Exploration of Data Enrichment and Data Search Mechanisms for NMR Spectra via Spectrum Analysis

Michalis Psalios

Supervisor: Simon Dobson

April 9, 2018

Abstract

The vast amount of data in scientific research has increased the need for a reliable management system that allows people to store and search data effectively. The School of Chemistry and the School of Computer Science at the University of St Andrews have created a web-based research data management system for their NMR (Nuclear Magnetic Resonance) facilities called NOMAD. NOMAD provides a fast and secure data store which allows users to search data based on various metadata.

The aim of the project is to extend the functionality of the NOMAD system and enhance data search based on the actual experiment data. To achieve this, a tool that allows users to perform NMR data analysis was created and a search engine based on the analysis data was implemented. A user study was conducted on the NMR analysis tool which evaluated the usability of the tool. The outcome of the user study suggests that users were confident using the tool from the first time.

0.1 Declaration

I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated.

The main text of this project report is 10884 words long, including project specification and plan but excluding appendices.

In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

0.2 Acknowledgements

I would like to thank Prof. Simon Dobson for supervising the project and giving me his valuable feedback throughout the development of this project. I would like to thank Dr. Tomas Lebl who helped with the chemistry aspects of this project. Moreover, I would like to thank Simone Ivan Conte who had the initial idea of the project and his constant feedback and suggestions for the whole duration of the project. I would like to thank the NOMAD team, who gave me access to their code repository. Finally, I would like to thank the St Andrews School of Computer Science for allowing me to use their facilities and resources.

Contents

1. Introduction

The rapid growth of technology has driven us to the information age, making it feasible to generate and record huge quantities of new data [1]. Data volumes are approximately doubling every two years creating enormous, complex datasets, called “Big Data” [2]. Big data introduces many challenges, including how we capture, store, analyse, search, share, transfer and update the data. The concepts of big data can be summarised in four, often-overlapping difficulties: Volume, Variety, Velocity and Veracity [3]. Volume describes the vast amount of data created every day. For example, with the Internet of Things (IoT) increase and constantly more devices being able to connect to the cyber world, this data will get larger. Velocity refers to the speed of processing this data. Variety denotes the different types of data and the issues faced by analysing a variety of data from different sources in a complex system. Finally, veracity refers to the noise and biases in data and how meaningful is the data to analyse in relation to the problem.

Data are essential part of research to ensure integrity and validation of results. Research data are a valuable resource that often requires a great deal of time and money to create. One ordinary problem with scientific data in general is the extreme loss of information between the lab and the publication [4]. In the digital age, this loss of information is unreasonable.

NOMAD (NMR Online Management and Datastore) system is trying to solve this problem for NMR data. NOMAD is a web-based research data management system for NMR facilities developed as a collaboration between the Schools of Chemistry and Computer Science [5]. It creates a process flow from the data acquisition to the publication stage. It also provides a secure and reliable data store, searchability and protection of the data.

NMR spectroscopy, or magnetic resonance spectroscopy (MRS), is a chemistry technique to observe local magnetic fields around atomic nuclei. NMR is widely used to identify organic compounds. NMR spectroscopy provides comprehensive data about the structure, dynamics, reaction state and chemical environment of molecules. NMR spectra analysis can be a complicated task because of the diverse and complicated molecular structures of compounds. However, at a time when the usage of artificial intelligence is growing significantly, it is essential to explore the potentials of computer-intelligence to assist humans in the analysis of NMR spectra and the interpretation of published data [4].

Such system requires a vast amount of data to be collected and stored in a dedicated database. The database must store the raw NMR spectrum data and the analysis of the data associated with the literature, so machine learning algorithms will be able to take into account content related to any molecular structure.

1.1 Project Aim

The NOMAD system allows users to run NMR experiments and group them into compounds. A structural formula is associated with each chemical compound which describes its molecular structure. Currently, the system provides search functionality based on experiments metadata (e.g., user, group, date created) and structural information (exact, substructure and similarity search). However, there is no search functionality based on the properties of the experiments. Users are able to preview the spectrum of each experiment but not to interact with it. In order to analyse the data, users must download the data, import them into an offline NMR analysis tool, perform the analysis and copy the generated report back to NOMAD. This process is time wasting, cumbersome and it often leads to errors. It also requires that users already have an NMR analysis tool.

This project aims to improve and automate this procedure, by allowing users to perform NMR analysis (i.e., classify spectrum data, identify peaks) on NOMAD. This will lead to a much faster process and it will solve the errors occurred during data transfer.

Additionally, performing the NMR analysis on the server allows all the information to be captured by NOMAD and stored in the database. This information will enrich the current metadata. The aim of the project is to create an advanced search engine based on this metadata.

1.2 Initial Objectives

1.2.1 Primary

- Summarise the relevant literature on signal processing for NMR spectra
- Peak picking from spectrum data
- Determining integral intensity of peaks - i.e., area under the curve within certain limits
- Classify multiplet information from NMR spectrum (i.e., all three chemical shift, integral, multiplicity)

- Implement an NMR Spectrum search engine based on multiplet information

1.2.2 Secondary

- Integrate MNova to perform automatic peak picking
- Peak picking of 2D spectra
- Implement advanced NMR spectrum search by peak's physical and chemical properties (i.e., singlet, doublet, triplet peak search) and advanced AI techniques (i.e., neural networks, SSVs)
- Implement a UI tool to let the user specify the search criteria easily
- Study how the peak search engine scales with the number of classified spectra

1.2.3 Tertiary

- Exploration of novel techniques for efficient and fast peak picking from spectrum data
- Integrate the peak's classification with the molecular structure predictor developed by Jake Rivett [6] for simple molecules/spectra
- Conduct a usability study on the UI tool to search the peaks

During this project, I managed to implement all primary objectives by creating an NMR analysis tool that allows users to perform peak picking on spectrum data, determine integral intensity of peaks, and classify multiplet information from NMR spectra. When an analysis on NMR data is performed, the multiplet information are stored in the database. The NMR analysis tool is deployed at `https://mp236.host.cs.st-andrews.ac.uk`.

Moreover, I implemented an NMR spectrum search engine based on peaks' chemical properties using the information acquired by the NMR analysis tool. I explored various ways to allow users to specify the search criteria easily and, as a result, a UI tool was implemented to provide users with a simple, fast way to search for NMR spectra. Finally, a usability study was conducted on the NMR analysis tool and the results were analysed.

The project has been open sourced. The code of the whole application can be found at `https://github.com/psalios/nmr-analysis`. The aim for open sourcing the software is to provide an NMR analysis tool to people, especially students, who do not own an NMR analysis software.

1.3 Approach

The NMR analysis tool and the NMR spectra search engine based on multiplet information presented in this work will mainly be used by chemists. In addition, the system will implement the process of NMR analysis. Therefore, a good understanding of this process was required. In order to make sure that I completely understood the analysis procedure, I had regular meetings with Dr. Tomas Lebl from the School of Chemistry who explained it to me. Dr. Lebl tested the application giving me feedback about the correctness and usability of the system. His main concerns were the interaction of the system with the users and how easily would chemists be able to use the tool. Therefore, the tool was designed from the beginning with the principles of human-computer interaction design in mind, while taking into account the chemistry aspects of the system.

1.4 Thesis Outline

The rest of the report is structured in the following format: **Chapter 2** establishes fundamental knowledge, literature and related work required to understand the complete system; **Chapter 3** describes the software engineering practices followed and the technologies used for the project; **Chapter 4** explains the overall design of the application; **Chapter 5** describes the detailed implementation considering the design of the system; **Chapter 6** considers the ethical issues raised during the evaluation of the system; **Chapter 7** talks about the evaluation results and the interpretation and analysis of these results; **Chapter 8** explains future work that could be done regarding this system; Finally, **Chapter 9** concludes the project.

2. Literature Review

2.1 NMR Analysis

NMR spectroscopy is an analytic chemistry technique used in quality control and research for determining the content and purity of a sample as well as its molecular structure [7]. For example, NMR can quantitatively analyse mixtures containing known compounds. For unknown compounds, NMR can be used to infer the basic structure directly. Once the basic structure is known, NMR can be used to determine molecular conformation in solution as well as studying physical properties at the molecular level such as phase changes, solubility, diffusion and conformational exchange. In order to achieve the desired results, a variety of NMR techniques are available [7]. The NMR behaviour of proton and carbon nuclei has been exploited by organic chemistry since they provide valuable information that can be used to deduce the structure of organic compounds. Therefore, these will be the focus of our attention.

2.1.1 Chemical Shift

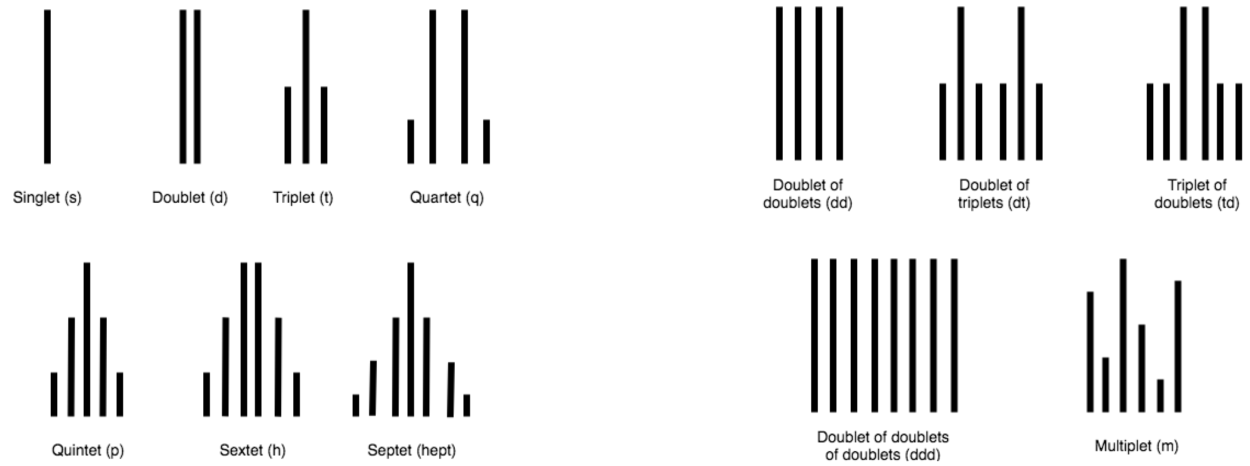
A chemical shift, or signal, is the resonant frequency of a nucleus relative to a *standard compound* in a magnetic field. The precise resonant frequency of the energy transition is dependent on the effective magnetic field at the nucleus. This field is affected by electron shielding which is in turn dependent on the chemical environment. As a result, information about the nucleus' chemical environment can be derived from its resonant frequency. Often the position and number of chemical shifts are diagnostic of the structure of a molecule [8]. Chemical shifts are measured in parts per million (ppm).

The standard compounds include: tetramethylsilane (TMS), 4,4-dimethyl-4-silapentane-1-sulfonic acid (DSS) and trimethylsilyl propionate (TSP). Unfortunately, some labs use non-standard methods for determining the chemical shifts position. This lack of standardisation makes it difficult to compare chemical shifts for the same compound between different laboratories. Chemical shift re-referencing is a post-assignment process of adjusting the assigned NMR chemical shifts to match the International Union of Pure and Applied Chemistry recommended standards in chemical shift referencing. It offers a means to correct these referencing errors and to standardise the reporting of chemical shifts across laboratories.

2.1.2 Proton-1 NMR Analysis

Proton-1, or 1H NMR, analysis is the application of nuclear magnetic resonance to hydrogen-1 nuclei in order to determine the structure of its molecules. The number of chemical shifts of the NMR spectrum indicates how many “different kinds” of protons are present and their position shows differences in the hydrogens’ chemical environments. The magnitude or intensity of the signal (area under curve) is proportional to the number of protons it presents. Therefore, the more hydrogens there are in the same chemical environment, the more intense the signal will be. Finally, the split of a signal, or spin-spin coupling, into several peaks indicates the number of nearby nuclei which have magnetic moments (see Fig. 2.1). The distance between two peaks of the same signal is called “coupling constant”.

Common 1H NMR Splitting Patterns



Simple Multiplicities

Complex Multiplicities

Figure 2.1: Multiplicity Types

For instance, Fig. 2.2 shows a fragment of a proton NMR spectrum. In this fragment, there are two chemical shifts, or multiplets. The first chemical shift is split into 8 peaks of approximately equal intensity. Thus, the multiplicity class of the first chemical shift is *ddd*. The second chemical shift is split into 6 peaks of approximate intensity proportions equal to [1, 2, 1, 1, 2, 1]. Therefore the multiplicity class of the second chemical shift is *dt*. Both chemical shifts have approximately the same magnitude, therefore, they have the same number of protons, in this case 1 proton each.

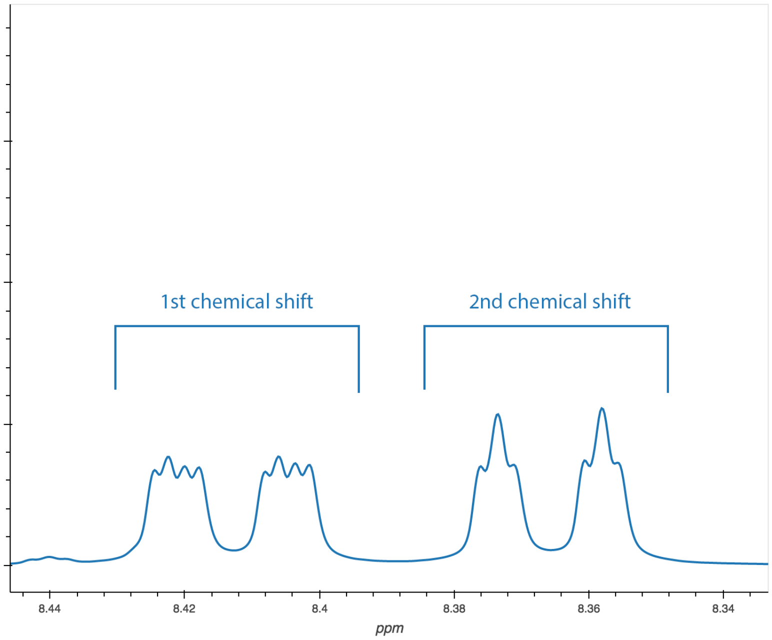


Figure 2.2: Proton Spectrum Example

2.1.3 Carbon-13 NMR Analysis

Carbon-13, or 13C NMR, analysis is the application of nuclear magnetic resonance to carbon. Similarly to the proton spectrum, the carbon spectrum allows the identification of carbon atoms in an organic molecule. The 13C NMR experiment is much less sensitive than the 1H. Therefore, the spin-spin couplings are rarely observed. For example, Fig. 2.3 shows a 13C NMR spectrum with 8 carbon atoms.

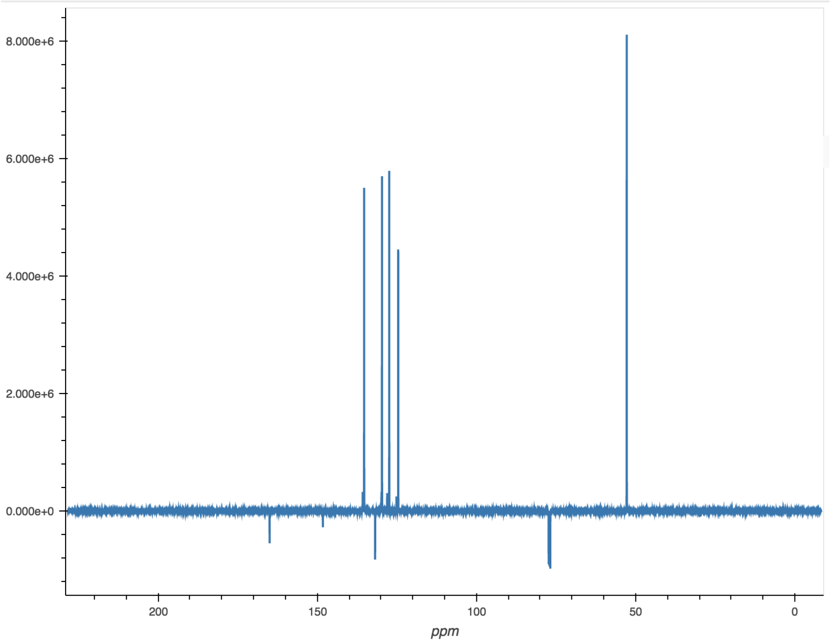


Figure 2.3: Carbon Spectrum Example

2.2 NMR Analysis Packages

This section presents several libraries that support operations on NMR data. Regarding this project, the most useful actions needed are: reading and parsing the NMR data and identify peaks on NMR data.

2.2.1 SPIKE

SPIKE (Spectrometry Processing Innovative KErnel) [9][10][11] is a Python library that allows the processing, displaying and analysis of datasets obtained from various Fourier-Transform (FT) spectroscopies. It currently supports the processing of 1D and 2D FT spectroscopies, implementing Real, Complex and HyperComplex n-dimensionnal Fourier Transform, and many other functionalities.

SPIKE can handle 1D and 2D NMR spectra. Although it has processing and analysing functions for the spectra, the library was created for generic usage, instead of dedicated NMR analysis. Therefore, some primary analysis functions (e.g., integral calculation and solvent filtering) are missing. Finally, the documentation of the package is incomplete, making it harder to understand its complete functionality and how to use it.

2.2.2 Nmrglue

Nmrglue [12] is a module for working with NMR data in Python. Nmrglue has the ability to read, write and convert various NMR file formats, such as Bruker, NMRPipe and Sparky. The files, which are represented in Python as dictionaries of spectral parameters and NumPy [13] ndarray objects, can be easily examined, modified and processed as desired.

Nmrglue provides a number of functions for processing NMR data such as apodisation, spectral shifting, Fourier and other transformations, baseline smoothing and flattening, and linear prediction modeling and extrapolation. In addition new processing schemes can be implemented easily using the nmrglue provided functions and the multitude of numerical routines provided by the NumPy and SciPy [14] packages. Nmrglue can be used to analysis NMR data, with routines to perform peak picking, multidimensional lineshape fitting (peak fitting), and peak integration provided within the package.

2.2.3 Nmrpro

Nmrpro [15] is a Python package for reading and processing different types of NMR spectra. It is a high-level wrapper of the Nmrglue package that lets developers handle NMR spectra more easily. Nmrpro has a unified representation of spectra. It reads NMR files, regardless of their format, to an NMRSpectrum object. Since all NMRSpectrum objects have the same structure, users do not have to concern about any format-specific parameters / header values. Furthermore, NMRPro adds implicit indexing using units, such as “ppm”, “hz” or “ms”. We can use these units to subset the spectrum similar to numpy.ndarray.

2.3 Plot Libraries

This section discusses various plotting libraries, what advantages each one has and what are their limitations. For this project, we need a plotting library that works well on the browser, handles large datasets and provides great interactivity.

2.3.1 Matplotlib

Matplotlib [16] is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook and web application servers.

Despite being fifteen years old, Matplotlib is still the most widely used library for plotting in the Python community. However, even though Matplotlib is a very capable package, it is designed for small to mediumscale visualisations. It is also not designed for interactive plots. This can be solved using the mpld3 project. However, mpld3 is built upon the foundation of HTML's SVG, which is not particularly well-suited for large datasets. Plots with more than a few thousand elements will have noticeably slow response for interactive features.

2.3.2 Plotly

Plotly [17] is an interactive, browser-based graphing library for Python. Plotly is a high-level, declarative charting library and it ships with over 30 chart types, including scientific charts, 3D graphs and statistical charts. It offers some charts that are not available in most libraries, like contour plots, dendrograms, and 3D charts.

On the other hand, Plotly does not involve jQuery but JavaScript only. This might causes problems in cross-browser compatibility and raises concerns about long-term support of the library. Moreover, it does not use HTML5 canvas element, therefore, it might raise concern on usage with old web browsers.

2.3.3 Bokeh

Bokeh [18] is a Python interactive visualisation library that targets modern web browsers for presentation. Its goal is to provide elegant, concise construction of novel graphics in the style of D3.js [19], and to extend this capability with high-performance interactivity over very large or streaming datasets. Bokeh can help anyone who would like to quickly and easily create interactive plots, dashboards, and data applications.

The main disadvantage of Bokeh library is that it is undergoing a lot of development. Therefore, newer versions of the library might not be entirely compatible with the code we write today. Additionally, it has relatively less visualisation options, when compared to D3.js [20].

2.4 Similar Systems

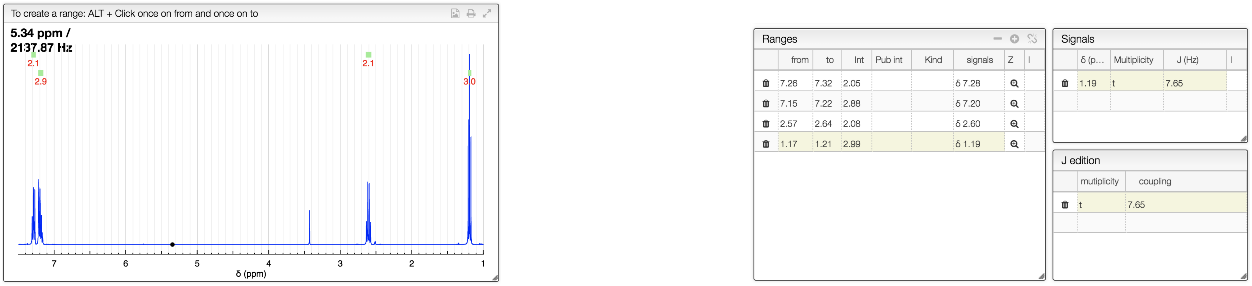
2.4.1 Mestrelab Mnova

Mestrelab Mnova [21] is a software suite to analyse and process chemical data. Mnova is a cutting edge NMR data processing and presentation package with many features, like spectral estimation (NMR prediction) built in. However, Mnova is a quite expensive application suite which will require a lot of money to obtain a license, especially from individuals. Furthermore, Mnova is a desktop based application which is not portable, and harder to maintain (users need to update the application).

2.4.2 C6H6

C6H6 [22] is an NMR repository which stores raw NMR data and provides a set of tools for spectra analysis and processing. It provides a web-based solution for storing, sharing, analysing, and interacting with NMR data. Its purpose is to provide assistance to improve the review process of publications by creating a streamline into the publication pipeline. The project is open source under the MIT licence.

The analysis tool provides a plot which shows the spectrum and several tables that allow the manipulation of data (see Fig. 2.4). The plot viewer displays the spectrum and provides only zoom capabilities. In order to enable data manipulation, users must add new row to the "Ranges" table and manually fill all the information. Finally, the analysis tools are coupled with the NMR data which makes them hard to decouple and reuse for NOMAD.



To create a range: ALT + Click once on from and once on to

Ranges

	from	to	Int	Pub int	Kind	signals	Z	I
🗑	7.26	7.32	2.05			δ 7.28	🔍	
🗑	7.15	7.22	2.88			δ 7.20	🔍	
🗑	2.57	2.64	2.08			δ 2.60	🔍	
🗑	1.17	1.21	2.99			δ 1.19	🔍	

Signals

	δ (p…	Multiplicity	J (Hz)	I
🗑	1.19	t	7.65	

J edition

	mutiplicity	coupling
🗑	t	7.65

C6H6 Plot viewer

C6H6 data manipulation

Figure 2.4: C6H6 NMR Analysis Tools

2.4.3 NMRShiftDB2

nmrshiftdb2 [23] is a database for organic compounds and their NMR spectra. It provides spectrum prediction as well as searchability on spectra, structures and other properties. The nmrshiftdb2 application is open source and the data is published under an open content license. It currently supports only 1D NMR spectrum data (e.g., 13C, 1H) but its architecture allows future extensions. The nmrshiftdb2 project does not provide functionality for manual analysis of the spectrum. The data are semi-automatically analysed by using an NMR shift prediction application and then presenting the result to reviewers.

2.5 Usability Test

Usability does not exist in any absolute sense. It is defined as the ease of use and learnability of a human-made object such as a tool or device, but it can only be expressed with reference to specific context. For example, usability in software engineering is the metric that defines the degree that a software application can be used by particular users to achieve its objectives with efficiency, satisfaction and effectiveness.

The abstraction of usability as a generic term means that there are no defined measures to calculate the usability. There are various usability tests, each more suitable for particular context. Each test traditionally uses a standardised questionnaire that asks the same questions in an identical format and records the responses in a uniform manner. Standardised questionnaires provide several advantages, such as reliability, validity, sensitivity and objectivity. There are various tests that have been developed to test the usability of a software system which are more suitable based on various constraints such as the type of the system and the resources availability. Popular standardised questionnaires include: Software Usability Measurement Inventory (SUMI), Post-Study System Usability Questionnaire (PSSUQ), and the System Usability Scale (SUS).

SUMI [24] is a questionnaire method for analysing products or prototypes in terms of usability and quality of use. It is a rigorously tested method of measuring software quality from the end user's point of view. SUMI questionnaire consists of 50 questions that measure users' perception of the efficiency, affect, helpfulness, control and learnability of a system. It is also backed up by a report generator which refers to a large standardisation database. The use of SUMI requires a license that costs approximately $700 per month.

PSSUQ is another usability quantification survey which uses Likert scale, a scale used to represent people's attitudes to a topic, to describe how much users agree or disagree with a statement. It is formed by 16 questions which measure users' perceived satisfaction with a product or system [25].

SUS [26] is a Likert-type survey that provides a reliable, low-cost scale that can be used for system usability. It is a simple, short questionnaire which includes 10 questions that must be answered by the users of the system. Recent psychometric analysis show that items 4 and 10 reliably measure the dimension of perceived "learnability" [25]. The SUS questionnaire was decided to be used during this project as it is robust with a small number of participants [27] and has the distinct advantage of being technology agnostic, meaning it can be used to evaluate a wide range of hardware and software systems [28]. The ten template questions used are given below:

1. I think that I would like to use this system frequently.

2. I found the system unnecessarily complex.

3. I thought the system was easy to use.

4. I think that I would need the support of a technical person to be able to use this system.

5. I found the various functions in this system were well integrated.

6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The odd numbered questions state a positive expression for the system while the even numbered questions a negative one. Each question is graded on a scale from 1 to 5 where 1 means “Strongly Disagree” and 5 means “Strongly Agree”. To interpret the results of the evaluation we subtract one from the user response for the odd numbered questions and subtract the user responses from 5 for even numbered items. This scales all values from 0 to 4, with four being the most positive response. Then, the values are added together to create a score between 0 and 40 and then multiplied by 2.5 we end up with a score in the range 0-100. The general guideline on the interpretation of SUS score is listed in Table 2.1.

Descriptive Statistics of SUS Scores for Adjective Ratings

No.	Rating	Count	Mean	Standard Deviation
7	Best imaginable	1	100	N/A
6	Excellent	69	85.58	9.473
5	Good	90	72.75	10.56
4	OK	36	52.01	12.13
3	Poor	15	39.17	12.38
2	Awful	0	N/A	N/A
1	Worst imaginable	1	25	N/A

Table 2.1: SUS Summary Statistics for Adjective Ratings [29]

3. Software Engineering Process

3.1 Methodology

3.1.1 Version Control

The project is completely independent of the current work done from the NOMAD team. Thus, the project was built outside of the NOMAD repository. I used the Git version control system and created a repository on GitHub hosting service. GitHub provides a reliable and user-friendly platform for a project development.

For the database, I used the Flyway open source database migration tool. All scripts used to create the database are stored in a folder and using Flyway we can migrate, undo and validate changes. Additionally, Flyway provides repair capabilities in case something goes wrong.

3.1.2 Build Tools

Python

For the NMR analysis tool and the UI tool the“venv” module was used to create lightweight virtual environments with their own site directories isolated from system site directories. The virtual environment has its own Python binary file and its own independent set of installed Python packages.

The *pip freeze* command returns the installed packages in Python requirements format. The dependencies where stored into a text file, called “requirements.txt”, allowing developers to easily install them to a different virtual environment.

Java

The backend of the NMR spectrum search engine was developed in Java. Both backend and frontend of the search engine have dependencies, which are managed using the Apache Maven project management and comprehension tool [30]. The client-side libraries are packaged into JAR archive files, called WebJars. With WebJars, developers can explicitly and easily manage web dependencies in their applications and the build tools will be aware of the web libraries versions.

3.2 Tools and Technologies

3.2.1 NMR Spectra Processing Library

The Nmrglue package is used for the analysis of NMR data during this project. The library is developed to be used with NMR data, making all of its features applicable to the project requirements. Nmrglue is also more customisable, compared to NMRPro, which gives greater flexibility and more advanced options. Finally, NMRPRo is a fork of Nmrglue which makes it dependable and harder to develop.

3.2.2 Plotting Library

The Bokeh library is used for the implementation of the NMR Analysis tool and the UI tool of the search engine. The Bokeh library provides great interactivity with the browser and high performance on large datasets. Even though 1D NMR data are quite small (few megabytes), multi-dimensional NMR spectra size increases significantly. Although the project currently does not support multi-dimensional NMR spectra, it is a main task for future versions. Therefore, this property of the library is essential. Additionally, Bokeh core library allows developers to easily extend it and build custom applications. This is inevitable due to the domain specific application we need.

The diagram below (see Fig. 3.1) explains the process flow of the Bokeh library and how it manages to represent the data on the browser. The Bokeh library consists of two standalone applications. One is the Bokeh backend library. It currently has multiple bindings in Python, R, Lua and Julia. The Python library is used in regards to this project, as it is the main library under development. Bokeh creates the plots on the server side and creates a JavaScript Object Notation (JSON) packet containing the generated plots. On the frontend, BokehJS receives this JSON file as input and presents the data to the browser. This creates modular environment by separation of concerns. This design has great advantages. For instance, heavy computations can be done in the backend while small interactions with the plot can be done on the frontend without sending any data to the server.

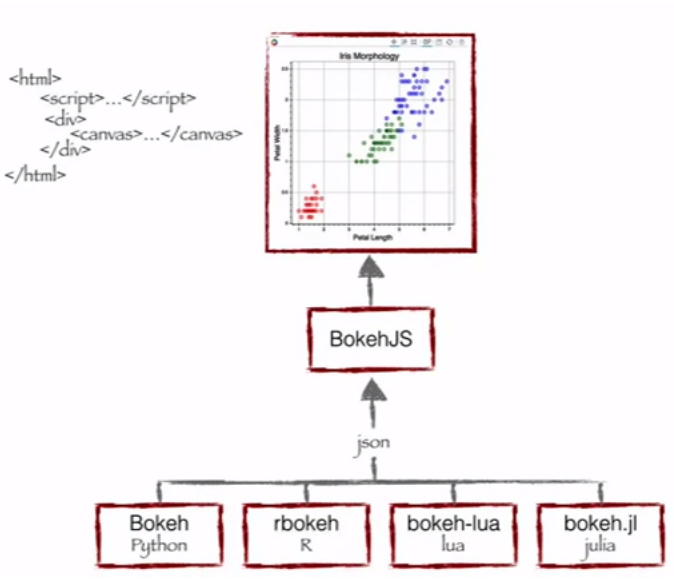


Figure 3.1: Bokeh Library Process Flow Diagram [20]

Bokeh consists of several components. The main component is the *document* component, which is a container for Bokeh Models (see Fig. 3.2). The *Plot* model is required in order to represent a plot of the NMR spectrum. The plot contains a toolbar which consists of a set of tools that interact with the plot. Another important component, essential for this application, is the widget component. Widgets are interactive controls which can be used with the Bokeh server to run arbitrary Python code, enabling complex applications.

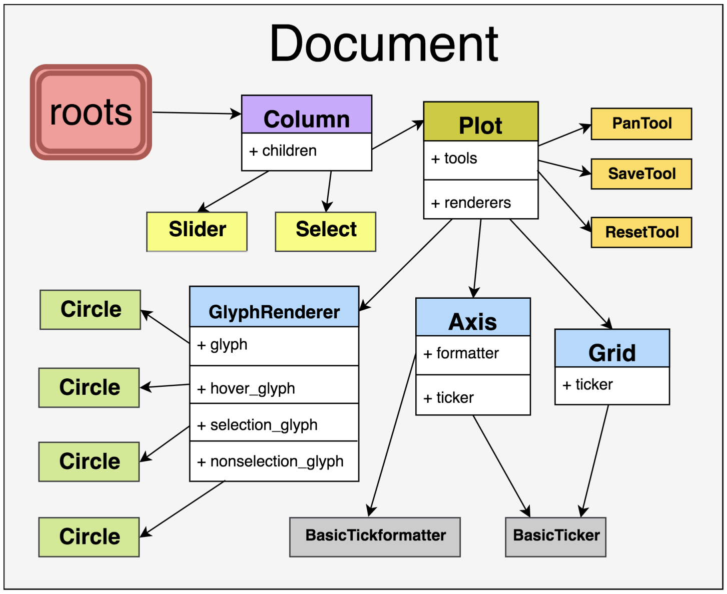


Figure 3.2: Bokeh Document Container [18]

3.2.3 Spring Boot - Thymeleaf

The NMR Spectrum Search Engine is developed in the Java programming language. NOMAD is based on Java and implementing the search engine in the same language minimises the integration effort. I am also familiar with Java web-application development which will make the development of the search engine easier and faster.

Spring is a popular framework in Java for building web and enterprise applications. Unlike other frameworks that focus on application-specific areas, Spring addresses a wide range of domains. It also simplifies integration with other Java frameworks like Java Persistence API (JPA).

Spring Boot is a project based on Spring framework that makes it easy to build standalone, production grade applications. It needs minimal configuration to start running but it also allows to override the default options. There is also no need to deploy WAR file as it contains an embedded Java Servlet Container such as Apache Tomcat or Jetty. Finally, Spring Boot provides production-ready features such as health checks, metrics and externalised configuration.

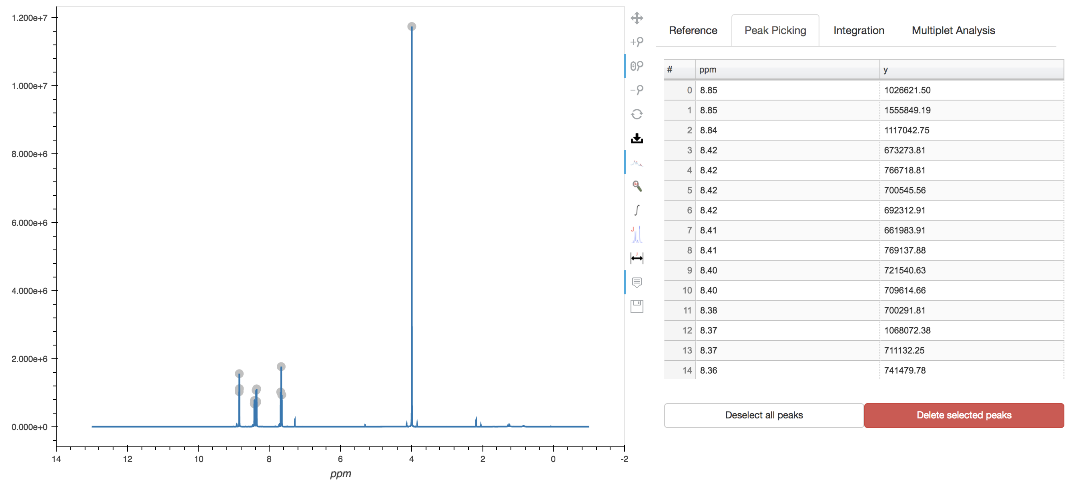
For the front-end of the web-application, the Thymeleaf template engine is used. Spring supports integration with Thymeleaf requiring no configuration to get it running. Thymeleaf code is more like HTML-ish view, compared to JavaServer Pages (JSP), creating more readable and understandable code. Finally, the Dialect (or Spring Standard Dialect) Expression Language is much more powerful than JSP Expression Language defining a set of features that cover most cases.

4. Design

The system presented in this scope was designed taking consideration of the human-computer interaction (HCI) principles such as consistency, simplicity, minimalistic design and error tolerance. System consistency is enforced by implementing a similar design for all components. In addition, each function requires similar sequence of actions. System simplicity is applied by automating the operations as much as possible and require minimal interaction by the user. Minimalistic design is fulfilled by creating the minimum number of components as possible whilst keeping the system functional with full capabilities. Error tolerance is implemented by minimising the room for error in the design of the components and validating the users' input values.

4.1 NMR Analysis Tool

The system design started by organising the components of the tool. Each component has several information, so a generic way was designed for users to review and modify them. For instance, the peak picking tool provides a list of the already found peaks so users can review and delete them if needed. To create a consistent design, each module was created inside a panel and each panel was placed into tabs next to the plot (see Fig. 4.1). This way, users can easily switch between the different components by changing the tab. Each module has a table that contains the information retrieved so far and users are able to inspect and modify them.



#	ppm	y
0	8.85	1026621.50
1	8.85	1555849.19
2	8.84	1117042.75
3	8.42	673273.81
4	8.42	766718.81
5	8.42	700545.56
6	8.42	692312.91
7	8.41	661983.91
8	8.41	769137.88
9	8.40	721540.63
10	8.40	709614.66
11	8.38	700291.81
12	8.37	1068072.38
13	8.37	711132.25
14	8.36	741479.78

Figure 4.1: NMR Analysis Tool Layout

The system consists of four components: a tool that allows users to change the chemical shift reference, a tool for peak picking on the NMR spectrum, a tool for integral calculation on given areas and a tool that allows users to perform multiplet analysis. All components are communicating with the backend of the application to perform heavy computations and to synchronise the other components.

The chemical shift referencing tool was firstly created using two input fields where users would write the required information (see Fig. 4.2). At the first input box, users would put the current chemical shift on the graph and at the second input box, the position where they would want it to move. This, however, is complicated as users should inspect the NMR spectrum and write the chemical shift manually. This can also lead to mistakes by accidentally writing an incorrect shift. To improve this, a tool that picks the current position on the plot and copies it to the old chemical shift input field was introduced and the old chemical shift input field was disabled (see Fig. 4.3). This automates the process and eliminates the issue discussed above. After the data selection, the information is sent to the server which dynamically shifts the plot.

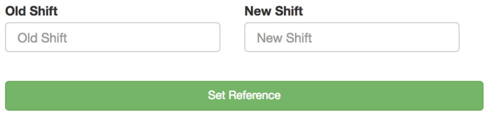


Figure 4.2: Manual Referencing Tool

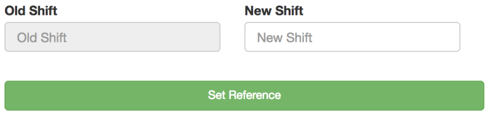


Figure 4.3: Automatic Referencing Tool

The peak picking function requires two tools, a tool that allows users to select an area of peaks on the plot and another to manually select individual peaks. The area selection tool should allow users to select an area on the plot and send the dimensions of the selected area to the backend, where an automatic process identifies the peaks in the selected area. The tool's design was initially inspired by the Bruker's TopSpin software package (see Fig. 4.5). The tool was designed as a box select tool, that allowed users to select a rectangular area where the automatic process would run (see Fig. 4.4). This design covers the requirement and it is easy to use. The design implemented provides the same functionality as the TopSpin's peak picking tool.

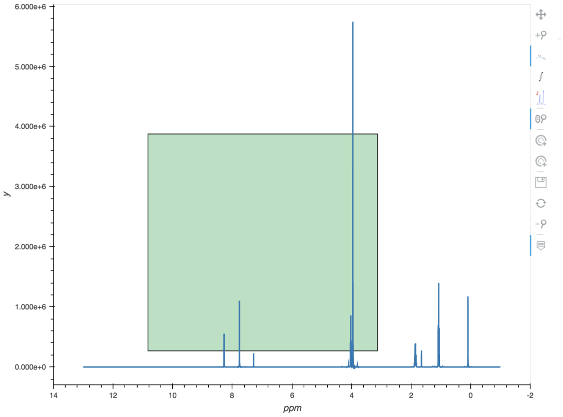


Figure 4.4: Box Select Peak Picking

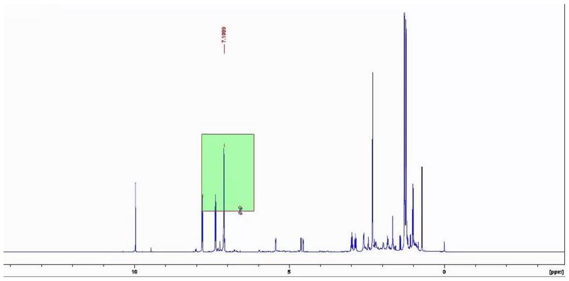


Figure 4.5: TopSpin Peak Picking Tool

However, there are many noises at the x-axis (near y=0) that are not valid peaks and users should not pick them. Therefore, in order to identify peaks in both positive and negative values, users must use the tool twice. The design of the tool changed to mitigate the issue with the noises and to make the process faster. The new design of the tool allows users to select area between two x-values [x1, x2] and an offset from the x-axis ($([\infty, y], [-y, -\infty])$) (see Fig. 4.6). This design allows users to select peaks on both positive and negative values at the same time, without worrying about the noises near the x-axis. This design is inspired by the Mnova software suite (see Fig. 4.7). Mnova's peak picking tool provides the same functionality but, however, it does not identify the negative peaks, unless users change the default settings.

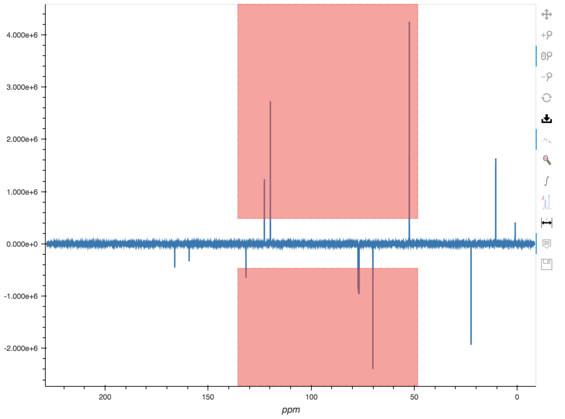


Figure 4.6: Positive-and-Negative Peak Picking Tool

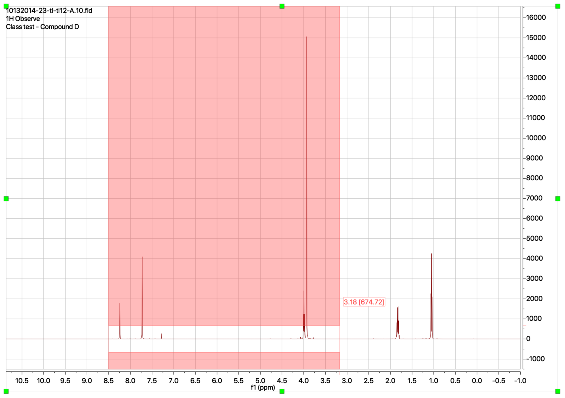


Figure 4.7: Mnova Peak Picking Tool

The tool that allows manual peak selection was firstly designed using an input field that allowed users to enter the chemical shift (x-position) of the peak. As discussed earlier, this requires a lot of manual configuration by the users. It might also lead to errors by accidentally writing the chemical shift wrong. Thus, the design of the tool was improved to allow users select the peak and display the value to the input field (see Fig. 4.9). This eliminates the issue with writing the correct chemical shift value. It also allows users to verify that the chemical shift selected is correct. On the other hand, this double check can be eliminated as peaks can be deleted at a later stage. Therefore, the tool was fully automated by automatically selecting the peak when users select a point on the plot. This makes the manual peak picking process faster.

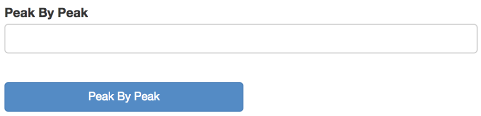


Figure 4.8: Manual Peak by Peak Tool

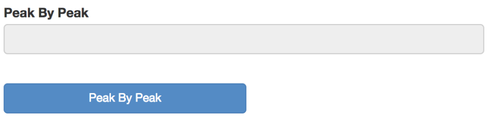
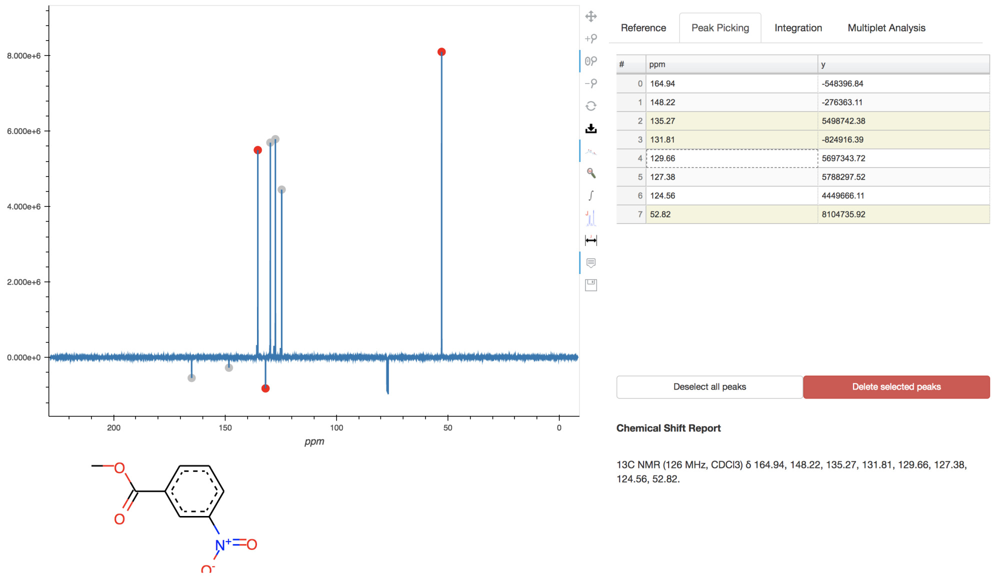


Figure 4.9: Peak by Peak Tool

When the data are selected, they are sent to the server-side of the application which calculates the peaks. When the user uses the box picking mechanism, the application finds the peaks in the selected area using an automatic process. Otherwise, when the user uses the manual peak picking tool the system finds the peak that is closest to the user's selected point. After the selection process is done, the peaks are added to a table which contains the coordinates of all the peaks. The peaks are also displayed in the plot as grey circles. When a user selects one or more peaks from the table, these are turned into red circles to distinguish between the non-selected ones (see Fig. 4.10). Finally, users are able to select multiple peaks from the table and delete them if necessary.



#	ppm	y
0	164.94	-548396.84
1	148.22	-276363.11
2	135.27	5498742.38
3	131.81	-824916.39
4	129.66	5697343.72
5	127.38	5788297.52
6	124.56	4449666.11
7	52.82	8104735.92

Chemical Shift Report

13C NMR (126 MHz, CDCl3) δ 164.94, 148.22, 135.27, 131.81, 129.66, 127.38, 124.56, 52.82.

Figure 4.10: Peak Picking Component

Mestrelab Mnova software suite displays the peaks on the plot having a small line above each peak and writing the peak position at the top of the plot (see Fig. 4.11). This allows users to check which peaks have been already processed and find their position quickly. Similarly, the TopSpin software displays the peaks in an almost identical design (see Fig. 4.12). I decided not to use the same design. As mentioned earlier, the peaks are displayed using a grey circle which turns red once the peaks are selected in the table (see Fig. 4.10). This design is useful for quickly spotting the peaks but it doesn't show their actual position. In order to find the position of a specific peak, users must either hover over the peak or check the peaks' table. This design was preferred as it keeps the plot simple and clean.

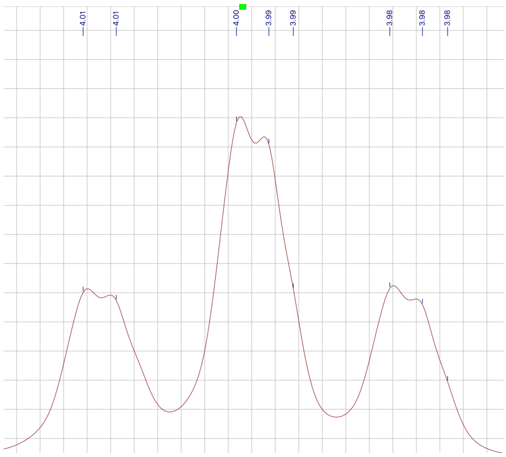


Figure 4.11: Mnova Software Peaks

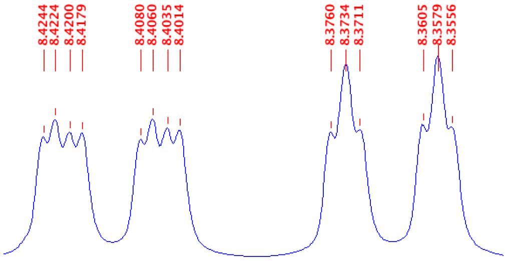


Figure 4.12: TopSpin Software Peaks

The integral computation tool should allow users to select an area which they want to calculate the integral between the curve and the x-axis. The tool design allows users to select a vertical region between two x-values [x1, x2] (see Fig. 4.13). When users select an area, the x-values are sent to the backend and the integral value between the curve in this space and the x-axis is calculated. The values' range and the integral value are added into the integration table and presented to the user. Users can click on one or more entries from the table and the selected regions will be displayed on the plot. Finally, users are able to modify the values by double-clicking at an entry row or delete them (see Fig. 4.14).

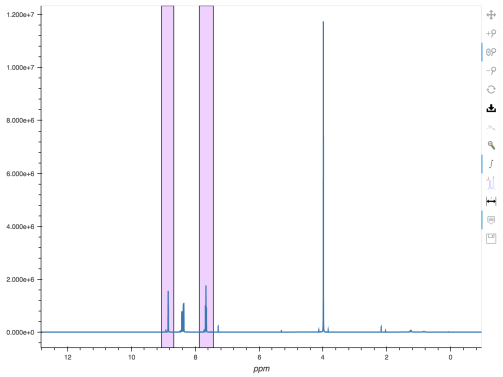
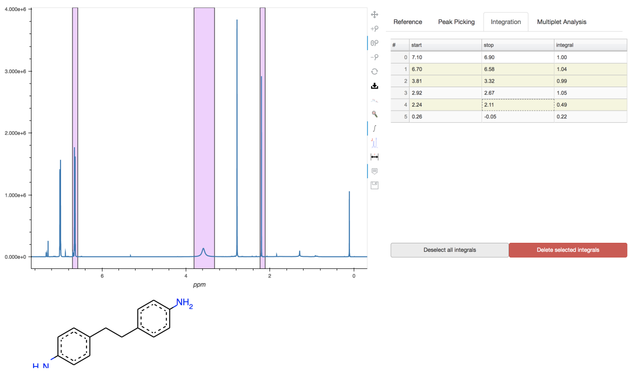


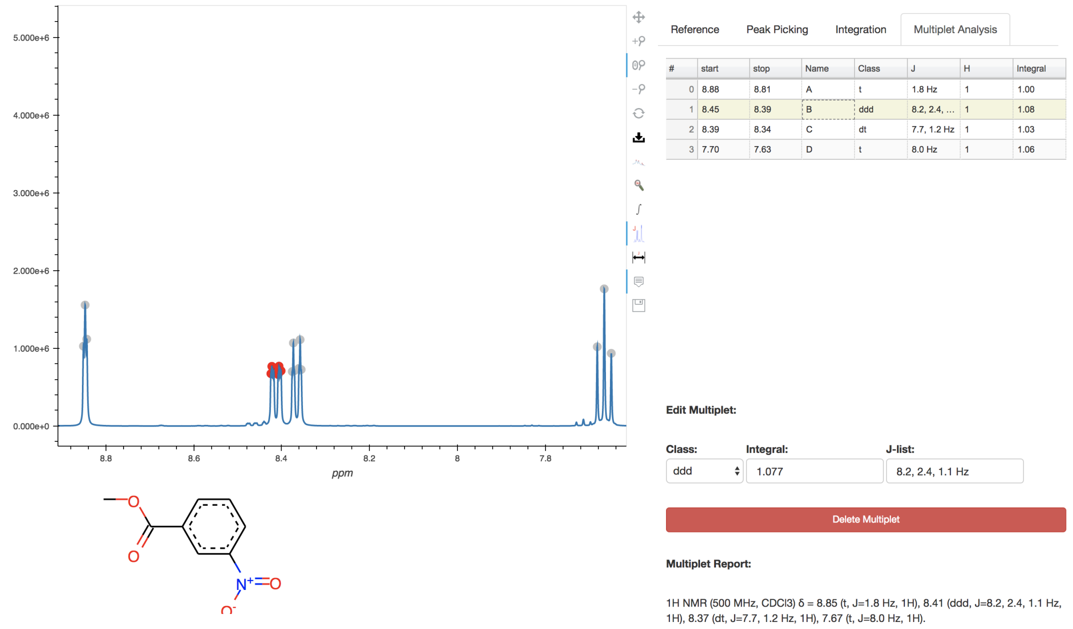
Figure 4.13: Integration Tool



#	start	stop	integral
0	7.10	6.90	1.00
1	6.70	6.58	1.04
2	3.81	3.32	0.99
3	2.92	2.67	1.05
4	2.24	2.11	0.49
5	0.26	-0.05	0.22

Figure 4.14: Integration Component

The multiplet analysis tool uses the same design as the area selection tool of peak picking. It allows users to select an area which is then sent to the server. The server-side of the multiplet analysis component has to perform several actions: peak picking, calculate integral, calculate the coupling constants and calculate the multiplicity class of the multiplet. The system currently supports the simple multiplicity classes (e.g., singlet, doublet, triplet, etc.) and many complex ones such as doublet of doublets, doublet of triplets etc. (see Section. 2.1). As NMR spectra can contain many impurities, in order to calculate the multiplicity class a heuristic operation that predicts the multiplicity class considering the noises on the spectrum was created. The prediction function creates a range of acceptable values of the next peak height. The bigger the error value, the less false positives are created but more false negatives are introduced. After some experimenting with several NMR spectra, I ended up with an error value equal to 1000000. Finally, even if the prediction is wrong, users can manually change it.



#	start	stop	Name	Class	J	H	Integral
0	8.88	8.81	A	t	1.8 Hz	1	1.00
1	8.45	8.39	B	ddd	8.2, 2.4, …	1	1.08
2	8.39	8.34	C	dt	7.7, 1.2 Hz	1	1.03
3	7.70	7.63	D	t	8.0 Hz	1	1.06

Edit Multiplet:

Class: ddd **Integral:** 1.077 **J-list:** 8.2, 2.4, 1.1 Hz

Multiplet Report:

1H NMR (500 MHz, CDCl3) δ = 8.85 (t, J=1.8 Hz, 1H), 8.41 (ddd, J=8.2, 2.4, 1.1 Hz, 1H), 8.37 (dt, J=7.7, 1.2 Hz, 1H), 7.67 (t, J=8.0 Hz, 1H).

Figure 4.15: Multiplet Analysis Component

The toolbar of the Bokeh library consists of all the tools created and some predefined tools for zooming, panning and saving an image of the plot. However, the tools are ordered in a predefined order based on their action (e.g., click, select, inspect) (see Fig. 4.16). This makes the use of the tool complicated as users would spend a lot of time to locate the correct tools. Therefore, a new toolbar was created, which sets the tools in a custom order (see Fig. 4.17). This increases the confidence of the users of what each tool does. Finally, switching between the tabs automatically changes the selected tool to the appropriate one. Similarly, activating a tool from the toolbar, automatically changes the active tab. This makes the use of the tool easier as users won't have to change both the tool and the tab.

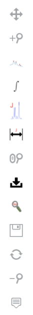


Figure 4.16: Bokeh Toolbar - Predefined Order

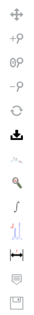


Figure 4.17: Bokeh Toolbar - Custom Order

After implementing all the components of the tool, I used the right-click gesture to provide alternative way for users to perform an action (see Fig. 4.18). All the tools are accessible in a list at right-clicking. This helps new users to quickly find a specific tool they need.

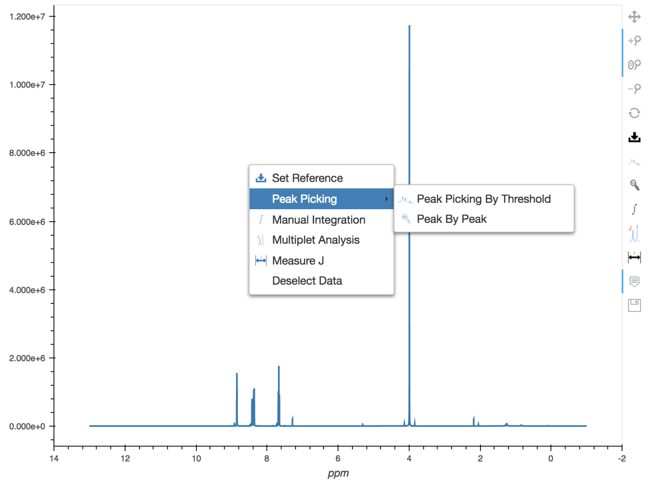
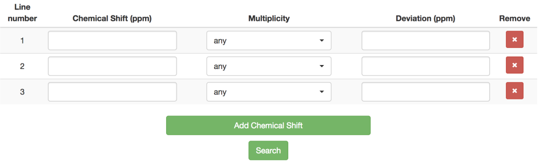


Figure 4.18: Right-Click Gesture

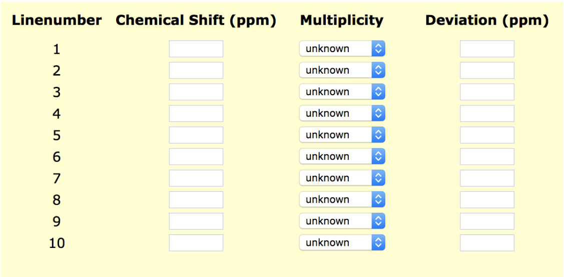
4.2 NMR Spectrum Search Engine

The NMR Spectrum Search Engine allows users to enter some of the chemical shifts of the compounds they are searching for and search for compounds that have peaks at the positions given. Users should be able to enter a range [x1, x2] that a chemical shift belongs to. I started designing the search engine by creating the input fields where users should be able to enter the chemical shift ranges. The number of inputs should be dynamic, so I created a button that creates an empty input field and a button next to each row to delete it. Each input row allows users to enter the chemical shift position, the multiplicity class and an error range (deviation). For instance, if the chemical shift is equal to 1.0 and the deviation is equal to 0.5 then the chemical shift must be between 0.5 and 1.5. This design was inspired by an NMR spectra similarity search engine developed at the University of Vienna [31]. Figure 4.19 shows the design of this project's NMR Spectrum search engine, while Figure 4.20 shows the design of the NMR Spectra similarity search engine. The similarity search engine from the University of Vienna has a fixed number of parameters, where some of them can be empty but there is no way to add more. It also does not perform automatic search, but sends the results by email. When users need to remove an entry, they have to clear all input fields. On the other hand, the search engine implemented as part of this project allows unlimited number of parameters, by dynamically adding and removing them. Also, when the parameters are submitted, the search engine returns the results to the user. Finally, users can simply remove a parameter by deleting it.



Line number	Chemical Shift (ppm)	Multiplicity	Deviation (ppm)	Remove
1		any		✕
2		any		✕
3		any		✕

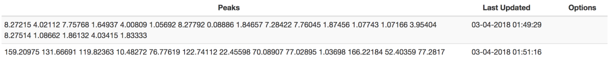
Figure 4.19: NMR Spectrum Search Engine Based on Multiplet Information



Linenumber	Chemical Shift (ppm)	Multiplicity	Deviation (ppm)
1		unknown	
2		unknown	
3		unknown	
4		unknown	
5		unknown	
6		unknown	
7		unknown	
8		unknown	
9		unknown	
10		unknown	

Figure 4.20: NMR Spectra Similarity Search - University of Vienna

When users perform the search, the engine searches the database and finds the list of spectrum that contain all of the given peaks. Then, the spectrum information are sent to the front end. Initially, the raw information were presented to the user, containing the complete list of peaks of each spectrum and when the data were captured (see Fig. 4.21). For chemistry perspective, the raw data are not very useful. Therefore, the design changed to display the results in a grid containing the chemical compounds images (see Fig. 4.22). When users click on a compound image they are able to preview the compound spectrum (see Fig. 4.23).



Peaks	Last Updated	Options
8.27215 4.02112 7.75768 1.64937 4.00809 1.05692 8.27792 0.08886 1.84657 7.28422 7.76045 1.87456 1.07743 1.07166 3.95404 8.27514 1.08662 1.86132 4.03415 1.83333	03-04-2018 01:49:29	
159.20975 131.66691 119.82363 10.48272 76.77619 122.74112 22.45598 70.08907 77.02895 1.03698 166.22184 52.40359 77.2817	03-04-2018 01:51:16	

Figure 4.21: Peaks list

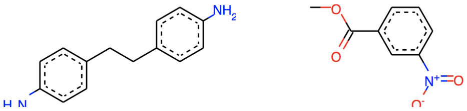


Figure 4.22: Images of Compounds

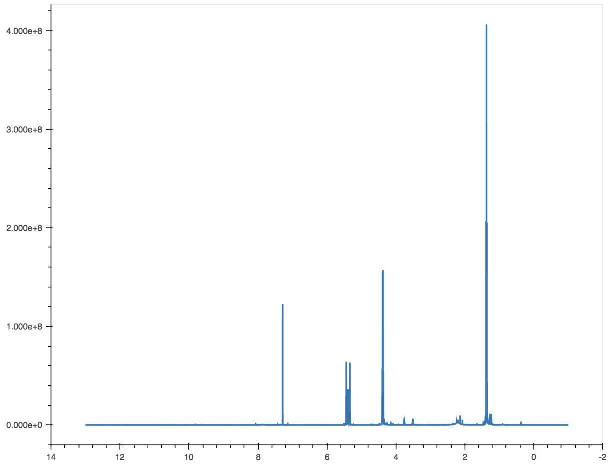


Figure 4.23: Spectrum Preview

After the search engine implementation, I explored various ways that would allow users specify the search criteria easily and perform search on the NMR spectra faster. A UI tool was designed and implemented that allows users to interact with a plot in order to set the search parameters (see Fig. 4.24). Users are able to zoom in the plot, select a range of values for a chemical shift or delete an already selected range. Selection of a range does not allow users to specify the multiplicity type, which by default is “any”. The purpose of the UI tool is to provide an alternative, simple, easy and fast way to search the spectrum using chemical shifts. When a range is selected, the system performs a search on the specified ranges. The results are then shown under the plot as compounds. When the user selects a compound then the spectrum of the compound is shown on the plot, along with the selected areas (see Fig. 4.25).

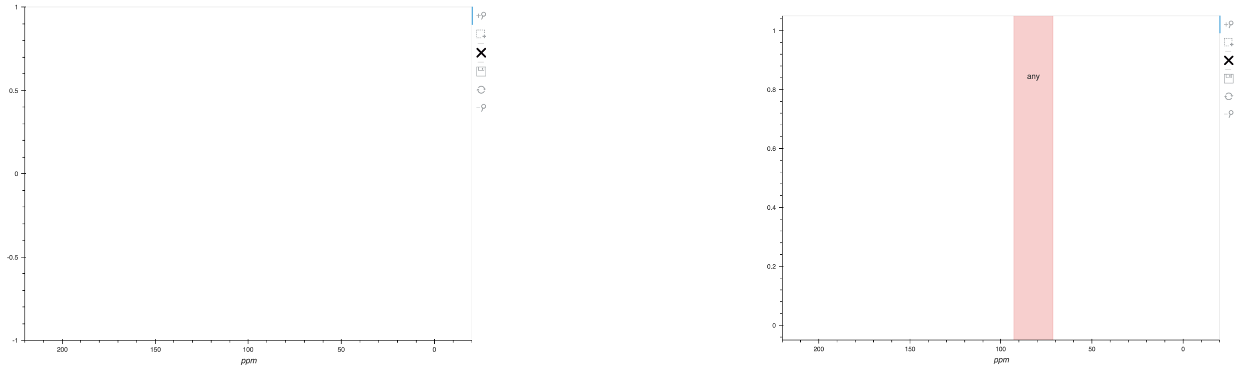


Figure 4.24: UI Tool

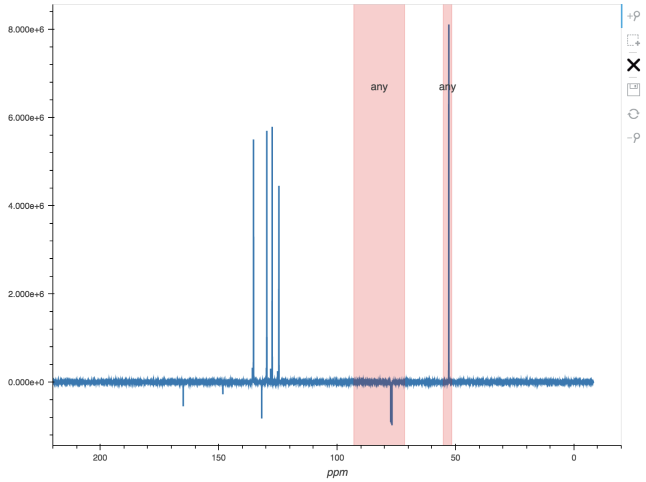


Figure 4.25: Spectrum Preview

4.3 Database

The database stores general information about the spectra and the multiplet information generated using the NMR analysis tool. The database was designed to reduce the duplication of data and ensure referential integrity. This was done by using the third normal form (3NF) to normalise the database design.

The information is split into two entities. One entity is the spectrum containing the spectrum identifier, and the time stamp when the spectrum was last modified. In order to allow multiple analysis of the same spectrum, as users might generate different results, a unique identifier for each spectrum entry in the table is created. The other entity is the chemical shift information. Each chemical shift has a unique identifier, the chemical shift position on the spectrum, the multiplet class of the shift and the spectrum id referencing the spectrum associated with the chemical shift. Chemical shifts are a logical extension of the spectrum. Therefore, when a spectrum is updated or deleted then its chemical shifts are updated or deleted as well. The main purpose of the database is to allow users search for NMR spectra using chemical shifts efficiently. To improve the speed of the database, additional indexing based on the chemical shifts was created. This database design allows users to easily search the spectrum based on the chemical shifts.

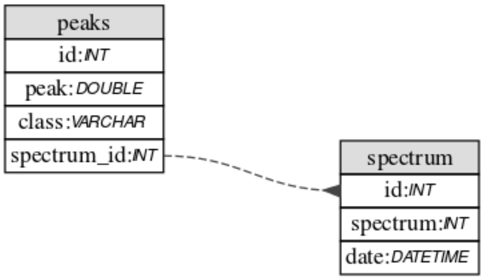


Figure 4.26: Database ER Diagram

5. Implementation

5.1 NMR Analysis Tool

The NMR Analysis Tool was built in Python using the Bokeh library to plot the data. In order to use the computing power of the server, the Bokeh Server was used, which creates a web application that communicates to a Tornado backend server using WebSockets. The flow of the program is based on events, as the backend responds to the frontend events. Additionally, the components of the tool are logically interconnected. Therefore, the observer pattern was utilised to exchange messages between components. The observer pattern allows the creation of one-to-many dependencies between the components without making the objects tightly coupled. It also ensures that when the state of one object changes, the dependent objects are updated automatically.

The NMR analysis tool development started by parsing the raw NMR data, using the Nmrglue package. Nmrglue supports the Bruker data format which is used by the School of Chemistry. The nmrglue package reads the data and returns a dictionary with the experiment parameters and a numpy.ndarray with the NMR data, which corresponds to the y values of the plot. Nmrglue also provides functionality that calculates the ppm scale by passing the properties of the experiment, which corresponds to the x-values of the plot. A basic plot in Bokeh library was created using these data, which displays the NMR spectrum and some basic tools (zooming, panning and saving an image of the plot).

An important issue I faced from the beginning of the components' implementation is that Bokeh tools do not support communication with the server. In order to send data to the server, the user must trigger a widget which will then trigger some action on the server-side. Therefore, in order to send data from the plot to the server, the tools must select the information from the plot and then widgets send the selected data to the server. This is very cumbersome and requires more effort by the users. A more user-friendly approach is to automatically send the information to the server after selecting data on the plot. In order to achieve this, the tools and widgets were combined into one component. This was implemented by creating hidden widgets on the web application and triggering them after each operation using JavaScript callbacks. Additionally, new widgets had to be introduced that extend the current widgets and store additional information. A dictionary object was introduced into each widgets which would be sent to the backend when the widget would be triggered. Finally, the tools would store the information in the widgets' dictionary using BokehJS.

Creating this cooperation between tools and widgets the NMR analysis components implementation was feasible. The chemical shift reference component sends the current and the new chemical shift to the backend. The backend verifies that the variables received are valid and performs the transformation. Finally, it notifies all the observers in order to update their data, as shifting the plot affects all the other components.

The peak picking component development consists of the creation of the two tools described in the design section. The area selection tool sends the selected area coordinates to the backend. The server performs peak picking to the NMR data using the Nmrglue package. The default method for peak picking is called *connected*. The connected method finds all nodes which are above a given threshold and connected to the initial point. For finding all segments the scipy.ndimage.label function is used for speed. This algorithm is not very sensitive, therefore, it is more suitable for noisy spectra but it does not identify many correct peaks. The peak picking algorithm was changed to *downward*, a more sensitive algorithm which results to more identified peaks. Downward method uses the flood fill algorithm to find all points connected to an initial node which are above a given threshold and to which a path exists in which each step of the path moves lower in intensity. This can be thought of as all points are accessible by a water drop following downward slopes from the initial node. This might result to more false positives which can then be removed. This approach seems better and more user-friendly than having more false positives and adding them manually. In order to perform the algorithm for both positive and negative values, the peak picking component also stores the opposite values of the NMR data and runs the peak picking algorithm twice. The manual peak picking tool selects a ppm value on the plot and sends it to the server. The server checks if the value received is in the NMR data list and then adds the peak to the peak list. Whenever the peak list changes (either addition or deletion) the backend checks whether the spectrum is carbon or proton. If it is a carbon spectrum, it recalculates the chemical shift report of the spectrum. Otherwise, it notifies the multiplet analysis to recalculate the multiplicity class of the multiplets affected by the change.

The integration tool allows users to select area on the plot and calculate the area between the curve and the x-axis. This is done using the numpy.trapz function which integrates along the given axis using the composite trapezoidal rule. As mentioned earlier, NMR analysis is not interested about the exact value of the integral, but for the proportion between the different integrals. Therefore, the first request that is sent to the backend is stored as the point of reference and set to 1, and the rest of the integrals are analogous to this value. When users change the value of an integral, then the point of reference is recalculated and all the integrals are updated based on the new point of reference. This ensures that the integrals are always proportional to each other.

When users select an area for multiplet analysis, the area coordinates are sent to the backend. Then, the multiplet analysis component performs peak picking using the peak picking component. In order to create a multiplet, at least one peak must be found in the selected area. Then, the component calculates the integral of the area using the integration component and predicts the multiplicity class of the multiplet. The multiplicity class is calculated using the dictionary 5.1. The table attribute describes the height difference between the peaks of the multiplet, *sum* is equal to the number of peaks of the class and j contains a list of pairs of peaks there coupling constant must be calculated (see Section 2.1). The multiplicity prediction algorithm goes through the multiplicity classes and checks if the sum is equal to the number of peaks identified. If so, it recursively goes through the multiplets and checks whether the proportion of heights correspond to the particular class. If none of the multiplicity classes defined is found then the default multiplicity class, called *multiplet*, is used.

Listing 5.1: Multiplicity Classes

```
MULTIPLETS = {
    ’s’:    {’table’: [1], ’sum’: 1, ’j’: []},
    ’d’:    {’table’: [1, 1], ’sum’: 2, ’j’: [[0,1]]},
    ’t’:    {’table’: [1, 2, 1], ’sum’: 3, ’j’: [[0,1]]},
    ’q’:    {’table’: [1, 3, 3, 1], ’sum’: 4, ’j’: [[0,1]]},
    ’p’:    {’table’: [1, 4, 6, 4, 1], ’sum’: 5, ’j’: [[0,1]]},
    ’h’:    {’table’: [1, 5, 10, 10, 5, 1], ’sum’: 6, ’j’: [[0,1]]},
    ’hept’: {’table’: [1, 6, 15, 20, 15, 6, 1], ’sum’: 7, ’j’: [[0,1]]},
    ’dd’:   {’table’: [[1, 1], [1, 1]], ’sum’: 4, ’j’: [[0,1], [0,2]]},
    ’ddd’:  {’table’: [[1, 1], [1, 1], [1, 1], [1,1]], ’sum’: 8, ’j’: [[0,1], [0,2], [0,4]]},
    ’dt’:   {’table’: [[1, 2, 1], [1, 2, 1]], ’sum’: 6, ’j’: [[0,1], [0,3]]},
    ’td’:   {’table’: [1, 1, 2, 2, 1, 1], ’sum’: 6, ’j’: [[0,1], [0,2]]},
    ’ddt’:  {’table’: [[1, 2, 1], [1, 2, 1], [1, 2, 1], [1, 2, 1]], ’sum’: 12, ’j’: [[0,1], [0,3],
        [0,6]]}
}
```

5.2 NMR Spectrum Search Engine

The system is structured using the client-server model. The client and the server communicate through the HTTP protocol. Clients initiate communication sessions with servers which await incoming requests. Figure 5.1 shows a high-level overview of the architecture of the system.

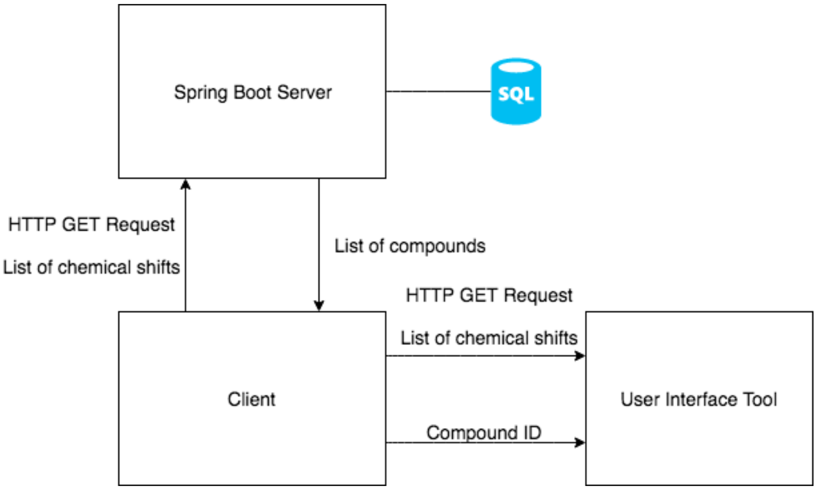


Figure 5.1: NMR Spectrum Search Engine Architecture

The backend server receives a list of doubles containing the chemical shifts, a list of strings containing the multiplicity types and a list of doubles containing the deviations. Spring Boot automatically performs input validation to the form parameters. Therefore, if any variable is not valid, it returns 400 "Bad Request" response code with an appropriate message. I created a custom error page that shows a JSON file containing the response code, the response error, the error message and the time stamp.

If the form parameters are correct, the server goes through the chemical shifts, the deviations and the multiplicity types in parallel, creates a range [shift - deviation, shift + deviation] and searches the database for spectrum that contain shifts in this range and with the requested multiplicity type. Then, it reads the image of the molecular structure of the compounds with the found spectrum and returns them to the frontend. It also creates a query string which contains the form parameters. This query string will be used as part of the request sent to the UI tool in order to show the preselected areas (see Fig. 4.24).

The frontend receives the compound images. It loops through the images and places them into a grid having three compounds in each row. It also goes through the request parameters (chemical shifts, deviations and multiplicity types) and dynamically creates the list of the chemical shifts on the front end. It also contains the UI tool as a different component. The UI tool resides on a different server so it is imported using an iframe window. When the page loads, the frontend makes the request to the UI tool server which will return the UI tool view to the frontend.

The UI tool is coupled with the NMR spectrum search engine. It is implemented in Python using the Bokeh library. A request to the UI tool has the same format as a request to the NMR Spectrum search engine. It contains all the required information about the chemical shifts passed as parameters. The Bokeh Server creates a new, empty plot with these areas selected. Each area is a rectangle that covers the whole height of the plot and contains a label stating what is the multiplicity type of the requested chemical shift. When a selection or deletion action is triggered, the UI tool adds the new area in the parameters list and uses the JavaScript property *window.top* to fire a new search request to the search engine. As the UI tool is in a different domain, the two different applications communicate by securely sending messages between windows using the JavaScript method *postMessage()*. When the user clicks on a compound then the search engine posts a message to the frame with the compound id. The UI tool has a listener that checks for inbound messages. When it receives a request, it sends the compound id to the backend server using widgets. Then, the server parses the NMR data using the Nmrglue package and updates the plot.

6. Ethics

The evaluation stage of the project obtained ethical approval in order to test the application with users and gather their feedback. All data being gathered during the evaluation phase were completely anonymous. Participants were informed and agreed to participate before taking the study, while being able to stop the evaluation at any time with no explanation.

The ethical approval letter can be found in the Appendix Section A.

6.1 Collaborations

This project would not be possible without the collaboration of the following people:

- Prof. Simon Dobson who supervised the project and gave me advice on the direction of the project.
- Dr. Tomas Lebl who shared his knowledge about the chemistry aspects of the project. Also, Dr. Lebl periodically tested the project and gave me constant feedback.
- Simone Ivan Conte who proposed the initial idea of the project. Moreover, Simone helped with the computer science aspects of the project giving me regular feedback.
- The NOMAD team who gave me access to their code repository. The existing code of the spectrum viewer acted as a starting point to generate ideas.

7. Evaluation

The evaluation process aimed to evaluate the NMR analysis tool implemented for the purposes of this project. The purpose of the evaluation was to assess the usability of the tool; how well users could learn and use the product to achieve their goals and how satisfied users were with this process.

7.1 Participants

The evaluation process was conducted with people from the School of Chemistry. The only requirement of the participants was that they had to be familiar with the NMR analysis process and preferably they had used an NMR analysis tool before.

The study was conducted with 12 participants. Only two of these participants decided that they wanted to run the process only once and the rest completed the whole process twice. The majority of the participants where PhD students and academics. From the study we can verify that all participants were familiar with the Mestrelab Mnova software suite (Section 2.4.1), most of them were familiar with the TopSpin processing software and few had used another application (see Fig. 7.1).

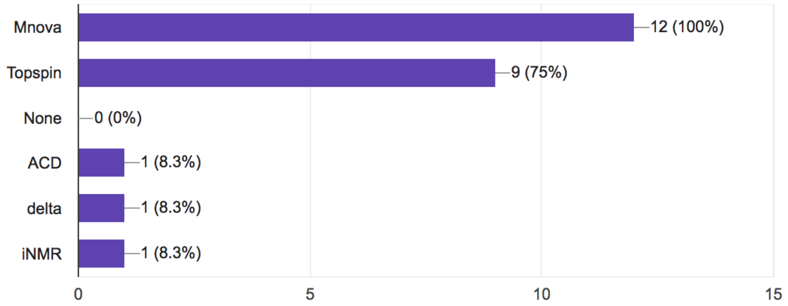


Figure 7.1: NMR Analysis tools used by the participants

7.2 Process

The users were not explained how the tool works. The participants were requested to perform certain NMR analysis operations to given NMR spectra and describe how easy or hard the actions were. More specifically, participants were requested to perform integral calculation and multiplet analysis to a proton spectrum and change the reference of the spectrum and peak picking to a carbon spectrum. They were asked to perform each action twice, firstly at a specific compound, which was the same compound for all participants, and then at a random one from a given list. This assists to comparing the feedback after participants learn how the system works. After that, participants were requested to complete the SUS questionnaire discussed above (see Section 2.5). The full questionnaire can be found in the Appendix Section B.

7.3 Analysis

The results of the evaluation process after the first execution of each action (see Fig. 7.2) found that most of the participants were confident in performing all the requested actions. Specifically, participants were more confident in using the multiplet analysis component. Even though it is the most complicated function, users found the design attractive. Participants found it obvious that they were able to manually change the multiplets from the panel under the table, compared to the integration component where users must double-click on the table entries. On the other hand, users found that the chemical referencing component was the hardest to use. This is because users spent some time trying to figure out how the tool works. Also, users found it hard to locate the old chemical shift at the graph.

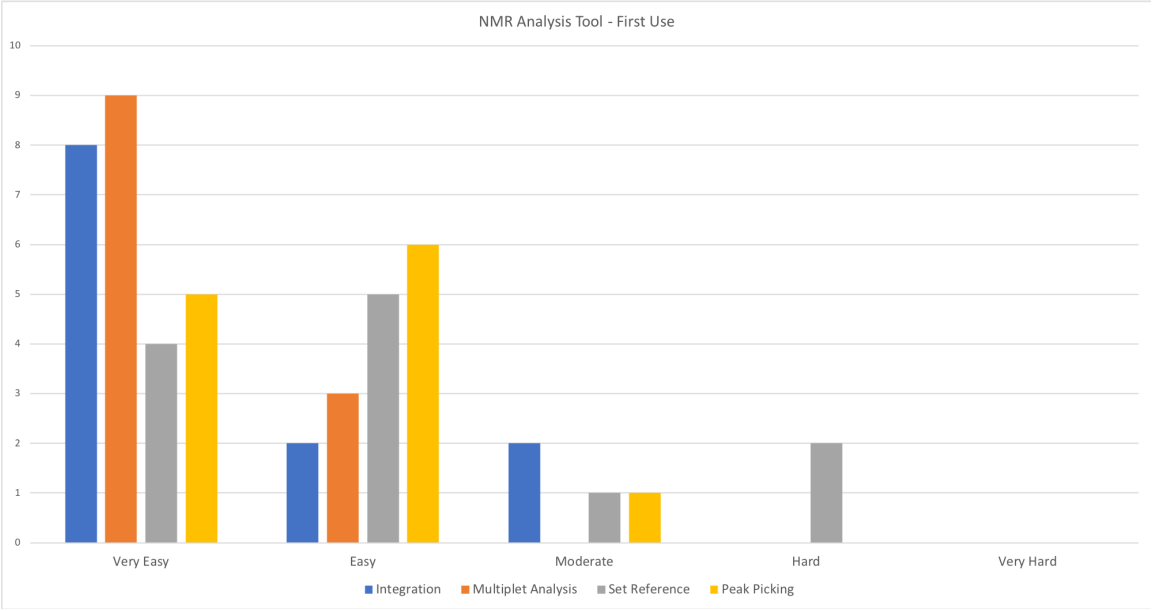


Figure 7.2: NMR Analysis Tool - First Evaluation

After the second execution of each action, participants clearly found the process easier (see Fig. 7.3). This is because users already knew how the tool operates and what each component does. Therefore, a quick tutorial that describes the tool would help first-time users.

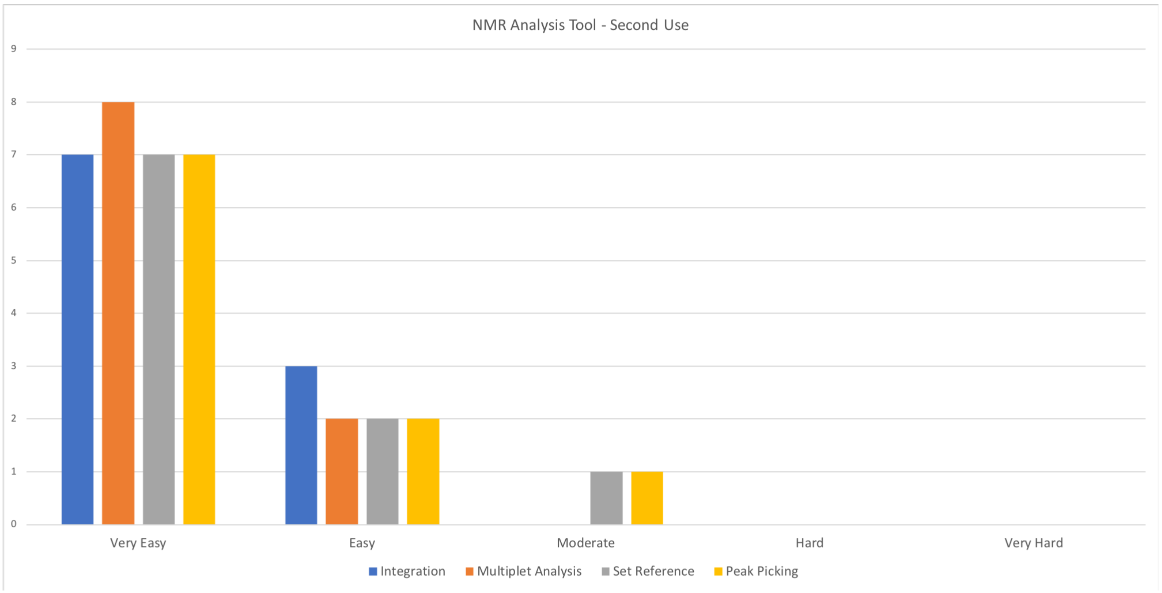


Figure 7.3: NMR Analysis Tool - Second Evaluation

After the use of the tool, the participants filled the SUS questionnaire. The data are presented in Figure 7.4. The data was analysed using the SUS interpretation process (see Section 2.5). The mean score of the feedback from all participants is *87.7*. This, based on the general guidelines of the SUS questionnaire, means that the NMR analysis tool is “Excellent” in terms of usability. Users found that it was easy to use the system and felt confident while using the tool from the first time.

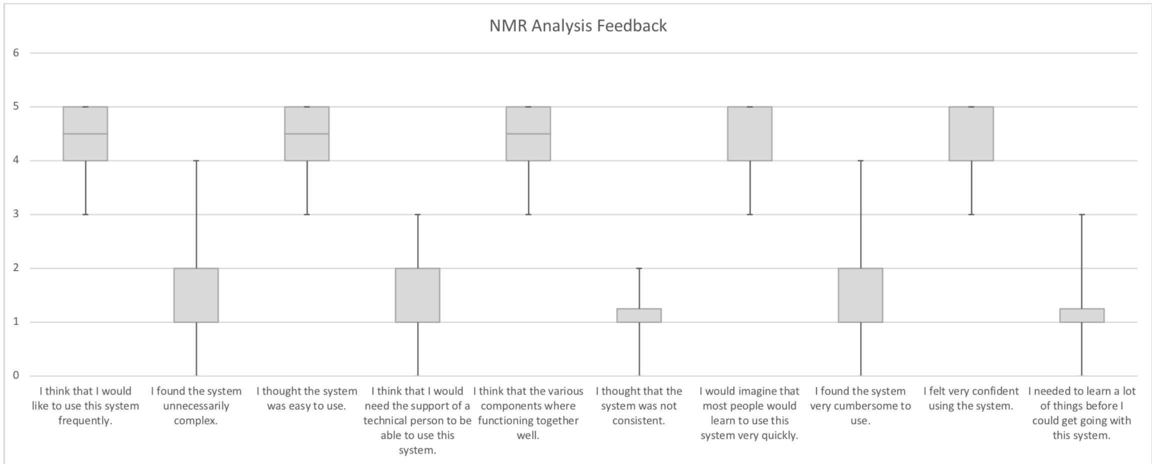


Figure 7.4: NMR Analysis Tool Feedback

7.4 Feedback from Users

Participants gave positive feedback on the NMR analysis tool. They were able to perform all the operations with minimal or no guidance and they felt very confident using the system from the first time. It is notable that no user found that they could switch between the operations using right-click gesture. However, after mentioning it, most users found it extremely useful. I assume that they found it helpful because of the unfamiliarity with the tool and that the right-click context menu uses string format to represent the tools.

Even though the feedback was very positive, users suggested some changes that would improve the usability and functionality of the system. A feature requested by some users is to change the mouse cursor to match the activated function. Even though the active functions are highlighted in the toolbar, users often paid attention only to the plot. Therefore, this will make it much easier to identify the active tool. Another recommendation regarding tool activation was the ability to switch between tools using keyboard shortcuts. As the frequency of use increases, so do the user’s desires to reduce the number of interactions and to increase the pace of interaction. Enabling keyboard shortcuts will be very helpful to an expert user. When users have a wrong function activated, or when they use a function incorrect they need a way to reverse their actions. That’s why some participants requested the implementation of undo functionality to provide easy reversal of actions. This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options [32].

Users also proposed some changes that improve the usability from chemistry perspective. Most participants, when requested to re-reference the plot, did not remember what was the correct position, even for the most common solvents. A suggestion, which is implemented by Mnova, is to provide a table of the most frequent solvents with the correct reference value. This eliminates the issue described above in a great extent. Users also suggested to implement a process that tracks the mouse pointer and automatically detects peaks for referencing. As the reference is always on a peak, some users found it cumbersome to manually find the peak by themselves. They expected that the referencing tool should highlight and auto-select the closest peak.

Finally, users spent significantly more time on the first NMR analysis procedure. The time difference is because users did not know how to use each function. Thus, many participants suggested the creation of a short, simple tutorial that briefly explains what each tool does and how it operates. This will also give more confidence to the users when they try the tool for the first time.

7.5 Conclusion of Evaluation

The evaluation of the NMR analysis tool was a worthwhile process. It provided valuable feedback for the strengths and weaknesses of the tool regarding the usability and the chemistry aspects. Users described how they felt when they firstly used the application and what they found hard to do, which is important for future versions of the tool. In general, users where able to successfully use the tool to complete the NMR analysis process from the first time.

8. Future Work

The NMR Analysis Tool and the NMR Spectrum Search Engine aim to enrich NMR data via Spectrum analysis. In the future, I plan to pursue additional work in collaboration with Dr. Tomas Lebl and the NOMAD team. This chapter intends to highlight the potential work that could be planned for the near future.

One of the aims of the project is to implement a web-application that utilises the open sourced NMR analysis tool. Users will be able to upload there NMR data and perform NMR analysis on the browser. The next iteration of the NMR analysis tool will focus on improving the design according to the feedback received from the conducted user study. The current version is user-friendly (see Chapter 7) but requires a little familiarity with the tool. It is important that the design must assist users to perform NMR analysis quickly and provide guidance when they do not know what to do.

Future work will consist into integrating the NMR analysis tool and the NMR Spectrum Search Engine into the NOMAD system. The users will be able to perform NMR analysis inside NOMAD and the experiment information will be associated with the data generated from the analysis. The search functionality will extend the current search functionality available and help users locate experiments and chemical compounds easier. After integration with NOMAD, the database of NMR analysis data will increase significantly. A study on how the search engine scales with the number of classified spectra must be conducted to analyse the performance of the engine. There are several strategies that can be introduced in order to improve the spectra search engine performance, such as adding secondary indexes in the database, database denormalisation and horizontal scaling.

Currently, the NMR analysis tool provides basic functionality for NMR analysis. There are several other, more advanced features that can be introduced, such as chemical assignments. The molecular file of the compound can be parsed and allow users to interact with the compound structure. Then, users would be able to assign the chemical elements of the compound structure to the chemical shifts of the NMR spectrum. This is important in order to verify that the NMR spectrum actually is a spectrum of the corresponding molecule. Finally, the increase of data in the database will allow the creation of automatic procedures, such as spectra prediction using the molecular structure of the compound and compound verification.

9. Conclusions

NOMAD is a successful system extensively used by the School of Chemistry. It stores a vast amount of data since 2012, allowing users to perform various actions such as searching, editing, grouping and downloading the data. It is currently under development and the NOMAD team maintains the current system and introduces new features. The main purpose is to make it as efficient as possible for researchers to store, locate and manipulate their data within the system.

This project explored data enrichment and data search mechanisms for NMR spectra via spectrum analysis. It used cutting edge technology and software engineering practices to implement a well designed application based on HCI principles. The project considers what is the audience and how they would be more familiar with the implementation by adopting design decisions from popular software.

Overall, the project met the requirements that were set from the start of the implementation. The NMR analysis tool provides a user-friendly environment with many features that allow complete NMR spectra analysis. The NMR Spectra Search Engine provides a simple environment that fits the purpose of the project. The UI Tool provides an alternative, faster way to set the parameters of the search. Even though the NMR Analysis Tool and the NMR Spectra Search Engine are going to be used for the same purpose, they are completely separated, allowing them to be used independently. The frontend and the backend of the search engine are also decoupled from one another, allowing developers to change the backend server or the frontend, by complying to the API schema.

Finally, a user-study was carried out to evaluate the usability of the NMR analysis tool. The results show that the participants were able to use the tool successfully without any knowledge. The study also gave valuable feedback for future versions to improve the usability and the chemistry aspects of the tool.

Bibliography

[1] N. R. Council *et al.*, *A question of balance: Private rights and the public interest in scientific and technical databases.* National Academies Press, 2000.

[2] K. G. Coffman and A. M. Odlyzko, ‘Internet growth: Is there a “moore’s law” for data traffic?’, in *Handbook of massive data sets*, Springer, 2002, pp. 47–93.

[3] D. Hunt, ‘Big data challenges: Volume, variety, velocity & veracity’, *NC State University Research*, pp. 2–8, 2014.

[4] D. Jeannerat, ‘Human-and computer-accessible 2d correlation data for a more reliable structure determination of organic compounds. future roles of researchers, software developers, spectrometer managers, journal editors, reviewers, publisher and database managers toward artificial-intelligence analysis of nmr spectra’, *Magnetic Resonance in Chemistry*, vol. 55, no. 1, pp. 7–14, 2017.

[5] S. I. Conte, F. Fina, M. Psalios, S. M. Reyal, T. Lebl and A. Clements, ‘Integration of an active research data system with a data repository to streamline the research data lifecycle: Pure-nomad case study’, 2017.

[6] J. Rivett, *Predicting Molecular Structures using SMILES strings and Bayesian Networks.* 2017. [Online]. Available: `http://sic2.me/resources/jake_sh_2017.pdf`.

[7] R. Hoffman, *What is nmr?* [Online]. Available: `http://chem.ch.huji.ac.il/nmr/whatisnmr/whatisnmr.html`.

[8] R. M. Silverstein, F. X. Webster, D. J. Kiemle and D. L. Bryce, *Spectrometric identification of organic compounds.* John wiley & sons, 2014.

[9] M.-A. Delsuc, L. Chiron and M.-A. Coutouly, *Introduction - spike documentation.* [Online]. Available: `https://spikedoc.bitbucket.io/introduction.html`.

[10] D. Tramesel, V. Catherinot and M.-A. Delsuc, ‘Modeling of nmr processing, toward efficient unattended processing of nmr experiments’, *Journal of Magnetic Resonance*, vol. 188, no. 1, pp. 56–67, 2007.

[11] M. A. van Agthoven, L. Chiron, M.-A. Coutouly, M.-A. Delsuc and C. Rolando, ‘Two-dimensional ecd ft-icr mass spectrometry of peptides and glycopeptides’, *Analytical chemistry*, vol. 84, no. 13, pp. 5589–5595, 2012.

[12] J. J. Helmus and C. P. Jaroniec, ‘Nmrglue: An open source python package for the analysis of multidimensional nmr data’, *Journal of Biomolecular NMR*, vol. 55, no. 4, pp. 355–367, 2013. DOI: `10.1007/s10858-013-9718-x`.

[13] T. E. Oliphant, *A guide to NumPy*. Trelgol Publishing USA, 2006, vol. 1.

[14] E. Jones, T. Oliphant, P. Peterson *et al.*, *SciPy: Open source scientific tools for Python*, [Online; accessed ¡today¿], 2001–. [Online]. Available: `http://www.scipy.org/`.

[15] A. Mohamed, C. H. Nguyen and H. Mamitsuka, ‘Nmrpro: An integrated web component for interactive processing and visualization of nmr spectra’, *Bioinformatics*, vol. 32, no. 13, pp. 2067–2068, 2016.

[16] J. D. Hunter, ‘Matplotlib: A 2d graphics environment’, *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: `10.1109/MCSE.2007.55`.

[17] P. T. Inc. (2015). Collaborative data science, [Online]. Available: `https://plot.ly`.

[18] Bokeh Development Team, *Bokeh: Python library for interactive visualization*, 2014. [Online]. Available: `https://bokeh.pydata.org`.

[19] M. Bostock, V. Ogievetsky and J. Heer, ‘D3 data-driven documents’, *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011, ISSN: 1077-2626. DOI: `10.1109/TVCG.2011.185`. [Online]. Available: `http://dx.doi.org/10.1109/TVCG.2011.185`.

[20] S. Ray, *Interactive data visualization using bokeh (in python)*, 2015. [Online]. Available: `https://www.analyticsvidhya.com/blog/2015/08/interactive-data-visualization-library-python-bokeh/`.

[21] M. R. Willcott, *Mestre nova*, 2009.

[22] L. Patiny, M. Zasso, D. Kostro, A. Bernal, A. M. Castillo, A. Bolaños, M. A. Asencio, N. Pellet, M. Todd, N. Schloerer *et al.*, ‘The c6h6 nmr repository: An integral solution to control the flow of your data from the magnet to the public’, *Magnetic Resonance in Chemistry*, 2017.

[23] S. Kuhn and N. E. Schlörer, ‘Facilitating quality control for spectra assignments of small organic molecules: Nmrshiftdb2–a free in-house nmr database with integrated lims for academic service laboratories’, *Magnetic Resonance in Chemistry*, vol. 53, no. 8, pp. 582–589, 2015.

[24] J. Kirakowski and M. Corbett, ‘Sumi: The software usability measurement inventory’, *British journal of educational technology*, vol. 24, no. 3, pp. 210–212, 1993.

[25] A. Garcia, *Ux research — standardized usability questionnaire*, 2013. [Online]. Available: `https://arl.human.cornell.edu/linked%20docs/Choosing%20the%20Right%20Usability%20Questionnaire.pdf`.

[26] J. Brooke *et al.*, 'Sus-a quick and dirty usability scale', *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.

[27] T. S. Tullis and J. N. Stetson, 'A comparison of questionnaires for assessing website usability', in *Usability professional association conference*, 2004, pp. 1–12.

[28] J. Brooke, 'Sus: A retrospective', *Journal of usability studies*, vol. 8, no. 2, pp. 29–40, 2013.

[29] A. Bangor, P. T. Kortum and J. T. Miller, 'An empirical evaluation of the system usability scale', *Intl. Journal of Human–Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008.

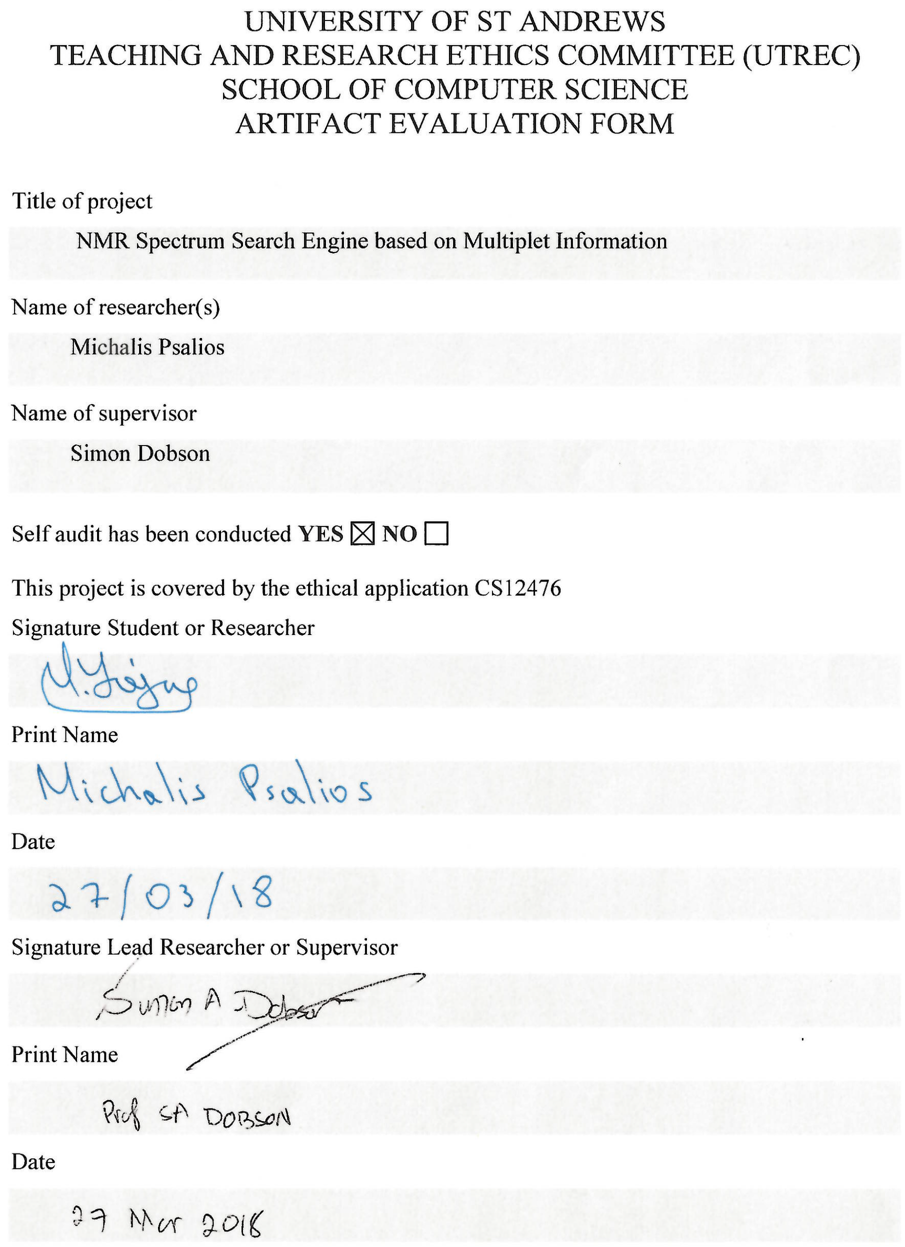
[30] F. P. Miller, A. F. Vandome and J. McBrewster, 'Apache maven', 2010.

[31] U. o. V. Norbert Haider, *Spectral similarity search with ranking.* [Online]. Available: `http://nmrpredict.orc.univie.ac.at/similar/eval.php`.

[32] B. Shneiderman, 'Shneiderman's eight golden rules of interface design', *Retrieved July*, vol. 25, p. 2009, 2005.

Appendices

A. Ethical Approval Letter



UNIVERSITY OF ST ANDREWS
TEACHING AND RESEARCH ETHICS COMMITTEE (UTREC)
SCHOOL OF COMPUTER SCIENCE
ARTIFACT EVALUATION FORM

Title of project

NMR Spectrum Search Engine based on Multiplet Information

Name of researcher(s)

Michalis Psalios

Name of supervisor

Simon Dobson

Self audit has been conducted **YES** ☒ **NO** □

This project is covered by the ethical application CS12476

Signature Student or Researcher

Print Name

Michalis Psalios

Date

27/03/18

Signature Lead Researcher or Supervisor

Simon A Dobson

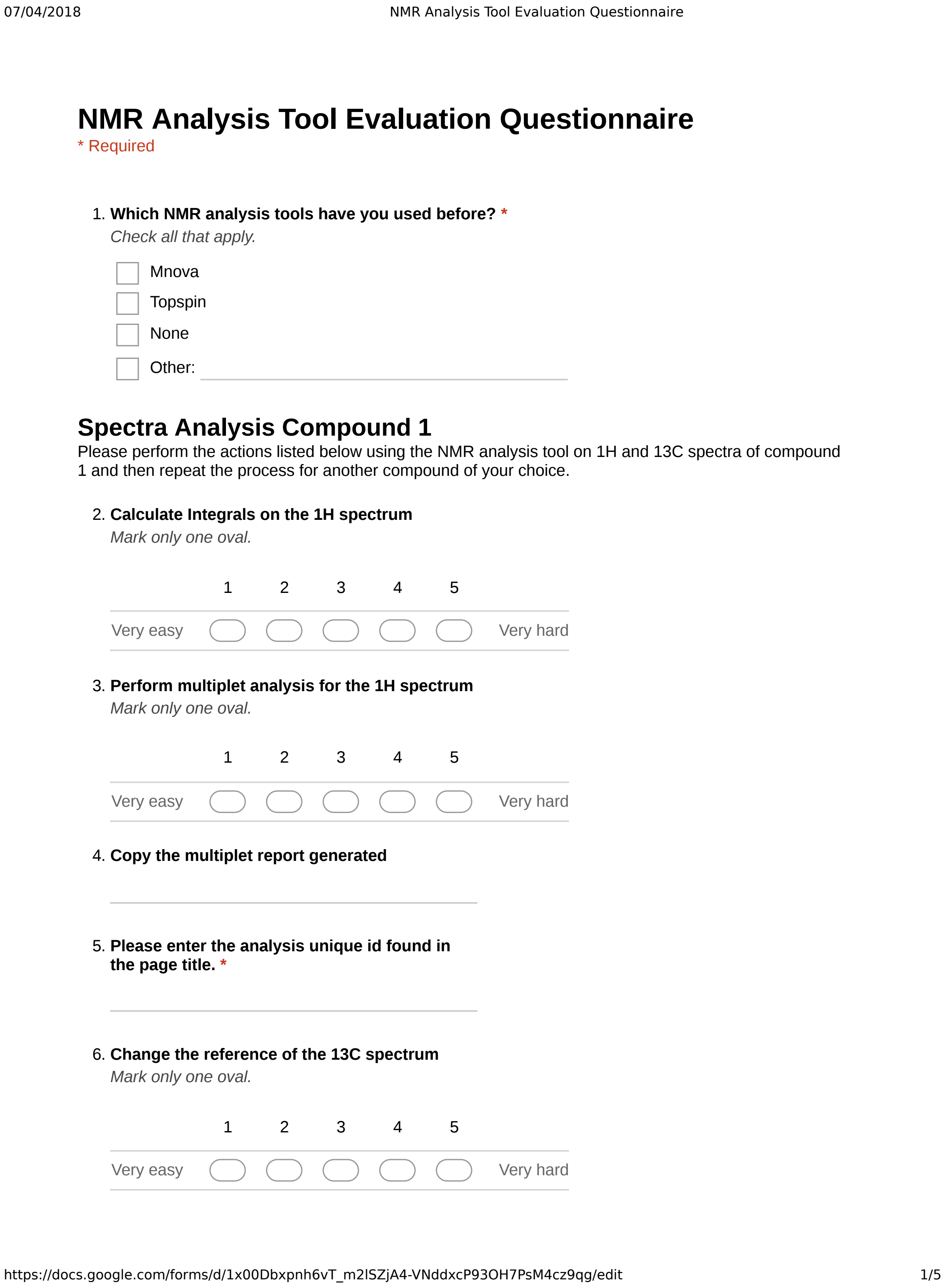
Print Name

Prof SA DOBSON

Date

27 Mar 2018

B. Evaluation Questionnaire



NMR Analysis Tool Evaluation Questionnaire

* Required

1. **Which NMR analysis tools have you used before? ***

   Check all that apply.

   □ Mnova

   □ Topspin

   □ None

   □ Other: ____________________

Spectra Analysis Compound 1

Please perform the actions listed below using the NMR analysis tool on 1H and 13C spectra of compound 1 and then repeat the process for another compound of your choice.

2. **Calculate Integrals on the 1H spectrum**

   Mark only one oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very hard

3. **Perform multiplet analysis for the 1H spectrum**

   Mark only one oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very hard

4. **Copy the multiplet report generated**

   ____

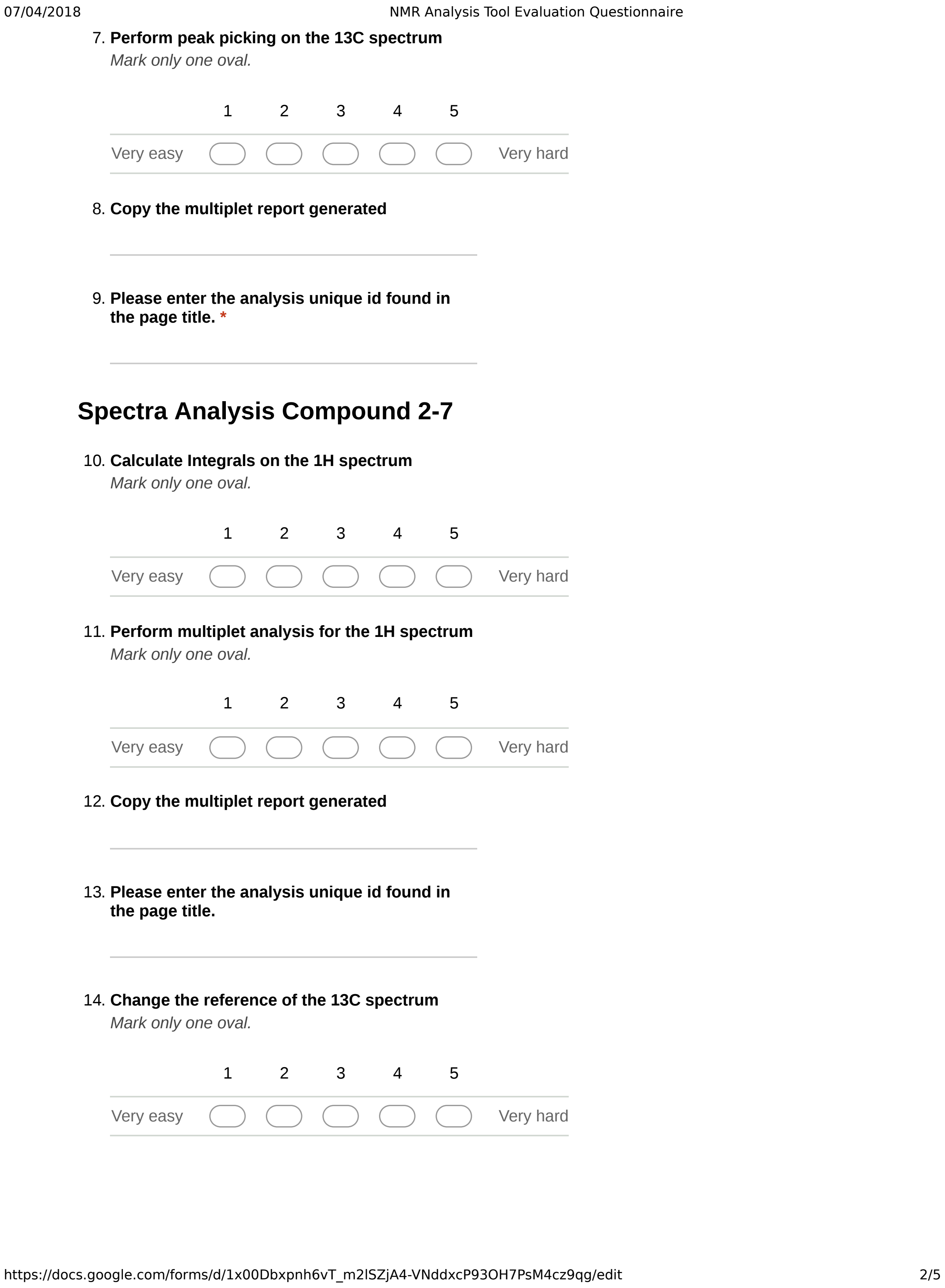
5. **Please enter the analysis unique id found in the page title. ***

   ____

6. **Change the reference of the 13C spectrum**

   Mark only one oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very hard



7. **Perform peak picking on the 13C spectrum**
Mark only one oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very hard

8. **Copy the multiplet report generated**

9. **Please enter the analysis unique id found in the page title. ***

Spectra Analysis Compound 2-7

10. **Calculate Integrals on the 1H spectrum**
Mark only one oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very hard

11. **Perform multiplet analysis for the 1H spectrum**
Mark only one oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very hard

12. **Copy the multiplet report generated**

13. **Please enter the analysis unique id found in the page title.**

14. **Change the reference of the 13C spectrum**
Mark only one oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very hard

15. **Perform peak picking on the 13C spectrum**
Mark only one oval.

	1	2	3	4	5	
Very easy	○	○	○	○	○	Very hard

16. **Copy the multiplet report generated**

____

17. **Please enter the analysis unique id found in the page title.**

____

Feedback

18. **I think that I would like to use this system frequently.**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

19. **I found the system unnecessarily complex.**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

20. **I thought the system was easy to use.**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

21. **I think that I would need the support of a technical person to be able to use this system.**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

22. **I think that the various components where functioning together well.**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

23. **I thought that the system was not consistent.**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

24. **I would imagine that most people would learn to use this system very quickly.**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

25. **I found the system very cumbersome to use.**
Mark only one oval.

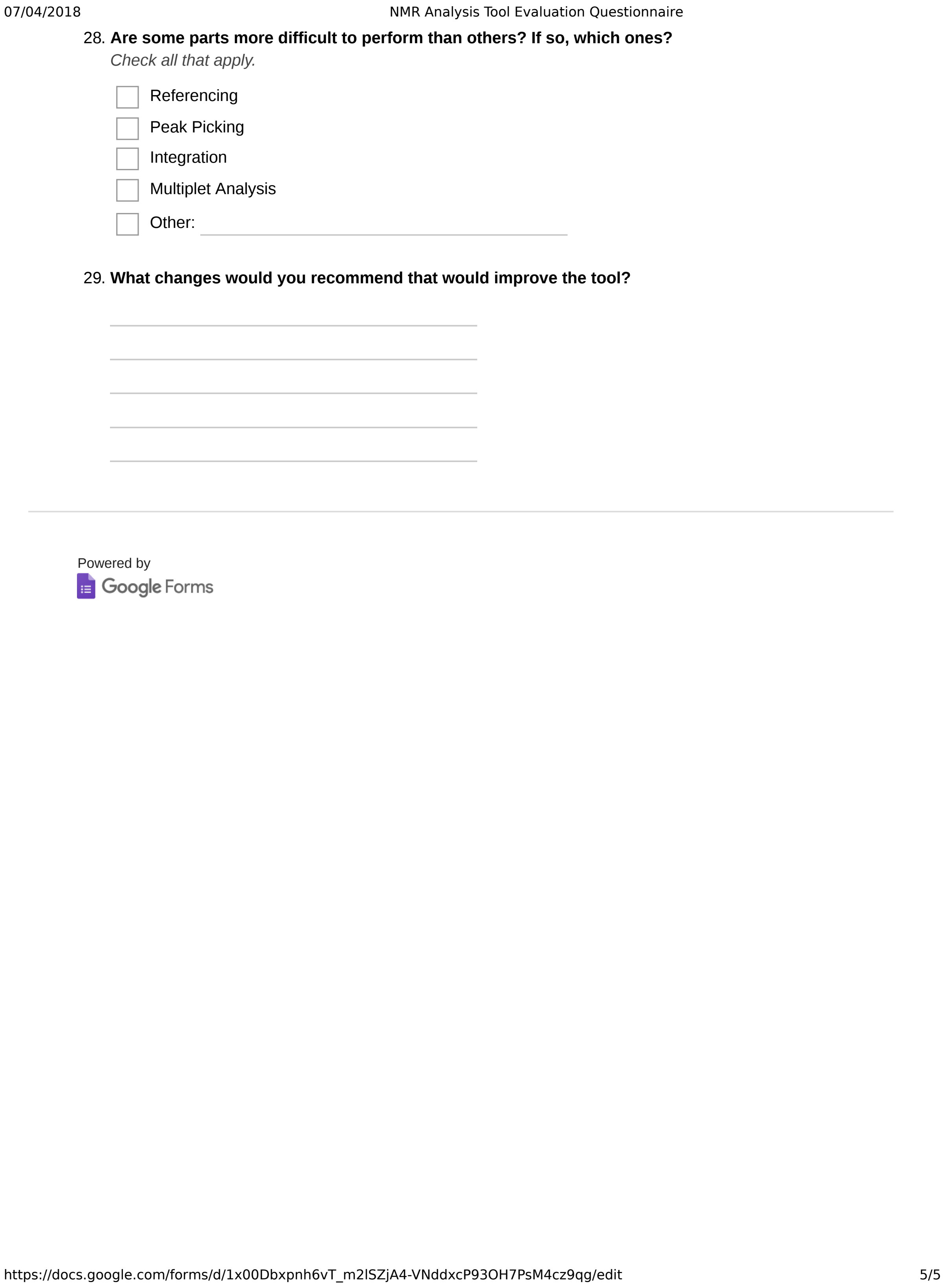
	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

26. **I felt very confident using the system.**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

27. **I needed to learn a lot of things before I could get going with this system.**
Mark only one oval.

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree



28. **Are some parts more difficult to perform than others? If so, which ones?**
Check all that apply.

- □ Referencing
- □ Peak Picking
- □ Integration
- □ Multiplet Analysis
- □ Other: ______________________

29. **What changes would you recommend that would improve the tool?**

____

____

____

____

____

Powered by
Google Forms

C. User Feedback Results

The results from the user feedback are stored in the submission folder, in a file called “Feedback.csv”.