in a software system.

- Use a crawler to collect the URLs and information on the General Practices

- Develop a system that evaluates the accessibility of those websites and stores the results in a regular fashion

- A dashboard that monitors the accessibility of websites over time

- Statistical analysis between compliance to accessibility guidelines, (i) general satisfaction with the GP and (ii) use of online services by the people registered in that practice

**Week1 - 2** (Sep 25 - Oct 8)

Plan:

documentation(what have done, reflection on planning, performance)

1. Devise a detailed plan, Gantt chart (1w)
2. Accessibility guidelines (4w)
   1. understand each criterion

4 main chapters of the guidelines:

Chap 1 for week 1

Chap 2 for week 2

Chap 3 & 4 for week 3

* 1. decide which criteria I may use to evaluate NHS website

1. Papers on accessibility evaluation (3w)
   1. do some research on previous papers about the website accessibility evaluation

- what methods/tools they used

- what data of the websites they collected

- what guidelines they used

- how they analyzed the data

1. Existing evaluation tools (4w)
   1. find some tools and check what accessibility criteria they can evaluate
   2. try to use some tools to evaluate the web page -> understand how to **embed** them into my system
   3. compare different -> choose some that can be used in my own work
   4. compare different -> quality / coverage / accuracy ->
   5. check agreement among different tools -> more reliable
   6. remember to credit the recourse
2. Crawler (4w) github
   1. learn to build a crawler to collect urls and info

- collect data

- store the data in the database (mysql?)

- sampling methodology

- use evaluation tools to test the stored web pages

1. Statistical analysis (Christmas, 2w)
   1. analyse the collected data
   2. evaluate the result of the tools

maybe need some manual checks? Evaluate manually

* 1. visualize the data (heat map, map etc.)
  2. think of improvements:

- use mathematical techniques/models

- machine learning models -> predictions?

* 1. evaluate the system: what guidelines are applied, what data is analyzed (strengths and weaknesses)

(2w exam period)

1. Dashboard (6w)
   1. learn to build a dashboard to display the result (both frontend and backend)
   2. automatically update

Q:

1. Deadlines?
2. Writing a pipeline: use existing tool and do some analysis -> there are not many things for me to do?

If I meet all the requirements mentioned in the project description, what grade can I expect to get?

To achieve a higher grade, what improvements can I make?

Improvement?

1. Make the dashboard look fancier? Add more function to it?
2. Write my own func to evaluate the accessibility? Do not use existing tools (or partially use the tool/api)
3. Use more techniques in data analysis? Eg. Machine learning?

Report

Intro

Describe the problem

Why it’s important

What others done

There’s a gap

Summary of what you did

Background

Expand most relevant (others work)

A list of gaps / improvements

Inspiration

Methodology

Experiment / pipeline

Documentation

Testing method

Architecture (uml, database, class design etc.) more conceptual diagram

Result & evaluation

Description artifacts (more detailed)

Stats if there are experiments

Discussion & conclusion

Discuss findings

Questions unanswered

**Week 3** (Oct 9 - Oct 15)

Web Content Accessibility Guidelines (WCAG) 2.1:

<https://www.w3.org/TR/WCAG21/>

- make Web content more accessible to people with disabilities.

- more usable by older individuals with changing abilities due to aging (improve usability for users in general).

Involves a wide range of disabilities: visual, auditory, physical, speech, cognitive, language, learning, and neurological disabilities.

Conformance level

<https://www.w3.org/TR/WCAG21/#conformance>

A: minimum level of conformance, fundamental and the Web page should try to satisfy all the Level A Success Criteria

AA: web page satisfies all the Level A and Level AA Success Criteria

AAA: web page satisfies all the Level A, AA and AAA Success Criteria

Use as Qualitative Metrics

Criteria (quick ref)

<https://www.w3.org/WAI/WCAG21/quickref/?currentsidebar=%23col_overview>

Perceivable Operable Understandable Robust

**Relevant criteria** which can be applied to test the accessibility of NHS: (perceivable)

**Text alternatives**

Graphs on the web page -> should have text alt

So that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language

**Adaptable**

Content which can be displayed in different ways, eg. Simpler layout without losing info/structure

When using a screen reader, the sequence of the information remains the same

For shapes on the web page, should not rely on the color/shape/location of the icons -> provide additional info in case people cannot perceive the shape

**Distinguishable**

Use different colors to indicate different actions/responses/elements (if cannot perceive the color, use text etc)

Text has a high contrast ratio, whether distinguishable from the background

Text can be resized without loss of content/functionality

**Essays**

*Benchmarking Web Accessibility Evaluation Tools: Measuring the Harm of Sole Reliance on Automated Tests*

Problems with automated tools:

Only report negative/positive result, may not include interpretation of the guidelines and the severe impact on users

May have narrow coverage, low completeness and accuracy

Need to evaluate the result of the evaluation tools

May require evaluate manually: sampling (eg. Select pages which include various content and functionality) -> find the tags/code which cause the accessibility barrier -> match against the report

Report of the tools may be in different forms (eg. Web report, XML files, emails)

Combine all kinds of report and store them in a common uniform report

*Tools for Web Accessibility Evaluation*

Apply several different automated tools, compare the results -> match them with the results I evaluated manually -> select the suitable tool according to each principle: perceivable, operable, understandable, robust (eg. Tool A has higher accuracy in evaluating perceivable guidelines, tool B better at operable part -> combine the results from different tools, generate a final report with higher accuracy)

(Crowdsourcing: users and volunteers are able to evaluate and improve the accessibility collaboratively, tho not viable in personal project)

**D3 js / echarts**

scrapy

Minimal viable product: crawler + online tools

**Week 4** (Oct 16 - Oct 22)

Found some existing tools and tried to use the tools to evaluate the NHS website,

Most are online tools, some are extensions

Various results for the same web page

**Siteimprove**, extension, has the most detailed result:

Display with different conformance level, categorize the severity as error, warning, review (need manual check)

For each issue, provide cause, related code -> point out which criterion it violates

This one reports the most issues among all the tools

**Wave**, online checker, API

Locate the code that violates the guidelines

**Google lighthouse** & **axe dev tool**, just found very few violations

The check maybe not that exhausted

Online: **Accessi.org** & **Achecker** & **TAW**

Started to learn to build a scrapy project

Planning to scrape the urls

Problem:

All clinic links & rating links -> page not found

Get the name of the clinic and search it using a search engine -> find its nhs page and go to its website (home page)

Check some API

Pipeline -> one tool -> more tools

Evaluate the results (manually check 1/2 pages, confusion matrix)

**Week 4** (Oct 23 - Oct 29)

防止这个网址出问题，先爬下所有医院名称，保存备份

Still learning scrapy

Scraped the name of the GP on nhs website -> store into data frame, including the name & location

Next try to find url of each GP -> search (name + nhs overview) in google -> make sure the first link is the surgery overview on nhs website -> (maybe some regex: check if the link contains ‘nhs.uk’) -> collect the official link

Google api

Some websites have an accessibility menu, can customize the website display, eg. Change contrast, text spacing, cursor...

Web can have **higher accessibility**

Powered by userway.org -> provide the same functionality for different websites

-> keep a record of the website which have this menu

Evaluate the extra accessibility of the menu

Eg. If a website has contrast issue but provides this menu -> it can resolve the contrast issue

Num of surgeries?

Fake user agent -> csv -> data frame -> search api

Google being duckduckgo

**Week 5** (Oct 30 - Nov 5)

403 access denied when scrape too frequently

Can re-access in 10 min

Fake user headers + rotate proxies?

Module: fake\_useragent

Deal with 403: wait for 10 min and retry

Remove duplicate: (Some duplicate gp names)

- in different location (have different postcode)

- Or they are actually the same one but can be accessed in several different towns due to the short distance between the gp and the towns (people in different towns can go the the same gp)

-> drop all duplicate names: gp with the same name usually has the same website

-> check the nhs overview: get gp’s link and their official location (town + postcode)