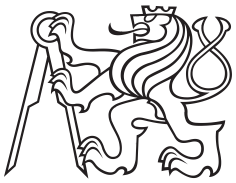
Bachelor's Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Application for data analysis from the world university rankings

Aziza Kuspekova

Supervisor: doc. Ing. Ivan Jelínek, CSc.
January 2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kuspekova** Jméno: **Aziza** Osobní číslo: **467508**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Aplikace pro analýzu dat ze světového žebříčku univerzit

Název bakalářské práce anglicky:

Application for data analysis from the world university rankings

Pokyny pro vypracování:

Cílem bakalářské práce je najít komplexní řešení pro analýzu dat a srovnání hodnocení univerzit. Bakalářská práce bude zaměřena na analýzu existujících mezinárodních hodnocení univerzit a srovnání stávajících jejich hodnocení, se zaměřením na analýzu dostupných dat z QS World University Rankings a analýza požadavků na hodnocení vysokých škol, jejich vážení a stanovení kritérií hodnocení. Výsledná aplikace poskytne přehled o shromážděných datech a výsledku analýzy prostřednictvím vhodného uživatelského rozhraní. Implementace bude řešena jako sada opakovaně použitelných programovacích modulů. Součástí bakalářské práce bude i návrh testování cílové implementace a ověření validity výsledků.

Seznam doporučené literatury:

Methodology | Ranking Web of Universities.” Webometrics, https://www.webometrics.info/en/Methodology. Accessed 15 January 2022.
O, Craig, and Chloe Lane. “QS World University Rankings 2022.” QS Top Universities, https://www.topuniversities.com/qs-world-university-rankings/methodology. Accessed 15 January 2022.
O, Craig, and Chloe Lane. “QS World University Rankings: Sustainable Development Goals.” QS Top Universities, https://www.topuniversities.com/university-rankings/world-university-rankings/sustain able-development-goals. Accessed 15 January 2022.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Ivan Jelínek, CSc. kabinet výuky informatiky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **02.02.2022** Termín odevzdání bakalářské práce: **10.01.2023**

Platnost zadání bakalářské práce: **30.09.2023**

doc. Ing. Ivan Jelínek, CSc. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. | Podpis studentky

Datum převzetí zadání

Acknowledgements

I would like to thank the supervisor of this bachelor's thesis, doc. Ing. Ivan Jelínek, CSc. for all genuinely helpful advice and help. This work would be impossible without the support of my family.

Declaration

Prohlašuji, že jsem předloženou práci vypracovala samostatně, a že jsem uvedla veškerou použitou literaturu.

V Praze, 5. ledna 2023

Abstract

World university rankings are gaining more attention in the last decade. Students are ready to go to another part of the world to get a degree in the most prestigious universities. Their choice is greatly influenced by the annual university ranking published by independent organisations. Each university wants to take a significant position in the ranking, but it needs to study and analyse what criteria negatively affect the overall score. An application for data analysis from ranking the best universities will help facilitate this process and allow universities to work with ranking data, analyse them and create visual reports.

Keywords: university ranking, data extraction, ranking analysis, web application, Spring, ReactJS, Microsoft Power BI

Supervisor: doc. Ing. Ivan Jelínek, CSc.

Abstrakt

Světový žebříček univerzit získává v posledním desetiletí stále větší pozornost. Studenti jsou připraveni vyrazit do jiné části světa, aby absolvovali nejprestižnější univerzity. Jejich výběr je do značné míry ovlivněn každoročním žebříčkem univerzit vydávaným nezávislými organizacemi. Když univerzita chce získat významnou pozici v žebříčku, musí si prostudovat a analyzovat, jaká kritéria pozitivně a negativně ovlivňují celkové skóre. Aplikace pro analýzu dat ze světového žebříčku univerzit tento proces usnadní a umožní univerzitám pracovat s daty hodnocení, analyzovat je a vytvářet vizuální sestavy.

Klíčová slova: žebříček univerzit, extrakce dat, analýza žebříčku, webová aplikace, Spring, ReactJS, Microsoft Power BI

Překlad názvu: Aplikace pro analýzu dat ze světového žebříčku univerzit

Contents

Figures

Tables

Listing

Chapter 1

Introduction

The project's primary goal is to design an application that would be able to collect data of the ranked universities from the annual QS World University Rankings (QS WUR). The application would provide an overview of collected data and the analysis result through a suitable user interface. The project will include research of existing rankings and their methodologies, a comparison of available data on universities and how to obtain them, and especially the specification of requirements for the analysis of the collected data.

Key steps of the project:

- analysis and comparison of existing international rankings of universities,
- analysis of available data from the QS WUR,
- analysis of existing solutions for data extraction from QS WUR,
- analysis of requirements for evaluation of universities, their weighting, determination of evaluation criteria,
- collection of functional and non-functional requirements for the target application,
- selection of appropriate environment and software tools for implementation,
- design of application architecture and interactive user interface.

The result of the project will include the implementation of the fundamental solution modules and will be a starting point for the final development and testing.

Chapter 2

Analysis and comparison of existing international rankings of universities

This chapter aims to analyse the currently existing well-known world university rankings, highlighting their pros and cons. This chapter's conclusion describes why QS WUR is chosen as the most suitable ranking for this work.

2.1 QS World University Rankings

QS World University Rankings is the world university ranking founded by Quarcquarelli Symonds, the leading provider of services, analytics, and insight to the global higher education sector. The QS WUR portfolio, inaugurated in 2004, has become the world's most popular source of comparative data about university performance. Their flagship website, www.TopUniversities.com, was viewed 149 million times in 2019, and over 94000 media clippings mentioning QS were published by media outlets across the world in 2019.[1]

QS WUR aim to inform prospective international students about their study choices and graduate outcomes.

QS is annually publishing university rankings, measuring the following indicators:

- academic reputation from a global survey,
- faculty-student ratio,
- citations per faculty from Scopus[1],
- employer reputation from a global survey,
- the proportion of international students,
- the proportion of international staff.

Table 2.1 lists the advantages and disadvantages of QS WUR.

Advantages	**Disadvantages**
Data available since 2014	Data is incomplete for less famous universities
The primary focus group of the ranking is students	Data from 2004 to 2014 is not available
An overall score is measured based on a wide variety of indicators	
Wide range of rankings, including ranking by region and by subject	

Table 2.1: QS WUR advantages and disadvantages

2.2 Academic Ranking of World Universities (ARWU)

ARWU was the first organization that published a widely known global university ranking, founded by Shanghai Jiao Tong University in 2003. ARWU presents the world's top 1000 research universities annually based on transparent methodology and objective third-party data.[3] The organisation compares higher education institutions based on the following six indicators:

- alumni of an institution winning Nobel Prizes and Fields Medals,
- the staff of an institution winning Nobel Prizes and Fields Medals,

[1]Scopus is the largest abstract and citation database of peer-reviewed literature: scientific journals, books and conference proceedings.[2]

- highly cited researchers,
- papers published in Nature and Science academic journals,
- papers indexed in Science Citation Index-Expanded[2] and Social Science Citation Index[3],
- per capita academic performance of an institution.

ARWU is primarily focused on the academic achievements of universities, measuring citations, research performance, and the number of Nobel Prizes winners between employees and alumni. More than 2500 universities are ranked by ARWU every year, and the best 1000 are published.[4]

Table 2.2 describes the advantages and disadvantages of ARWU university ranking.

Advantages	**Disadvantages**
Measuring academic excellence-appealing for master's and PhD applicants	Only the world's top 1000 research universities are published annually
An overall score is measured based on a wide variety of indicators	

Table 2.2: ARWU advantages and disadvantages

2.3 Times Higher Education (THE) World University Rankings

THE is a higher education data provider for the world's research-led institutions. Founded in 2004, it annually evaluates universities around the globe. Overall rankings, featuring more than 1,500 institutions, are accompanied by a series of subject-specific rankings to help students determine where to study. The ranking is focused on measuring research, teaching and impact performance of the universities.[5] THE's rankings have a stable methodology and calculate an overall score based on five following ranking indicators called "scores":

[2]Science Citation Index-Expanded is a citation index originally produced by the Institute for Scientific Information and created by Eugene Garfield.

[3]The Social Sciences Citation Index is a commercial citation index product of Clarivate Analytics.

- teaching score,
- research score,
- citations score,
- industry income score,
- international outlook score.

Table 2.3 presents the advantages and disadvantages of THE university rankings.

Advantages	**Disadvantages**
Stable methodology	Registration is required
Overall score is measured based on the wide variety of indicators	
Complete ranking data for the last five years	
Wide range of rankings, including ranking by region and by subject	

Table 2.3: THE World University Rankings advantages and disadvantages

2.4 Webometrics Ranking of World Universities(also Ranking Web)

Webometrics Ranking of World Universities is a ranking which uses both webometric(all missions) and bibliometric(research mission) indicators. Its primary objective is to promote Open Access[4] to knowledge generated by the university.[6] Webometrics is the largest independent, objective and free academic ranking that evaluates universities every six months. Since it is a research project, its methodology is not stable, and it is changing according to new findings or the availability of sources.

Table 2.4 lists the advantages and disadvantages of Webometrics Ranking of World Universities.

[4]Open access (OA) is a set of principles and a range of practices through which research outputs are distributed online, free of access charges or other barriers.

Advantages	Disadvantages
One of its core principles is being independent	The methodology is not stable, and it is changing according to new findings or the availability of sources
It has ranked 31000 institutions	The number of indicators is poor, so it is difficult to count the actual rank and understand the weak sides of the university
	Data for the previous years is not easily reachable or missing at all

Table 2.4: Webometrics Ranking of World Universities advantages and disadvantages

2.5 Conclusion

It is impossible to compare the rankings with each other since each of them uses different ranking methodologies and evaluation criteria. Some focus on evaluating the research activities, others on teaching quality and internationalisation. QS World University Rankings and THE World University Rankings are chosen as primary source of information because of their stable methodologies, ranking data consistency, and the client's requirements discussed in section 6.1. Furthermore, the performed analysis showed that these rankings have an outstanding reputation, long history of publications and a group of professionals standing behind their data. Other described university rankings, such as Academic Ranking of World Universities and Webometrics Ranking of World Universities, also have unique methodologies and data and might be considered as a source of information for the future development of this work, such as university comparison between different rankings.

Chapter 3

Analysis of available data from the QS World University Rankings

This chapter analyses the data available on the QS ranking webpage[1]. QS World University Rankings contains a wide range of rankings of universities, such as rankings by subject, region, etc. There is currently available data for the years 2014-2023.

3.1 List of the rankings page

The list of the rankings page provides an overview of all the universities ranked in the selected year. It contains three tabs that show different data for each university:

- university rankings tab — shows basic university information, such as name, location, overall score,
- ranking indicators tab — shows the overall score, international students ratio, international faculty ratio, academic reputation, employer reputation, citations per faculty, faculty student ratio,
- SDG ratings — show the Sustainable Development Goals indicators, such as environmental impact and social impact.

[1] www.topuniversities.com

Every row in the list contains a link to the university's detailed information page.

3.1.1 University rankings tab

The university rankings tab is located on the list page, described in the section 3.1, and it shows an ordered list of the universities, their rank and overall score.

Figure 3.1 shows how the web page and the university rankings tab look for the year 2023.

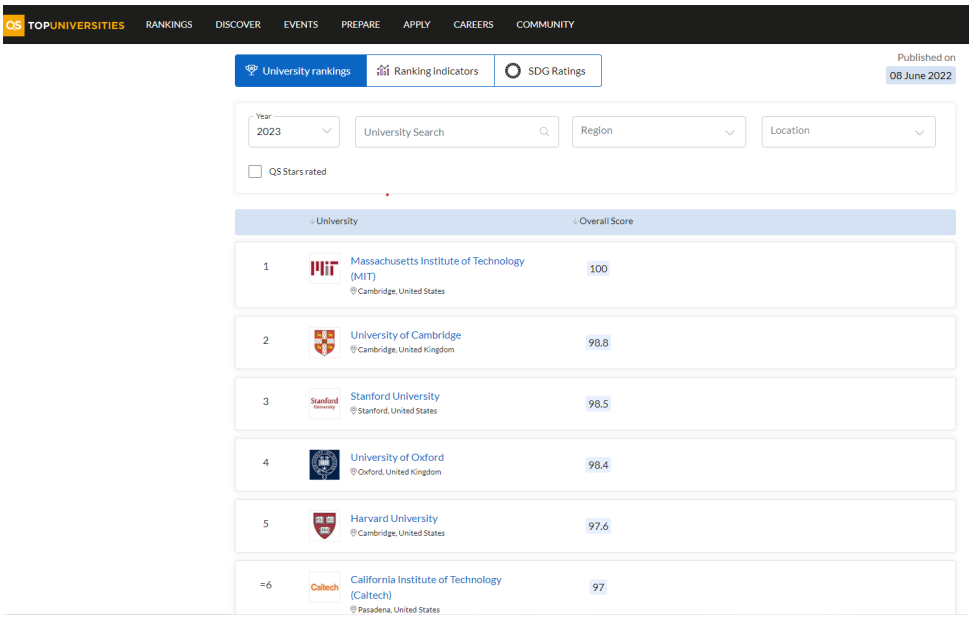


Figure 3.1: QS WUR 2023: University rankings tab

Table 3.1 provides information about fields and values present in the university rankings tab.

Name	Description	Example
Rank	The rank of the university among all the ranked universities	161
Name	Name of the university	Aarhus University
Location	City and country where the university is located	Aarhus, Denmark
Overall Score	An overall score	37.5

Table 3.1: University rankings tab data fields

3.1.2 Rankings indicators tab

The ranking indicators are the metrics used in the QS methodology. Universities are evaluated according to the following indicators, which are valid for the years 2017-2022:

- overall score,
- international students ratio,
- international faculty ratio,
- faculty-student ratio,
- citations per faculty,
- academic reputation,
- employer reputation.

During the years 2014-2017 universities were also evaluated according to the following subject indicators:

- social sciences and management,
- life sciences and medicine,
- engineering and technology,
- arts and humanities.

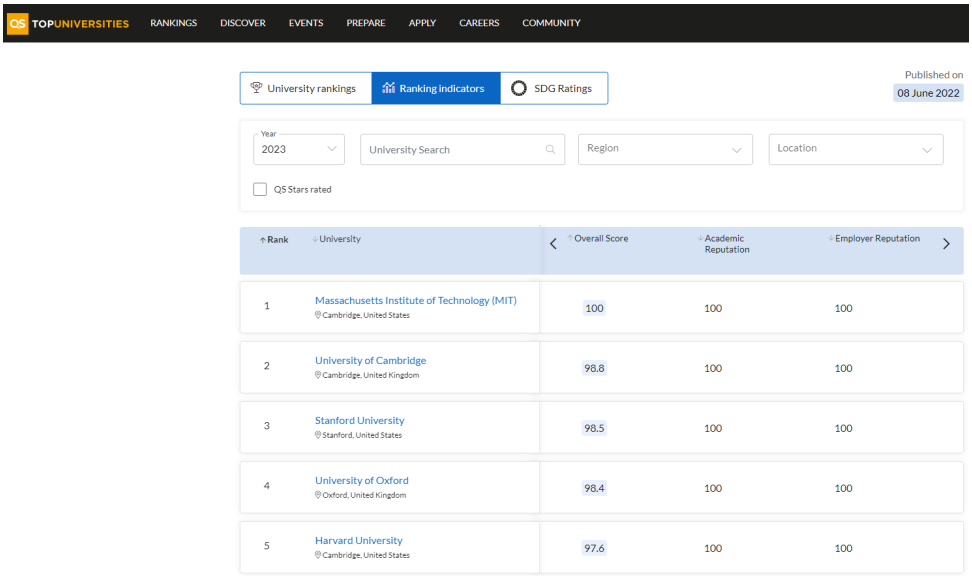


Figure 3.2: QS World University Rankings 2023: Ranking indicators tab

Name	Description	Example
International Students Ratio	The proportion of international students	5.6
International Faculty Ratio	The proportion of international staff	8.6
Faculty-student Ratio	Number of faculty members per student	69.6
Citations per Faculty	Total number of citations received by all papers produced by an institution across five years by the number of faculty members at that institution	92.1
Academic Reputation	Teaching and research quality, based on Academic survey	41.7
Employer Reputation	Employees are asked to identify those institutions from which they source the most competent, innovative and effective graduates	44.4
Social Sciences and Management	Metrics of the ranking by subject which was included in QS WUR in the years 2014-2016	85.3
Life Sciences and Medicine	Metrics of the ranking by subject which was included in QS WUR in the years 2014-2016	92.1
Engineering and Technology	Metrics of the ranking by subject which was included in QS WUR in the years 2014-2016	65
Arts and Humanities	Metrics of the ranking by subject which was included in QS WUR in the years 2014-2016	87.6

Table 3.2: Rankings indicators tab data fields

Figure 3.2 shows how the web page and the ranking indicators tab look for the year 2023.

Table 3.2 explains each ranking indicator's meaning and gives its example value.

3.1.3 SDG Ratings tab

In 2021 QS World University Rankings introduced the SDG Ratings tab. SDG stands for the Sustainable Development Goal, established by the United Nations in 2015. SDG Ratings measure an institution's research focus on the SDGs, which are split into environmental and social impact. "*Institutions have been awarded a medal — Gold, Silver, Bronze or Candidate — which reflects the impact of their output in a particular area of sustainable development.*"[7]

Figure 3.3 shows how the web page and the SDG Rankings tab look for the year 2023.

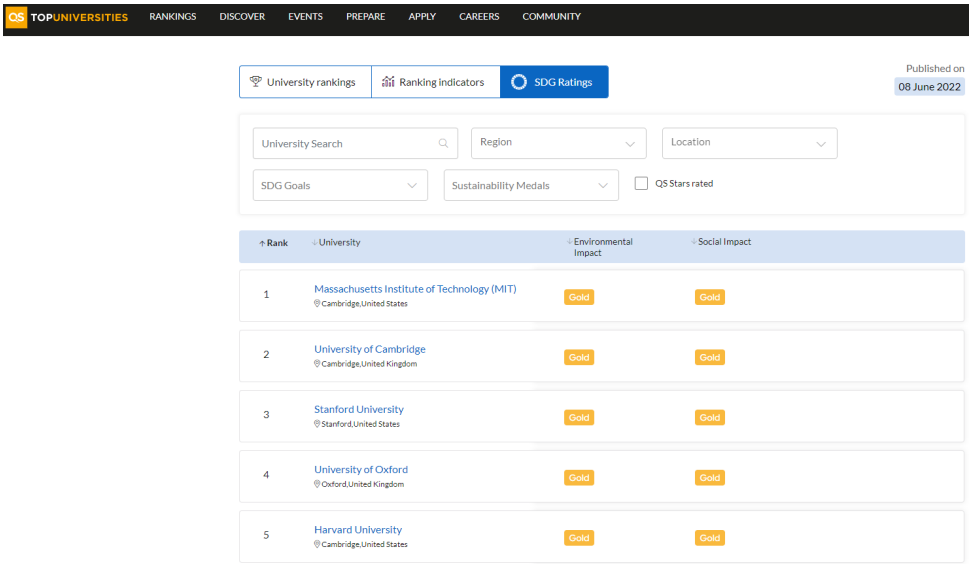


Figure 3.3: QS World University Rankings 2023: SDG Rankings tab

Table 3.3 describes the meaning of columns in the SDG Ratings tab.

Name	Description	Example
Environmental Impact	Metric to evaluate the environmental impact of an institution according to United Nations' SDG	Gold
Social Impact	Metric to evaluate social impact of an institution according to the United Nations' SDG	Silver

Table 3.3: SDG Ratings tab data fields

Chapter 4

Analysis of existing solutions for data extraction from QS World University Rankings

This chapter aims to analyse existing solutions to one of the project's goals — the ways to collect university rankings data.

4.1 Approaches

Initial research identified the following existing approaches to access data of QS World University Rankings:

- usage of Top Universities Scraper[1],
- usage of official supplement available at `www.topuniversities.com`,
- usage of official excels provided by the QS WUR.

[1] https://apify.com/vbartonicek/topuniversities-scraper

4.2 Usage of Top Universities Scraper

Top Universities Scraper is an API created and shared by Vratislav Bartonicek at Apify platform. It is intended to get a list of top universities based on the QS World University Rankings in various text file formats such as HTML table, JSON, CSV, Excel, XML or RSS feed.[8] It is configurable to run in two modes:

- basic mode — to get a list of ranked universities with basic data like name, country, and link to its detail,
- detailed mode — to get detailed information about the universities by visiting its detail pages.

Apify platform offers a one-month free trial version for newly registered users; after, a monthly subscription costs $49.[9] Apify users do not have access to the code of the mentioned solution, so it impossible to modify or improve it.

4.3 Usage of official supplement

An official supplement is a free guide yearly published by the QS WUR. It includes tables with data and insights into the world's top universities, commentary and analysis of this year's results. It can be downloaded in PDF file format by registered users.[10]

The data in the supplements for different years has no strictly defined structure making it challenging to analyse.

4.4 Usage of official excels

For registered users, QS WUR provides "Rankings excel tables". "Rankings excel tables" is a file in an Excel format that includes the following data:

- Overall rank & score,
- Classifications,
- Academic Reputation (rank & score),
- Employer Reputation (rank & score),
- Faculty Student Ratio (rank & score),
- Citations per Faculty (rank & score),
- International Faculty (rank & score),
- International Students (rank & score).[11]

Currently, there are available tables for the years 2021 and 2022. The structure and content of the "Rankings excel tables" are different for available years.

4.5 Conclusion

Ranking data necessary for future analysis has to cover criteria such as accuracy, completeness and authenticity.

Table 4.1 describes the advantages and disadvantages of the existing solutions mentioned in this chapter. The investigation of the existing solutions demonstrates that they do not meet the requirements above and can not be considered an appropriate solution.

Solution	PROS	CONS
Top Universities Scraper	It is a ready-to-use solution to get the data for future analysis.	There is no ability to modify or configure the solution's code. The tool can get unsupported with time. The tool can contain errors that would prevent successful data collection. Paid solution.
Official supplements	A ready-to-use solution to get the data for future analysis. Free for registered users.	Incomplete data. Supplements provide only information about selected universities. Complicated data processing because of the provided format. Registration is required.
Official excels	Free for registered users. The data is provided in machine-readable format.	Incomplete data. Only data for the years 2021 and 2022 are available. Registration is required.

Table 4.1: Pros and cons of the existing solutions

Chapter 5

Analysis of requirements for evaluation of universities

This chapter describes the QS World University Rankings' evaluation methodology, explains the meaning and weighting of ranking indicators required to assess universities.

5.1 The QS World University Rankings methodology

The rankings methodology is based on six metrics: academic reputation, employer reputation, faculty-student ratio, citations per faculty, international faculty ratio and international student ratio.

Table 5.1 describes the meaning of the ranking indicators and their weight from the overall score in percentage.

Indicator name	Weight from the overall score in percentage	Description
Academic reputation	40	An indicator is based on the annual QS Academic Survey, which collects higher education experts' opinions on research and teaching quality areas.
Employer reputation	10	An indicator is based on QS Employer Survey, which asks employers to identify those institutions from which they source the most competent, innovative and effective graduates.[12]
Faculty-Student Ratio	20	Calculated by dividing the number of students validated by QS by the Faculty figure validated by QS, it aims to serve as a proxy measure for the learning and teaching environment of the institution.[13]
Citations per faculty	20	An indicator measures the research level of a university and calculates the total number of citations received by all papers produced by an institution across five years by the number of faculty members at the institution. All citation data is sourced using Elsevier's Scopus database, the world's largest repository of academic journal data.[12]
International Faculty Ratio	5	The number of faculty staff who contribute to academic teaching or research or both at a university for a minimum period of at least three months and who are of foreign nationality as a proportion of overall faculty staff.[14]
International Student Ratio	5	The total number of undergraduate and postgraduate students who are foreign nationals and who spend at least three months at your university as a proportion of the total number of undergraduate students and postgraduate students overall.[15]

Table 5.1: QS World University Rankings methodology

5.2 Data required for analysis

Based on the methodology and the analysis described in chapter 3, the following fields have been chosen as crucial for future proof-of-concept application and analysis:

- name of the university,
- rank,
- location,
- overall score,
- international students ratio,
- international faculty ratio,
- faculty-student ratio,
- citations per faculty,
- academic reputation,
- employer reputation.

Chapter 6

Design

The following chapter focuses on functional and non-functional requirements, entity relationship diagram (ERD) and wireframes put on the target application for data collection and proof-of-concept application for universities rankings overview and their analysis.

6.1 Functional requirements

- **FR1:** The application for data collection has to gather available university rankings and indicators described in section 5.2.
- **FR2:** The application for data collection has to focus on the data from QS World University Rankings and THE World University Rankings. This work does not include the collection and processing of data from other sources.
- **FR3:** Collected data has to be saved in the database.
- **FR4:** The proof-of-concept application for data overview and analysis has to provide the following modules:
  - overview of the university rankings,
  - analysis of university rankings,
  - comparison of multiple universities and their rankings,
  - management of data collection works.

6.2 Non-functional requirements

- **NFR1:** The application for data collection has to be designed and implemented in a way that allows adding new modules for the collection of the data from other university rankings sources.

- **NFR2:** The application for data collection is expected to be able to process 1000 university rankings in less than one hour.

6.3 Characteristics and user classes

6.3.1 Use Case Model

The use case diagram describes a set of use cases that the system performs with the help of external users of the system, where each user performs a specific role.

Figure 6.1 illustrates the use case diagram for the ranking analysis application. The system has three roles: Admin(or Administrator), Moderator and Basic User.

The administrator's role allows the user to manage the internal processes of a system. Admin can create, update and delete Moderators and Basic users. He manages the database and provides technical system support.

Moderator is a user that is responsible for the content of an application. Moderator has access to the system executor, which extracts the data from an external web page, loads and saves them into the database.

Basic user is able to view the content, download the data in CSV format.

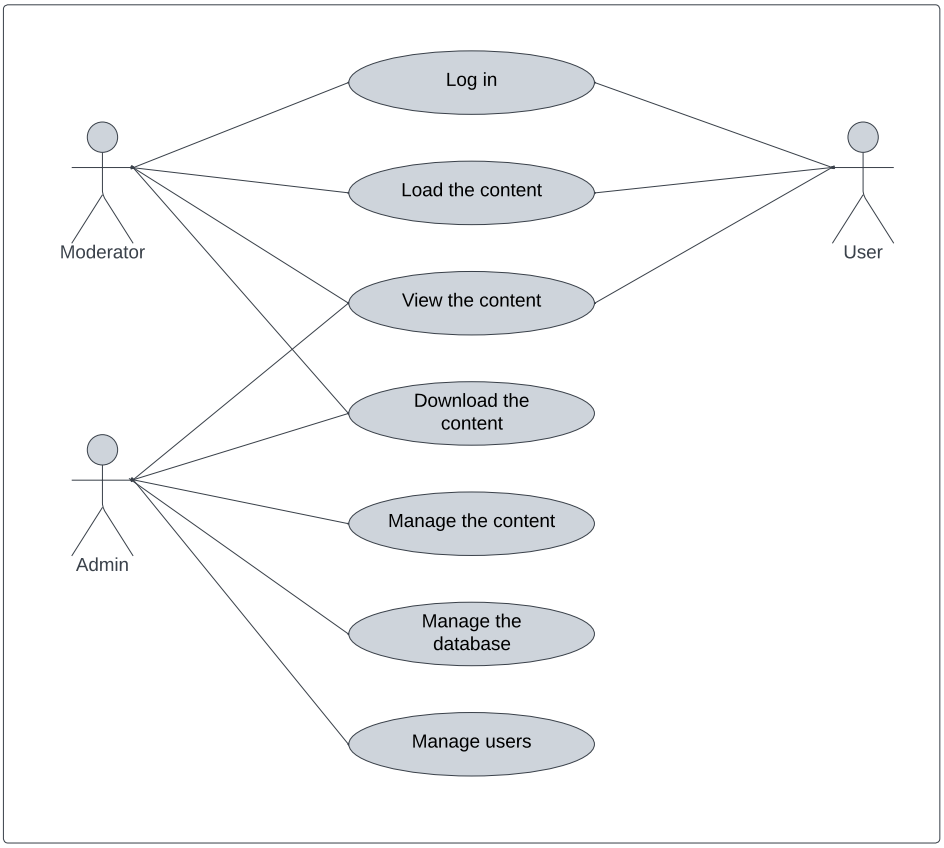


Figure 6.1: Use case diagram

6.4 Entity relationship diagram

The entity relationship diagram (ERD) illustrates the relationship of the entity set that is stored in a database. It takes an essential role in the design process of the relational database. The building blocks of the ERD are entities with attributes and relationships between them. "*When we speak of an entity, we normally speak of some aspect of the real world which can be distinguished from other aspects of the real world. A relationship is some association between entities. As a real-world aspect, an entity is characterised by a number of properties or attributes. Values assigned to attributes are used to distinguish one entity from another.*"[16]

Figure 6.2 illustrates the ERD for the THE ranking, which contains three entities: university, ranking score and rank. In THE ranking, each university has a rank and ranking score record per year, and this is represented by a one-to-many relationship between the university and the ranking score and

the university and rank entities in the ERD.

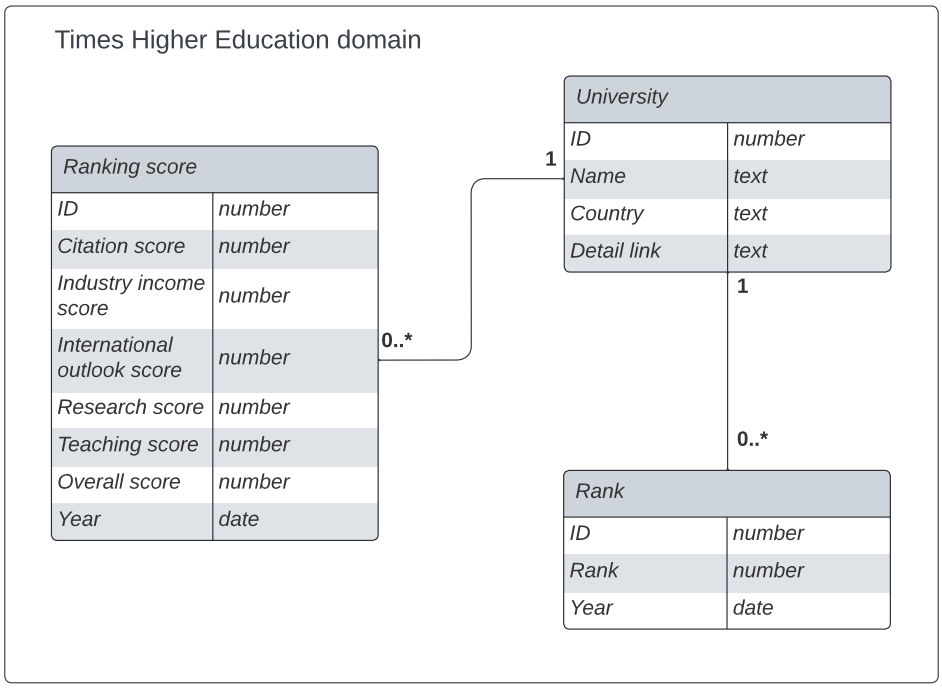


Figure 6.2: Logical entity relationship diagram for THE WUR

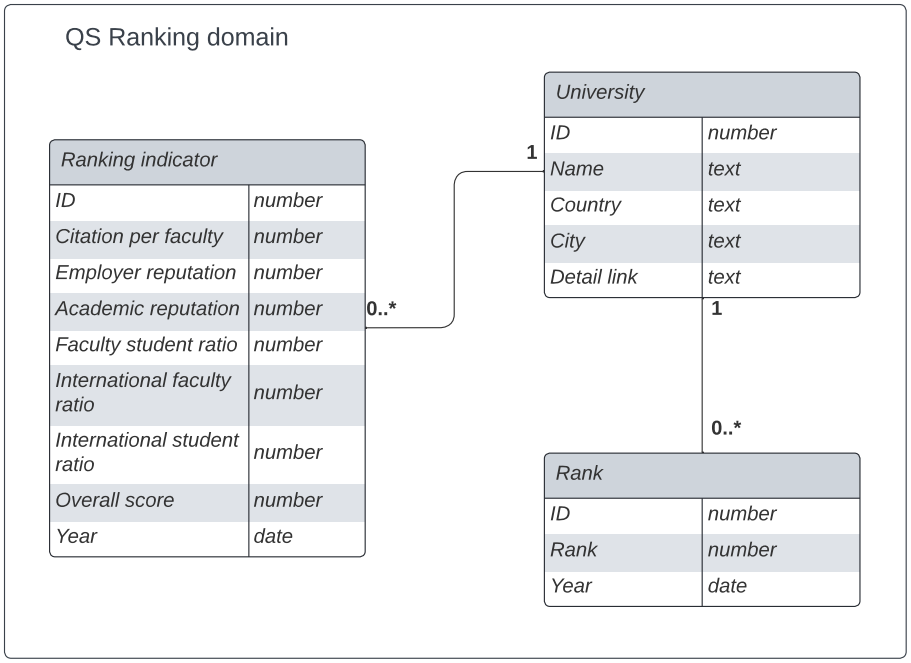


Figure 6.3: Logical entity relationship diagram for QS WUR

Figure 6.3 illustrates the ERD for the QS ranking, which also consists of three entities: university, ranking indicator and rank. The relationship structure and cardinality between entities are the same as on THE ranking diagram in figure 6.2.

6.5 Prototype

Prototype describes the main functionality of an application and its final look. The core functionalities of an application are represented on three web pages: ranking overview, ranking analysis and university comparison. On each page, the user can switch between the data of QS WUR and THE WUR.

The ranking overview page, shown in figure 6.4, contains an ordered list of universities. It allows users to filter the data by year, university and country.

Figure 6.4: Rankings overview page

The ranking analysis page, shown in figure 6.5, allows users to convert data spreadsheets into understandable visual forms for future analysis. Data of each indicator is visualised within graphs for the selected university in a specified report period.

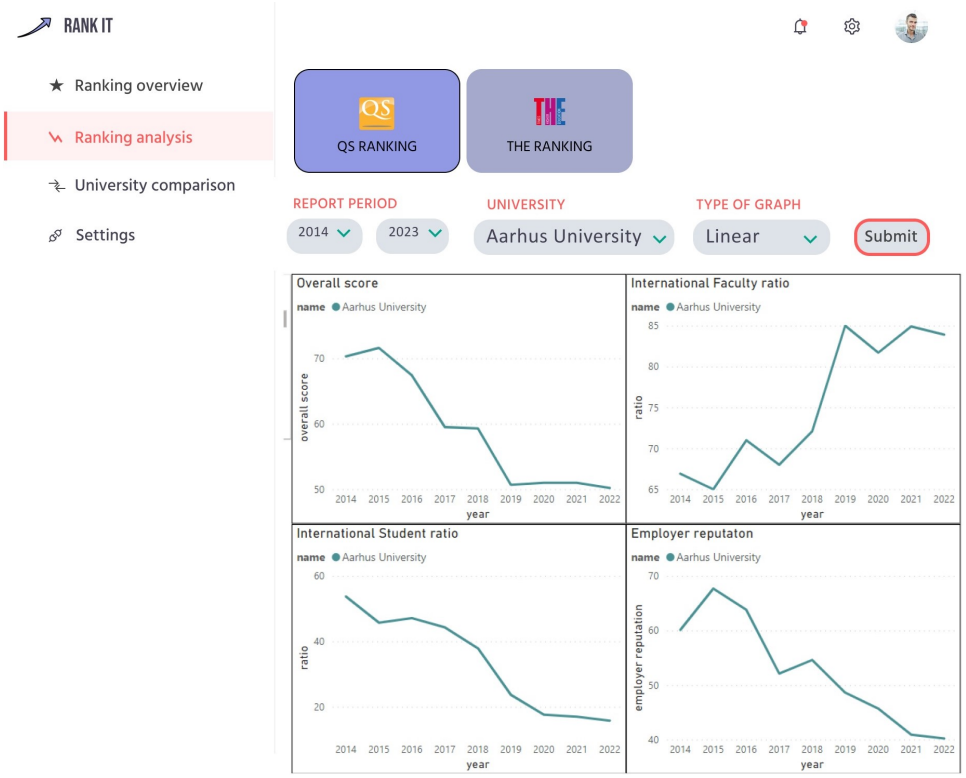


Figure 6.5: Ranking analysis page

Figure 6.6: Universities comparison page

The university comparison page, shown in figure 6.6, allows users to compare selected universities by scores of indicators. Comparison is visualised within graphs for each indicator in the set report period.

Chapter 7

Selection of appropriate software tools and environments for implementation

This chapter describes the tools and technologies used to implement the application for data collection and analysis.

7.1 Server-side

7.1.1 Java

The data scraping application is implemented in the programming language Java version 17. Java is a general-purpose, concurrent, class-based and object-oriented language.[17] Java is chosen for the implementation of the application for its tooling, community support and previous experience with this language.

7.1.2 Spring

In addition to the Java programming language, the Spring Boot framework version 2.6.1 was used. Spring Boot has developed from another popular

framework in the Java programming language world called Spring. Spring was released two decades ago, in 2002, and with time evolved into a massive tooling set consisting of more than 20 different modules.[18] Spring Boot Starter Web, JPA, Security, Validation, Actuator packages were used for implementation.

7.1.3 Jsoup

Jsoup is an open-source project distributed under MIT license. "*It provides a very convenient API for fetching URLs and extracting and manipulating data, using the best of HTML5 DOM methods and CSS selectors.*"[19] This dependency is required for the data scraping application to parse scraped HTML pages and extract pieces of information about universities' rankings. The latest version of this library, 1.14.3, is used.

7.1.4 Selenium

The Selenium tool is used to automate the process of launching and controlling web browsers. "*Selenium is an open source umbrella project for a range of tools and libraries aimed at supporting browser automation*".[20] It provides an Application Programming Interface (API) to access all major browsers and emulate user interaction with them. Selenium is required to access pages at `https://topuniversities.com` and process them. The major version 4 was used for the implementation.

7.1.5 PostgreSQL

The PostgreSQL database system is used for storing and retrieving scraped data. It is an object–relational database management system (ORDBMS) maintained as an open-source project.[21] To work with the database in Java, the PostgreSQL JDBC driver needs to be downloaded and configured. PostgreSQL was chosen for its performance, simplicity of use and other tools that improve developer experience, such as pgAdmin.[22]

7.1.6 Other libraries

There are other, less spread libraries worth mentioning regarding the development of the data scraping application. Junit[1] major version 5 and Mockito[2] major version 4 are used for unit testing. Java JWT [3] is an open-source library used to simplify JSON Web Token management. "*JSON Web Token (JWT) is an open standard (RFC 7519) [23] that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.*"[24] JWT is required to implement authentication and authorisation for the client application to access the web API of the data scraping application.

7.2 Client-side

7.2.1 TypeScript

The implementation of the client-side application for presentation and analysis of gathered data about universities is performed with TypeScript programming language. "*TypeScript is a strongly typed programming language that builds on JavaScript, giving you better tooling at any scale.*"[25] TypeScript compiles into plain JavaScript, which is supported by all major browsers. The TypeScript was chosen for its popularity in development of web applications and its similarity to Java programming language.

7.2.2 React

"*React is a JavaScript library for building user interfaces.*"[26]. As mentioned above, TypeScript compiles into JavaScript, making React and TypeScript a good combination of tools for developing front-end applications. React was chosen as a core library to support the implementation of the client-side application for the presentation and analysis of gathered data about universities.

[1] `https://junit.org/junit5/`
[2] `https://site.mockito.org/`
[3] `https://github.com/jwtk/jjwt`

7.2.3 Material UI (MUI)

MUI provides a robust, customisable, and accessible library of foundational and advanced components, enabling building the design system and developing React applications faster.[27] MUI simplifies the process of building the client-side application.

7.2.4 Microsoft Power BI

"*Microsoft Power BI is a leading business intelligence and analytics platform that supports both self-service data visualisation and exploration as well as enterprise BI deployments.*"[28] This tool allows building dashboards with university rankings data visualisations, which then can be embedded into the client-side application for data analysis.

7.2.5 Other libraries

Nowadays, almost every developer takes advantage of already existing packages and tools available on the market while developing their application. There are other packages used for the development of the client-side application. "*Axios is a simple promise-based HTTP client for the browser*".[29] Axios is used to perform HTTP calls from client to server application.

7.3 Conclusions

A combination of multiple programming languages, libraries and tools was considered for the implementation of client-side and server-side applications. Technologies described in this chapter were chosen because most of them are open-source, modern, have good community support and fulfil the requirements for the development of the applications.

Chapter 8

Implementation

This chapter describes the overall architecture of the system and the implementation of the server-side and client-side applications.

8.1 Application architecture

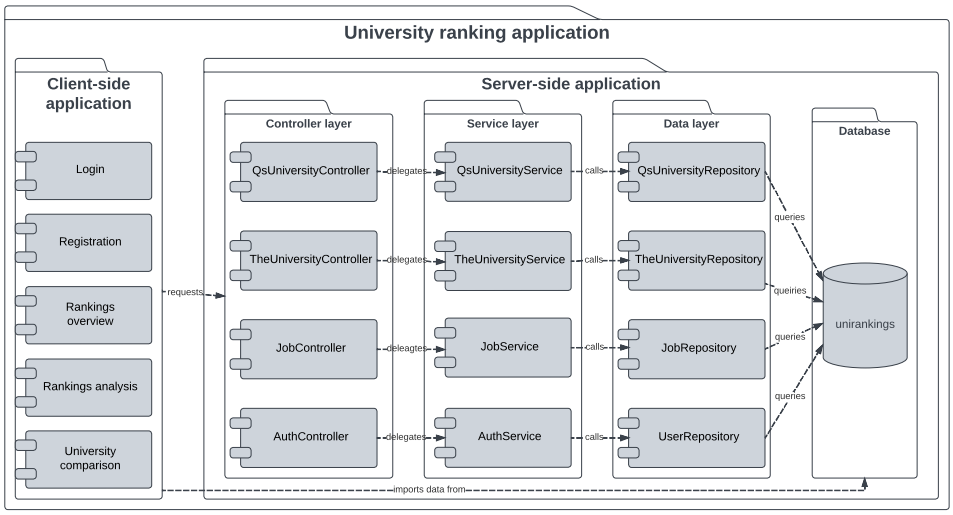


Figure 8.1: Architecture diagram

The diagram in figure 8.1 shows the application's architecture, which follows the well-established three-tier architecture for client-server applications. "*Three-tier architecture is a client-server software architecture pattern in which the user interface (presentation), functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms.*"[30] The diagram presents an overview of the layers, modules and components in the system and how they interact between each other.

8.2 Server-side application

8.2.1 Controller layer

The controller layer in the server-side application consists of 4 main components annotated with Spring `@RestController`. `@RestController` and other annotations from the Spring Web module help to express request mappings, request input, exception handling, and more.[18] The following figure, 8.1, shows the handler's implementation for the university detail request.

```
@GetMapping(value = "/universities/{id}", produces =
    "application/json")
public ResponseEntity<University>
    getUniversity(@PathVariable("id") long id) {
  University university = service.findUniversity(id);
  if (university != null) {
    return new ResponseEntity<>(university, new HttpHeaders(),
        HttpStatus.OK);
  } else {
    return new ResponseEntity<>(HttpStatus.NOT_FOUND);
  }
}
```

Listing 8.1: Controller method for the university detail resource

The application resources, such as `University` or `Job`, are represented in JSON format. "*JSON is a lightweight, text-based, language-independent syntax for defining data interchange formats.*"[31] The object conversion is done with the help of the Jackson library, which is implicitly included in the Spring Web module.

The controller layer is primarily focused on handling incoming HTTP

requests, delegating their processing to the service layer, and then returning the HTTP response. The communication with the service layer happens through the injected services. For example, code 8.2 shows how `TheUniversityController` gets access to `TheUniversityService`.

```
private final UniversityService service;
public UniversityController(UniversityService service) {
  this.service = service;
}
```

Listing 8.2: `TheUniversityController` constructor

8.2.2 Service layer

As mentioned in the previous section, 8.2.1, the controller layer delegates the processing of the requests to the service layer, also known as a business layer in the layered architecture pattern. "*A Service Layer defines an application's boundary and its set of available operations from the perspective of interfacing client layers. It encapsulates the application's business logic, controlling transactions and coordinating responses in the implementation of its operations.*"[32]

The service layer in the application is represented by the following four components: `QsUniversityService`, `TheUniversityService`, `JobService`, `AuthService`.

Both `QsUniversityService` and `TheUniversityService` focus on retrieving data for QS and THE university rankings, respectively.

The `JobService` defines logic for running a new job to collect information from QS and THE university rankings. The flow of running the job is implemented in `runNewJob` method in `JobServiceImpl`.

The `AuthService` contains logic to register and log in users in the system. The `registerUser` method implements validation rules for registering a new user, such as the existence of the user with the same username or email already in the system. The `authenticateUser` method focuses on the authentication process itself and the creation of a JSON Web Token for authentication of the consequent requests from the authenticated user. The JWT generation logic is encapsulated in the `JWTUtils` class, which contains utility methods to

generate and validate the token. The code 8.3 shows the `generateJwtToken` method.

```
public String generateJwtToken(Authentication authentication) {
   UserDetailsImpl userPrincipal = (UserDetailsImpl)
       authentication.getPrincipal();
   return Jwts.builder()
   .setSubject((userPrincipal.getUsername()))
   .setIssuedAt(new Date())
   .setExpiration(new Date((new Date()).getTime() +
       jwtExpirationMs))
   .signWith(SignatureAlgorithm.HS512, jwtSecret)
   .compact();
}
```

Listing 8.3: `generateJwtToken` method

All of the mentioned services access the data layer with the help of Spring JPA repositories instances, which are managed by the Spring framework.

8.2.3 Data layer

"*This layer provides access to data hosted within the boundaries of the system, and data exposed by other networked systems; perhaps accessed through services. The data layer exposes generic interfaces that the components in the business layer can consume.*"[33]

In the application, the data layer is implemented with the help of Spring Data repositories, which provide access to standard methods for managing data, such as `findAll`, `save`, `delete`, etc. Four repository interfaces extend from Spring `JpaRepository`, which adds sorting and filtering capabilities for the data fetching operations on top of standard CRUD operations. Code 8.4 shows how the `TheUniversityRepository` class defines a method that uses filtering API to filter the found universities by name and country.

```
University findFirstByNameAndCountry(String name, String country);
```

Listing 8.4: `findFirstByNameAndCountry` method

"*The goal of the repository abstraction of Spring Data is to reduce the effort required to implement data access layers for various persistence stores significantly.*"[34] Even though, Spring Data simplifies the implementation of basic methods for data access, it also provides the ability to define custom,

more sophisticated queries for data retrieval using Java Persistence Query Language (JPQL) defined in Java Persistence API specification. Code 8.5 presents an example of such a query from the `TheUniversityRepository` class.

```
@Query("SELECT u FROM theUniversity u JOIN FETCH u.ranks r "
       + "JOIN FETCH u.rankingScores rs "
       + "WHERE r.year = rs.year "
       + "AND (:year IS NULL OR r.year = :year) "
       + "AND (:country IS NULL OR u.country = :country) "
       + "AND (:name IS NULL OR u.name LIKE '%:name%')")
List<University> findUniversities(@Param("year") String year,
    @Param("country") String country, @Param("name") String name);
```

Listing 8.5: `findUniversities` method and the `@Query` annotation

8.2.4 Database

The application uses a relational database from PostgreSQL for storing data about universities, users and jobs. Figure 8.2 shows the structure of the database and its objects.

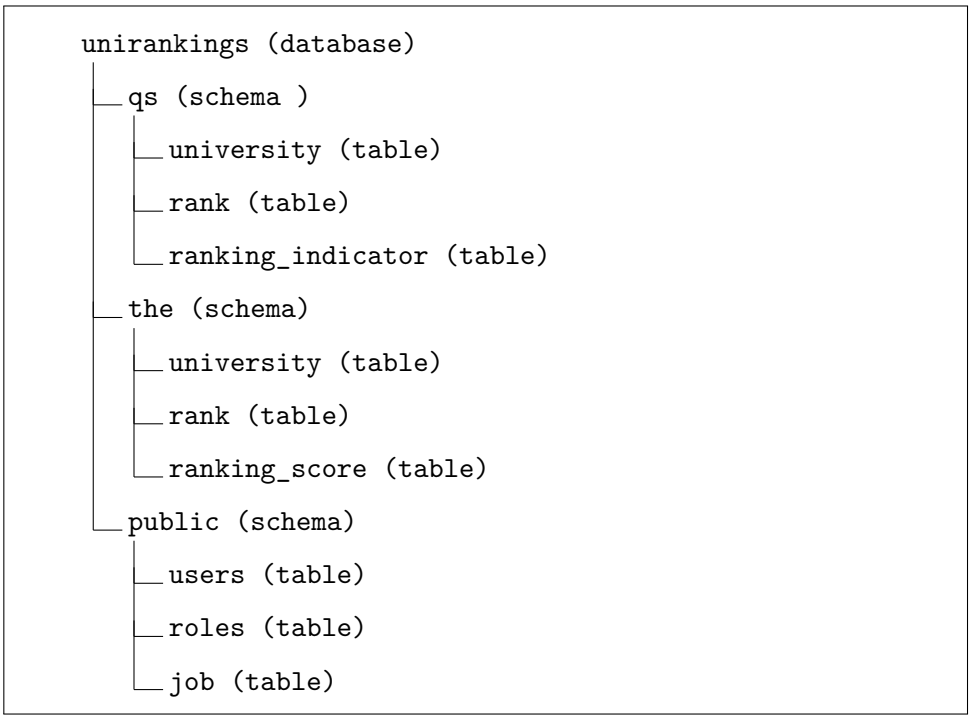


Figure 8.2: Database structure

The `qs` and `the` schemes are dedicated to storing data about QS and THE university rankings, respectively. The `public` schema is for storing data about jobs, users and their roles.

8.3 Extraction process

This section provides an overview of the extraction process, which is responsible for the data extraction from the web pages of university rankings.

8.3.1 Launching web browser

Both `topuniversities.com` and `timeshighereducation.com` are web applications with dynamic content which makes it difficult to perform web scraping. "*Web scraping is a data science technique that deploys scripts for the extraction of structured data from websites.*"[35] Using a web browser is one of the approaches to data scraping from dynamic websites.

The Selenium tool, described in 7.1.4, is used to automate the process of launching and controlling web browsers. Class `BrowserFactory` implements `IBrowserFactory` interface that provides an API to set up and create a new `WebDriver`. "*WebDriver is an API and protocol that defines a language-neutral interface for controlling the behaviour of web browsers.*"[36]

8.3.2 Extraction flow

The extraction flow for QS WUR is located in `RankingsListExtractor` class in `topuniversities.extractor` sub-package. Its activity diagram is outlined on figure 8.3.

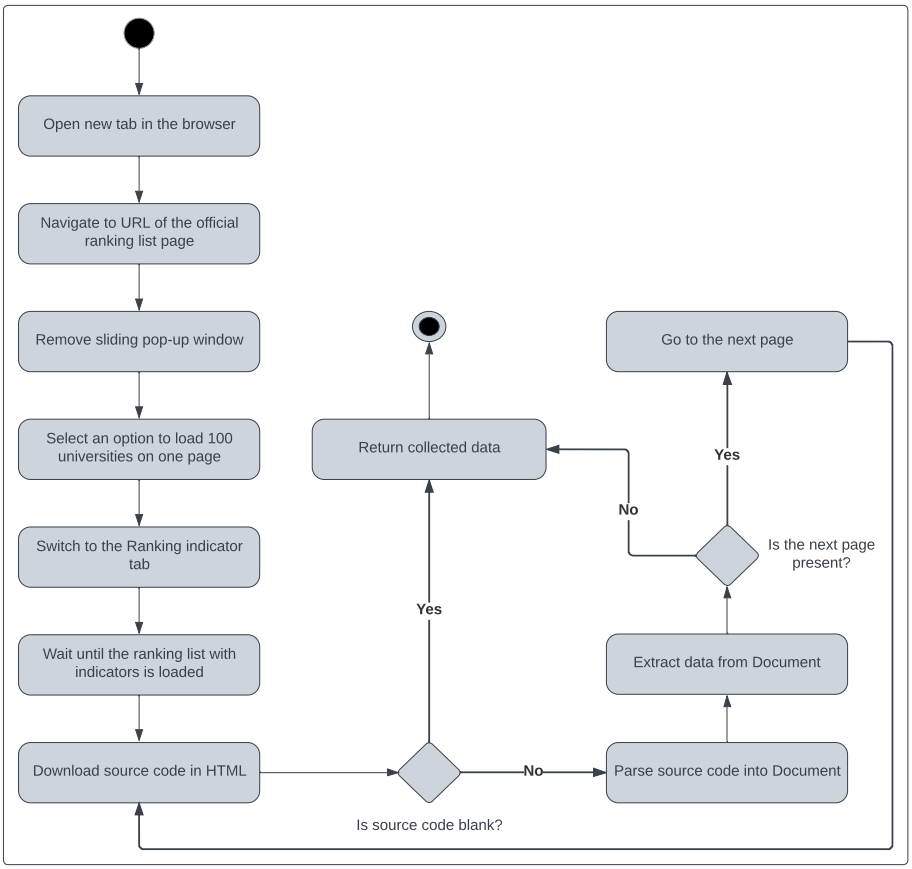


Figure 8.3: Activity diagram for the QS WUR extraction process

The extraction process consists of 5 core steps:

- launching the web browser in the headless mode,
- navigating to the university rankings URL for the given year,
- downloading of the page source code in the HTML format,
- parsing of the HTML document,
- selecting the required data.

After the browser is launched, as described in section 8.3.1, all interactions with it are done with the help of the `RankingsListPage` class, which implements a page object design pattern. The page object is an object-oriented class which acts as an interface for the web page that is being automated

or tested.[37] The `RankingsListPage` implements methods to remove sliding pop-ups from the page, expand the number of results per page, switch to the ranking indicators tab in the table with university rankings, check if the next page is available, and navigate to the next page.

8.3.3 Parsing

The HTML document is parsed with the help of Jsoup library, described in section 7.1.3. The `org.jsoup.nodes.Document` returned from `Jsoup.parse` method provides API for extracting the data using CSS and XPath selectors. Code 8.6 shows how rows in the table with ranking indicators are selected for further processing in the `AbstractRankingIndicatorParser` class.

```
private List<UniversityRankingIndicator>
    parseRankingIndicators(Document document) {
  Elements rankingIndicatorsTable =
      document.select("#ranking-data-load_ind");

  Elements indRowTags =
      rankingIndicatorsTable.select("div.row.ind-row");

  return indRowTags.stream()
          .map(this::parseRankingIndicatorTag)
          .collect(Collectors.toList());
}
```

Listing 8.6: `parseRankingIndicators` method

8.3.4 Model

`UniversityRankingIndicator` and `UniversityRankingScore` classes define the structure of the extracted data for QS WUR and THE WUR, respectively. Both of the classes are Plain Old Java Objects (POJO), which provide access to all the information collected during the extraction and parsing phase.

8.4 Client-side application

This section describes the overall structure of the client-side application code and details of selected implemented components.

8.4.1 Code structure

The `src` folder in the project consists of the following subfolders and core files:

- `components` — reusable React components for the pages,
- `model` — constants, enumerations and interfaces that define the model for the application objects,
- `pages` — high-level React components that represent different pages,
- `services` — reusable services for authorisation and data-fetching logic,
- `App.tsx` — root React component which defines routing between pages and common layout,
- `index.tsx` — the starting point of the application.

8.4.2 Routing

To enable navigation between pages of the application, the client-side routing is implemented with the help of the React Router library. This library allows to define routes, where each route specifies the URL path it matches and the component that it renders.

```
<Route path="/ranking-overview" element=
        {<RequireAuth> <RankingOverviewPage /> </RequireAuth>}
/>
<Route path="/" element={<MainPage />} />
```

Listing 8.7: Client-side routing definition for ranking overview and main pages

The code 8.7 from `App.tsx` shows route definitions for `RankingOverview` and `MainPage` components, which represent the main page and rankings overview page, respectively. The `RankingOverviewPage` component is wrapped with the `RequireAuth` component, which implements part of the authorisation process in the application.

8.4.3 Authentication and authorisation

As mentioned in 8.4.2, the `RequireAuth` component is responsible for verifying the access rights of the current user to the given page. In case the user is not logged in or doesn't have the correct roles to access the web page, the user is redirected to the login page or to the home page.

`RegistrationPage` and `RegistrationContainer` components implement the registration page. The login page is implemented in `LoginPage` and `LoginContainer` components. Both pages use `AuthService` from the `service` folder, which implements logic for registration, login, and logout operations.

When users log in to the application, their data are stored in `Storage` object for the `Document`'s origin, which can be used to access the current origin's local storage space.[38] Code 8.8 shows the implementation of the `login` function.

```
  try {
    const response = await axios.post(
      `${AUTH_API_URL}/signin`,
      {
        username,
        password,
      }
    );

    if (response.data.token) {
      localStorage.setItem('user', JSON.stringify(response.data));
    }

    return response.data;
  } catch (e) {
    return Promise.reject('HTTP to sing in user failed.');
  }
```

Listing 8.8: `login` function

8.4.4 Communication with server-side application

The communication between client-side and server-side applications is done via HTTPS, an extension of the HTTP. The `axios` and `axios-hooks` packages were used to have access to the HTTP client, which helps to perform HTTP requests and handle HTTP responses. The code 8.9 from the `AdminPage` component shows how data about jobs are loaded with the help of `useAxios` hook.

```
const [{ data: beJobs = [], loading, error }] = useAxios<Job[]>(
   {
      url: JOBS_API_URL,
      headers: authHeader(),
   },
   {
      useCache: false,
   }
);
```

Listing 8.9: Jobs data loading implementation

Authorisation header is required to be passed in the HTTP request headers to authorise client requests. `AuthHeader` is utility function, which uses information about user stored in local storage to create authorisation header.

8.4.5 User interface components

Most of the application's user interface is built with the help of the Material UI component library.

The `Table` component is designed to display sets of data with support for sorting and pagination. This component is used in admin and rankings overview pages to show information about executed extraction works and university rankings. `Table` is highly customisable and supports all relevant application requirements.

Figure 6.4 illustrates the "Ranking overview" page with a table of ranked universities and their indicators.

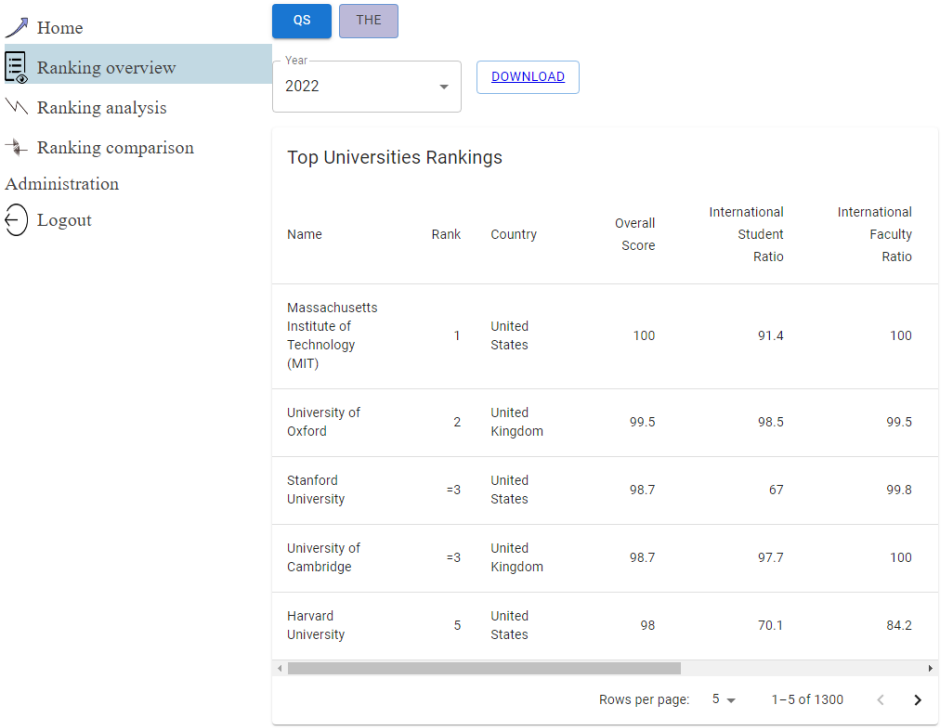


Figure 8.4: Ranking overview page

Another popular component in the application, which has presence almost in every page is `Grid`. The `Grid` component implements a grid system, which creates visual consistency between layouts while allowing flexibility across various designs.[39] Using grid system helps to define the page layout and ensure the application is responsive across different devices.

Other MUI components that are used in the application include `Button`, `FormControl`, `Select`, etc.

8.4.6 "Ranking analysis" and "University comparison" pages

The "Ranking analysis" and "University comparison" pages help users of the application to analyse university rankings and compare universities using data visualisations. Data visualisation helps to represent data collected during the extraction phase into easily understood and useful information.

Microsoft Power BI, described in section 7.2.4, allows building data visuali-

sations and grouping them into a dashboard, which then can be published to the web. After the dashboard is published, it is embedded into the client-side application with the help of `<iframe>` HTML element.

Microsoft Power BI connects to the PostgreSQL database, which is populated with university rankings during the extraction phase, with the help of NpgSQL[1] data provider, which is shipped with Power BI Desktop.

The dashboard on the "Ranking analysis" page includes multiple tiles for different ranking indicators. The tiles are customisable within slicers to select the university of interest, report period and type of graph. The graphs show how the rankings of the selected university developed over time. Figure 8.5 illustrates the dashboard of the "Ranking analysis" page.

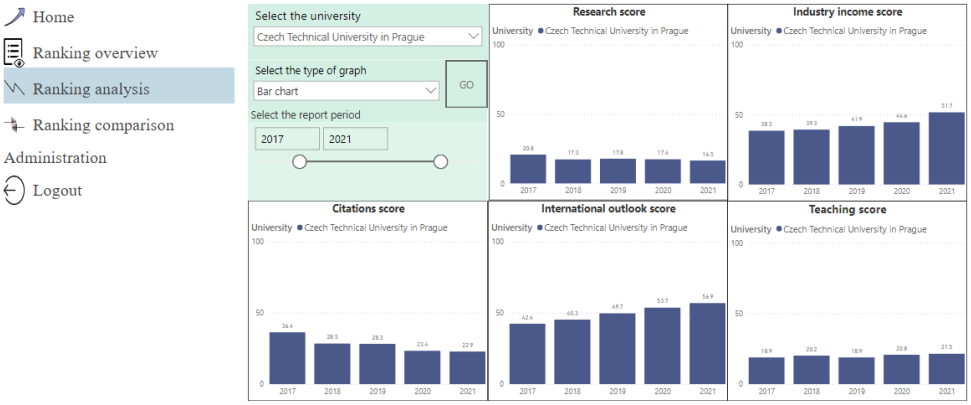


Figure 8.5: Ranking analysis page

The "University comparison" page contains similar visuals and the slicer that allows selecting multiple universities simultaneously for comparison. Figure 8.6 illustrates the dashboard of the "Ranking comparison" page.

[1] https://www.npgsql.org/

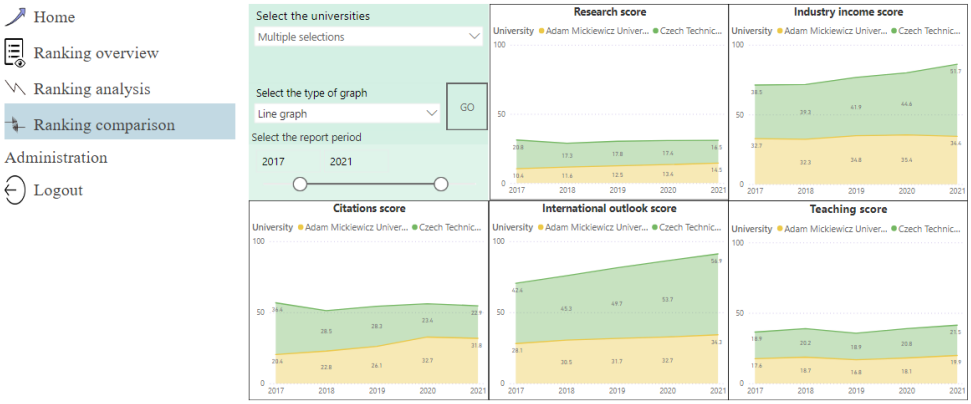


Figure 8.6: Ranking comparison page

Chapter 9

Testing

This chapter describes the process of testing the created system. Implemented functionality was verified with the help of unit, end-to-end and usability testing. Unit testing was automated, which reduced the risk of breaking the existing functionality while introducing new.

9.1 Unit tests

The server-side application is covered extensively with unit tests. The line-coverage measured with the help of JetBrains IDE[1] reaches 80%. The following list provides an overview of created test classes and what they test:

- `RankingsListExtractorTest` — covers the extraction logic and flow,
- `ParsingModelMapperImplTest` — checks that extracted data are correctly mapped to the database model,
- `RankingsListPageTest` — verifies the behavior of page objects that are used during the data extraction phase,
- `RankingScoreParserImplTest` — tests different scenarios when parsing HTML documents,

[1] IDE — Integrated Development Environment

- `AuthServiceImplTest` — checks user authentication and registration processes,
- `JwtUtilsTest` — verifies JWT-related logic.

The list doesn't include some of the implemented tests' classes that do not verify complex pieces of logic, such as `UniversityServiceImplTest`, `LoginRequestTest`, etc.

9.2 End-to-end tests

The following section focuses on the test scenarios and execution of end-to-end tests. "*End-to-end testing is a methodology used in the software development lifecycle (SDLC) to test the functionality and performance of an application under product-like circumstances and data to replicate live settings.*"[40]

9.2.1 Testing scenarios

Before starting with the test execution, it is important to design testing scenarios which cover the application use cases.

Test case 1. Successful login

Description	User is able to log in
Precondition	User is registered
Expected result	User is logged in
Steps	

1. Open Login page
2. Enter valid credentials into the form
3. Click Login button

Test case 2. Failed login

Description	User fails to log in with invalid credentials
Precondition	User is registered
Expected result	User is not logged in and sees error message “Entered credentials are invalid”

Steps

1. Open Login page
2. Enter invalid credentials into the form
3. Click Login button

Test case 3. Ranking overview for THE WUR

Description	User is able to see THE WUR for different years
Precondition	User is logged in
Expected result	User sees the correct data for each available year

Steps

1. Open the “Ranking overview” page
2. Select the “THE” tab
3. Select year 2022 from the “Year” select
4. Observe data in the table
5. Repeat steps 3 and 4 for different years

Test case 4. Ranking overview for QS WUR

Description	User is able to see QS WUR for different years
Precondition	User is logged in
Expected result	User sees correct data for each available year

Steps

1. Open the “Ranking overview” page
2. Select the “QS” tab
3. Select year 2022 from the “Year” select
4. Observe data in the table
5. Repeat steps 3 and 4 for different years

Test case 5. Ranking overview data filtering

Description	User is able to filter data by year and country
Precondition	User is logged in
Expected result	User filters data by multiple criteria

Steps

1. Open “Ranking overview” page
2. Select the “QS” tab
3. Select year 2022 from the “Year” select
4. Select Denmark country from the “Country” select
5. Observe data in the table change accordingly
6. Repeat steps 2, 3, 4 and 5 for different rankings, years and countries

Test case 6. Ranking analysis

Description	User is able to see the ranking analysis for the selected ranking, university, and years
Precondition	User is logged in
Expected result	User sees the ranking analysis in the form of multiple line charts

Steps

1. Open the “Ranking analysis” page
2. Select the “THE” tab
3. Select the report period years
4. Select the university
5. Observe line charts for the ranking indicators
6. Repeat steps 2, 3, 4 and 5 for different rankings, years and universities

Test case 7. Ranking comparison

Description	User is able to see the ranking comparison for the selected universities

Precondition	User is logged in
Expected result	User sees the ranking comparison in the form of multiple line charts

Steps

1. Open the “University comparison” page
2. Select the “THE” tab
3. Select the report period years
4. Select at least two universities to compare
5. Observe line charts for the ranking indicators
6. Repeat steps 2, 3, 4 and 5 for different rankings, years and universities

9.2.2 Test execution

Testing scenarios described in 9.2.1 were executed and helped to uncover the following errors:

1. the error message shown in case of invalid credentials is different than expected,
2. the country is missing for multiple universities and years in the “Ranking overview” page for QS WUR,
3. ordering of data in the “Ranking overview” page is incorrect due to empty or “n/a” values for ranking indicators,
4. the “Overall score” line chart in “Ranking analysis” and “Ranking comparison” pages is missing for some of the universities due to invalid data.

9.3 Usability testing

Usability testing is a method that allows the evaluation of a product by testing it with real users. It provides direct information about how people use computers and their problems with the concrete interface being tested.[41]

Usability testing of an application will be conducted in the following steps:

1. Participants selection.
2. Completion of the tasks by participants.
3. Completion of the questionnaire by participants.
4. Testing results analysis.

9.3.1 Preparation for the test

Participants selection

For the usability testing, five potential users have been selected:

- three academics,
- two prospective students who are looking for a university to study in.

Tasks given to the participants

Each user received the following tasks:

1. Log in.
2. Download the data from the QS WUR for the year 2020.

3. Download the data from the THE WUR for the year 2017.
4. Visualise the data for Aarhus university in THE WUR.
5. Visualise the data for the comparison of the University of Oxford and the University of Cambridge in THE WUR.

Usability testing questionnaire

1. Were you able to find all the information you were looking for?
2. How can we improve our website?
3. Did you encounter any difficulties while browsing?
4. How long did it take you to complete this task?
5. If you could change one feature, which one would it be?
6. How would you describe your overall experience with the application?

9.3.2 Testing results analysis

Both academics and students finished all tasks without noticeable prolongings. Academics outlined the following issues:

- After clicking "Enter" in the password field during the login, the "Login" button is not clicked. User has to click the button with the mouse.
- The years in the "Ranking overview" page are not ordered, so it is difficult to quickly choose the correct one.
- It is difficult to understand which ranking provider is selected "QS" or "THE" in the "Ranking overview" page.
- There are no names and description of the pages, so it is difficult to understand the purpose of each page without additional context provided.
- There is no option to save the state of the "Ranking analysis" and "Ranking comparison" pages.
- There is no option to download the report of the "Ranking analysis" and "Ranking comparison" pages.

Prospective students noted the following:

- After login, the “Profile icon” is not seen and it is impossible to control setting of the application,
- On the “Login” page, the navigation bar is not located the same as in other pages
- It is not clear, what are “THE” and “QS” and if these are buttons in the “Ranking overview”
- Button “Download” doesn’t contain any explanation of what is going to be downloaded and in which format
- Pages do not have titles and descriptions which would improve the speed of completing the tasks.

Advice from both groups can be summed up in the following three points:

- improve the navigation bar to be able to hide it and have better view on the data,
- add more tooltips and descriptions,
- add the source of information for all the pages.

Results of usability testing show that main points of improvement are application documentation and navigability.

Chapter 10

Conclusion

This chapter represents and summarises accomplished goals and sets the plan for the project's future.

10.1 Conclusion

The thesis's main goal was to develop a web application to help higher education institutions improve their position in the world university rankings by collecting and preparing ranking data for analysis. The application development included a collection of functional and non-functional requirements, selecting appropriate environment and software tools for implementation, and designing application architecture and user interface. The project also contained the research of the World University Rankings (WUR), analysing and selecting relevant rankings, studying available ranking data, and exploring existing solutions for data extraction. All the mentioned objectives were accomplished.

10.2 Future work

The result of the project is a starting point for the development of an application. The collection and preparation of data for analysis were successful.

However, missing indicators of some universities can cause errors in data visualisation, so it is crucial to find a way to overcome such issues.

The next step is finishing the role-based login implementation to set users' permissions and privileges based on their roles. Furthermore, the user interface has to be enhanced according to the designed wireframes.

The main plan for the future of the work is to expand the analytical part of the application, for example, creating more customisable dashboards and adding an overview of the comparison of the same university in different rankings.

Appendix A

Bibliography

[1] About QS. [online], Oct. 2022, [cit. 2022-11-17]. Available from: https://www.topuniversities.com/about-qs

[2] What is Scopus about? [online], Mar 2022, [cit. 2022-11-17]. Available from: https://service.elsevier.com/app/answers/detail/a_id/15100/supporthub/scopus/related/1/session/

[3] About ARWU. [online], Oct. 2022, [cit. 2022-11-17]. Available from: https://www.shanghairanking.com/

[4] ARWU methodology. [online], Oct. 2022, [cit. 2022-11-17]. Available from: https://www.shanghairanking.com/methodology/arwu/2022

[5] About THE. [online], Oct. 2022, [cit. 2022-11-17]. Available from: https://www.timeshighereducation.com/world-university-rankings/about-the-times-higher-education-world-university-rankings

[6] Webometrics. Ranking Web of Universities. Methodology. [online], Oct. 2022, [cit. 2022-11-17]. Available from: https://www.webometrics.info/en/Methodology

[7] O'Callaghan, C. QS World University Rankings: Sustainable Development Goals. [online], Oct. 2021, [cit. 2022-11-17]. Available from: https://www.topuniversities.com/university-rankings/world-university-rankings/sustainable-development-goals

[8] Bartoniček, V. Apify API. [online], Oct. 2022, [cit. 2022-11-17]. Available from: https://apify.com/vbartonicek/topuniversities-scraper

[9] Plans and pricing. [online], [cit. 2022-11-17]. Available from: https://apify.com/pricing

[10] QS World University Rankings 2021 Supplement. [online], Mar 2022, [cit. 2022-11-17]. Available from: https://www.topuniversities.com/student-info/qs-guides/qs-world-university-rankings-2021-supplement

[11] QS World University Rankings 2022 Result Tables (Excel). [online], Mar 2021, cit.[2022-11-17]. Available from: https://www.qs.com/portfolio-items/qs-world-university-rankings-2022-result-tables-excel/

[12] L, L. QS World University Rankings 2022. [online], Oct. 2021, [cit. 2022-11-17]. Available from: https://www.topuniversities.com/qs-world-university-rankings/methodology

[13] QS Quacquarelli Symonds. Faculty-Student Ratio – QS Quacquarelli Symonds. [online], July 2021, [cit. 2022-11-17]. Available from: https://support.qs.com/hc/en-gb/articles/4403961809554-International-Faculty-Ratio

[14] QS Quacquarelli Symonds. International Student Ratio – QS Quacquarelli Symonds. [online], Oct. 2021, [cit. 2022-11-17]. Available from: https://support.qs.com/hc/en-gb/articles/4403961727506-International-Student-Ratio

[15] QS Quacquarelli Symonds. International Student Ratio – QS Quacquarelli Symonds. [online], Oct. 2021, [cit. 2022-11-17]. Available from: https://support.qs.com/hc/en-gb/articles/4403961727506-International-Student-Ratio

[16] Beynon-Davies, P. Database Systems. 2004.

[17] Gosling, J.; Joy, B.; et al. *The Java Language Specification, Java SE 8 Edition*. Addison-Wesley Professional, first edition, 2014, ISBN 013390069X.

[18] Spring. Spring description. [online], Oct. 2022, [cit. 2022-11-17]. Available from: https://docs.spring.io/spring-framework/docs/5.0.0.RC2/spring-framework-reference/overview.html

[19] Hedley, J. Jsoup description. [online], Oct. 2022, [cit. 2022-11-17]. Available from: https://jsoup.org/

[20] Selenium. [online], Oct. 2022, [cit. 2022-11-17]. Available from: https://www.selenium.dev/documentation/

[21] The PostgreSQL Global Development Group. PostgeSQL. [online], Oct. 2022, [cit. 2022-11-17]. Available from: https://www.postgresql.org/about/

[22] The pgAdmin Development Team. pgAdmin. [online], Oct. 2022, [cit. 2022-11-17]. Available from: https://www.pgadmin.org/

[23] Jones, M. B.; Bradley, J.; et al. RFC7519. [online], May 2015, [cit. 2022-11-17]. Available from: https://tools.ietf.org/html/rfc7519

[24] JSON Web Token. [online], Oct. 2022, [cit. 2022-11-17]. Available from: https://jwt.io/introduction

[25] Typescript. [online], Oct. 2022, [cit. 2022-11-17]. Available from: https://www.typescriptlang.org/

[26] Getting Started – React. [online], Oct. 2022, [cit. 2022-11-17]. Available from: https://reactjs.org/docs/getting-started.html

[27] The React Component Library You always wanted. [online], [cit. 2022-11-17]. Available from: https://mui.com/

[28] Powell, B. *Mastering Microsoft power bi: Expert techniques for effective data analytics and business intelligence*. Packt Pub., 2018.

[29] Promise based HTTP client for the browser and node.js. [online], [cit. 2022-11-17]. Available from: https://axios-http.com/

[30] Eckerson, W. W. Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications. *Open Information Systems*, volume 10, no. 1, 1995.

[31] Crockford, D.; Morningstar, C. Standard ECMA-404 The JSON Data Interchange Syntax. 12 2017, doi:10.13140/RG.2.2.28181.14560.

[32] Fowler, M.; Rice, D.; et al. *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional, 2002.

[33] Microsoft Patterns & Practices Team. *Microsoft Application Architecture Guide*. USA: Microsoft Press, second edition, 2009, ISBN 073562710X.

[34] Pollack, M.; Gierke, O.; et al. *Spring Data*. Oreilly and Associate Series, O'Reilly Media, Incorporated, 2012, ISBN 9781449323950. Available from: https://books.google.cz/books?id=Mk4HcBE9WYcC

[35] Munzert, S.; Rubba, C.; et al. *Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining*. Wiley Publishing, first edition, 2014, ISBN 111883481X.

[36] Getting started. [online], Jan 2022, [cit. 2022-11-17]. Available from: https://www.selenium.dev/documentation/webdriver/getting_started/

[37] Mukulsingodiya@mukulsingodiya. Page Object Model (POM). [online], Jul 2019, [cit. 2022-11-17]. Available from: https://www.geeksforgeeks.org/page-object-model-pom/

[38] Window.localStorage - web apis: MDN. [online], [cit. 2022-11-17]. Available from: https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage

[39] React grid component - material UI. [online], [cit. 2022-11-17]. Available from: `https://mui.com/material-ui/react-grid/`

[40] Combine API and UI testing for confidence at every layer of your application. [online], [cit. 2022-11-17]. Available from: `https://smartbear.com/solutions/end-to-end-testing`

[41] Nielsen, J. *Usability Engineering*. Elsevier Science & Technology, 1994.

Appendix B

Acronyms

API Application Programming Interface. v, 16, 32, 33, 38, 40, 42

ARWU Academic Ranking of World Universities. v, 4, 5, 7

BI Business intelligence. vi, 34, 46

CRUD Create Read Update Delete. 38

CSS Cascading Style Sheets. 32, 42

CSV Comma-separated values. 16, 24

DOM Document Object Model. 32

ERD entity relationship diagram. 25, 27

HTML HyperText Markup Language. 16, 32, 41, 42, 46

HTTP Hypertext Transfer Protocol. 34, 36, 37, 45

HTTPS Hypertext Transfer Protocol Secure. 45

JDBC Java Database Connectivity. 32

JPA Java Persistence API. 32, 38, 39

JPQL Java Persistence Query Language. 39

JSON JavaScript Object Notation. 16, 33, 36

JWT JSON Web Token. 33, 37

MIT Massachusetts Institute of Technology. 32

MUI Material UI. vi, 34, 45, 46

ORDBMS object–relational database management system. 32

POJO Plain Old Java Objects. 42

QS Quacquarelli Symonds. 3, 4, 9, 11, 25, 37, 40

QS WUR QS World University Rankings. v, viii, ix, 1, 3, 4, 7, 9, 10, 12–16, 19, 20, 23, 26, 27, 40, 42, 50, 52

RFC Request for Comments. 33

RSS Really Simple Syndication. 16

SDG Sustainable Development Goal. v, viii, ix, 10, 12–14

THE Times Higher Education. v, 5, 6, 25, 27, 37, 40

THE WUR THE World University Rankings. viii, ix, 6, 7, 23, 26, 27, 42, 49

URL Uniform Resource Locator. 32, 41, 43

XML Extensible Markup Language. 16