**Chapter 1**

**INTRODUCTION**

In this chapter,basic introduction of the system is given which consists of problem description which include what are the problems that are faced normally that are solved by this system and scope of the project which include the sources required and rules that are applied on the system.

**1.1 EXISTING SYSTEM**

Normally question papers were generated manually, which was time consuming. To overcome this problem new system is generated that uses keywords from syllabus to generate question considering the rules and restriction, paper is generated very effectively and efficiently. The question paper generations system that currently exists is totally based on database. It just randomly selects the questions from database and generates question paper.

In this system instead of adding each question into database for question paper generation, CSV files are used to generate question paper based on keywords that are extracted from that CSV file.

**Chapter 2**

**PROPOSED SYSTEM**

This system changes the process of creating question paper for teachers which can save a lot of time and effort. The paper is generated just like teachers would like to create. Every question is selected based on weight age of that chapter and randomly therefore whole syllabus is covered with questions generated from each chapter. Thus it saves a lot of time which can be used by teachers to do some other important activities. Teachers can also create question paper for term tests held in colleges for 20 marks so that teachers can put their valuable time in other important activities like preparation for lectures, presentation and conferences etc. This system helps the users mainly the teachers for efficient question paper generation.

**2.1 SCOPE**

Firstly System will convert the syllabus PDF to CSV file which will allow the user to add file to database.PDF is converted first to word file. Then word file is converted into excel file, and then excel file is converted to CSV and then it is imported into database. After that using a database query, keywords are extracted and question paper is generated.

In this system CSV file is used because it is easy to add the CSV file into database. CSV file is loaded in the database in such way that all of its column and rows are automatically inserted in the table so it can be easy to access. This system has reduced scope because the system cannot detect table from PDF which consist of syllabus of all subjects. The system is applicable to only single subject syllabus table. So, user should upload PDF file which consist of syllabus in single table format given by university.

**2.2 FEATURES**

Advanced question paper generation system fully automates the process of question paper generation. User just needs to provide the PDF file according to university format. This system will use the information provided in PDF file and different algorithms will be applied on information gathered and finally it will generate question paper based on selected keywords (topics) by appending it to the question.PDF of question paper consisting of 12 questions each of 10 marks .The questions covers whole syllabus including all modules. The numbers of questions are more from module which has more number of hours. Each question has two sub questions a, b therefore paper consists of total 6 questions. This generated PDF file can be mailed to colleges and universities.

**Merits**

1. The questions can be selected from entire syllabus which is unpredictable.
2. This system can offer different variety in question generation means it will ask

Questions from any topic.

3) If syllabus changes then just by providing new syllabus PDF file system can generate new

Question paper.

**Chapter 3**

**SYSTEM ANALYSIS**

In this chapter the description of the papers are given that as per module. That includes the functions and the algorithms that are going to be used in this system are studied using papers that are published.

**3.1 MODULE**

Comma-separated values (CSV) file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. An official standard for the CSV file format does not exist, but RFC 4180 provides a de facto standard for many aspects of it. In popular usage, however, the term CSV may denote some closely related delimiter-separated formats, which use a variety of different field delimiters. These include tab-separated values and space-separated values, both of which are popular. Such files are often even given a .CSV extension, despite the use of a different field separator than the comma.

**3.1.1 KEYWORD EXTRACTION**

Fang Yuan in year 2005 has proposed a method for extracting information from PDF files. The rule set is constructed for manually or from learned training data for information extraction and modified later. There are many Information Extraction (IE) approached can be used e.g. STALKER, CCWRAP. Main task of IE algorithm to find out symbol of left and right side of each attribute. But these methods cannot be applied directly to PDF files.

The process of Information extraction consists of following modules.

1. Construction of PDF files parser.

PDF document consist of various information such as text, images and table information. PDF file parser parses this information.

1. Construction of tag injector.

In PDF files information is organized in the form of co-ordinates, so for extracting information just inject user defined tags for example font size, font type, empty row etc.

1. The process of tag preprocessor.

It process the user defined tags.

**3.1.2 EXTRACT TABLE FROM PDF**

Burcu Yildiz in year 2014 has proposed a way to decompose table and store extracted data in structured data format (XML) for easy reuse. Table often contains high density information. PDF file is taken as an input. Table Extraction is process of decomposing table information in document. In this process author made use of PDF to html tool. For each text chunk in PDF file it returns text in XML with following attributes. Because table presents and structures data with high density therefore the task of Table Extraction is very important. But this task is not easy because tables can be of varying formats. In this paper input files used is generally PDF files. Table Extraction is task of detecting and decomposing table information in a document. The PDF to html tool returns text chunks and their absolute coordinates in the PDF file in same order s they were inserted in the original file.

* **Attributes**

Attributes are the properties related to the text

* 1. Top=vertical distance from the top of page

1. Left=horizontal distance from the left border of page
2. Width=width of text part
3. Height=height of text part
4. Font= this attribute describe the size, family and color of text part [6].

* **Heuristics**

Here author has studied and explored different types of table and their structure and given some heuristics. The heuristics used can be grouped in two categories

1. Heuristics intended to recognize table.

This task deals with the problem of identifying a construct as a table. The level of difficulty of this task depends, among others, on the document in which the table is embedded. As we deal with an XML document which does not mark-up tables, we have to identify a portion of text elements as a table only by means of the knowledge of the absolute coordinates of the text elements.

* 1. Heuristics intended to decompose table.

After detecting a part of a file as a table, the next step is to decompose the table as close to the original as possible. This task includes the correct identification of header elements, their spanning behavior (i.e., how many columns or rows are spanned), the correct assigning of data cells to header elements, and so on.

**3.1.3 GENERATION OF PAPER FROM QUESTION BANK**

The structure of question database, paper database and template database are discussed .First question type is designed including subjective and objective questions. The test database contains variety of question with difficulty levels. Using the system user can choose paper setting manual.

**Construction Program**

1) General principle

According to the course syllabus, combining with the teaching purpose, requirements and contents of course, the proposition scheme is formulated. The specific requirements are as follows: extensive covering range focused key points, moderate difficulty, rich question types, moderate questions amount, and complete attachment, standardized and unified format.

2) Determination of proposition outline

It is mainly made based on the professional characteristics of our school and the allocation of teaching hours. Based on the course syllabus, the system can accommodate the requirements of the knowledge updating and education reform and examine the basic knowledge and basic skills of students and the capability of applying knowledge to solve practical problems.

3) Selection of question source

The question source is mainly from the various reference books, and it is enriched by self-designed questions. Because of its richness of the question database, we can select the questions calmly. There are three levels of question selection, they are master, understand and know, and the excessive repetition rate can be avoided.

4) Determination of question coverage

There are many contents in the entire course, the knowledge contents of the examination should cover the main points of the course, and the coverage ratio should be not less than 80%. The test should fully examine the master extent for students to course knowledge. At the same time, the knowledge points should be considered reasonably and comprehensively to test the ability of utilizing overall knowledge.

5) Setting of question type

Firstly, the question type is designed, including choice, fill in the blank, calculation, analysis, error correction, mapping, and so on. It includes objective questions and subjective questions. The former can adapt the requirement of the current standardized tests, mainly to test basic concepts, knowledge breadth, and judging ability of the students, including choice, fill in the blank. Subjective questions can test the ability of students to analyze the issues and use the knowledge to solve problems, including computing, analysis, error correction, mapping problems. Generally, the latter has higher difficulty.

**Functions**

Through user interface user can access following functions:

1. **Test question database maintenance:** The question can added, deleted and modifiedfrom database.
2. **Manual paper combination:** In proposed system user can select the required questionand add it into paper.
3. **Intelligent paper combination:** In proposed system question are selected randomlyaccording to specific requirement and conditions.
4. **Test paper generation:** It is the combination of manual and intelligent papercombination.
5. **Other function:** Other function like setting, system exit, library open etc [1].

**3.2** **WORKING**

In this chapter the description of the project is given which includes which are the conditions that should be satisfied to start the process, the functions that are used in this system and system requirements to run this system.

**3.2.1 THEORITICAL ISSUES**

Currently system creates question paper on theoretical subjects only. For creating questions on mathematical subjects and for questions like draw diagram and explain algorithm etc. our system is not ready yet. For creating questions like specified above system can make use of EBook so that page references can be used to find questions like these. System can also make use of CSV files of only problems/sums in a particular subject so that CSV files can be imported in the database to create question paper directly. These kinds of questions can be generated by this system in future with help of some modifications.

**3.2.2 PRE-PROCESSING**

First to initialize the process a PDF file of syllabus according university format is needed. This PDF should include all topics in each subject in tabular format. The table should include Module No, Module Name, and Hours. But this information cannot be used from PDF file directly; therefore it should be converted to Comma Separated Value (CSV) file. Once CSV file is ready, it can be easily imported into the database.

To convert PDF file into CSV by following process:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |
|  | PDF FILE |  |  | WORD FILE |  |  | EXCEL FILE |  |  | CSV FILE |
|  |  |  |  |  |  |  |  |  |  |  |



Fig 3.2.2: File Conversion

Above Fig 3.2.2 represents the conversion of PDF file to CSV file. After this process is completed the system imports this CSV file into the database for further process. Finally the user will just have to select Class and Subject to create question paper and according to the selection of particular CSV file will be loaded from the database for question creation.

**3.2.3 STRING TOKENIZERS**

The string tokenized class allows an application to break a string into tokens. The tokenization method is much simpler when it is used by the String Tokenizer class. The String Tokenized methods do not distinguish among identifiers, numbers, and quoted strings, nor do they recognize and skip comments. The set of delimiters (the characters that separate tokens) may be specified either at creation time or on a per-token basis.

**3.2.4 STEP-WISE PROCEDURE**

**Syllabus Engine**

Rules are defined on the syllabus engine and user should provide the syllabus PDF as per university pattern. Syllabus mainly consist of modules, topics and subtopic.

**Pattern Composer**

Pattern of the syllabus should be such that it should be according to weight age of each module. Paper generated must be of 100 marks.

**Generate Question Paper**

After the sub-topic/topic keywords are searched, randomly it will get appended to the question and question is generated. With the help of this system it can be make sure that question will not be repeated again and again.

**3.2.5 STRING TOKENIZED ALGORITHM**

Basically in this system the algorithm takes a string which is in the comma separated format, so string tokenized class is used to separate each string with a delimiter. First it reads column/subtopic from database and tokenizes that string using delimiter. Take a string array and make object of StringTokenizer class. This class tokenize string using delimiter comma (,) and generates tokens. String has number of tokens so the process is repeated until all tokens are used.

**3.2.6 MAPPING ALGORITM**

Mapping Algorithm (mapping of number of questions and hours)

1. Simply read the generalized table which has no. of questions according to no. of hours and syllabus table.
2. Map the hour’s column in each table using some SQL operation like join operation
3. Format should also be in the following form:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Module 1 |  | Number of questions 2 |  |
|  |  |  |  |  |
|  |  |  |  |  |



Fig 3.2.6: Format of Mapping

This algorithm will be used for mapping of modules and questions that mean how many questions should be generated from the respected module. If module has higher weight age then more questions generated from that module, if module has lesser weight age then less questions generated from that module. Mapping should be such that entire syllabus is covered and number of questions must be equal to 12 and paper should be of 100 marks.

**3.3 SYSTEM REQUIREMENTS**

**Software requirements**

1. Netbeans Integrated Development Environment (IDE)
2. Java Development Kit (JDK)
3. XAMPP database system

**Hardware requirements**

* 1. 1 GB RAM
  2. Dual Core Processor and Windows 7/XP/Vista

**3.4 FRONT END**

**Java JFrame** - JFrames are swing components. A swing component is a part of a graphic user interface (GUI). Frames, windows, text boxes, buttons, switches and many other parts of a GUI application are all components. The root class of all swing components is the JComponent class and other swing components including JFrame are subclasses of JComponent. JComponent is an abstract class, so it cannot be instantiated directly but can be subclassed, which is useful if you want to write your own custom GUI component.

A JFrame is a top-level component, meaning that it contains other components but is not contained itself. It's onscreen appearance is dictated by the platform but generally it is a window that the user can resize, move, maximize, minimize, and has its own title bar.

Probably the best way to get familiar with a JFrame as well as components in general is to run a short example program that creates one.

**3.5 BACK END**

**MySQL using XAMPP**- XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends,consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

* 1. **FEASIBILITY**

A Feasibility Study determines whether a project is worth doing. The process followed for making this determination is called a Feasibility Study. This type of study determines whether a project can and should proceed. Once it has been determined that a project is feasible, the analyst can proceed and prepare the project specifications that finalize the project specification.

The following are the various types of feasibility studies that can be undertaken:

**Technical Feasibility**

This is concerned with specifying the equipments and the software to satisfy the user requirements. The technical needs of the system vary considerably but might include:

• The facility to produce outputs in a given time.

• Response time under certain conditions.

• Ability to produce a certain paper at a specified speed.

• Facility to communicate data to a distant location through email.

Technical feasibility centers on the existing computer system, hardware, software etc. and to what extent it can support the system. In examining the technical feasibility, the configuration of the system is given more importance than the actual software. The configuration should provide the complete picture of the system requirements, for example how many modules are required and how these units are interconnected so that they would operate smoothly. Specific hardware and software products can then be evaluated keeping in view the logical needs.

**Economic Feasibility**

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system.

It is not done to analyse the new system. We assumed that the benefit of the project is greater than the cost. So if we can develop the project easily then it is used for the evaluation of the proposed. We calculate the cost/benefit analysis and we assume that the benefit is feasible so we start developing the project. It is an analysis of the cost to be incurred in the system and benefits the derivable from the system. An economic Feasibility Study should demonstrate the net benefit of the proposed course of action in the context of direct and indirect benefits and costs to the organization and to the public as a whole. It should be required for both pilot and long-term activities, plans and projects.

**Feasibility Study Report:**

The result of the Feasibility Study provides us with the following facts:

* The automated system would increase the efficiency of the system.
* The automated system would increase customer's satisfaction.
* The automated system has many requirements such as Efficiency cost effectiveness, prompt service, Reliability.
* The automated system should be simple to use, incorporate all necessary services and maintainable.

**Chapter 4**

**PROJECT DESIGN**

**4.1 FLOW CHART**

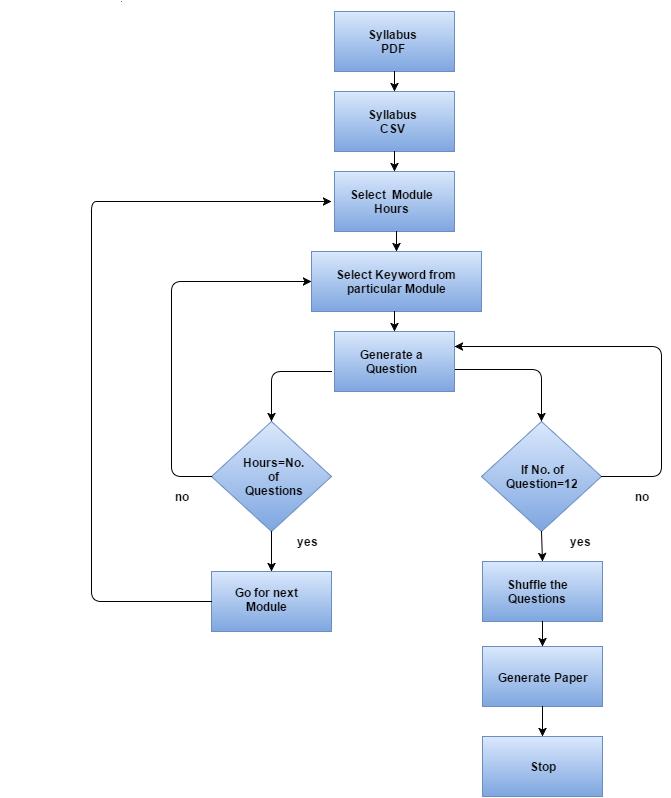


Fig 4.1: Flow Chart

**4.2 DFD DIAGRAM**

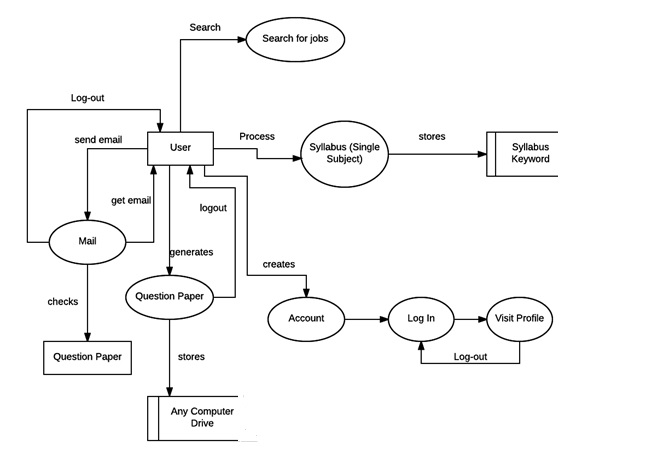


Fig 4.2:DFD Diagram

**4.3 USE CASE**

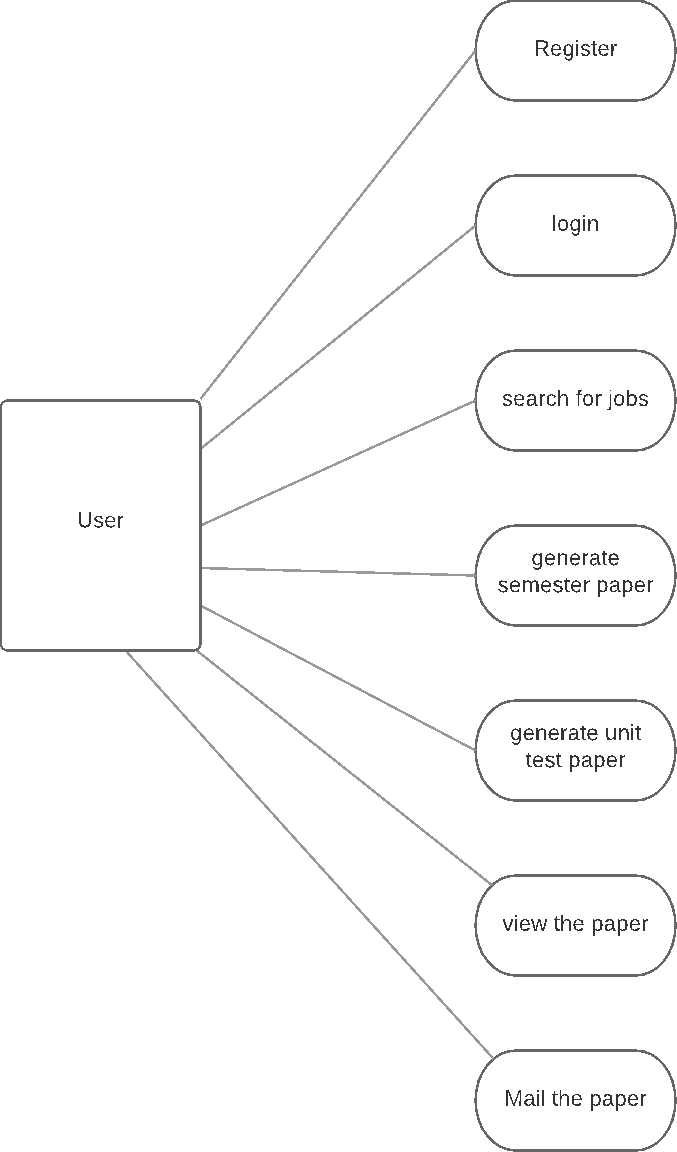


Fig 4.3:Use Case Diagram

**4.4 SEQUENCE**

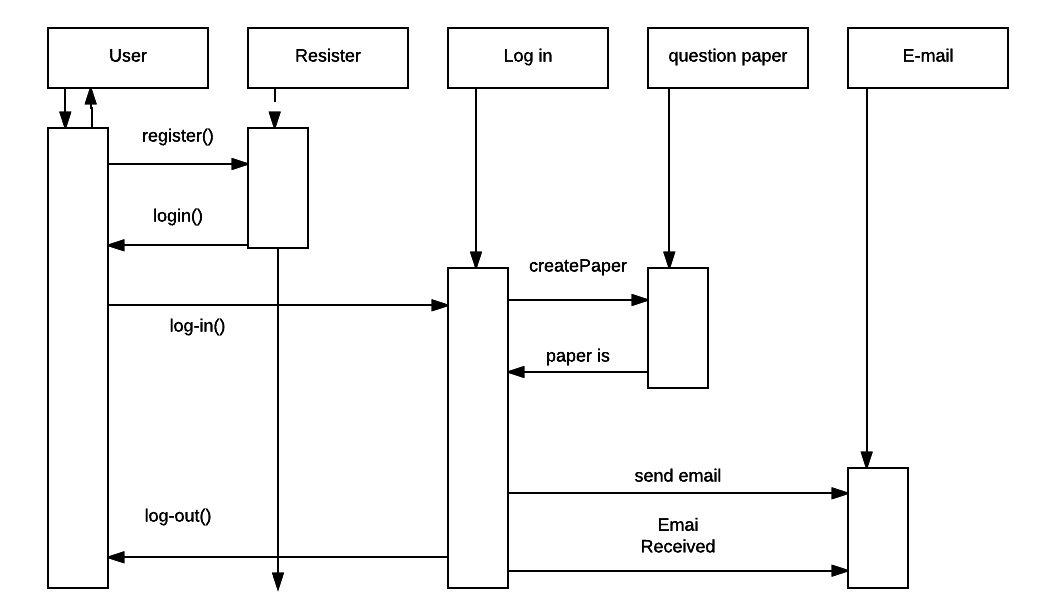


Fig 4.4:Sequence

**4.5 CLASS DIAGRAM**

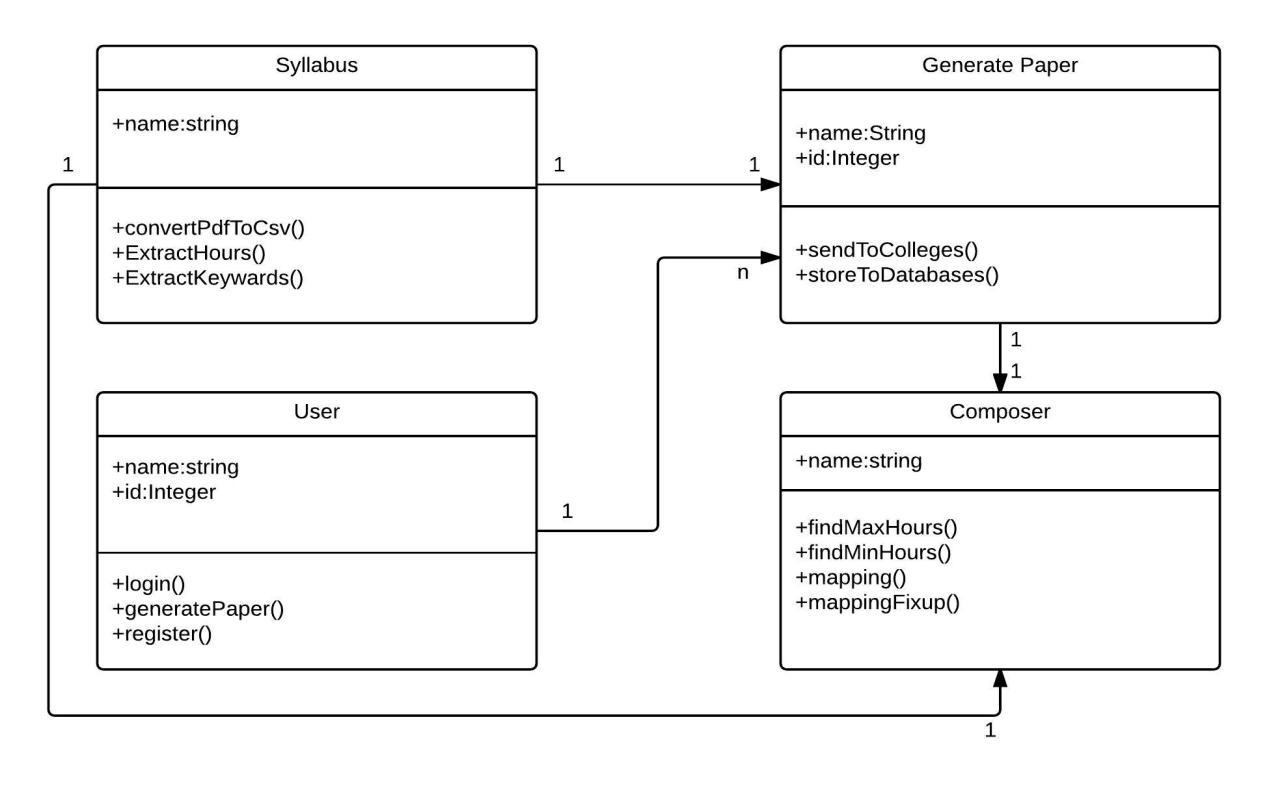
****

Fig 4.5:Class Diagram

**4.6 ER DIAGRAM**

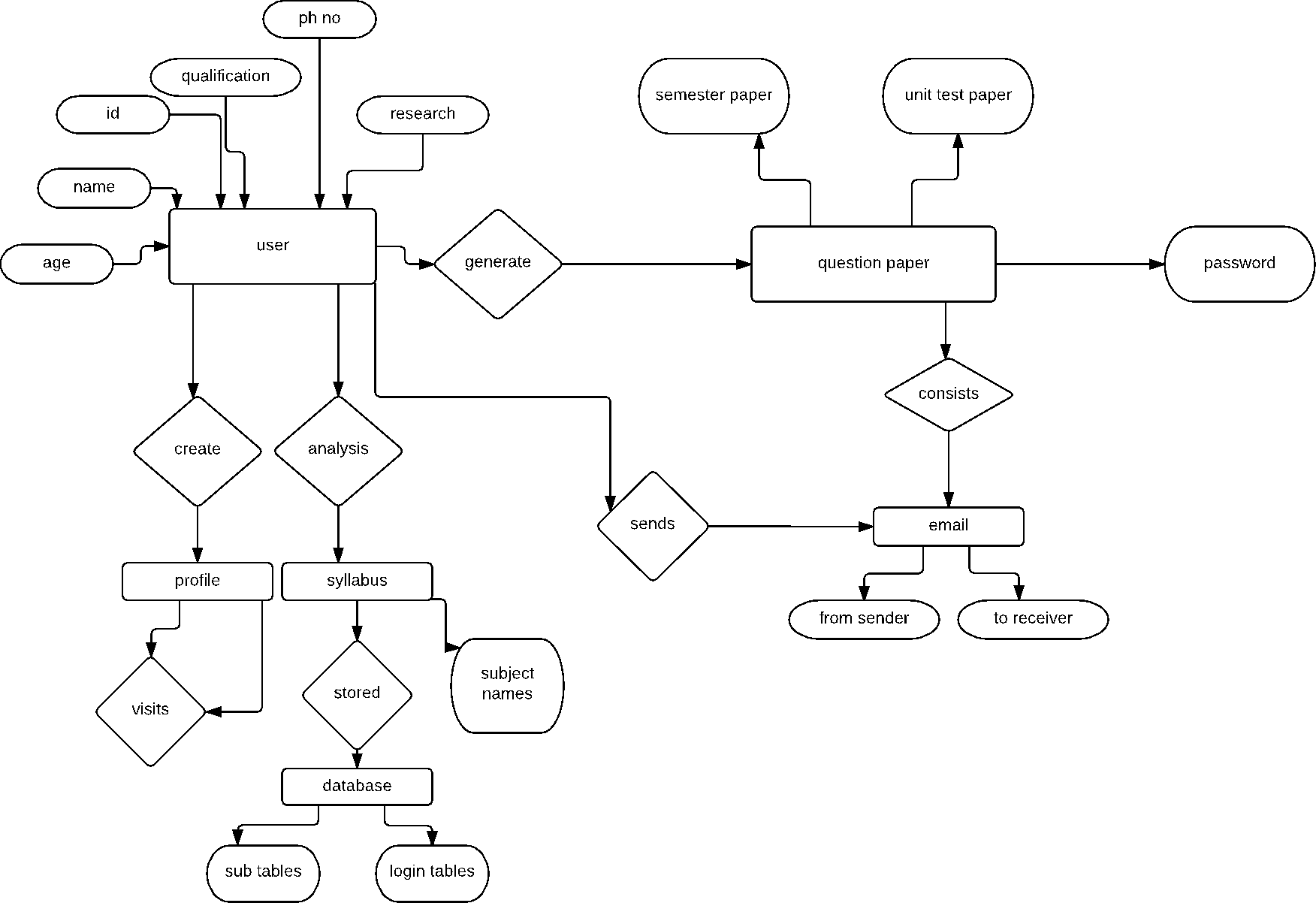


Fig 4.6:ER Diagram

**Chapter 5**

**EXPERIMENTAL RESULTS**

In this chapter implementation of the system is described which includes step wise procedure, algorithms that are used in this system and flow of the system using flow chart. The results that are expected from this system and demerits of the existing system are also described.

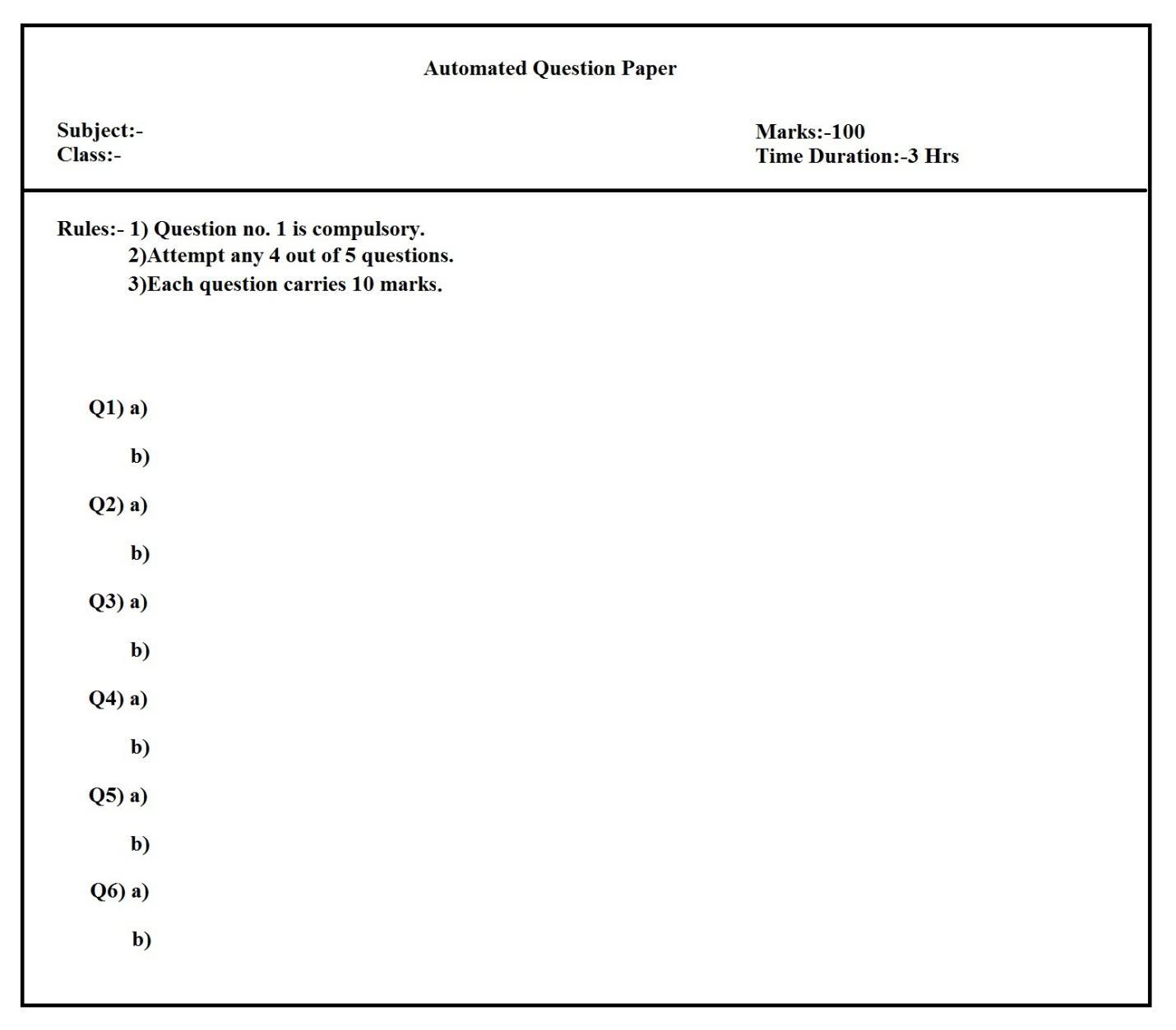


Figure 5.1: Question paper pattern

**5.1 SCREENSHOTS**

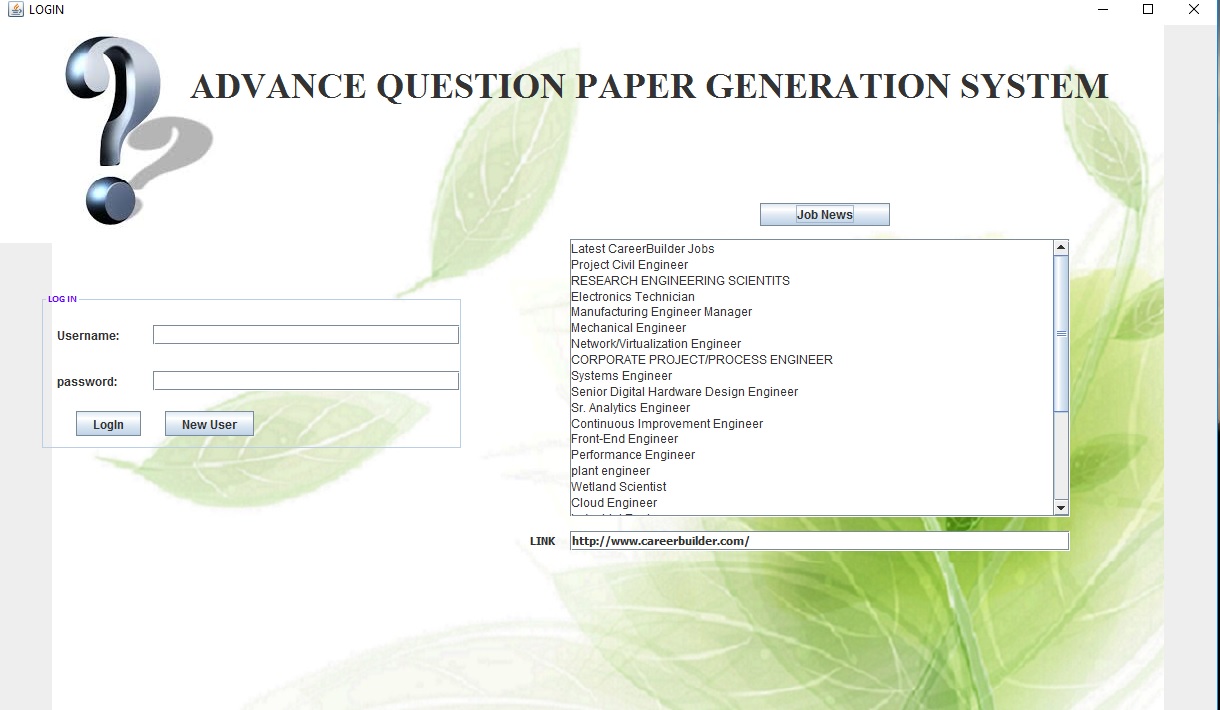


Figure 5.1.1: Login page

Above fig 5.1.1 represents the login window of Advance Question Paper Generation System. For login user have to insert username and password. If the user is not registered yet than he has to click on New User button to register. There is also Job News button which provides us various feeds about current job opportunities.

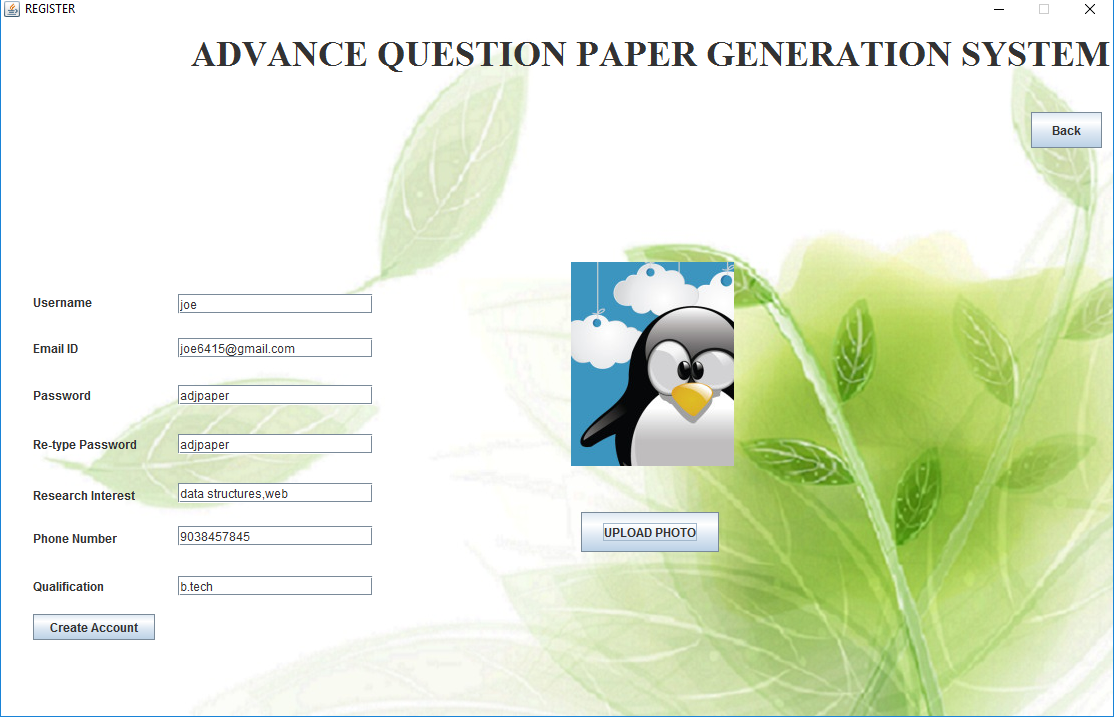


Fig 5.1.2: Register page

Above Fig 5.1.2 represents the register window of Advance Question Paper Generation System. When user click on New User button on login window user is directed to this window. In this window user have to fill all the fields and upload is photo to create an account.

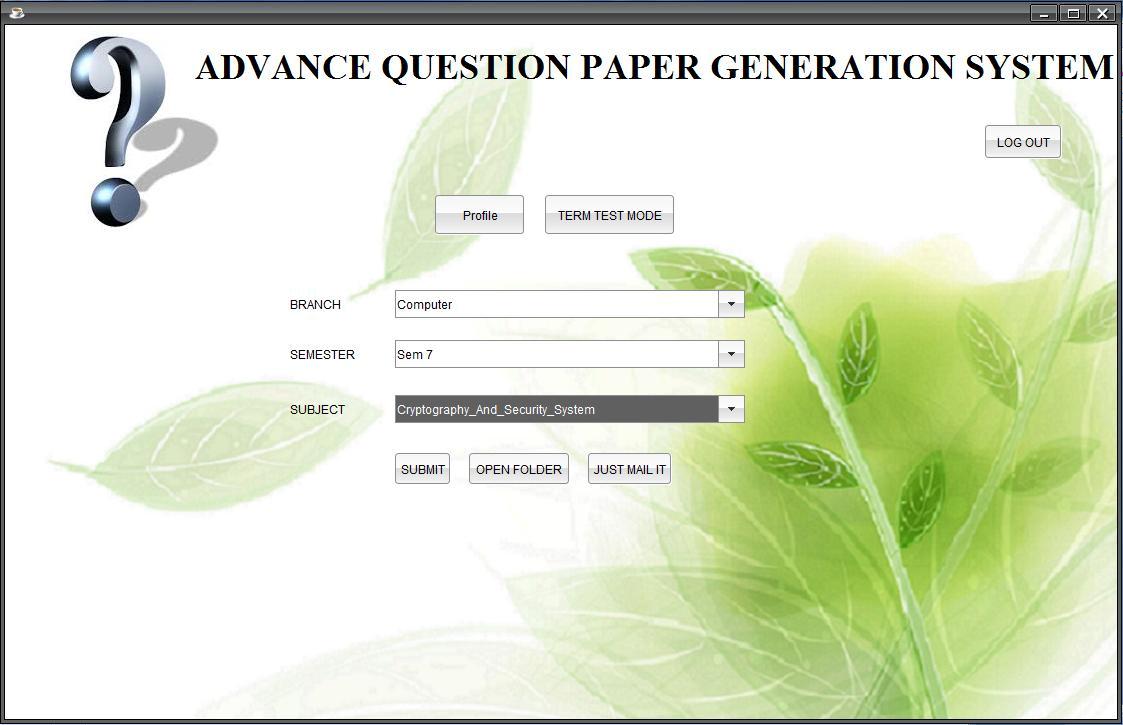


Figure 5.2.3: Composer page for MU

Above Fig 5.2.3 represents the Mumbai University mode window of Advance Question Paper Generation System. In this mode user can create question paper for Mumbai University. User just have to select branch, semester and subject and when click on submit button the question paper is created in E drive of computer. When user click on open folder button E drive is opened and user can see the generated paper. The generated paper is also password protected. There is also a button to go to Term Test mode and button to mail the paper

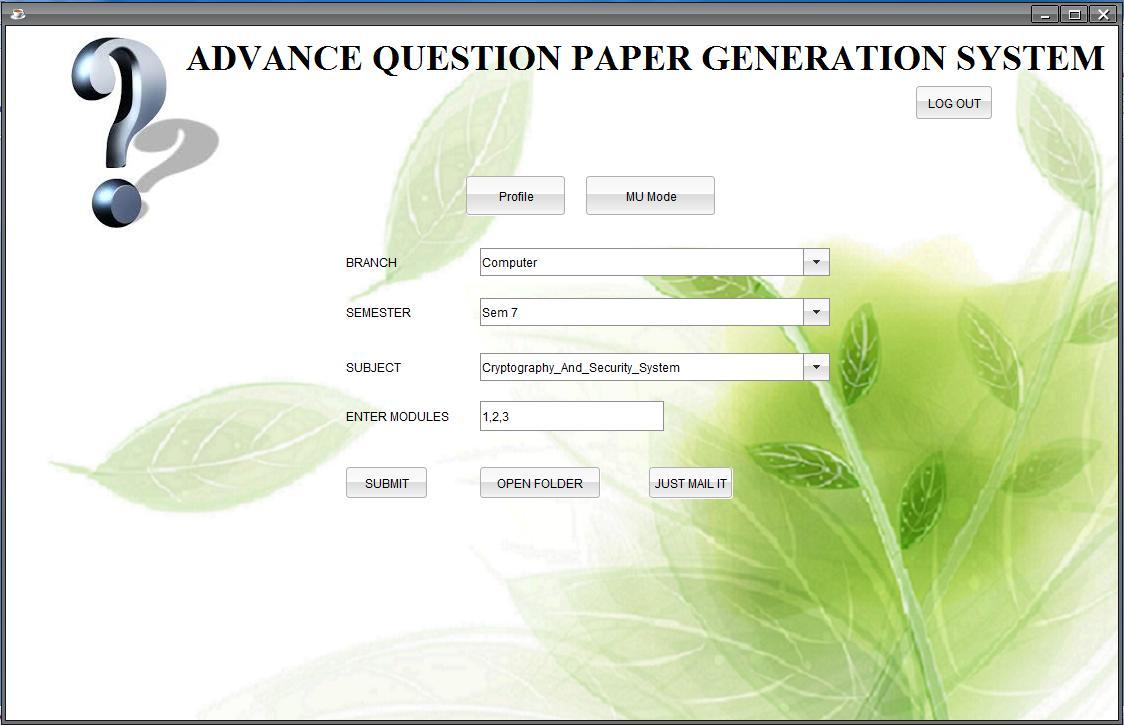


Figure 5.2.4: Composer page for term test

Above Fig 5.2.4 represents the Term Test mode window of Advance Question Paper Generation System. In this mode user can create question paper for Term Test held in colleges. User just have to select branch, semester and subject and when click on submit button the question paper is created in E drive of computer. When user click on open folder button E drive is opened and user can see the generated paper. The generated paper is also password protected. There is also a button to go to Mumbai University mode and button to mail the paper

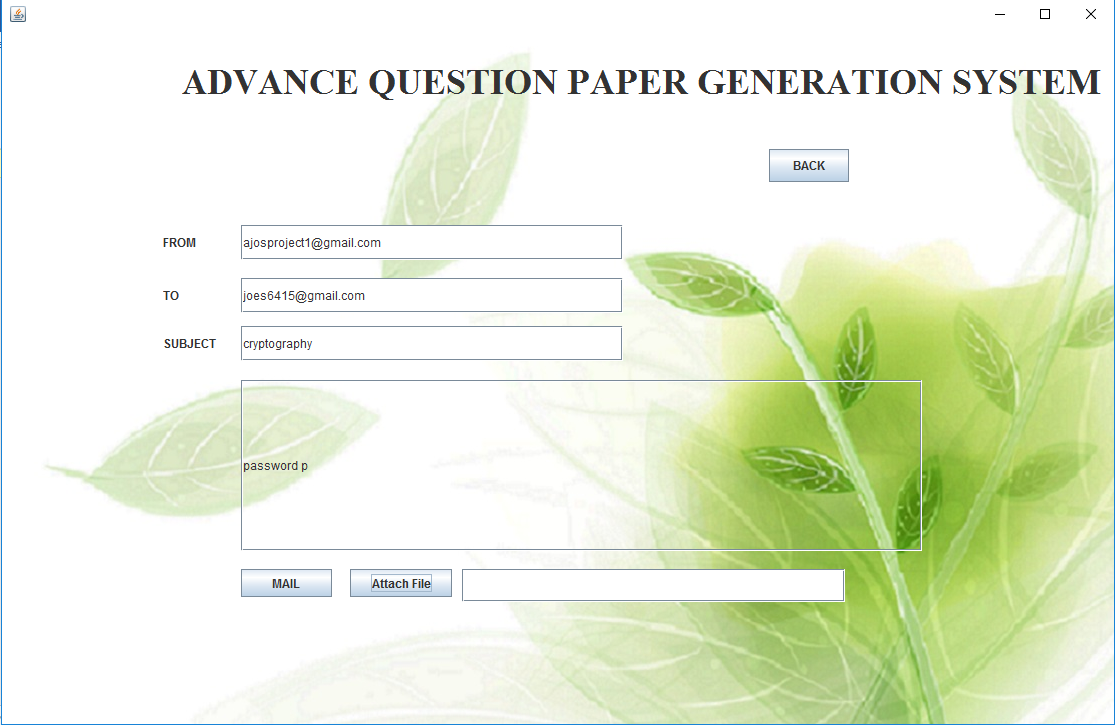


Figure 5.2.5: Mail page

Above Fig 5.2.5 represents the Mail window of Advance Question Paper Generation System. In this window user can mail the generated question paper to anyone including a person or colleges. Along with paper user have to provide the password to unlock the PDF file of question paper.

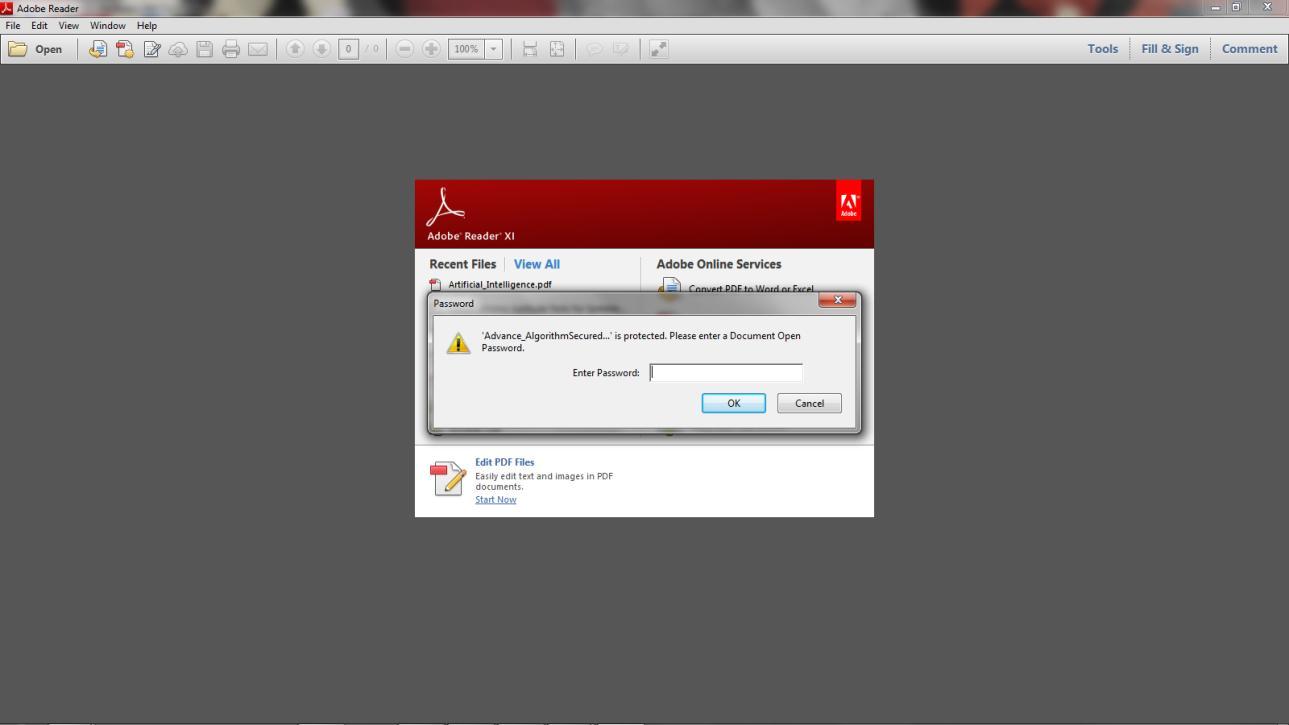


Figure 5.2.6: password page for paper

Above Fig 5.2.6 shows that while opening the generated question paper Adobe Reader asks for password therefore the generated question paper can be viewed by legitimate user only.

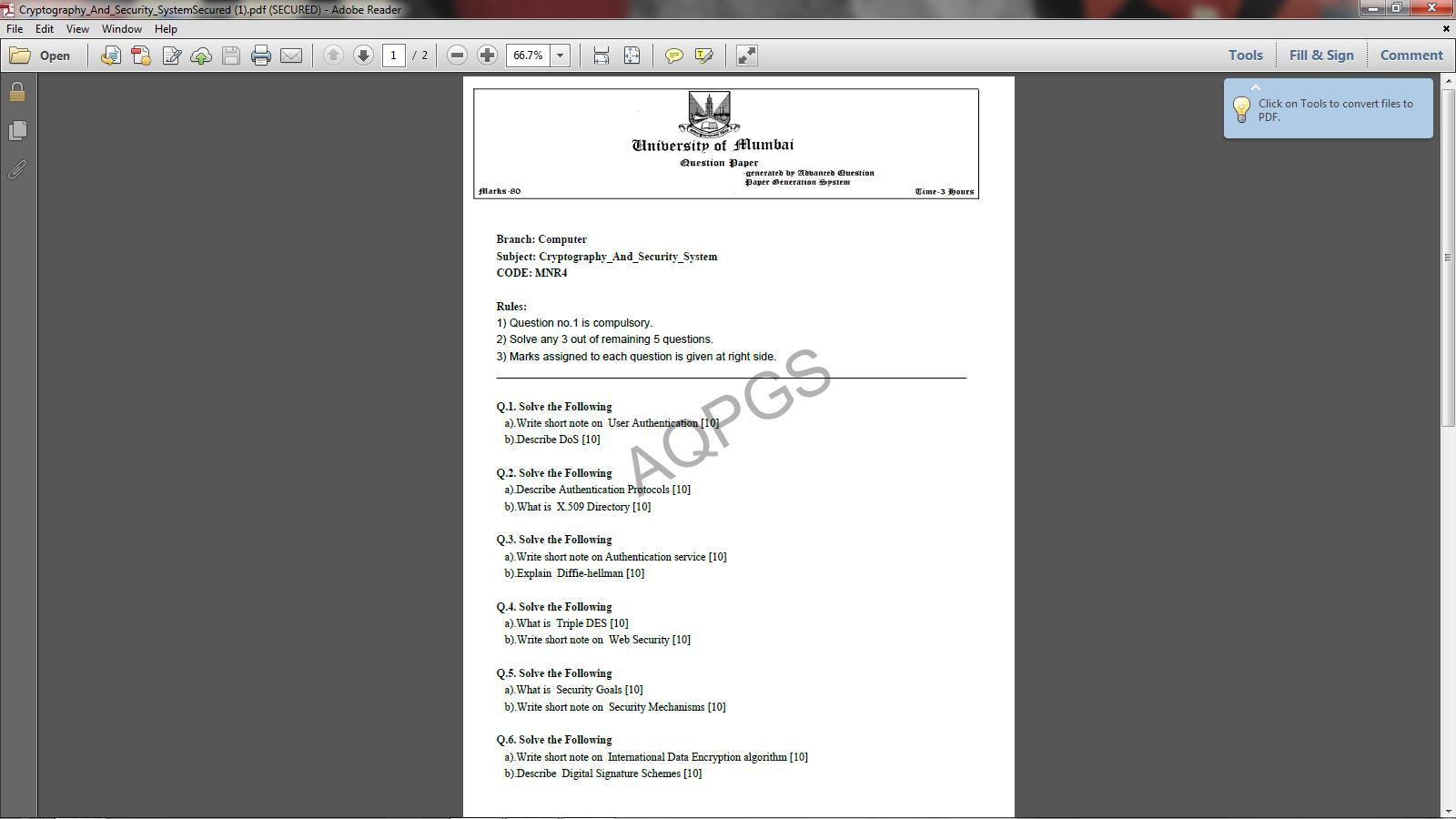


Figure 5.2.6: Composer page for MU

Above Fig 5.2.6 shows the generated question paper of subject Cryptography and system security for Mumbai University of 80 marks. This paper includes 6 questions of 20 marks each with option out of which student have solved any 4 question including first question. No matter how many times you create question paper of the same subject you will get different question every time.

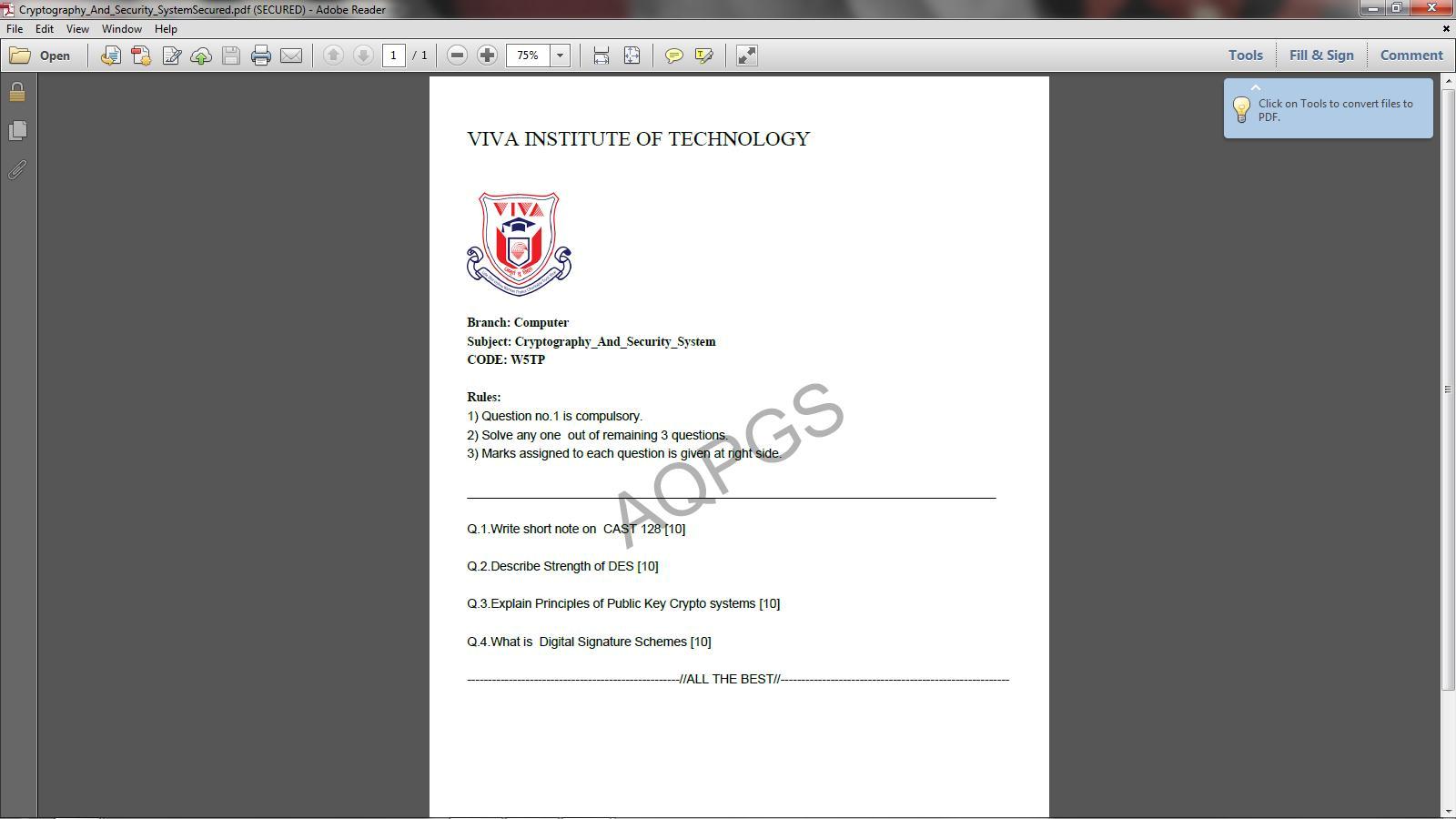


Figure 5.2.7: Composer page for MU

Above Fig 5.2.7 shows the generated question paper of subject Cryptography and system security for Term Test of 20 marks. This paper includes 4 questions of 10 marks each with option out of which student have solved any 2 question including first question. No matter how many times you create question paper of the same subject you will get different question every time.

**5.2 CODE**

import static automaticpaper.Login1\_1.jobs;

import static automaticpaper.Login1\_1.sourceCode;

import static automaticpaper.rssFeeder.readRss;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.WindowEvent;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.MalformedURLException;

import java.net.URL;

import java.sql.\*;

import java.util.Random;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.\*;

import javax.swing.text.DefaultCaret;

public class Login1 extends javax.swing.JFrame {

public static String username1;

public static String sourceCode="";

Connection con = null;

PreparedStatement pst = null;

ResultSet rs = null;

static String pwd=null;

public static String uname = null;

//String user1=null;

//private jframe frame;

// private Login1 frame;

/\*\*

\*

\*/

public Login1() {

setContentPane(new JLabel(new ImageIcon("C://Users//Joyan//Desktop//11.jpg")));

initComponents();

//this.setLocation(500,500);

/\* Random gen = new Random();

Color color = new Color(gen.nextInt(240), gen.nextInt(240),

gen.nextInt(240));

this.getContentPane().setBackground(color);\*/

this.setTitle("LOGIN");

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

public void close() {

WindowEvent WinClosingEvent = new WindowEvent(this, WindowEvent.WINDOW\_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(WinClosingEvent);

}

// @SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel3 = new javax.swing.JLabel();

jScrollPane2 = new javax.swing.JScrollPane();

jTextArea1 = new javax.swing.JTextArea();

jPanel1 = new javax.swing.JPanel();

txt\_password = new javax.swing.JPasswordField();

jLabel1 = new javax.swing.JLabel();

login = new javax.swing.JButton();

jLabel2 = new javax.swing.JLabel();

txt\_username = new javax.swing.JTextField();

register = new javax.swing.JButton();

jLabel4 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

news = new javax.swing.JButton();

jLabel5 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

newname = new javax.swing.JTextArea();

LINK = new javax.swing.JTextField();

jLabel7 = new javax.swing.JLabel();

jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 36)); // NOI18N

jLabel3.setText("ADVANCE QUESTION PAPER GENERATION SYSTEM");

jTextArea1.setColumns(20);

jTextArea1.setRows(5);

jScrollPane2.setViewportView(jTextArea1);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setBounds(new java.awt.Rectangle(100, 0, 400, 100));

setIconImages(null);

addWindowListener(new java.awt.event.WindowAdapter() {

public void windowOpened(java.awt.event.WindowEvent evt) {

formWindowOpened(evt);

}

});

getContentPane().setLayout(null);

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "LOG IN", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Calibri", 1, 10), new java.awt.Color(102, 0, 255))); // NOI18N

jPanel1.setOpaque(false);

jLabel1.setText("Username:"); // NOI18N

login.setText("LogIn");

login.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

loginMouseClicked(evt);

}

});

login.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

loginActionPerformed(evt);

}

});

jLabel2.setText("password:");

txt\_username.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

txt\_usernameActionPerformed(evt);

}

});

register.setText("New User");

register.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

registerActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, jPanel1Layout.createSequentialGroup()

.addGap(19, 19, 19)

.addComponent(login)))

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 80, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(txt\_username, javax.swing.GroupLayout.DEFAULT\_SIZE, 307, Short.MAX\_VALUE)

.addComponent(txt\_password)))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(24, 24, 24)

.addComponent(register)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 32, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(txt\_username, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 24, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(txt\_password, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(login)

.addComponent(register))

.addContainerGap())

);

getContentPane().add(jPanel1);

jPanel1.setBounds(40, 270, 423, 159);

getContentPane().add(jLabel4);

jLabel4.setBounds(32, 116, 0, 159);

jLabel6.setFont(new java.awt.Font("Times New Roman", 1, 36)); // NOI18N

jLabel6.setText("ADVANCE QUESTION PAPER GENERATION SYSTEM");

getContentPane().add(jLabel6);

jLabel6.setBounds(190, 30, 919, 65);

news.setText("Job News");

news.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

newsActionPerformed(evt);

}

});

getContentPane().add(news);

news.setBounds(760, 180, 130, 23);

jLabel5.setIcon(new javax.swing.ImageIcon(getClass().getResource("/automaticpaper/images.jpg"))); // NOI18N

getContentPane().add(jLabel5);

jLabel5.setBounds(0, 0, 230, 220);

newname.setColumns(20);

newname.setRows(5);

jScrollPane1.setViewportView(newname);

getContentPane().add(jScrollPane1);

jScrollPane1.setBounds(570, 216, 500, 280);

LINK.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

getContentPane().add(LINK);

LINK.setBounds(570, 510, 500, 20);

jLabel7.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

jLabel7.setText("LINK");

getContentPane().add(jLabel7);

jLabel7.setBounds(530, 510, 30, 20);

pack();

}// </editor-fold>

private void txt\_usernameActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code her

}

private void formWindowOpened(java.awt.event.WindowEvent evt) {

try {

// TODO add your handling code here:

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/csv\_db", "root", "");

//frame= new Login1();

} catch (SQLException ex) {

Logger.getLogger(Login1.class.getName()).log(Level.SEVERE, null, ex);

}

}

@SuppressWarnings("empty-statement")

private void loginMouseClicked(java.awt.event.MouseEvent evt) {

}

private void loginActionPerformed(java.awt.event.ActionEvent evt) {

String sql = "select \* from login where username=? AND Password=?";

// TODO add your handling code here:

try {

pst = con.prepareStatement(sql);

// pst = con.prepareStatement(sql1);

pst.setString(1, txt\_username.getText());

pst.setString(2, txt\_password.getText());

rs = pst.executeQuery();

pwd=txt\_password.getText();

int count = 0;

// System.out.println(sql1);

uname = txt\_username.getText();

//System.out.println(user);

String password=txt\_password.getText();

//composer4.username1=user;

//GlobalVars.addValuesToGlobalMap("username", user);

if (rs.next()) {

// System.out.println(count);

// count=count+1;

//System.out.println(count);

// JOptionPane.showMessageDialog(null, "username and password is correct");

// frame.dispose();

// System.out.println(count);

//frame.dispose();

this.dispose();

composer4 c = new composer4();

c.setVisible(true);

//Paper\_info s= new Paper\_info();

//s.setVisible(true);

} else {

JOptionPane.showMessageDialog(null,"Invalid username and password");

}

register r = new register();

// Profile profile = new Profile();

// Profile.name.setText("ABC");

// Profile.name.setEnabled(false);

// Profile.qualification.setText(this.qualification1.getText());

// Profile.qualification.setEnabled(true);

// Profile.phone.setText(this.phone1.getText());

// Profile.phone.setEnabled(true);

// Profile.research.setText(this.research1.getText());

// Profile.research.setEnabled(true);

} catch (HeadlessException | SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

}

private void registerActionPerformed(java.awt.event.ActionEvent evt) {

register register = new register();

register.setVisible(true);

this.dispose();

}

public static String readRss(String urlAddress) throws IOException {

try{

URL rssUrl=new URL(urlAddress);

BufferedReader in=new BufferedReader(new InputStreamReader(rssUrl.openStream()));

int count=0;

String line;

while ((line = in.readLine()) != null ) {

int titleEndIndex = 0;

int titleStartIndex = 0;

while (titleStartIndex >= 0&& count!=10) {

titleStartIndex = line.indexOf("<title>", titleEndIndex);

if (titleStartIndex >= 0) {

titleEndIndex = line.indexOf("</title>", titleStartIndex);

System.out.println();

sourceCode += line.substring(titleStartIndex + "<title>".length(), titleEndIndex) + "\n";

// count++;

// System.out.println(count);

}

}

}

/\* while((line = in.readLine())!= null)

{

if(line.contains("<title>"))

{

int firstPos = line.indexOf("<title>");

String temp = line.substring(firstPos);

temp = temp.replace("<title>", "");

int lastPos = temp.indexOf("</title>");

temp = temp.substring(0, lastPos);

sourceCode += temp+"\n";

}

}\*/

/\* while((line = in.readLine())!= null)

{

if(line.contains("<title>"))

{

int firstPos = line.indexOf("<title>");

String temp = line.substring(firstPos);

int fp = temp.indexOf("</title>");

int lastPos = line.indexOf("</title>");

System.out.println(firstPos);

System.out.println(lastPos);

System.out.println(fp);

}

}\*/

in.close();

return sourceCode;

}catch(MalformedURLException e){

System.out.println("Something went wrong reading URL");

}

catch(IOException e){

System.out.println("hello");

}

/\*if (sourceCode.contains(",")) {

jobs = sourceCode.split(",");

}

System.out.println(jobs.length);

for(int i=2;i<jobs.length;i++){

System.out.println(jobs[i]);

}\*/

//catch(MalFormedURLException e){

//System.out.println("MalFormed URL");

//}

return null;

}

private void newsActionPerformed(java.awt.event.ActionEvent evt) {

newname.setText(sourceCode);

LINK.setText("http://www.careerbuilder.com/");

System.out.println();

// newname.setText("LINK:-http://www.careerbuilder.com/RTQ/rss20.aspx?rssid=RSS\_PD&num=25&geoip=false&ddcompany=false&ddtitle=false&cat=JN004");

/\* try{

// URL rssUrl=new URL("http://www.careerbuilder.com/RTQ/rss20.aspx?rssid=RSS\_PD&num=25&geoip=false&ddcompany=false&ddtitle=false&cat=JN004");

// BufferedReader in=new BufferedReader(new InputStreamReader(rssUrl.openStream()));

String line;

while ((line = in.readLine()) != null) {

int titleEndIndex = 0;

int titleStartIndex = 0;

while (titleStartIndex >= 0) {

titleStartIndex = line.indexOf("<title>", titleEndIndex);

if (titleStartIndex >= 0) {

titleEndIndex = line.indexOf("</title>", titleStartIndex);

sourceCode += line.substring(titleStartIndex + "<title>".length(), titleEndIndex);

newname.setText(sourceCode);

System.out.println();

}

}

}

/\*

if (sourceCode.contains(",")) {

jobs = sourceCode.split(",");

}

System.out.println(jobs.length);

for(int i=2;i<jobs.length;i++){

System.out.println(jobs[i]);

}\*/

/\* while((line = in.readLine())!= null)

{

if(line.contains("<title>"))

{

int firstPos = line.indexOf("<title>");

String temp = line.substring(firstPos);

temp = temp.replace("<title>", "");

int lastPos = temp.indexOf("</title>");

temp = temp.substring(0, lastPos);

sourceCode += temp+"\n";

}

}\*/

/\* while((line = in.readLine())!= null)

{

if(line.contains("<title>"))

{

int firstPos = line.indexOf("<title>");

String temp = line.substring(firstPos);

int fp = temp.indexOf("</title>");

int lastPos = line.indexOf("</title>");

System.out.println(firstPos);

System.out.println(lastPos);

System.out.println(fp);

}

}\*/

// in.close();

//return sourceCode;

// }catch(MalformedURLException e){

// System.out.println("Something went wrong reading URL");

// }

// catch(IOException e){

// System.out.println("hello");

//catch(MalFormedURLException e){

//System.out.println("MalFormed URL");

//}

//return null;

// newname.setText(sourceCode);

/\* String abc="<html>";

String pqr="<marquee>";

rss1.setText(jobs[2]);

rss2.setText(jobs[3]);

rss3.setText(jobs[4]);

rss4.setText(jobs[5]);

rss5.setText(jobs[6]);

rss6.setText(jobs[7]);

rss7.setText(jobs[8]);

rss8.setText(jobs[9]);

rss9.setText(jobs[10]);

rss10.setText(jobs[11]);

rss11.setText(jobs[12]);

rss12.setText(jobs[13]);

pqr=pqr+"</marquee>";

abc=abc+"</html>";\*/

// DefaultCaret caret = (DefaultCaret)newname.getCaret();

//caret.setUpdatePolicy(DefaultCaret.ALWAYS\_UPDATE);

//JScrollPane scrollPane = new JScrollPane();

//scrollPane.setViewportView(newname);

//scrollPane.set

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) throws IOException {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

/\* try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

//UIManager.setLookAndFeel("com.jtattoo.plaf.acryl.AcrylLookAndFeel");

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Login1.class

.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Login1.class

.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Login1.class

.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Login1.class

.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}\*/

//</editor-fold>

System.out.println(readRss("http://www.careerbuilder.com/RTQ/rss20.aspx?rssid=RSS\_PD&num=25&geoip=false&ddcompany=false&ddtitle=false&cat=JN004"));

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Login1().setVisible(true);

}

});

}

// public javax.swing.JLabel;

public static javax.swing.JTextField qualification1;

public static javax.swing.JTextField research1;

public static javax.swing.JTextField phone1;

public static javax.swing.JTextField username;

//public javax.swing.JLabel jLabel1;

// Variables declaration - do not modify

private javax.swing.JTextField LINK;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JTextArea jTextArea1;

private javax.swing.JButton login;

private javax.swing.JTextArea newname;

private javax.swing.JButton news;

private javax.swing.JButton register;

private javax.swing.JPasswordField txt\_password;

private javax.swing.JTextField txt\_username;

// End of variables declaration

}

**Chapter 6**

**CONCLUSION AND FUTURE SCORE**

**Conclusion**

To create effective question it is important to automate the complete procedure of question paper generation. This system uses searching algorithm and questions can be generated that are relevant to syllabus and generate a question paper according to weight age of each chapter. Thus system simplifies the whole process of question paper generation. And further security can be added in future to system.

**Future Scope**

In future this system can be extended using Natural Language Processing (NLP) for generating questions.NLP can be used in system to make sure that same question will not be generated again and again, and with the help of NLP user can add calculation type and cases type questions in question paper.

**Chapter 7**

**REFERENCES**

[1] Peijiang Chen, ”Construction of Test Question Database for Electrical and Electronic

Technology”, International Conference on Electronic & Mechanical Engineering and Information Technology, IEEE,August 2011.

[2] Syahaneim Marzukhi**,**Muhammad Firdaus Sulaiman. “Automatic Generation Question System to Enhance Preparation of Learning Assessment”.

[3] Fang Yuan, Bo Liu,. “A New Method of Information Extraction from PDF Files”,The Fourth International Conference on Machine learning and Cybernetics, IEEE,August 2005.

[4] Vaibhav M. Kale, Arvind W Kiwelekar. “An algorithm for question paper template generation in question paper generation system”. IEEE, 2013.

[5] Jing Mei Li, Jing Li, Jian Pei Zhang,Nan Ding. “Research on intelligent test paper generation based on multi-variable asymptotic optimization”. IEEE ,2010.

[6] Burcu Yildiz, Katharina Kaiser, Silvia Miksch. PDF2table: “A method Extract table information from PDF files’. IEEE, 2014.

[7]Ashok M., Tulasi .B. “Framework For Automatic Paper Generation System”, International Journal of Computer Applications, IEEE,2014.

[8] S. S. Sheik,† Sumit K. Aggarwal,” A FAST Pattern Matching Algorithm”, Bioinformatics

Centre andSupercomputer Education and Research Centre, Indian Institute of Science, J. Chem. Inf. Comput. Sci, 2004.

[9] Mr. Rahul B. Diwate, Prof. Satish J. Alaspurkar,”Study of Different Algorithms for

Pattern Matching”, International Journal of Advanced Research in Computer Science.

1. Vasile Rus, Brendan Wyse,” Question Generation Shared Task and Evaluation Challenge, Proceedings of the 13th European Workshop on Natural Language.
2. Husam Ali ,Yllias Chali,” Automatic Question Generation from Sentences, University of

Lethbridge, TALN 2010.

[12] S. Chen ,S. Diggavi,” Efficient String Matching Algorithms for Combinatorial Universal

Denoising”. Generation (ENLG), Association for Computational Linguistics, September

2011.