**AUTOMATIC QUESTION PAPER GENERATOR SYSTEM**

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING**

BY

**Aanchal Jawere (EN16CS301002)**

**Anchal Soni (EN16EL301041)**

Under the Guidance of

**Prof. Nitesh Tejra**



**Department of Computer Science & Engineering**

**Faculty of Engineering**

**MEDI-CAPS UNIVERSITY, INDORE- 453331**

**May, 2020**

**AUTOMATIC QUESTION PAPER GENERATOR SYSTEM**

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING**

BY

**Aanchal Jawere (EN16CS301002)**

**Anchal Soni (EN16EL301041)**

Under the Guidance of

**Prof. Nitesh Tejra**



**Department of Computer Science & Engineering**

**Faculty of Engineering**

**MEDI-CAPS UNIVERSITY, INDORE- 453331**

**May, 2020**

**Report Approval**

The project work **“Automatic Question Paper Generator System”** is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

Internal Examiner

Name:

Designation

Affiliation

External Examiner

Name:

Designation

Affiliation

**Declaration**

We hereby declare that the project entitled **“Automatic Question Paper Generator System”**  submitted in partial fulfillment for the award of the degree of Bachelor of Technology in ‘Computer Science and Engineering’ completed under the supervision of **Prof. Sachin Solanki, Assistant Professor, Department of Computer Science and Engineering,** Faculty of Engineering, Medi-Caps University Indore is an authentic work.

Further, we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

**Aanchal Jawere(EN16CS301002)**

**Anchal Soni(EN16EL301041)**

**Date-14/05/2020**

**Certificate**

I, **Prof. Nitesh Tejra** certify that the project entitled “**Automatic Question Paper Generator System”** submitted in partial fulfillment for the award of the degree of Bachelor of Technology by **Aanchal Jawere (EN16CS301002)** and **Anchal Soni (EN16EL301041)** is the record carried out by them under my guidance and that the work has not formed the basis of award of any other degree elsewhere.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Prof. Nitesh Tejra**

Computer Science Engineering

Medi-Caps University, Indore

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Suresh Jain

Head of the Department

Computer Science & Engineering

Medi-Caps University, Indore

**Acknowledgement**

We would like to express our deepest gratitude to Honorable Chancellor, **Shri R C Mittal**, who has provided us with every facility to successfully carry out this project, and our profound indebtedness to **Prof. (Dr.) Sunil K Somani**, Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted our morale. We also thank **Prof. (Dr.) D K Panda**, Dean, Faculty of Engineering, Medi-Caps University, for giving us a chance to work on this project. We would also like to thank our Head of the Department **Prof. (Dr.) Suresh Jain** for his continuous encouragement for betterment of the project.

We express our heartfelt gratitude to our Internal Guide, **Prof. Nitesh Tejra**, Assistant Professor, Department of Computer Science and Engineering, Medi-Caps University, without whose continuous help and support, this project would ever have reached its completion.

It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

**Aanchal Jawere(EN16CS301002),**

**Anchal Soni (EN16EL301041)**

B.Tech. IV Year

Department of Computer Science and Engineering

Faculty of Engineering

Medi-Caps University, Indore

**Abstract**

Curiosity is the fuel for discoveries and learning, and we humans are always curious about learning novel things. Asking different types of questions is a tendency of a good learner and so the students ask a variety of questions from their teachers. Similarly to examine the students, the teachers also ask questions from the students and also evaluate their performances. Because of their inconsistent minds in certain situations, sometimes humans are not very skilled in asking good questions and because of that we have come with the idea of a system with the help of which we would be able to generate the questions from a text automatically. The system is known as the Automated Question Paper Generator System, which is a fast and secure system and which can also generate questions in a random way. The document or pdf file or simply text can be provided as an input to this proposed system. NLP technology is used for preparing the proposed system. Traditional method of preparation of question papers can be ineffective at times owing to bias, repetition, security concerns and also tiring and challenging tasks for educators. Also due to shortage of teaching staff in any institute, paper creation becomes a tough task. Our application aims to provide fast operations, data storage and high security for all its tasks and can be helpful to many educational institutes and NGO based institutes.

**Keywords:** Question Paper Generation, stop words, NLP, Key phrase Extraction, Triplets Generation, Sentence Segmentation, Text Lemmatization, word tokenization, dependency parsing.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S No.** | **Particulars** | **Page No.** |
|  | Report Approval | ii |
|  | Declaration | iii |
|  | Certificate | iv |
|  | Acknowledgement | v |
|  | Abstract | vi |
|  | Table of Contents | vii |
|  | List of figures | viii |
| Chapter 1 | Introduction | 1 |
|  | 1.1 Introduction | 2 |
|  | 1.2 Literature Review | 2 |
|  | 1.3 Objectives | 3 |
|  | 1.4 Significance | 3 |
|  | 1.5 Problems in Existing System | 5 |
|  | 1.6 Research Design | 6 |
|  | 1.7 Source of Data | 7 |
| Chapter 2 | System Requirement Analysis | 8 |
|  | 2.2 Platform Specifications | 9 |
|  | 2.3 Procedures adopted. | 9 |
| Chapter 3 | System Analysis | 12 |
|  | 3.1 Information Flow Representation | 13 |
|  | 3.1.1 Use CaseDiagram | 13 |
|  | 3.1.2 Activity Diagram | 16 |
|  | 3.1.3 Sequence Diagram | 17 |
|  | 3.1.4 Class Diagram | 18 |
| Chapter 4 | Design | 20 |
|  | 4.1 Implementation | 21 |
|  | 4.2 Algorithm Design for Operation | 22 |
| Chapter 5 | Results and Discussions | 23 |
|  | 5.1 Testing Objective | 24 |
|  | 5.2 Test Cases | 24 |
|  | 5.3 Testing Principles | 26 |
| Chapter 6 | Conclusion | 28 |
| Chapter 7 | Future Scope | 30 |
| Chapter 8 | References | 32 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure** | **Particulars** | **Page no.** |
| 1.1 | Working Model | 07 |
| 3.1 | Use Case Diagram | 16 |
| 3.2 | Activity Diagram | 17 |
| 3.3 | Sequence Diagram | 18 |
| 3.4 | Class Diagram | 19 |
| 4.1 | Working Algorithm (Main Function) | 22 |
| 4.2 | Working Algorithm (iqpgFunction) | 22 |
| 4.3 | Input | 25 |
| 4.4 | Output | 25 |

**Chapter-1**

**Introduction**

**1.1 Introduction**

Traditionally question papers were generated manually. Preparing any exam paper is a very challenging task for the educators because they have to check whether there is any repetition in the pattern or not, and other than that security is one of the major concerns for them , also due to lack of teaching staff in any institute, creation of paper is not at all an easy task. So, here we are proposing an Intelligent Automatic Question Paper Generator System which provides storage of the data, fast operations, and high security for all its tasks. It can be helpful to many educational as well as NGO institutes. Tasks performed by this system are automated and because of that storage space, security is not a concern anymore. Proposed system works upon Natural Language Processing and is fast due to computer based automation, streamlined, randomized and unbiased, secure and generates unique questions and overcomes the problems of Human based paper generation.

**1.2 Literature Review**

Automatic question paper generation has been an area of ​​interest for many researchers and a lot of research has been done for the question paper generation system. Researchers are very interested in this field and they are showing their interest too. The reason for the interest is basically the time taken by the professor to write the thesis. And every researcher has undergone the university processes. AUTOQUEST was one of the first automated QG systems proposed to support novices to learn English. Kunichika et al, who proposed a method for generating questions based on both syntactic and semantic information (space, time, and agent) so that he could generate more types of questions (where, when and who).[4] Most recently, Mostov and Chen proposed an approach for generating deep questions based on a situation model. It can generate questions about what, how and why. Several approaches to automatic QG with multiple selection of reading materials have been proposed. Coniam deleted every nth-word in the text to be a test item, and distractors were identified by choosing the same part of speech and similar word frequency to a tagged corpus. Mitkov and Ha extracted key terms that are noun phrases with a frequency that exceeds a certain threshold.

**1.3 Objective**

In today’s age, education is the most important way of achieving success. When we discuss education, it is imperative to mention tests and examinations. Examinations prepare students in their quest for knowledge. So, having a proper examination paper and format is quite necessary. Now the traditional method of generating question paper has been manual. In this method certain officials chalk out the question paper. But this method can be ineffective at times owing to bias, repetition and security concerns. We have proposed an Automated process of Question Paper Generation which is fast, streamlined, randomized and secure. Every task performed by this system is automated so that storage space, bias and security is not a concern anymore. Furthermore, we have proposed a new algorithm which ensures total randomization of questions and avoids repetitions.

**1.4 Significance**

In today’s ambition, examination plays an important role in ensuring students’ academic progress and the era of information technology is now being replaced by the productive application of the technology. In the "information society" the production of utility from knowledge is important for the development of society. For various examinations conducted in any academic course in one year, teachers are required to make a variety of question papers as per the guidelines and evaluation requirements of the autonomous college. It is a tough task for the teachers to cover all the characteristics of the course results and avoid duplication of questions in successful exams. Many times, this entire element may degrade the standard of the question paper. Automated creation of exam paper provides a platform to create a streamlined exam paper and at the same time automation is easy to incorporate many elements that determine the quality of the question paper.An efficient question paper is generally suitable for ordinary students, but it also engages the demanding item for clever students. Thus automatically creating a question paper from a description entered by a teacher uses a word-labeled question bank that is currently required by the hour. Here we are implementing a system which accordingly creates the question paper from this semantically labelled question bank.

It is necessary to categorize queries as different class of queries need different methods for automatic generation of Question-type

* **Yes/No Questions:** also known as polar questions, whose expected answer is either yes or no.
* **Wh-Questions:** Wh-questions use interrogative words, such as why, when, who, where, what, which, etc,… to request information.
* **Fill-in-the-blanks Questions**: also known as cloze questions. Fill-in-the-blank questions could be a sentence with one or a lot of blanks in it with four alternatives to fill those blanks.

Quality of questions generated for any exam paper through Automatic Question Paper Generator would be :

* Appropriate : A good question is appropriate. It focuses on memorizing only the content included in your lesson and aligns well with the overall learning objectives.
* Concise : A good question is usually concise and crisp. This leaves out any unnecessary information that students have to spend time understanding correctly.
* Understandable : A good question is formulated in clear, easily understandable language without any ambiguity. Students should understand what the question requires, even when they do not know the answer.
* Purposeful : A question without a specific purpose has no value. The purpose helps evaluate the question on the basis of some defined benchmarks. A good question can seek both innate and specific knowledge.
* Guiding But Not Leading : A good question guides the students to understand the concept in the image.But at the same time, it does not give them specific answers.
* Stimulates Thinking : A good question requires learners to think and remember through the concepts taught.It does not protect them by asking explicitly.
* Single-Dimensional : A good question focuses on one dimension at a time. If there are multiple ideas to evaluate, it is better to split them into multiple questions.[2]

Difference Between Manual Process And Automated Process Of Generating Question Paper :

|  |  |
| --- | --- |
| **Manual Paper Generation** | **Automated Paper Generation** |
| It is a human process. | It is an automated process. |
| In this process repetition in paper may occur. | It is a random as well as unbiased process |
| Less Security. | Higher security. |
| It is a slow process. | It is a faster process. |
| It has less varieties of questions. | It has more varieties of questions. |

**1.5 Problem in Existing System**

The main task is to create questions automatically based on the given text, ie. sentences or groups of sentences.The basic input is a positive or declarative sentence or group of sentences. The output should be different kinds of questions depending on the type of sentences. Basically we can say that the input would simply be the text whereas output would be a number of questions. The basic goal of this system is to offer a solution to the problem of first breaking down the paragraph and then turning it into questions.[3]

There are different types of problems that exist in the current manual system of question paper generation. These problems are:-

* Low security as paper is not secured using any mechanism.
* Patterns or repetitions may occur in paper.
* Slow as human labor is involved.
* Less variety of different types of questions.

Thus there are many problems associated with manual human based question paper generation, therefore to overcome this we came up with the ‘Automatic Intelligent Question Paper Generator’.

**1.6 Research Design**

The Automatic Question Paper Generator System is developed using the Python programming language. In a fully functional system, there is a repository of syllabus, questions and pattern of question papers. It takes a simple text , a document or a pdf file as an input and provides a list of questions as an output. Process that is followed for generating questions automatically has the following steps :

**Step 1: Sentence Segmentation:** The very first step in the pipeline is to break the text in the document into separate sentences. Now that we’ve divided our document into sentences, we can process them one at a time.

**Step 2: Word Tokenization:** The second step in our pipeline is to break this sentence into tokens or separate words, which is called tokenization. In tokenization, we’ll just split apart words whenever there’s a space between them. And we’ll also treat punctuation marks as separate tokens since punctuation also has meaning.

**Step 3: Predicting Parts of Speech for Each Token:** Next, we’ll look at each token and try to guess its parts of speech whether it is a noun, a verb, an adjective. Knowing the role of each word in the sentence will help us start to figure out what the sentence is talking about.

**Step 4: Text Lemmatization & Identifying Stop Words:** Lemmatization means figuring out the most basic form or lemma of each word in the sentence. Stop words are usually identified by just checking a list of known stop words.

**Step 5: Dependency Parsing:**  The next step is to figure out how all the words in a sentence relate to each other. This is called dependency parsing. So far, we’ve treated every word in our sentence as a separate entity. But sometimes it makes more sense to group together the words that represent a single idea or thing. We can use the information from the dependency parse tree to automatically group together words that are all talking about the same thing.

**Step 6: Question Creation:** At this point, we already have a useful representation of our sentence. We know the parts of speech for each word, how the words relate to each other and which words are talking about named entities. Now we’ll create ‘Wh’ questions from this information.

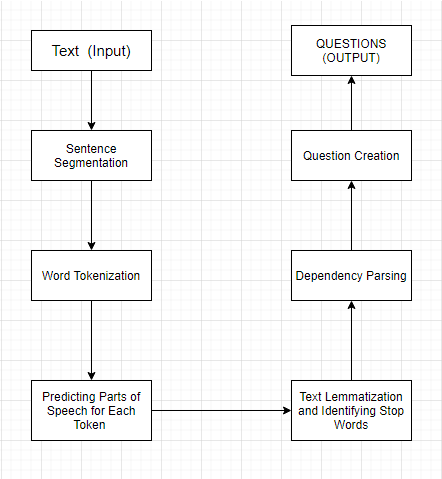


Fig 1.1 Working Model

**1.7 Source of Data**

[1] <https://www.researchgate.net/publication/276886742_Automatic_Question_Generation_from_Text>

[2] <https://elearningindustry.com/characteristics-of-a-good-question-7>

[3]<https://www.ijcaonline.org/archives/volume166/number10/gangar-2017-ijca-914138.pdf>

[4] <https://www.computer.org/csdl/journal/lt/2012/03/tlt2012030251/13rRUwfZC2p>

**Chapter-2**

**System Requirement Analysis**

**2.1 Experimental Set-up**

|  |  |
| --- | --- |
| **Tools/Technologies** | **Description** |
| Operating System | Any modern OS can be chosen for development of Intelligent Question Paper Generator. |
| Python | To implement the project python is chosen as a platform for its more interactive community and lots of libraries. |
| Anaconda IDE | Anaconda IDE is to be used as the main development tool to develop this project. |
| NLP | Natural Language Processing (NLP) is the sub-field of Computer Science especially Artificial Intelligence (AI) that is concerned about enabling computers to understand and process human language. |

**2.2 Platform Specification:**

* **Software Requirements:**

1. Python 3.0 (With required libraries)
2. MySQL Database
3. Pycharm IDE
4. Html, CSS, JavaScript Editor
5. Web browser (Any Modern)

* **Hardware Requirements:**

1. Supported device types: Thin client (minimum).
2. For developing software any standard computer is sufficient.
3. Database for storing questions & student data.
4. Server.

**2.3 Procedures Adopted:**

The method which we have used for question generation is ‘Syntactic and Semantic Analysis using Natural Language Processing’. Natural Language Processing or NLP is the sub-field of AI that is focused on enabling computers to understand and process human languages. We have used pipelining in NLP, doing anything complicated in machine learning usually means building a pipeline. The idea is to break up input text into very small pieces and then use machine learning to solve each smaller piece separately. Then by logically chaining them together to generate questions.

The steps followed are:

**Step 1: Sentence Segmentation:**

The first step in the pipeline is to break the text apart into separate sentences. Now that we’ve split our document into sentences, we can process them one at a time.

**Step 2: Word Tokenization:**

The next step in our pipeline is to break this sentence into separate words or tokens. This is called tokenization. Tokenization is easy to do in English. We’ll just split apart words whenever there’s a space between them. And we’ll also treat punctuation marks as separate tokens since punctuation also has meaning.

**Step 3: Predicting Parts of Speech for Each Token:**

Next, we’ll look at each token and try to guess it's part of speech whether it is a noun, a verb, an adjective and so on. Knowing the role of each word in the sentence will help us start to figure out what the sentence is talking about.

**Step 4: Text Lemmatization & Identifying Stop Words:**

Lemmatization means figuring out the most basic form or lemma of each word in the sentence. Stop words are usually identified by just checking a list of known stop words.

**Step 5: Dependency Parsing:**

The next step is to figure out how all the words in a sentence relate to each other. This is called dependency parsing. So far, we’ve treated every word in our sentence as a separate entity. But sometimes it makes more sense to group together the words that represent a single idea or thing. We can use the information from the dependency parse tree to automatically group together words that are all talking about the same thing.

**Step 6: Question Creation:**

At this point, we already have a useful representation of our sentence. We know the parts of speech for each word, how the words relate to each other and which words are talking about named entities. Now we’ll create ‘Wh’ questions from this information.

Question creation is even more difficult than sentence parsing. Recent advances in machine learning have resulted in new approaches that are more accurate, but it isn’t perfect yet.

**Chapter-3**

**System Analysis**

**3.1 Information Flow Representation**

Information flow representation is a UML behavior diagram which shows the exchange of information between system entities at some high levels of abstraction. Information flows may be useful to describe circulation of information through a system by representing aspects of models not yet fully specified or with fewer details. Information flows do not specify the nature of the information, mechanisms by which it is conveyed, sequences of exchange, or any control conditions. Information items can be used to represent the information that flows through a system along the information flows before details of their realization have been designed.

**3.1.1 Use Case Diagram :**

The purpose of a use case diagram is to capture the dynamic aspect of a system. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified. We are using, use case diagrams to model the outside view. Use case diagrams are representing high level requirement analysis of a system. Requirements of a system are analyzed; the functionalities are captured in use cases.

**1) Login**

* Name: Login
* Brief description: To access the platform user must register and after registration user can access the platform.
* Flow of Events:
  + The user first registers himself.
  + Then users can login to access the platform.
* Pre-Conditions: The user must register on the platform.
* Post-Conditions: After Login students can give test and view reports and teachers can enter text and generate questions and can view student performance.

**2) Registration**

* Name: Registration
* Brief description: There are two types of user that can access the platform : teacher and student. To access the platform the user must register.
* Flow of Events: Eligible students can register and teachers can also register.
* Pre-Conditions: None.
* Post-Conditions: After registration users can access the platform.

**3) Set test**

* Name: Set test.
* Brief description: The teacher can enter text and generate questions and save them to the database and then select questions from the database and form a test.
* Flow of Events:

o The teacher must register and then login to the access platform.

o The teacher then enters text or enters file location for question generation.

o The questions that are generated are saved into the database.

o The teacher then selects questions from the database and set tests.

* Pre-Conditions: The teacher first registers and then login into the platform. Then generate questions after entering text.
* Post-Conditions: After setting text now students can give text.

**4) Give test**

* Name: Give test
* Brief description: The student can give a test.
* Flow of Events:

o The student must register and then login to the access platform.

o The student then selects the subject and can give text.

* Pre-Conditions: The student must register and login. Then students can give tests.
* Post-Conditions: None

**5) View profile**

* Name: View profile
* Brief description: The user can view their profile. The student can also view their profile and evaluate their result.
* Flow of Events:

o The user must register and then login.

o Then the user can view their result.

* Pre-Conditions: the user must register and then login
* Post-Conditions: the user can view their profile.

**6) View result**

* Name: View result.
* Brief description: the user can register and then login. The student must give a test to view their result. The teacher can view the results of different students.
* Flow of Events:

o The user first register and then login themselves.

o The student then gives a test.

o Then students can view their result.

* Pre-Conditions: The student must register and then give a test.
* Post-Conditions: None

**7) View student profile**

Name: View student profile

* Brief description: the teacher can view the profile of different students and can view the report of different students. The student can view their profile.
* Flow of Events:

o The user first register then login themselves.

o The student can view their profile.

o The teacher can view their profile as well as the profile of different students.

* Pre-Conditions: The student must register.
* Post-Conditions: The user can view the profile.

**8) Logout**

* Name: Logout.
* Brief description: when user are done with work then it will log user out
* Flow of Events:

o The user must register then login and then access the platform.

o The user can logout.

* Pre-Conditions: The user must register and login into platform
* Post-Conditions: This will logout users from the platform.

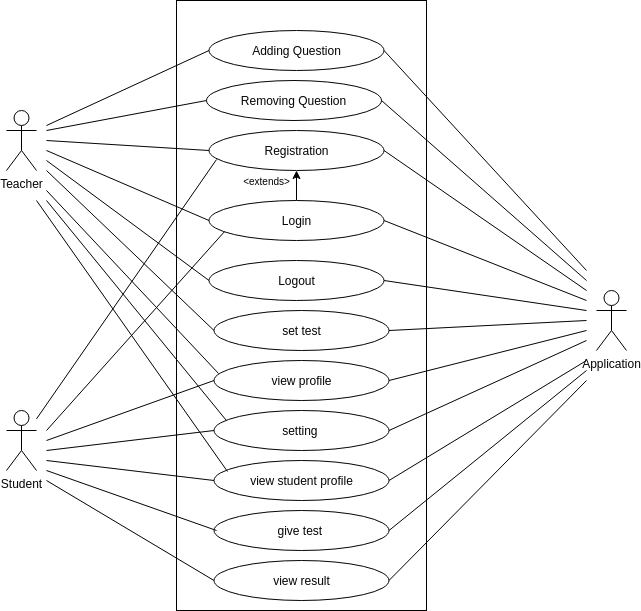


Fig 3.1 Use Case Diagram

**3.1.2 Activity Diagram**

Activity diagram is another important diagram in UML for describing the dynamic aspects of the system.  Activity diagram represents the flow from one activity to another activity. The activity can be described as an operation of the system. The start and ending of the diagram is represented through respective symbols.

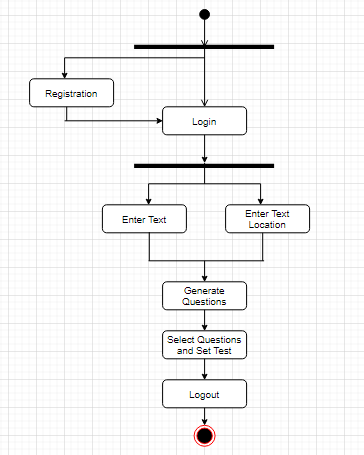


Fig 3.2 Activity Diagram

**3.1.3Sequence Diagram**

A Sequence diagram shows interaction among objects in terms of exchanges of messages arranged in time.

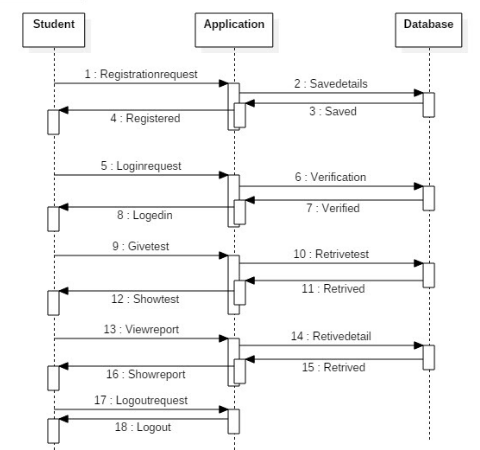


Fig 3.3 Sequence Diagram

**3.1.4Class Diagram**

A Class diagram is a UML diagram that describes the static structure of a system. It describes a

System’s classes, their attributes, operations, and the relationships among objects. We have two class one applications and a database in our program.

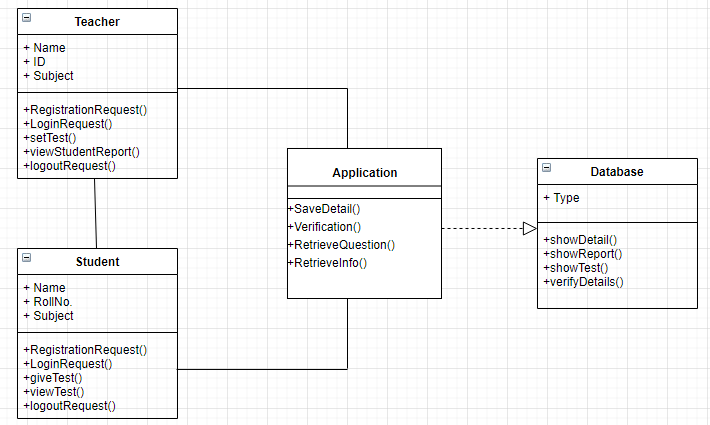


Fig 3.4 Class Diagram

**Chapter-4**

**Design**

**4.1 Implementation**

The project is implemented in the programming Language Python. Python is chosen as a platform for its more interactive community and lots of libraries . Python is a general purpose, dynamic, high level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.Python is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development.

Compared to other programming languages,python supports a huge number of libraries . The libraries we used in our application development is shown below:

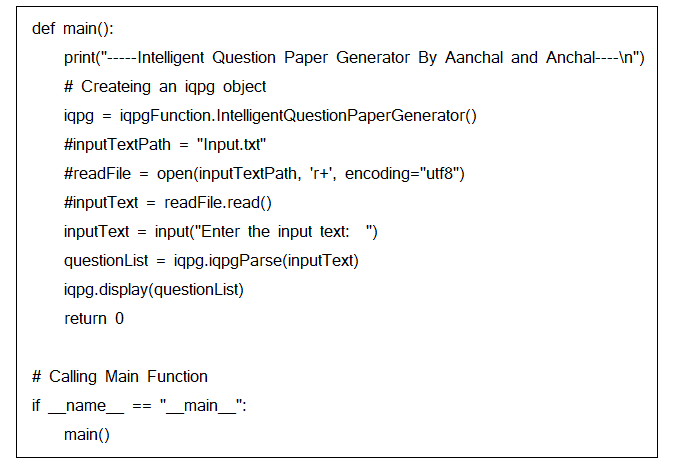
1. Spacy
2. NLTK

|  |  |
| --- | --- |
| **Tools/Technologies** | **Description** |
| Operating System | Any modern OS can be chosen for development of Intelligent Question Paper Generator. |
| Python | To implement the project python is chosen as a platform for its more interactive community and lots of libraries. |
| Anaconda IDE | Anaconda IDE is to be used as the main development tool to develop this project. |
| NLP | Natural Language Processing (NLP) is the sub-field of Computer Science especially Artificial Intelligence (AI) that is concerned about enabling computers to understand and process human language. |

Design goals are important properties of the system to be optimized, and which may affect the overall design of the system.Note an important difference between design goals and requirements: Requirements include specific values that must be met in order for the product to be acceptable to the client, whereas design goals are properties that the designers strive to make "as good as possible", without specific criteria for acceptability.

**4.2 Algorithm Design for Operations**

Proofofworkingalgorithm ofAutomatedQuestionPaper Generator System:

Fig 4.1 Working Algorithm (Main Function)

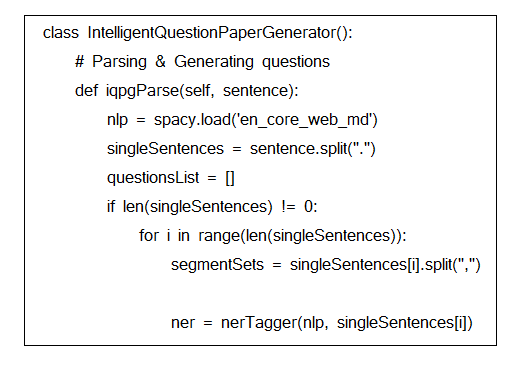


Fig 4.2 Working Algorithm (iqpgFunction)

**Chapter-5**

**Results and Discussion**

**5.1 Testing Objective**

Software Testing has different goals and objectives.The major objectives of Software testing are:

* Finding defects which may get created by the programmer while developing the software.
* Gaining confidence in and providing information about the level of quality.
* To prevent defects.
* To make sure that the end result meets the business and user requirements.
* To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
* To gain the confidence of the customers by providing them a quality product.

Software testing helps in finalizing the software application or product against business and user requirements. It is very important to have good test coverage in order to test the software application completely and make it sure that it’s performing well and as per the specifications.

While determining the test coverage the test cases should be designed well with maximum possibilities of finding the errors or bugs. The test cases should be very effective. This objective can be measured by the number of defects reported per test cases. Higher the number of the defects reported the more effective are the test cases.

**5.2 Test Cases:**

Input and Output can be seen in the snapshot below. The following snapshot is of generating questions for a given piece of text.

INPUT:

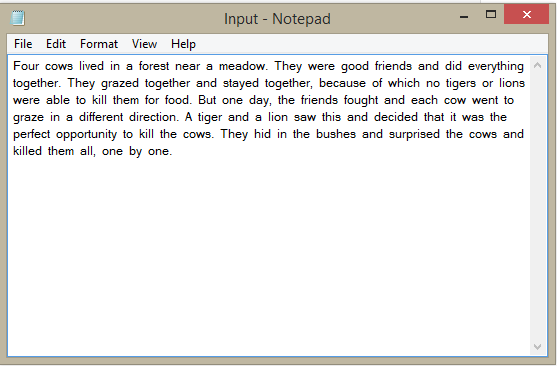
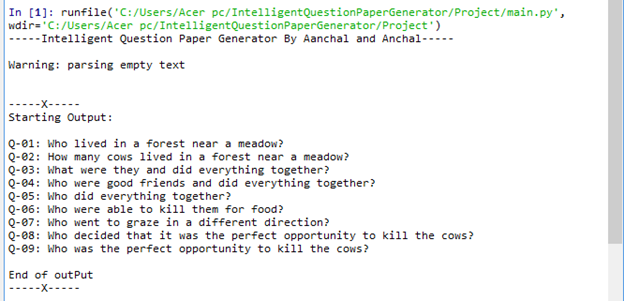


Fig 5.1 Input

OUTPUT:

Fig 5.2 Output

**5.3 Testing Principles**

Software testing is a process of executing a program with the aim of finding the error. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software. There are seven principles in software testing:

1. Testing shows presence of defects
2. Exhaustive testing is not possible
3. Early testing
4. Defect clustering
5. Pesticide paradox
6. Testing is context dependent
7. Absence of errors fallacy

* Testing shows presence of defects: The goal of software testing is to make the software fail. Software testing reduces the presence of defects. Software testing talks about the presence of defects and doesn’t talk about the absence of defects. Software testing can ensure that defects are present but it can not prove that software is defects free. Even multiple testing can never ensure that software is 100% bug-free. Testing can reduce the number of defects but not remove all defects.
* Exhaustive testing is not possible: It is the process of testing the functionality of a software in all possible inputs (valid or invalid) and pre-conditions is known as exhaustive testing. Exhaustive testing is impossible means the software can never test at every test case. It can test only some test cases and assume that software is correct and it will produce the correct output in every test case. If the software will test every test case then it will take more cost, effort, etc. and which is impractical.
* Early Testing: To find the defect in the software, early test activity shall be started. The defect detected in early phases of SDLC will be very less expensive. For better performance of software, software testing will start at initial phase i.e. testing will perform at the requirement analysis phase.
* Defect clustering: In a project, a small number of the module can contain most of the defects. Pareto Principle to software testing states that 80% of software defects comes from 20% of modules.
* Pesticide paradox: Repeating the same test cases again and again will not find new bugs. So it is necessary to review the test cases and add or update test cases to find new bugs.
* Testing is context dependent: Testing approach depends on context of software developed. Different types of software need to perform different types of testing. For example, The testing of the e-commerce site is different from the testing of the Android application.
* Absence of errors fallacy: If a built software is 99% bug-free but it does not follow the user requirement then it is unusable. It is not only necessary that software is 99% bug-free but it is also mandatory to fulfill all the customer requirements.

**Chapter-6**

**Conclusion**

The application will support two types of user: Teacher and Student.

* Teachers as a user will have access to add/delete subjects, enter text matter, setting up parameters for a test, database management. Teacher is the administrator of the system and thus has full control of the system and can also view results of students.
* Students as a user can give tests using his/her login credentials and see their results.

A point wise description of expected outcome when applying ‘Intelligent Question Paper Generator’ in real life scenarios:

* Intelligent Question Paper Generator will generate a well formatted question paper in a matter of a few seconds i.e. saving a lot of time when compared to traditional systems.
* With the use of this system for question paper generation there are no chances of exam paper getting leaked as paper is generated just before the test.
* It will save a lot of time for teachers and thus will improve efficiency.

If students are to transfer knowledge into practice, testing is a must to foster a student's ability to think quickly and analytically rather than only to recall facts. There are many problems associated with manual human based question paper generation, therefore to overcome this we came up with ‘Intelligent Question Paper Generator’.

In the coming semester we will work to get an advanced approach for this project , extend the project and will focus on eradicating any of the limitations we are facing around.

**Chapter-7**

**Future Scope**

**7.1 Future Scope**

This software can be widely used in educational institutes etc. It is a web based application so user location doesn’t matter. Admin can access this software anywhere anytime. This system also evaluates the candidate’s capability and skills efficiently. It is a fully automated system which results in fast results. In this system there is no need of transporting paper through police/security vans to all colleges. This system provides unbiased results. The usage of this system reduces human effort and saves time and resources to an extent.

This project can be further extended to generate yes/no type questions and fill in the blanks type questions. Further automatic answer copy checking systems can be added for checking the answer sheets and providing the result to the student.

**REFERENCES**

[1] Xuchen Yao, Gosse Bouma, Yi Zhang, “Semantics-based Question Generation and Implementation”, Department of Computer Science, Johns Hopkins University

[2] Bangera Shanika Ashok Shanthi, Harshitha, Leona Josline Rego, Manasa K, “Automated Exam Question Generator”, International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-0056, (Volume-6, Issue-5)

[3] A.Immauel and Tilasi.B, “Framework for Automatic Examination paper generation system”, International journal of computer sciences and technology, vol 6, no1, pp.128-130

[4] www.semanticscholar.org , “Automated Question paper generation System”