



TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
HIMALAYA COLLEGE OF ENGINEERING
Chyasal-09, Lalitpur
[CT-654]
A THIRD-YEAR MINOR PROJECT REPORT
ON
QUESTION PAPER GENERATOR
AND MODERATOR

Submitted by:

Ashok Ghimire (18758)

Resarch Paija Pun (18771)

Sabal Thapa (18774)

Kishor Humagain (18791)

Submitted to:

Department of Computer and Electronics Engineering

May,2022

**“QUESTION PAPER GENERATOR
AND MODERATOR”**

[CT-654]

**“A THIRD YEAR MINOR PROJECT REPORT
SUBMITTED FOR PARTIAL FULFILLMENT OF DEGREE
OF BACHELOR’S IN COMPUTER ENGINEERING”**

SUPERVISOR

Er. Himal Chand Thapa

Tribhuvan University

Institute of Engineering

Himalaya College of Engineering

Chyasal-09, Lalitpur

Submitted by:

Ashok Ghimire (18758)

Resarch Paija Pun (18771)

Sabal Thapa (18774)

Kishor Humagain (18791)

Submitted to:

Department of Computer and Electronics Engineering

May, 2022

ACKNOWLEDGEMENT

With the accomplishment of this project, we would like to say few words of appreciation to those people who have guided and assisted us.

Firstly, we are grateful to Institute of Engineering (IOE) for including minor project in the syllabus of BCT III/II. Similarly, we are thankful to Himalaya College of Engineering (HCOE) for providing us with the resources to assist in the completion of the project. We are also grateful to our respected HOD Er. Ashok Gharti Magar and Deputy HOD Er. Devendra Kathayat for their advices and supports. We would like to express our gratitude to our project coordinator Er. Ramesh Tamang and our supervisor Er. Himal Chandra Thapa who has guided us throughout our project.

Lastly, we are also thankful to all our colleagues and our respective teachers for their support.

Ashok Ghimire (18758)

Resarch Paija Pun (18771)

Sabal Thapa (18774)

Kishor Humagain (18791)

ABSTRACT:

Examination process has always been the traditional and the current system of examination is lacking to provide easy question paper generation and fair examination. So, this report analyzes how efficiently the question papers can be made and how safely and quickly the teachers can shift towards modern way of generation of the question paper.

We have come up with the solution to shift the traditional practice to modern technology-oriented generation of question paper. We have successfully made the Question Paper Generator and Moderator, which is fast, randomized, streamlined and secure. Using our system, the model question papers can be generated very fast with the click of a button. Every generated question paper are unique which makes fair generation of the question paper. The system works starting with the input or the bunch of characterized questions given by the users in the form of pdf file. The pdf file is then scanned using OCR to extract the text i.e., questions. The questions are then stored in database with respect to their mark and then they are generated in question paper format when user request the system to generate the question paper.

Keywords: *Examination, Questions, OCR, Regular Expression, Automation, Randomization, Secure, Printed Text Recognition, Threshold*

TABLE OF CONTENTS

ACKNOWLEDGEMENT	i
ABSTRACT:.....	ii
TABLE OF CONTENTS	iii
LIST OF FIGURES.....	v
LIST OF TABLES	vi
LIST OF ABBREVIATIONS	vii
1. INTRODUCTION.....	1
1.1 Back ground and Motivation.....	2
1.2 Problem Statement	3
1.3 Objectives	4
1.4 Features	4
1.5 APPLICATIONS	5
2. LITERATURE REVIEW	6
3. REQUIREMENT ANALYSIS	8
3.1 Software and Hardware Requirements	8
3.2 Functional Requirements	10
3.2.1 Use Case Diagram.....	11
3.3 Non-Functional Requirements	12
3.4 Feasibility Study.....	13
3.4.1 Technical Feasibility	13
3.4.2 Operational Feasibility	13
3.4.3 Legal Feasibility	13
3.4.4 Economic Feasibility	13
4. System Design.....	14
4.1 System Flow Diagram.....	14

4.2. Data Flow Diagram (DFD)	15
4.2.1 DFD-Level-	15
4.2.2. DFD- Level 1	16
4.3 Sequence Diagram	17
4.3.1. Sequence Diagram for Admin	17
4.3.2 Sequence Diagram for Teacher	18
4.4 ER Diagram.....	19
4.5 Activity Diagram.....	20
4.5.1. Activity Diagram for Admin	20
4.5.2. Activity Diagram for Teacher	21
5. METHODOLOGY	22
5.1 Incremental Model	22
5.2 Tools Used.....	31
5.2.1 Python.....	31
5.2.2 HTML.....	31
5.2.3 CSS.....	31
5.2.4 Javascript.....	32
5.2.5 Django	32
5.2.6 VS-Code.....	32
6. SYSTEM TESTING	34
6.1 Unit testing	34
6.2 Integration Testing	38
7. RESULT ANALYSIS	39
8. CONCLUSION	40
8.1 Further Enhancement	40
References	41
APPENDICES	42

LIST OF FIGURES

Figure 1 : Use Case Diagram	11
Figure 2 : System Flow Diagram	14
Figure 3 : DFD-level-0 Diagram.....	15
Figure 4 : DFD-Level-1 Diagram.....	16
Figure 5 : Sequence Diagram for Admin	17
Figure 6 : Sequence Diagram for Teacher	18
Figure 7 : ER Diagram	19
Figure 8 : Activity Diagram for Admin	20
Figure 9 : Activity Diagram for Teacher.....	21
Figure 10 : Incremental Model.....	22
Figure 11 : System Architecture.....	23
Figure 12 : Flowchart for OCR	26
Figure 13 : Flowchart for Regular Expression.....	30

LIST OF TABLES

Table 1: Hardware and Software Requirements	9
Table 2: Test case for user login	34
Table 3: Test case for Dashboard input.....	35
Table 4: Test case for OCR scanning.....	36
Table 5: Test case for regular expression.....	37

LIST OF ABBREVIATIONS

1. HOD: Head of Department
2. OCR: Optical Character Recognition
3. DFD: Data Flow Diagram

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

Technology today has become the driving force for everything. Governments invite a lot of money in institutions to update and improve the technological world, the major contributors of this technology movement are educational institutions. Education has become the key to national development. Universities and colleges are being opened and established in all parts of the country, with this number of new universities the critical question remains to be what is the quality of students being produced at these universities. The most traditional way of determining students understanding has been examinations. Examinations play a very important role in the evaluation of students and determining students' progress. Therefore, the quality of examination is the key to the credibility of the institution. The traditional way of preparing exams has always been that the lecturers, professors, instructors, set the question paper and them submits it for review by the examining board or senate. This method introduces a lot of human errors and high possibility of paper leakages. Since the papers are done in hard copy, storage also becomes a problem to those institutions wishing to maintain a database for past examination papers.

A design of suitable automated system for generating question papers and managing related data may prove vital in an Educational Institute. In this paper, we have proposed an integrated automated system that stores questions related to a particular course and prints a question paper based on its syllabus and curriculum which will be based on website and done from it. We have implemented a role-based hierarchy which restricts access to the users. The system also deploys security mechanisms that prohibit duplication of question papers. There are provisions to enter and edit data suitable to any educational organization with complete freedom for specifying courses, semesters, syllabus and pattern. This enables an educational institute to generate question ensuring security and non-repetitiveness of question papers and is a boon for organizations with limited staff and resources. Our system aims to provide fast operations, data storage and high security for all its tasks. The evolution of

traditional and existing Question Paper Generation systems and the need for an automated system is most essential for this present time. In this project, we have been working to make our system of Question Paper Generation. This system will be using OCR to scan the documents or questions and add it into the database.

1.1 Background and Motivation

OCR stands for "**Optical Character Recognition.**" It is a technology that recognizes text within a digital image. It is commonly used to recognize text in scanned documents and images. OCR software can be used to convert a physical paper document, or an image into an accessible electronic version with text. For example, if we scan a paper document or photograph with a printer, the printer will most likely create a file with a digital image in it. The file could be a JPG or PDF, but the new electronic file may still be only an image of the original document. We can then load this scanned electronic document it created, which contains the image, into an OCR program. The OCR program which will recognize the text and convert the document to an editable text file.

A **Regular Expression** is a method used in programming for pattern matching. Regular expressions provide a flexible and concise means to match strings of text. Regular expressions are also known in short form as regex or regexp. Regular expressions can be incredibly powerful. Essentially, if the pattern can be defined, a regular expression can be created. A simple pattern might be something as simple as finding all situations where a sentence ends in "that" and is replaced with "which". The pattern could get more complex by doing the same replacement but only on the 3rd and 5th occurrence of a match. Or it could get even more complicated by using different sets of matching characters depending on the frequency and location of previous matching characters. Regular expression allows us to surf through a file containing the large amount of random words, any characters or alphanumeric characters in random order. When surfing, we can match the pattern given by our

regular expression with the random group of words, characters or alphanumeric characters. We can extract the texts of our required pattern from the matched list and use them for our own purpose.

As most human working processes, this system suffers due to bias. There might be some questions which are repeated in many question papers as the professor has a personal inclination towards them. So there is no guarantee of pure randomly generated question paper. Other problems that may plague this system are non-availability of staff and resources, systematic errors, natural calamities and accidents. Also, the security of the system can be easily compromised if leverage over the person responsible for generating question papers is obtained. Other limitations include:-

- Lack of storage space
- Prone to damage
- Inefficient document transportation
- Supply costs
- Poor environmental credentials
- Limited collaboration
- Editing problems

1.2 Problem Statement

Generating the question paper in a traditional way is troublesome and the teacher themselves should have to prepare the paper consuming more time. Therefore, question paper generator will save the time consumed for preparing the papers manually. As generating paper is very time consuming and is needed to type the

questions every time although the questions are similar, this system helps to moderate the papers as well as generate the papers. Also, if teacher wishes to update questions in the database they can update with the help of this system. We can avoid time consumption to generate question paper at the certain time.

1.3 Objectives

- To generate the question paper in certain amount of time that saves the consumption of time as occupied by manual paper generation.
- To moderate the paper i.e. to add any new questions into the database such that new questions can be provided for examinations.

1.4 Features

Following are some of the features of the Question Paper Generation and Moderation System:-

- Simple interface which enhances the ease of updating data.
- Generates and prepares the Question Paper in specific time.
- This system will be website-based.
- Questions can be easily edited.

The Question Paper Generation and Moderation System provides various advantages to the user when compared to the traditional system. Listed below are some of the advantages of the system: -

- User can generate test papers randomly and instantly, saving a lot of time.
- The algorithm enables randomization of questions so avoids the often repetition of questions.

- A new question can be added to the database at any instant and different sets of test papers could be generated.

1.5 APPLICATIONS

This Question Paper Generation and Moderation System has certain specific applications. Firstly, it is applicable to generate the papers in very short period of time which will ultimately save the time that is consumed during the traditional way of generating the papers. Similarly, if the teacher wishes to add new questions it can be instantly added into the database and the question can be given in the next examination. Teachers often tend to repeat a lot of questions that had been asked before. So with this system, the paper can be generated in short period of time and there is no need to frequently type the same questions always. Also, new questions can be added into the database and new questions may also appear in the paper.

Other applications are:

- This system can be used by all educational institutions for the quicker generation of papers.
- This system can be used to moderate the papers and add new questions instantly.

CHAPTER 2

LITERATURE REVIEW

2. LITERATURE REVIEW

Like many other systems, Question Paper Generator system has been developed previously by various personnel. Examinations are the important aspect that determines the future of students. We can also say that examinations fully determine the future of students. However, teachers have been frequently facing difficulties and is being more challenging to create papers in traditional ways.

Mrunal Fatangare [1] have suggested a solution to choose challenging, well framed questions and make it easy for the instructor to generate them within a short period of time. This can be wiped out of a couple of taps of the hand because it is an Android application, therefore accessible at any time and place. It contains various modules which enable the system to affect all questions easily. The module like admin module, user module, and question entry and question management makes it an easy task. Their proposed system uses randomization algorithm which includes Bloom's Taxonomy levels.

Amit Khairnar [2] he proposed a system that enables college authorities to automatically generate question papers out of existing question bank in the database. The system will have the capability to process different unique sets of papers automatically. It takes over the entire tedious task and does the manual work swiftly and efficiently.

Mihir Joisher [3] presented the shuffling algorithm which is applied to randomize the questions generated from the database. The algorithm applied by them is quite straightforward and also extremely easy to understand. The implementation of this algorithm can be easily performed in other systems as well. This system is mainly used in colleges, institutions, to produce and test the paper setters who mainly want to have a huge record of questions in the question paper. Classification and organization of huge quantity of queries about unlike levels of toughness from methodical as well as non-scientific subjects related to several

classes. This system mainly presents the practices of procedures in Randomized Generated Question Paper System to overcome the above mentioned problem. The main part of the procedures is to deliver a random method in the group of systems, thus different sets of questions could be mainly generated without replication and duplication.

Prateek Pisat [4] have proposed a system that focuses on the development of detailed and analytical question paper with stats. The algorithm applied is much complex as the questions are meant to be comprehensive and are inclined towards systematic and logical thinking. There are basically two modules in the system.

Vijay Purohit [5] who proposed a system that is based on Bloom's Taxonomy. It is an adaptive system however the data entered is assumed to be completely error free which can affect the overall accuracy of the system. The objective of this software for colleges and institutions is to exploit the potential of expertise around the world in collecting questions and effectively managing the collected questions. Then inculcate the system with intelligence to generate the question papers as per the requirement (Marks, Skills, Difficulty Level, Syllabus, Type, Test) of the course teachers of the institutions for particular courses

CHAPTER 3

REQUIREMENT ANALYSIS

3. REQUIREMENT ANALYSIS

3.1 Software and Hardware Requirements

As it is web based, the external software requirement for the client side is not specific. Client can access from any web browsers. For developers, certain software are used to develop the web page. Simple developer tools with low memory space can be used to develop the project.

Due to the low processor load utilized by our project, high-end hardware requirement is not an issue for the client. But for the developer to build the overall project from scratch to complete, hardware requirement of mid-range can be good pick. Moreover, server-side hardware can be quite complex according to the user's engagement.

Table 1: Hardware and Software Requirements

	Hardware	Software/Language/Framework
Client	Mobile, Computer Any PC above 2GB RAM Any mobile above 540MB RAM	Web Browser Windows 7 and above
Developer	Computer Above (I5 7-Gen, CPU 1.8GHZ, 8-GB RAM)	Visual Studio Code HTML, CSS, JS, Db.sqlite3, Django Windows 7+ OS
Server	3.6 GHZ or fast CPU 10 GB HDD or 5GB SSD (Developer specification for local host server)	Django Live server (Can be implemented in online server)

3.2 Functional Requirements

Login:

Admin and Teachers have unique username and password. They can login to our system simply entering correct login username and password.

Interface:

Our user interface is simple, responsive and interactive. User will have easy idea on how to use our system.

Uploading raw questions:

Admin has right to upload the raw data (group of questions) in pdf format to save to the database. Only admin can modify the questions in database.

Dashboard:

Teachers will have a dashboard where they can fill up the fields to generate the question paper.

Viewing Question paper:

The generated question paper can be viewed in web page after proceeding to generate. The question paper can also be downloaded for further use.

3.2.1 Use Case Diagram

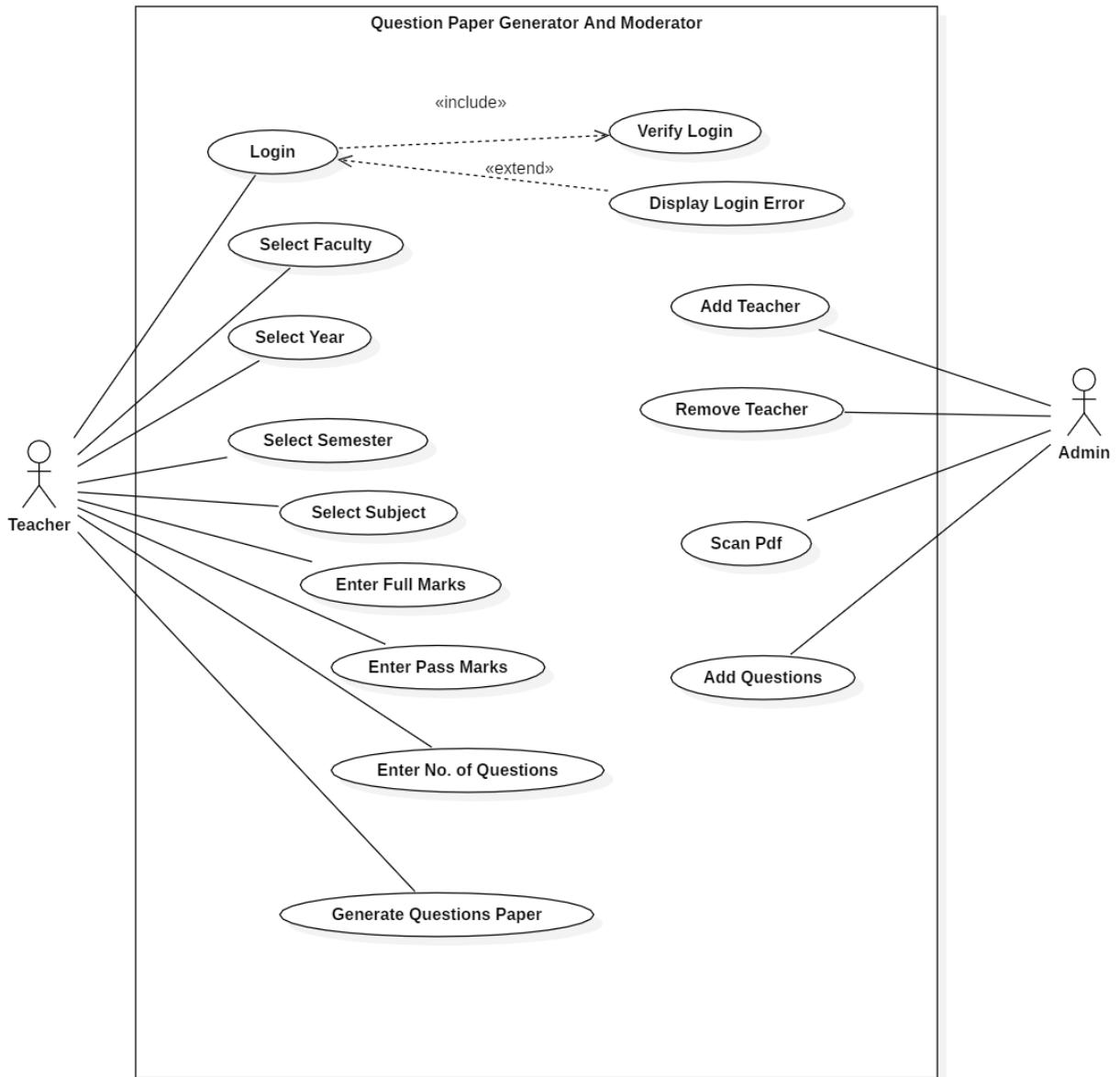


Figure 1 : Use Case Diagram

3.3 Non-Functional Requirements

Time Availability:

Currently the project is run in localhost Django-server. So, we can access the web-app as long server is run.

Performance:

OCR scanning of a pdf and storing of questions in database can be executed in less than 8 seconds.

Login Security:

Only admin can add teachers to the system by providing login credentials.

System Security:

A user trying to access other webpage's url will be redirected to login page if the user isn't authenticated.

Maintainability:

Our project is easy to extend with new functionality.

Individual table for Subjects:

Questions can be fed to OCR system as much as admin can and they will be stored subject-wise in the tables of database.

Uniqueness:

The contents of tables in database are unique to each other. This will generate unique questions.

3.4 Feasibility Study

3.4.1 Technical Feasibility

This system is capable of handling the user data as well as questions in a secure database. All the technology required by the system is easily available and is modifiable.

3.4.2 Operational Feasibility

This system provides a simple user interface. This system also provides reliable services to its users. Thus, this system is operationally feasible.

3.4.3 Legal Feasibility

This system follows all the legal contents and doesn't function against any laws. So, it is legally feasible.

3.4.4 Economic Feasibility

Developing this system doesn't require too much investments so it is economically feasible.

CHAPTER 4

SYSTEM DESIGN

4. System Design

4.1 System Flow Diagram

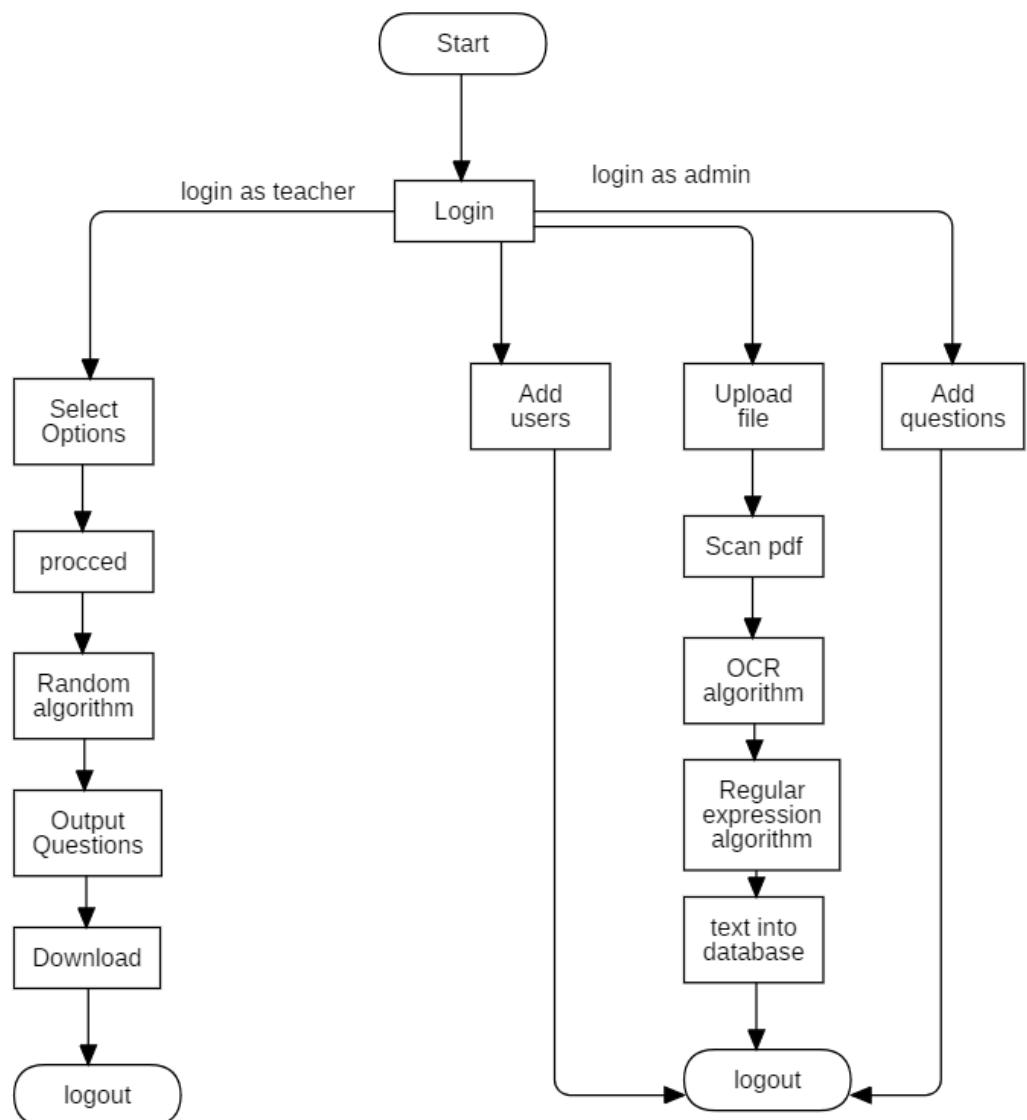


Figure 2 : System Flow Diagram

4.2. Data Flow Diagram (DFD)

4.2.1 DFD-Level-0

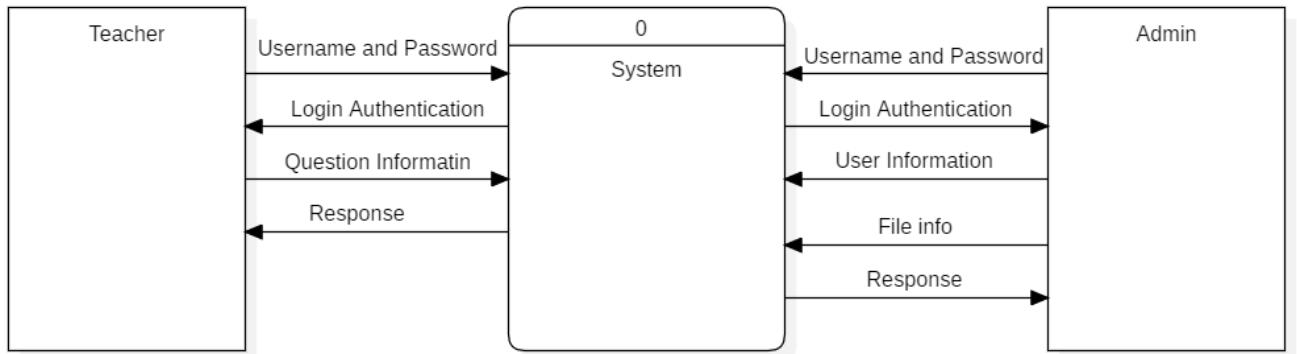


Figure 3 : DFD-level-0 Diagram

4.2.2. DFD- Level 1

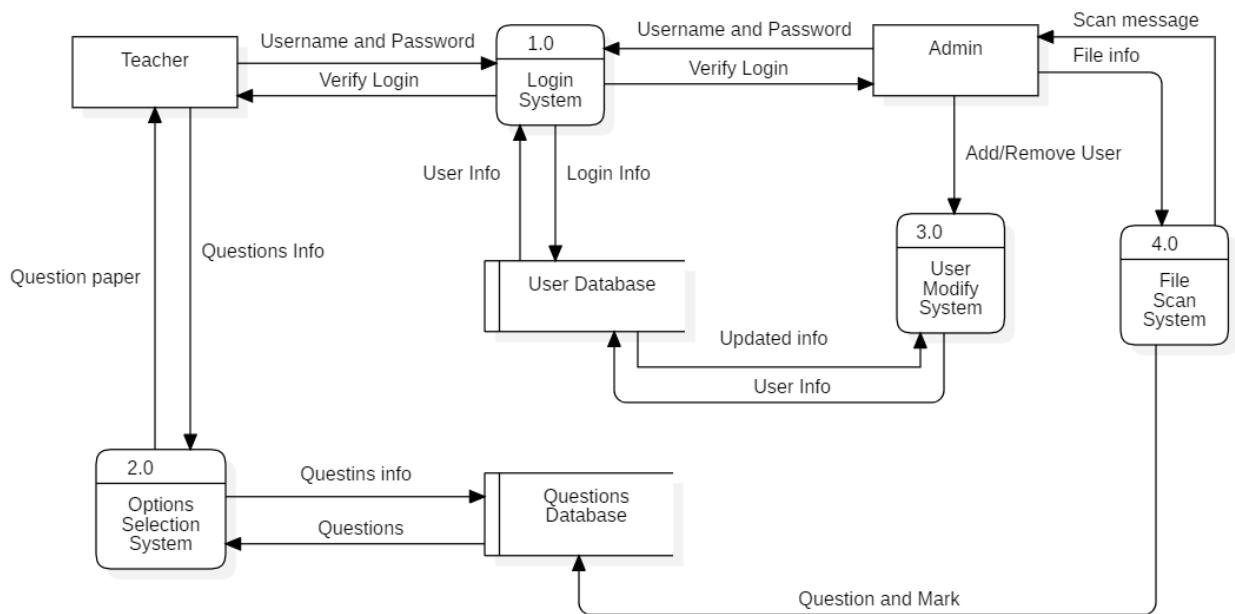


Figure 4 : DFD-Level-1 Diagram

4.3 Sequence Diagram

4.3.1. Sequence Diagram for Admin

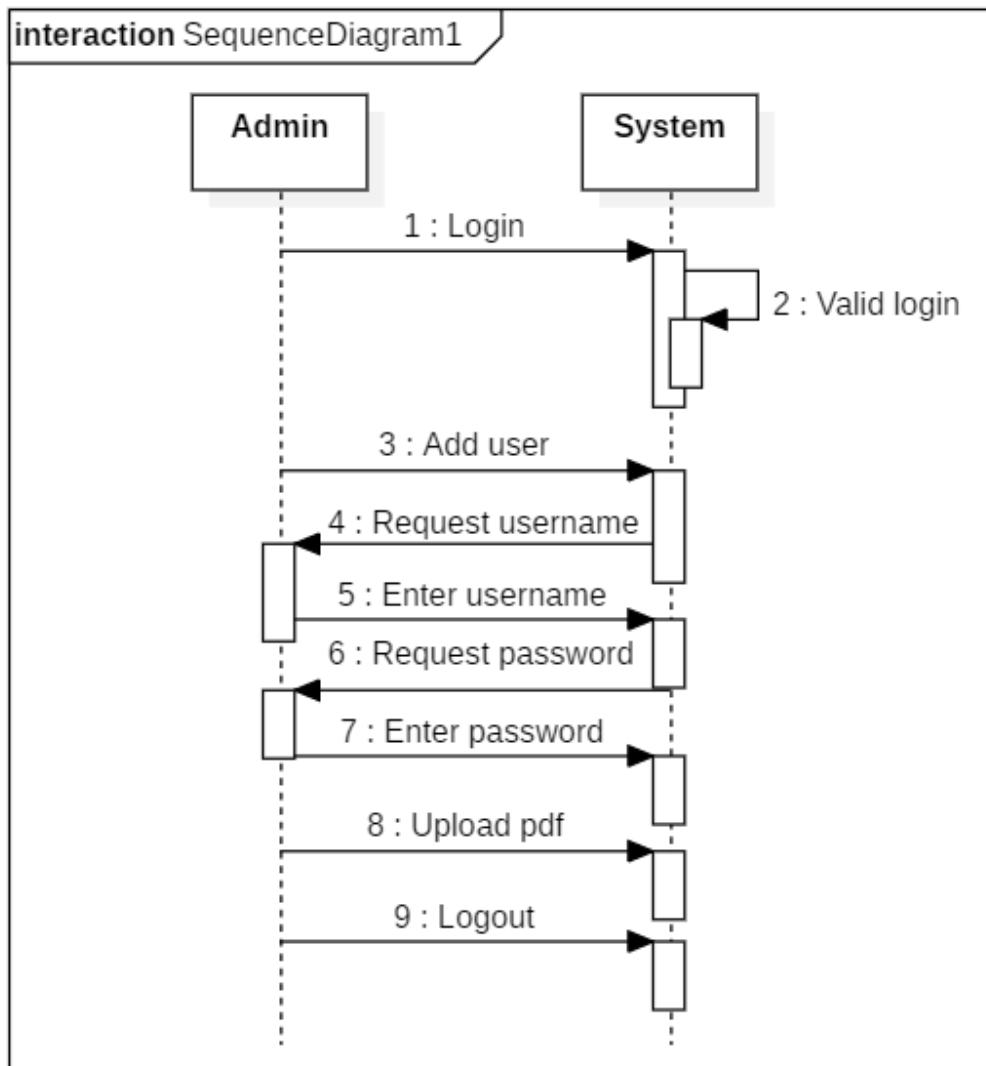


Figure 5 : Sequence Diagram for Admin

4.3.2 Sequence Diagram for Teacher

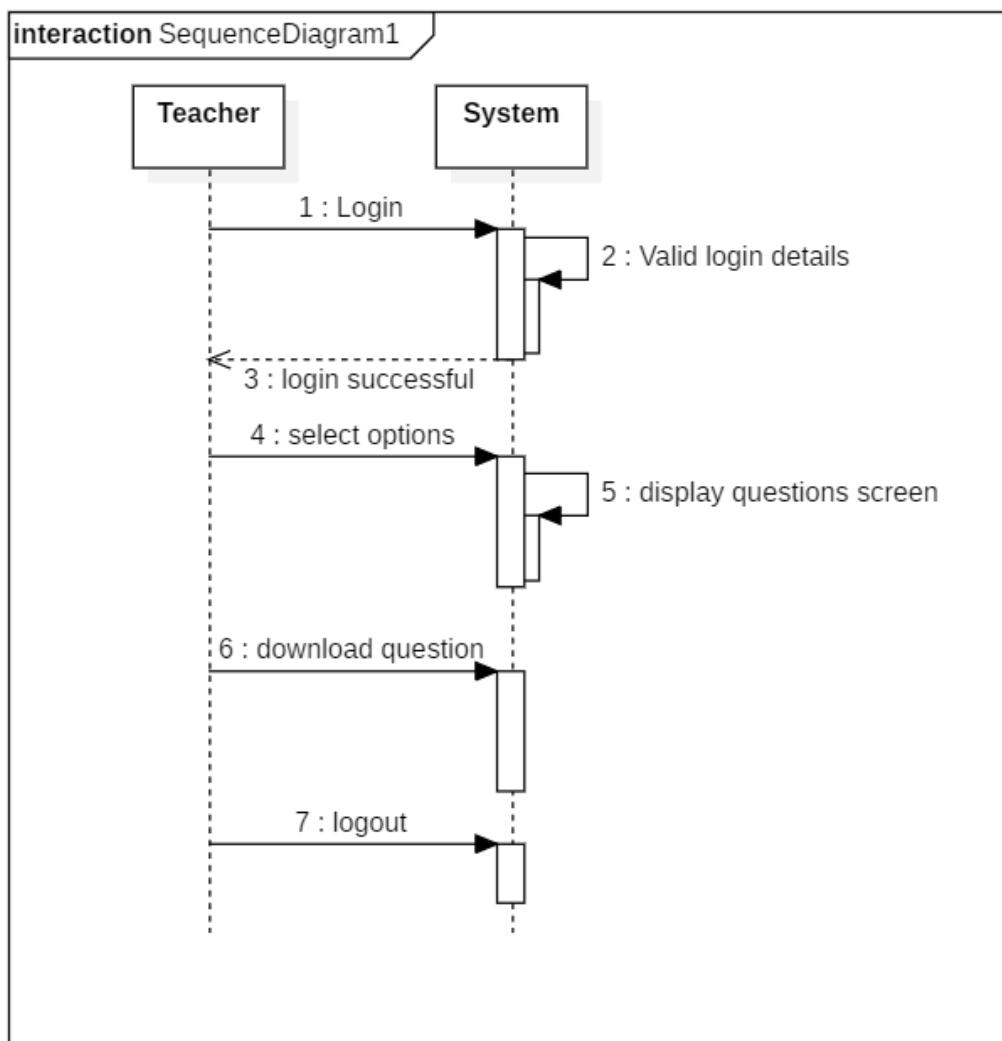


Figure 6 : Sequence Diagram for Teacher

4.4 ER Diagram

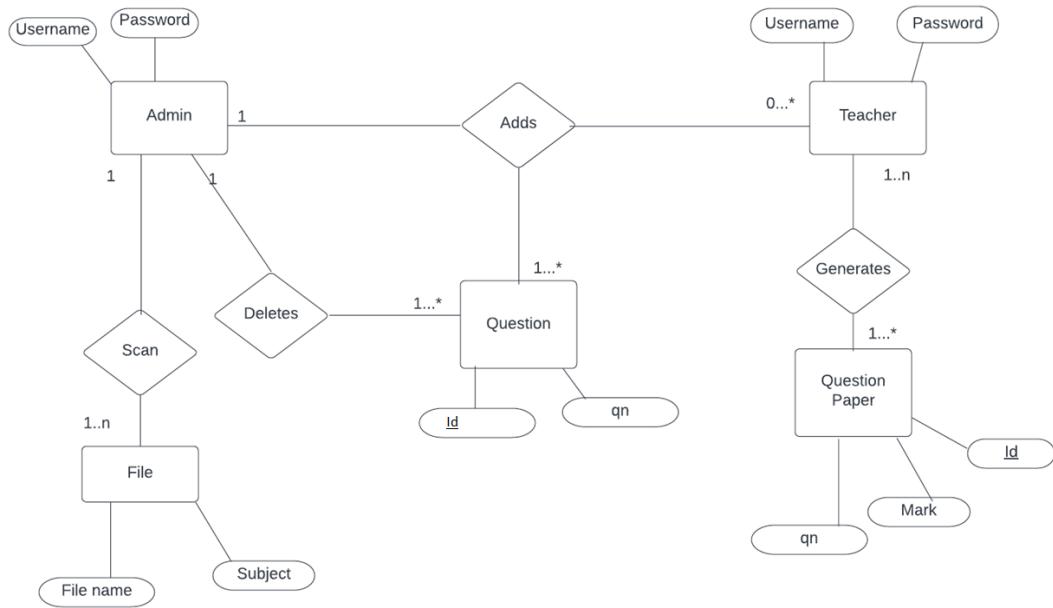


Figure 7 : ER Diagram

4.5 Activity Diagram

4.5.1. Activity Diagram for Admin

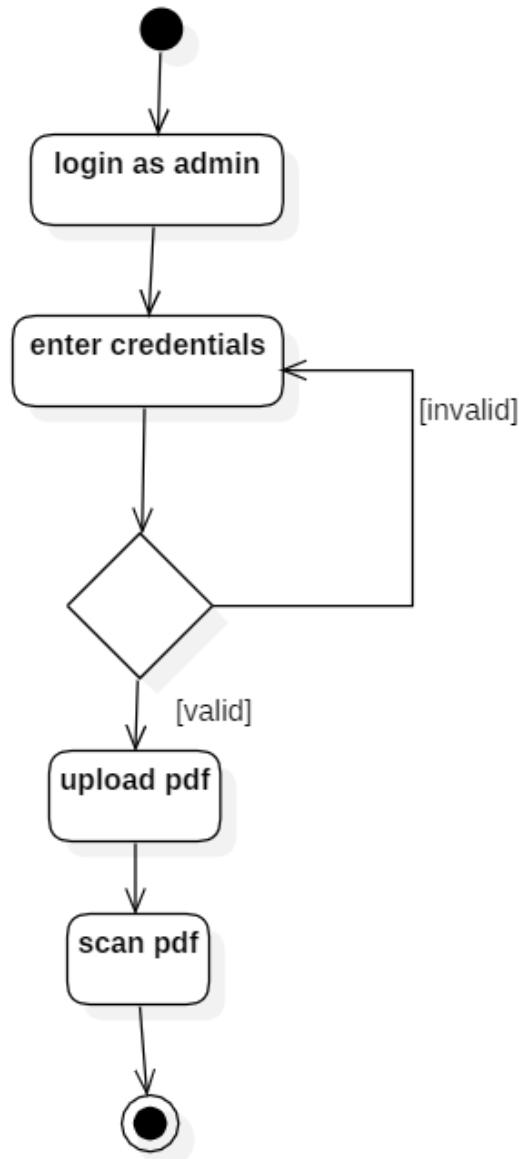


Figure 8 : Activity Diagram for Admin

4.5.2. Activity Diagram for Teacher

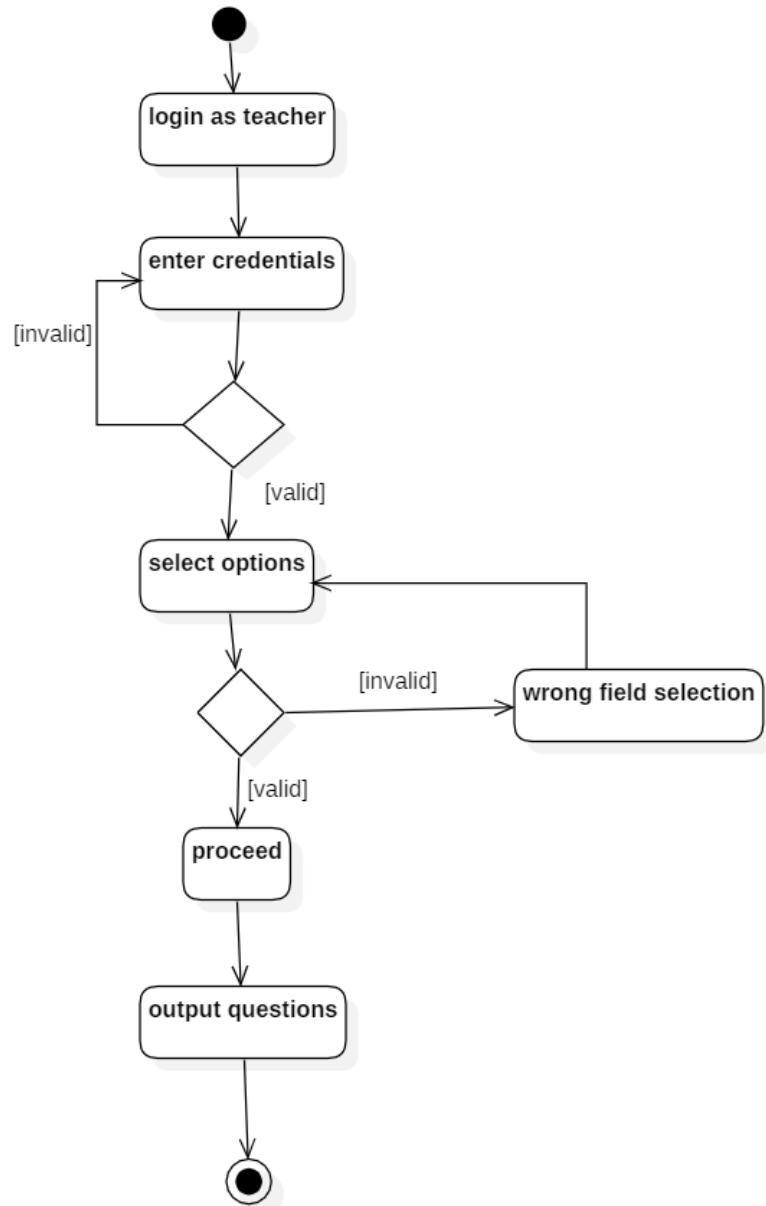


Figure 9 : Activity Diagram for Teacher

CHAPTER 5

METHODOLOGY

5. METHODOLOGY

5.1 Incremental Model

Incremental model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance. This model combines the elements of the waterfall model with the iterative philosophy of prototyping.

The product is decomposed into a number of components, each of which is designed and built separately (termed as builds). Each component is delivered to the client when it is complete. This allows partial utilization of the product and avoids a long development time. It also avoids a large initial capital outlay and subsequent long waiting period. This model of development also helps ease the traumatic effect of introducing a completely new system all at once.

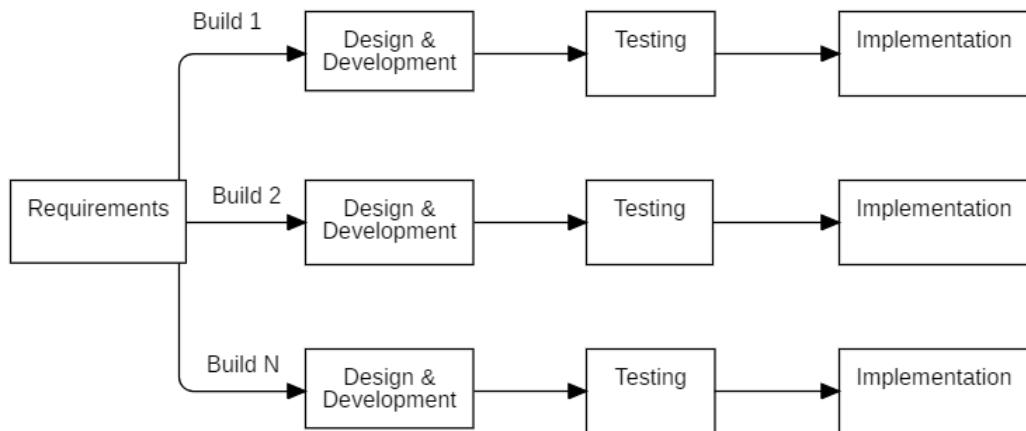


Figure 10 : Incremental Model

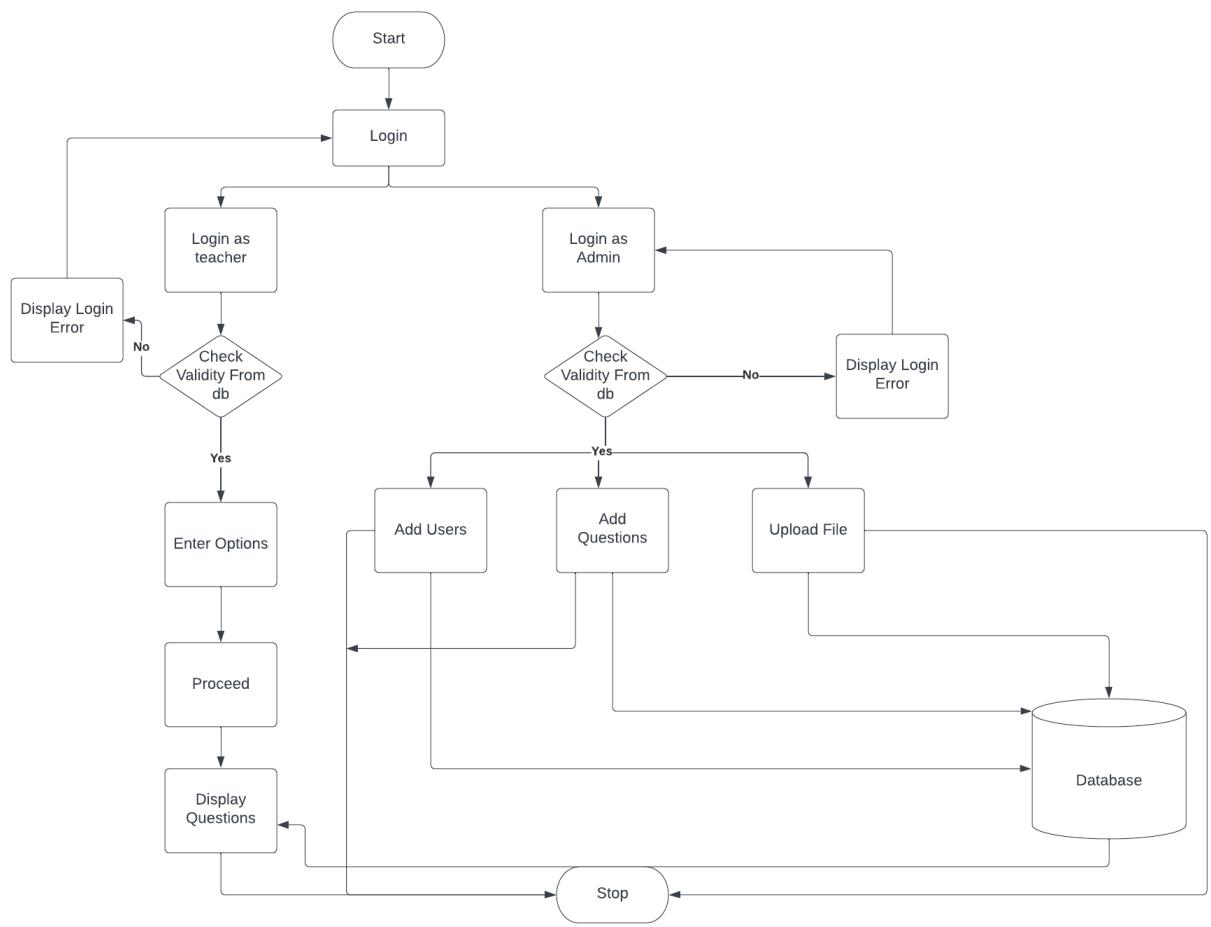


Figure 11 : System Architecture

The problem with the existing Zambia examination question paper generation method has led us to develop this project which ensures the fast and secured question paper generation. The approaches we have taken to generate the question paper are through the OCR scanning of the raw questions sets, setting them in order in database by extracting the useful information from scanned file and re-accessing the database to display the questions in exam question paper format.

We start the generation of our final output (which is question paper generation) by feeding the raw data into the system. The admin has right to update the file in pdf

format which is given to the OCR for scanning. On clicking the link to scan the pdf the OCR is set to process the pdf's raw data. The raw questions stored in the pdf should have special formatting to be fed into our system. For a given question, it should start with question numbering, the second field should have the question and the third field should define the marks of respective question. We have an OCR module called pdf miner to implement the text scanning functionality. It recognizes the unique characters from the pdf and extracts the texts. The texts are stored in text file. The text file is then used by the regular expression code to extract only meaningful texts. The regular expression defines the certain rules under which the text expressions or patterns are selected from the huge number of random data. When the extracted text pattern matches with the regular expression, the information including the question and its marks is saved into database table of that respective subject.

The question paper generation is easy and fast with some fields to fill on. The teacher can login into the system with the unique login credential provided by the admin. When a teacher is validated by the system, they are redirected into dashboard. In dashboard, teacher is provided with a form to input the full marks, pass marks, no of question weighing four and two marks, and the faculty should be chosen for generating question paper of respective subject. The form data from html template is received by Django in backend to work further on accessing and taking values from database based on the input given by the user. When teacher clicks on proceed button, s/he is redirected to the output page where the random questions are generated according to the number of weight they carry satisfying the full marks given. The full layout of the normal examination question paper is supposed to displayed which can be viewed by the teacher.

OCR:

So, how does OCR work? As you read these words on your computer screen, your eyes and brain are carrying out optical character recognition without you even noticing! Your eyes are recognizing the patterns of light and dark that make up the characters (letters, numbers, and things like punctuation marks) printed on the screen and your brain is using those to figure out. Computers can do this too, but it's really hard work for them. The first problem is that a computer has no eyes, so if you want it to read something like the page of a book, you have to present it with an image of that page generated with an optical scanner or the pdf format of that book. OCR program detects the characters by the pattern recognition and the feature recognition. The pattern recognition recognizes the character or group of characters by defining the pattern of the individual character in context of how it is made or drawn in graphical like way. The character differs from the individual to individual. It would be lot easier to recognize character if the patterns are same or produced by the same machine. The feature recognition method has datasets of fonts for a character to match the character with the pre-provided fonts.

In this project, we have implemented OCR for scanning the questions provided by the teachers which are the raw data for our database. The admin uploads the pdf format questions provided by the teachers to our system. After the file is uploaded the OCR scanning link can be clicked so that the admin can scan the questions. The link redirects to the OCR module where the processing of pdf can be done and the texts are extracted from the pdf. After OCR scanning, we use regular expressions to extract the useful information from the scanned texts to store the questions in database.

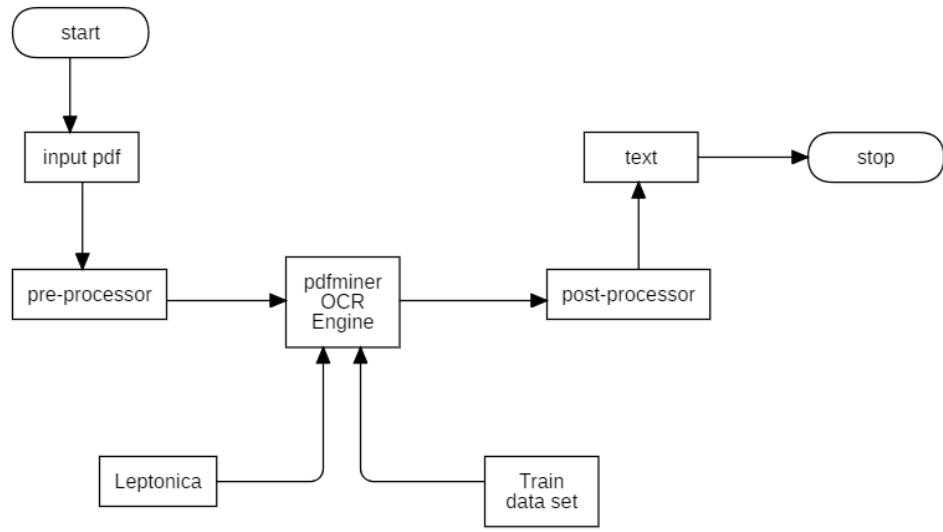


Figure 12 : Flowchart for OCR

General steps of OCR Working:

1. Input pdf:

The pdf containing the questions are fed into the OCR as inputs for the further processing and manipulation of the pdf's data.

2. Pre-Processor:

Pre-processing is done when the input is received in OCR system. Pre-Processing involves the following techniques:

Binarization:

In layman's terms Binarization means converting a colored image into an image which consists of only black and white pixels (Black pixel value=0 and White pixel value=255). As a basic rule, this can be done by fixing a threshold (normally threshold=127, as it is exactly half of the pixel range 0–255). If the pixel value is

greater than the threshold, it is considered as a white pixel, else considered as a black pixel.

So, the crucial part of binarization is determining the threshold. The threshold can be calculated using local minima and local maxima method.

$$C(i,j) = \frac{I_{max} - I_{min}}{I_{max} - I_{min} + E}$$

Where,

I_{max} = Maximum pixel value in the image

I_{min} = Minimum pixel value in the image,

E = Constant value

$C(i,j)$ is the threshold for a defined size of locality in the image (like a 10x10 size part). Using this strategy we'll have different threshold values for different parts of the image, depending on the surrounding lighting conditions but the transition is not that smooth.

Skewness:

While scanning a document, it might be slightly skewed (image aligned at a certain angle with horizontal) sometimes. While extracting the information from the scanned image, detecting & correcting the skew is crucial.

We can use project profile method for the skewness correction. In this method, we take binary image and project it horizontally (taking the sum of pixels along rows of the image matrix) to get a histogram of pixels along the height of the image. We rotate the image at various angles to find the skew angle and we can correct the skewness by rotating the image through an angle equal to skew angle in opposite direction.

Noise Removal:

The main objective of the Noise removal stage is to smoothen the image by removing small dots/patches which have high intensity than the rest of the image. Noise removal can be performed for both Colored and Binary images.

OCR engine:

OCR engine has Leptonica and group of datasets to manipulate the pre-processed data. Leptonica is a pedagogically-oriented open-source library containing software that is broadly useful for image processing and image analysis applications. Dataset is the collection of the predefined large amount of data for matching the data to given input data. NIST dataset is used in our OCR engine. NIST has 800000 characters images which are used to training our OCR algorithm and extract the meaningful data from it.

Post-Processor:

Post Processing involves data cleaning steps for documents that were digitized, such as pdf. One step in this process is the identification and correction of spelling and grammar errors generated due to the flaws in the OCR system.

Text:

Finally, our OCR process is completed. The result after the post-processing is the plain text which is saved in text file in our system.

Regular Expression:

A regular expression is a method used in programming for pattern matching. Regular expressions provide a flexible and concise means to match strings of text. Regular expressions are also known in short form as regex or re . Regular expressions can be incredibly powerful. Essentially, if the pattern can be defined, a regular expression can be created. A simple pattern might be something as simple as finding all situations where a sentence ends in "that" and is replaced with "which". The pattern could get more complex by doing the same replacement but only on the 3rd and 5th occurrence of a match. Or it could get even more complicated by using different sets of matching characters depending on the frequency and location of previous matching characters. Regular expression allows us to surf through a file containing the large amount of random words, any characters or alphanumeric characters in random order. When surfing, we can match the pattern given by our regular expression with the random group of words, characters or alphanumeric characters. We can extract the texts of our required pattern from the matched list and use them for our own purpose

As regular expression helps to find the pattern from our file. In this project we have attempted to match the question and mark pattern from the given raw question paper. The question paper consists of no of questions with question number and the marks assigned to the question. A question with a tail question can also be detected. We matched the questions with the respective marks and saved into the database of the respective subject.

Regular expressions are used for syntax highlighting systems, data validation and in search engines such as Google, to try to determine an algorithmic match to the query a user is asking. Data is everything in our digital world and the regular expression helps us to extract the data from the unmanaged order of data.

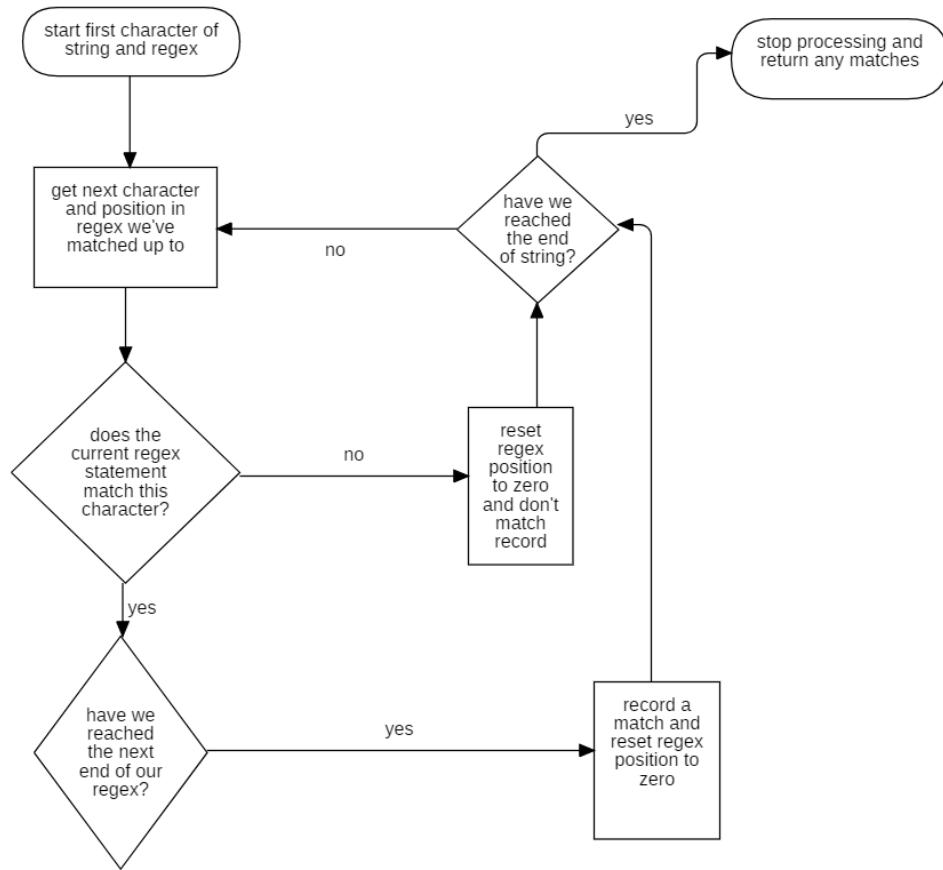


Figure 13 : Flowchart for Regular Expression

5.2 Tools Used

5.2.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. It can be used in many applications like front end web development, backend web developing, database management, data mining, artificial intelligence etc.

5.2.2 HTML

HTML stands for ‘Hyper Text Markup Language’. HTML, or Hypertext Markup Language, is a markup language for the web that defines the structure of web pages. It is one of the most basic building blocks of every website, so it's crucial to learn for the front end web development. It can integrate CSS and Javascript to make more interactive websites. HTML is the skeleton of the webpages.

5.2.3 CSS

CSS stands for ‘Cascading Style Sheet’. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files. CSS is the skin of the webpages.

5.2.4 Javascript

Javascript is the scripting language which helps to make the webpages interactive. Javascript is the cross platform language that contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements. Client-side JavaScript extends the core language by supplying objects to control a browser and its Document Object Model (DOM). For example, client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation. Basically, JavaScript is the circulatory, digestive, and respiratory systems that brings the structure and the skin to life.

5.2.5 Django

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support. Django can be used to build almost any type of website from content management systems, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, JSON, XML, etc.).

5.2.6 VS-Code

Visual Studio Code is the powerful and interactive code editor which is lightweight and supports various programming languages. It has powerful developer tooling, like IntelliSense code completion and debugging. VS Code helps with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more.

Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease. It is widely used in the programming world due to simple to learn environment for beginners and the advance toolkits for developer.

CHAPTER 6

SYSTEM TESTING

6. SYSTEM TESTING

Software testing is the act of examining the artifacts and the behavior of the software under test by validation and verification. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Testing is the important procedure which is most required before any product launch to avoid any further complications. We have implemented following testing procedures in our project.

6.1 Unit testing

We had tested every module and functions of our project while developing the project. We used unit testing for the following test cases.

Test case for user login

Table 2: Test case for user login

S.N	Test cases	Expected Output	Our Output
1	Enter valid user name and password	Login to our system and redirect to dashboard	Failed to login
2	Enter valid user name and password	Login to our system and redirect to dashboard	Same as Expected

Test case for Dashboard input

Table 3: Test case for Dashboard input

S.N	Test cases	Expected Output	Our Output
1	Enter the full marks, pass marks, questions amount and select valid subject	Redirect to output page with the generated question paper	Failed to generate questions
2	Enter the full marks, pass marks, questions amount and select valid subject	Redirect to output page with the generated question paper	same as expected
3	Enter the question amount exceeding the full marks given	Redirect to dashboard and display message full marks not met	Same as expected

4	Leave the field while filling the faculty and subject	Redirect to dashboard with message “Fill all the fields first”	Same as expected
---	---	--	------------------

Test case for OCR scanning

Table 4: Test case for OCR scanning

S.N	Test cases	Expected Output	Our Output
1	Enter the valid path for scanning document	Document scanned successfully and saved data to database	same as expected
2	Enter invalid path name for OCR feeding	No data saved to database	Same as expected

Test case for Regular Expression:

Table 5: Test case for regular expression

S.N	Test cases	Expected Output	Our Output
1	Provide text file with the valid pattern	Successfully matched pattern and store in respective database along with marks	Pattern was not recognized
2	Provide text file with the valid pattern	Successfully matched pattern and store in respective database along with marks	same as expected
3	Provide text file with the non-matching pattern	The questions not matched are not stored in database	Same as expected

6.2 Integration Testing

Integration testing is performed using the black box method. This method implies that a testing team interacts with an app and its units via the user interface – by clicking on buttons and links, scrolling, swiping, etc. They don't need to know how code works or consider the backend part of the components. We have integrated each module which were tested in unit testing. The overall system was tested after integration to identify the faults in the system and to debug the system.

CHAPTER 7

RESULT ANALYSIS

7. RESULT ANALYSIS

This system provides a very easy and simple way of generating question paper. Teacher generates question paper in very small time as per his/her need. Admin upload pdf file and extract question from pdf along with marks and store in database. Admin add new teacher in the system and provide username and password. This system gives the correct output as mentioned earlier in proposal. This system run successfully without any errors.

CHAPTER 8

CONCLUSION

8. CONCLUSION

With the completion of this project, the main aim of this project is achieved, that is easy way of generating and moderating question paper. This application integrates various functionalities to work together on providing the final output of our project. Our project reduces the traditional tedious job of writing question paper. Various resources like video tutorials, text tutorials and internet learning materials were used during this project.

8.1 Further Enhancement

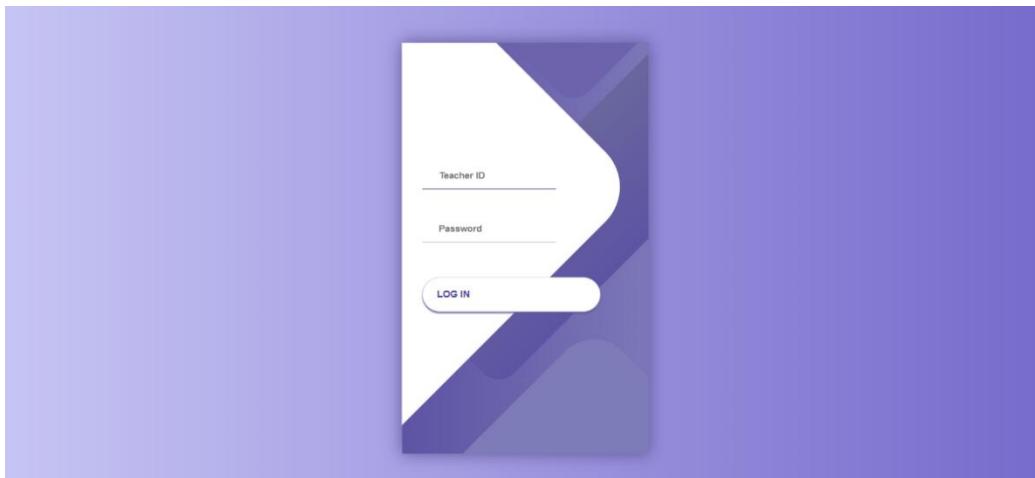
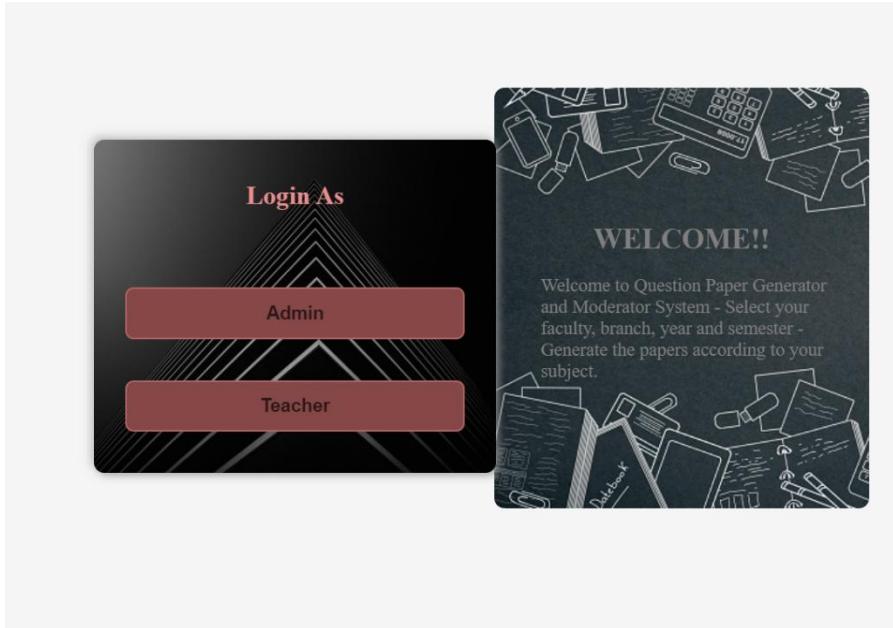
To further enhance the functionality of our system following features are to be integrated into our system:

- Student login for providing the generated random question papers to each student
- To save the generated question papers in question bank

References

- [1] M. Fatangare, "“Android Based Exam Paper Generator”,” “*Android Based Exam Paper Generator*”, 2018.
- [2] A. Khairnar, "“Automatic Question Paper”,” “*Automatic Question Paper*”, 2018.
- [3] M. Joisher, "“AutomaticQuestion PaperGeneration”,” “*AutomaticQuestion PaperGeneration*”, vol. 4, 2015.
- [4] P. Pisat, "“Question Paper Generator”,” “*Question Paper Generator*”, 2017.
- [5] V. K. Purohit, "“Design of Adaptive Question Bank”,” “*Design of Adaptive Question Bank*”, 2012.
- [6] S. Mori, "“Historical Review of OCR Research and Development”,” 1992.
- [7] A. A. Chaudhary, "“OCR of Bangla Character Using Neural Network”,” 2002.
- [8] Software Requirements Specification for project iTest.
- [9] The International Journal of Multi-Disciplinary Research on Examination System, 2014.
- [10] International Research Journal of Engineering and Technology (IRJET), 2019.
- [11] Fonseca Cacho, Jorge Ramon, "Improving OCR Post Processing with Machine Learning Tools", 2019.
- [12] "www.youtube.com/Python Backend Web Development Course (with Django), free code camp channel," [Online].
- [13] "www.youtube.com/regular expression with Corey Schafer," [Online].
- [14] "https://docs.djangoproject.com," [Online].
- [15] C. Wood, "Explaininthatstuff.com |OCR," [Online].

APPENDICES



The screenshot shows a web-based administration interface. On the left, a sidebar menu includes 'Dashboard' (selected), 'Aqg' (with sub-options: Dbmss, Ess, Files admins, Oss), and 'Authentication and Authorization' (Groups, Users). The main content area has two sections: 'Aqg' (Dbmss, Ess, Files admins, Oss) and 'Authentication and Authorization' (Groups, Users). A 'Recent actions' sidebar on the right lists five entries, each with a timestamp: 'osTyp e.pdf' (22 hours, 49 minutes ago), 'Changed Pdf file.', 'osTyp e.pdf' (22 hours, 50 minutes ago), 'Changed Pdf file.', 'Os object [72]' (22 hours, 50 minutes ago), 'Os object [73]' (22 hours, 50 minutes ago), 'Os object [74]' (22 hours, 50 minutes ago), and 'Os object [75]' (22 hours, 50 minutes ago).

The screenshot shows a form titled 'Question Paper Generator and Moderator'. The form fields include:
Full marks: 20
Pass marks: 8
No_four: 4
No_two: 2
Faculty: BCT
Year: 3
Part: II
Subjects: DBMS
A 'PROCEED' button is at the bottom.

HIMALAYA COLLEGE OF ENGINEERING

Chaysal-9, Lalitpur

Students are requested to give answers in their own words!!

Subject: DBMS

Full Marks: 20

Pass Marks:8

Group 'A'

(marks = 2)

- 1) What is Database Management System?
- 2) Why physical data independence is important in data modeling?

Group 'B'

(marks = 4)

- 1) Explain the distinction between static and dynamic hashing with approximate example.
- 2) Explain 3NF and BCNF.
- 3) Define transaction and explain the various states of transaction.
- 4) Why is database system more preferred over file system? Explain briefly.

THANK YOU !