TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Computer Systems

Senanayaka Arachchige Yasith Hasantha Ariyasena 194334IASM

BIBLIOGRAPHIC DATA MINING FROM ESTONIAN RESEARCH INFORMATION SYSTEM

Masters's Thesis

Supervisor: Aleksei Tepljakov
Ph.D.

Tallinn 2021

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutisüsteemide instituut

Senanayaka Arachchige Yasith Hasantha Ariyasena 194334IASM

BIBLIOGRAAFILISTE ANDMETE KAEVANDAMINE EESTI TEADUSUURINGUTE INFOSÜSTEEMI ALUSEL

magistritöö

Juhendaja: Aleksei Tepljakov
Ph.D.

Tallinn 2021

Author's declaration of originality

I hereby certify that I am the sole author of this thesis and this thesis has not been presented for examination or submitted for defence anywhere else. All used materials, references to the literature and work of others have been cited.

Author: Senanayaka Arachchige Yasith Hasantha Ariyasena

10.05.2021

Abstract

Bibliographic metadata presents a valuable resource since it is essential in the process of analyzing research output and relations. However, extracting metadata becomes a challenge since the data is not available in specific formats. This research is about developing a system for bibliographic data mining of research papers from the research publications which are published in the Estonian research information system (ETIS). Since most of the publications in ETIS do not have Digital object identifiers (DOI), one of the main objectives of this project is to generate bibliographic metadata in BibTeX format for every publication that can be used in text editors such as LaTeX to generate bibliographies quickly.

GROBID and Crossref tools are used further to parse the PDF files of the research papers and analyze the references to generate the BibTeX. The system is designed to integrate different functions with a web application. Output data is validated and tested with external tools to ensure that the system works without any issues.

The thesis is written in English and contains 79 pages of text, 6 chapters, 19 figures and 8 tables.

Nomenclature

DOI	*Digital Object Identifier*
ETIS	*Estonian Research Information System*
API	*Application Programming Interface*
CRF	*Conditional Random Fields*
GUI	*Graphical User Interface*
PDF	*Portable Document Format*
TEI	*Text Encoding Initiative*
XML	*Extensible Markup Language*
REST	*Representational State Transfer*
ORCID	*Open Researcher and Contributor ID*
MVC	*Model–view–controller*
OS	*Operating System*
JSON	*JavaScript Object Notation*
CSV	*Comma-Separated Values*
URL	*Uniform Resource Locator*
GUID	*Globally Unique Identifier*
HTML	*Hypertext Markup Language*
DOM	*Document Object Model*
HTTP	*Hypertext Transfer Protocol*
AJAX	*Asynchronous JavaScript And XML*

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Bibliographic Data

The bibliography is all the list of sources that we use in our publications. These publications can be conference proceedings, newspaper articles and journals, patents, periodicals, reports, and books. Therefore, the bibliography data structure can be different from publication to publication. However, most of the time, bibliography data should include the author's names, title, published year, publisher, DOI reference (if the publication has been assigned the latter). The importance of bibliographic metadata mining is broadly discussing since the number of research publications increased every year [1], and because of that, it is a challenging and time-consuming task for the researcher to find the publication's information that can be used for their research works.

Data mining can be interpreted as a process of parsing a large amount of raw data and extracting valuable data for use effectively. However, data mining is not a new thing to the modern digital era, and the concept is using for over the decades, but now it has become a greater focus with modern technology and practices are used in different areas. Since the increase of research publications, solutions for bibliographic data management have been introduced from time to time, and software, online tools can be found. BibTeX is a popular reference management tool used to format the bibliographic data and be integrated with type-setting software like LaTeX. This research mainly focuses on bibliographic data

mining from publications published in the Estonian research information system. The system is developed so that bibliography data can be extracted and parsed to present them in BibTeX format. Also, the data is stored and maintained in a digital file library that can be used for future projects.

1.2 Research Questions and Gaps

After the preliminary research, it was found out that the DOI identifier assigned to each publication can generate the BibTeX files using external tools. Therefore, finding DOI identifiers and parse them is one of the goals of this project. However, not all the ETIS publications registered with a DOI identifier. Therefore, it is an additional work for researchers to cite those publications in their research documents, and it might be a problem if there is a large set of references that need to cite. On the other hand, a researcher also needs to explore the bibliographies cited in those research papers. Since all the publication does not have a DOI identifier, researchers must explore the PDF files, and use other tools to generates the citation data. It will give more value to the researchers if the citation data extraction can be done from a single place. Also, this project will give visibility to publications that need to register with a DOI identifier.

EITS provides open APIs for developers to extract the data of the publications listed on its website. After the initial testing of the API responses, most publications do not have source links to download the papers as PDF files. As mentioned previously, it is essential to have PDF files to mining the bibliography data and to maintain the digital library. Therefore, the data extraction cannot be entirely dependent on the API, and other mechanisms also considered downloading the PDF files, such as web scraping methods.

If a publication has a DOI identifier, it is possible to generate BibTeX data using BibTeX generating software and online tools. Nevertheless, to generate such data in a centralized system, such a service must be integrated with the system to parses the DOI and return the BibTeX data. Crossref [2], DoiOrg [3] are services that provide API access to developers to build their solutions. If a publication does not have a DOI identifier, the system should use ETIS API data for generating the BibTeX file. It might not be possible to generate

the fully structured BibTeX formats in such situations as the required data may not be available.

As mentioned before, it is required to process the PDF files if the papers are publicly available for downloads or accessible through the TalTech network since the solution will be published inside the university network. Therefore, the system is limited to the users inside the university network. There are metadata extraction tools found to process the PDF files, but there is no comprehensive evaluation for those heuristics tools since their data mining approaches and methods are different. Recent researchers' different tools evaluated, and the GROBID [4] highlights as high rank among other tools. It uses the conditional random fields (CRF) approach to extract the metadata, but still, the data might not be 100% accurate, and exceptions might occur as the research publications are structured in different formats [5]. So, the extracted data need to be appropriately validated to minimize the errors.

1.3 Project Goals

The main objective of this thesis is to implement a system for bibliographic data mining from ETIS. Since this is software-related research, the following core modules have been identified as the project's main goals, and the developed system should handle these.

Extracting research publications from the ETIS system using its open API is one of the core functions of this project. This module should handle the ETIS API calls, validate, map with data transfer objects, and pass the response data to other modules. This module should also filter out the data by checking the DOI identifier and other required fields to generate the BibTeX data.

The developed system should generate and save the BibTeX data for all the ETIS publications and their bibliographies, regardless of the DOI availability. This function should integrate with Crossref APIs and GROBID web services to parse the DOI identifiers and PDF files. Available PDF files should download and save on a local server using a web scraping algorithm since the ETIS API does not return downloadable links for publica-

tions. Therefore, setup GROBID web services will be another important goal and investigate how these services can integrate into the system to handle the functions. GROBID services mainly use for extracting the metadata from the PDF files. These micro-services need to publish as a separate solution by decoupling from the primary system, and it will allow using these services for other researchers.

A web application needs to be implemented by integrating the above modules. The web app should have an interactive UI for end-users to retrieve the research publications, view BibTeX tags, view bibliography information, and download PDF files. Since the system is integrated by modules that use different tools and technologies, every scenario and integration must be tested and appropriately validated to ensure interoperability between the modules. Therefore, apart from the main goals, the following sub tasks must achieved for project success:

- Test and validate open API data with actual data published on the `https://www.etis.ee/` website.
- Test and validate BiBTeX tags with actual data.
- Validate the accuracy of the output data from the GROBID micro-services.
- Test interoperability of the modules.

1.4 Thesis Outline

The thesis is structured as follows,

Chapter 2 gives the literature review of the thesis. The reader can understand the Estonian research information system, data mining technologies used in the project, and an overview of the BibTeX format.

In Chapter 3, the system design and the methodologies are explained using the system diagrams, GUI designs. Also, the API specifications are discussed here. The reader gets knowledge and familiarity with how the system integrates and data flow works with

different modules.

Chapter 4 explains technical aspects of the project, development methodologies, and how the core modules developed.

In Chapter 5 and Chapter 6, test results of the implementation and post developments are explained.

Chapter 2

Literature Review

2.1 Estonian Research Information System

The Estonian Research Information System is a centralized platform for researchers, institutions, and various research outputs in Estonia. Furthermore, it is a national information system, also a channel for submitting, reviewing research publications. Researchers can use the ETIS website to access publicly available publications. Therefore, its an informative tool for researchers and R&D institutions. Several parties use this as their internal research information system as well. It is developed by the Estonian Ministry of Education and Research and operated by the Estonian Research Council [6].

As this research project implemented for extracting bibliography data from the ETIS publication, preliminary research was carried out for understanding the data structures of the publications, search parameters, and data types in the ETIS.ee website and the web services. A prototype is developed to gather the following statistics (as of 2021-04-15),

Table 2.1: ETIS publication statistics

CONTENT TYPE	COUNT
Publications	252,101
Profiles	13,935
Publications without DOI	207,476
Publications with DOI	44,625

Out of 252,101 publications, 82.2% of publications do not have a DOI identifier. Therefore, citing such publications with an automation solution gives a greater value to the end-users.

2.2 Review of Bibliography Data Mining Technologies

Every year, the ever increasing number of publications made a high demand to develop tools to extract accurate data from unstructured documents and maintain digital libraries in a structured way [7]. Therefore, several research projects are carried out to identify and introduce the best approaches and tools to extract the bibliographic metadata from scientific publications.

2.2.1 GROBID

Parsing PDF files in ETIS is one of the goals, as stated before. Therefore, finding a suitable tool or a library is essential for this project, and it must be used as micro-services, integrate with other interfaces. Several types of PDF file extraction software and tools are identified, such as ParsCit [8], Cermine [9], and GROBID.

GROBID is one of the open-source tools [10] used in this project as this tool has several methods to identify and extract the metadata and can be integrated easily into other interfaces. GROBID is a JAVA-based tool, and the latest version supports only run-on Linux and Mac Operating systems [11]. Evaluations done by the researchers identified that GROBID is a better solution for the metadata extraction without using any external tools and very effective compared to other open-source tools [12]. Joseph Boyd evaluated GROBID to use metadata mining from high-energy physics research papers in `https://inspirehep.net/` [13]. Few other projects also used GROBID libraries for metadata extraction; Mark and Joeran used GROBID for their research to parse the citations and retraining to get meaningful data as output [14]. Also, it is used for large-scale research like "The COVID-19 Open Research Dataset" to parse the PDF files to TEI XML format [15].

2.2.2 Crossref

Since 20% of the publications in ETIS possessed a DOI identifier, parse DOI identifiers with an existing tool is the ideal way to generate the BibTeX data. Crossref is one of the agencies that provide a broad range of services that specifically support scholarly content worldwide [16]. It was founded in 2000 as a not-for-profit memberships association for scholarly publishers [17]. Crossref mainly worked with a DOI-based system to identify academic content and cross-publisher reference linking to the original location of a publication [18]. Crossref provides open APIs for access metadata from publicly available research contents. Besides the BibTeX, it also provides funding data, license information, full-text links, ORCID Ids, abstracts, and bibliography data [19]. Therefore, the developed system is integrated with Crossref REST APIs to generate the BibTeX and enhance the system features with other available options such as full-text links, bibliography data.

2.3 Review of BibTex Data

BibTeX is a reference management tool, can be used in typesetting systems like LaTeX. It has a specific format with file extension as .bib [20]. BibTeX files make life easier for users by automating extensive reference lists, such as a Ph.D. thesis or research papers. Since it has a stable and straightforward data structure that is easy to edit, BibTeX has become popular and standard to generate bibliographic data. Also, some external tools are developed because the advantage of using BibTeX is high [21]. Therefore, BibTeX is used to generate a bibliography for publications that do not have DOI identifiers. Sample BibTeX data format is given in Figure 2.1.

It has three parts, Entry type, Cite key and Fields. In the latest version of the BibTeX, there are 14 entry types. The cite key is a unique identification for the entry, which combines letters and digits. Lastly, the bibliographic data store in the fields section, and the corresponding filed type and values are decided by the entry type. Entry types and fields are discussed more in section 4.2.3.

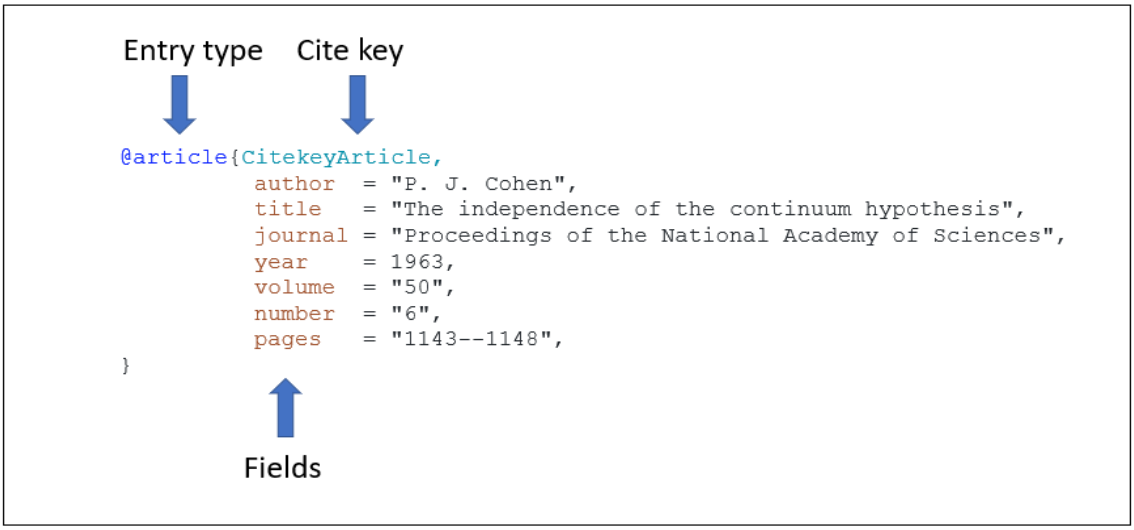


Figure 2.1: Example of a BibTeX entry

Additionally, other modern packages can be highlighted when it comes to bibliography management, such as BibLaTeX. In BibLaTeX, The LaTeX macros handle the formatting of the bibliography and, it uses its data program called ”biber” to process the data [22]. BibLateX gives several advantages when compared with BibTeX. Such as support for full Unicode, remote data sources, highly customizable bibliography labels [23] . Therefore, from the sustainability perspective, BibTeX is an aging system for typesetting bibliographies, and BibLaTeX should be used instead, where applicable. However, due to compatibility considerations—many scientific conferences and journals still lack BibLaTeX support— in this work, we use BibTeX. If needed, it is relatively easy to convert between the two formats; hence this is not a significant obstacle for the present work.

Chapter 3

Methodology and Design

3.1 System Overview

The system is designed in a way that should achieve the goals stated in Section 1.3. The system consists of different tools and technologies and core system design to develop by using Microsoft .Net Core framework. End-user is interacted with a Web Application to search and view the data. The first step is to retrieve the publication information using ETIS API, and for other tasks, the following services and methods are used in the pipeline. Figure 3.1 illustrates with a flowchart that how the system works.

- Identify the DOI — Implemented algorithm to filter out the DOI identifiers from the API response.
- Generates BibTeX for publication with DOI — Call Crossref REST client services.
- Generates BibTeX for publications without DOI — Implemented algorithm to format the BibTeX by ETIS API response.
- Download PDF files
  - If PDF available in ETIS web page — Implemented a web scraping algorithm to get the downloadable link.

  - If PDF file not available in ETIS — Check the publication has DOI; if yes, then call the Crossref REST services to get the downloadable link.

- Extract bibliographic data in a publication.

  - If DOI available — Use Crossref REST services.
  - If DOI not available — Check the availability of the PDF, and call GROBID web services to extract the data from PDF file.

- Generates BIbTeX for bibliographies in publications.

  - If the publication's DOI available — Use Crossref REST services.
  - If the publication's DOI not available — Check the availability of the PDF, and call GROBID web services to parse the PDF file.

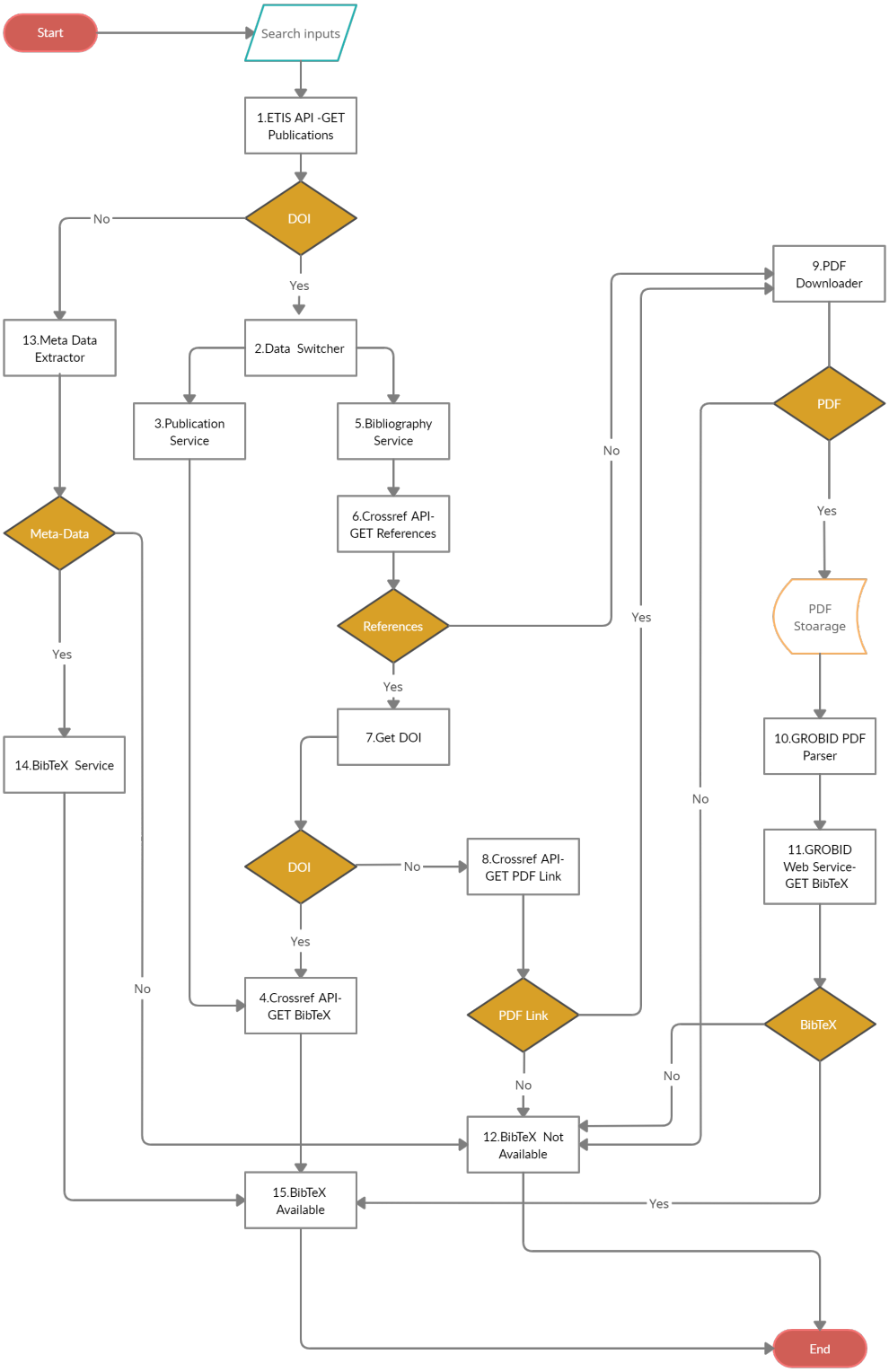


Figure 3.1: System flow chart

The flow starts with the user search inputs by passing to the ETIS API call. Since the goal is to find the relevant BibTeX data for publications and their bibliographies, the main flow consists of different approaches and techniques. Items 1,2,3, and 4 show the first approach to finding DOI identifiers and generating the BibTeX by Crossref services. Item 2 is a data switcher that divides the flow into two paths, as shown in items no 3 and 5, publication and bibliography services.

Bibliography service (item 5) mainly works for generating BibTeX for publication's bibliography lists. Since items 1,2,5, and 6 in the flow have the DOI number, it first checks the availability of the bibliography list with Crossref and extracts the DOI identifiers for each bibliography in the retrieved list. This flow finally connects to item 4 to get the BibTeX for DOI available bibliographies.

The second approach illustrates the flow from items 8 to 11, bibliographic data mining from PDF files of the publications. For the Bibliography service (item 5), in two scenarios, it routes to PDF downloader to search the BibTeX for bibliography list from publication's PDF file. scenario 1: if it does not found any bibliography from Crossref; scenario 2: if the Crossref does not have any DOI identifier for the bibliography records.

The above two cases check the PDF files and parse with GROBID services to extract the bibliography list and output the BibTeX. The third approach generates the BibTex with ETIS meta-data listed with items 1,13,14, and 15 in the flow. Item 13 extracts the meta-data from ETIS API contents and, if it is available, then passes to the BibTeX service to generate and arrange the BibTeX formats.

The flow finally shows the BibTeX for publication and its bibliographic list. If nothing has been found throughout the flow, it will show up as "N/A" content. Also, during the flow, the available PDF files will be stored in PDF storage.

3.2 System Architecture

The system is integrated with different components. The data flow starts with the user inputs in the front end, designed as a Web application. The core system is designed to implement in .Net Core MVC architecture. The front end has two components, ETIS profiles and ETIS publications. ETIS profile section provides the user to retrieve the publication for a person and generates the BibTeX. Also, users can get individual publications by the ETIS publication module. The search inputs pass to the search engine, and it triggers the ETIS API client service. Crossref and GROBID external tools use in the system for parse DOI and PDF files. An algorithm is designed for bibliographic data mining in each publication using both GROBID and Crossref as a hybrid solution that helpful for increase the data availability for the end-user. A web scraping method uses to extract downloadable links for PDF files since the API does not return the link. This module is integrated with GROBID to extract the reference list from the PDF files. The Latest version of GROBID services should be hosted in a Linux OS, as it does not support Windows [24]. Therefore, the core system will connect the GROBID with a API client service. Redis cache is designed to be implemented to increase performance. The core system is connected to a local file server to store the research papers and BibTeX files. Figure 3.2 illustrates the system diagram and how data flows between components.

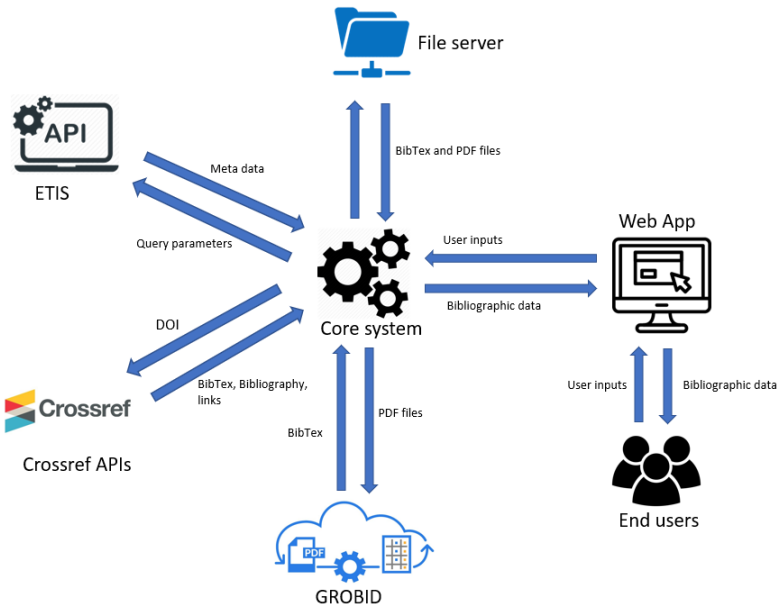


Figure 3.2: System architecture

3.3 API Architecture and Definitions

The solution has three API client services: ETIS, GROBID, and Crossref, designed to work asynchronously. The system is using .Net Core MVC architecture that flows the data from external APIs to View. Moreover, the Redis cache integrates here to integrate with the REST services to fasten the response. Figure 3.3 illustrates how APIs connect to the core architecture.

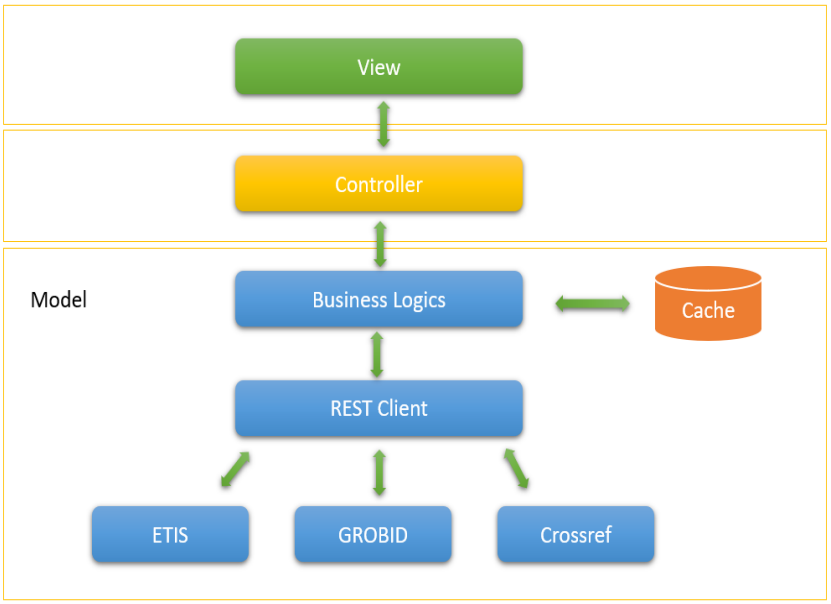


Figure 3.3: API architecture

The View is designed to implement using Bootstrap controls and client events with jQuery libraries. Every user interaction is asynchronously passed to the Controller to handle, work with the Model and finally render the output with View [25]. All the business logic and operations are implemented in Model, so it represents the application's state. The Controller manipulates these Model states and passes them to the View as ViewModel. The business logic contains the REST services connected to ETIS, GROBID, and Crossref, external API services. the Redis cache connects to the business logic services that store the frequently use data.

3.3.1 ETIS APIs

ETIS provides open APIs for developers to work with the publicly available content, and it does not require any user agreements to use the APIs in projects. There are 14 APIs available for different services. The response data can be returned as JSON, XML, CSV, and EXCEL. Also, a paging mechanism is enabled for each GET call. Below are API URLs for different services,

- Base URL: https://www.etis.ee:7443/api/
- Query parameters: /getitems?Format=json&Take=5&Skip=0

Table 3.1: ETIS APIs

SERVICE	CALL
Institution	/institution/
CV Est	/cvest/
CV Eng	/cveng/
Publication	/publication/
Mentorship	/mentorship/
Industrial Property	/industrialproperty/
Cooperation Offer	/cooperationoffer/
Product Service	/productservice/
Collection	/collection/
Scientific Equipment	/scientificequipment/
Classifier	/classifier/
Project	/project/
Excluded	/excluded/
Program	/programme/

Out of those, following APIs are used in this project,

To retrieve the research publications,

- URL: `https://www.etis.ee:2346/api/publication/getitems?Format=json&Take=5&Skip=0`

- Query parameters- take, skip, PersonId, PersonName, PersonEstonianIdCode, InstitutionId, InstitutionName, InstitutionRegNo, Title, PublishingYearMin, PublishingYearMax, TypeName, EditionTitle, ClassificationCode, ClassificationName, PublicationStatus (1-is published,2-is being published), SearchWord, DateCreated, DateModified, DateArchived, SearchType (1-begining, 2-fuzzy, 3-precise)
- Output – response data includes the details of publication, author, institution, projects, DOI, ISSN, ISBN, publisher, published year etc. (see Appendix B.2)

To retrieve the profiles,

- URL: `https://www.etis.ee:7443/api/cveng/getitems?Format=json&SearchType=2&Take=5&Skip=0`
- Query parameters-take, skip, PersonId, PersonName, PersonEstonianIdCode, InstitutionId, InstitutionName, OccupationStatus, ResearchAreaCode, ResearchAreaName, SearchWord, DateCreated, DateModified, DateArchived, SearchType (1-begining,2-fuzzy, 3-precise)
- Output : response data includes the details of person,publication, education, projects, etc.(see Appendix B.1)

To retrieve the classifications,

- URL: `https://www.etis.ee:7443/api/classifier/getitems?Format=json&SearchType=2&Take=5&Skip=0`
- Query parameters: take, skip, Code, Name, ClassifierName, InstitutionId, DateCreated, DateModified, DateArchived, SearchType (1-beginning,2-fuzzy, 3-precise)
- Output : response data includes the classification name, code, and id.(see Appendix C)

3.3.2 GROBID Web Serives

The latest version of the GROBID supports only Linux and Mac OS as this is implemented mainly with JAVA. It allows developers to use their demo server `https://cloud.science-miner.com/grobid/` which provides UI console and web services for tests and developments. Since this project implements on Microsoft technologies, we used the GROBID demo server for initial implementation and tests; this thesis explained the services with the test link. However, the live system will be published in a Linux environment, and the test link will be replaced in the production environment.

GROBID provides 14 webservices to parse the PDF files for different requirements [26].

Table 3.2: GROBID APIs

SERVICE	CALL
Extract the header	/api/processHeaderDocument
Convert the fully document into TEI XML	/api/processFulltextDocument
Extract bibliographical references	/api/processReferences
Parse a raw date string	/api/processDate
Parse header section in pdf and return author names	/api/processHeaderNames
Parse a raw name from a bibliographical list	/api/processCitationNames
Parse a raw affiliation	/api/processAffiliations
Parse a raw bibliographical list	/api/processCitation
Return JSON annotations	/api/referenceAnnotations
Augment PDF with annotations	/api/annotatePDF
Extract the patent and non-patent citations in UTF-8 text	/api/processCitationPatentTXT
Extract patent and non-patent citations encoded in ST.36	/api/processCitationPatentST36
Extract and parse the patent and non-patent citations	/api/processCitationPatentPDF
Excluded	/api/citationPatentAnnotations

In this project, we mainly need to extract the bibliography references list from each available PDF. Therefore we used the processReferences API service that capable of extract and converted all the references into TEI XML or BibTeX format.

- URL: `https://cloud.science-miner.com/grobid/api/processReferences`

- Method: POST/PUT
- Input: PDF file
- Request media type: multipart/form-data
- Response type: application/xml
- Response data: includes the data available in references such as author, title, year, publisher etc.

To get the output in BibTeX format, header key and a value needed to be added to the API call as Accept: application/x-bibtex.

3.3.3 Crossref APIs

Crossref provides a rich API platform for developers to work with metadata of scholarly content. Following REST API calls are used in our project to retrieve and format the data output.

To get all the details by DOI identifier,

- URL: `https://api.crossref.org/works/{doi}`
- Method: GET
- Input: DOI identifier
- Response type: application/json
- Response data: includes the data available for a passed DOI such as author, title, year, publisher, abstract, references, reference link to original publication. (see Appendix B.3)

To get the publication's BibTeX data,

- UR: `https://api.crossref.org/works/{doi}/transform/application/x-bibtex`

- Method: GET
- Input: DOI identifier
- Response type: application/x-bibtex
- Response data: BibTeX data for the passed DOI identifier

To transform BibTeX to bibliographic text,

- UR: `https://api.crossref.org/works/{doi}/transform/text/bibliography`
- Method: GET
- Input: DOI identifier
- Response type: application/text
- Response data: Bibliographic entry

3.4 Design of the Graphical User Interface

3.4.1 ETIS Profile Search Interface

This UI is designed for users to search for a particular author and get their publications. UI consist of a description of the page purpose followed by the search controls. There is one generic search text box and three other advanced filters if the user wants to narrow the search quarry. Users can select other search filters by clicking the ”Advance Search” link button.

A loading panel will be indicated to the user till the results are returned to the View.

- Generic search: will match any text in the profile data array and return the values.
- Person name: the person’s name and return only records that match the author’s name in the profile array.ex: James Hamilton Love.
- ETIS classification: classification code of the publication ex: 3.1
- Institution name: ex: Tallinn University of Technology

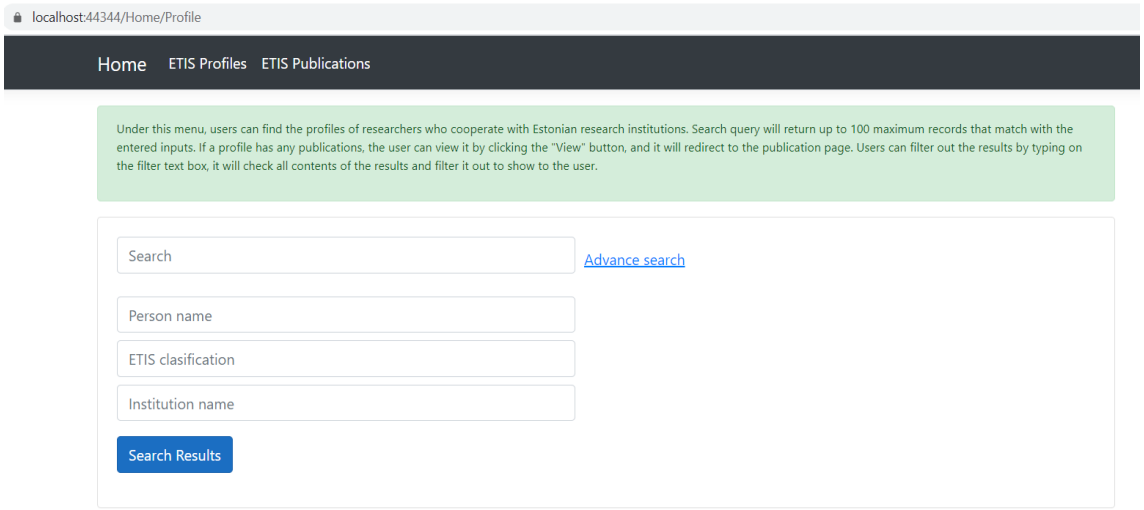


Figure 3.4: Search profile interface

3.4.2 Display Profile Results

After the user searched the profiles, the results display in a table view. The results consist of the following fields.

- Filter text box- this provides the user to filter out the result from the table. The input will match every content from the table data.
- Name: name of the author
- Email: email address of the author. If not available, it will show as “N/A.”
- No of publication: count of the publication that author has in ETIS
- View publications (action buttons): this button will redirect the user to the ETIS Publication page to view all the publication information. If there are no publications available, the button will be disabled, and text will be displayed as “N/A.”

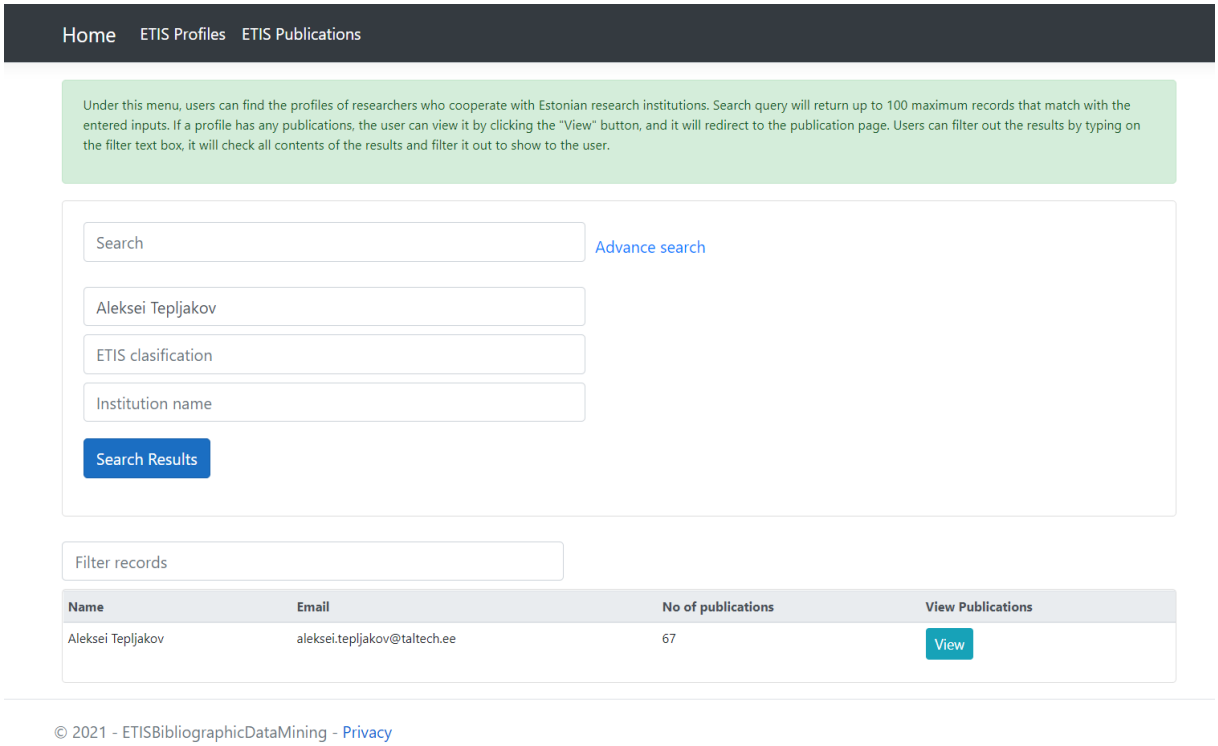


Figure 3.5: Example of profile search results interface

3.4.3 ETIS Publication Search Interface

This UI is designed for users to search a particular publication and get its details. UI consist of a description of the page purpose followed by the search controls. There is one generic search text box and four other advanced filters to narrow the search quarry. Users can select other search filters by clicking the ”Advance Search” link button.

- Generic search: will match any text in the publication data array and return the values.
- Person name: the person’s name and will return only records match with author name in publication array. ex: James Hamilton Love
- Title- the title of a publication
- ETIS classification: classification code of the publication ex: 3.1
- Institution name: ex: Tallinn university of technology

A loading panel will be indicated to the user till results are returned to the view.

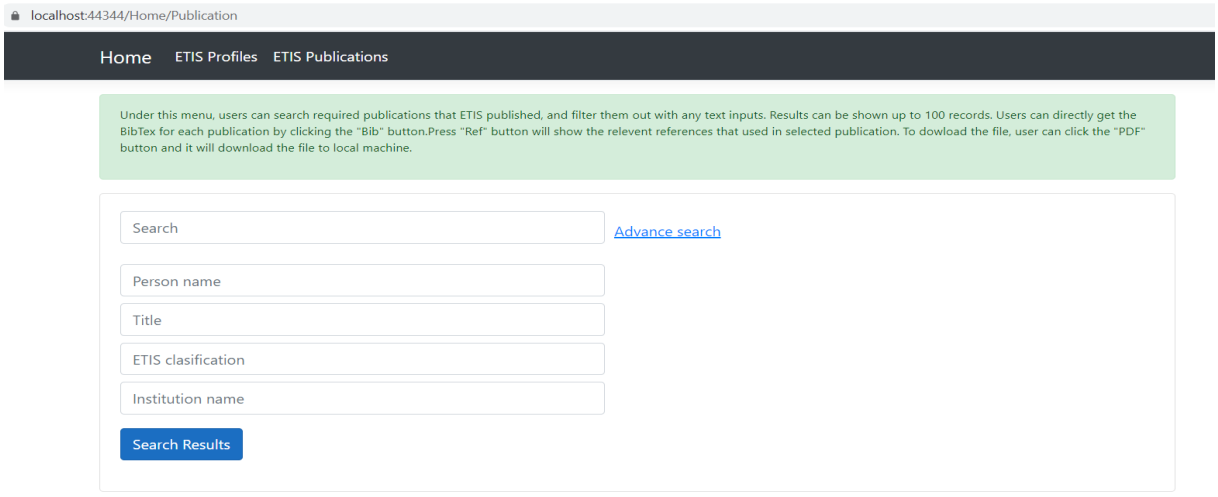


Figure 3.6: Search publication interface

3.4.4 Display Publication Results

After the user searched the publication, the results will be displayed in a table view. The results consist of the following fields.

- Filter text box: this provides the user to filter out the result from the table. The input will match every content from the table data.
- Year- published year of the publication.
- Classification code: classification code of the publication that comes from the ETIS.
- DOI: DOI identifier of the publication. If not available will show as “N/A.”
- Publication- short description of the publication
- Actions (action buttons): there are three action buttons.

  Bib: opens a modal that contains BibTeX data of the publication.

  Ref: opens a modal that contains reference list of the publication.

  Pdf: download the PDF file if available.

A button will be disabled if the relevant data not found for the action.

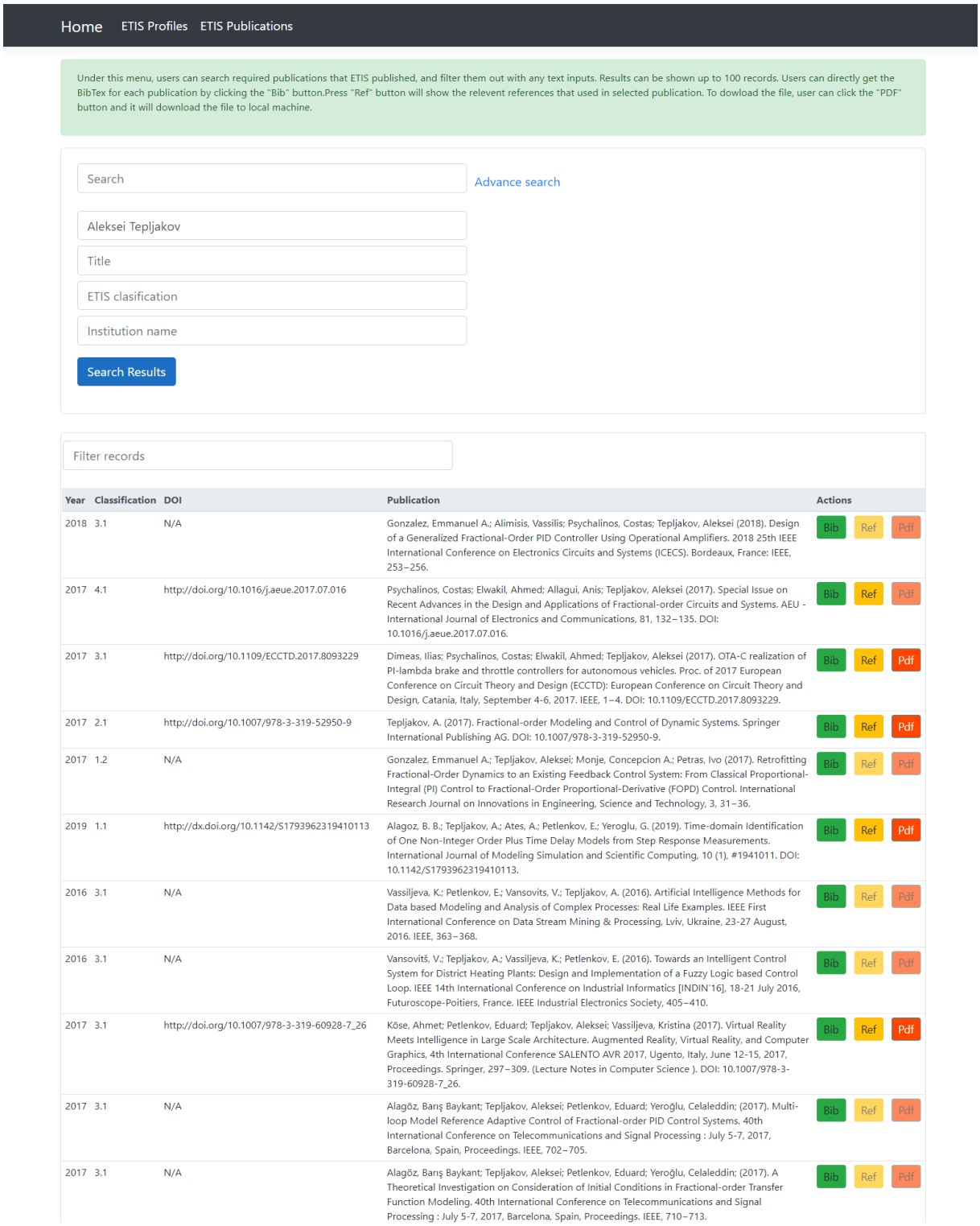


Figure 3.7: Example of publication results interface

3.4.5 Display BibTeX

When users click the “Bib” button in publication results, a popup modal will be opened containing the BibTeX data. This modal consists of three buttons, copy the content to the clipboard, save the content to the local machine as a .bib file, and the close button to exit from the modal. There is a loading view if the server takes time to process the data.

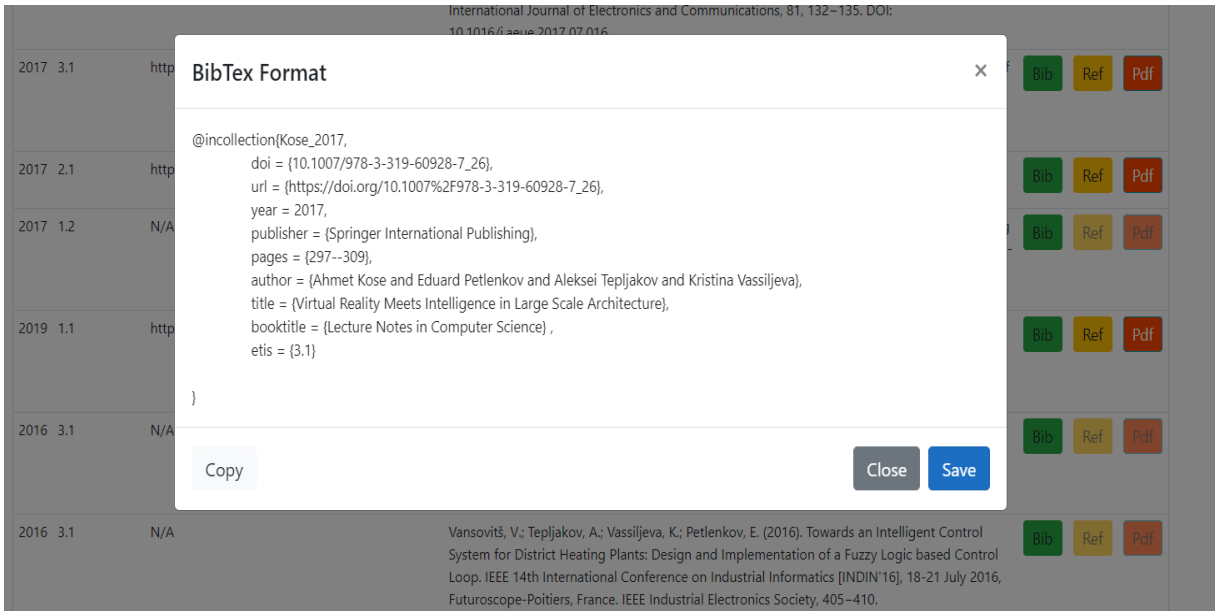


Figure 3.8: Example of BibTeX data modal interface

3.4.6 Display Bibliographic List

Reference modal is designed to open and show the list of references that have for a particular publication. This modal will be open when the user clicks the “Ref” button in the table row action button. The list of references will be displayed as a collapse list that user can expand each row to view the relevant BibTeX data.

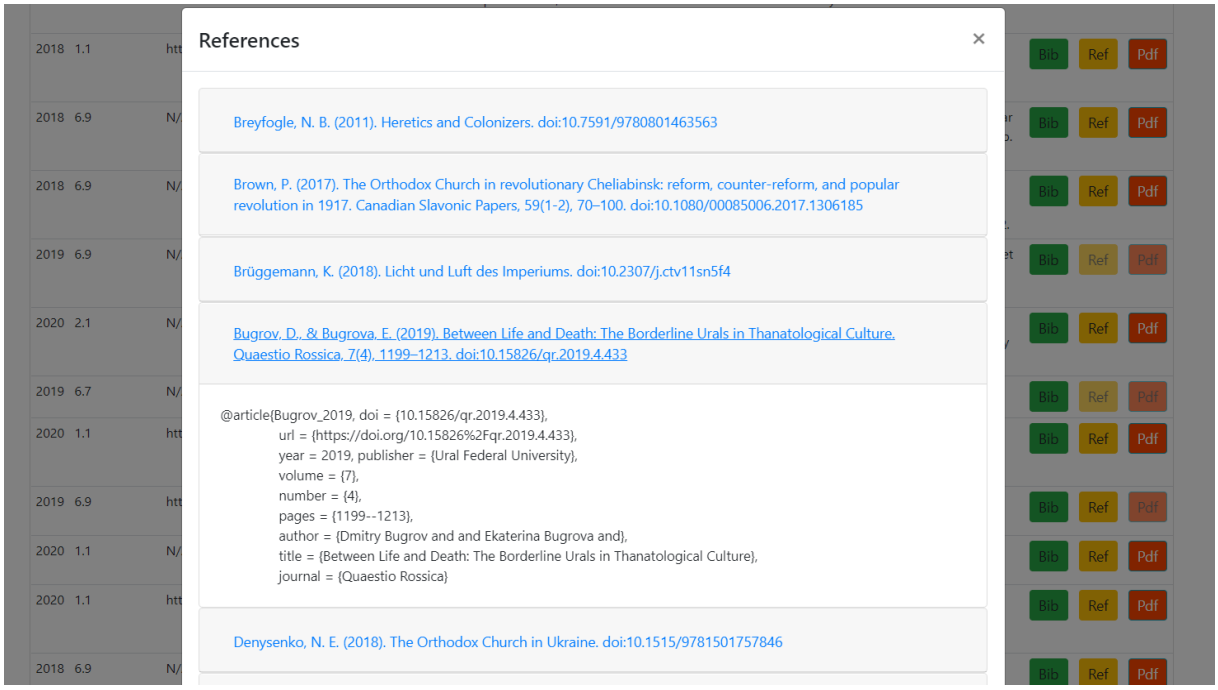


Figure 3.9: Example of bibliographic list interface

3.4.7 Display Publications by Profile

This interface is designed for viewing the publication that comes from the Profile search page. The required parameters (profile GUID, publication count) pass in the URL to capture by the code. This page is a bit different from the standard publication page, as this showed two buttons for downloading all the BibTeX and PDF files of the selected author. And these files are download as a .zip file.

Figure 3.10: Example of publications by profile

Chapter 4

System Development

4.1 Development Environment and Tools

Agile software development approach [27] used for system implementation and mainly developed using Microsoft technologies. Microsoft Visual Studio 2019 is used for the develop the application, using ASP.Net Core framework 3.1 with C# language. As explained in the design phase, the system is implemented in MVC architecture, extended with jQuery and Bootstrap components. Front-end GUI is implemented using HTML. jQuery and JavaScript use for client-side programming. Apart from the standard Microsoft libraries, few NuGet packages are used, such as HTMLAgilityPack for parsing HTML pages and DOM [28], RestSharp to make synchronous and asynchronous calls to external API resources over HTTP [29].

Since the system is used external services, the performance and the user experience need to be considered. Therefore we used Redis as a caching tool. Also, we do not use any relational databases now and planned to work everything on the fly. So, Redis is the ideal tool for replacing the database functions as its non-relational database with high performance distributed caching engine, provides a low latency key-value pair caching mechanism [30]. Redis is built and test to run on Linux and Mac OS, listening on incoming connections on a port such as 6379. However, fortunately, Microsoft provides a port to run Redis on Windows machines. To communicate .Net core application with the

Redis server, we used Microsoft.Extensions.Caching.Redis NuGet package [31, 32, 33].

Git is used for application source code maintains and version control. The final solution will be available at GitHub.

4.2 Core modules

4.2.1 Search Engines

Since the Web App has two main modules as ETIS profiles and ETIS publications, two search engines are developed.

SEARCH PROFILES

This search engine accepts four search filters.

- Generic text: this will find any match in the profile list.
- Person name: this will find only the person's name in the profile list.
- Classification code: this will find the relevant research classification area in the profile list.
- Institute name: this will find the profiles by person institution.

The search engine works asynchronously to finds the records and return the data for a relevant match. The maximum record count is 100 per search. A Loading indicator on the page is implemented to notify the user.

The search button invokes the jQuery button click function `$('#btnSearch').click()`

```
$('#btnSearch').click(function () {
        var personName = $("#txtName").val();
        var instituteId = $.trim($('#txtInstitue').val());
```

```
            var clasificationCode = $.trim($('#txtClasfication').val())
                ;
            var searchText = $.trim($('#txtSearch').val());
            var take = 100;
            var skip = 0;
            $("#loadingPanelTable").show();
            $.ajax({
                url: '/Home/GetEtisProfiles',
                type: 'GET',
                dataType: 'json',
                cache: true,
                async: true,
                data: { personName, instituteId, clasificationCode,
                    searchText, take, skip },
                success: function (data) {
                }
            });
```

The function will do an Ajax call to the Home Controller method `GetEtisProfiles()`, which invokes the ETIS API client services in the `IEtisApiClientService` interface.

```
public IActionResult GetEtisProfiles(string personName, string
   instituteId,string clasificationCode, string searchText, int take,
   int skip)
```

There are two methods implemented in `IEtisApiClientService` interface.

```
public interface IEtisApiClientService
{
    Task<List<ProfileModel>> GetProfiles(string personName,
                                         string instituteId,
                                         string clasificationCode,
                                         string searchText,
                                         int take,int skip);

    Task<List<PublicationModel>> GetPublications(string searchText,
                                                 string personId,
```

```
                                                    string publicationId,
                                                    int take,int skip);
}
```

`GetProfiles()` – make an async HTTP request to the ETIS API and return the corresponding profile data in JSON format. Necessary data fields are captured when deserializing the JSON into data Model as follow,

- Guid: unique identifier for the record
- PersonGuid: profile identification id
- PersonName: name of the person, includes both first and last names.
- Email: email address of the person
- Publications: list of publications that person has.

Returned data Model is bind to the HTML table in View. So, the method hierarchy as below,

```
$('#btnSearch').click()->GetEtisProfiles()->GetProfiles()
```

SEARCH PUBLICATIONS

This search engine accepts five search filters.

- Generic text: this will find any match in the publication list.
- Person name: this will find only the person's names in the publication list.
- Title: this will find the title of the publication in the publication list.
- Classification code: this will find the relevant research classification area in the publication list.
- Institute name: this will find the publication by author institution.

The query will return relevant matches up to a maximum record of 100 per search. An asynchronous call makes the Home Controller method by button click event function in jQuery and returns the data to the View.

The search button invokes the jQuery function `getPublications()`

```
function getPublications(searchText, personId, publicationId, count,
   personName, classificationCode, title, instituteName) {
        var take = count;
        if (take === null) {
            take = 50;
        }
        var skip = 0;
        $("#loadingPanelTable").show();
        $("#panelPublicationList").hide();
        $.ajax({
            url: '/Home/GetEtisPublications',
            type: 'GET',
            dataType: 'json',
            cache: true,
            async: true,
            data: { searchText, personId, publicationId, take, skip,
               personName, classificationCode, title, instituteName },
            success: function (data) {
            }
        });
}
```

Function will do a Ajax call to the Home Controller method `GetEtisPublications()` which invoke the ETIS API client method, `GetPublications()` function in `IEtisApiClientService` interface and the `UpdatePdfLink()` in `IEtisFileDownloadService` interface.

```
public IActionResult GetEtisPublications(string searchText, string
    personId, string publicationId, int take, int skip,  string
    personName, string classificationCode, string title, string
    instituteName)
```

`GetPublications()` – make an async HTTP request to the ETIS API and return the corresponding publications data in JSON format. Only required data fields are captured when deserializing the JSON into the data Model and return to the View. Also, this Data is used for generates the BibTeX data as well.

- Guid: unique identifier for the record
- Title: title of the publication
- DOI: digital object identifier of the publication
- PublicationTypeNameEng: publication type in English
- AuthorsText: names of the authors, includes both first and last names.
- PublishingYear: publish year of the publication.
- ClassficationCode: ETIS classification code
- Periodical: if the publication is an article, then the journal of the article
- Editor: name of the editor
- Publisher: name of the publisher
- BookTitle: if the publication is a book, then book title
- DisplayInfo: description of the publication

The function call hierarchy as below,

$('\#btnSearch').click() -> GetEtisPublications() -> GetPublications()$

As mentioned before, this function also updates the PdfLink data model with a downloadable link of the publication by using `UpdatePdfLink()`. This method is elaborated in detail in the next module.

4.2.2 PDF Files Extraction Module

PDF file download method, $UpdatePdfLink()$ works as below pseudo code,

```
IF ETIS page has the PDF file
    THEN  webscrape the page and return the link.
ELSE IF publication has DOI
    THEN Get link from Crossref
        IF Crossreff has the link
            THEN Download the file;
        ELSE
            Check the original location with DOI and download the
               file.
                IF original location has a downloadable file
                    THEN  Download the file;
                ELSE
                    RETURN PDF not found;
                ENDIF;
        ENDIF,
ELSE
    RETURN Pdf not found;
ENDIF,
```

This function is executed in two ways to get the PDF file. It checks the ETIS web page first to download the file from the web page. If not successful, then call the Crossref APIs if the DOI identifier is available. It saves a copy of the file in the local server to maintain the publication file library, and each file is renamed with publication GUID, which comes from the ETIS, which helps avoid duplication records. The following code block illustrates the implemented logic.

GetDowloadLink()

`IEtisFileDownloadService` interface class has the `GetDowloadLink()` method implemented. This function is capable of download the pdf files from the ETIS web pages. We used web scraping techniques to identify the downloadable link that contains on the publication details page. The web page consists of a GUID number of the publication;

therefore, we used that number to generate the link. Extracted link mapped to the publication data Model that passes to the View. Therefore, GUID set as input parameter for this method.

EXAMPLE

- Guid of the publication: e7200319-fe8a-43bf-908e-447db2ca178f
- URL: `https://www.etis.ee/Portal/Publications/Display/e7200319-fe8a-43bf-908e-447db2ca178f?lang=ENG`

```
public async Task<string> GetDowloadLink(string publicationId)
{
    var webLink = $"https://www.etis.ee/Portal/Publications/Display/{
       publicationId}?lang=ENG";
    var html = await _client.GetStringAsync($"{webLink}");
    var htmldoc = new HtmlDocument();
    htmldoc.LoadHtml(html);
    var tableTag = htmldoc.GetElementbyId("
       Publication_Attachment_file_links");
    if (tableTag == null)
        return "";
    var hrefTag = tableTag.SelectNodes("//tbody/tr/td/a").First();
    var pdfLink = hrefTag.Attributes[0].Value;
    var file = $"https://www.etis.ee{pdfLink}";

    return file;
}
```

Function used the HTMLAgilityPack NuGet package. The HttpClient class instance makes an async call to the webpage and gets the content and results form as HtmlDocument, which can be a query as a list. In the ETIS web page DOM, PDF files are referenced inside a table tag with id `Publication_Attachment_file_links`. The algorithm captures this id and goes through each table row nodes to find the `href` tag that includes the link of the file and return to the caller.

If the PDF file is not available on the ETIS web page, the algorithm checks the DOI identifiers availability and searches the publication PDF link with Crossref. After analyzing

Figure 4.1: ETIS HTML Page

these links' behavior, it is noticed that most of the links do not have the .pdf extension, and the PDF files are available in a hosted environment with an embedded PDF viewer.

Ex: `https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-epa.2019.0877`

Most of the web browsers are compatible to show these types of links. However, retrieve the PDF file from the code using binary data is not effective and time-consuming to implement since the URL has different header parameters and there is no specific pattern. Therefore, a different approach is considered and implemented to download the PDF files using the Google Chrome browser.

It noticed that PDF files that open with the Google Chrome browser keep a cache file in the following location in a Windows machine.

`C:\Users\UserName\AppData\Local\Google\Chrome\UserData\Default\Cache`

Therefore, this location is used to copy the PDF file to the server location where the GROBID module takes the PDF files to be processed. The implemented algorithm considers the following steps.

- Clear all the data from the above cache location.
- Open the Google Chrome browser from the code with the PDF link
- Sleep the thread for a while until google chrome loads fully the PDF
- Copy the latest cache file from the cache location with has application/pdf format

- Rename the file with publication GUID
- Move the file to the server location to process with GROBID

Each file in the directory is accessed in a loop and deleted to remove the cache from the Chrome application; it might give an exception for specific files since the file might hold from other processes. These errors are avoided with try-catch blocks. The PDF link opens with `System.Diagnostics.Process.Start()` method with URL passed as an argument. However, the cache file takes little time to generates in the folder. Therefore, System sleeps for 1000ms. The algorithm finally takes all the files in that folder and checks for application/pdf file type. If any file is found, it will rename with the publication GUI and copy to another folder. Few drawbacks have been found with this approach. Since the program accesses the location that belongs to the Google Chrome application, some files might be locked by other processes to read. Also, for each link, the Chrome application opens, and that can be caused to high memory utilization in the server. Therefore, processing a large number of files might take a long time. However, this area needs further research to optimize the performance. Such as with the use of scheduled jobs.

4.2.3 BibTeX Generation

The publication details are displayed on the ETIS Publication web page where the users can check the BibTeX data on the results page by clicking the user action button in each table row. The data displays in a Modal popup view that the user can copy the content. BibTeX generation is categorized into two parts,

- BibTeX for publications
- BibTeX for publication's bibliographies

BibTeX for publications

Each publication is checked and filtered out the DOI identifiers. If a publication contains a DOI, this module will get the relevant BibTeX by an Ajax call to Home Controller and an

async call to the Crossref HTTP client service.`GetBibTex()` method is implemented to call the Crossref API and return the result to the Home Controller. One important thing here is, we manipulate the Crossref result to add ETIS classification code into the BibTeX format and rearrange the contents to show up in the popup modal. Following jQuery code implemented in the front end to re-format the BibTeX contents,

```
var removeBracket = str.substring(0, str.length - 1);
var array = removeBracket.split(",");
var bitex = array.join(",<br>     ");
bitex += ',<br>      etis = {' + removeLastChar(
   classification) + '}<br>     ';
bitex += '<br>}';
$('#bibBody').html(bitex);
```

BibTeX data returned as string from Crossref; first, it removes the last curly bracket and split all the contents by a comma. Then rejoin again with `<br>` and spaces, that will help display the data nicely and readable in the modal. ETIS classification code is added to the content as the last element, and finally, it binds the entire string as HTML code to the modal body.

If a publication does not contain a DOI, an algorithm is implemented to generate the BibTeX data according to the standard format by using the data that available from the ETIS API. Oren Patashnik has mentioned fourteen different standard BibTeX entry types in his research [34]. Every entry has standard data fields and unique fields that are mostly used. Table 4.1 illustrates the BibTeX entry types.

Table 4.1: BibTeX entries

ENTRY	REQUIRED FIELDS	OPTIONAL FIELDS
article	author,title,journal,year	volume, number, pages, month, note
book	author/editor,title,publisher, year	volume or number, series, address, edition, month, note
booklet	title	author,howpublished, address, month, year, note
conference	Same as inproceedings	Same as inproceedings
inbook	author/editor,title,chapter and/or pages, publisher, year	volume or number, series, type, address, edition, month, note
incollection	author,title,booktitle, publisher, year	editor, volume or number, series, type, chapter, pages, address, edition, month, note
inproceedings	author,title,booktitle, year	editor, volume or number, series, pages, address, month, organization, publisher, note
manual	title	author, organization, address, edition, month, year, note
mastersthesis	author, title, school, year	type, address, month, note
misc	none	author, title, howpublished, month, year, note
phdthesis	author, title, school, year	type, address, month, note
proceedings	title, yearr	volume, number, pages, month, note
techreport	author, title, institution, year	editor, volume or number, series, address, month, organization, publisher, note
unpublished	author, title, note	type, number, address, month, note

In this project, we checked and validated the data fields available in ETIS API for generates the BibTeX and form the format accordingly. Therefore, the following strategies and validations are considered when doing the implementation.

- Map the BibTeX entry types with ETIS publication types.
- Check mandatory fields are available or not.
- Map option fields.
- Remove diacritics and other characters which not supported by BibTeX.

- Format the data as BibTeX entry.

There are nine ETIS publication types found in API responses, and mapping has been done as in table 4.2.

Table 4.2: BibTeX mapping

ETIS PUBLICATION TYPE	MAPPED BIBTEX ENTRY TYPE
article in a journal	article
Convert the fully document into TEI XML	article
article in a newspaper or web portal	article
article / chapter in a book	article
book / monograph	book
conference presentation	inproceedings
dissertation	phdthesis
other electronic publications	misc
editing	misc
other	misc

The `GenerateBibTex()` method is implemented to provide the BibTeX data according to mentioned details. The citation key is implemented by combining the author and the year of the publication. Since the Estonian alphabet has diacritics characters such as š, ä, ö, and õ., we used String. Normalize approach to remove those characters from the BibTeX string. First, we normalize to Form D by splitting the full text from the diacritics character and getting the base string, and removing the non-spacing characters. Then the string array Normalized back to Form C [35].

```
private string RemoveDiacritics(string text)
{
    var baseString = text.Normalize(NormalizationForm.FormD);
    var strBuilder = new StringBuilder();
    foreach (var charctor in baseString)
    {
        var unicodeCategory = CharUnicodeInfo.GetUnicodeCategory(
            charctor);
        if (unicodeCategory != UnicodeCategory.NonSpacingMark)
        strBuilder.Append(charctor);
```

```
    }
    return strBuilder.ToString().Normalize(NormalizationForm.FormC);
}
```

BibTeX for publication's bibliographies

Bibliography extraction from publication is handle in two ways.

- Get bibliography by Crossref if DOI available.
- Get bibliography by GROBID if PDF file available.

The pseudo-code for extraction bibliography as below,

```
IF Publication has DOI
    THEN  Get references from Crossref
        IF References is not null
            FOR each publication in reference list DO
                Get DOI identifier;
                    IF Publication has DOI
                        THEN  Get BibTeX from Crossref
                    ELSE
                        RETURN BibTeX not found;
                    ENDIF;
            END FOR
        ELSE IF publication has PDF file
            THEN Get references from GROBID
                IF References is not null
                    Check DOI identifier;
                        IF Publication has DOI
                            THEN  Get BibTeX from Crossref
                        ELSE
                            Generate BibTeX with GROBID;
                        ENDIF;
                ELSE
                    RETURN BibTeX not found;
                ENDIF;
        ELSE
            RETURN References not found;
```

```
        ENDIF;
ELSE IF publication has PDF file
    THEN Get BibTeX from GROBID
        IF BibTeX is null
            RETURN BibTeX not found;
        ELSE
            RETURN BibTeX;
        ENDIF;
ELSE
    RETURN References not found;
ENDIF;
```

Crossref APIs provide the service to get the research publication's bibliographies by the DOI identifier. Therefore, the algorithm first checks to get the references with the Crossref API and parse the DOI for each with the above-explained BibTeX services. If this fails, it switches to the GROBID by checking the PDF file availability to extract the bibliography from the papers. GROBID is capable of parsing the PDF files to extract the bibliographies and return them in BibTeX format. However, the implemented method uses a hybrid solution for bibliography extraction with Crossref. If the GROBID service discovered any DOI identifier inside a bibliography record, it calls the Crossref service to resolve the DOI and get the BibTeX data. In this way, it increases the accuracy and the availability of the data. Also, this function removes diacritics characters from the BibTeX data, and finally, the generated bibliography and BibTeX pass to the popup modal in the web page. The modal has Bootstrap collapse panels for each bibliography. Users can recognize each record by its title and click to expand the title to view the BibTeX. The "Ref" button will be disabled if there is no reference to display.

4.2.4 File Library and Download Module

As explained in the early sections, when the system is processing the publication, it updates the data model with the PDF download link. If the algorithm found these links, it saves a copy of the file in the local server and renames it with publication GUID. The file count will be gradually increased when users search for more publications in the web app. Therefore, we maintain all the publications as a digital library and might be helpful

for future research tasks. Also, the system provides access to the users to download each publication to the user's machine. If the system could not find any file, the button will be disabled.

If a user retrieved the publication using the ETIS Profile search engine and directed to the publication page, the system allows the user to download PDF and .bib files as a batch. The system has two buttons for downloading the PDF and BibTeX files in ZIP file format. This module searches all the research papers and the BibTeX data that match the author's profile and makes a Zip file in the local server for downloads. These files are renamed with the publication title, that the user can identify the file quickly. The Zip file will be renamed with the author's name. An important fact here is that since the system is run in the TalTech network and uses its subscription access for publication downloads, the system is only available for students and others inside the network. So, we do not breach any security concerns.

Chapter 5

Results Validation

BibTeX data is generated in three ways in the implemented system: Crossref, GROBID, and system algorithms. All three approaches tested and validated separately were also applied in a .bib file to produce a bibliography in a LaTeX editor to check the accuracy of the actual output.

WITH CROSSREF

BibTeX data generated with Crossref API is compared and validated with external online tools, and it found out that the output is 100% accurate and format is the same in every tool other than the ETIS classification code. Several tests are carried out for DOI identifiers by validate with online tool `https://www.doi2bib.org/` and following results are gathered.

Table 5.1: Test results of BibTeX data generates with Crossref

DOI	GENERATED IN SYSTEM?	VALIDATED IN DOI2BIB	RESULT
10.1109/ECCTD.2017.8093229	Yes	match 100%	Success
10.1109/TSP49548.2020.9163557	Yes	match 100%	Success
10.3390/app9224829	Yes	match 100%	Success
10.1109/BEC.2016.7743753	Yes	match 100%	Success
10.1109/TSP.2019.8769086	Yes	match 100%	Success

An example of a test case shows in the following figures,

Input DOI: 10.1109/ECCTD.2017.8093229

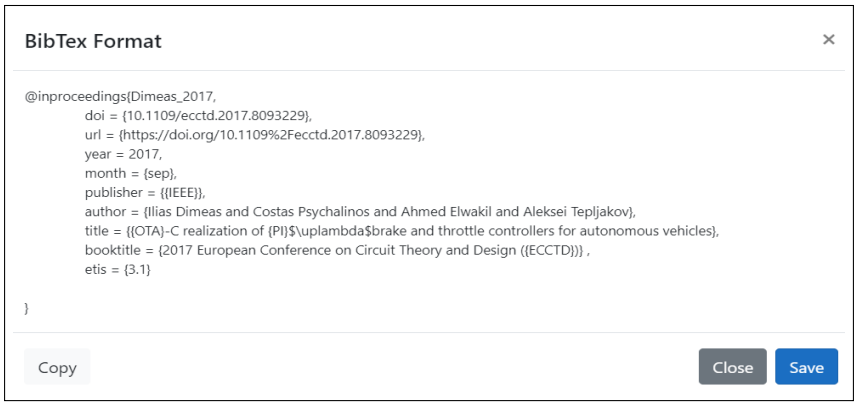


Figure 5.1: Example of system generated BibTeX with DOI

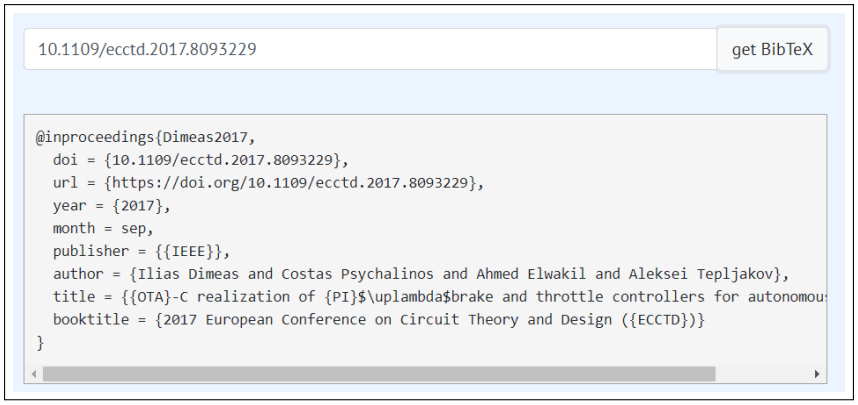


Figure 5.2: Example of BibTeX generated with doi2bib online tool

WITH GROBID

The system is used GROBID to generates the BibTeX for each reference in the publications. The accuracy of the generated BibTeX formats depends on the information available on each reference and the style used to format those. Randomly selected three research paper's bibliography lists are validated with the system-generated BibTeX and generated data applied to the LaTeX editor to check how the citation appears.

Paper 1: `https://www.etis.ee/File/DownloadPublic/d95f4f61-aeba-41df-92ba-d957c39592c2?name=Fail_Lissovski_Prague_2007.pdf`

Paper 2: `https://www.etis.ee/File/DownloadPublic/6a9537b9-021e-4c77-a98b-59becc169dc2?name=Fail_RTUCON2015-051.pdf`

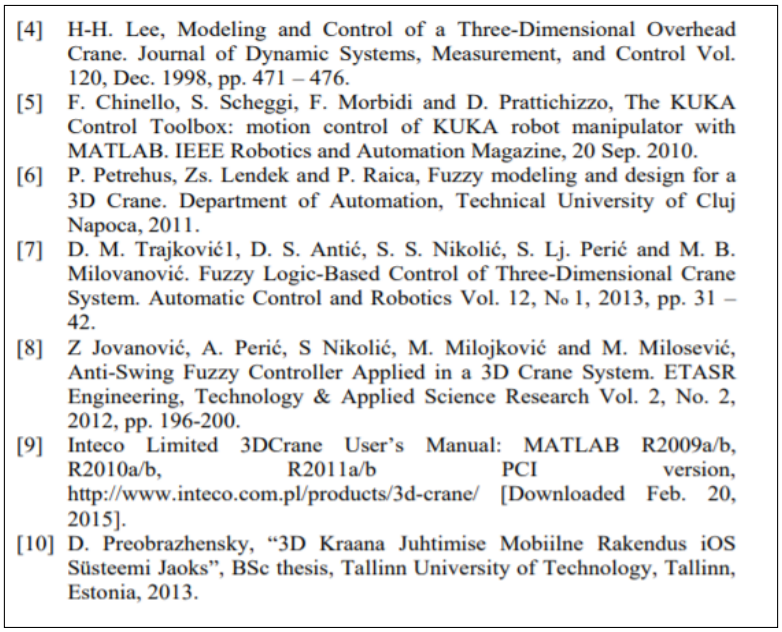
Paper 3: https://www.etis.ee/File/DownloadPublic/8fe3955a-0973-4eb2-9a06-b0901c96f686?name=2016_aBEC.pdf

Table 5.2: Test results of BibTeX data generates with Crossref

INPUT	CAPTURED THE BIBLIOGRAPHY?	GENERATE BIBTEX?	GENERATED IN LaTeX?	RESULT
Paper 1	Yes	Yes	Yes	Success
Paper 2	Yes	Yes	Yes	Success
Paper 3	Yes	Yes	Yes	Success

An example of a test case shows in the following figures,

Input: Paper 2 — Validate reference number 4



[4] H-H. Lee, Modeling and Control of a Three-Dimensional Overhead Crane. Journal of Dynamic Systems, Measurement, and Control Vol. 120, Dec. 1998, pp. 471 – 476.

[5] F. Chinello, S. Scheggi, F. Morbidi and D. Prattichizzo, The KUKA Control Toolbox: motion control of KUKA robot manipulator with MATLAB. IEEE Robotics and Automation Magazine, 20 Sep. 2010.

[6] P. Petrehus, Zs. Lendek and P. Raica, Fuzzy modeling and design for a 3D Crane. Department of Automation, Technical University of Cluj Napoca, 2011.

[7] D. M. Trajković1, D. S. Antić, S. S. Nikolić, S. Lj. Perić and M. B. Milovanović. Fuzzy Logic-Based Control of Three-Dimensional Crane System. Automatic Control and Robotics Vol. 12, No 1, 2013, pp. 31 – 42.

[8] Z Jovanović, A. Perić, S Nikolić, M. Milojković and M. Milosević, Anti-Swing Fuzzy Controller Applied in a 3D Crane System. ETASR Engineering, Technology & Applied Science Research Vol. 2, No. 2, 2012, pp. 196-200.

[9] Inteco Limited 3DCrane User's Manual: MATLAB R2009a/b, R2010a/b, R2011a/b PCI version, http://www.inteco.com.pl/products/3d-crane/ [Downloaded Feb. 20, 2015].

[10] D. Preobrazhensky, "3D Kraana Juhtimise Mobiilne Rakendus iOS Süsteemi Jaoks", BSc thesis, Tallinn University of Technology, Tallinn, Estonia, 2013.

Figure 5.3: Sample of PDF bibliography list

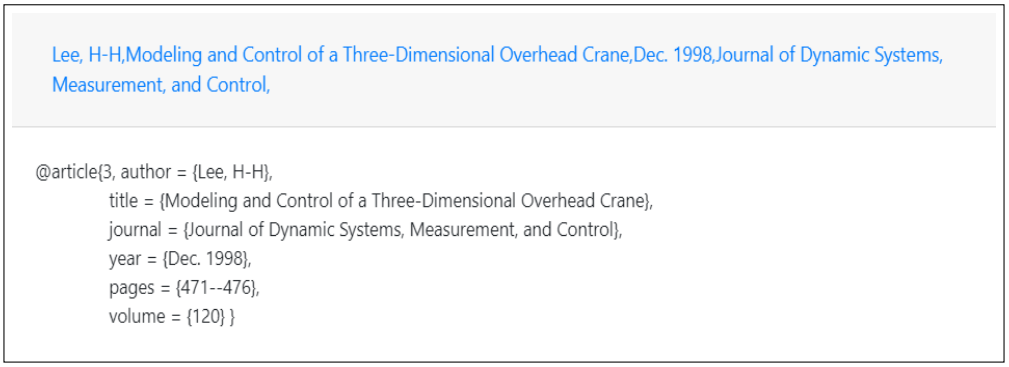


Figure 5.4: Example of system generated BibTeX with GROBID

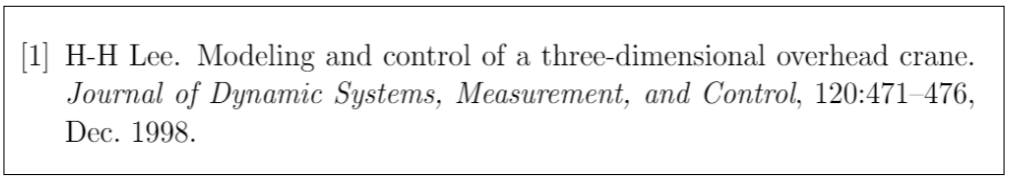


Figure 5.5: Example of system generated BibTeX in LaTeX environment

After analyzing the test results, it shows that GROBID is successfully extracted all the bibliographies from the mentioned three PDF files and converted them into BibTeX format. The example shows how GROBID captured the fourth reference from second research paper and how the data present in the implemented system. Further, the BibTeX entry is added to the Overleaf online LaTeX editor, complied with, and checked the output to validate the usability of the data. It used the ”unsrt” style to format the bibliography. Compared to reference number 4 in the figure 5.3 with figure 5.5, it proves that the generated BibTeX can be used in the LaTeX environment without any issues. Since the bibliography style of the input data differs from figure 5.5, slight differences can be seen in the ”pages” and the ”volume” data fields.

WITH SYSTEM ALGORITHM (WITHOUT DOI)

Publications that do not have DOI identifiers are generated with available data with the implemented algorithm. The most available publication types (proceedings, article, book, Ph.D. thesis) in ETIS are tested and validated to pass the following conditions,

- BibTeX should contain all the mandatory fields for each type. Otherwise, type should generate as misc

- BibTeX file should process in the LaTeX environment.

Table 5.3: BibTeX entry type test results

BIBTEX TYPE	REQUIRED FIELDS	AVAILABLE IN OUTPUT?	GENERATED IN LaTeX?	RESULT
@inproceedings	author	Yes	Yes	Success
	title	Yes	Yes	
	booktitle	Yes	Yes	
	year	Yes	Yes	
@article	author	Yes	Yes	Success
	title	Yes	Yes	
	journal	Yes	Yes	
	year	Yes	Yes	
@book	author/editor	Yes	Yes	Success
	title	Yes	Yes	
	publisher	Yes	Yes	
	year	Yes	Yes	
@phdthesis	author	Yes	Yes	Success
	title	Yes	Yes	
	school	Yes	Yes	
	year	Yes	Yes	

An example of a test case of @inproceedings BibTeX type shows in the following figures,

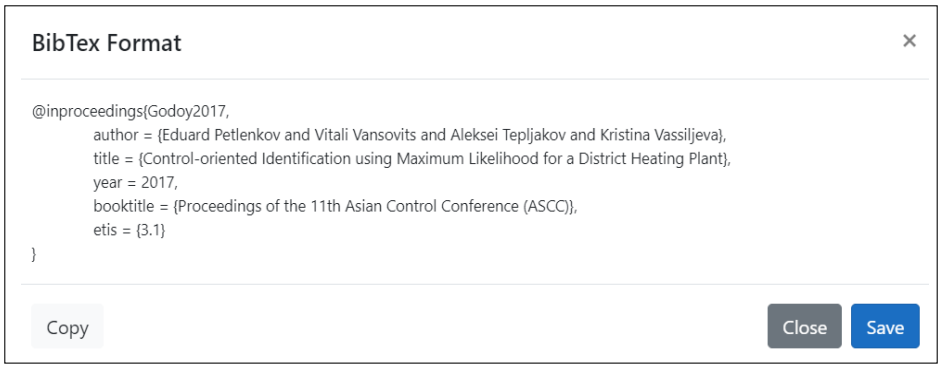
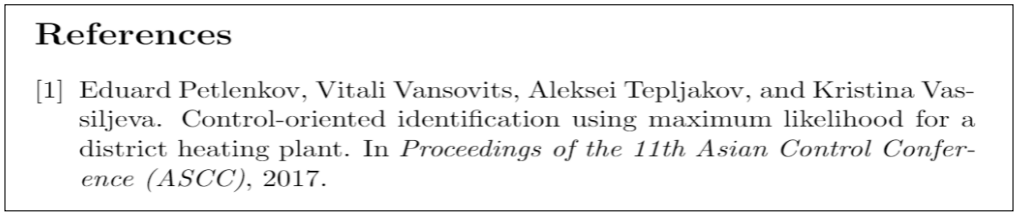


Figure 5.6: Example of system generated BibTeX with ETIS data



References

[1] Eduard Petlenkov, Vitali Vansovits, Aleksei Tepljakov, and Kristina Vassiljeva. Control-oriented identification using maximum likelihood for a district heating plant. In *Proceedings of the 11th Asian Control Conference (ASCC)*, 2017.

Figure 5.7: Example of system generated BibTeX with ETIS data in LaTeX environment

Since this module uses the ETIS meta-data to generate the BibTeX, output data is validated with definitions stated in the table 4.1. Table 5.3 represents only the most available publication types in the ETIS system. The results prove that the output complies with the standard BibTeX definitions. If ETIS meta-data is not enough to format the BibTeX for a required entry type, the system creates a ”@misc” entry. Figure 5.7 shows the bibliographic record that compiled BibTeX by Overleaf online editor with ”unsrt” style.

In summary, all three approaches showed positive results, proving the implemented system can handle the stated project goals.

Chapter 6

Post Developments and Enhancements

The research findings are mainly related to the ETIS data and developed for use as a web application. However, the core system can scale up for other integrations such as microservice solutions, web API for third parties and other data sources to output the BibTeX data. As the next step following areas need to be improved to enhance the user experience in the application,

- PDF file extraction for embedded links.
- Search queries with more search filters.
- Background task to increase the performance. (process PDF files, batch download)

Since the system is integrated with GROBID micro-services, there are more areas that can be implemented into the system. Such as extract other data in PDF files (Abstract, introduction, conclusion, etc.).

Conclusion

In conclusion, bibliographic information is becoming essential among researchers as publications are growing fast across the world. This project aims to extract the bibliographic metadata from the ETIS system and generate BibTeX data since the ETIS data shows that 80% of the publications do not have registration with DOI.

The core system is integrated with several modules which are implemented with mainly Microsoft technologies. GROBID and Crossref have been considered for the implementation for parsing the PDF files and generate BibTeX. ETIS data is extracted by using their open APIs. Publication's PDF files and BibTeX data are saved in a local digital library that users can download as a batch file. The core system is connected with a web application for end-users to retrieve the extracted data.

Based on the output data, it is efficient to use GROBID services for extracting the PDF contents and format the data. Even though few of its services are integrated with this project, it opens several other areas for researchers interested in data mining and exploring more on other services highlighted in this thesis.

All the BibTeX data entry types available in ETIS are tested with external tools and an actual LaTeX environment to ensure the usability of the system outcome, proving that the implemented system can handle the requirements stated in the thesis goals. Several new research areas are found during the implementation that can enhance the current system.

Bibliography

[1] M. Khabsa and C. L. Giles, "The Number of Scholarly Documents on the Public Web," *PLoS ONE*, vol. 9, no. 5, p. e93949, may 2014. [Online]. Available: https://doi.org/10.1371%2Fjournal.pone.0093949

[2] L. J. Wilkinson, "Labs - Crossref," https://www.crossref.org/labs/, Mar 2018.

[3] "DOI Resolution Documentation," https://www.doi.org/factsheets/DOIProxy.html, accessed 18-Apr-2021.

[4] "Introduction - GROBID Documentation," https://grobid.readthedocs.io/en/latest/Introduction/, accessed 18-Apr-2021.

[5] M. Lipinski, K. Yao, C. Breitinger, J. Beel, and B. Gipp, "Evaluation of header metadata extraction approaches and tools for scientific PDF documents," in *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries - JCDL '13*. ACM Press, 2013. [Online]. Available: https://doi.org/10.1145%2F2467696.2467753

[6] "ETIS user manual," https://www.etag.ee/wp-content/uploads/2012/05/ENG2_User_manual_long.pdf, 2016, accessed 18-Apr-2021.

[7] J. Beel, B. Gipp, S. Langer, M. Genzmehr, E. Wilde, A. Nürnberger, and J. Pitman, "Introducing mr. DLib,," in *Proceeding of the 11th annual international ACM/IEEE joint conference on Digital libraries - JCDL '11*. ACM Press, 2011. [Online]. Available: https://doi.org/10.1145%2F1998076.1998187

[8] I. Councill, C. L. Giles, and M.-Y. Kan, "ParsCit: an open-source CRF reference string parsing package," in *Proceedings of the Sixth International Conference on*

Language Resources and Evaluation (LREC'08). Marrakech, Morocco: European Language Resources Association (ELRA), May 2008. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2008/pdf/166_paper.pdf

[9] D. Tkaczyk, P. Szostek, M. Fedoryszak, P. J. Dendek, and Ł. Bolikowski, "CERMINE: automatic extraction of structured metadata from scientific literature," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 18, no. 4, pp. 317–335, jul 2015. [Online]. Available: https://doi.org/10.1007%2Fs10032-015-0249-8

[10] P. Lopez, "GROBID: Combining Automatic Bibliographic Data Recognition and Term Extraction for Scholarship Publications," in *Research and Advanced Technology for Digital Libraries*. Springer Berlin Heidelberg, 2009, pp. 473–474. [Online]. Available: https://doi.org/10.1007%2F978-3-642-04346-8_62

[11] P. Lopez, "GROBID," https://github.com/kermitt2/grobid, 2008–2021.

[12] D. Tkaczyk, A. Collins, P. Sheridan, and J. Beel, "Machine Learning vs. Rules and Out-of-the-Box vs. Retrained," in *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*. ACM, may 2018. [Online]. Available: https://doi.org/10.1145%2F3197026.3197048

[13] J. Boyd, "Automatic Metadata Extraction The High Energy Physics Use Case," 2015.

[14] M. Grennan and J. Beel, "Synthetic vs. Real Reference Strings for Citation Parsing, and the Importance of Re-training and Out-Of-Sample Data for Meaningful Evaluations: Experiments with GROBID, GIANT and Cora," 2020.

[15] L. L. Wang, K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Eide, K. Funk, R. M. Kinney, Z. Liu, W. Merrill, P. Mooney, D. Murdick, D. Rishi, J. Sheehan, Z. Shen, B. B. S. Stilson, A. D. Wade, K. Wang, C. Wilhelm, B. Xie, D. A. Raymond, D. S. Weld, O. Etzioni, and S. Kohlmeier, "CORD-19: The Covid-19 Open Research Dataset," *ArXiv*, 2020.

[16] Ynnig, "Why Crossref? - Crossref," https://www.crossref.org/education/why-crossref/, 2020.

[17] R. Lammey, "Using the crossref metadata API to explore publisher content," *Science Editing*, vol. 3, no. 2, pp. 109–111, aug 2016. [Online]. Available: https://doi.org/10.6087%2Fkcse.75

[18] A. Brand, *Metadata demystified : a guide for publishers / Amy Brand, Frank Daly and Barbara Meyers.*, Bethesda, Md., Hanover, Pa., 2003.

[19] "Metadata retrieval, REST API," https://www.crossref.org/education/retrieve-metadata/rest-api/.

[20] "A complete guide to the BibTeX format," https://www.bibtex.com/g/bibtex-format /, 2019.

[21] J. Fenn, "Managing citations and your bibliography with bibtex," *The PracTEX Journal,(4)*, 2006.

[22] "biber – a BibTeX replacement for users of BibLaTeX," https://ctan.org/pkg/biber.

[23] "BibLaTeX – Sophisticated Bibliographies in LaTeX," https://ctan.org/pkg/biblatex.

[24] "Troubleshooting and known issues - GROBID Documentation," https://grobid.readthedocs.io/en/latest/Troubleshooting/#windows-related-issues.

[25] S. Smith, "Overview of ASP.NET Core MVC," https://docs.microsoft.com/en-us/aspnet/core/mvc/overview, 2018.

[26] "GROBID User Manual, GROBID Service API," https://grobid.readthedocs.io/en/latest/Grobid-service/#grobid-service-api, accessed 18-Apr-2021.

[27] J. Li, "Agile Software Development," 2012.

[28] "What's Html Agility Pack," https://github.com/zzzprojects/html-agility-pack, accessed 18-Apr-2021.

[29] "Rest Sharp, user manual, Getting Started," https://restsharp.dev/getting-started/, accessed 18-Apr-2021.

[30] J. L. Carlson, *Redis in Action*. USA: Manning Publications Co., 2013.

[31] S. Dasgupta, ".NET Core — How to use Redis Cache to boost application performance," https://medium.com/@saurabh.dasgupta1/about-7fb96fb1f80d, accessed 18-Apr-2021.

[32] M. Murugan, "Redis Caching in ASP.NET Core – Distributed Caching Detailed," https://codewithmukesh.com/blog/redis-caching-in-aspnet-core/, accessed 18-Apr-2021.

[33] S. Smith, "Working with a distributed cache," https://aspnetcore.readthedocs.io/en/stable/performance/caching/distributed.html, accessed 18-Apr-2021.

[34] O. Patashnik, "BibTeXing. documentation for general BibTeX users," *Electronic document accompanying BibTeX distribution*, 1988.

[35] Microsoft Inc., "String.Normalize Method," https://docs.microsoft.com/en-us/dotnet/api/system.string.normalize?redirectedfrom=MSDN&view=net-5.0#overloads, accessed 18-Apr-2021.

Appendix A

Source Code

The web application's full source code is available at GitHub.https://github.com/YasithAri/ETIS/tree/master. The application can run locally with Microsoft Visual Studio 2019 with .Net Core framework 3.1. The local server should have Redis cache up and running; otherwise, the application will not run as expected. Please add the Redis server address in the option.Configuration parameter inside ConfigureService(IServiceCollection services) method in Startup.cs class.

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddControllersWithViews();
    services.AddControllersWithViews().AddRazorRuntimeCompilation();

    services.AddMvc();
    services.AddDistributedRedisCache(option =>
    {
        option.Configuration = "127.0.0.1";
        option.InstanceName = "master";
    });
}
```

NB! If the application runs outside the Taltech Network, the research paper download option might not be available for specific papers due to privacy policies.

Appendix B

API Responses

B.1 ETIS GET Profile API Json response


```
[
{
    "Guid": "5fe0c484-2729-48a6-9c79-2eca67badff4",
    "PersonGuid": "5fe0c484-2729-48a6-9c79-2eca67badff4",
    "PersonName": "Lincoln John Theo",
    "DateOfBirth": "",
    "DateOfDeath": "",
    "Email": "theolj@cput.ac.za",
    "Phone": "",
    "Phone2": "",
    "Description": "",
    "Homepage": "",
    "ResearcherId": "",
    "OrcId": "",
    "OtherNames": [],
    "OccupationInfos": [],
    "EducationPaths": [
    {
        "Guid": "3bd35dd2-679d-40c9-bf4b-7bd1da000f4d",
        "DisplayInfo": "Bachelor of Arts (BA), Arts Faculty,
           University of Cape Town, South Africa",
        "PeriodEndDate": "31.12.1993",
```

```
            "PeriodStartDate": "01.01.1991"
        },
        {
            "Guid": "161d09db-b7a0-4862-8bb3-f0d7e6288a1c",
            "DisplayInfo": "Bachelor of Laws (LLB), Law Faculty,
               University of Cape Town, South Africa",
            "PeriodEndDate": "31.12.1996",
            "PeriodStartDate": "01.01.1994"
        },
        {
            "Guid": "f8cc8efa-dfd4-449a-8ff1-9f4903ed4df8",
            "DisplayInfo": "Certificate in Legal Practice (CLP),
               School for Legal Practice, University of Cape Town,
                South Africa",
            "PeriodEndDate": "31.12.1998",
            "PeriodStartDate": "01.07.1998"
        },
        {
            "Guid": "347e8e89-8714-40df-b199-39465a17414d",
            "DisplayInfo": "Attorneys Admission Certificate (AAC),
               Cape Law Society, Cape Provincial Division of the
               High Court of South Africa",
            "PeriodEndDate": "31.12.2000",
            "PeriodStartDate": "01.01.1999"
        },
        {
            "Guid": "96d97b97-8603-449c-9275-692a721a575f",
            "DisplayInfo": "Masters in Social Science (M Soc Sci (
               African Studies)), Harry Oppenheimer Centre for
               African Studies, University of Cape Town, South
               Africa",
            "PeriodEndDate": "31.12.2005",
            "PeriodStartDate": "01.01.2003"
        }
        ],
        "ResearchActivities": [
        {
            "Guid": "cee28b9a-57fe-476b-a47f-6a122e697cbb",
            "DisplayInfo": "Head of Film Programme within a Media
```

```
            Department, serve on faculty committees: research,
            ethics, teaching and learning",
        "PeriodEndYear": null,
        "PeriodStartYear": 2014
    }
    ],
    "CreativeActivities": [],
    "QualificationAdditionalInfos": [],
    "ResearchAwards": [
    {
        "Guid": "8d302939-f51d-464b-8285-e9305a2bdd6d",
        "DisplayInfo": "CHE/HELTASA South African National
            Excellence in Teaching Award",
        "Year": 2016
    }
    ],
    "ResearchDirectionInfos": [
    {
        "Guid": "00000000-0000-0000-0000-000000000000",
        "DisplayInfo": "film and media, screenwriting"
    }
    ],
    "Publications": [

    }
    ],
    "Degrees": [],
    "Mentorships": [],
    "ScientificEquipments": [],
    "Projects": [],
    "ResearchAreasCercs": [
    {
        "Guid": "46751221-6594-477d-837b-a8f101317cd9",
        "Code": "H330",
        "Name": "Dramatic art "
    }
    ],
    "ResearchAreasEtis": [
    {
```

```
                "Guid": "46751221-6594-477d-837b-a8f101317cd9",
                "Code": "2. ",
                "Name": "ETIS CLASSIFICATION: 2.  Culture and Society;
                    2.5.  Aesthetics and Arts Research; CERCS
                    CLASSIFICATION: H330 Dramatic art ; SPECIFICATION:
                    film and media, screenwriting"
            }
            ],
            "ServiceAdditionalInfos": [
            {
                "DisplayInfo": "Associate Professor, Media Department,
                    Faculty of Informatics \& Design, Cape Peninsula
                    University of Technology (CPUT), Cape Town, South
                    Africa",
                "Guid": "d6987e3c-7bd8-48b0-9159-1a02e33fa543"
            }
            ],
            "IndustrialProperties": []
        }
        ]
```


B.2 ETIS GET Publication API Json response


```
{
   "Guid": "b052513f-51a1-4a66-941c-d735b8496895",
   "DisplayInfo": "Tepljakov, A. (2017). Fractional-order Modeling and
        Control of Dynamic Systems. Springer International Publishing
       AG. DOI: 10.1007/978-3-319-52950-9.",
   "PublicationTypeName": "raamat / monograafia",
   "PublicationTypeNameEng": "book / monograph",
   "OpenAccessTypeName": "suletud",
   "OpenAccessTypeNameEng": "closed",
   "OpenAccessLicenceName": "",
   "OpenAccessLicenceNameEng": "",
   "Authors": [
   {
       "Guid": "6aa88d90-8623-454d-8627-6b16661de1de",
```

```
        "IdCode": null,
        "Name": "Aleksei Tepljakov",
        "RoleName": "Autor",
        "RoleNameEng": "Author"
    }
    ],
    "AuthorsText": "Tepljakov, A.",
    "Languages": "Inglise",
    "LanguagesCode": "EN",
    "LanguagesEng": "English",
    "Title": "Fractional-order Modeling and Control of Dynamic Systems
       ",
    "TitleHtml": "",
    "TitleTranslation": "",
    "IssueTitle": "",
    "University": "",
    "Editors": "",
    "Periodical": "",
    "ConferenceDescription": "",
    "PublishingPlace": "",
    "PublishingHouse": "Springer International Publishing AG",
    "Issn": "2190-5053",
    "Isbn": "978-3-319-52949-3",
    "Binding": "",
    "Number": null,
    "Part": null,
    "SupplementIssue": "",
    "SpecialIssue": "",
    "Series": "Springer Theses",
    "SeriesBinding": "",
    "PublishingYear": 2017,
    "PagesCount": "173",
    "PagesEnd": "",
    "PagesStart": "",
    "PublicationStatus": "Ilmunud",
    "PublicationStatusEng": "Published",
    "IsOpenAccess": "Ei",
    "IsOpenAccessEng": "No",
    "IsPublic": true,
```

```
    "ClassificationCode": "2.1.",
    "ClassificationDatabaseSubtype": "",
    "ClassificationName": "Monograafiad",
    "ClassificationNameEng": "Scholarly monographs",
    "PublicFile": false,
    "Url": "",
    "Doi": "http://doi.org/10.1007/978-3-319-52950-9",
    "BookDoi": "",
    "Institutions": [
    {
        "BusinessRegNo": "74000323",
        "Guid": "405f8348-9cea-4cae-b604-00d0b1c49a6c",
        "Name": "Tallinna Tehnikaulikool, Infotehnoloogia teaduskond,
           Arvutisusteemide instituut",
        "NameEng": "Tallinn University of Technology , School of
           Information Technologies, Department of Computer Systems"
    }
    ],
    "InstitutionsAsFreeText": null,
    "Comment": "",
    "Keywords": "",
    "KeywordsEng": "",
    "KeywordsAsFreeText": null,
    "UserKeywords": null,
    "Projects": [],
    "FullTextLocation": "",
    "ReferencingDatabase": "",
    "DissertationTypeName": "",
    "DissertationTypeNameEng": "",
    "DateCreated": "2017-02-15T16:01:20.7723387",
    "DateModified": "2020-02-12T16:33:01.9601533",
    "WOSdocumentType": null,
    "WOSfieldsOfResearch": null
}
```

B.3 Crossref API Json response


```
// 20210425164245
// https://api.crossref.org/works/10.1016/j.optmat.2007.11.035/

{
    "status": "ok",
    "message-type": "work",
    "message-version": "1.0.0",
    "message": {
        "indexed": {
            "date-parts": [
            [
            2021,
            1,
            3
            ]
            ],
            "date-time": "2021-01-03T07:23:37Z",
            "timestamp": 1609658617748
        },
        "reference-count": 29,
        "publisher": "Elsevier BV",
        "issue": "3",
        "license": [
        {
            "URL": "https://www.elsevier.com/tdm/userlicense/1.0/",
            "start": {
                "date-parts": [
                [
                2009,
                1,
                1
                ]
                ],
                "date-time": "2009-01-01T00:00:00Z",
                "timestamp": 1230768000000
            },
```

```
        "delay-in-days": 0,
        "content-version": "tdm"
    }
    ],
    "content-domain": {
        "domain": [

        ],
        "crossmark-restriction": false
    },
    "short-container-title": [
    "Optical Materials"
    ],
    "published-print": {
        "date-parts": [
        [
        2009,
        1
        ]
        ]
    },
    "DOI": "10.1016/j.optmat.2007.11.035",
    "type": "journal-article",
    "created": {
        "date-parts": [
        [
        2008,
        3,
        3
        ]
        ],
        "date-time": "2008-03-03T20:19:02Z",
        "timestamp": 1204575542000
    },
    "page": "567-574",
    "source": "Crossref",
    "is-referenced-by-count": 30,
    "title": [
    "VUV spectroscopy of double phosphates doped with rare earth
```

```
        ions"
    ],
    "prefix": "10.1016",
    "volume": "31",
    "author": [
    {
        "given": "J.",
        "family": "Legendziewicz",
        "sequence": "first",
        "affiliation": [

        ]
    },
    {
        "given": "M.",
        "family": "Guzik",
        "sequence": "additional",
        "affiliation": [

        ]
    }
    ],
    "member": "78",
    "reference": [
    {
        "key": "10.1016/j.optmat.2007.11.035_bib1",
        "author": "Dexter",
        "volume": "108",
        "first-page": "630",
        "year": "1957",
        "journal-title": "Phys. Rev.",
        "DOI": "10.1103/PhysRev.108.630",
        "doi-asserted-by": "crossref"
    }

    ],
    "container-title": [
    "Optical Materials"
    ],
```

```
    "original-title": [

    ],
    "language": "en",
    "link": [
    {
        "URL": "https://api.elsevier.com/content/article/PII:
            S0925346707003667?httpAccept=text/xml",
        "content-type": "text/xml",
        "content-version": "vor",
        "intended-application": "text-mining"
    },
    {
        "URL": "https://api.elsevier.com/content/article/PII:
            S0925346707003667?httpAccept=text/plain",
        "content-type": "text/plain",
        "content-version": "vor",
        "intended-application": "text-mining"
    }
    ],
    "deposited": {
        "date-parts": [
        [
        2018,
        12,
        31
        ]
        ],
        "date-time": "2018-12-31T09:28:23Z",
        "timestamp": 1546248503000
    },
    "score": 1.0,
    "subtitle": [

    ],
    "short-title": [

    ],
    "issued": {
```

```
            "date-parts": [
            [
            2009,
            1
            ]
            ]
        },
        "references-count": 29,
        "journal-issue": {
            "published-print": {
                "date-parts": [
                [
                2009,
                1
                ]
                ]
            },
            "issue": "3"
        },
        "alternative-id": [
        "S0925346707003667"
        ],
        "URL": "http://dx.doi.org/10.1016/j.optmat.2007.11.035",
        "relation": {
            "cites": [

            ]
        },
        "ISSN": [
        "0925-3467"
        ],
        "issn-type": [
        {
            "value": "0925-3467",
            "type": "print"
        }
        ],
        "subject": [
        "Electrical and Electronic Engineering",
```

```
        "General Computer Science",
        "Atomic and Molecular Physics, and Optics",
        "Electronic, Optical and Magnetic Materials"
        ]
    }
}
```