

**FINAL REPORT**

**Self-learning Named-Entity Recognition Platform**

Name of the Team Members (in alphabetical order), *2002169*; Chen Xin Xin, *2001893*; Chong Wei Wen, Reuben, *2001135*; Jevan Ng Jia Qi, *2002841*; Leong Kah En, *2002738*; Whitney Tan Wenhui

Name of the Industry Supervisor: **Mr Rek Chong**; Name of the Company: **Element**

Name of the SIT Academic Supervisor(s): **Dr Muhamed Fauzi BIN ABBAS**



**ABSTRACT**

Element is one of the leading independent providers of destructive and non-destructive testing services for metals, and non-metals in ensuring safety, quality, compliance, and fit for purpose of these materials. Within the company, Excel has been adopted as their current form of software that acts as a form of a database for storing data. The company would like to increase the efficiency of performing data analytics on the Excel files that have been created by their scientists/engineers through a newly created web application platform that utilizes machine learning to build a machine learning model that aids in cleaning the data. This project targets to build and train a Named Entity Recognition Model using artificial intelligence to perform data cleansing which will be hosted on a web application for users to extract entities from the Excel files into a consolidated list of distinct values. To achieve the NER model’s accuracy, it has gone through thorough training to ensure the produced result is ideal.



# 

# ACKNOWLEDGMENTS

First of all, this project would not have been possible without the opportunity given by our client, Mr Rek Chong, and his company, Elements Material Technology to work on this project and for putting faith in us to develop a product while meeting their expectation. Thus, we would like to express our special thanks of gratitude to our client and the company for allowing our team to have the chance in working closely with them whilst gaining valuable experiences.

We would also like to extend our gratitude to our client, Mr Rek Chong, for taking time out of his busy schedule for his valuable guidance and support as an industrial supervisor for this project. From the start of the project, he would meet with us every fortnight to review the progress of work that we have made and provide us with constructive feedback to ensure that we are going in the right direction.

Aside from the project review, he has shared with us his experiences and knowledge in this field of industry to educate us on information related to his company and their background. Even on days when we are not scheduled for a meeting, he would stay in contact with us via email to address our clarifications and approach us for more information on our project solution.

We would also like to express our warm thanks to our academic supervisor, Mr Muhamed Fauzi Bin Abbas, for overseeing this project. Throughout the course of this project, he has taken the time to schedule a meeting with us to discuss how our team is coping and to raise any concerns. We deeply appreciate the effort put in to check on us, giving us advice and ways, we can improve further in achieving the deliverables required in this project module.

Once again, we would like to express our sincere appreciation to Mr Muhamed Fauzi Bin Abbas, Mr Rek Chong, and everyone who has played a part in contributing to the completion of this project.

# CONTENTS PAGE(s)

[ACKNOWLEDGMENTS](#_heading=h.vmbfs9a6syt6) **2**

[CONTENTS PAGE(s)](#_heading=h.30j0zll) **3**

[GLOSSARY](#_heading=h.g7ua8lx4ds70) **6**

[INTRODUCTION](#_heading=h.3znysh7) **6**

[Motivation for the project](#_heading=h.2et92p0) 6

[Project Domain and Background](#_heading=h.8s2eju4occcw) 7

[Project Objective](#_heading=h.3dy6vkm) 7

[PROJECT TIMELINE](#_heading=h.1t3h5sf) **7**

[Project Management Plan](#_heading=h.4d34og8) 7

[Project Management Plan Review](#_heading=h.ozadz785zzoa) 9

[Software Development Methodologies Applied](#_heading=h.17dp8vu) 9

[Deadlines and Deliverables](#_heading=h.3rdcrjn) 9

[TEAM ORGANISATION](#_heading=h.z85aarj3miip) **10**

[Chosen Team Approach](#_heading=h.lnxbz9) 10

[Communication Strategies](#_heading=h.35nkun2) 10

[Team Members Roles and Responsibilities](#_heading=h.1ksv4uv) 11

[REVIEW OF RELEVANT LITERATURE](#_heading=h.44sinio) **12**

[Name Entity Recognition](#_heading=h.2jxsxqh) 12

[Regular Expression (RegEx)](#_heading=h.z337ya) 12

[Review Summary](#_heading=h.3j2qqm3) 13

[REQUIREMENTS ELICITATION](#_heading=h.1y810tw) **13**

[Requirements Elicitation Techniques](#_heading=h.4i7ojhp) 13

[Stakeholder](#_heading=h.2xcytpi) 13

[Functional Requirements](#_heading=h.1ci93xb) 13

[Non-Functional Requirements](#_heading=h.3whwml4) 14

[*Performance*](#_heading=h.2bn6wsx) *14*

[*Maintainability*](#_heading=h.qsh70q) *14*

[*Usability*](#_heading=h.rvfm878w81ew) *14*

[REQUIREMENTS PRIORITISATION](#_heading=h.1pxezwc) **14**

[USE CASE DIAGRAM](#_heading=h.2udpykzbdc9z) **16**

[Use Case Diagram Description](#_heading=h.2p2csry) 16

[USE CASE DESCRIPTIONS](#_heading=h.147n2zr) **16**

[USER STORIES](#_heading=h.3o7alnk) **17**

[File Management](#_heading=h.23ckvvd) 17

[Data Processing](#_heading=h.qtugym4anig4) 18

[USER STORIES PRIORITISATION](#_heading=h.32hioqz) **18**

[ARCHITECTURE DIAGRAM](#_heading=h.1hmsyys) **18**

[Architecture Diagram Description](#_heading=h.41mghml) 19

[Flask-API Web Microservice](#_heading=h.lwu3inbjuaj7) 19

[FEATURES IMPLEMENTED](#_heading=h.8ly1o3b2mzam) **20**

[Upload](#_heading=h.1v1yuxt) 20

[Extract Entities](#_heading=h.4f1mdlm) 20

[Download](#_heading=h.7eaaxo2u0plj) 21

[TESTING](#_heading=h.2u6wntf) **21**

[Black Box Testing](#_heading=h.19c6y18) 21

[UI/UX Testing](#_heading=h.28h4qwu) 24

[LIMITATIONS](#_heading=h.9em6q3cr56w4) **24**

[Accuracy of the Learning Model Dependent on Training Data](#_heading=h.1mrcu09) 24

[CHALLENGES FACED](#_heading=h.4k668n3) **25**

[Training Process is Time Consuming](#_heading=h.2zbgiuw) 25

[Inaccurate Word Tokenizers](#_heading=h.bjrj8zeo1ab8) 25

[FUTURE WORKS](#_heading=h.3ygebqi) **26**

[Support Multiple Format Conversion](#_heading=h.3cqmetx) 26

[File Password Protection](#_heading=h.hbrq3krqen6x) 26

[TECHNOLOGY AND TOOLS](#_heading=h.1rvwp1q) **26**

[Project Management](#_heading=h.4bvk7pj) 26

[Communication Tools](#_heading=h.2r0uhxc) 26

[Development Tools](#_heading=h.1664s55) 26

[Data Sources](#_heading=h.25b2l0r) 26

[APPLICATION OF CONCEPTS TAUGHT](#_heading=h.kgcv8k) **27**

[ICT 1002: Programming Fundamentals](#_heading=h.34g0dwd) 27

[ICT 1004: Web Systems and Technologies](#_heading=h.y9bdegr5scub) 27

[ICT 1007: Operating Systems](#_heading=h.p3wlslqi7thb) 27

[ICT 1008: Data Structures and Algorithms](#_heading=h.hpmftwgymcsp) 27

[ICT 1009: Object Oriented Programming](#_heading=h.qfuoraqi0nyz) 28

[ICT 1010: Computer Networks](#_heading=h.glh73tup3899) 28

[ICT 2101: Introduction to Software Engineering](#_heading=h.9z38zbe5dfn3) 28

[ICT 2103: Information Management](#_heading=h.auinf78xugn8) 28

[ICT 2106: Software Design](#_heading=h.opjki9jq40iz) 28

[ICT 2108: Software Modelling and Analysis](#_heading=h.hyusdq5s8tcb) 28

[INTRODUCTION AND APPLICATION OF NEW CONCEPTS](#_heading=h.1jlao46) **29**

[Web-related](#_heading=h.43ky6rz) 29

[*ReactJS*](#_heading=h.2iq8gzs) *29*

[Flask-API](#_heading=h.d7mqyxnqlfqq) 29

[Machine Learning related](#_heading=h.xvir7l) 29

[FINAL OUTCOMES AND RESULTS OF THE PROJECT](#_heading=h.1x0gk37) **30**

[LESSONS LEARNT](#_heading=h.4h042r0) **30**

[REFERENCES](#_heading=h.loiecwi0kjp8) **31**

[APPENDICES](#_heading=h.itvqxpafoag2) **32**

[APPENDIX A: GANTT CHARTS](#_heading=h.sz8a4adm31v8) 32

[APPENDIX B: REQUIREMENTS ELICITATION INTERVIEW MINUTES](#_heading=h.wt231yt243xc) 34

[APPENDIX C: WIREFRAME](#_heading=h.9qfb02sqd13x) 37

[APPENDIX D: USE CASE DESCRIPTIONS](#_heading=h.kujygaw962pw) 38

[APPENDIX E: ARCHITECTURE DIAGRAM](#_heading=h.opk9yx4x2280) 40

[APPENDIX F: BLACK BOX TESTING DOCUMENT](#_heading=h.unwxq591x8yn) 41

# GLOSSARY

This section contains the table for the definitions, acronyms and abbreviations used within this document.

**Table 1. Glossary Table**

| **Term** | **Description** |
| --- | --- |
| NER | Short for Named Entity Recognition helps to identify and classify named entities mentioned in unstructured text into pre-defined categories |
| ML | Short for Machine Learning |
| SpaCy | An open-source software library for advanced natural language processing. |
| NLTK | Short for Natural Language Toolkit used for writing in Python programming language. |
| RegEx | Short for Regular Expression |
| MERN | Short for MongoDB, Express, React, Node |
| API | Short for Application Programming Interface, to allow access to the features or data  of an operating system, application, or other services |
| UI | Short for User Interface |
| UX | Short for User Experience |
| HTTP | Short for Hypertext Transfer Protocol |
| CSV | Short for Comma-separated values |

# INTRODUCTION

This section comprises the motivation for the project, project domain and background, and project objectives.

## Motivation for the project

Data is often loosely entered into spreadsheets as an easy method to capture information in many parts of the business. These data are unstructured text containing spelling mistakes, short-form writings and comments. In most cases, these unstructured texts are key information such as product, service, people, etc which are used for identification or classification purposes. Therefore, it presents a need for the data to be properly reviewed and cleansed. It is often extremely laborious to manually identify and categorize these texts, thus the demand for unstructured text identification. Our team was requested by the client to develop a web tool and a machine learning model used for Named-Entity Recognition (NER) to extract key entities from input files which helps to speed up the data cleansing process.

## 

## Project Domain and Background

The project domain revolves around several sectors in Singapore including building products, defence, medical devices, power generation, telecommunications, and many other industry sectors. Element Materials Technology is a world-class provider of testing, inspection, and certification services to many of the world's most renowned corporations. [1] From early research and development through complex regulatory approvals and into production, Element’s global laboratory network of scientists, engineers, and technologists assists customers in achieving assurance over product quality, sustainable outcomes, and market access.[2]

The background of this project looks at the current method of logging information and how the concept of a database/web tool can assist the company in using machine learning to increase efficiency in their data analysis. The company currently operates 350 labs around the world. It is difficult to determine which lab is using which specifications or test methods because this information is not kept centrally. As a result, in their current situation, they are limited in their ability to perform data analysis after storing it in Excel spreadsheets. The client expressed their concern that their current organisation does not have the capability to develop a database/web tool because most of their scientists/engineers prefer logging information into excel sheets. Hence, they naturally utilize Excel as their main platform for entering data.

Furthermore, as hundreds of their employees use their own Excel spreadsheet, there is no chance for data analysis, and it will be difficult to build a standardized glossary of industry-specific legal terms that can be used within the organization. Therefore, our created platform will assist the employees in quickly cleaning up the data so that the data team can use it to conduct efficient analysis and linking of industry-specific legal terms.

## Project Objective

The primary objective of this project is to design and develop a web application and a machine learning model. These components encompass the following aspects respectively:

Web application:

* Excel upload
* Database for storing the text identified
* Facilitate the use of the ML model through microservices

NER model:

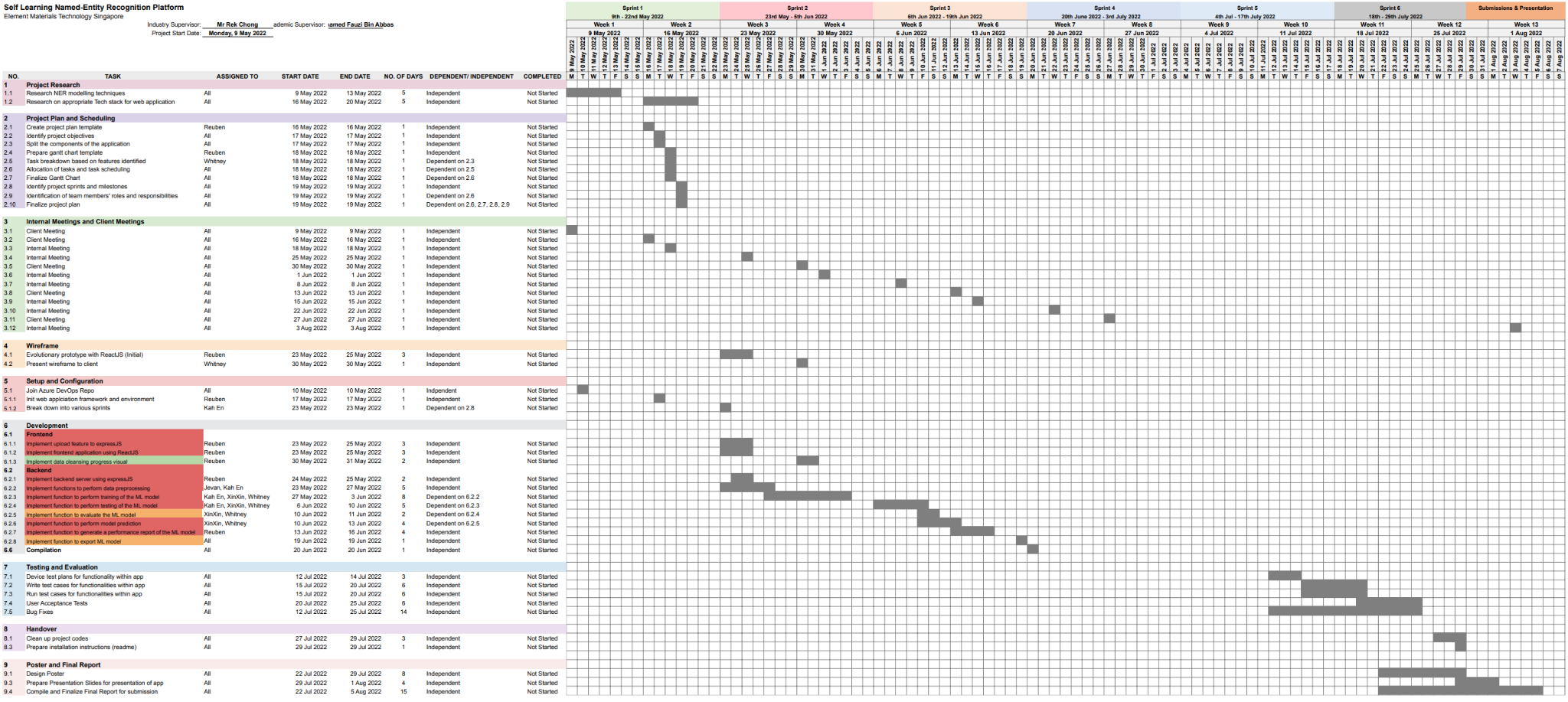
* Detect name entities from raw Excel inputs
* Classify name entities into predefined categories
* Output Excel in a format which client desires

# PROJECT TIMELINE

This section comprises the Gantt chart representing the timeline of events throughout this project, a review of the changes made to the timeline in the Gantt Chart, the software development methodology used and the deadlines and deliverables for each sprint.

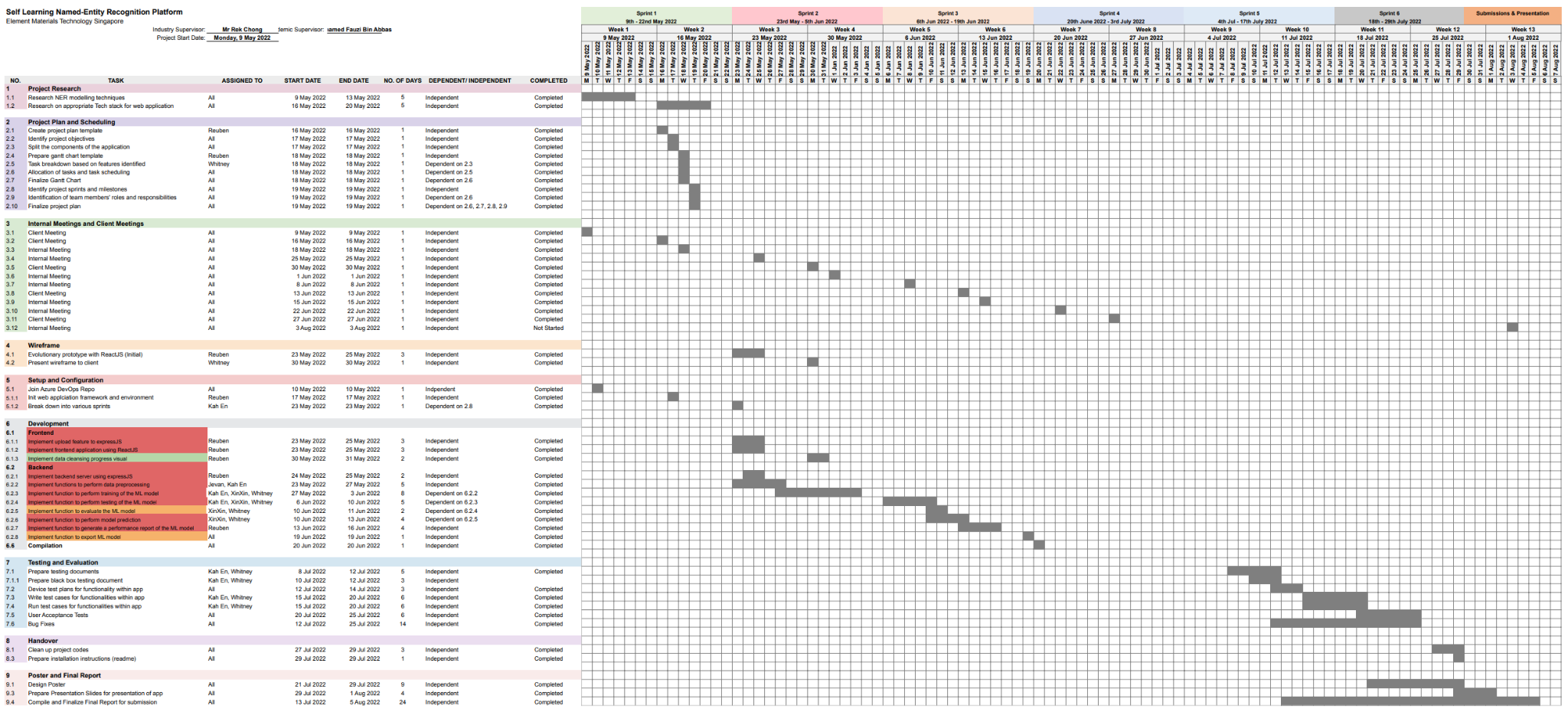
## Project Management Plan

The following consist of the Gantt Chart submitted at the beginning of the project and the latter is the latest Gantt Chart upon completion of this project. Alternatively, you may refer to [Appendix A: Gantt Charts](#_heading=h.sz8a4adm31v8) or the external link provided to view the resource.



**Figure 1: Gantt Chart (Start of Project)**

[ITP Team 4 Gantt Chart (Start of Project)](https://docs.google.com/spreadsheets/d/1TK4QCX5R8uvKz_-tRvo41WRg3qutBWkVGNFvw137dU0/edit?usp=sharing)



**Figure 2: Gantt Chart (End of Project)**

[ITP Team 4 Gantt Chart (End of Project)](https://docs.google.com/spreadsheets/d/1dqx2xNxsYWTOe5dypCwvWqiQzys7_9tZ_dOzo9SK1Pw/edit#gid=1769656556)

## Project Management Plan Review

Client meetings were held bi-weekly to keep the client up to date on the team's progress during each sprint. As our client is situated overseas, the meetings were held online through Microsoft Teams. At the end of each sprint review session, the client would provide the team with feedback regarding our progress and design decision. This has greatly benefitted the team as it helped to contribute to the action items for the following sprint. With this project management plan, we were able to complete the tasks for each sprint on time.

In the initial Gantt Chart, we did not take into account the test-driven development (TDD) approach and allocated it for testing at the end of the project. However, in the early phases of the project, we realised that testing should be done concurrently along with development, thus, we adopted TDD which is reflected in the Gantt Chart.

## Software Development Methodologies Applied

Our team adopted the agile software development methodology, this allowed for increment development based on the client’s requirements.

## Deadlines and Deliverables

The following table lists the start and end dates for each sprint and the deliverables for each sprint.

**Table 2. Deadlines and Deliverables**

| **Sprint** | **Period** | **Deliverables** |
| --- | --- | --- |
| 1 | 9th May - 22nd May | * Research on NER techniques * Research on appropriate web application framework * Research on tech stack application |
| 2 | 23rd May - 5th June | * Identification of individual tasks, roles, and responsibilities * Finalize features of the app * Web application prototype |
| 3 | 6th June - 19th June | * ML model training and application |
| 4 | 20th June - 3rd July | * Testing and Bug Fixing |
| 5 | 4th July -17th July | * Testing and Bug Fixing * Report and slides |
| 6 | 18th July - 29th July | * Final release * Report, Presentation slides |

# 

# TEAM ORGANISATION

This section comprises the approach selected for our team’s organization, communication strategies, and individual members’ roles and responsibilities.

## Chosen Team Approach

Our team appointed a leader to handle the majority of the communication and documentation work between the group and the client, this provides for a consistent medium for communication. As this project consists of 2 different development projects, the web application, and the machine learning model, we were split into two separate teams based on individual preferences and interests. This helped to improve the team’s motivation and has allowed us to work more effectively. Another benefit of this approach is the ability to work on the two development projects concurrently which helped to boost efficiency. Ultimately, this approach has helped us achieve our sprint objectives on time to satisfy the client’s needs.

## Communication Strategies

Our team split the term “communication” into, 1) day-to-day communication, 2) bi-weekly internal meetings and 3) bi-weekly client meetings.

Day-to-day communication is the platform to resolve any daily problems encountered that are related to the project, or any other issues that the member is facing related to the project. This communication was done via telegram group messages where every member participated actively to help resolve day-to-day issues. The team also utilised the message pinning feature of Telegram to temporarily keep track of general client requests so that we could craft our client emails appropriately.

Bi-weekly internal meetings were conducted on weeks when there are no client meetings scheduled. This allowed the team to have a separate meeting without the client to integrate the week’s worth of work as well as to have further discussions. The meeting also helped the team to resolve any other issues before the upcoming client meeting. This meeting was done over discord group video calls.

Bi-weekly client meetings were done to update the client on our sprint results and receive feedback. This was done over Microsoft Teams.

For documentation collaboration work, i.e. report, Gantt chart, slides, diagrams, etc. our team used google shared drive to collaborate on these tasks.

For technical code, the client has very kindly set up an Azure DevOps pipeline for us. It uses Git and is very similar to GitHub which all of us are familiar with.

## Team Members Roles and Responsibilities

The following tables depict our team members’ individual roles and responsibilities.

**Table 3. Team Members Roles and Responsibilities**

| **Team Member** | **Roles and Responsibilities** |
| --- | --- |
| Chen XinXin | As a team member, my main responsibilities is the exploration and development of the learning model that will be imported in the microservice application. This consists of compiling training data and the training of a custom learning model that suits the client’s use case, exportation and implementation of the model so as to produce an output Excel with the extracted entities detected in the format that our client demands. |
| Chong Wei Wen Reuben | As a team member, my main responsibilities revolve around the development of the web application. This consists of, architectural design, application data flow design, document design for MongoDB, dynamic frontend design using ReactJS, server implementation using ExpressJS, backend implementation using MongoDB and NodeJ, web microservice design and implementation using Flask-API to host the machine learning model, documentation such as application documentation, report, slides and poster. |
| Leong Kah En | As a team member, my main responsibilities are the data cleansing, exploration, and development of the learning model that will be imported into the microservice application. This includes compiling training data to feed into a custom learning model tailored to the client's use case, doing testing of the web application as well as documentation such as application documentation, reports, slides, and poster. |
| Whitney Tan Wenhui | As a team leader, my main responsibilities are communicating between the client and my team members to arrange meetings, as well as the data cleansing, exploration, and development of the learning model that will be imported into the microservice application embedded in the web application's backend. This includes compiling training data to feed into a custom learning model tailored to the client's use case, doing testing of the web application as well as documentation such as application documentation, reports, slides, and poster. |
| Jevan Ng Jia Qi | As a team member, my main responsibilities are the integration of the web application and learning model, configuration of processed data to be returned and uploaded to the database as well as the testing of overall performance (time taken for processing and upload) of the integrated application. |

# REVIEW OF RELEVANT LITERATURE

## Name Entity Recognition

**Table 4. Name Entity Recognition Libraries Comparison**

| **spaCy** | **NLTK** |
| --- | --- |
| Object-Oriented | Mainly a string processing library |
| Provides the most efficient NLP algorithm for a given task | Provides access to many algorithms |
| Easy to learn and use | Complicate to learn and use |
| Suitable for app developers | Suitable for researchers |

Being object-oriented, spaCy functions return objects instead of strings or arrays whereas NLTK functions will return a processed string. It is easier for the user to explore with the spaCy library as compared with NLTK since users do not need to check the documentation to understand the context of the string. While NLTK provides the access to many algorithms, spaCy acts as a service that focuses on completing a specific task and it already provides the most efficient algorithm. This results in spaCy returns result faster than NLTK.

In summary, spaCy is suitable for app developers and is prioritized in completing the task whereas NLTK is suitable for scholars and researchers to explore the algorithms.

## Regular Expression (RegEx)

A regular expression is a string of characters that specifies a text search pattern. A regex character pattern can be defined or later used to locate a specific pattern within a larger body of text. RegEx will be incredibly practical for Element’s use case of creating a wide range of matches (Numbers, capital letters, specific letters, characters, with one or more repetitions, etc.) to extract the required contents of an excel spreadsheet and build an industry-specific glossary for the services, specifications, and accreditation that are primarily used in the organisation. From our team’s review, RegEx can be used for data cleansing by verifying whether the input matches the text pattern, searching data or looking for a specific match in a large body of text, and replacing text matching the patterns with other text. With these features, users can specify a pattern for extracting necessary contents from an excel document, which will be contributed to the NER as training data for the ML model to learn and gradually build a standardized glossary of industry-specific legal terms to make smart predictions to identify and extract the correct entities.

**Table 5. Advantages and Disadvantages of Regular Expression**

| **Advantages** | **Disadvantages** |
| --- | --- |
| * Flexible * Replace blanks, special characters and null values * Split the text by delimiters or a specific character * Replace specific words such as spelling errors or typos * Search/Match a specific string of text | * Time-consuming * Complex behaviour can be difficult to design and debug * Pattern has to be created to match the pattern of treated words precisely |
|
|
|

However, the user will have to manually create patterns to precisely match the pattern of treated words, which will be time-consuming, and some treated words may not be extracted because there is no specified pattern to be matched. As such, our team has considered how these RegEx patterns provide an efficient way of data cleansing and can be an alternative method to be incorporated into our ML model, and web application, and used in a local context.

## Review Summary

Although powerful RegEx patterns can be created to perform data cleansing and extract specific information from text, they are limited to string patterns, which may not be the best method for Element's use case.

In conclusion, the library we have chosen to develop our learning model is spaCy as our goal for this project is to develop a solution that aims to solve the problem for our client efficiently and all the attributes in spaCy suit our needs.

# REQUIREMENTS ELICITATION

This section discusses the different requirements elicitation techniques employed in this project, the stakeholders identified for the product, and various functional and non-functional requirements identified for this project.

## Requirements Elicitation Techniques

Our team has decided to use **Interview** and **Prototyping** as our chosen requirements elicitation techniques to gather requirements for this project.

As our client was certain of the features to be implemented for this project, we have decided to use an **Interview** as a requirement elicitation technique to gather more information about the requirements our client would require of the web application. From the initial client meeting, the client introduced to us the work background and how the newly created web application will be useful in their work. After hearing what the client mentioned, our team was able to raise more questions to understand and clarify further the requirements of the project so that we were all on the same page. The meeting minutes can be found in [Appendix B: Requirements Elicitation Interview Minutes](#_heading=h.wt231yt243xc).

After understanding the requirements of the web application identified by the client, our team has also adopted **Prototyping** as another form of requirements elicitation technique. Our team used a coded prototype which acts as a solution close to the ready-to-release version of the end product we have in mind. During the client meeting, we showcased the prototype to allow the client to fully visualize the user interface of the web application. The coded prototype acts as a good foundation for a fully-functioning application. From there, the client was able to give us feedback on any design issues or features to be corrected. The coded wireframes can be found in [Appendix C: Wireframe](#_heading=h.9qfb02sqd13x).

## Stakeholder

The identified stakeholder of the web application is to be implemented for a user under the company of Element.

The user is able to utilize the web application platform to upload a file that is needed to be analyzed further. The user will be able to upload a selected file that contains columns of data to be analyzed by the Named Entity Recognition (NER) model. The user will be able to view the uploaded file on the web application and extract entities from it. The NER model will extract the entities and return a file analyzed by the model. The analyzed file will be used for further analysis.

## Functional Requirements

The functional requirements for the web application are:

1. The application will allow users to upload a file.
2. The application will allow the system to extract entities from the uploaded file.
3. The application will allow users to view the extracted entities.
4. The application will allow the system to display the progress of the ML model.
5. The application will allow users to download excel file results from the NER model.

## Non-Functional Requirements

The following are some of the non-functional requirements identified for the web application.

### *Performance*

1. The loading time within the web application to upload a file should not have a response time of more than 5 seconds.
2. The loading time within the web application to extract entities of a file should not have a response time of more than 30 seconds.

### *Maintainability*

1. Developers should be able to manage the system when a component is down and needs to be fixed in less than 2 hours.
2. Developers should be able to easily modify and extend the web application.

### *Usability*

1. The web application is designed to be a user-friendly interface.
2. Users should be easy to learn and be able to use the web application in a single page visit.
3. Users should be able to quickly perform their tasks.

# REQUIREMENTS PRIORITISATION

This section comprises the tables showing the prioritisation of requirements gathered from the interview and the cost estimates for these requirements.

|  | **AHP, Sub-Step 1: Compare requirements pairwise (Value)** |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Rid | Crit: Value | R1 | R2 | R3 | R4 | R5 |
| R1 | Allow users to upload a file | 1 | 5 | 3 | 3 | 3 |
| R2 | Allow the system to extract entities from the uploaded file | 1/5 | 1 | 6 | 3 | 1 |
| R3 | Allow users to view the extracted entities | 1/3 | 1/6 | 1 | 3 | 1 |
| R4 | Allow the system to display the progress of the ML model | 1/3 | 1/3 | 1/3 | 1 | 1/5 |
| R5 | Allow users to download excel file results from the NER model | 1/3 | 1 | 1 | 5 | 1 |

**Figure 3: AHP Sub Step 1 (Value)**

|  | **AHP, Sub-Step 1: Compare requirements pairwise (Cost)** |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Rid | Crit: Cost | R1 | R2 | R3 | R4 | R5 |
| R1 | Allow users to upload a file | 1 | 1/5 | 1/3 | 1 | 1/3 |
| R2 | Allow the system to extract entities from the uploaded file | 5 | 1 | 3 | 2 | 1 |
| R3 | Allow users to view the extracted entities | 3 | 1/3 | 1 | 3 | 1/5 |
| R4 | Allow the system to display the progress of the ML model | 1 | 1/2 | 1/3 | 1 | 1/5 |
| R5 | Allow users to download excel file results from the NER model | 3 | 1 | 5 | 5 | 1 |

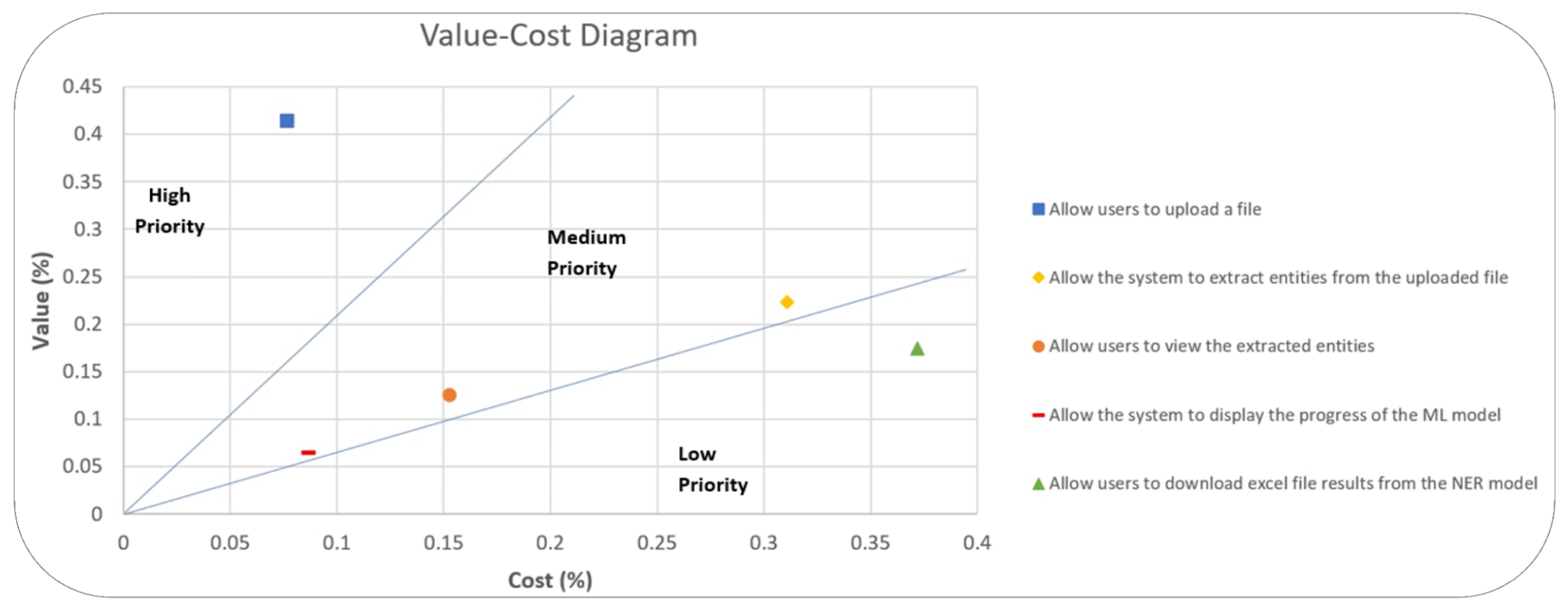
**Figure 4: AHP Sub Step 1 (Cost)**

|  | **AHP, Sub-Step 2: Evaluate how Crit distributes among regs (Value)** |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rid | Crit: Value | R1 | R2 | R3 | R4 | R5 | Relative Value |
| R1 | Allow users to upload a file | 0.45 | 0.67 | 0.26 | 0.2 | 0.48 | 0.414 |
| R2 | Allow the system to extract entities from the uploaded file | 0.09 | 0.13 | 0.53 | 0.2 | 0.16 | 0.223 |
| R3 | Allow users to view the extracted entities | 0.15 | 0.02 | 0.09 | 0.2 | 0.16 | 0.125 |
| R4 | Allow the system to display the progress of the ML model | 0.15 | 0.04 | 0.03 | 0.07 | 0.03 | 0.065 |
| R5 | Allow users to download excel file results from the NER model | 0.15 | 0.13 | 0.09 | 0.03 | 0.16 | 0.174 |

**Figure 5: AHP Sub Step 1 (Value)**

|  | **AHP, Sub-Step 2: Evaluate how Crit distributes among regs (Cost)** |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rid | Crit: Cost | R1 | R2 | R3 | R4 | R5 | Relative Cost |
| R1 | Allow users to upload a file | 0.08 | 0.07 | 0.03 | 0.08 | 0.12 | 0.077 |
| R2 | Allow the system to extract entities from the uploaded file | 0.38 | 0.33 | 0.31 | 0.17 | 0.37 | 0.311 |
| R3 | Allow users to view the extracted entities | 0.23 | 0.11 | 0.10 | 0.25 | 0.07 | 0.153 |
| R4 | Allow the system to display the progress of the ML model | 0.08 | 0.16 | 0.03 | 0.08 | 0.07 | 0.087 |
| R5 | Allow users to download excel file results from the NER model | 0.23 | 0.33 | 0.52 | 0.42 | 0.37 | 0.372 |

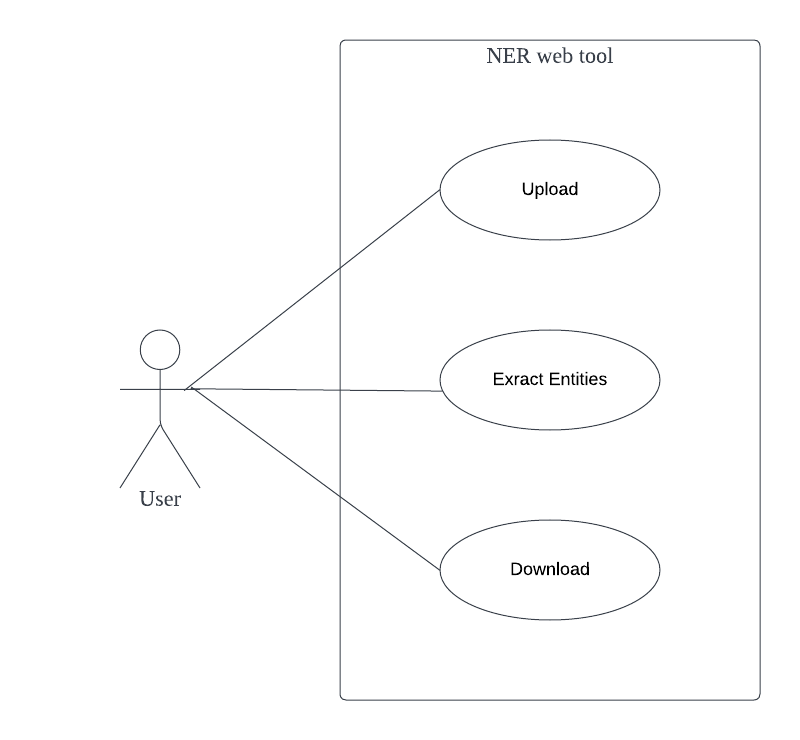
**Figure 6: AHP Sub Step 2 (Cost)**

****

**Figure 7: Value Cost Diagram**

# USE CASE DIAGRAM

This section comprises the use case diagram for our web application.



**Figure 8: Use Case Diagram**

## Use Case Diagram Description

In the following use case diagram, there is only one type of actor, the user. A user will be able to upload excel files to the web server and extract entities using the machine learning model. The user will also be able to download the results returned by the model.

# USE CASE DESCRIPTIONS

This section comprises some of the important use case descriptions for the use cases identified in the previous section. Alternatively, you may refer to [Appendix D: Use Case Descriptions](#_heading=h.kujygaw962pw) for all the use case descriptions.

**Table 6. Upload Use Case**

| User Case Title: | Upload |
| --- | --- |
| User: | Data Scientist |
| Brief Description: | Upload excel files |
| Pre-condition: | - |
| Post-condition: | User is able to extract entities found in excel data |
| Basic Flow: | 1. The user selects the browse button 2. The user selects the files that he/she wants to upload 3. The user confirms and presses the upload button 4. The system returns a success message 5. The use case ends successfully |
| Alternate Flow: | File found in web server |
| 4.1. The system returns duplicate file message  4.2. The use case ends successfully |
| No files selected |
| 4.1. The system returns no file selected message  4.2. The use case ends successfully |
| Files are not in .xlsx extension |
| 4.1. The system returns a wrong file extension message  4.2. The use case ends successfully |

**Table 7. Extract Entities Use Case**

| User Case Title: | Extract Entities |
| --- | --- |
| User: | Data Scientist |
| Brief Description: | The web server sends a request to the NER microservice to extract entities from the uploaded files |
| Pre-condition: | - |
| Post-condition: | Web server excel file with results from NER model |
| Basic Flow: | 1. The user clicks on the extract entities button 2. The system returns a loading message while processing data 3. The system returns a success message 4. The use case ends successfully |
| Alternate Flow: | Web server/ NER microservice is down |
| 3.1. The system returns an error message  3.2. The use case ends successfully |

**Table 8. Download Use Case**

| User Case Title: | Download |
| --- | --- |
| User: | Data Scientist |
| Brief Description: | User is able to download the excel file results from NER model. |
| Pre-condition: | - |
| Post-condition: | User has successfully downloaded the results. |
| Basic Flow: | 1. The user selects the download button. 2. The system sends the download file to the user’s system 3. The use case ends successfully |

# USER STORIES

This section comprises the user stories identified for the features in this web application.

## File Management

**Table 9. File Management Features**

| Features | User Stories |
| --- | --- |
| Upload | As a user, I want to be able to upload raw excel data, so that I can extract entities from the data. |
| Delete Files | As a user, I want to be able to remove unwanted raw data files from the web server, so that if I uploaded a wrong file, I can remove them. |
| View Files uploaded | As a user, I want to be able to view the files that I have already uploaded, so that I can keep track of the files that I am extracting entities from. |
| Download | As a user, I want to be able to download excel file results that are stored on the web server, so that I can get the results of the NER model. |

##### 

## Data Processing

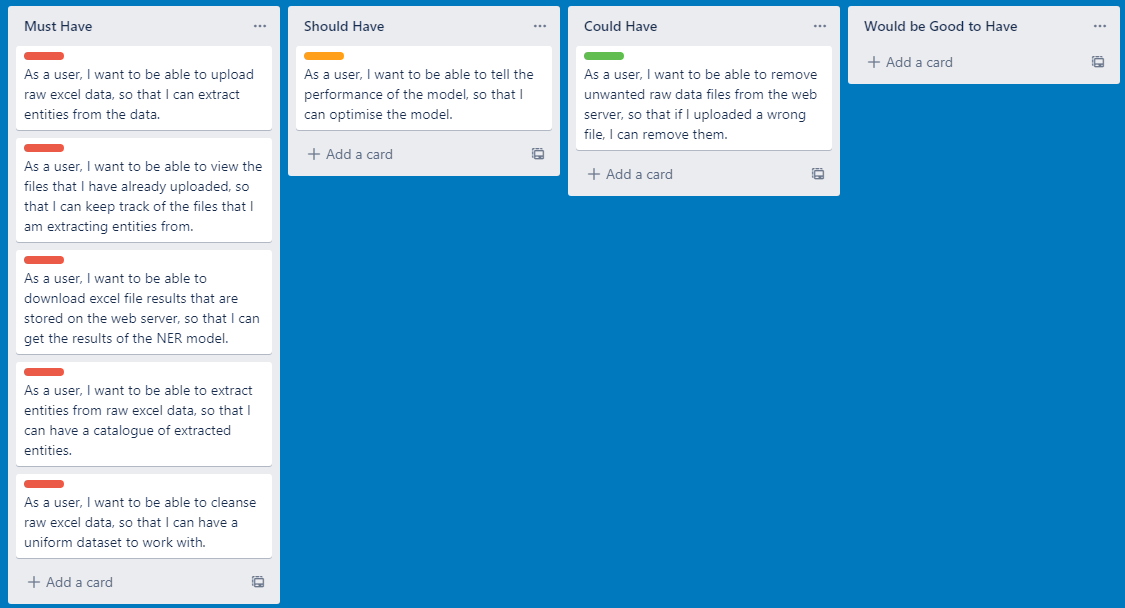
**Table 10. Data Processing Features**

| Features | User Stories |
| --- | --- |
| Extract Entities | As a user, I want to be able to extract entities from raw excel data, so that I can have a catalogue of extracted entities. |
| Data cleansing | As a user, I want to be able to cleanse raw excel data, so that I can have a uniform dataset to work with. |
| Performance | As a user, I want to be able to tell the performance of the model, so that I can optimise the model. |

##### 

# USER STORIES PRIORITISATION

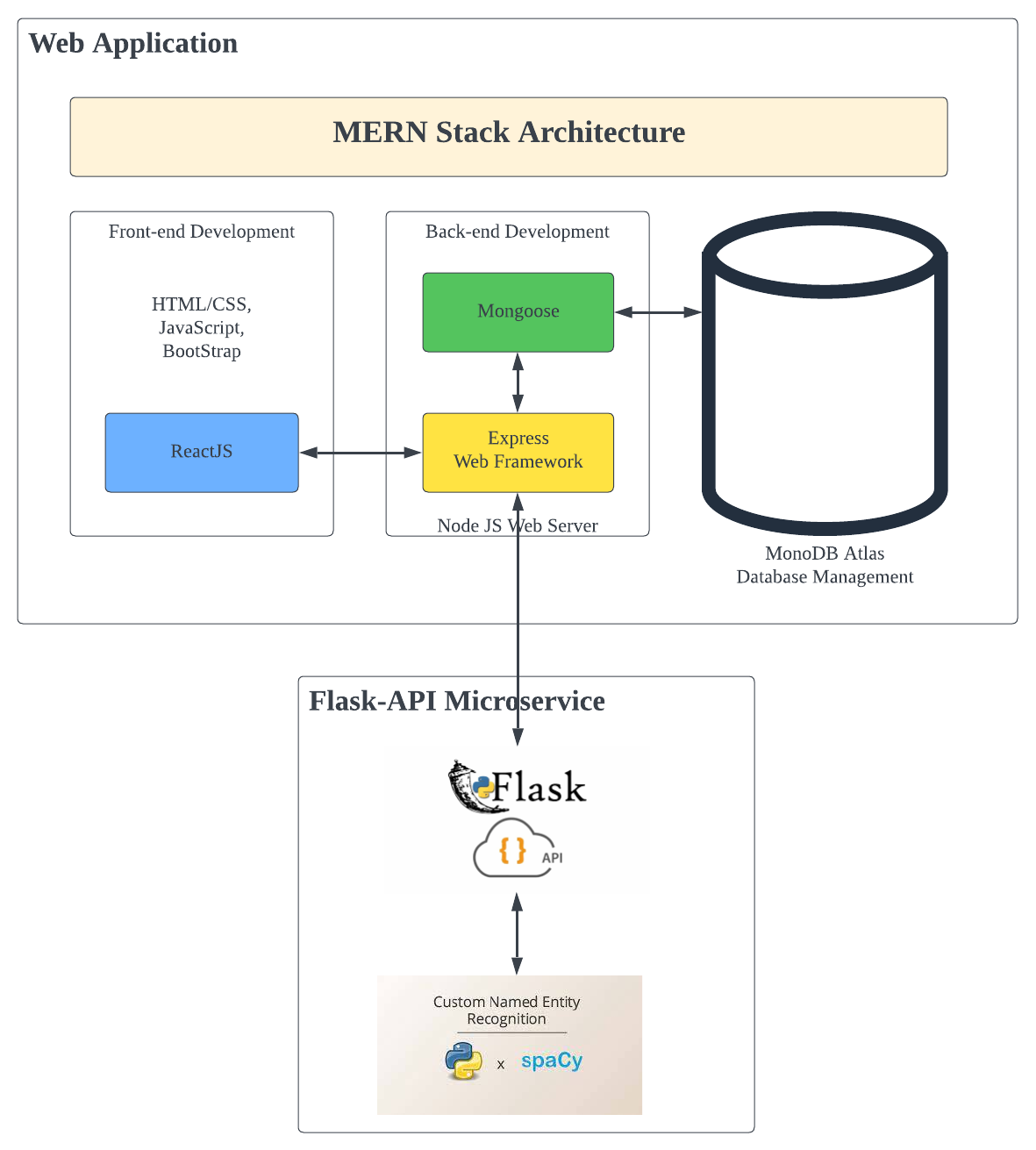
This section shows the Trello Board where we mapped the users’ stories derived from our requirements based on MOSCOW prioritisation.



**Figure 9: User Stories Prioritisation**

# ARCHITECTURE DIAGRAM

This section comprises the architecture diagram for our web application. Alternatively, you may refer to [Appendix E: Architecture Diagram](#_heading=h.opk9yx4x2280) to have a better view of the resource.



**Figure 10: Architecture Diagram**

## Architecture Diagram Description

The following architecture diagram highlights the overview architecture of the entire application. For this project, we designed and implemented a web application using the MERN stack architecture and a microservice hosting the developed NER model for the web application to call. This ensures that the web application and ML microservice are decoupled and independent, allowing the client to constantly change/ use/ implement multiple different NER models in the microservice without collateral on the web application code base.

## Flask-API Web Microservice

To decouple the machine learning model and the web application, the team developed a flask-api microservice to host the learning model, independent of the web application. This allows for an independent deployment, loosely coupled, and maintainable system. The client is allowed to host different machine learning models in this microservice and continually integrate and develop the microservice to include many different learning models without having much of an impact on the web application.

# FEATURES IMPLEMENTED

This section contains a list of features that were implemented in the web application.

## Upload

**Table 11. Upload Feature**

| **Screenshot** |
| --- |
|  |
| **Description** |
| The upload feature allows the users to ‘upload file’ to the web application which will be used for extracting entities in step 2. |

## Extract Entities

**Table 12. Extract Entities Feature**

| **Screenshot** |
| --- |
|  |
| **Description** |
| The extract entities feature allows the users to enable the NER model to ‘extract entities’ from the file that they have uploaded in step 1. |

## 

## Download

**Table 13. Download Feature**

| **Screenshot** |
| --- |
|  |
| **Description** |
| The download feature allows users to ‘download results’ of the extracted entities from the file that they have uploaded in step 1. |

# TESTING

This section comprises various testing techniques carried out during this project. They included Black Box testing and UI/UX Testing on the web application.

## Black Box Testing

Our team has created a document comprising various test cases for in-depth coverage to test different

functionalities, each with specific instructions to guide the users along the tests with screenshots. You may refer to [Appendix F: Black Box Testing Document](#_heading=h.unwxq591x8yn) to view all the test cases for end users. Each test case includes a test scenario, test case description, detailed test steps and expected results as shown in Table 20. The users are required to adhere to the test steps when carrying out a test and record any observations and the success/failure upon completion of each test case. We have created an online form for users to record their findings as they carried out the test so that our team can consolidate the results and find out any errors arising from failed test cases.

**Table 14. Upload Excel Test Case in Black Box Testing Document**

| Test Scenario | Upload Excel |
| --- | --- |
| Test Case | Check response upon uploading an excel file in the web application |
| Pre-Condition(s) | Nil |
| Test Steps | 1. Click on the ‘Choose File’ button.      1. Proceed to select the excel file with the .xlsx extension from your file directory 2. Click the ‘Upload File’ button. |
| Test Data | Sample File.xlsx |
| Expected Results | At Step 1, the ‘File Successfully Uploaded’ message should be displayed.      In Step 2, the preprocessed file should be shown. |
| Actual Results |  |
| Pass/Fail |  |

**Table 15. Extract Entities from Uploaded File Test Case in Black Box Testing Document**

| Test Scenario | Extract Entities from Uploaded File |
| --- | --- |
| Test Case | Check response of NER service. |
| Pre-Condition(s) | Uploaded file to be shown. |
| Test Steps | 1. Click on the ‘Extract Entities’ button. |
| Test Data | Nil |
| Expected Results | At Step 2, the excel file’s entities have been successfully extracted.    In Step 3, the excel file containing all extracted entities should be shown. |
| Actual Results |  |
| Pass/Fail |  |

**Table 16. Downloading Extracted Entities Result Test Case in Black Box Testing Document**

| Test Scenario | Download Extracted Entities Result |
| --- | --- |
| Test Case | Check the result of extracted entities to be shown and downloaded in the local directory. |
| Pre-Condition(s) | Uploaded file has been extracted in Step 2. |
| Test Steps | 1. Click on the ‘Download Results’ button. |
| Test Data | Nil |
| Expected Results | At the bottom of the web browser, it will show that the result has been downloaded as a ZIP file. |
| Actual Results |  |
| Pass/Fail |  |

## 

## UI/UX Testing

Our team has closely followed React’s documentation to design the user interface for our web application. During the development of the web application, our team utilise React Hooks as it produces a simpler and cleaner code structure while implementing the functionalities quickly and effectively. For making HTTP get and post between React express from the web browser and NER services, we closely adhered to the Axios documentation. Due to the short amount of time given for this project, our team came up with a simple web design that addresses the client's project requirements. During the client meeting, the client provided constructive feedback on the web application's user interface to ensure that it is to the client's standard.

# LIMITATIONS

This section highlights the limitations identified in this web application and the learning model applied.

## Accuracy of the Learning Model Dependent on Training Data

This project is using a blank model, that is empty and untrained with any data. Unlike those pre-existing models that spaCy has provided, the blank model has not been trained with those commonly used labels such as ‘PERSON’ or ‘ORG’. Therefore, we have to add in our customised labels, such as ‘SERVICES’, and compile our training data to train the model to recognise entities correctly.

Being an empty model, the training data we feed to the model is their solely learning source. If the training data provides the wrong information, such as wrong entities being learnt, it will affect the accuracy of the model and might produce inaccurate results.

The way to mitigate this limitation is by double-checking the training data that has been compiled before feeding it to the model to learn.

* 1. **Compiling of Training Data May Be Time Consuming**

The training data has to be following a specific data format that spaCy demands and highlight the key entities to be learned by specifying the starting and ending indices. This means that the training data has been compiled manually as any machine would not know what are the key entities to be recognised by the learning model. This may cause the compiling process of the training data to be time-consuming if there is a large dataset that needs the model to be trained with.

The way to mitigate this limitation is only to compile data that is unique and the model has not learned before. For those data that have a similar pattern, such as ‘SEP 1520’ and ‘SEP 1572’, the learning model can recognise the pattern and does not need to learn it again.

# CHALLENGES FACED

This section highlights some of the challenges faced in the course of development during the project.

## Training Process is Time Consuming

During the planning stage of this project, we did not anticipate the amount of data that needs to be trained by the learning model, which is a very time-consuming and labour-intensive process and thus, the one-week duration we planned on training is insufficient. When the training process is still undergoing and there is no model to work with, this set back our progress since the following development will require the model, to begin with.

To resolve this problem, the team agreed to come out with a sample model for the client to ensure that what we are working on is in line with the expectations of what the client wants us to deliver for this project.

## Inaccurate Word Tokenizers

Tokenization is the process of breaking down sentences into smaller segments, to produce tokens of words. The order of steps for the tokenizing of text:

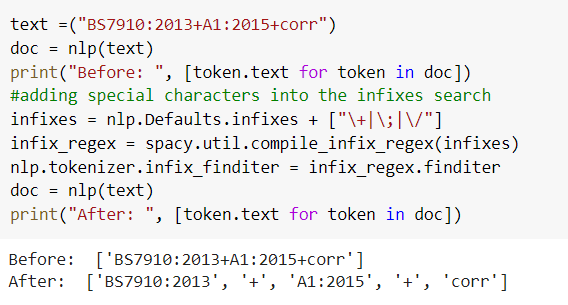
1. Split the text based on whitespaces which are similar to a split() function
2. Tokenizer checks if substring matches the exception rules.

For example, ‘don’t’ will be split into two tokens whereas ‘U.K.’ will be kept as one.

1. Tokenizer checks for the prefix, suffix and infix in the substring, for punctuations such as commas and periods.

However, some text is tokenized incorrectly and it resulted in errors while training the model in recognising the entities. For example, we want to train the model to recognise ‘BS7910:2013’ from a string "BS7910:2013+A1:2015+corr". There is an error while training the model since the tokenizer failed to break down the ‘BS7910:2013+A1:2015+corr’ into tokens of words and the machine presume that ‘BS7910:2013’ is a partial word thus, refusing to take it as a valid entity to be learnt.

To resolve this problem, we modified the infix rules by adding more special characters in the search to tokenize words that suit our use case.



**Figure 11: Python Snippet to Modify Tokenizer Rule**

# FUTURE WORKS

This section identifies some of the future works that encompass this project.

## Support Multiple Format Conversion

Currently, our web application only supports one file upload format:.xlxs. To make this web application more useful in context, we believe that supporting many other format conversions, such as CSV, word document, and so on, will bring out the full potential of the application to a larger group audience and improve the efficiency of data cleaning and entity extraction for the organisation.

## File Password Protection

When a user downloads the results from our web application, the results of the extracted entities from the ML model are encrypted in a ZIP file format. Password protection can be used to increase security in sensitive data, making the download file more secure.

# TECHNOLOGY AND TOOLS

This section discusses the technology and tools used throughout the course of this project.

## Project Management

We managed our project by using the Gantt chart and continually updated the progress on each of our responsibilities to keep track of the tasks to be completed. We used the Microsoft Azure DevOps environment that has been set up by our client which uses Git for the development and integration of our project.

## Communication Tools

We made use of Telegram for our daily communication and Discord for voice/ video call discussions. We also used Microsoft Teams for client communication.

## Development Tools

We utilized various frameworks to implement the entire software. ReactJS for front-end development, ExpressJS for server-side development, Mongo Atlas for a cloud database storage solution, NodeJS for server-side programming for its event-driven, non-blocking I/O and Flask-API for the development of the microservice. We used Postman to test and develop the RESTful endpoint in the microservice and the backend system.

## Data Sources

The following table shows the list of data sources used in this web application.

**Table 17. Data Source**

| **Data Source** | **Description** |
| --- | --- |
| Excel Sample Data | Made use of the sample data provided by the client, to be used for project purposes when training the machine learning model. |

# APPLICATION OF CONCEPTS TAUGHT

This section comprises the application of knowledge and concepts in this project from modules taught in previous academic semesters.

## ICT 1002: Programming Fundamentals

In this module, we learnt about the fundamentals of Python programming which includes the basic concept, such as file manipulation. We applied the concepts taught in this module by reading the input file from the web application to process and write the results on an output file which our user could download. Other intermediate operations such as merging and changing the arrangement of the file are also being demonstrated as a part of the requirements from our client. The approach to arriving at our solution may be different from how we are being taught, but the concepts applied are the same and this module helped us in the development of this project since we managed to pick up the knowledge learnt quickly.

## ICT 1004: Web Systems and Technologies

In this module, we learnt about the LAMP tech stack, development and deployment of web applications using AWS solutions. With the fundamental knowledge of what a tech stack is, our team member was able to make use of its concepts and translate them to the implementation of the MERN tech stack, although both have their significant differences, the initial exposure and fundamental knowledge have helped the team member to understand how web technologies are. We also learnt about vanilla HTML, JavaScript, BootStrap etc. which were heavily utilized in our project. Even though we used a frontend framework, ReactJS, the fundamental knowledge of web development was essential and played a critical role to aid our implementation. Our team member was able to pick up the framework quick enough with his understanding of Javascript from this module to aid in development. All in all, the application of these concepts in our project, has increased the efficiency of coding and helped up learnt new concepts more quickly.

## ICT 1007: Operating Systems

In this module, specifically, we were exposed to the WSL2 interface in windows and familiarised ourselves with the Linux operating system. With the use of the subsystem, we were able to develop the flask microservice application in WSL. Even though it was not a requirement to develop in WSL2, we decided to use what we were exposed to in this module to the fullest. Aside from subsystems, the concepts in this module such as memory, stack, scheduling, threads etc, have helped us understand how JavaScript works asynchronously with google chrome’s v8 engine. This has given us a deeper understanding of how asynchronous calls work, JavaScript’s call stack etc, in a single-threaded environment. This understanding was crucial when developing the backend of the web application as it is common practice to have a non-blocking UI for any sort of application to enhance user experience.

## ICT 1008: Data Structures and Algorithms

Add in any sort of things yall used for ur model optimization if any

In this module, we gained fundamental knowledge about data structures and were exposed to some basic data structures like trees, queues, stacks etc, and we learned how they could be used to optimize code or to aid in implementation. The information learnt in this module was especially crucial to this project as data structure played a huge role when implementing the solution. As the web application is constantly passing around JSON objects, we needed to understand what JSON objects were and how to construct/ deconstruct them or serialize and deserialize them to optimize message transfer. We utilized knowledge about dictionaries and applied our understanding to help us work with JSON objects. All in all, the application of these concepts in our project, has increased efficiency in performance and aided in implementation.

## ICT 1009: Object Oriented Programming

In this module, we learnt about object-oriented concepts. Although our web application does not contain classes etc, the knowledge about objects and the object-oriented paradigm has helped our team member to understand how ReactJS works better. Because ReactJS is component-based, and each component acts somewhat similarly to how a class would act, in fact, you could convert each component source code into a class-based source code however it is not commonly practised unless you are developing a library. From this module, we learnt how to create classes, their purpose, and more, and applied the knowledge when designing the React components. All in all, the understanding of how classes work has helped us work more efficiently with libraries.

## ICT 1010: Computer Networks

In this module, we learnt about ports, IP addresses, synchronous and asynchronous message transfer, how packets are delivered, how we can intercept and analyze packets, and how to build HTTP packets etc. This knowledge has helped us in the development of the backend solution. As the server, frontend, database and microservice all needed to communicate with each other through HTTP GET/ POST/ PUT/ DELETE calls, knowledge from this module helped us pick up on how the communication works. Aside from communication, when deploying our application locally, we required 3 different ports as we had a frontend, backend and microservice to deploy. Understanding ports from this module helped us in the implementation of the application.

## ICT 2101: Introduction to Software Engineering

This module has equipped us with important fundamental concepts in the software engineering process of a project. Specifically, our team utilized the agile approach and learnt how agile is helpful for incremental development especially since we are meeting the client after every 2 weeks. We also learnt about project integration and management from this module and applied it using Git which was also taught in this module.

## ICT 2103: Information Management

This module taught us fundamental concepts on database modelling and suitable database design for software projects. We were exposed to both relational and non-relational database technologies. In our project, we decided to use a NoSQL database solution through the use of MongoDB which was also taught in the module. We were also exposed to cloud technologies in this module which we did use in our project in the form of Mongo Atlas.

## ICT 2106: Software Design

This module taught us about software design and how it is used for code maintainability. Some concepts like microservices and software patterns have helped us gain a better understanding of how we could approach the design of our application. Most importantly, this module has stressed greatly decoupling solutions when needed. For our application, as we needed to decouple the machine learning model and the web application, we made use of microservices architecture in the software solution. All in all, the concepts taught in this model have helped us to design our software solution as best to the client’s needs.

## ICT 2108: Software Modelling and Analysis

This module gives us the in-depth knowledge and concepts we need to properly model and analyse requirements in the software development process. We recognised the significance of requirements elicitation as a critical component of the process for effectively gathering accurate and correct requirements, as it is a common cause of failure in software projects. This has aided us in communicating with and eliciting requirements from our client at the outset of this project. We used the techniques learned in this module in our project, such as conducting unstructured interviews rather than structured interviews, so that we could openly discuss the perceived problem, prompting supplementary questions in relation to their previous answers and exploring potentially overlooked issues. We've also learned effective interviewing techniques, such as how to make recommendations to our clients about web application features and ML model functionalities in order to understand the feasibility and importance of these features to them.

# INTRODUCTION AND APPLICATION OF NEW CONCEPTS

This section comprises the various new concepts learnt and applied throughout the course of this project.

## Web-related

### *ReactJS*

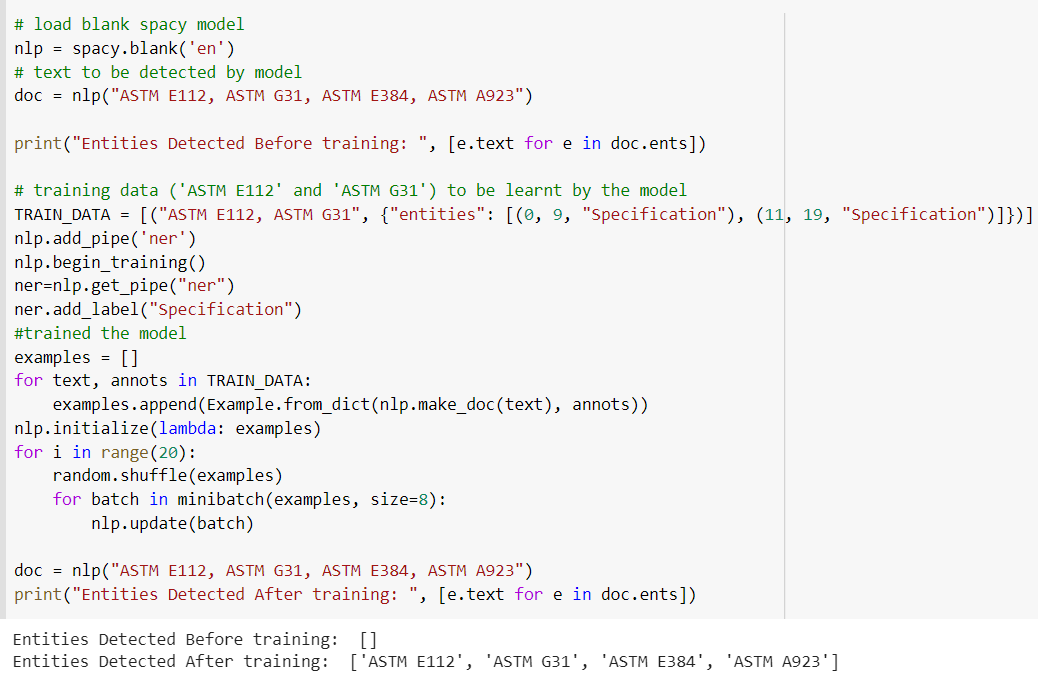
The ReactJS framework was used for front-end development to facilitate fast and interactive user interfaces for web applications. We learnt about how data is passed between react components and how to build these components. In an MVC architecture, react would often be in the view layer.

### *Flask-API*

The Flask-API framework was used to create an ML microservice. The Flask-RESTful extension was used to quickly build the REST endpoint to request the ML service. We learnt that this is a common use case/ architecture when needing to add machine learning models to a website built with Nodejs.

## Machine Learning related

The training process of the learning model with the training data is a part of machine learning. Once the model has been trained, not only it will be able to recognise the entities consisting of those training data, but also the pattern of such entities and thus be able to predict and detect those entities if they match with the pattern.



**Figure 12: Example of Outputs Produced by Model Before and After Trained**

The figure above shows an example of the results produced by a model before and after the training process has taken place. In addition, the trained model is not only able to detect the entity that is being fed to it, but also able to detect entities that have a similar pattern.

# FINAL OUTCOMES AND RESULTS OF THE PROJECT

Looking back at the initial objectives set during our project plan, it is safe to say that we have achieved most of the objectives set previously. Our web application allows users to upload raw excel data in the form of .xlsx and process it to extract entities. Our machine learning model makes use of 2 different methods to extract these entities, regex and machine learning based. The application then allows users to download the entities for their purposes.

# LESSONS LEARNT

Overall, this project has given our team an enriching and fruitful experience. For most of us, we had first-hand experience of working with a company to develop a product. We also learned a lot about time management on our own as, during the course of the project, we had complete freedom in our time management while still needing to deliver sprint updates, etc. As none of us had prior experience in machine learning, it was a whole new experience for the part of the team that needed to design and build the model. Most of us also only had the basic experience of web application development, furthermore, the tools used in school were simple and not what we used for this project, it was an enjoyable experience for the part of the team that needed to design and build the web application. Having a short span of 3 months to design the model, design the web architecture, accommodate the client’s requirements and ensure that both the machine learning portion and the web application integrate smoothly was a challenge as well as a learning experience. All in all, these lessons learned are beneficial experiences that will add to the portfolio of our experiences as we embark on future projects.

# 

# REFERENCES

[1] More Sectors [Internet]. Element. 2022 [cited 20 July 2022]. Available from: <https://www.element.com/more-sectors#:~:text=Element%20is%20a%20leading%20provider,Power%20Generation%20and%20Telecommunications%20sectors>.

[2] Purpose, Vision and Values [Internet]. Element. 2022 [cited 20 July 2022]. Available from: <https://www.element.com/about-element/purpose-and-values>

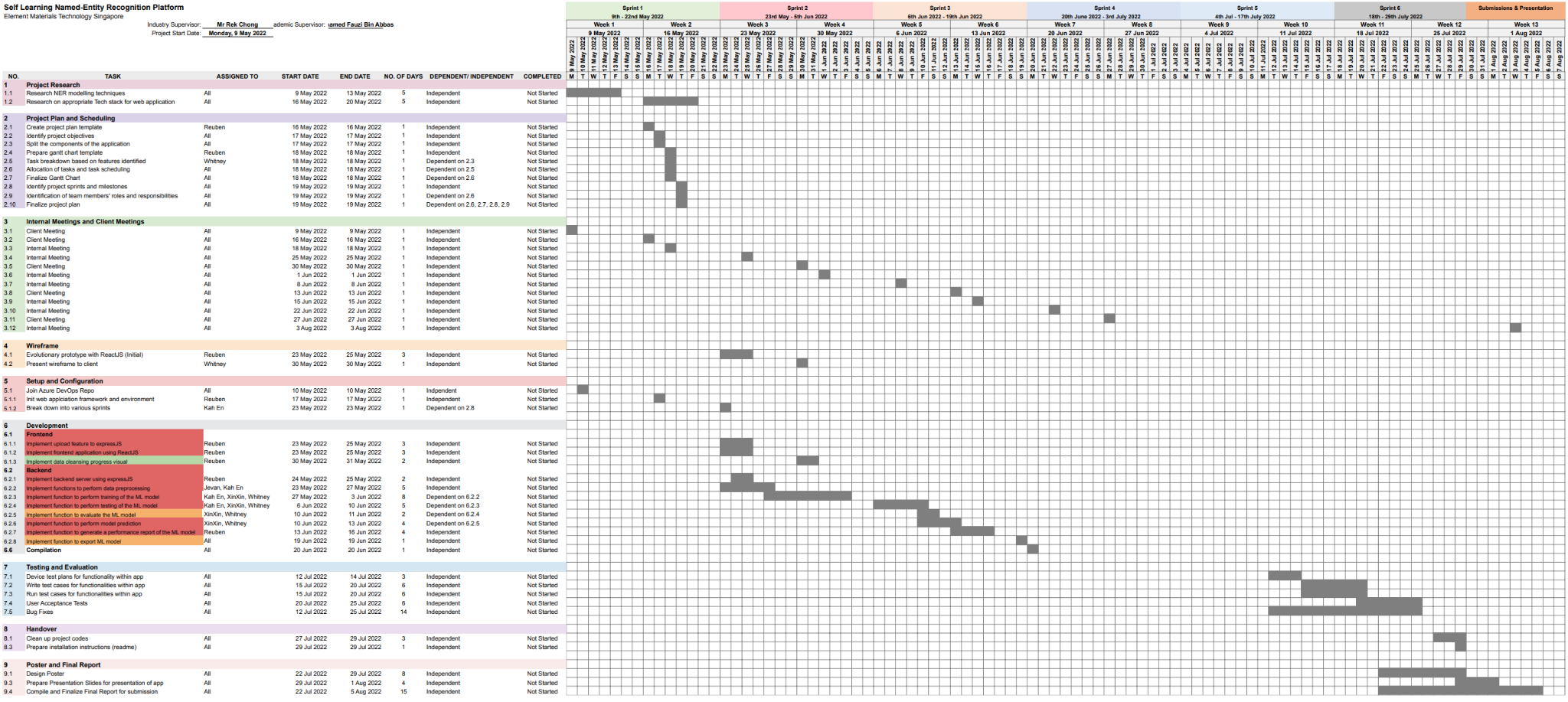
# 

# 

# APPENDICES

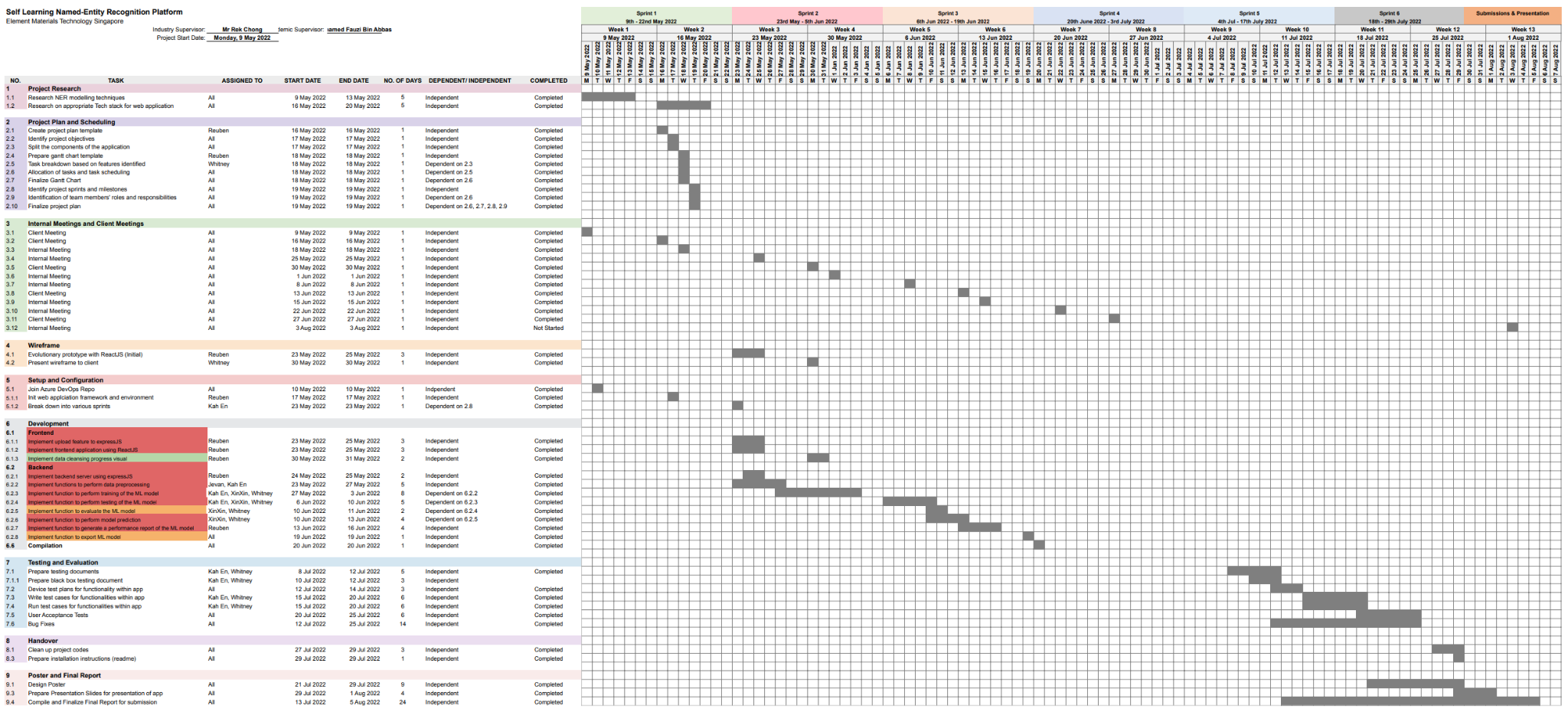
## APPENDIX A: GANTT CHARTS

Please refer to this link for the completed Gantt Chart: <https://docs.google.com/spreadsheets/d/1TK4QCX5R8uvKz_-tRvo41WRg3qutBWkVGNFvw137dU0/edit?usp=sharing>

****

**Figure 13: Gantt Chart (Start of Project)**

Please refer to this link for the completed Gantt Chart: <https://docs.google.com/spreadsheets/d/1dqx2xNxsYWTOe5dypCwvWqiQzys7_9tZ_dOzo9SK1Pw/edit?usp=sharing>



**Figure 14: Gantt Chart (End of Project)**

## APPENDIX B: REQUIREMENTS ELICITATION INTERVIEW MINUTES

**Table 18: Interview Minutes**

| Interviewer: Mr Rek Chong  Date: 16th May 2022  Venue: Microsoft Teams Meeting  Subject: Named-Entity Recognition Platform Requirements | | |
| --- | --- | --- |
| **No.** | **Interviewer Question** | **Interviewee Response** |
| **1.** | In the previous meeting, we noticed the excel sheet that was shown to us as an example of the data that we will be working on. Could you provide us with some details or definitions for each column so that we can understand its meaning and purpose of it as well as the data it contains in the columns? | *\*Client shows and explains sample excel sheet data\**  The first column, Key, is literally a key which links back to our original dataset. Service names are like your items or products. For example, when you’re at the supermarket, you will see different brands of milk. So these service names are just different types. But as you can see in the description, it’s almost free text writing. So how do you make sure you can extract. For example, ‘Liquid Penetrant Inspection - PT’, is actually one test/service. Another example I’ve used before like ‘United States’ or ‘United Kingdom’, individual words don’t mean anything but if you combine them it becomes something. The specification is like as it says here ‘ISO’ if you treat it as an item, it will be the same as a product. So the only difference here is some of these specifications are in different versions due to free text writing. People tend to put semicolons or commas or dashes to denote the second specification. So again, it requires a bit of cleaning and looking at the pattern that you can see in the dataset. For accreditation, the only clean-up is we do not want n/a, so if you could extract out that would be great. So how do we clean all this up, is what we’re looking for in the models. |
| **Supplementary Questions** | **Interviewee Response** |
| For cells with ‘n/a’, do you recommend removing it and leaving it blank? | Yes, you can replace it as blank. |
| After that, do we still take into account that row or column? | Ultimately, when you clean each column up individually, you will still want to link it back to the key. Because the key will give you the relationship back to everything. |
| **2.** | Could you provide us with a list of material testing standards that are commonly used with your current system? For example, the service name and specification column, the terms that are being used and we should take note of? As I can see there are some abbreviations being used. | Under the specification column, these are some terms for the standards like ‘BS EN ISO’ and ‘ASTM’. ‘ASTM’ is one of the most common terms. So these terms, you will have to research by yourselves. |
| **3.** | What is the background of this project and its purpose behind it? Could you give us an example of a situation/scenario of using this project solution? | There are countries and labs there. The key objective is that one of the problems we have in Element is that we grow by acquisition, so we buy a lot of companies every year. Last year we bought 12 companies, imagine having those 12 companies coming in with their own product name and own skill number. It will be hard to ask someone to do the mapping of the product and it will take forever to do. That is why we want to use AI and some sort of machine learning. Ultimately, what we’re trying to get to is if my customer calls me up and requests to do a bend test versus ASTM A370, which location do we do this service within our 250 labs. There is another separate set of files where we have 76 keys linking to different locations/countries and contexts as well, which is easier to clean as compared to this sample excel file. That is why we’re giving you this as a challenge.  Imagine you pulling out these terms like ‘Fatigue Testing’, so ideally you want to do some correlation or relationship back to ‘ASTM’. What the insides sales are doing now is that when a customer is asking for a fatigue test, they will go to this excel file and filter search the column on service name containing ‘Fatigue Test’ and list out cells that contain this term. Under specification, they will filter search the specific term ‘ASTM E10’. This is a time-consuming process due to free text writing.  This is why we need to come up with a data model where ‘Fatigue Test’ has this relationship with this specification. Think of it as almost like Facebook, friends of friends where you don’t want to build that relationship statically but instead more like a graph. Where one ‘Fatigue Test’ can have multiple relationships and the same ‘ASTM’ can have multiple relationships with something else as well. But before we get there, we have to clean this data up. |
| **4.** | What is the kind of listing are you looking for when extracting the entities? | Ideally, we can get to a listing where if I take away the duplicates it will be left with 197 records. The uniqueness of the listing that we can tie back to, could be a one to many. For an instance, in Key 1388, there will be three accreditation. So that’s the sort of listing we’re looking for. |
| **Supplementary Questions** | **Interviewee Response** |
| For the output file, how would it look like once it’s done extracting? | Ideally, it will be a few files of different listings. So maybe one file for Service Name, one file for Specification and one file for Accreditation. In each of the files, there should be a key that links to the service name/accreditation. |
| Will the user specify the condition and the file will be output? | Either you can do a website where they just upload the file and say these are my four columns (key, service name, specification, accreditation), or dynamically you can read the number of columns. For example if I have five columns, you’re gonna output four files because one of the column is the key. So always assume that there is always a key column where you can link everything back and any subsequent columns will have an individual file. This way it will be more advanced and dynamic but if you can do statically just based on these three columns (service name, specification, accreditation), more than happy to, since you guys only have three months to do it. |
| **5.** | In terms of frontend, how would the process be? For example, the user uploading the file and then producing the output? | Very simply, the user uploading the file. Also make sure there is always a key column which is your validation and the rest of the columns can be anything. For instance, if I give you another file with ‘Country’ column, then you produce the file. So the input is file upload and outputs are individual listings as individual files based on the number of columns, linking to a certain key.  The user interface is quite simple:   1. Upload 2. Do model training and learning 3. Output as individual file for download |
| **6.** | Are there any data cleansing rules when we train the model? For example, word replacement or punctuation removal, etc? | Yes, we have to look at the pattern. For instance, under the specification column, there’s ‘DIN EN 60068-2-1’, where there are ‘-1’ and ‘-2’ which are the versions. In this example, we have to keep the dashes. For this, I don’t have the answer just yet so we will have to look at the dataset to know more. |
| **7.** | For the frontend, my team suggested using ReactJS. Would it be okay for us to move in that direction? | Yes, sure that’s fine. |
| **8.** | Could you provide us with any reference materials/links or access to the previous group’s material if it would be helpful for us in any way? | Yes, I can give you access to what they have done previously. You can access using Azure DevOps under ‘Repos’ where the previous group have uploaded their documentation, powerpoint, different version of the coding, models, which is basically their source code. They try to use ‘Boards’ as their project management, so that’s what you can potentially do too. |
| **9.** | In the project brief there is a component where there will be a “report to track the performance of the model’, what data will there be shown in this report? | I think it will be good to show the accuracy. Because when you are doing your modelling, you have to decide on the scoring of your models. If you can show that on your report or on the website, at least you know that when you’re training it you know how this model is performing. With this validation, you can know how well it is performing using the accuracy as you iterate through different models and trying out different solutions to see how that score range over time. |
| **10.** | For the testing phase in the model, will there be any other data that we will be provided? | I’d say you can stick to the sample file provided, which contains almost 17,000 rows of records. So if you split them up, that will be the dataset you use. |

## APPENDIX C: WIREFRAME

**Table 19: Wireframe**

|  |
| --- |
| This screenshot shows the main screen when the user first opens the web application. The user will be able to upload a file in step 1 and in step 2, the user will be able to extract the entities from the uploaded file. |

## APPENDIX D: USE CASE DESCRIPTIONS

**Table 20. Upload Use Case**

| User Case Title: | Upload |
| --- | --- |
| User: | Data Scientist |
| Brief Description: | Upload excel files |
| Pre-condition: | - |
| Post-condition: | User is able to extract entities found in excel data |
| Basic Flow: | 1. The user selects the browse button 2. The user selects the files that he/she wants to upload 3. The user confirms and presses the upload button 4. The system returns a success message 5. The use case ends successfully |
| Alternate Flow: | File found in web server |
| 4.1. The system returns duplicate file message  4.2. The use case ends successfully |
| No files selected |
| 4.1. The system returns no file selected message  4.2. The use case ends successfully |
| Files are not in .xlsx extension |
| 4.1. The system returns a wrong file extension message  4.2. The use case ends successfully |

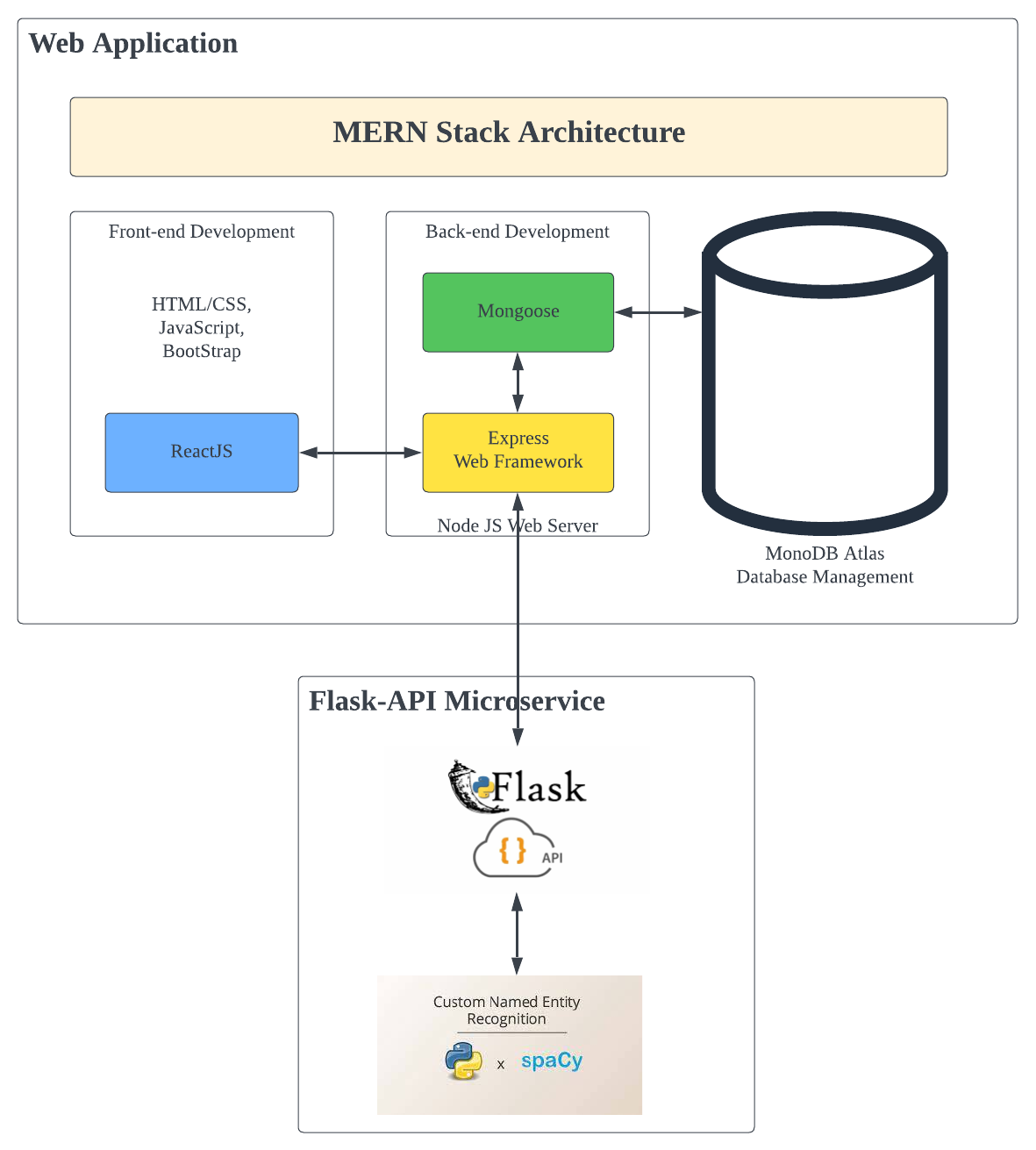
**Table 21. Extract Entities Use Case**

| User Case Title: | Extract Entities |
| --- | --- |
| User: | Data Scientist |
| Brief Description: | The web server sends a request to the NER microservice to extract entities from the uploaded files |
| Pre-condition: | - |
| Post-condition: | Web server excel file with results from NER model |
| Basic Flow: | 1. The user clicks on the extract entities button 2. The system returns a loading message while processing data 3. The system returns a success message 4. The use case ends successfully |
| Alternate Flow: | Web server/ NER microservice is down |
| 3.1. The system returns an error message  3.2. The use case ends successfully |

**Table 22. Download Use Case**

| User Case Title: | Download |
| --- | --- |
| User: | Data Scientist |
| Brief Description: | User is able to download the excel file results from NER model. |
| Pre-condition: | - |
| Post-condition: | User has successfully downloaded the results. |
| Basic Flow: | 1. The user selects the download button. 2. The system sends the download file to the user’s system 3. The use case ends successfully |

## APPENDIX E: ARCHITECTURE DIAGRAM



## APPENDIX F: BLACK BOX TESTING DOCUMENT

**PARTICIPANT CONSENT FORM**

*The consent form is to seek your permission to participate in the ICT2211 Integrative Team Project Black Box Testing conducted by second year undergraduate students in the Singapore Institute of Technology, as a part of their project coursework.*

*We value your effort and for taking the time to help us evaluate the following web application for our project.*

*We would like to seek your understanding in providing honest results after the completion of the test cases in this document.*

*Upon completion of the test cases, please proceed to the following web address below:*

**DATA PRIVACY**

All the personal information (i.e. the email address) provided will be handled securely, with the access restricted to only the people involved in this project. This data will not be shared with any third-party or used for commercial purposes.

**Test Case 1**

| Test Scenario | Upload Excel |
| --- | --- |
| Test Case | Check response upon uploading an excel file in the web application |
| Pre-Condition(s) | Nil |
| Test Steps | 1. Click on the ‘Choose File’ button.      1. Proceed to select the excel file with the .xlsx extension from your file directory 2. Click the ‘Upload File’ button. |
| Test Data | Sample File.xlsx |
| Expected Results | At Step 1, the ‘File Successfully Uploaded’ message should be displayed.      In Step 2, the preprocessed file should be shown. |
| Actual Results |  |
| Pass/Fail |  |

**Test Case 2**

| Test Scenario | Extract Entities from Uploaded File |
| --- | --- |
| Test Case | Check response of NER service. |
| Pre-Condition(s) | Uploaded file to be shown. |
| Test Steps | 1. Click on the ‘Extract Entities’ button. |
| Test Data | Nil |
| Expected Results | At Step 2, the excel file’s entities have been successfully extracted.    In Step 3, the excel file containing all extracted entities should be shown. |
| Actual Results |  |
| Pass/Fail |  |

**Test Case 3**

| Test Scenario | Download Extracted Entities Result |
| --- | --- |
| Test Case | Check the result of extracted entities to be shown and downloaded in the local directory. |
| Pre-Condition(s) | Uploaded file has been extracted in Step 2. |
| Test Steps | 1. Click on the ‘Download Results’ button. |
| Test Data | Nil |
| Expected Results | At the bottom of the web browser, it will show that the result has been downloaded as a ZIP file. |
| Actual Results |  |
| Pass/Fail |  |

## 