# FORMAT CHECKING OF FYP REPORTS (WORD FILES)


**Muhammad Ali Tahir**

**Farhan Bashir**

**Syed M. Sami Hassan**

*Supervised By*

**Sir Zain ul Abideen**

*Submitted for the partial fulfilment of BS Computer Science degree to the*

*Faculty of Engineering & CS*

**DEPARTMENT OF COMPUTER SCIENCE**

**NATIONALUNIVERSITY OF MODERN LANGUAGES ISLAMABAD**

**June, 2023**

# ABSTRACT

This project is a web application which takes a FYP report file in word format as an input and compares it with the already defined actual format of the “National University of Modern Languages FYP report” and highlights the mistakes in the uploaded file. Logging file is also generated which states the validation result of each and every validation such as heading’s size and font, logos, chapter and paragraph headings, minimum word counts, table of contents and figures etc. The main purpose of the project is to save the time of both, the students, and the supervision panel as they won’t have to check the report manually. It is surely effective as there is no such system introduced before and efficient as the mistakes are marked out after reading the file.

User interacts with the web application, where he/she enters their names, email and upload their FYP report and after validation they get the autogenerated email (SMTP Services are used to do so) which contains Validated log text file (which contains each and every validation) and Final Year Project Report's file containing “Red Highlighted Errors”. Student and Admin can also download them directly from portal. Spire.doc library is used which extracts the element from the word file and compares them with the already defined format which are stored in the form of tables in the database. Asp.net is used for the frontend development, where there is a portal for Students and also for Administration.

There are some limitations of the project which include the issues due to use of Spire.doc library for document processing cannot validate auto-generated table of contents, tables in the chapters and images in a file which are placed randomly in different chapters. We tried to cope with these limitations as much as possible and came up with alternate solutions such as validating captions of images and tables from the table of content with content in chapters rather than checking images and tables directly, also to handle data in table of content we applied logic for manual table of content rather than checking auto-generated table of content because of limitations.

Future work could involve implementing additional validation mechanisms to address these limitations. This could include developing custom algorithms to verify the structure and accuracy of table of contents and images. Our application is validating first 7 pages from database and other dynamically from table of content, future work could be to developer whole systems dynamically where Admin just has to enter the FYP Sample Report with desired format and our application automatically validates on the basis of format provided rather than using database.

# CERTFICATE

Dated: _____

## Final Approval

It is certified that the project report titled **'Format Checking of FYP Reports (Word files)'** submitted by **Muhammad Ali Tahir**, **Farhan Bashir** and **Syed M. Sami Hassan** for the partial fulfilment of the requirement of **"Bachelor's Degree in Computer Science"** is approved.

## COMMITTEE

**Dr. Noman Malik**

Dean Engineering & CS: Signature: _____

**Dr. Sajjad Haider**

HoD Computer Science: Signature: _____

**Mr. Farhad Muhammad Riaz**

Head Project Committee: Signature: _____

**Mr. Zain ul Abideen**

Supervisor: Signature: _____

# DECLARATION

We here by declare that our dissertation is entirely our work and genuine/original. We understand that in case of discovery of any PLAGIARISM at any stage, our group will be assigned an F(FAIL) grade and it may result in withdrawal of our bachelor's degree.

Group members:

| **Name** | **Signature** |
|---|---|
| Muhammad Ali Tahir | _____ |
| Farhan Bashir | _____ |
| Syed M. Sami Hassan | _____ |

# ACKNOWLEDGEMENT

We are thankful to Allah Almighty for giving us knowledge for completing this project and we are also highly thankful to all the people who pray for our success specially our friends. We would like to acknowledge and give our warmest thanks to our supervisor Mr. Zain ul Abideen who made this work possible and is guidance and advice carried us through all the stages of our project. We wouldn't have been able to finish the work before the time without his assistance.

Moreover, we would like to thank Mr. Farhad and Dr. Qurat ul Ain Safdar for their invaluable guidance and support throughout the development of this project. Their expertise and assistance were instrumental in shaping our ideas and ensuring that we stayed on track. We would also like to thank our committee members for letting our defence be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you.

# TABLE OF CONTENT

**Chapter** **Page**

# LIST OF FIGURES

# LIST OF TABLES

# Chapter 1

# INTRODUCTION

## 1.1. Introduction:

This web application is helping the users to check the format of their FYP reports. It helps them to check whether their report is according to the defined format provided by the university or not. Before, the students had to provide a hard copy of the report and the panel had to check the FYP reports manually for the mistakes. But this web application helps save time for both the panel and the students.

## 1.2. Project Domain:

The project focuses on the domain of education, specifically on the validation and formatting of Final Year Project (FYP) reports for the "National University of Modern Languages." It aims to automate the process of checking and highlighting formatting errors in FYP reports by comparing them with the predefined format of the university's report. By providing a web application for students to upload their reports and generating validation logs and automated emails, the project streamlines and saves time in the manual validation process. The administration portal allows supervisors to search and verify the validation status of student reports using provided tokens. The project domain encompasses aspects of document management, report formatting, and streamlining the assessment process within an educational institution.

## 1.3. Problem Identification:

As we are aware that every FYP report is consisted of 6 chapters and is checked manually by the panel which is very time consuming as they have to check the format of the entire report according to the format provided by the university. Rather than consuming time on the actual content, most of the time is spent on checking the content formatting. The panel have to tell again and again about the mistakes identified in the report.

The existing manual validation process of Final Year Project (FYP) reports at the "National University of Modern Languages" is time-consuming and prone to errors. Students and supervision panels face challenges in ensuring adherence to the predefined format, resulting in inconsistencies and limited feedback. The lack of automation leads to inefficiencies and difficulties in tracking the validation status of reports. These problems hinder the timely and accurate assessment of FYP reports, causing inconvenience and delays for both students and supervisors.

### 1.3.1. Proposed Solution:

To address these issues, a web application is proposed to automate the validation process and highlight formatting errors in FYP reports. By comparing reports with the predefined format, the system saves time and improves accuracy. It generates detailed validation logs and autogenerates emails with highlighted errors, providing valuable feedback to students. The administration portal allows for easy tracking of validation status using unique tokens. With this solution, the project aims to streamline the validation process, ensure consistency, and enhance the overall efficiency of FYP report assessments at the university.

### 1.3.2. Objectives:

Through our system, there is no need for checking format of FYP report manually. Students just upload their FYP reports on a web portal and if there is any error in report formatting the updated result is sent to the user where all of the mistakes are highlighted. So, teachers time would not waste, and Student are informed automatically without interference of the teacher itself. The main objectives are:

- To design a system that saves teachers time by checking the format by itself.
- To design a system that informs the students automatically by sending them an email of validation results.

### 1.3.3. Project's Scope:

The project's scope encompasses the validation of Final Year Project (FYP) reports in Word format for the "National University of Modern Languages." It focuses on validating formatting elements such as headings, logos, and word counts. The system generates a logging file to provide detailed feedback on errors found, and an autogenerated email is sent to students containing the validated log and the report file with highlighted errors. The administration portal enables supervisors to track and verify the validation status using unique tokens. Integration with the Spire.doc library and ASP.NET frontend development ensures seamless extraction and comparison of report elements.

The project's scope excludes in-depth content analysis and focuses solely on formatting validation. It does not involve modifying the predefined format but aims to streamline the manual validation process by automating error identification. The project provides a user-friendly interface for students to upload their reports and receive feedback, as well as an administration portal for efficient oversight. The project's scope is limited to FYP reports,

Word document input, and the predefined format of the "National University of Modern Languages."

## 1.4. Effectiveness / Usefulness of the System:

The project proves effective by automating the validation of FYP reports, saving time for students and supervision panels. It accurately detects formatting errors, provides feedback for improvement, and ensures consistency in the assessment process. The administration portal enhances oversight, while the user-friendly interface improves the overall experience. With scalability and accessibility, the project streamlines the validation process, benefiting all stakeholders involved in the FYP report evaluation.

## 1.5. Resource Requirement:

The project requires a variety of resources, such as a server or hosting infrastructure for the web application deployment, database storage for predefined format tables and validation logs, SMTP services for sending automatically generated emails, and integration of the Spire.doc library for extracting and comparing Word file elements. For creating the user interfaces, the ASP.NET framework and frontend programming know-how are additionally required. To ensure effective performance and the management of many user interactions, sufficient computing resources, including processing power and storage, must be allotted.

### 1.5.1. Hardware Requirement:

- **Computer:** A computer or server system capable of running the web application and hosting the necessary software components. This can be a desktop computer or a laptop
- **Storage Device:** Sufficient storage capacity to store the web application files, database, and uploaded FYP report files. Drives with enough space to accommodate the expected volume of files such as Hard drive or solid-state drive (SSD)
- **Memory (RAM):** Adequate memory to support the concurrent execution of the web application, database, and other necessary software components.
- **Network Infrastructure**: A stable network connection with sufficient bandwidth to handle user interactions, file uploads, and email notifications. This can be a wired or wireless network connection depending on the deployment environment.

### 1.5.2. Software Requirement:

- An **operating system** intended for servers, such as Linux or Windows Server

- Internet Information Services (IIS) for Windows, a **web server programme**
- Maintaining and archiving the validation log, default format table, and other pertinent data in a **database management system (DBMS)**, such as Microsoft SQL Server, My SQL or PostgreSQL
- As it is stated in the project specifications, the **frontend development** uses the ASP.NET framework.
- Using the **Spire.doc library** to extract components from Word documents and compare them to a predetermined format.
- The FYP report file with highlighted faults and the validated log text file are sent to users using **SMTP** services.
- The **development tools** like **Visual Studio** and **Microsoft SSMS** are used for integration purposes.

### 1.5.3. Data Requirement:

- **FYP Report Files:** The project requires the FYP report files in Word format as the main input data. These files are be uploaded by students for validation.
- **Predefined Format Tables:** The project needs predefined format tables that define the expected formatting requirements for FYP reports. These tables are stored in a database and are used for comparison with the uploaded reports.
- User Information: User information, including student names and email addresses, are captured during the interaction with the web application for generating autogenerated emails and facilitating communication.

## 1.6. Report Organization:

There are six chapters in the report. The project's introduction is covered in length in the first chapter, along with the project's scope, efficiency, and resource needs. The background and current system information is covered in the second part of this study.

System specifications and needs are covered in detail in the third chapter. The fourth chapter discusses system modelling and design, which includes an overview of the system's architecture, use cases, data flow, and system diagram. The testing and validation of systems are the focus of the fifth chapter. This chapter discusses system testing in addition to the necessary results. The final chapter of the project concludes by summarizing its overall purposes, prospective applications, and objectives.

# Chapter 2

# BACKGROUND AND EXISTING SYSTEMS

This chapter is dealing with the literature review of the projects similar to ours. There is no need for checking format of FYP report manually. Students just upload their FYP reports on a web portal and if there is any error in report formatting the updated result is sent to the user where all of the mistakes are highlighted.

## 2.1. Related Literature Review:

In this section, we have discussed the literatures that are similar to our system. The overview of these literatures is given below:

### 2.1.1 PubLayNet: Largest Dataset Ever for Document Layout Analysis:

In addition to its large-scale dataset, PubLayNet introduces a robust methodology for generating accurate ground truth annotations. It addresses the scarcity of annotated datasets in this field by combining two existing datasets and augmenting them with a novel large-scale dataset. PubLayNet contains over one million document images from a diverse range of sources and covers various layout elements. This research paper not only presents a ground-breaking dataset but also provides a valuable resource for advancing document layout analysis techniques through the utilization of state-of-the-art deep learning models.

### 2.1.2. Validating process variables of sourcing in an assessment of multiple document comprehension:

The research paper focuses on validating process variables related to sourcing in the assessment of multiple document comprehension. It explores the influence of various factors on participants' ability to comprehend and integrate information from multiple documents. The study employs a mixed-methods approach, combining qualitative and quantitative data analysis. The findings highlight the significance of variables such as source credibility, source diversity, and source conflict in determining comprehension outcomes. The paper contributes to a deeper understanding of the cognitive processes involved in sourcing and provides insights for designing effective assessments of multiple document comprehension.

### 2.1.3. Summary of Revied Literature:

Table 2.1: Summary of Reviewed Literature

| Year | Authors | Contribution | Techniques | Limitations |
|---|---|---|---|---|
| 2019 | Wenhao He, Kaiyue Pang, Tao Huang, and Jimmy Lin. | Automatic annotation of document layout. | Data Combination and Augmentation, Deep Learning Models, Ground Truth Generation and Evaluation Metrices | PubLayNet does not contain relationships between the layout elements, e.g., a paragraph and a section title |
| 2018 | Bråten, Stadtler, & Salmerón | To demonstrate how process data from an assessment of multiple document comprehension can be used to represent sourcing | Generalized linear mixed models (GLMM) | Although the MDC assessment was designed to record source access, the operationalized process variables do not claim exhaustiveness in the identification of possible cognitive functions of sourcing. |

## 2.2. Related Systems/Applications:

There are some similar existing systems in the market. The overview of some of the systems is given below.

### 2.2.1. Scribendi:

Scribendi is an online editing and proofreading service that offers professional assistance for a wide range of written documents. With a team of experienced editors, Scribendi helps improve grammar, punctuation, clarity, and overall quality. Users

submit their documents to the platform for editing, ensuring error-free and well-written content. Scribendi serves students, authors, businesses, and individuals who seek to enhance the effectiveness of their writing. It provides different service levels, including proofreading, editing, and formatting, while focusing on editing existing content rather than generating new material.

### 2.2.2. Spire.office:

Spire.Office is a software component suite by e-iceblue, offering .NET components for office document handling. It enables developers to create, modify, convert, and extract data from office files programmatically. With features like document generation, manipulation, conversion, and formatting, it simplifies the automation and processing of Word, Excel, PowerPoint, and PDF documents within .NET applications. Spire.Office provides a comprehensive set of tools for dynamic document creation, content manipulation, format conversion, and data extraction, streamlining office document handling tasks for developers.

### 2.2.3. Summary of Existing Systems:

Table 2.2: Summary of Existing Systems

| Year | System | Contribution | Limitations |
|---|---|---|---|
| 1997 | Scribendi | Online editing and proofreading | Subject Matter Expertise, Dependency on user input, Limited colloaboration |
| 2014 | Spire.office | Document generation, manipulation, conversion, and formatting | Platform dependency, learning curve, File format capability |

## 2.3. Identified problem from existing work:

The lack of relationship representation between layout components in the PubLayNet dataset, the incomplete capture of cognitive functions related to sourcing in multiple document

comprehension assessment, the limitations in user collaboration and expertise dependency in Scribendi, and platform dependence and potential format limitations in Spire.Office are among the issues that have been identified.

## 2.4. Selected Boundary for Proposed Solution:

With our proposed solution, we implemented some functionalities to address the needs of the project. Firstly, the system validates the FYP report file by comparing it with the predefined format of the "National University of Modern Languages FYP report." This helps highlighting any formatting mistakes such as incorrect heading sizes, fonts, logos, chapter and paragraph headings, minimum word counts, and table of contents. Additionally, the system generates a logging file (Validated Text File) that provides a comprehensive validation result for each aspect of the report. Users are able to upload their FYP report, receive an autogenerated email with the validated log text file, and download the report file containing red-highlighted errors.

However, some capabilities could not be implemented in our system owing to restrictions or reasons. For example, the system does not validate the auto-generated table of contents and images in the report. This is due to a limitation with the Spire.doc library, which does not support the validation of these specific elements. While we strive to provide comprehensive validation, this particular aspect requires manual checking by the users or the supervision panel. Nonetheless, the system still offers the significant time-saving benefits by automating the validation process for various other formatting aspects of the FYP report.

# Chapter 3

# SYSTEM REQUIREMENTS AND SPECIFICATION

We outlined the non-functional requirements for the developed system in this chapter together with the module/software requirements.

## 3.1. System Specifications:

Our system is based on a web application that'll help the administration and the students to identify the mistakes in the inserted FYP report. It saves the time of both the administration and the students as they won't have to check the format of the file manually.

The frontend of the application is built on **asp .net**. Backend of the application is built using **.net framework** with **Spire doc** package. The system is using code first approach.

## 3.2. System Module:

Based on the functionality and interactions between users described in the project abstract, the requirements for the interface for the project can be defined. For out project, the following modules are necessary:

### 3.2.1. User Registration/Login:

Create a user registration and login interface for administrators and students. To access the system, users should be able to either create accounts or log in with their credentials.

### 3.2.2. File Upload:

Provide a user-friendly interface for users (administrators and students) to upload FYP report files in Word format. The user interface requires to enable file selection and offer confirmation after a successful file upload.

### 3.2.3. User Information Form:

Users may provide their email addresses and names on a form by presenting it. This data is used to generate an email that is sent to the user with the results of the validation.

### 3.2.4. Downloading Validated Files:

Provide options for downloading for the verified FYP report file with problems noted and the validated log text file for users. The online application should allow users to download these files directly.

### 3.2.5. Email Notification:

Automatically send an email to the user that contains the FYP report file with errors highlighted in it and the validated log text file. Send the email to the user-provided email address using SMTP services.

### 3.2.6. Dashboard and Navigation:

Create a simple dashboard or navigation layout that gives users access to the application's numerous features, such as file uploading, validation history, and the setting of user profile.

## 3.3. Functional Requirements:

Here are some functional requirements for the project:

### 3.3.1. User Registration and Authentication:

- Allow the users (students and administration) to register and create accounts.
- Provide a login mechanism to authenticate users and ensure secure access to the system.

### 3.3.2. File Upload and Validation:

- Enable users to upload their FYP report in Word Format
- Extract the content from the uploaded file using the Spire.doc library.
- Compare the extracted content with the predefined format of the "National University of Modern Languages FYP report."
- Validate various elements such as headings, font styles, logos, chapter and paragraph headings, word counts, table of contents, and figures.
- Highlight formatting errors in the uploaded FYP report file.

### 3.3.3. Logging and Reporting:

- Generate a logging file that records the validation results for each validation rule or requirement.
- Include details such as validation success, errors found, and specific locations of formatting errors within the FYP report.
- Store the logging file in a text format or a structured format in the database.

### 3.3.4. Email Notification:

- Automatically generate an email to the user (student) after the validation process.

- Include the validated log text file as an attachment in the email.
- Attach the FYP report file with highlighted errors for the user to download.

### 3.3.5. Database Management:

- Store user registration information, uploaded files, validation results, and tokens in a database.
- Ensure proper data management, including retrieval, storage, and security.

### 3.3.6. Error Handling and Validation Feedback:

- Display clear and informative error messages if there are issues with file upload or validation.
- Provide detailed feedback on the specific formatting errors found in the FYP report.

### 3.3.7. File Download:

- Allow users (students and administration) to download the validated FYP report file with highlighted errors.
- Enable direct file downloads from the web application.

## 3.4. Non-Functional Requirements

Although this application can function without certain prerequisites, it might not live up to expectations. The following conditions are not functioning for our application:

### 3.4.1. Usability:

A key aspect in making sure the project is user-friendly is its usability. The user interface of the web application shall be simple and intuitive, allowing users to navigate, upload their FYP reports, and submit their information with ease. The process of validation is directed by clear instructions the entire time. Users surely find it helpful to review their errors thanks to the automatically generated email that contains the validated log text file and the FYP report with highlighted errors. The project's overall goal is to maximize usability by reducing the time and effort users must expend, expediting the validation procedure, and offering a smooth and effective experience for both students and the administration.

### 3.4.2. Reliability:

The reliability of the project is a top priority, ensuring consistent and accurate results. This system is designed to handle various scenarios and inputs, effectively validating FYP reports against the predefined format. There is thorough testing to find and fix any potential mistakes or problems. Additionally, the project incorporates error handling mechanisms, data backups, and system monitoring to minimize downtime and ensure the application remains stable and reliable for users throughout their interaction with the system.

### 3.4.3. Performance:

In order to ensure an effective and responsive user experience, the project requires to attain optimal performance. Large FYP reports is processed more quickly thanks to the validation process's speedy and precise execution. The system is ready to deal with numerous concurrent user interactions, ensuring lag-free performance even during busy times. In order to find and fix any potential bottlenecks, performance testing and optimization strategies are utilized, which produces a high-performing system that fulfils user expectations.

### 3.4.4. Scalability:

To satisfy evolving demands from users, the project is built with scalability in mind. The system design is scalable horizontally as well as vertically as required. It is feasible to handle increasing user loads and bigger data volumes thanks to the database and server infrastructure. The application also adjusts to make the best use of system resources and reduce bottlenecks, guaranteeing smooth performance even as the user base and data size grow.

### 3.4.5. Availability:

The project intends to achieve high availability, making sure that users can access the web application whenever they need it. The application is installed on a dependable hosting infrastructure, redundancy is added to crucial components, and automatic monitoring systems is put up, among other steps, to reduce downtime. The right alerts and schedule is used in the event of maintenance or upgrades to prevent any adverse impacts on user accessibility.

# Chapter 4

# SYSTEM MODELING AND DESIGN

This chapter is going over the system model and design. We deal with how the architectural diagram, sequence diagram, data diagram and use case diagram helps us demonstrate the project and how the students and admin can communicate with the system is shown in the diagrams.

## 4.1. System Design and Analysis:

The system design and analysis of the project involve analyzing the requirements, designing the system architecture, implementing validation logic, and developing the user interface. The requirements analysis helps define the project's scope and objectives, while the system architecture design determines the components, modules, and interactions. Throughout the process, risk assessment, feasibility studies, and scalability considerations are performed to ensure a viable and efficient solution.

## 4.2. System Diagram:

Following diagram is the overview of whole system where there is a web application which can be accessed by students or admin. Some functionalities are same for both admin and student which include login, signup, dashboard and profile. While some are specified for students such as upload a FYP Report, send messages, manage his profile and finally view his validation record i.e., he can view the validation record which has been generated through his account and admin can view messages which are sent by student, view all of the student validation record and manages his profile. All of the data is managed with the help of database which fetches and stores record such as validated text and word files, validation history and email is sent with these files being attached to it.

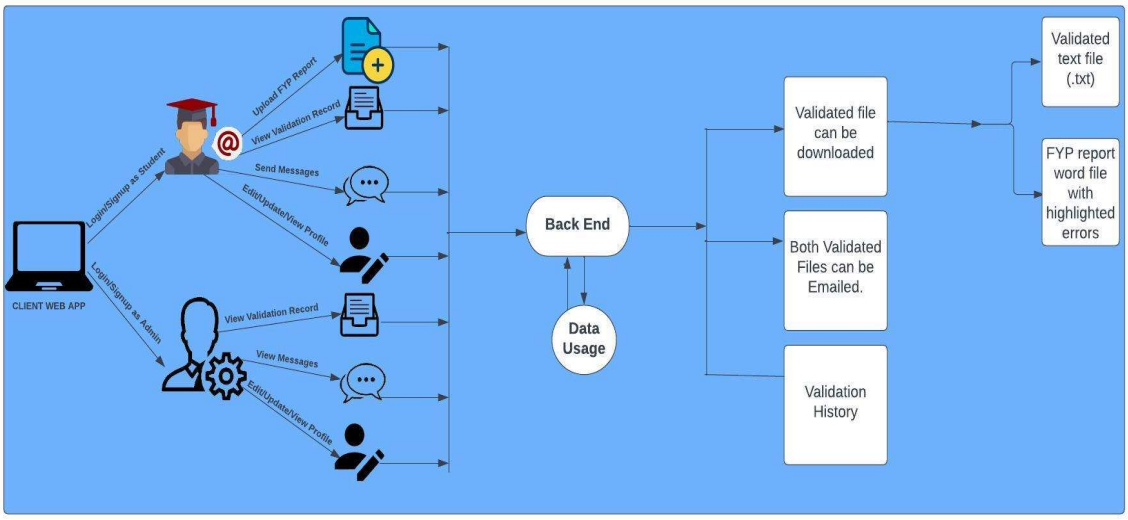


Figure 4.1: System Diagram

## 4.3. Use Case Diagram:

Following use case diagram illustrates the interactions and functionalities of a system involving students and administration. It encompasses use cases such as registration, login, uploading FYP reports, format checking, generating validated files, generating emails, accessing the dashboard, managing profiles, viewing validation records, contacting administrators, and viewing messages. Students can register and log in to upload their FYP reports, which are then checked for formatting errors. Validated files are generated, and an email is sent to students. The dashboard serves as the central hub, allowing users to manage profiles, view validation records, contact administrators, and access messages. These use cases ensure effective communication and efficient system utilization.

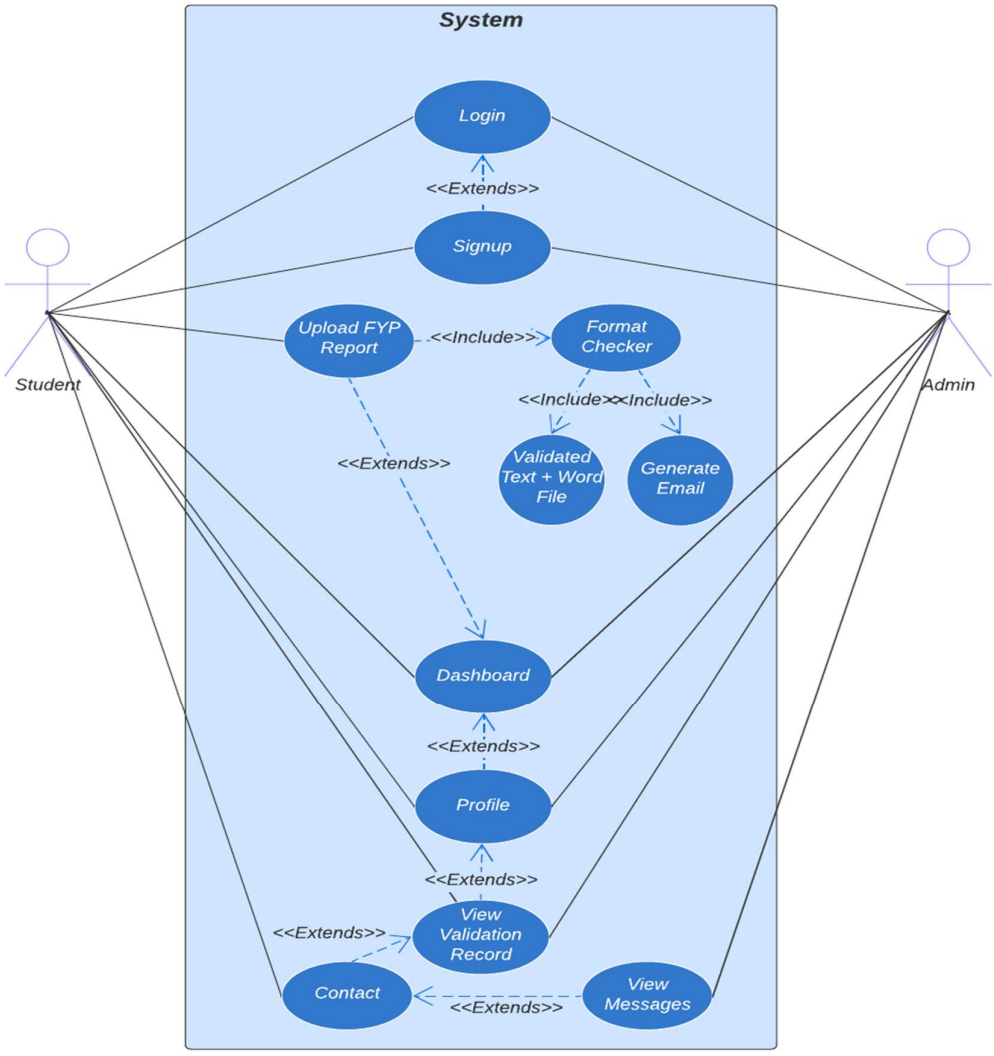


Figure 4.2: Use Case Diagram

## 4.4. Full Dress Use Case/Detailed Use Case:

Use Case 1: Signup

Use Case 2: Login

Use Case 3: Upload FYP Report

Use Case 4: Format Checker

Use Case 5: Generate Validated Text and Word file

Use Case 6: Generate Email

Use Case 7: Dashboard

Use Case 8: Profile

Use Case 9: View Validation Record

Use Case 10: Contact

Use Case 11: View Messages

**Actors:**

- Student

- Administration

**Use Cases:**

**Use Case 1: Registration**

- Description: Allows student and admin to register and create accounts in the system, also a default profile picture is settled in the beginning which could be changed by editing in profile.

- Actors: Student / Administration

- Pre-conditions: None

- Post-conditions: Student and admin are registered and can log in to the system.

**Use Case 2: Login**

- Description: Allows student and admin to log in to the system.

- Actors: Student/ Administration

- Pre-conditions: Student/Admin must be registered.

- Post-conditions: Student/Admin is authenticated and gains access to their account.

**Use Case 3: Upload FYP Report**

- Description: Enables students to upload their FYP report file in Word format.

- Actors: Student

- Pre-conditions: Student must be logged in.

- Post-conditions: FYP report file is uploaded and ready for validation.

**Use Case 4: Format Checker**

- Description: Compares the uploaded FYP report with the predefined format of the "National University of Modern Languages FYP report" and highlights any formatting errors.

- Actors: Student

- Pre-conditions: FYP report file must be uploaded.

- Post-conditions: FYP report is validated, and formatting errors are identified.

**Use Case 5: Generate Validated Text and Word file**

- Description: Generates a validation text file that records the validation result of each validation rule or requirement and also a word file which highlights the errors from report.

- Actors: Student

- Pre-conditions: FYP report must be validated.

- Post-conditions: Validated Text and Word file are generated and stored for future reference.

**Use Case 6: Generate Email**

- Description: Automatically generates an email to the student containing the validated log text file and the FYP report file with highlighted errors.

- Actors: Student

- Pre-conditions: FYP report must be validated.

- Post-conditions: Email is sent to the student with the validation results and download links.

**Use Case 7: Dashboard**

- Description: It is the main hub of our application, from where student and admin interact with other options.

- Actors: Student/Administration

- Pre-conditions: Student/Admin must be logged in.

- Post-conditions: Students and admin redirect to multiple pages to perform specific operation.

**Use Case 8: Profile**

- Description: Enables student and admin to view, edit and update their profile where they can also upload their profile picture.

- Actors: Student/Administration

- Pre-conditions: Student/Administration must be logged in.

- Post-conditions: edited details are stored in database for future use.

**Use Case 9: View Validation Record**

- Description: Compares the uploaded FYP report with the predefined format of the "National University of Modern Languages FYP report" and highlights any formatting errors.

- Actors: Student/Administration

- Pre-conditions: Student/Administration have to be logged in and record has to be present in the database.

- Post-conditions: Validation record is fetched from database which is viewed by admin or student and they can download the validated text and word file.

**Use Case 10: Contact**

- Description: Allows student to view contact information and send messages which admin could read.

- Actors: Student

- Pre-conditions: Student must be logged in.

- Post-conditions: Message is sent to the admin and stored in the database.

**Use Case 11: View Messages**

- Description: Allows the admin to view all the messages sent by students

- Actors: Administrator

- Pre-conditions: Admin must be logged in.

- Post-conditions: All the messages are fetched from database to view.

## 4.5. Activity Diagram:

The activity diagram for the "Format Checking of FYP Reports" outlines the flow of activities involved in the validation process. It starts with the login and registration process and then stating all the activities of admin and student. The most important functionality includes student uploading their FYP report, followed by the system validating the format. If errors are detected, the system highlights them in the report. A unique token is generated for the validated report, and an autogenerated email is sent to the student with the validation result and error-highlighted report. The process concludes at the end, ensuring a systematic and efficient validation workflow.

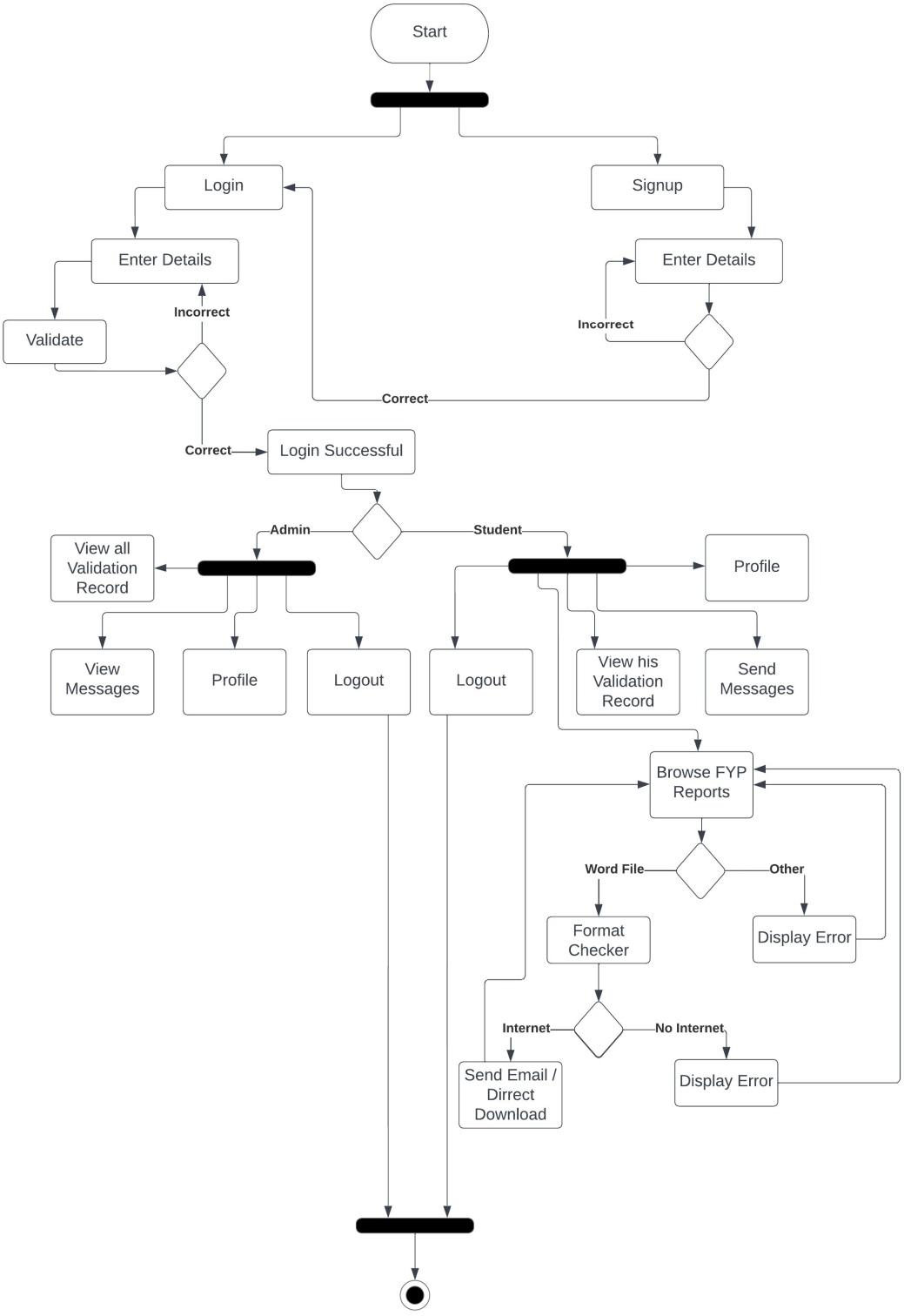


Figure 4.3: Activity Diagram

## 4.6. Dataflow Diagram:

The DFD for our system shows the flow of data between various components of the system. It depicts the input, processing, and output of data within the system. The diagram includes entities such as students, administration, along with data flows representing FYP reports, validation results, messages, profile, and email notifications. The DFD visually represents how data is transformed and shared between different components, providing an overview of the system's data flow architecture.

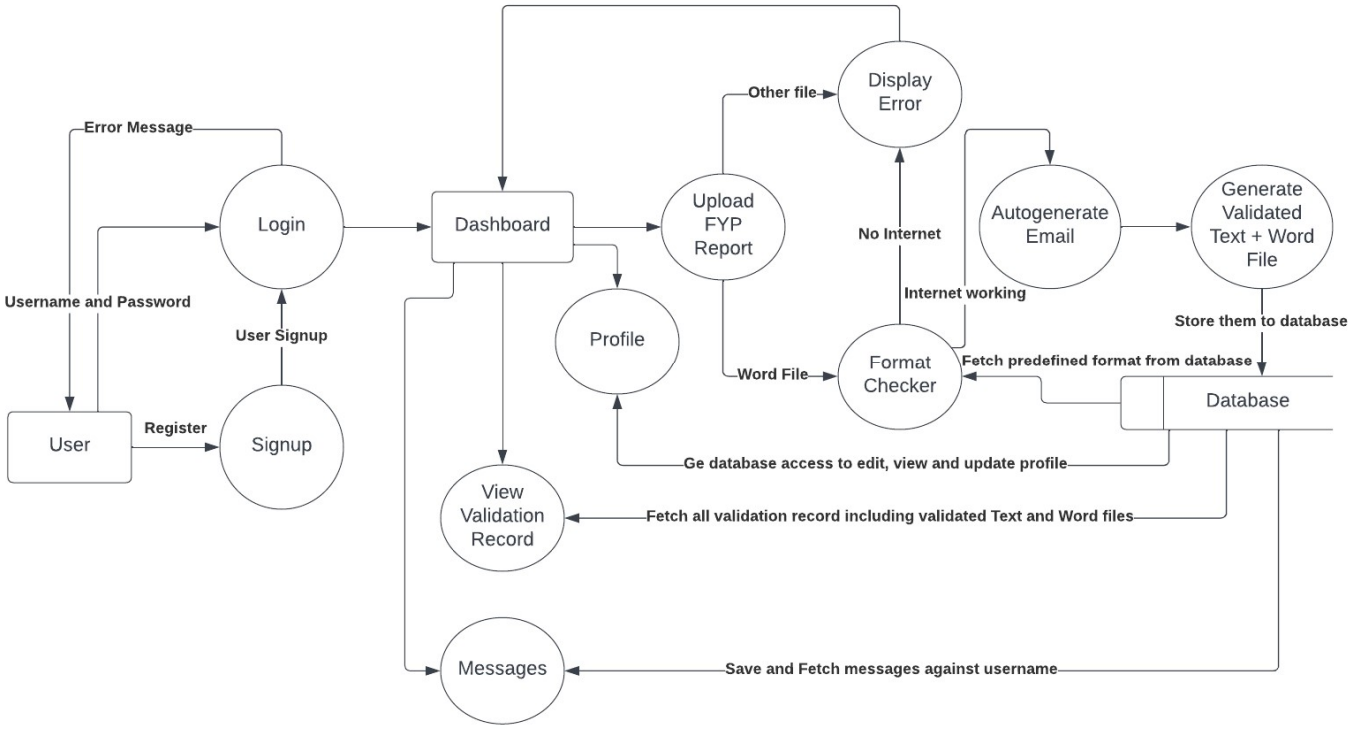


Figure 4.4: Dataflow Diagram

# Chapter 5

# SYSTEM TESTING AND VALIDATION

This Chapter discusses the functions that are basically used to test the functions and the method by which the Application passes through various tests and provides us with the desired outcome. The test case functions are described in this chapter. This chapter is organised as follows. It describes the system testing and benefits, as well as the methodologies utilized to test the functions. In this chapter, many project case scenarios are described.

## 5.1. System Testing:

System Testing has a crucial role in ensuring the quality and reliability of the "FYP Report Validation System." It involves testing the entire system as a whole to validate its functionality, performance, and usability. The testing process includes various activities such as validating the upload and parsing of FYP reports, verifying the comparison of report format with predefined standards, and checking the generation of validation results and error-highlighted reports. Additionally, system testing involves testing the email notification functionality, token generation, and searching and viewing functionalities for administration. Rigorous testing is conducted to identify and fix any issues or bugs, ensuring that the system meets the desired requirements and provides accurate validation results to users.

## 5.2. Testing Technique:

We first go over functional approaches, which enables us to test an application against its prerequisites. They are added to or made explicit in the sections and non-functional needs. There are many different kinds of testing techniques. Among these are system testing, unit testing, and integrated testing. These are a few examples of testing methods. Techniques for non-functional testing are used for non-functional requirements that describe how the system should operate. When is the data reloaded into the application? What are the speed and reaction time? How long, for instance, should a backend call take to react to a specific request? How long does the screen take to load and how quickly the back sender receives the content.

## 5.3. Test Cases:

A test case is a group of procedures or actions that are used to determine whether an application satisfies both the system's requirements and the performance operations or procedures correctly. The application needs to be completely rewritten if any of the test cases fail. the achievement of the intended results. to follow the instructions provided in order for the programme to execute effectively. Both the functional and non-functional criteria's requirements must be met by the application.

### 5.3.1. Sign-up Test Case:

Our web application helps the user, either admin or student to sign-up on the portal by providing their personal information like name, email and password. He has to enter the name carefully because that name is used in other modules like format checking, generating email and view record. In the signup default image is passed when user presses signup button as we don't need to ask for the image rather, he can upload his image when he is signed in.

Table 5.3: Sign-up Test Case

| **GENERAL INFORMATION** | | | |
|---|---|---|---|
| **Test Stage:** | ☐ Unit ■ Functionality ☐ Integration ☐ System ☐ Interface<br>☐ Performance ☐ Regression ☐ Acceptance ☐ Pilot<br>Specify the testing stage for this test case. | | |
| **Test Date:** | 4/10/2023 | **If applicable. System date:** | Null |
| **Tester:** | Muhammad Ali Tahir<br>Farhan Bashir<br>Syed M. Sami Hassan | **Test Case No:** | 01 |
| **Description** | User's sign-up to the application | | |
| **Results:** | ■ Pass ☐ Fail | **Incident Number** | null |
| **INTRODUCTION** | | | |
| **Requirement to be tested:** | The software should be accessible to the user along with browser and internet connection for this application to execute whichfalls into this category. | | |
| **Roles and Responsibilities:** | Muhammad Ali Tahir, Farhan Bashir and Syed M. Sami Hassan created an account with their relevant information inside the web application. | | |

| | |
|---|---|
| **Set Up Procedures:** | After running the project and loading the application in your browser, click the signup button. Sign up has two options. Either sign up as admin or as a student. Itprovides options for new users to create their new account. User enter confidential information such as their name, email address and password. |
| **Stop Procedures:** | Close the browser tab. |
| **ENVIRONMENTAL NEEDS** | |
| **Hardware:** | Laptop or desktop |
| **Software:** | Visual Studio, Browser |

| | |
|---|---|
| **Procedural Requirements:** | Accessing the signup form, entering valid user information, verifying successful registration, and testing error handling with invalid inputs. The process includes validation of success messages, redirection to the appropriate page, and verification of stored user data. Edge cases and special characters are also considered for thorough testing. The test results should be documented, and any issues or failures should be reported for further analysis and resolution. |
| **TEST** | |
| **Test Items and Features:** | Laptop or Desktop |
| **Input Specifications:** | For checking whether the user sign up is successful or not. |
| **Procedural Steps:** | Go to the application, open the web application, and sign up to the application. Enter the valid details and click on the sign-up button. |
| **Expected Results of Case:** | A user can easily create an account with the help of the web application both as admin as well as the students. |

### 5.3.2. Sign-In Test Case:

| ACTUAL RESULTS | |
|---|---|
| **Output Specifications:** | Users were signed up successfully. |

By providing the username and password that are being obtained from the database, the application enables the user to sign in for the application.

Table 5.4: User Sign-in Test Case

| GENERAL INFORMATION | | | |
|---|---|---|---|
| **Test Stage:** | ☐ Unit ■ Functionality ☐ Integration ☐ System ☐ Interface<br>☐ Performance ☐ Regression ☐ Acceptance ☐ Pilot<br>Specify the testing stage for this test case. | | |
| **Test Date:** | 4/12/23 | **If applicable. System date:** | Null |
| **Testers:** | Muhammad Ali Tahir<br>Farhan Bashir<br>Syed M. Sami Hassan | **Test Case No:** | 02 |
| **Description** | Sign-In to the portal. | | |
| **Results:** | ■ Pass ☐ Fail | **Incident Number** | null |
| **INTRODUCTION** | | | |
| **Requirement(s) to be tested:** | A user should have access to the used software, browser and internet connection to execute the application whichfalls into this category. | | |
| **Roles and Responsibilities:** | Muhammad Ali Tahir, Farhan Bashir and Syed M. Sami Hassan created an account and then signed in with their relevant credentials inside the | | |

| | |
|---|---|
| | application. |
| **Set Up Procedures:** | When the web browser has loaded the application. By entering your credentials, you may sign in. The process of creating a new account dashboard involves new users. The user interacts with the dashboard buttons, each of which has a specific function. |
| **Stop Procedures:** | Close the browser tab. |

| **ENVIRONMENTAL NEEDS** | |
|---|---|
| **Hardware:** | Laptop or desktop |
| **Software:** | Visual Studio, Browser |
| **Procedural Requirements:** | Go to the project application and launch the web application. Enter your credentials and click on the sign-in button. |
| **TEST** | |
| **Test Items and Features:** | Laptop or Desktop |
| **Input Specifications:** | For checking whether the user sign in is successful or not. |
| **Procedural Steps:** | Go to the application, open the web application, and sign in to the application. Enter the valid details and click on the sign-in button. |
| **Expected Results of Case:** | A user can easily create an account with the help of the web application both as admin as well as the students. After that the user can sign in any time using the valid credentials of the account. |
| **ACTUAL RESULTS** | |
| **Output Specifications:** | Users were signed in successfully after entering the valid data. |

### 5.3.3. File Uploading:

The application allows the user to sign in for the application by providing the username and thepassword that is being fetched from the database. After that, the user can upload the FYP report on to the portal.

Table 5.5: File Uploading Test Case

| GENERAL INFORMATION | | | |
|---|---|---|---|
| **Test Stage:** | ☐ Unit ■ Functionality ☐ Integration ☐ System ☐ Interface<br>☐ Performance ☐ Regression ☐ Acceptance ☐ Pilot<br>Specify the testing stage for this test case. | | |
| **Test Date:** | 4/15/23 | **If applicable, system date** | Null |
| **Tester:** | Muhammad Ali Tahir<br>Farhan Bashir<br>Syed M. Sami Hassan | **Test Case Number:** | 01 |
| **Test Case Description** | Uploading FYP report on the portal | | |
| **Results:** | ■ Pass ☐ Fail | **Incident Number** | null |
| **INTRODUCTION** | | | |
| **Requirement(s) to be tested:** | A user should have access to the used software, browser and internet connection to execute the application whichfalls into this category. | | |
| **Roles and Responsibilities:** | Muhammad Ali Tahir, Farhan Bashir and Syed M. Sami Hassan created an account and then signed in with their relevant credentials inside the application. After that they uploaded the FYP report as a word file on the portal. | | |

| | |
|---|---|
| **Set Up Procedures:** | After loading the application in the web browser. Enter the credentials and click the sign in button. After successfully signing in, enter your email and name and upload the FYP report for validation. |
| **Stop Procedures:** | Close the browser tab. |

| **ENVIRONMENTAL NEEDS** | |
|---|---|
| **Hardware:** | Laptop or desktop |
| **Software:** | Visual Studio, Browser |
| **Procedural Requirements:** | Go to the project application and launch the web application. Enter your credentials and click on the sign-in button. After that upload you FYP report as a Word file. |
| **TEST** | |
| **Test Items and Features:** | Laptop or Desktop |
| **Input Specifications:** | For uploading only word file and checking if the application is accepting the report in any other format |
| **Procedural Steps:** | Go to the application, open the web application, and sign in to the application. Enter the valid details and click on the sign-in button. After signing in, upload the FYP report as a word file. |
| **Expected Results of Case:** | A user can easily create an account with the help of the web application both as admin as well as the students. After that the user can sign in any time using the valid credentials of the account and upload their file on to the portal |
| **ACTUAL RESULTS** | |
| **Output Specifications:** | Users were signed in successfully after entering the valid data. The FYP report file was uploaded successfully and only as a Word file. |

### 5.3.4. Validation Test:

The application allows the user to sign in for the application by providing the username and thepassword that is being fetched from the database. After that, the user can upload the report in word file for validation according to the pre-defined format.

Table 5.6: Validation Test Case

| GENERAL INFORMATION | | | |
|---|---|---|---|
| **Test Stage:** | ☐ Unit ■ Functionality ☐ Integration ☐ System ☐ Interface<br>☐ Performance ☐ Regression ☐ Acceptance ☐ Pilot<br>Specify the testing stage for this test case. | | |
| **Test Date:** | 4/15/23 | **If applicable, System Date** | Null |
| **Tester:** | Muhammad Ali Tahir<br>Farhan Bashir<br>Syed M. Sami Hassan | **Test Case No.:** | 03 |
| **Test Case Description** | Uploading FYP report on the portal and their its validation according to the pre-defined format. | | |
| **Results:** | ■ Pass ☐ Fail | **Incident Number** | null |
| **INTRODUCTION** | | | |
| **Requirement(s) to be tested:** | A user should have access to the used software, browser and internet connection to execute the application whichfalls into this category. | | |
| **Roles and Responsibilities:** | Muhammad Ali Tahir, Farhan Bashir and Syed M. Sami Hassan created an account and then signed in with their relevant credentials inside the application. After that they uploaded the FYP report as a word file on the portal and provided their email. Click the submit button to start the validation. | | |

| | |
|---|---|
| **Set Up Procedures:** | After loading the application in the web browser. Enter the credentials and click the sign in button. After successfully signing in, enter your email and name and upload the FYP report for validation purpose. |
| **ENVIRONMENTAL NEEDS** | |
| **Hardware:** | Laptop or desktop |
| **Software:** | Visual Studio, Browser |
| **Procedural Requirements:** | Go to the project application and launch the web application. Enter your credentials and click on the sign-in button. After that upload you FYP report as a Word file. |
| **TEST** | |
| **Test Items and Features:** | Laptop or Desktop |
| **Input Specifications:** | For uploading only word file and checking if the application is accepting the report in any other format. After uploading and clicking the submit button, the validation begins. |
| **Procedural Steps:** | Go to the application, open the web application, and sign in to the application. Enter the valid details and click on the sign-in button. After signing in, upload the FYP report as a word file. After uploading it click the submit button to start the validation. The app marks the area red which is containing errors. |
| **Expected Results of Case:** | A user can easily create an account with the help of the web application both as admin as well as the students. After that the user can sign in any time using the valid credentials of the account and upload their file on to the portal. After clicking the submit button, the report is submitted and the validated files can be downloaded. |
| **ACTUAL RESULTS** | |
| **Output Specifications:** | Users were signed in successfully after entering the valid data. The FYP report file was uploaded successfully and only as a Word file. The validation was completed and the validated report along with the error description file was available for download. |
| **Stop Procedures:** | Close the browser tab. |

### 5.3.5 Email Services Test:

The application allows the user to sign in for the application by providing the username and thepassword that is being fetched from the database. After that, the user can upload the report in word file for validation. After the validation, an email is sent to the provided email address along with the validated word file and the error file attached to it.

Table 5.7: Email Services Test

| GENERAL INFORMATION | | | |
|---|---|---|---|
| **Test Stage:** | ☐ Unit ■ Functionality ☐ Integration ☐ System ☐ Interface ☐ Performance ☐ Regression ☐ Acceptance ☐ Pilot<br>Specify the testing stage for this test case. | | |
| **Test Date:** | 4/15/23 | **System Date, if applicable:** | Null |
| **Tester:** | Muhammad Ali Tahir<br>Farhan Bashir<br>Syed M. Sami Hassan | **Test Case Number:** | 01 |
| **Test Case Description** | Sending email to the users | | |
| **Results:** | ■ Pass ☐ Fail | **Incident Number** | null |
| **INTRODUCTION** | | | |
| **Requirement(s) to be tested:** | A user should have access to the used software, browser and internet connection to execute the application whichfalls into this category. | | |

| | |
|---|---|
| **Roles and Responsibilities** | Muhammad Ali Tahir, Farhan Bashir and Syed M. Sami Hassan created an account and then signed in with their relevant credentials inside the application. After that they uploaded the FYP report as a word file on the portal and provided their email. Click the submit button to start the validation. An email should be sent to the provided mail. |
| **Set Up Procedures:** | After loading the application in the web browser. Enter the credentials and click the sign in button. After successfully signing in, enter your email and name and upload the FYP report for validation purpose. After that email should be sent to provided mail. |
| **Stop Procedures:** | Close the browser tab. |
| **ENVIRONMENTAL NEEDS** | |
| **Hardware:** | Laptop or desktop |
| **Software:** | Visual Studio, Browser |
| **Procedural Requirements:** | Go to the project application and launch the web application. Enter your credentials and click on the sign-in button. After that, upload you FYP report as a Word file and click submit. Email with the validated file and error containing file should be mailed to the provided mail address |
| **TEST** | |
| **Test Items and Features:** | Laptop or Desktop |
| **Input Specifications:** | For uploading only word file and checking if the application is accepting the report in any other format and checking whether an email is sent to the provided mail or not |
| **Procedural Steps:** | Go to the application, open the web application, and sign in to the application. Enter the valid details and click on the sign-in button. After signing in, upload the FYP report as a word file and mail the validation file to the provided mail |

| | |
|---|---|
| **Expected Results of Case:** | A user can easily create an account with the help of the web application both as admin as well as the students. After that the user can sign in any time using the valid credentials of the account and upload their file on to the portal and validation file should be mailed to the provided mail. |
| **ACTUAL RESULTS** | |
| **Output Specifications:** | Users were signed in successfully after entering the valid data. The FYP report file was uploaded successfully and only as a Word file. An email was sent to the user's provided mail with the validation file and the error description file to the user's provided mail. |

# Chapter 6

# CONCLUSION

This chapter discusses the project's achievements and conclusions as well as any potential future work that might need to be done. It is going to cover the work that has been done on this application up till now. And what could be the future work on this project. What objective can be met to improve the system's appeal and effectiveness? What improvements can be made to the system's user interface, user backend, fronted, and database endpoint from blackened to make it better and more interesting to use? The chapter outlines the successes that have been made utilizing the system and the future improvements that are anticipated.

## 6.1. Milestones Achieved:

- **Requirement Gathering:** This milestone involves understanding the requirements of the project, including the predefined format of the "National University of Modern Languages FYP report" and the specific validation criteria to be implemented.
- **System Design and Planning:** This milestone includes designing the system architecture, database structure, and user interface. It involves defining the tables in the database to store the predefined format, implementing the logic for comparing the uploaded FYP report with the format, and planning the overall project timeline.
- **Implementation and Integration:** This milestone marks the development of the web application using ASP.NET MVC and integrating the Spire.doc library for document processing. It involves implementing the validation logic, creating the user interface for uploading reports, and integrating SMTP services for sending autogenerated emails.
- **Testing and Quality Assurance:** This milestone includes testing the application for its functionality, performance, and usability. It involves validating the validation process against various test cases, ensuring the accurate identification of mistakes, and conducting usability testing to ensure a smooth user experience.
- **Deployment and User Acceptance:** This milestone involves deploying the web application to a production environment, configuring the hosting infrastructure, and conducting user acceptance testing. It includes tasks such as setting up the administration portal, generating unique tokens for users, and ensuring the email notifications are properly delivered.
- **Documentation and Handover**: The project reaches completion with the documentation of the project, including the validation process, system architecture, and user guides. It involves preparing for the handover of the project to the stakeholders and providing support and maintenance plans.

## 6.2. Conclusion:

Our project is a web application that simplifies the validation process of Final Year Project (FYP) reports at the National University of Modern Languages. It compares uploaded reports with the predefined format, efficiently identifying mistakes in formatting, heading sizes, fonts, and more. By generating a logging file, users receive a comprehensive validation result along with the validated word file which highlights every single mistake that was define. The user-friendly interface allows students to easily upload reports, receive autogenerated emails containing validated logs and error-highlighted reports, and conveniently download files from the portal. Also, every execution result is kept in the database along with time of executions and student can download previously validated word and log files. The administration portal enables efficient tracking of validated reports using automatically generated tokens.

The key features of our project include accurate report validation, automated email notifications, seamless user interface, and efficient storage of predefined formats. Talking about the predefined format, the format of first seven static pages is kept in the database as there are some fixed words which must be there, other than that every chapter is validated on the base of table of content and every caption of image and table is validated on the base of list of tables and list of figures. Hence whatever there is in table of content, list of figures and list of tables has to be present in sequence in chapters. By automating the validation process, we save significant time for both students and the supervision panel, eliminating the need for manual checking. The developed system has achieved its objectives by improving the quality and adherence to the predefined format. Users appreciate the user-friendly interface and prompt delivery of autogenerated emails. The integration of Spire.doc library and ASP.NET ensures robust document processing and frontend development. Overall, our project effectively enhances efficiency, accuracy, and convenience in the FYP report validation process at the National University of Modern Languages.

## 6.3. Limitations:

There are some limitations and future work of this project as there are limitations in the library **SPRIRE.DOC.** Limitation of the project by Spire.doc library used for document processing include:

- Our system validates the manual table of content as auto-generated table of content is not handled by Spire.Doc.
- As it does not validate the auto-generated table of content, so we had to apply validations on table of content and chapters headings to be validated.
- As we are validating captions of images and tables from List of figures and List of tables, we were not able to validate their images and as well as tables as they could be randomly placed in chapters and also due to the limitations of SPIRE.DOC library which didn't allowed us to do so.

## 6.3. Future Work:

Future Work could involve implementing additional validation mechanisms to address limitations form our work. This could include developing custom algorithms to verify the structure and accuracy of table of contents and conducting image analysis to ensure proper placement and formatting. The accuracy and completeness of the validation process would be significantly improved with improvements in these areas.

### 6.3.1. Make it Dynamic:

As there are always changes occurring in the defined format of FYP Report from the university, so it is a better approach to handle validations dynamically rather than depending on predefined format in the database. Our work is already 60% dynamic as other than first 7 pages validations are applied on the basis of table of content which student has entered. But the first 7 pages use the validations which are predefined in the database. By making it dynamic, no one has to worry about applying changes in database for the first 7 pages rather the format is set by inserting a FYP SAMPLE REPORT which has the correct format and our system would fetch the validations from that Sample Report and will compare it with FYP Report which is inserted by student. Now if the defined format is changed by the university, we just have to insert the FYP Sample Report with updated format and it is good to validate student FYP Reports with the new format. This will not only work for Department of computer science and National University of Modern Languages but it could be used in every university or offices.

### 6.3.2. Handle limitations of SPIRE.DOC:

Overcome the limitations from our work as SPIRE.DOC has some limitation which restricted us to validate auto-generated table of content, pictures and tables whose captions are mentioned in list of figures and list of tables (as we were able to validate captions mentioned in list of tables and list of figures but due to limitations of Spire.Doc we were not able to validate their images and tables). In order to do that, consider exploring other document processing tools or libraries that provide more thorough validation capabilities for Microsoft Word files. Research and explore options to integrate or collaborate with such tools to enhance the validation functionality of this format checker.

### 6.3.3. Normalize use of it in universities:

As of now there occurs some limitations which are discussed above, they limit the use of it as if the FYP coordinator has to apply some changes to the FYP format he has to go into the database of our system and then deal with coding stuff. By handling the above limitations and future work discussed normalize the use of it in our university and other universities as it has the potential to do so.

# APPENDICES

## Definition of Terms:

**SMTP**

SMTP stands for “Simple Mail Transfer Protocol”.

**SSMS**

SSMS stands for “SQL Server Management Studio”.

**MVC**

MVC stands for “Model View Controller”.

**DFD**

DFD stands for “Dataflow Diagram”.

**ASP.NET**

ASP.NET stands for “Active Server Pages Network Enabled Technologies”.

# REFERENCES


Nuget, ".NET Library to Create, Manipulate & Convert Word Documents", [Online]. Available: https://www.nuget.org/packages/Spire.Doc/

Microsoft, "Handle requests with controllers in ASP.NET Core MVC", [Online]. Available: https://learn.microsoft.com/en-us/aspnet/core/mvc/controllers/

Iceblue, "Find and Highlight Text in Word", [online]. Available

https://www.e-iceblue.com/Tutorials/Spire.Doc/Spire.Doc-Program-Guide/NET-Word-Find-and-Highlight-Text-in-Word-with-C-VB.NET.html

Guillaume Endignoux, Olivier Levillain, Jean-Yves Migeon, "A Pragmatic Approach to PDF Parsing and Validation", [Online]. Available

https://ieeexplore.ieee.org/abstract/document/7527763

OV Mazurets , OV Kovalchuk , VO Slobodzyan, "Using specialized software extensions to automate work with digital documents of educational materials", [Online]. Available

http://journals.khnu.km.ua/vestnik/pdf/tech/pdfbase/2018/2018_1/(257)%202018-1-t.pdf#page=61

John Ciliberti, "Data Validation Using ASP. NET Core MVC", [Online]. Available

https://link.springer.com/chapter/10.1007/978-1-4842-0427-6_9