SORBONNE UNIVERSITÉ
CRÉATEURS DE FUTURS DEPUIS 1257

Assignment 3 – webapp. Book search engine.

BM Bui-Xuan

Web/mobile application : A web/mobile application is different from a web page by their user interaction : a page is reduced to be uniquely static. User interaction can be explicit : the application changes in response to an action that the user has just performed. In the case of a side effect, or a consequence of older user actions, the interaction is called an implicit UI.

Search engine for a library : In this assignment, we refer to library any database containing sufficiently many text documents. One such example is given at The Gutenberg Project, where documents are stored in various formats, including ASCII text format. Much as with The Gutenberg Project's database, a library can store tens of thousands of text documents. It is humanly challenging to manually search for the right document when answering to questions such as *What are all the books telling the story of King Sargon ? Has King Sargon ever came to Saigon harbor ? As a tourist, what can we expect in actual Saigon district ?*, and so on.

A search engine in such a library is a web/mobile application whose primary feature is to point its users to the right document, according to a search by keyword. Another feature could be to point the users to a recommended document following their search history.

Content analysis and performance analysis : The content relevance of a search engine is a subjective score resulting from a user test campaign. It could be simplified as rule #12 of The Joel Test. Contrasting to this human biased measure, performance ratings of a search engine can be determined by stress tests and statistics over increasingly large datasets.

1 Statement of assignment 3 – webapp

The project consists in developing a web/mobile application offering a search engine for a certain library of text documents. The first requirement is about the data layer : one need to collect sufficiently many text documents, either by hard-storage on disk, or pointers to contents on distant servers such as at The Gutenberg Project's. The minimum size of the library must be 1664 books. The minimum size of each book must be 10^4 (ten to the four ; ten thousand) words.

Next, it is required to build the server logic and the client views offering the main features of a search engine. Here, each project team can determine its own userstories, however, they must include :

- **Explicit feature "Search" :** Search documents by keyword. On user input a string S, the application returns a list of text documents whose index table contains S.
- **Explicit feature "Advanced search" :** Search documents by RegEx. On user input a string $RegEx$, the application returns : either a list of text documents whose index table contains a string S matching $RegEx$ as regular expression (refer to Lecture 1 of UE DAAR for a formal definition of regular expressions) ; or a list of text documents containing a string S matching $RegEx$ as regular expression (Warning : this option may cause the application to slow down considerably).
- **Implicit feature of ranking :** Ordering the presentation of the documents returned by above features. In response to a search or an advanced search, the web/mobile application returns the list of documents ordered by relevance, according to some mathematical definition of ranking : by decreasing number of occurrences of the keyword/regEx in the document, by decreasing centrality ranking of Jaccard graph (refer to the ending slides of Lecture 5 for a formal definition of centrality ranking and the Jaccard distance/graph). It is required to implement at least one of the following centrality rankings : *closeness*, *betweenness*, or *pagerank*. It is also required in the final report of the

project to : recall the definition of the centrality ranking measure in use ; the implemented algorithm computing this measure ; as well as examples of the computation on well chosen samples of the > 1664 books in your project team's database.

- **Implicit feature of recommendation :** Suggestion of documents with a content similar to the last search. Along with the response to a search request, the web/mobile application also returns : either a list of documents which are vertices of the Jaccard graph (cf. Lecture 5) in the neighbourhood of the highest ranked documents matching the search request (according to the above feature of ranking) ; or a list of most "clicked" documents by other users when doing a similar search request.

The project is essentially the construction of a web/mobile application. It is important to include a demo version during the oral presentation or the video pitch of the project (see description of oral presentation and video pitch below). It is also very important that the application can be run on various client machines during the demo (laptops, smartphones with iOS and/or Android, etc).

As for evaluation, most important aspects are : presentation of the tests (relevance test and performance test) ; the composition quality of the report ; as well as the oral presentation or the video pitch of the project. There will be 3 assessment related to the project : one for the code ($\approx 30\%$ project assessment) ; one for the report ($\approx 40\%$) ; and one for the oral presentation ($\approx 30\%$).

Report composition : Each project team should focus on the following aspects, for any implemented algorithm (including those already presented in the Lectures such as Aho-Ullman, KMP, radix tree, Jaccard distance, closeness, betweenness, pagerank) :

- problem definition and the data structure in use.
- analysis and theoretical presentation of known algorithms from literature ("state of the art" presentation).
- factual and relevant arguments backing any choice, amelioration, and critics relative to these algorithms.
- testing : methodology for obtaining every testing dataset. In particular, all references from other works must be properly cited (any form of plagiarism is prohibited).
- performance tests : please, favor bar charts, frequency diagrams, and other types of dashboard-like charts, to plain, raw, columns of figures.
- a discussion on the results of the performance tests would be much appreciated.
- if applicable, please include user test and relevance test methodologies.
- close the report with a conclusion and perspectives of search engines for textual documents.

A typical report will include between 10 and 15 pages. The recommended number is 12. The higher limit is mandatory : pages 16+ will not be read by the examiner.

Oral presentation 20min or video pitch 5min, with demo : Each project team has the choice to either perform an on site oral presentation of the project or instead send a video pitching the project.

Oral presentation : The upper limit of duration for the oral presentation is 20 minutes. The following points are mandatory :

- Introduction ($\approx$ 7 minutes) : team members ; project goals ; functional analysis (use cases and/or user stories) ; some first exemplified wireframes (some recommended tools are balsamiq, framerX, sketch) ; list of software technologies in use (along with a comparison chart with concurrent technologies and arguing the project team's choices) ; possibly with a HR presentation (Gantt chart or Scrum backlog or just a basic table of man days per principal steps of the project).
- Technical part ($\approx$ 10 minutes) : general architecture of the web/mobile application ; presentation of the client layer (principal views, actual graphical rendering compared to the planned sketch, etc) ; presentation of the data layer (database technology in use, index tables conception, Jaccard graph storage, etc) ; presentation of the server logic layer (algorithms).

- Demo and conclusion part ($\approx$ 3 minutes) : a demo is mandatory where the web/mobile application must run on at least 2 distinct client machines connected to the same LAN/WLAN (provided by each project's team). However, it would be better to have 3 machines for the demo : one server machine and two client machines. The difference between 2 and 3 machines is that the client machines will not display from localhost.

It is recommended to have between 18 and 22 slides for a 20 minutes talk.

Video pitch : Instead of presenting the project on campus as described above, each project team has the choice to send a video pitching the project instead. This video is a highlight video, and must last for roughly 5 minutes (+/- 10%, no more, no less). The video should resume the essential points of an oral presentation described above. The video must include a sequence of demo of the project. There will be a penalty applied to every project team choosing not to present the project on campus but the video pitch does not include a sequence of solid demonstration of the project.

Constraints :

- Project team of 2 or 3 members.
- Compress the final package in one unique file, including : documentation (report $\approx$ 10-15 pages); commented code source; the video pitch if applicable; as well as all necessary materials for the examiner. However, the final compressed file must not exceed 30-ish Mo.
- Please use the following naming format for the compressed file : daar-assignment3-SURNAME1-SURNAME2-SURNAME3.piki, where piki could belong to $\{tgz, zip, rar, 7z, etc\}$. This naming convention is important for an automatic classification of your files on the PC of the poor person who will have to assess 70+ student projects. (There should be a penalty applied to every project team not using the naming convention...).
- This file must be uploaded on Moodle. Deadline for the final package (including the video pitch if applicable) : November 23rd, 2025, 23 :59.
- Oral presentation : November 17th, 2025, 10h45-12h45. The order of presentations will be determined during TME5 and TME6 sessions.