**NORTH SOUTH UNIVERSITY**

**CSE 499A Section 21**

**Project Paperless**

Technical Report

**Instructor**DR. MD SHAHRIAR KARIM

**Due Date  
31st** December 2019, **Tuesday**

Group: **Bright Sparks**

SUMAIYA NOORJAHAN SHAHADAT HOSSAIN PAVEL **151 2065 642 161 2332 042**

FAREHA ALAMGIR TANZIM AL DIN AHMED **152 1464 642 162 1203 042**

FAHAD BIN BARI SHOVO **161 0162 042**

**Introduction**

Every year a huge amount of paper is being used in Thesis/Report Submission in the Universities. Interestingly, a huge portion of these bulky paper works are simply discarded because it contained errors or due to inappropriate formatting. And the resubmission process is not only time consuming but also wastes a huge amount of paper. Storing these documents also requires a larger space, thus few proportions is kept. It is not only waste of space and resource but also has a negative impact on the environment. And also, sometimes the owner loses the documents after a long period of time, which is another issue. A large amount of greenhouse gases is being released during the processing of paper.

In this world of growing technology, it has now become more meaningful to store a digital version of these documents. Day by day more and more organizations are incorporating "Go paperless" concept because not only does it makes storing documents easier and reduces the chance of losing them but also contributes towards a greener environment. And this concept is particularly beneficial to students because it not only opens the doorway to share knowledge across the globe, but also ensures that all the course materials are available at their fingertips, anytime, anywhere!

To solve the aforesaid problem, we propose a convenient, secure and more efficient, “Greener” solution that reduces the carbon footprint of academic document management system. Precisely, would allow online submission and storage of these important documents such as thesis, reports etc. and would also ensure the exact formatting and other aspects of the documents prior to submission. And an all-in-one system where the students can create and upload their course materials, so they never have to worry about them again.

**Method and Design**

The proposed system would consist of a centralized database and server to maintain and store the course contents uploaded by the students, and also the submitted thesis papers. The system would not only help to store the thesis papers but would also perform checks on the document formatting to identify whether it is correctly formatted and plagiarism checks prior to submission. It is quite unique up until now, along with the fact that the platform tries to provide an all in one solution for students and as well as the teachers. Students and teachers would have their respective accounts with students being able to create, store and view course contents, and submit, store and view the thesis papers and teachers would be able to view those course contents, revise those contents upon requirement and view thesis papers or upload the templates which would be used for the format checking.

Each student is required to have an account to use this service. To create an account, a student must fill up the form provided by the system, with the student identification number provided by the University. Internally, the system would validate the request by acquiring information from the university server and create a new account. Registered students would then be allowed to use the services provided by the system. As mentioned earlier, each student would be allowed to create course content, revise previously created course contents, add collaborators for content creation, submit a new thesis file, revise the submission upon requirement, and view submitted thesis files.

Similarly teachers would also require to have their respective accounts, and to create an account, a teacher is also required to fill up the respective form provided by the system, where they would require to provide with the teacher identification number provided by the University, and after a successful validation, the new account would be created. Teachers would be able to upload template files, which the system would use to validate the format of the document, and they can also verify any thesis paper submission requested by a student, and view the thesis file any time after it is successfully submitted by a student. Additionally, teachers would be able to revise the course contents created by the students, upon requirement.

This section discusses the typical steps required to submit a thesis file using the proposed system. After a student has uploaded a thesis paper for submission, the file would first be stored into a temporary storage in the server. The system would then compare the format of the submitted document with the template uploaded by the teacher or the department for the particular submission, and then notify the student of certain errors, if there is any, otherwise the system would send a confirmation request to the concerned teacher. The document can be revised any number of times during this period. After a successful validation from the teacher, the document would then be permanently stored into the server. During this process, a digital signature would be attached to the document for authenticity. Any revision of the document from this point onwards would require passing through the above-mentioned steps again.

The students would be able to view their submitted thesis papers for free for a limited period of time. But teachers would always be able to view the papers those were submitted to them.

For the course content creation part, the registered students would be provided with an interface where they can create the content. The interface would contain a WYSIWYG HTML editor with popular formatting options where the students can easily type in their contents. The editor would also allow the students to add equations with a friendly interface. It is to be noted that, it would also support equations written in the Latex and ASCII Math formats. A live preview of the content would be shown to the students to help them create aesthetically pleasing layouts. After saving the contents, the students would be allowed to revise them any number of times required. A student may add collaborators to help him/her in the creation or revision process. The system would contain dedicated pages to display these uploaded course materials, and any registered student would be able to view those contents regardless of their departments.

Faculty members would be able to view and revise the contents uploaded by the students. Faculty members are acknowledged to review the uploaded contents concerned with their respective courses, to maintain the reliability of the service.

**Tools**

The system would be implemented using HTML, CSS, JavaScript, Python, Django (Python-Web Framework).

Since the system would be implemented using the Client-Server architecture, using a web application would ensure an easier solution for a wide variety of devices. Only a device with a working Internet Connection and a Web-Browser would be required to use this service.

HTML5, CSS3 and JavaScript would be used to build the client side of the application. The user interface would be a responsive interface to ensure a seamless experience across most of the devices. The server side would be built using Python (Django).

Python is one of the most used Programming Languages that are being used now, due to its strong emphasis on readability and efficiency compared to other languages like PHP. It is very easy to learn compared to many other languages and many complex functionalities can be implemented with very few lines of codes. Apart from that, Python is also very flexible, that is, it has several robust integrations with other programming languages. For example, CPython - integrated with C, Jython – integrated with JAVA and so on. And since, it is quite popular among the programming community, it offers a wide range of resources or packages.

Python also offers a very high-level web-framework, Django, that encourages rapid development, and clean and pragmatic design. It is very robust, extremely fast and secure, and also very scalable all at the same time. Organizations like Instagram, The New York Times, The Guardian, MIT, NASA, National Geographic and many more have incorporated Django into their systems.

This system would incorporate Django to maintain the server, and Python-Docs module for the word document handling.

This section contains information about the additional libraries used for the respective tasks. Note that, this is not the final version of the implemented technologies, and more would be added if required in the future.

Back – end:

* Python Django for the website architecture.
* Python-docx for Word Document processing.
* PyPDF2 for PDF Document processing.
* PyLaTeX for Latex Document processing.

Front – end:

* HTML, CSS, JavaScript for the client-side interface.
* Bootstrap 4 for beautification of the client-side interface.
* CK Editor for content creation and revision. It is a WYSIWYG HTML editor which can generate HTML contents based on user inputs.
* Wiris (Integrated with CK editor) provide with simpler interface to add mathematical and chemical equations (generates user defined equations in MathML format).
* MathJax to display Mathematical and Chemical equations (written in Latex, ASCII Math or MathML formats) on the webpage.
* Ace Editor to allow adding code snippets into the webpage.

**Timeline**

The project mainly consists of four main phases: Front-end or UI, Back-end or Server, Document Processing for checking validation, and Testing phases. Below is a proposed timeline of the work flow.

* Month 1: Data collection, learning required dependencies (if required), creating documentation, and development of the front-end / UI, and creating Thesis document Templates. (Consisting usage of HTML, CSS, JavaScript, and also Microsoft Office Word for creating document templates).
* Month 2: Development of the back-end / server-side application, and development of modules for document processing. (Consisting usage of Python (Django), and Python-docs mainly for document handling).
* Month 3: Testing and finalizing. (Though we would focus on Test Driven Development procedures).

**Recent Works**

* Created and updated Project Documentations.
* Generated cover page based on inputs from the user with University Name and Logo. Course title, section, project title, instructor’s name, group name, group member(s) name(s) etc. were taken as inputs form the user (terminal based).
* Created basic HTML forms, with CSS, JavaScript, Bootstrap, for the User Interface. Forms to Register, Login; Dashboards for Students, Teachers. etc.
* Implemented algorithms to test the properties of a word document. Page orientation, size, margins, paragraph alignment, line-spacing, font name, size, color etc. were displayed in the console using the module.
* Implementation of Back – end of the website started.
* Trialed a HTML page to generate a Thesis Cover page based on user inputs.
* Trialed CK-editor and MathJax to take contents from the user and display it in the web browser in real-time.
* Trialed Plagiarism checking algorithms.

Learning Django, Python Web Framework.