

# Document Submission System

## Software Requirements Specification

### Software Engineering Project| Group 28

*List of your Names:*

<b>Name</b>	<b>Position</b>	<b>Email</b>	<b>Phone</b>
<b>Jun Wee Tan</b>	Team Leader	101231636@student.swin.edu.au	0493461576
<b>Adrian Sim Huan Tze</b>	Team Member	101225244@student.swin.edu.au	0474642003
<b>Xin Zhe Chong</b>	Team Member	103698851@student.swin.edu.au	0406259449
<b>K.M. Yovinma M. Konara</b>	Team Member	102426323@student.swin.edu.au	0411101072
<b>Richard Ly</b>	Team Member	103340644@student.swin.edu.au	0424391144
<b>Sandali Jayasinghe</b>	Team Member	102849357@student.swin.edu.au	0470617473

*SWE40001, SOFTWARE ENGINEERING PROJECT A, SEMESTER 1 AND 10/04/2022*


## - Document Change Control

Version	Date	Authors	Summary of Changes
0.1	07/04/2022	Adrian Sim Huan Tze	The entire document is formatted to the Cambria font style with 10-11 font size. All remarks are removed and put in the comment panel.
0.2	08/04/2022	Adrian Sim Huan Tze	The overview of User Interfaces is added. Brief description of Hardware Interfaces is added.
0.3	08/04/2022	Jun Wee Tan	Completed section 5 – High-Level system architecture
0.4	09/04/2022	Adrian Sim Huan Tze	Wireframes of Login, Assignment Submission and Admin Interfaces are added along with the descriptions.
0.5	09/04/2022	Xin Zhe Chong	Completed section 2.3 - Acceptance Criteria, 2.4 - Documentation, 3 – Functional Requirements, 4 – Non-functional Requirements.
0.6	10/04/2022	Richard Ly	<ul style="list-style-type: none"> <li>Added more Definitions, Acronyms and Abbreviations in Chapter 1.3 - Definitions, Acronyms and Abbreviations.</li> <li>Rewrote Chapter 1.2 - Scope</li> <li>Reviewed document for typo errors.</li> </ul>
0.7	10/04/2022	Sandali Jayasinghe	Completing sections 6.3, 6.4 -software interfaces and communication interfaces. Reviewed document for grammar and font errors.
0.8	10/04/2022	Adrian Sim Huan Tze	Table of Contents added. Revised and reviewed the entire document.
0.9	10/04/2022	Jun Wee Tan	Final review on the entire document
1.0	16/05/2022	Jun Wee Tan	Revised and added: Section 5 High-level system architecture Section 3 functional requirement <ul style="list-style-type: none"> <li>task 9: Send a summary of MCQ results to the convenor via email</li> </ul>
1.1	16/05/2022	Adrian Sim Huan Tze	Final review on every section before submission Changes made: 1. Functional Requirements 2. Scope

## - Document Sign Off

Name	Position	Signature	Date
Jun Wee Tan	Team Leader champion		10/04/2022
Adrian Sim Huan Tze	Team Member		10/04/2022
Xin Zhe Chong	Team Member		10/04/2022
Richard Ly	Team Member		10/04/2022
Sandali Jayasinghe	Team Member		10/04/2022
K.M. Yovinma M. Konara	Team Member		10/04/2022

## - Client Sign off

Name	Position	Signature	Date
Caslon Chua	Client		13/05/2022
<b>Organisation</b>			
Department of Computing Technologies at Swinburne University of Technology			

## Contents

-	Document Change Control.....	2
-	Document Sign Off .....	3
-	Client Sign off.....	3
1	Introduction.....	5
1.1	Purpose.....	5
1.2	Scope.....	5
1.3	Definitions, Acronyms and Abbreviations .....	6
2	Overall Description .....	7
2.1	Product Features .....	8
2.2	System Requirements.....	8
2.3	Acceptance Criteria .....	9
2.4	Documentation .....	10
3	Functional Requirements .....	11
4	Non-Functional (Quality) Requirements .....	17
5	High-Level System Architecture.....	19
5.1	Domain Model .....	19
5.1.1	High-level system architecture .....	21
6	Interface Requirements .....	23
6.1	User Interfaces .....	23
6.1.1	Login Interface.....	24
6.1.2	Assignment Submission Interface .....	25
6.1.3	Admin Interface .....	26
6.2	Hardware Interfaces .....	27
6.3	Software Interfaces .....	27
6.4	Communication Interfaces .....	27
7	References .....	28

# 1 Introduction

The software that will be designed is a Document Submission System. This application will be developed by the team with the purpose of enabling Swinburne students to submit their work and analyse it to produce a text-based analytic report which will check for plagiarism through the extraction of keywords from the submitted document and followed by answering a series of multiple-choice questions.

## 1.1 Purpose

The purpose of this document is to outline the requirements needed for this project to be deemed successful. The documentation for this project will provide a framework for Team 28 to follow to ensure the project is completed.

The target audience for the SRS is the client, supervisor and the team to be used as a reference in understanding and comprehend the nature of this project.

## 1.2 Scope

The scope of the Software Requirements Specification or SRS is to cover what is needed to develop the Document Submission System.

The application should have the features of authenticating users such as Students, Convenors and Admin to prevent unauthorized access with the help of Login module. One of the key functionalities of the application is to enable students to upload their assignments (drag-and-drop) to the portal. The system should be able to accept only PDF documents.

This application will be able to extract words and analyse submitted PDF documents and will look to convert selected sentences and formulate it into questions quickly and then perform an analysis of the submitted work to produce a text-based analytic report which will include the name and doc ID, name of student and ID, keywords used, multiple-choice questions and answers submitted by students, sentiment analysis results, web search results and references used.

Once that process has been completed, the application will then produce a series of questions within a quick timeframe to the students that they will need to answer for integrity testing. A summary of the questions and answers from the students will be emailed to the convenor at the end of the day.

This project will not be able to perform these tasks in any other languages aside from English. The document submission system is not to be confused for another "Turnitin" as this application will only extract words in the document and perform a web search to see if there are solutions already listed. Noticeable examples include sites such as StuDoc, Course Hero and other sites which has solutions to university-based work.

### 1.3 Definitions, Acronyms and Abbreviations

- **REST** - Representational State Transfer
- **HTTPS** - Hypertext Transfer Protocol Secure
- **API** – Application Programming Interface
- **AWS** –Amazon Web Services
- **RDS** – Relational Database Service
- **SRS** – Software Requirements Specification
- **SQAP** – Software Quality Assurance Plan
- **GUI** – Graphical User Interface
- **GITHUB** – Git and Hub
- **STP** – Software Test Plan
- **SDLC** –Software Development Cycle
- **NPL** – Natural Processing Language
- **DB** – Database
- **PDF** – Portable Document Format
- **MCQ** – Multiple Choice Questions
- **PHP** – A side scripting language used for general purposes
- **AWS EC2** – Amazon Web Services Elastic Compute Cloud
- **CSS** – Cascading Style Sheets
- **IDE** – Integrated Development Environment

## 2 Overall Description

The document submission system is not to be mistaken with a plagiarism checking system that is most common among document submission. Most document submission systems where assignments are submitted would be for the purpose of plagiarism or grammar check and would often appear as a monthly subscription service where both functionalities are included. This document system is designed with the purpose of being accessible to all types of users found in the public and will not be restricted strictly for academic purposes.

This product is a web-based application that is newly created to allow the submission of text-only documents in PDF format. The system will conduct analysis on the document submitted and generate a minimum of five questions along with answers that the individual who submits the document will have to answer to finalize submission. Furthermore, the system will also produce a summarized version of the document contents, which will be stored on a database for the data administrator to access. Finally, based on keywords and the document title found in the analysis of the document, a web search will be conducted in which the information discovered will be uploaded to a database on a cloud-based server. The solution will not be a part of a larger system even though cloud-based services will be utilized for hosting purposes.

Diagram below presents a representation of how the product components work with each other to produce the expected outcomes:

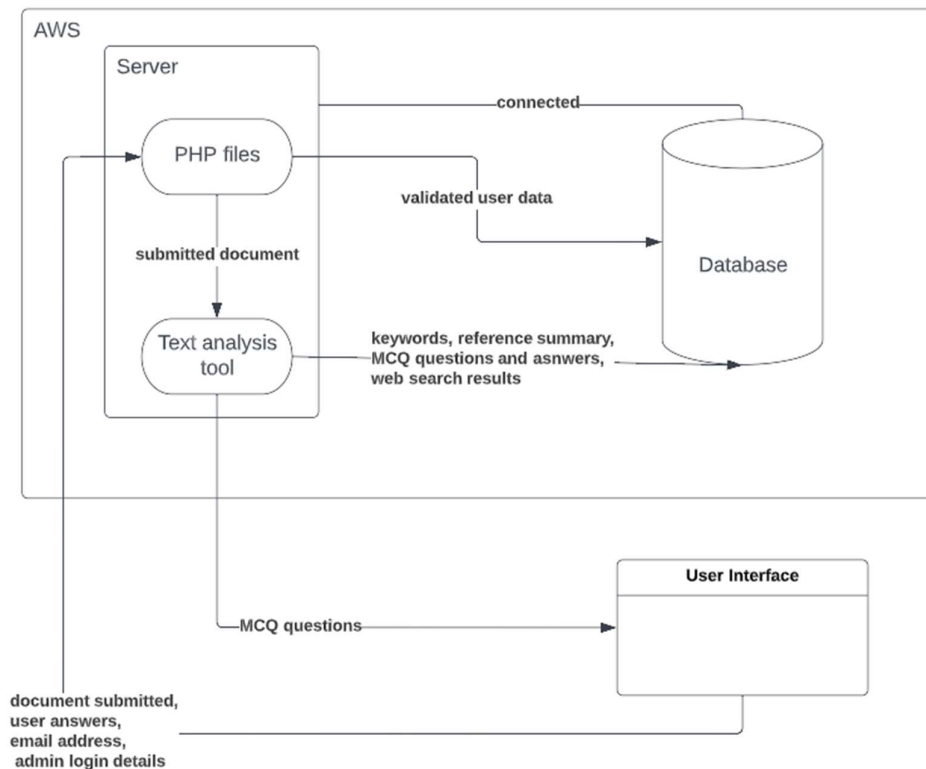


Figure 1 Product components used in system

## 2.1 Product Features

The main product features of the product are identified as following:

- Generate 5 Multiple Choice Questions (MCQ) with answers via analysis
- Ability to record the individuals' personal answers to the MCQs generated
- Ability to analyse documents and extract keywords and document title from the document submitted
- Send an email notifying convenor about the summary of MCQs and answers
- Store analysis information such as analysis report, keywords extracted, MCQ and answers to MCQ by an individual and the actual answers for MCQs in a table in the database
- Ability for an admin to access data related to analysis results in database
- Minimalistic design for easy interpretation and usage for end-users
- Produce an analysis report from document submitted for the convenor to view. The report will consist of the following
  - Name of document submitted and document ID (generated by database)
  - Name of Student and the student's ID (ID provided by the student)
  - Keywords extracted from document
  - MCQ questions generated, along with the answers by the student and actual answers
  - Score of how many questions the student correctly answered
  - Results from sentiment analysis (writing style, and the use of emotion)
  - Web search results based on the keywords and document title searched
  - List of references used in the document (if any available)

## 2.2 System Requirements

In this context, it is important that the terms “development” and “deployment” mean:

- Development stage describes the tools required for the creation of software to be provided to the end users
- Deployment stage describes the components that would be handed over to the client when the system has finished developing

The minimum requirements to develop and deploy the document submission system is as follows:

Component	Software	Implementation stage	Purpose
Server	✓	Deployment and Development	To host web application for usage for client and developers and testing for developers The server would be acquired by a cloud service provider such as Amazon Web Services (AWS) EC2 The user interface will be available through the server (updated with the newest version)



Github repository	✓	Deployment and Development	Any updates or changes to the system during and after project development will be documented and available here for the end users to view
Relational Database System (RDS)	✓	Development	To store text analysis results such as keywords, MCQs and their answers generated, the user's personal answers to MCQs and reference summary in a protected and organized manner The RDS would be acquired by a cloud service provider such as AWS or Google Cloud
User Interface	✓	Deployment	To display relevant analysis results such as MCQs generated for the user to answer To display and request for user prompts such as email addresses and display notifications relevant to document submission To access admin login
IDE	✓	Development	Utilizes IDEs such as Visual Studio Code, Visual Studio, Atom or Notepad++ for creation of user interface and the web server.
Text Analysis Tool	✓	Development	For conducting text analysis to extract, summary content, keywords and content for MCQ generation, along with word frequency checking for web search purposes Tools such as Google Cloud NLP, IBM Watson and MonkeyLearn can be utilized to create customized text analysis

As the solution is a web-application, there will be no hardware requirements for the deployment of the product.

## 2.3 Acceptance Criteria

### Project

The following criteria applies to all aspects of the project:

- The project goals and objectives (as defined in the Project Plan) are met.
- All deliverables meet the requirements specified within the document.
- Communication between the project team and the client is consistent and clear.
- The project outcomes are delivered within the appropriate timeframes:
  - Prototype System and accompanying documentation (if applicable): End of Semester 2, 2022
- At the project's completion, all the deliverables and supporting artefacts are transferred to the client.

### **Research Report**

The Research Report must comply with the following acceptance criteria

- The report provides a clear indication as to the state of contracting cheating and plagiarism.
- The report contains enough supporting information to allow the client to make an informed decision regarding the continuation of the project.

### **Prototype System**

- The prototype system is of a similar quality to common industry counterparts, with appropriate concessions made due to its “proof-of-concept” nature.
- The prototype system is accompanied by user documentation that a university level computer science or engineering student could reasonably use to recreate the system.

## **2.4 Documentation**

Documentation to be delivered with the project as follows:

- User Manual – this document will also act as a technical menu which will contain details of how to operate and maintain the system, along with how to contact for support or resolve any technical issues
- Git repository with the code required to run the prototype system

### 3 Functional Requirements

#### Task 1: Logging in to the Document Submission System

Task: Logging in to the Document Submission System	
Purpose:	Ensure that only authorized students can access the Document Submission System.
Precondition:	The student wants to submit a document and visits the web application.
Frequency	Average 1000 to 2000/day. Mostly occurring on weeks 5, 6, 10, and 12 in the semester.
Critical:	More than 2000 users trying to access the system at the same time.
Subtasks:	Example Solution:
1. Validate user input	The system checks if the student enters the credentials before logging them in
2. Check existing record	The system compares the credentials with existing records in the database and returns a result
3. Redirecting to the home page of the Document Submission System	The student will be redirected to the home page of the Document Submission System to start uploading their documents. Authorized login required.

**Task 2: Submit a Document**

<b>Task: Submit a Document</b>	
Purpose:	Allow the system to start the document analysis.
Precondition:	The student wants to submit a document.
Frequency	Average 1000 to 2000/day. Mostly occurring on weeks 5, 6, 10, and 12 in the semester.
Critical:	More than 2000 students are submitting the document at the same time
Subtasks:	Example Solution:
1. Check for file extension Problem: Invalid file extension	The system checks if the file extension from the submitted document is within the list of acceptable file extensions. E.g. .docx, .pdf System will alert the student to convert the file type or to submit a different document.
2. Validate user input.	The system checks if the user has selected the units or not before accepting their submissions.
3. Save document	The document is automatically saved by the system into the database.

**Task 3: Analysing the Document**

<b>Task: Analysing the Document</b>	
Purpose:	Ensure that the document submitted is not plagiarised.
Precondition:	The system receives the submitted document.
Frequency	Average 1000 to 2000/day. Mostly occurring on weeks 5, 6, 10, and 12 in the semester.
Critical:	More than 2000 documents requiring analysis at the same time.
Subtasks:	Example Solution:
1. Perform keyword extraction	The system extracts the keywords from specific sections of the document. E.g. (Introduction, abstract, conclusion)
2. Perform analysis	The system uses a Natural Language Processing algorithm (NLP) to perform analysis on the document such as syntax analysis, sentiment analysis, entity analysis, entity sentiment analysis, and text classification.
3. Storing the results	The system stores the results of the analysis into the database.

**Task 4: Generate the MCQ questions and answers**

<b>Task: Generate the MCQ questions and answers</b>	
Purpose:	Provides evidence if the student plagiarised the document.
Precondition:	The student successfully submitted a document, the system completes the document analysis.
Frequency	At least 5 per document submission
Critical:	
Subtasks:	Example Solution:
1. The system generates the multiple-choice questions.	The system generates at least 5 multiple choice questions based on the details extracted the document.
2. The system generates the correct and incorrect answers for the questions.	The system generates 4 to 5 answers for each of the questions. Only 1 correct answer is generated.
3. Save MCQ questions, answers and student's answers.	The details of the MCQ is automatically saved by the system into the database.

**Task 5: Generate Analysis Report**

<b>Task: Generate Analysis Report</b>	
Purpose:	Provide the information to the convenor for analysing and making informed decision on whether the student has plagiarised content.
Precondition:	The student submits the MCQ answers, and the system successfully analyses the document.
Frequency	With each document submission
Critical:	
Subtasks:	Example Solution:
1. Collect analytic data	System queries database to provide statistics on Sentiment Analysis, References Summary, the MCQ questions, correct answers, and student's answers.
2. Generate report	The system outputs the result of the analysis to a PDF document.

**Task 6: Send a summary of MCQ results to the convenor via email**

<b>Task: Send a summary of MCQ results to the convenor via email</b>	
Purpose:	Provide a summary of students' MCQ results to convenor as reference.
Precondition:	The student submits the MCQ answers, and the system successfully analyses the document.
Frequency	One summary email to convenor per day
Critical:	More than 2000 records needed to be sent every day
<b>Subtasks:</b>	<b>Example Solution:</b>
1. Gather student results that have same convenor.	Retrieve information (Student Id, Student MCQ question and result) from database that have same convenor email. Combine and format them for output.
2. Send notify email to convenor. <b>Problem:</b> Convenor is not a subscriber in document submission system notification service	1. Triggers notify function at every day 9:00 am 2. Get the combined information 3. Loop through convenor subscriber list and find corresponding convenor email. 4. Send Student Id, MCQ question and results to convenor email. <b>Problem solution:</b> add convenor to subscriber list.



## 4 Non-Functional (Quality) Requirements

The major non-functional requirements of the project are as follows:

- Accessibility
- Usability
- Durability
- Modifiability

The above requirements apply to the deliverables in the following ways:

### **Prototype System**

#### ***Accessibility***

- The system should be able to run on all computers running on Windows 10 or newer, with an internet connection.

#### ***Usability***

- User documentation should be reasonably understood by 90% of university level computer science or information technology students.
- The graphical user interface (GUI) of the system should be easily comprehensible by 90% of university level students.

#### ***Durability***

- The document submission system cloud database (AWS RDS db.t2.micro instance) should be able to store up to 20 GB of data (documents, MCQ questions, answers, results) with at least 90% efficiency and less than 1% of downtime.
- The extra 20 GB of backup storage will help document submission system to backup data automatically

#### ***Modifiability***

- Design of the system should be at least 50% adaptable to different types of use cases (modular) to simplify future development.

#### ***Security***

It is important that the system prevents any unauthorized access and all data that is stored is safe.

To meet the required Security the system must:

- Have a log in system to stop unauthorized users from accessing
- Restrict the access that different users have to different parts of the system.
- Safely store information so that it cannot be accessed outside of the application.

### ***Availability***

It is important the system can handle many requests from many users at once. With a capacity of 2000 users, if the system can't handle this it will increase waiting times overall. Furthermore, the slowdown of the system may cause frustration in staff or students.

- Can process 500 submissions simultaneously without any slowdown.
- Be always available.
- Any downtime of the system for maintenance can be done without disrupting the system
- In the case of unforeseen circumstances where the system needs to be reset, the down time is no more than 10 minutes.

## 5 High-Level System Architecture

### 5.1 Domain Model

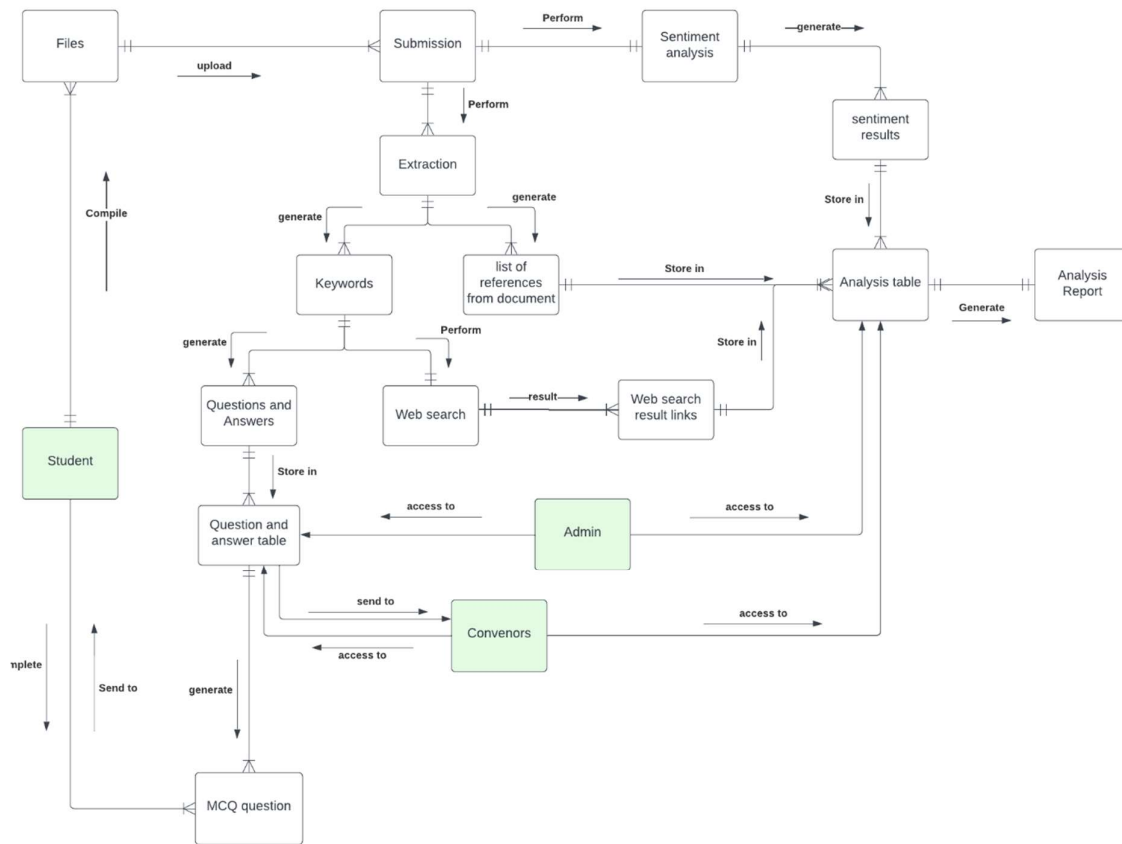


Table 1: Document Submission System – Domain Model

The above domain model diagram outlines the domain model of the Document submission system. It interprets the relationships of each domain entity involved in this document submission system. There are three main actors in this system: Students, Admin, and Convenors. Below are the responsibilities for each role:

#### Student

- Responsible for compiling files and submitting files to document submission system.
- Responsible to answer generated MCQ answer that is available on webpage after their submission.

#### Admin

- Responsible for managing all the data, including auto generated questions and submitted document analysis status.
- Able to control the question quality and type from question bank to student.
- Able to control the student multiple choice question test result for sending to correct convenor.
- Able to retrieve any data from cloud database for further analysis purposes.

**Convenors**

- Responsible to receive student multiple choice question test result report via email.
- Responsible to determine student's plagiarism based on the provided information from Document Submission System. (Listed under section 2.1)

Students will have to compile their works in PDF format at the initial stage. Then, they can submit it to the submission system provided with their email. The submission system will receive the document and perform two different analysis processes.

For the first analysis process, the whole submission will go through the sentiment analysis using Google Cloud NLP. This process will perform deep and complete research on each paragraph submitted. Every outcome of it including writing style and paragraph emotion will be stored in an analysis table for the report generating processes.

For the second analysis process, the system will perform an extraction of keywords from the document such as title, introduction, extract, and conclusion. The extracted sentences or keywords will go through two sub-processes which is generate question and perform web search.

The Generate Question Process will generate questions and answers by keywords given from student submission. The MCQ question will be given to submitter for testing integrity. All the questions and answers will be stored in the question and answer table for both admin and convener to review.

When the student has finished and submitted the question provided by the system, the result will be graded and stored in the question and answer tables as documentation. All the information, including MCQ questions, student testing results, text analysis results and sentiment analysis results, will be compiled and generated into an analysis report document.

The Web Search Process will perform a web search by using extracted keywords from student submission. The top five result of website links will be recorded and store in analysis table for generate the analysis report.

The system will send only the student's MCQ test result to respective convenors for plagiarism evaluation. The convenor can login to document submission system and have a complete view of student document analysis report.

### 5.1.1 High-level system architecture

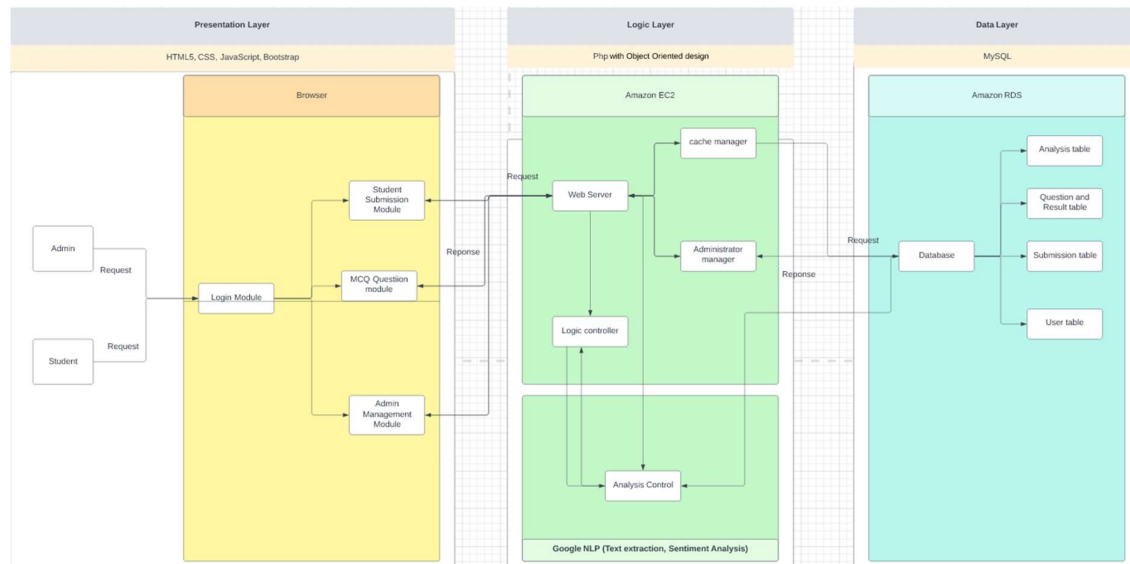


Table 2: Document Submission System – High-Level System Tier Architecture

The above diagram is the High-Level System Tier Architecture of the Document Submission System. It is divided into three tiers: **Presentation Tier**, **Logic Tier**, and **Data Tier**.

The presentation Tier will include all the user interfaces and the interaction between the end-user and the system. The logic layer, which is run under the Amazon EC2 service with external Google NLP service, will be responsible for collecting all the information sent from the user browser, computing it, and requesting storing data in Data Layer. The Data Layer will run under the Amazon RDS service, responsible for storing and managing data in the database.

End-user will have to use a web browser to enter the document submission web application. The first graphic user interface is the login module. Users need to log in with their credentials. If the credential is student, the system will direct to student submission page for submitter to submit PDF document. After submitting the document, student will be redirected to MCQ question module to complete the MCQ testing. This module is for students to get tested on their integrity of work. The Analysis Management Module is the page that generates the student's submissions analysis report.

In the logic layer, the webserver will be the agent to receive or respond to any HTTP request message from the user. The logic controller is a controller that will help the system to control the calculation of the data, such as the extraction of sentences from the paragraph. Cache Manager will act as a buffer to store temporary data for validation before going into the database. The administrator manager will be the sentinel to control all the verification and validation of user credentials. Analysis control will take charge of all analysis parts, such as text analysis, and will be linked to external tools Google Cloud NLP for analysis purposes.

As for the data layer, four tables will be stored in AWS RDS: Question and Result table, Analysis table, Submission table and User table. The Question and Result table will keep all the generated questions for the student as well as students MCQ results. The analysis table will contain all analysis results from student-submitted documents, such as writing style, and the use of emotion. Submission table will store the student's submission PDF document. The user table will store all the end-user and admin credentials.

## 6 Interface Requirements

The document submission system will have four entities: user interfaces, hardware interfaces, software interfaces, and communication interfaces. It will follow the high-level system architecture to communicate with other entities.

Hardware interfaces will be one of the critical entities in this system. The user must have an internet connection to use the document submission system. Hardware interfaces will connect to the software interfaces entity as all the web applications must have a server to operate. This project will use virtual servers such as AWS RDS and AWS EC2 to host our data and web server.

Next, the software interface entity will have a connection communication interface entity. The communication entity will help us set up all communication protocols such as HTTP protocol and FTP protocol for file transferring.

The Communication interfaces entity will have a link to the user interfaces entity. The user interfaces entity will be responsible for the graphic user interface, and it is the only entity that have direct interaction with the user.

### 6.1 User Interfaces

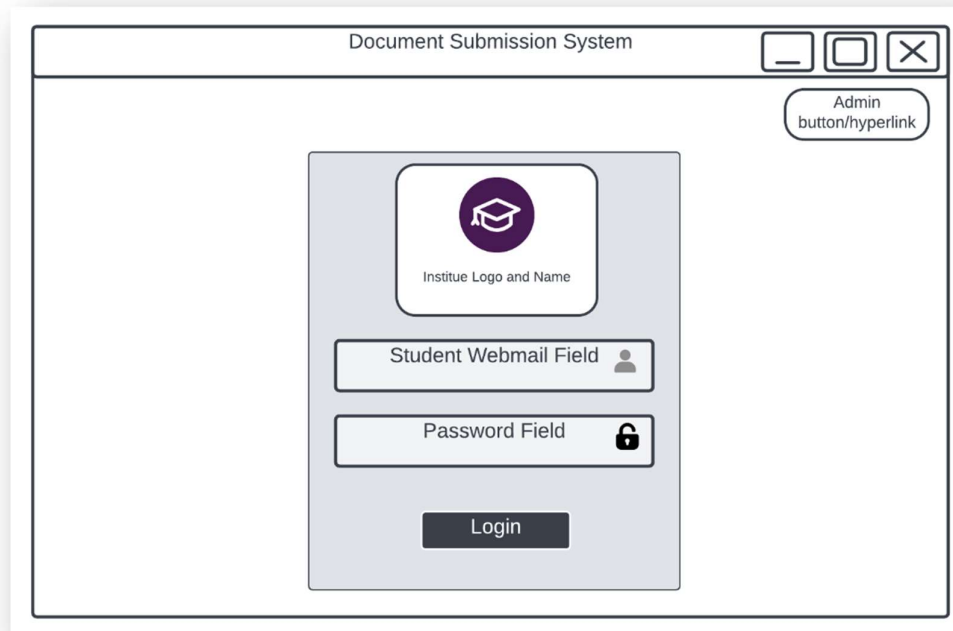
The User Interfaces of our web application is designed to be user-friendly and easy to understand. The design practices applied when designing the interfaces include:

- Keep interface simple and clear in language
- Maintain consistency and use common UI elements
- Focus on page layout

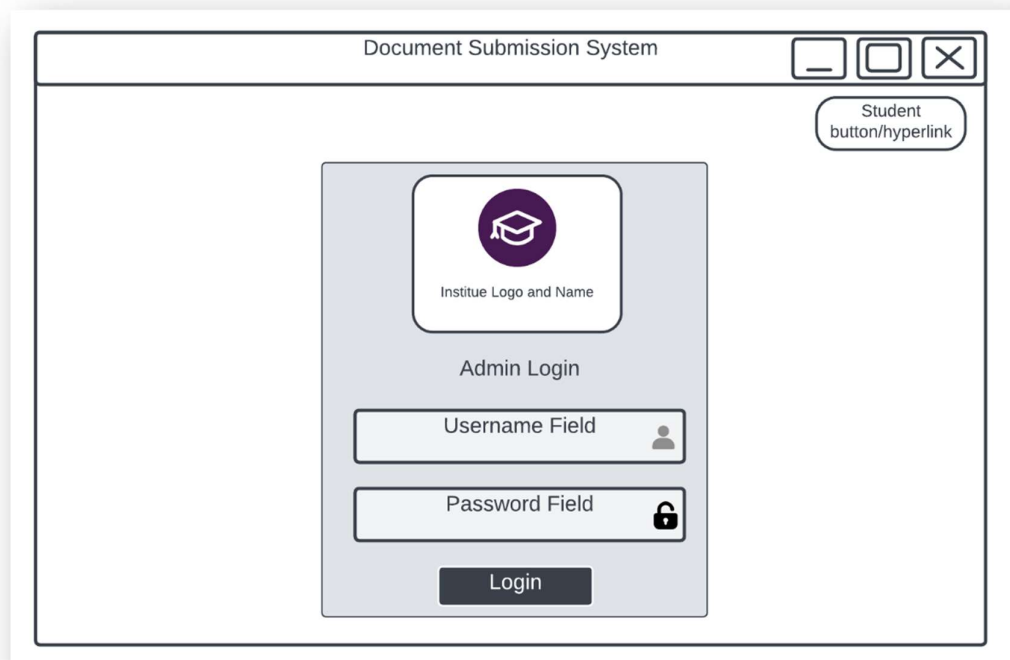
The user interfaces for the software should be compatible to any browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, Opera and Internet Explorer by which users can gain access to the system with any of the browser stated above. English language is the primary language in our application as it is a universal language which can be understand by everyone. Moreover, users are expected to have fundamental digital literacies such as browsing, reading information and using proper buttons.

User interfaces are further illustrated in detail below:

### 6.1.1 Login Interface



Login Interface (Student) will have **two input fields** placed at the center of the browser which will prompt the user to enter student webmail and password. There will be a **Login button** to login user. There is also a **hyperlink/button** which navigates user to admin login at the top right corner.





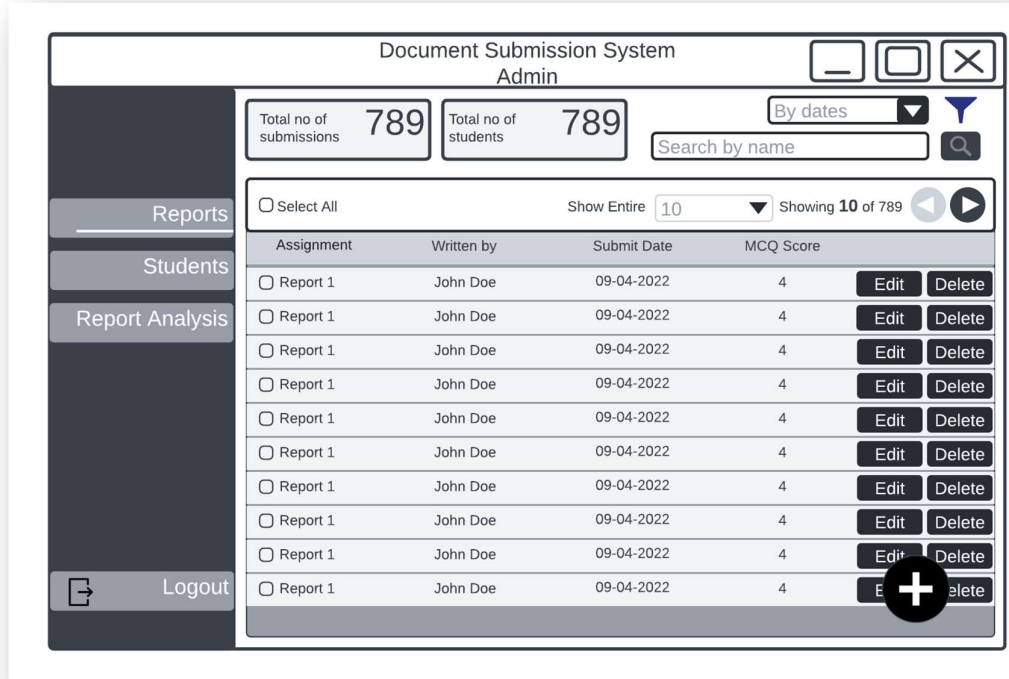
Login Interface (Admin/Convenor) will be the same as Student Login. It will have **two input fields** placed at the center of the browser which prompt the admin to username and password. There will be a **Login button** to login admin. There is also a **hyperlink/button** which navigates admin to student login located at the top right corner.

### 6.1.2 Assignment Submission Interface

The mockup shows a web browser window titled "Document Submission System Student". The header bar is dark blue and contains a user profile icon, the text "Hello, Adrian", and the date "21/04/2022". The main content area is divided into three sections. On the left is a sidebar with a "Submission" button (with a document icon) and a "Logout" button (with a door icon). The central area features a large "Upload" button with a cloud and arrow icon, an "Add Convenor" text input field, and an "Accept Terms and Conditions" checkbox. On the right, the "Uploaded Assignment" section shows "Assignment 1" with a "100%" progress bar and a "Preview File" button with a close icon. At the bottom right are "Submit" and "Cancel" buttons.

Assignment Submission Interface will have a **big input box** which prompts the user to upload his/her assignment. During submission, user can add comments to the assignment by inputting them to the **comment field** under the big box. A **check box** is located below the comment field where student must check the box to accept the terms and conditions. Uploaded assignments will be displayed under the **right panel**. A **cross button** is placed right next to the uploaded assignment for the user to delete the wrong assignment. There are two buttons located at the bottom right of the page which are Submit and Cancel buttons. **Submit button** submits the assignment while **Cancel button** calls off the submission. There is a **Logout button** located at the top left corner.

### 6.1.3 Admin Interface



The admin interface will be used to manage data in the web application. There will be a **navigation pane** on the left to navigate users to switch to other data table. On the top, there are **two big cards** which display and summarize the total number of submissions and students. A **select option field** is provided at top right to filter data in different ways followed by a search bar which prompts admin to enter keywords for searching assignment. There is also a **check box** on top of the table to enable admin to select all records in one time. Another select option field is provided for admin to adjust the range of the displayed entries. **Pagination** is designed above the table as a simple navigation method that splits an abundant amount of data within the tables into smaller parts. Each **record** will be display in a table form with relevant information. Each record has two buttons which are Edit and Delete buttons. **Edit** button allows admin to make changes on each record while **Delete** button removes the record from the database. A **floating button** is placed at bottom right corner to enable admin to insert or add new record to the table. **Logout** button is located at the bottom of the navigation pane.

## 6.2 Hardware Interfaces

Since the document submission system is of a web application which must run over the internet, all the hardware should have the functionality of establishing a stable connection to the internet either a wired connection or wireless connection. Some examples of hardware that the software will interface to include WIFI Modem, WAN – LAN and Ethernet Cross-Cable.

## 6.3 Software Interfaces

The document submission system includes a web application developed with **HTML, CSS, JavaScript and Bootstrap**. The web browser will communicate with the **Amazon EC2** – the cloud-based server used in this software to operate and manage the requests. The document submission system uses the **Amazon Relational Database Service** to store and retrieve data about the users and the questions. System will communicate with **Text Analysis tools** like Google NLP, IBM Watson and MonkeyLearn.

## 6.4 Communication Interfaces

When communicating over the internet and servers the document submission will use the HTTP protocols. The communication architecture of the system must follow the **client-server** model. The communication established between client and server should use a **REST**-compliant web service. The protocol used in the REST API for message transmission should be **HTTP Secure** (HTTPS). The communication between client tier and server tier must be **stateless**.

## **7 References**