Acknowledgment

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We highly indebted to (**Dr.Eng. Samy Elmokadem** the supervisor of the project) for his guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We highly indebted to (**Dr. Bahi Eldin Mortagi , Eng. Alber Shawky Aziz)** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We indebted also to (**Dr. Waleed Wahba**) for their guidance and consultation in the implementation of the project.

We would like to express our gratitude towards our parents & member of (October 6 university) for their kind co-operation and encouragement which help us in completion of this project.

We would like to express my special gratitude and thanks to industry persons for giving us such attention and time. Our thanks and appreciations also go to my colleague in developing the project and people who have willingly helped us out with their abilities.

Finally, we would like to express our thanks and gratitude to all who helped us to complete this project. We do not forget (**Dr. Moussa Ibrahim the Dean of the College**) to help us with this project and our continuous support.

Contents List

Chapter 1:

Chapter 2:

Chapter 3:

Chapter 4:

Chapter 5:

List of Screens/Figures

1. Project Management

2. System Analysis

3. System Design

4. Implementation

Abstract

Generation of question papers through a question bank is an important activity in learning management systems and educational institutions .the quality of question paper is based on various design constraints such as whether a question paper assesses different skills, whether it covers all units from the syllabus of a course and whether it covers various difficulty levels Preparing the exam questions is very challenging, tedious and time consuming for the instructors.

Thus, with the help of this app that present the solution in form of Automatic Question Paper Generator System (QGS) which enables instructors to submit questions in manual or automatic way from spreadsheet, then use it to generate various types of exams e.g. quizzes – final – midterm, that are consistent with Intendent Learning Outcomes (ILOs) standards.

And Enables students to view previous exams samples. The design process performs the scrutiny and composes the examination paper using an efficient algorithm with a high rate of success. From the entered input, the examination paper will be generated automatically.

the final paper may be stored as ‘Word Document’ files. The system shows characteristics like simple operation, a great interface, good usability, immense security, and high stability along with reliability.

Glossary

Term	Definition
Admin	Add, Edit and Update the information of the users and create exams (midterm, final), assignment and quiz.
Student	The person who can pass the quiz and view the previous examinations
Instructors	The person responsible for the questions in the bank questions through which the final exam and mid-term and the development of assignments.
Quiz	The product of the application that is created through the question bank by the instructor.
Exam	The product of the application that is created through the question bank by the instructor and placed in (docs) or (pdf).
Assignments	The product of the application that is created through the question bank By the instructor.

Chapter 1

1.2 Summary

Question bank is an application designed to collect and index questions according to certain criteria like ILOs and question type, Then generate various types of exams.

The Questions bank is one of the systems used in many universities and educational applications for pre-prepared purposes. The bank of questions is prepared according to the quality system that is called **intended learning outcomes (ILOs).**

Through our project, we tried to use the Questions bank to help the members of the faculty at the university to put the (final, midterm, quizzes, assignments) exams as in the college exam paper.

Doctors can also place assignments in all subjects through the questions bank for each subject. In short, we want to help all doctors develop paper tests and duties according to the quality system (**ILOs**) as well as electronic tests. We are grateful to them for their academic years.

Doctors can allow students to view previous generated exams as a help tool for them to achieve maximum benefit of the course.

1.3 Problem

The bank depends on the questions, to creating questions and storing them in the system in different ways. The system must contain most of these methods to facilitate the use of this system to the user.

The university seeks to raise the level of education and assist its members using technical means during the teaching process. One of the most important means that the university seeks to develop is the method of preparing the exam using these means of technology.

Another of the problems facing the project is to follow the university's quality education system within the university which is named **intended learning outcomes (ILOs).**

One of the problems that we encountered during the project implementation is the design of the exam paper and how to make it identical to the examination paper for the entire College of Information Systems and Computer Science.

Some university doctors prefer to put several models of exams and this wastes a very large time in preparing it.

1.4 Purpose

The main purpose of the project is to change the way of creating exams in accordance with the thought of the university in making education easier for all members of the educational system.

The university seeks to promote the system of education within the university and from this point of view we began to inquire about the development of the system of examinations within the university.

This is done through the development of a bank questions for each educational course within the university that makes it easy for university doctors to take the exam easily.
This system enables university doctors to put examinations through the system of quality education within the university, which is called **intended learning outcomes (ILOs).**

The goal of this project is not only to change the system of the exam paper, but also to develop a system that allows the university doctors to demonstrate the Quiz and also allows the assignment of duties through a special paper.

In short, we want to develop a complete system that helps university to develop paper exams (final, midterm) and (Quiz) and the assignment by randomly selecting questions from the bank questions that we have prepared in advance.

1.5 Objectives

The first objective of the questions bank Process is identifying standards for establishing a bank of questions and determining corresponding control measures.

Second Objective and one of the most important objectives of the project is the development of the question bank through the system of quality **intended learning outcomes (ILOs)** and the development of an assistant to know the type of question.

Eeducational material is divided into a group of chapters to facilitate the user to easily develop the exam by selecting the parts that he wants to be present in the exam.

The Third objective is generating the exams in a form of docx file to easily modified it after generation.

The exam is being developed according to the quality system **intended learning outcomes (ILOs)**. Also, by choosing the number of questions and the type of each question and the number of points within each question. The exam is randomly generated by the bank of questions which choosing by instructors before working for the exam.

The main objective of this application is to compile the questions of materials within one bank and use them in the development of exams that are established within the university (the final and midterm) and also the quizzes that will be on the same application and it is also possible to do the homework on this site and print them for delivery to students.

The final objective is to assist students in obtaining their previous exams by allowing them to see these exams. Also, through this, the student can take a quiz exam on the application.

1.6 Figure 1.1 Time Plan

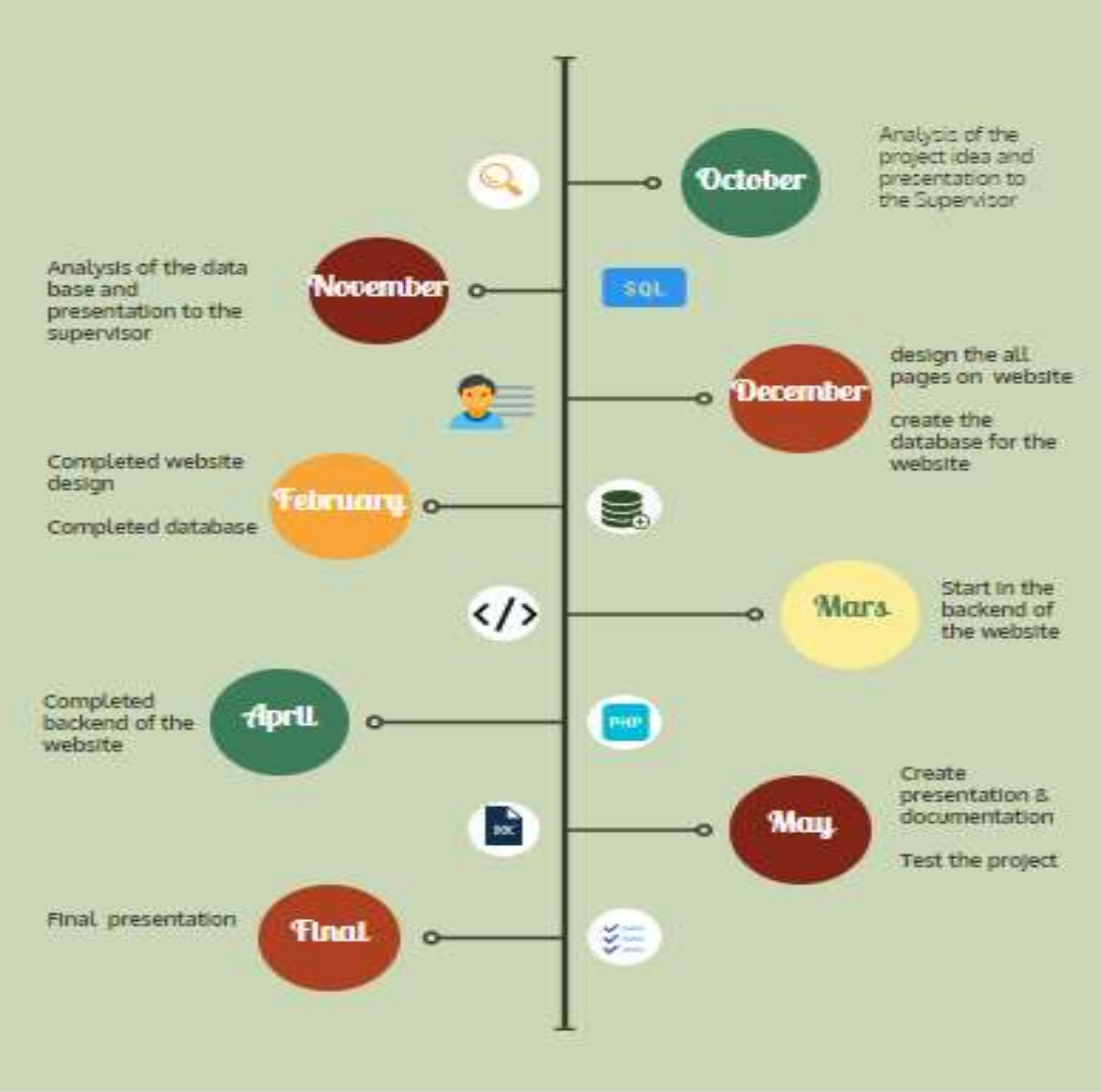
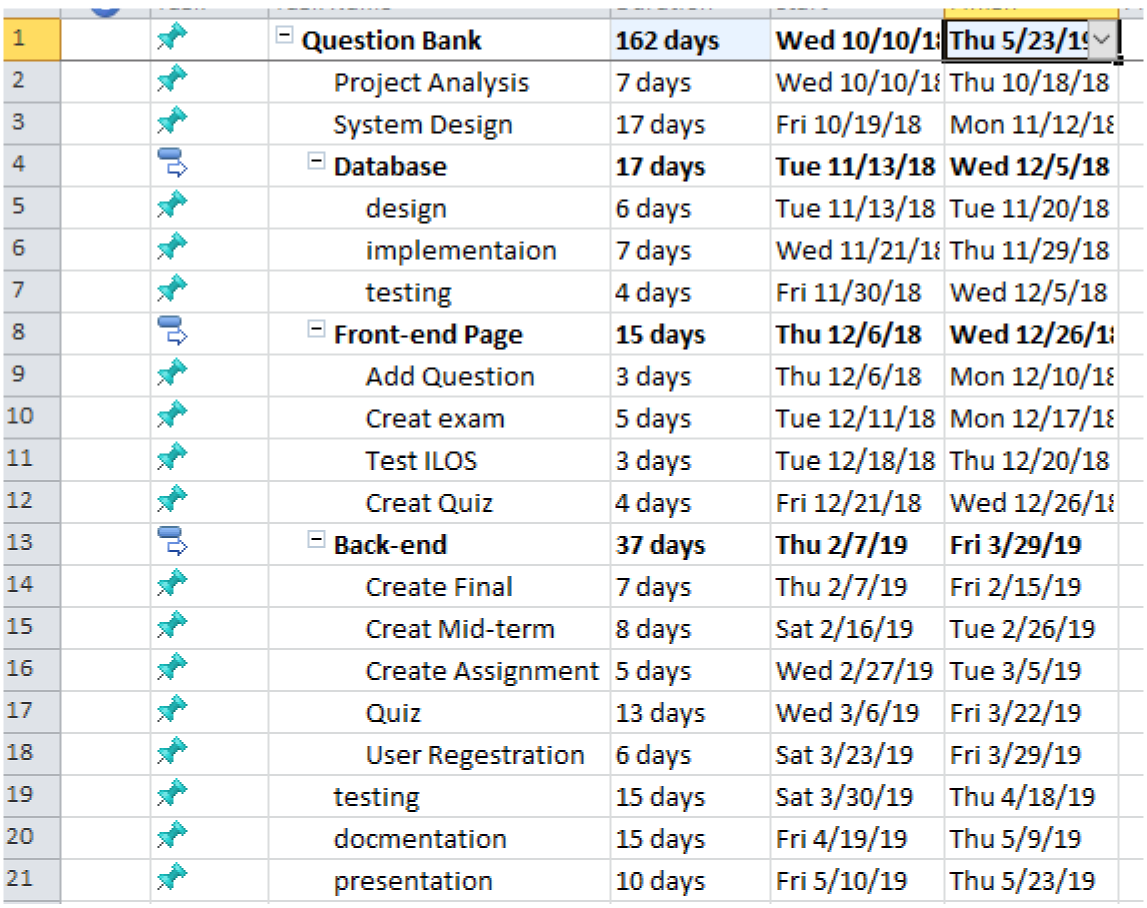
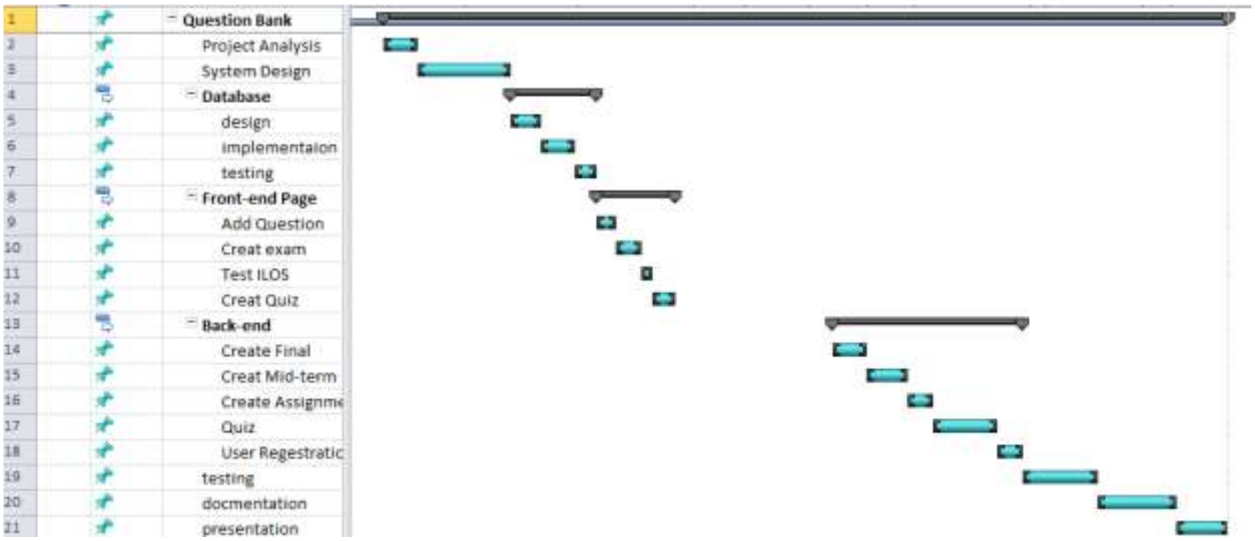

October
Analysis of the project idea and presentation to the Supervisor
November
Analysis of the data base and presentation to the supervisor
SQL
December
design the all pages on website
create the database for the website
February
Completed website design
Completed database
Mars
Start in the backend of the website
April
Completed backend of the website
PHP
May
Create presentation & documentation
Test the project
DOC
Final
Final presentation

Figure 1.2 Time Plan [On Microsoft Project]



1	Question Bank	162 days	Wed 10/10/1	Thu 5/23/1
2	Project Analysis	7 days	Wed 10/10/18	Thu 10/18/18
3	System Design	17 days	Fri 10/19/18	Mon 11/12/18
4	Database	17 days	Tue 11/13/18	Wed 12/5/18
5	design	6 days	Tue 11/13/18	Tue 11/20/18
6	implementaion	7 days	Wed 11/21/18	Thu 11/29/18
7	testing	4 days	Fri 11/30/18	Wed 12/5/18
8	Front-end Page	15 days	Thu 12/6/18	Wed 12/26/1
9	Add Question	3 days	Thu 12/6/18	Mon 12/10/18
10	Creat exam	5 days	Tue 12/11/18	Mon 12/17/18
11	Test ILOS	3 days	Tue 12/18/18	Thu 12/20/18
12	Creat Quiz	4 days	Fri 12/21/18	Wed 12/26/18
13	Back-end	37 days	Thu 2/7/19	Fri 3/29/19
14	Create Final	7 days	Thu 2/7/19	Fri 2/15/19
15	Creat Mid-term	8 days	Sat 2/16/19	Tue 2/26/19
16	Create Assignment	5 days	Wed 2/27/19	Tue 3/5/19
17	Quiz	13 days	Wed 3/6/19	Fri 3/22/19
18	User Regestration	6 days	Sat 3/23/19	Fri 3/29/19
19	testing	15 days	Sat 3/30/19	Thu 4/18/19
20	docmentation	15 days	Fri 4/19/19	Thu 5/9/19
21	presentation	10 days	Fri 5/10/19	Thu 5/23/19

Figure 1.3 Time Plan [On Microsoft Project]



1	Question Bank
2	Project Analysis
3	System Design
4	Database
5	design
6	implementaion
7	testing
8	Front-end Page
9	Add Question
10	Creat exam
11	Test ILOS
12	Creat Quiz
13	Back-end
14	Create Final
15	Creat Mid-term
16	Create Assignme
17	Quiz
18	User Regestratic
19	testing
20	docmentation
21	presentation

Chapter 2

2.1 User Requirements

Functional Requirements

- **Admin:**
  - The system contains some admins, and each have different privileges
  - Admin has only username one password
  - Admin can export list of admins in the system if he has the credentials

- **Users:**
  - The system has depended on the instructor who has interested in making exam online.
  - While registrations we save for each instructor his/her name, email, password, subject, Questions and previous exam.
  - While registrations we save for each student his/her name, email, password and previous exam.
  - Admin has the ability to search about specific participant or even filter them
  - Admin can control the user [EDIT AND DELETE]
  - users can register.

- **Question bank:**
  - The system of Question bank depends on the question will instructor add it.
  - Each question has address, type, subject, number and possible answer
  - Admin can control the questions [EDIT, DELETE AND ADD]

- **Subject:**
  - Each subject has name, Id, Questions, instructor name, chapters and year.
  - Instructor who responsible to addition this Question
  - Admin can control the subject [EDIT, DELETE AND ADD]

- **Final exam system:**

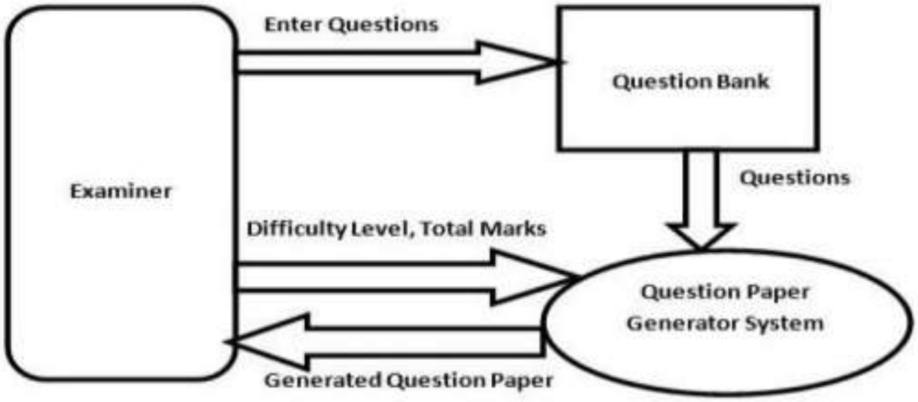


Figure 2.1 Process of Final Exam System

- Instructor can make a final exam from the Questions bank in any time
- Each final exam has subject, date, id, year, degree, Questions, credit hour and semester.
- Admin can control the final exam [EDIT, DELETE AND ADD].
- Student can view the previous exam only.

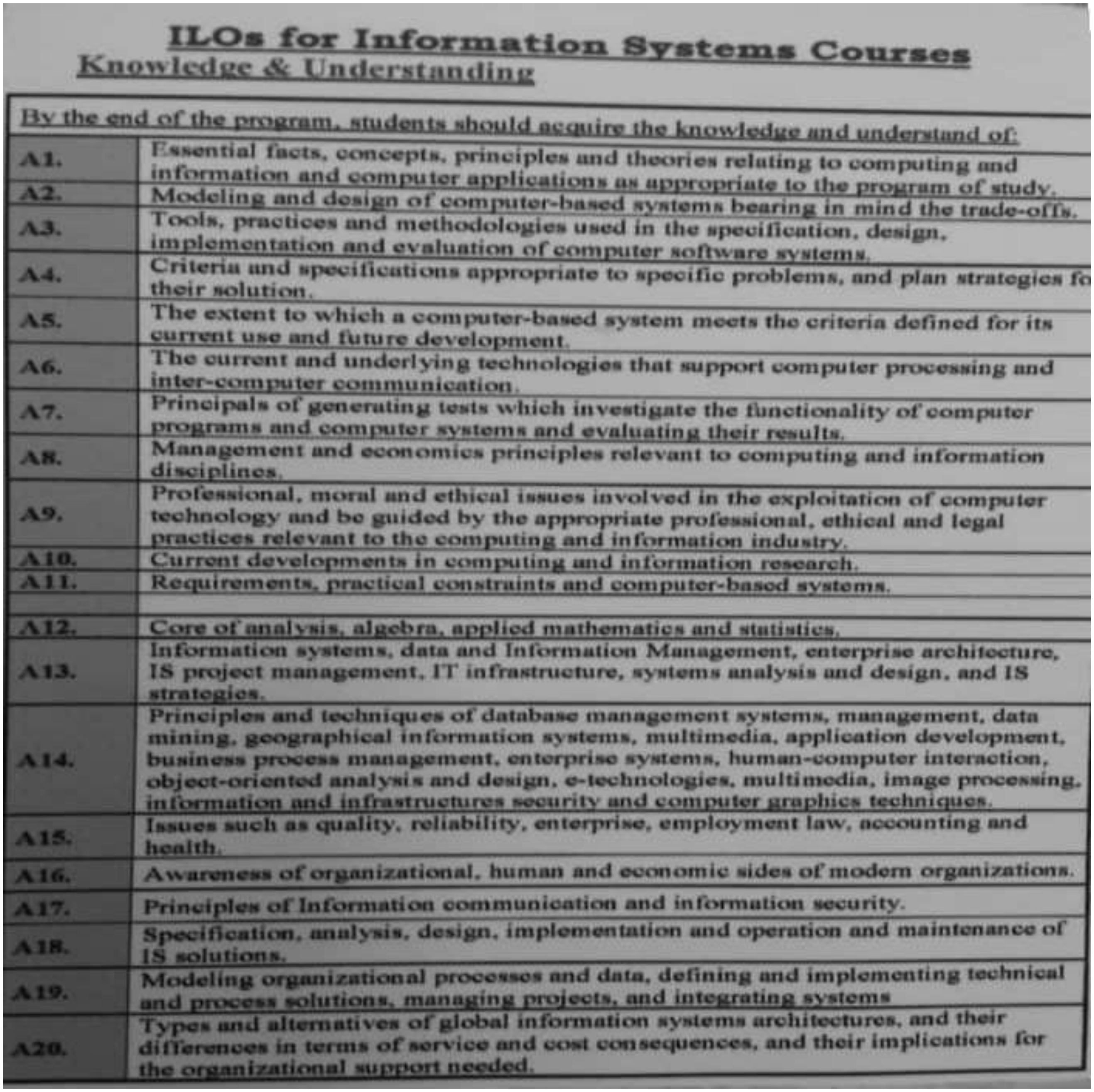
- **Mid-term exam system:**
    - Instructor can make a mid-term exam from the Questions bank in any time.
    - Each final exam has subject, date, id, year, degree, Questions, credit hour and semester.
    - Admin can control the mid-term exam [EDIT, DELETE AND ADD].
    - Student can view the previous exam only.

- **Assignments system:**
    - Instructor can make an assignment from the Questions bank in any time.
    - Each final exam has subject, date, chapter and Questions.
    - Admin can control the mid-term exam [EDIT, DELETE AND ADD].
    - Student can view the previous exam only.

- **[ILOs] system:**
    - Instructor who responsible to addition this **intended learning outcomes (ILOs)** for any questions.
    - Instructor can test the [ILOs] type for any questions.

Intended Learning Outcomes

1st: Figure 2.2: Knowledge and Understanding



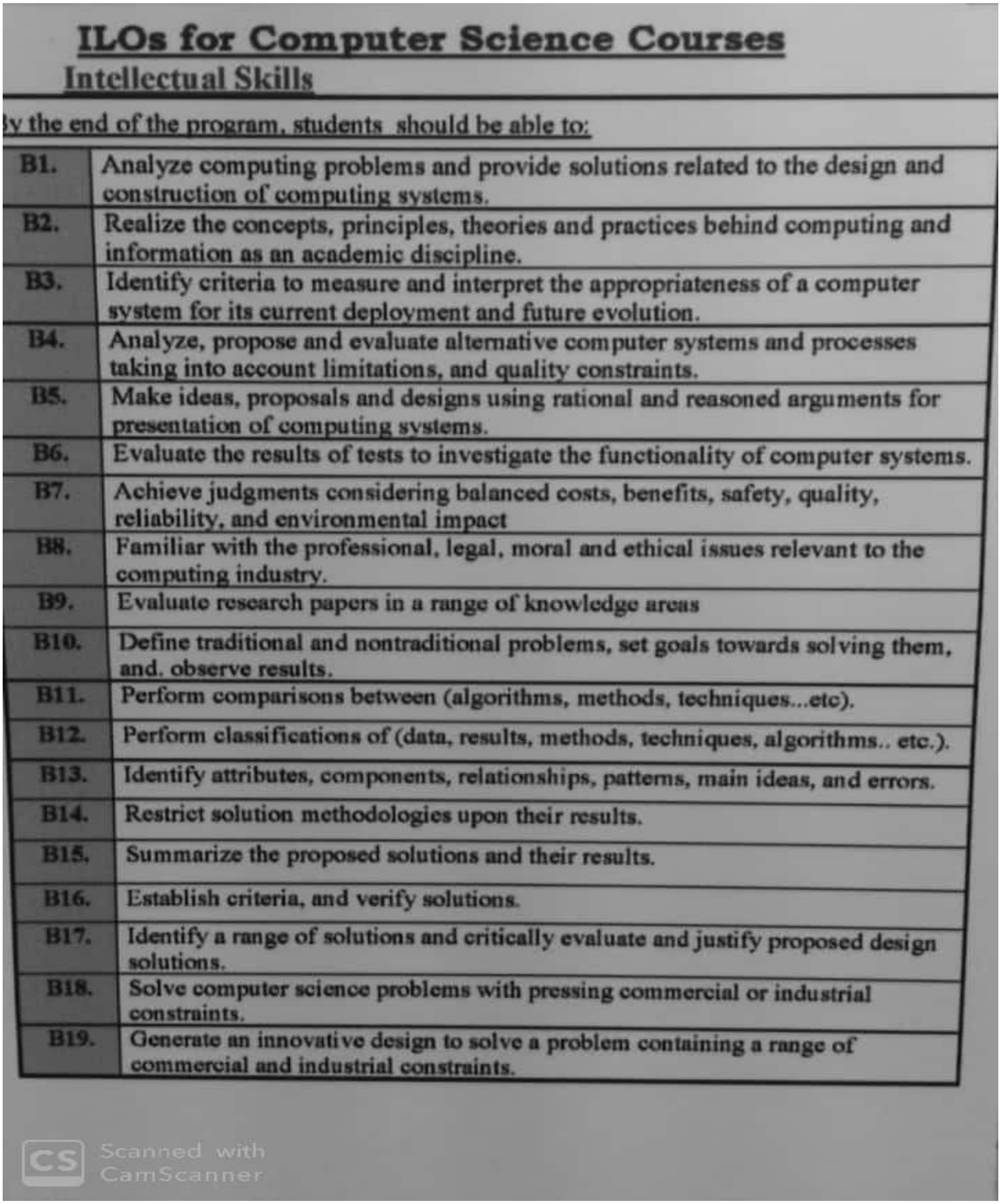
ILOs for Information Systems Courses

Knowledge & Understanding

By the end of the program, students should acquire the knowledge and understand of:	
A1.	Essential facts, concepts, principles and theories relating to computing and information and computer applications as appropriate to the program of study.
A2.	Modeling and design of computer-based systems bearing in mind the trade-offs.
A3.	Tools, practices and methodologies used in the specification, design, implementation and evaluation of computer software systems.
A4.	Criteria and specifications appropriate to specific problems, and plan strategies fo their solution.
A5.	The extent to which a computer-based system meets the criteria defined for its current use and future development.
A6.	The current and underlying technologies that support computer processing and inter-computer communication.
A7.	Principals of generating tests which investigate the functionality of computer programs and computer systems and evaluating their results.
A8.	Management and economics principles relevant to computing and information disciplines.
A9.	Professional, moral and ethical issues involved in the exploitation of computer technology and be guided by the appropriate professional, ethical and legal practices relevant to the computing and information industry.
A10.	Current developments in computing and information research.
A11.	Requirements, practical constraints and computer-based systems.
A12.	Core of analysis, algebra, applied mathematics and statistics.
A13.	Information systems, data and Information Management, enterprise architecture, IS project management, IT infrastructure, systems analysis and design, and IS strategies.
A14.	Principles and techniques of database management systems, management, data mining, geographical information systems, multimedia, application development, business process management, enterprise systems, human-computer interaction, object-oriented analysis and design, e-technologies, multimedia, image processing, information and infrastructures security and computer graphics techniques.
A15.	Issues such as quality, reliability, enterprise, employment law, accounting and health.
A16.	Awareness of organizational, human and economic sides of modern organizations.
A17.	Principles of Information communication and information security.
A18.	Specification, analysis, design, implementation and operation and maintenance of IS solutions.
A19.	Modeling organizational processes and data, defining and implementing technical and process solutions, managing projects, and integrating systems
A20.	Types and alternatives of global information systems architectures, and their differences in terms of service and cost consequences, and their implications for the organizational support needed.

Intended Learning Outcomes

2nd: Figure 2.3: Intellectual Skills



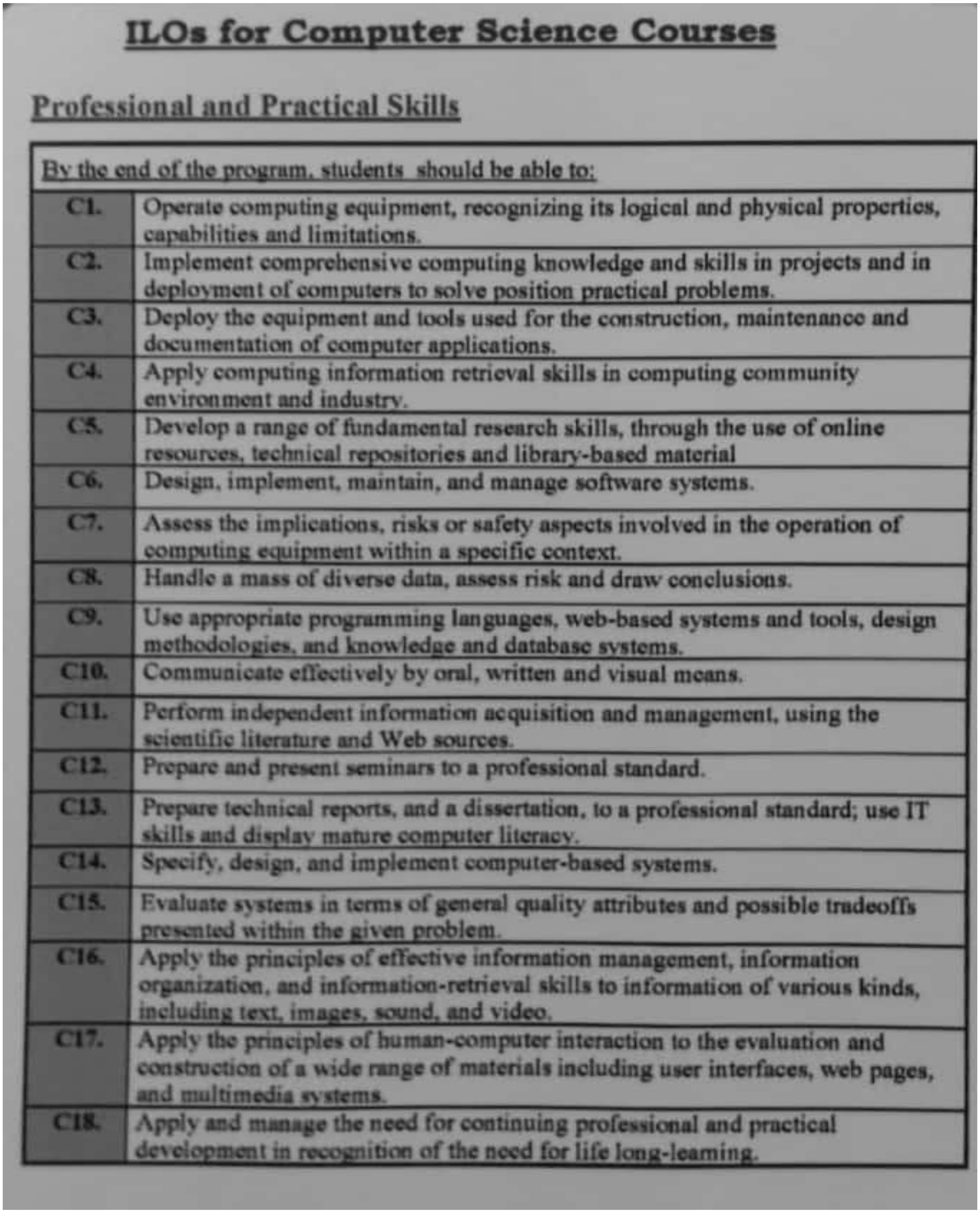
ILOs for Computer Science Courses

Intellectual Skills

y the end of the program, students should be able to:

B1.	Analyze computing problems and provide solutions related to the design and construction of computing systems.
B2.	Realize the concepts, principles, theories and practices behind computing and information as an academic discipline.
B3.	Identify criteria to measure and interpret the appropriateness of a computer system for its current deployment and future evolution.
B4.	Analyze, propose and evaluate alternative computer systems and processes taking into account limitations, and quality constraints.
B5.	Make ideas, proposals and designs using rational and reasoned arguments for presentation of computing systems.
B6.	Evaluate the results of tests to investigate the functionality of computer systems.
B7.	Achieve judgments considering balanced costs, benefits, safety, quality, reliability, and environmental impact
B8.	Familiar with the professional, legal, moral and ethical issues relevant to the computing industry.
B9.	Evaluate research papers in a range of knowledge areas
B10.	Define traditional and nontraditional problems, set goals towards solving them, and. observe results.
B11.	Perform comparisons between (algorithms, methods, techniques...etc).
B12.	Perform classifications of (data, results, methods, techniques, algorithms.. etc.).
B13.	Identify attributes, components, relationships, patterns, main ideas, and errors.
B14.	Restrict solution methodologies upon their results.
B15.	Summarize the proposed solutions and their results.
B16.	Establish criteria, and verify solutions.
B17.	Identify a range of solutions and critically evaluate and justify proposed design solutions.
B18.	Solve computer science problems with pressing commercial or industrial constraints.
B19.	Generate an innovative design to solve a problem containing a range of commercial and industrial constraints.

Intended Learning Outcomes
3rd: Figure 2.4: Practical Skills



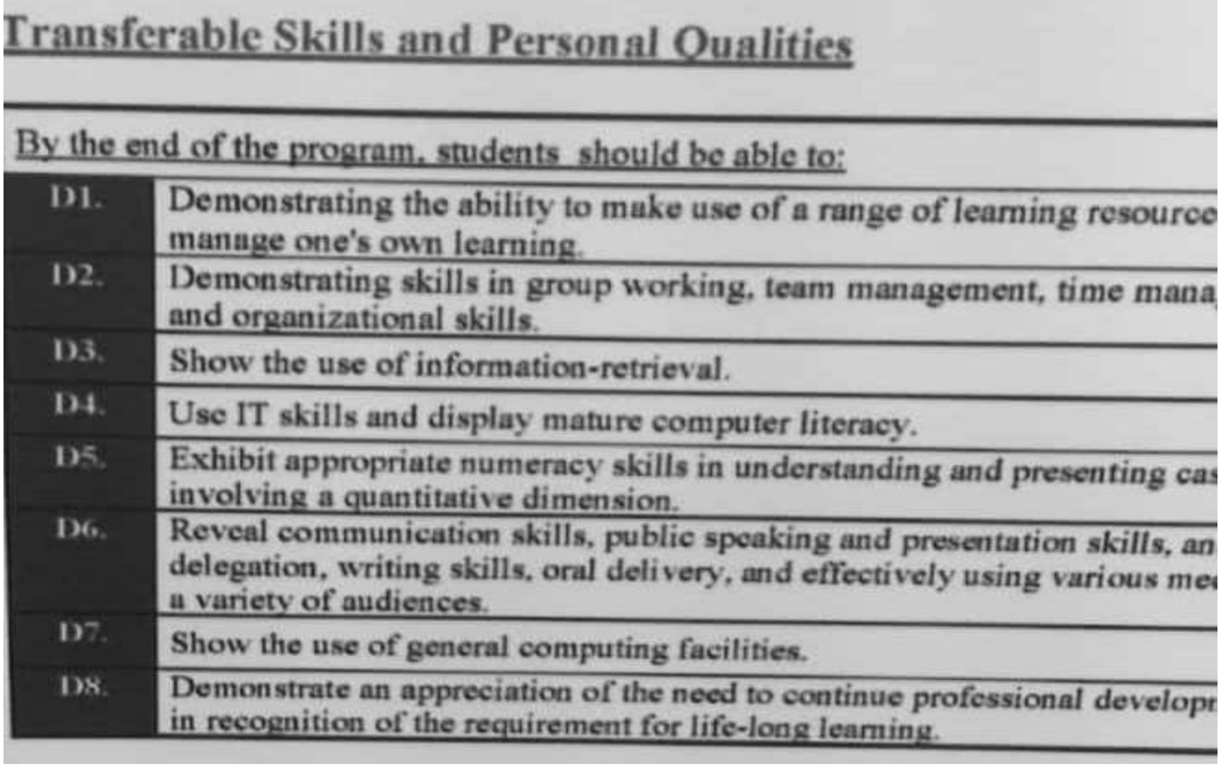
ILOs for Computer Science Courses

Professional and Practical Skills

By the end of the program, students should be able to:	
C1.	Operate computing equipment, recognizing its logical and physical properties, capabilities and limitations.
C2.	Implement comprehensive computing knowledge and skills in projects and in deployment of computers to solve position practical problems.
C3.	Deploy the equipment and tools used for the construction, maintenance and documentation of computer applications.
C4.	Apply computing information retrieval skills in computing community environment and industry.
C5.	Develop a range of fundamental research skills, through the use of online resources, technical repositories and library-based material
C6.	Design, implement, maintain, and manage software systems.
C7.	Assess the implications, risks or safety aspects involved in the operation of computing equipment within a specific context.
C8.	Handle a mass of diverse data, assess risk and draw conclusions.
C9.	Use appropriate programming languages, web-based systems and tools, design methodologies, and knowledge and database systems.
C10.	Communicate effectively by oral, written and visual means.
C11.	Perform independent information acquisition and management, using the scientific literature and Web sources.
C12.	Prepare and present seminars to a professional standard.
C13.	Prepare technical reports, and a dissertation, to a professional standard; use IT skills and display mature computer literacy.
C14.	Specify, design, and implement computer-based systems.
C15.	Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.
C16.	Apply the principles of effective information management, information organization, and information-retrieval skills to information of various kinds, including text, images, sound, and video.
C17.	Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, and multimedia systems.
C18.	Apply and manage the need for continuing professional and practical development in recognition of the need for life long-learning.

Intended Learning Outcomes

4th: Figure 2.5: Transferable Skill



Transferable Skills and Personal Qualities

By the end of the program, students should be able to:

D1.	Demonstrating the ability to make use of a range of learning resource manage one's own learning.
D2.	Demonstrating skills in group working, team management, time mana and organizational skills.
D3.	Show the use of information-retrieval.
D4.	Use IT skills and display mature computer literacy.
D5.	Exhibit appropriate numeracy skills in understanding and presenting cas involving a quantitative dimension.
D6.	Reveal communication skills, public speaking and presentation skills, an delegation, writing skills, oral delivery, and effectively using various me a variety of audiences.
D7.	Show the use of general computing facilities.
D8.	Demonstrate an appreciation of the need to continue professional develop in recognition of the requirement for life-long learning.

2.2 System Requirements

System requirements are all of the requirements at the system level that describe the functions which the system as a whole should fulfill to satisfy the stakeholder needs and requirements, and is expressed in an appropriate combination of textual statements, views, and non-functional requirements; the latter expressing the levels of safety, security, reliability, etc., that will be necessary.

System requirements play major roles in systems engineering, so we will discuss in this section the system requirements such as:

- **Performance Requirements:**
  - The system and each functionality implemented in way which has the shortest time to perform tasks
  - The power of sending and receiving requests through the server is the important perspective, as the system design to serve thousands of people
  - Application designed with caching the data in order to benefit from the previous loaded data
- **Usability Requirements:**
  - The system designed using high levels of user control which help actors to easily use the system
  - The way of sending and reading message designed in effective UX method to serve the usability of the system
- **Interface Requirements:**
  - Application front-end developers choose the colors of dashboard which suitable for the eyes and will be comfortable for all of the users
- **Operational Requirements:**
  - Easy way in data entering, querying and filtering the data
  - Admin can perform the tasks in dashboard without any knowledge
  - Each part of the system designed to be easily operational and both participants and the admins can use the system as the use the applications they are usually use.
- **System Security:**
  - Back end server(s), including data encryption and transmission
  - Administrator controlled username and password access
  - Information Security Industry Standard encryption and SSL certifications (currently 128-bit)
  - Automatic timeout/log-off
  - Privileges check system for both admin

- Integrated technical safeguards to ensure a high level of privacy and security

- System Validation:

- In data entering phase the system ensures the validation of data such as email and password constrains
- Input masks
- Record data completeness requirements

- Communications Requirements:

- Communication between the User and the system will be through the worldwide web

2.3 System Analysis

In this chapter we will discuss each diagram in the systems such as:

- **Sequence Diagram**

  The sequence diagram captures the time sequence of the message flow from one object to another

- **Use case Diagram**

  To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating.

- **Class Diagram**

  Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application

2.4 Sequence Diagram

ADMIN FUNCTIONS [Figure 2.6, 2.7]

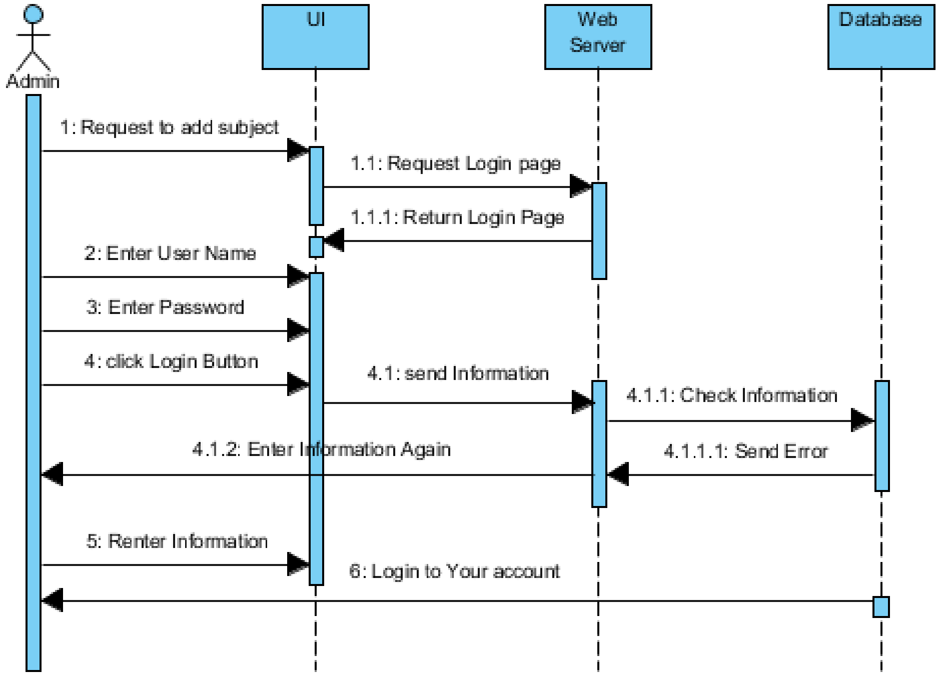


FIGURE 2.6: ADMIN LOGIN

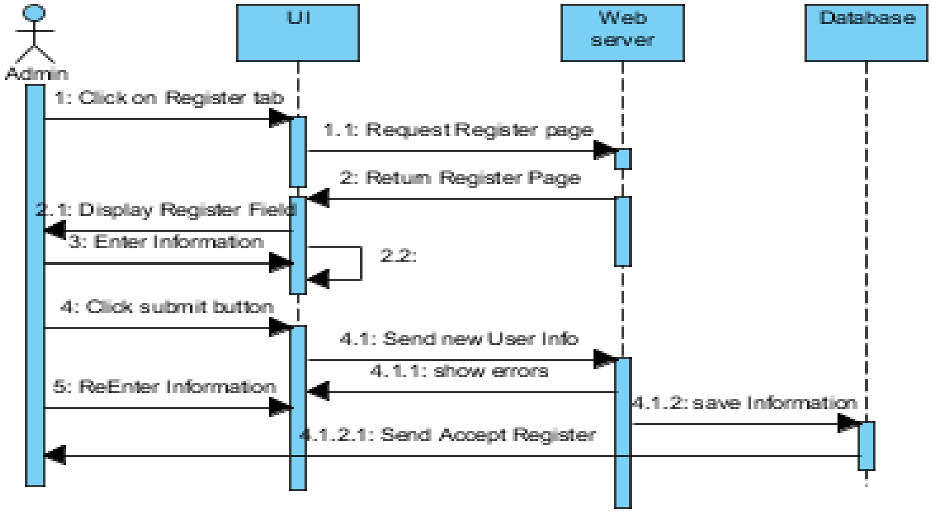


FIGURE 2.7: ADMIN REGISTER

2.4 Sequence Diagram
ADMIN FUNCTIONS [Figure 2.8 ,2.9]

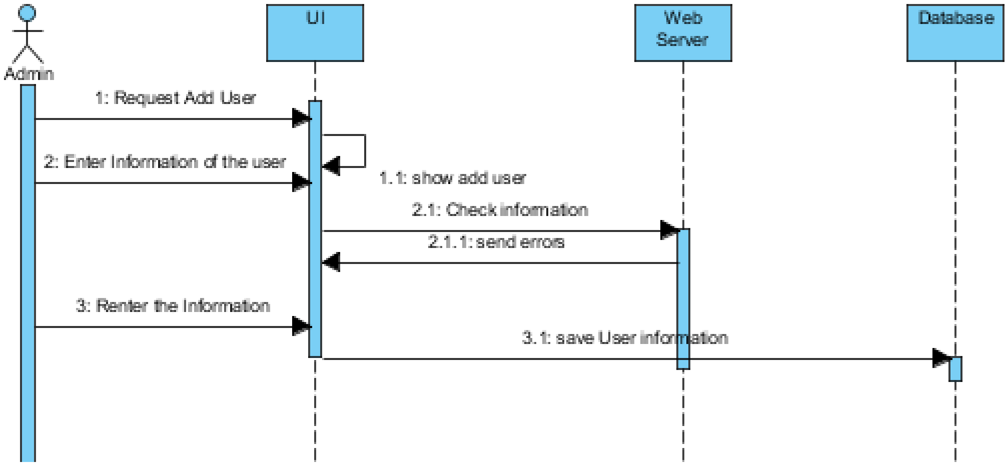


FIGURE 2.8: ADMIN ADD USER

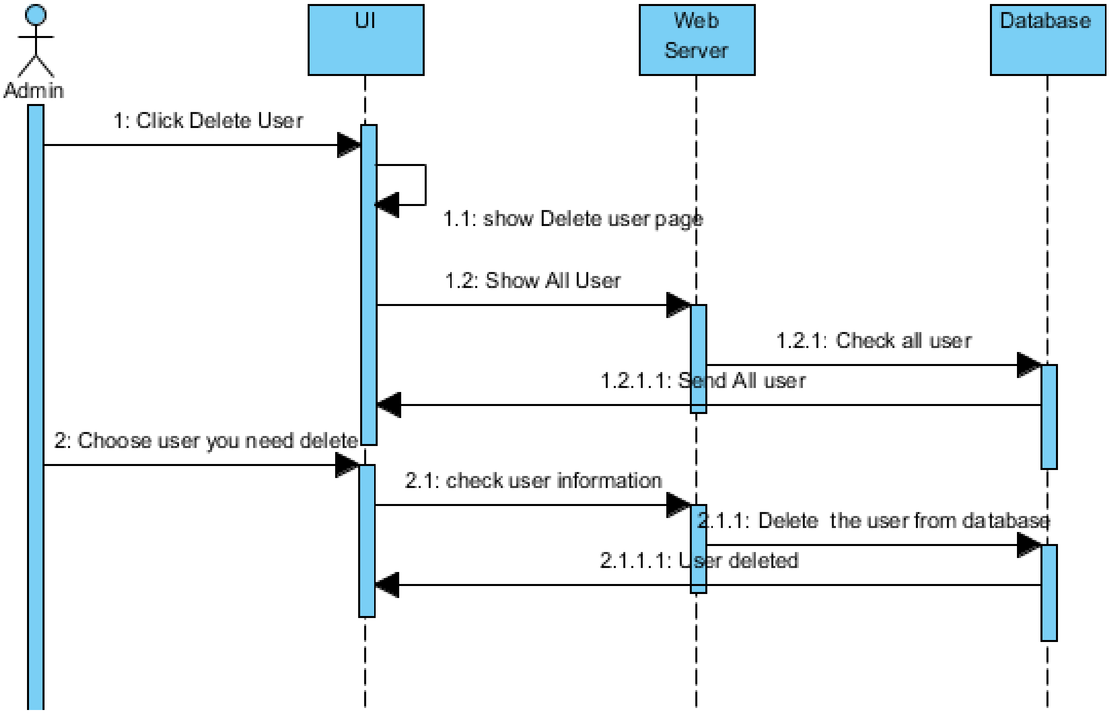


FIGURE 2.9: ADMIN DELETE USER

Sequence Diagram
Figure 2.10 ,2.11 ADMIN FUNCTIONS

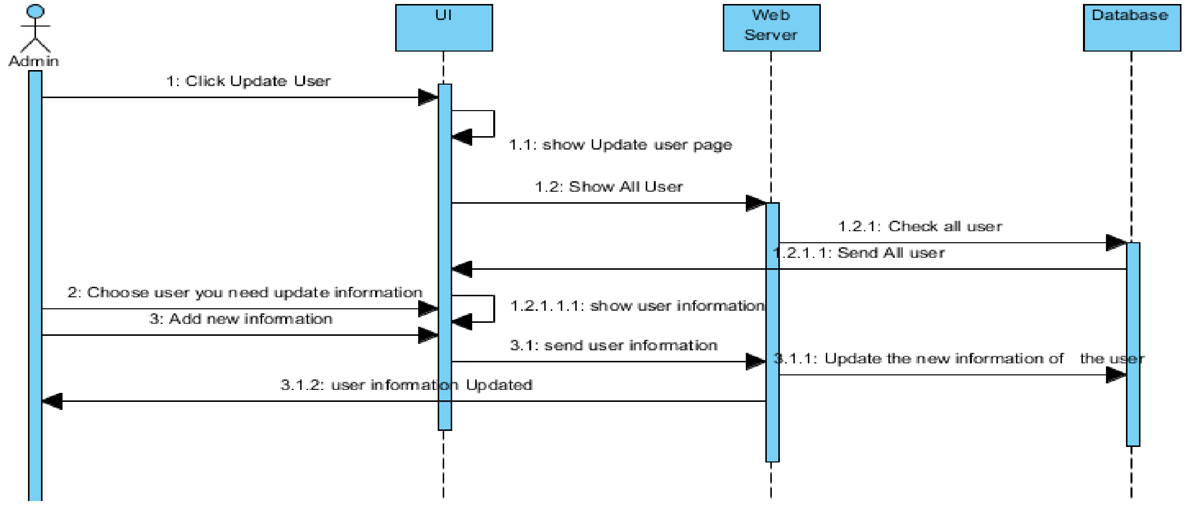

Admin
UI
Web Server
Database
1: Click Update User
1.1: show Update user page
1.2: Show All User
1.2.1: Check all user
1.2.1.1: Send All user
2: Choose user you need update information
1.2.1.1.1: show user information
3: Add new information
3.1: send user information
3.1.1: Update the new information of the user
3.1.2: user information Updated

FIGURE 2.10: ADMIN UPDATE USER

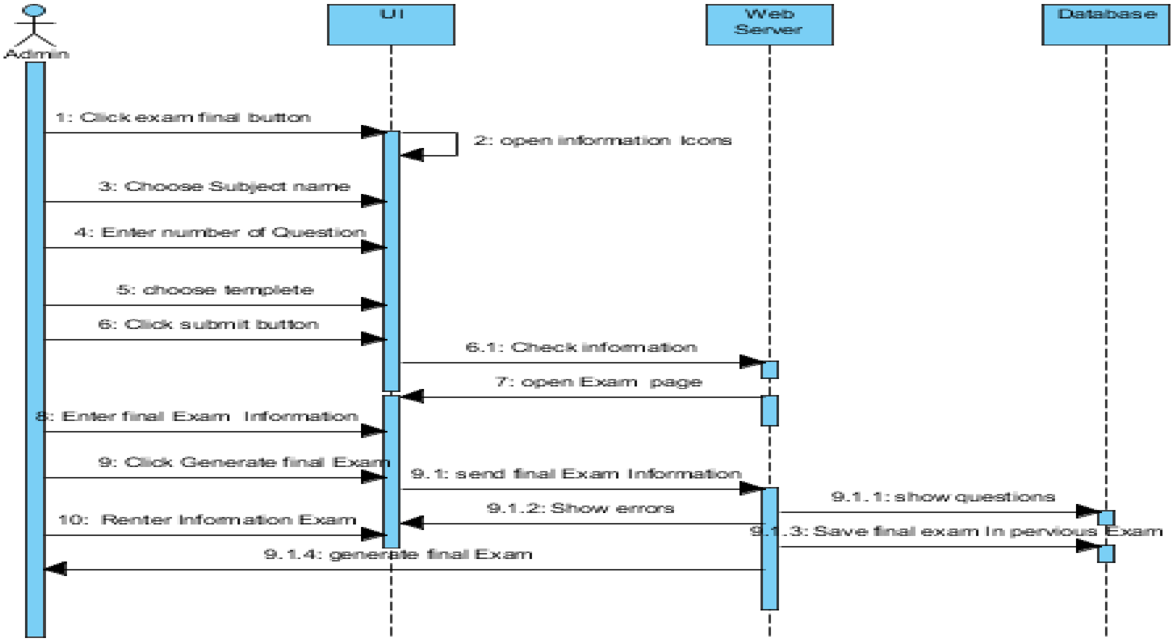

Admin
UI
Web Server
Database
1: Click exam final button
2: open information Icons
3: Choose Subject name
4: Enter number of Question
5: choose templete
6: Click submit button
6.1: Check infomation
7: open Exam page
8: Enter final Exam Information
9: Click Generate final Exam
9.1: send final Exam Information
9.1.1: show questions
9.1.2: Show errors
10: Renter Infomation Exam
9.1.3: Save final exam In pervious Exam
9.1.4: generate final Exam

FIGURE 2.11: ADMIN FINAL EXAM

Sequence Diagram

Figure 2.12 ,2.13 ADMIN FUNCTIONS

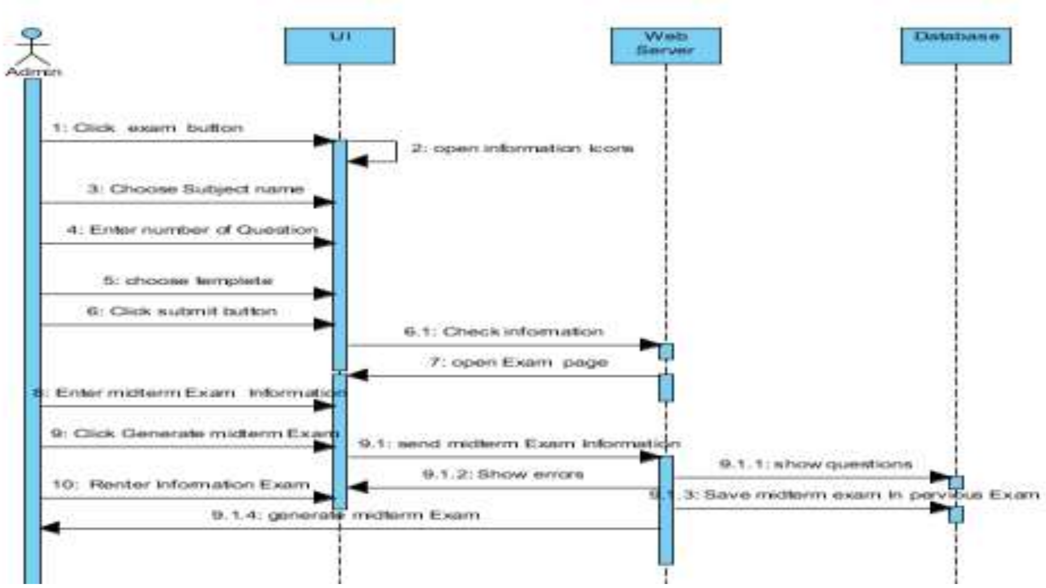

Admin
UI
Web Server
Database
1: Click exam button
2: open information Icons
3: Choose Subject name
4: Enter number of Question
5: choose templete
6: Click submit button
6.1: Check infomation
7: open Exam page
8: Enter midterm Exam Information
9: Click Generate midterm Exam
9.1: send midterm Exam Information
9.1.1: show questions
9.1.2: Show errors
10: Renter Information Exam
9.1.3: Save midterm exam in pervious Exam
9.1.4: generate midterm Exam

FIGURE 2.12: ADMIN MIDTERM EXAM

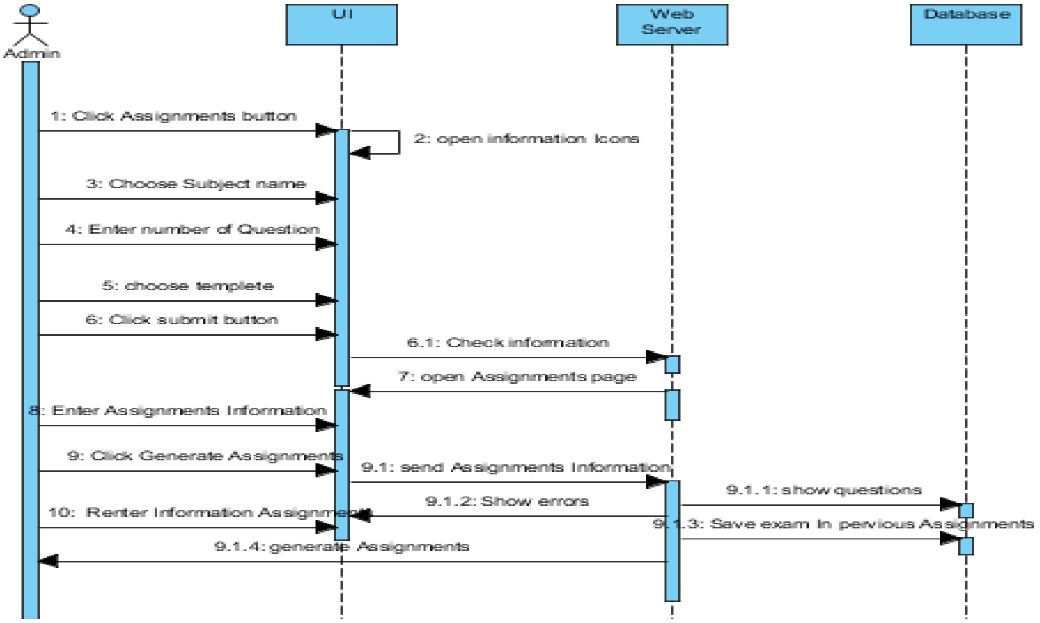

Admin
UI
Web Server
Database
1: Click Assignments button
2: open information Icons
3: Choose Subject name
4: Enter number of Question
5: choose templete
6: Click submit button
6.1: Check information
7: open Assignments page
8: Enter Assignments Information
9: Click Generate Assignments
9.1: send Assignments Information
9.1.1: show questions
9.1.2: Show errors
10: Renter Information Assignments
9.1.3: Save exam in pervious Assignments
9.1.4: generate Assignments

FIGURE 2.13: ADMIN ASSIGNMENTS

Sequence Diagram
Figure 2.14 ADMIN FUNCTIONS

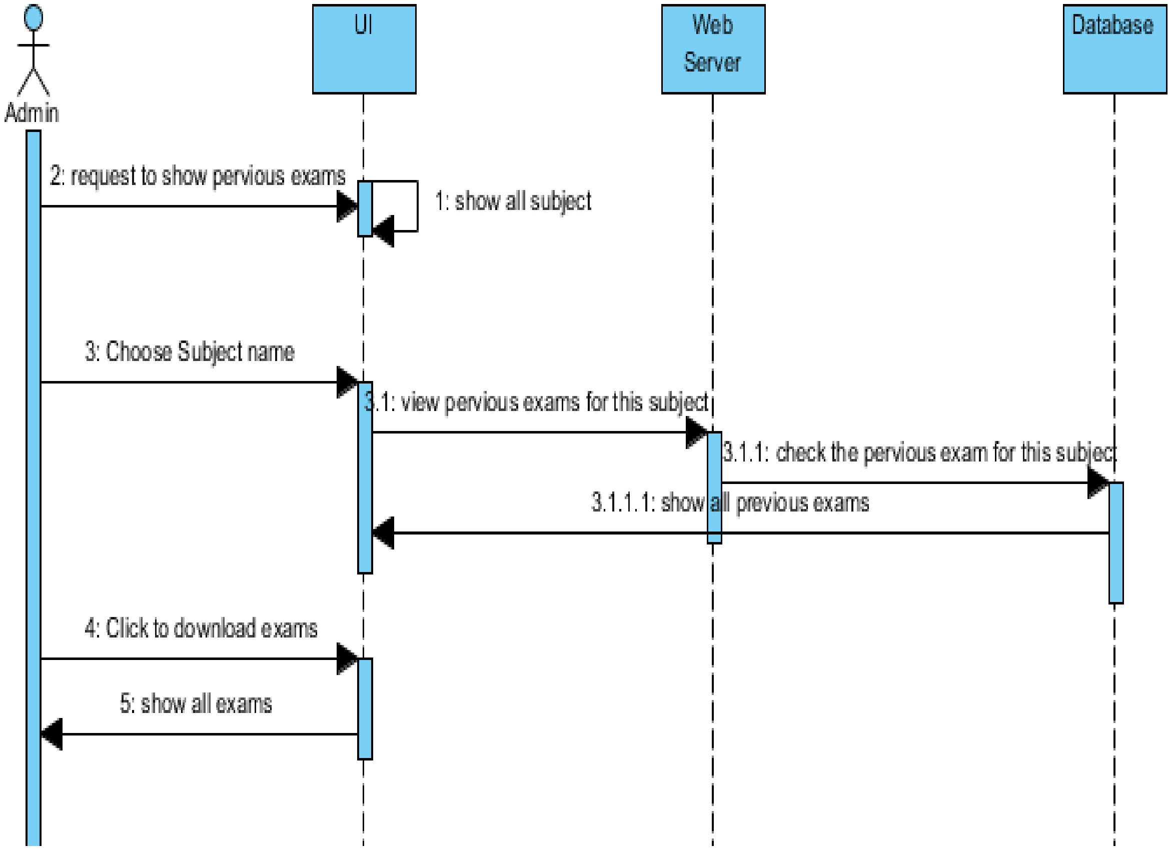

Admin
UI
Web Server
Database
2: request to show pervious exams
1: show all subject
3: Choose Subject name
3.1: view pervious exams for this subject
3.1.1: check the pervious exam for this subject
3.1.1.1: show all previous exams
4: Click to download exams
5: show all exams

FIGURE 2.14: ADMIN VIEW PERVIOUS EXAM

Sequence Diagram
Figure 2.15 INSTRUCTOR FUNCTIONS

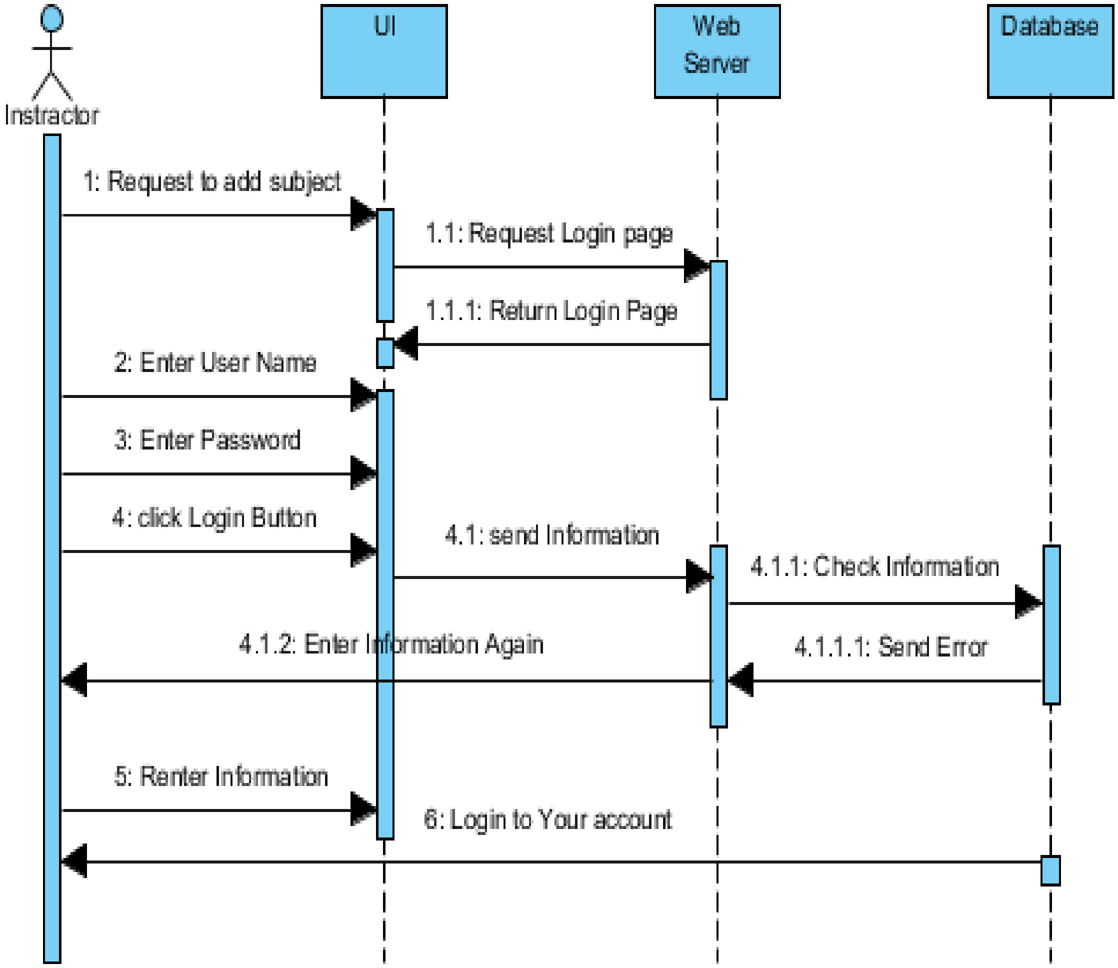


FIGURE 2.15: INSTRUCTOR LOGIN

Sequence Diagram
Figure 2.16, 2.17 INSTRUCTOR FUNCTIONS

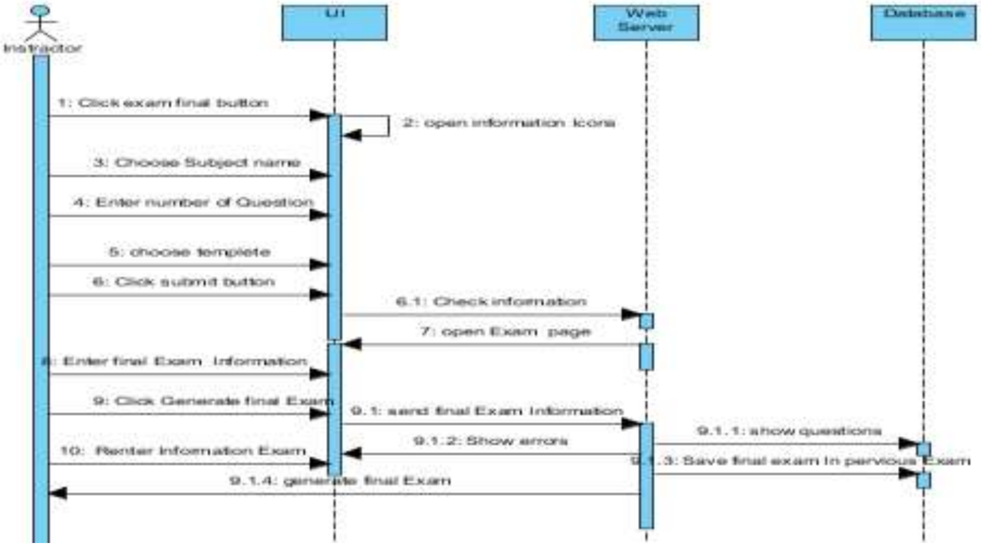

Instractor
UI
Web Server
Database
1: Click exam final button
2: open information Icons
3: Choose Subject name
4: Enter number of Question
5: choose templete
6: Click submit button
6.1: Check information
7: open Exam page
8: Enter final Exam Information
9: Click Generate final Exam
9.1: send final Exam Information
9.1.1: show questions
9.1.2: Show errors
9.1.3: Save final exam In pervious Exam
10: Renter Information Exam
9.1.4: generate final Exam

FIGURE 2.16:INSTRUCTOR FINAL EXAM

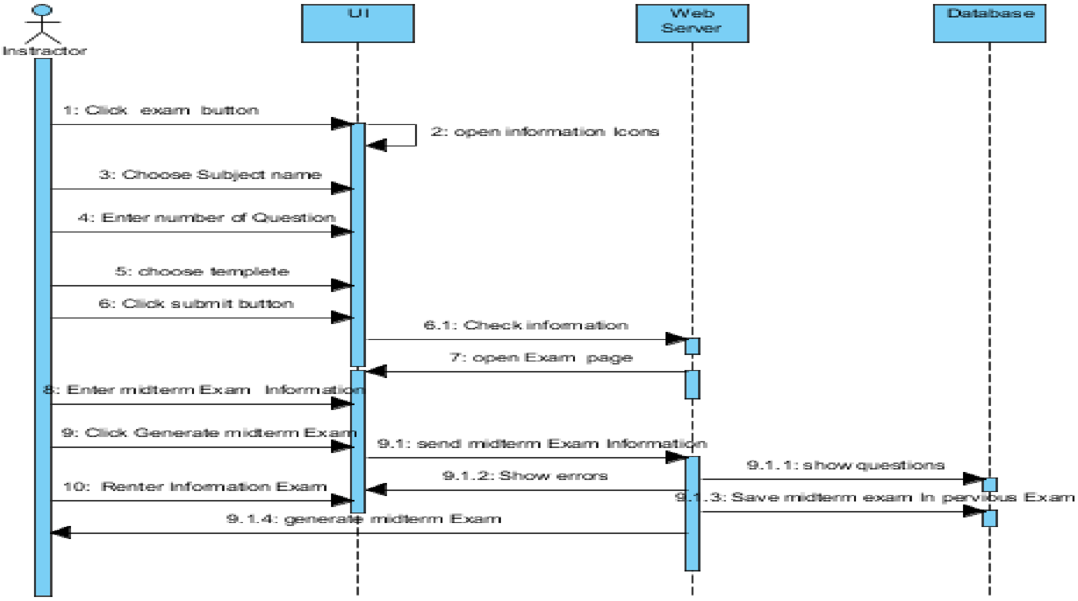

Instractor
UI
Web Server
Database
1: Click exam button
2: open information Icons
3: Choose Subject name
4: Enter number of Question
5: choose templete
6: Click submit button
6.1: Check information
7: open Exam page
8: Enter midterm Exam Information
9: Click Generate midterm Exam
9.1: send midterm Exam Information
9.1.1: show questions
9.1.2: Show errors
9.1.3: Save midterm exam In pervious Exam
10: Renter Information Exam
9.1.4: generate midterm Exam

FIGURE 2.17: INSTRUCTOR MIDTERM EXAM

Sequence Diagram

Figure 2.18 INSTRUCTOR FUNCTIONS

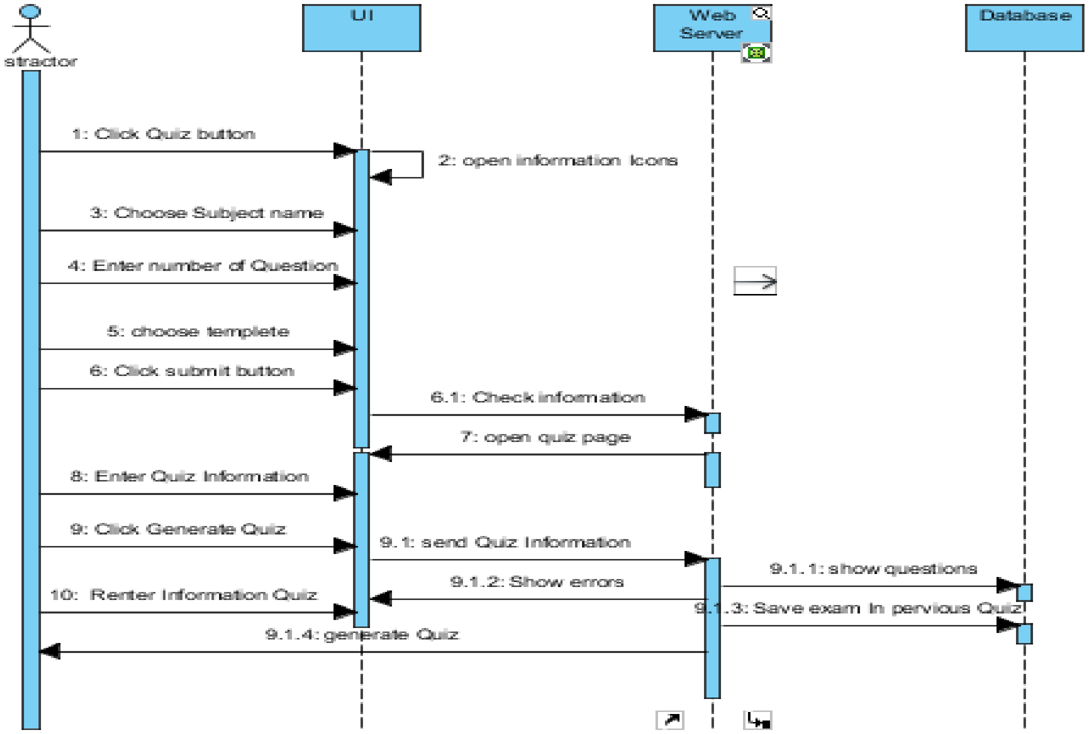


FIGURE 2.18: INSTRUCTOR QUIZ EXAM

Sequence Diagram
Figure 2.19, 2.20 STUDENT FUNCTIONS

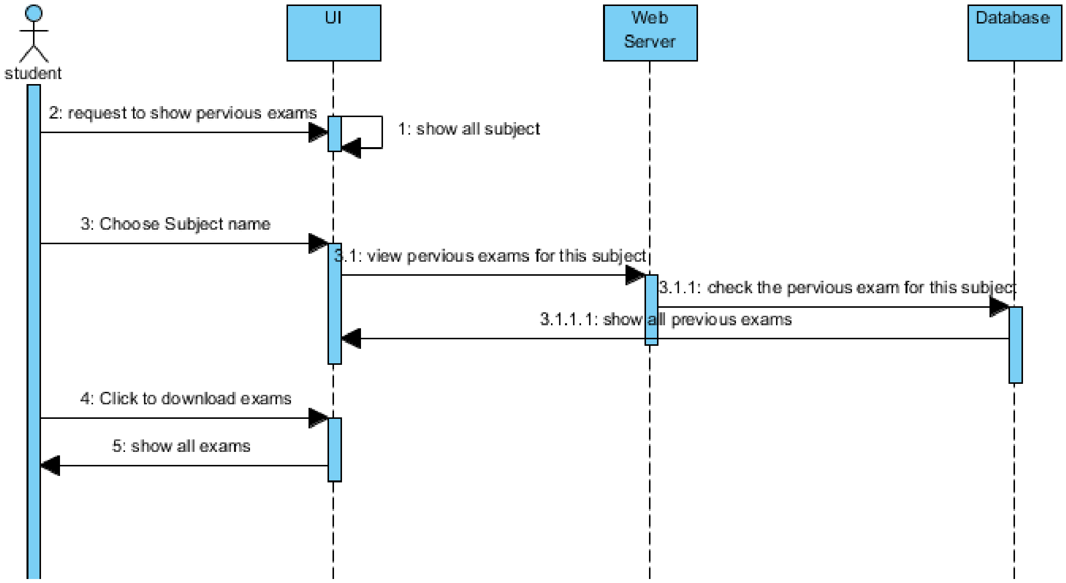

student
UI
Web Server
Database
2: request to show pervious exams
1: show all subject
3: Choose Subject name
3.1: view pervious exams for this subject
3.1.1: check the pervious exam for this subject
3.1.1.1: show all previous exams
4: Click to download exams
5: show all exams

FIGURE 2.19: STUDENT LOGIN

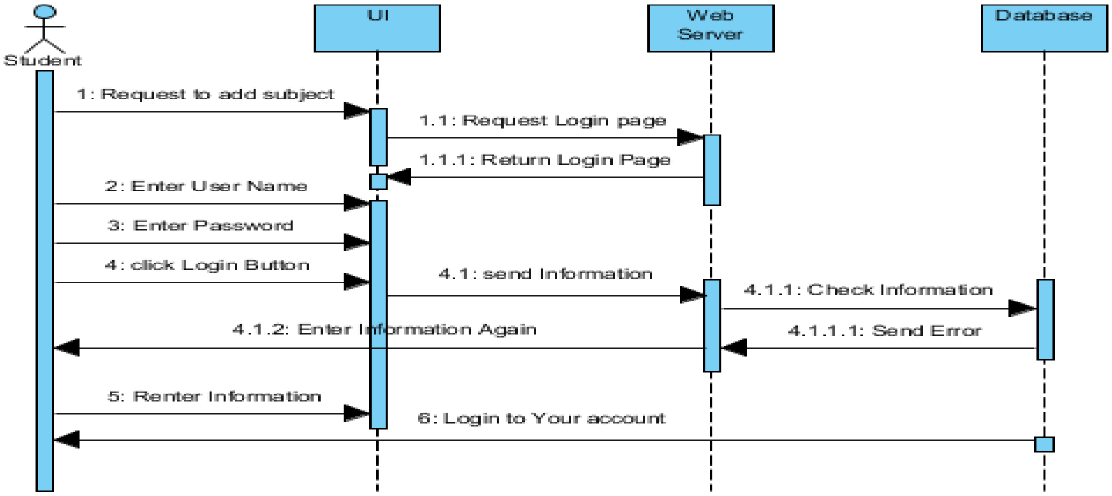
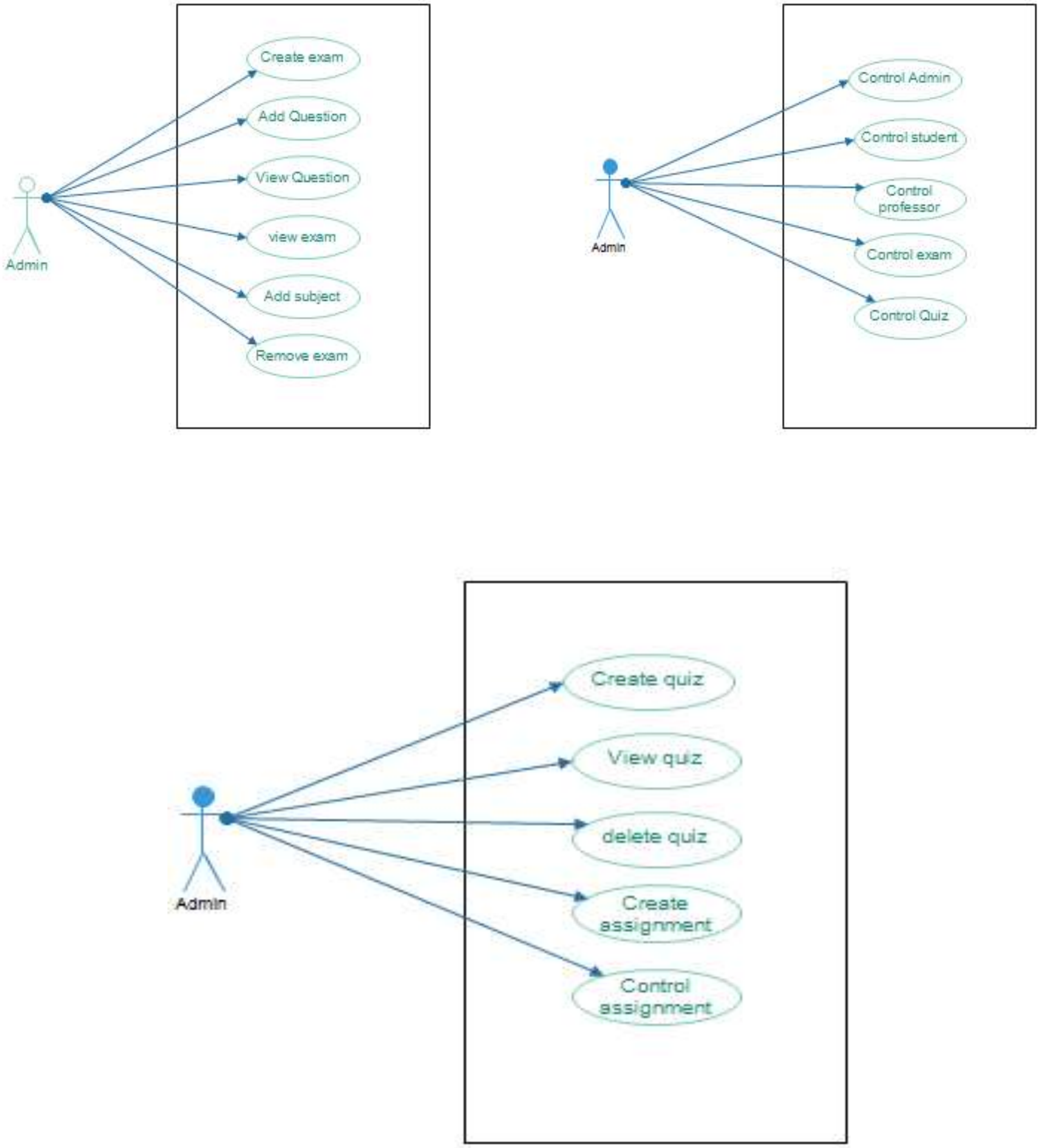


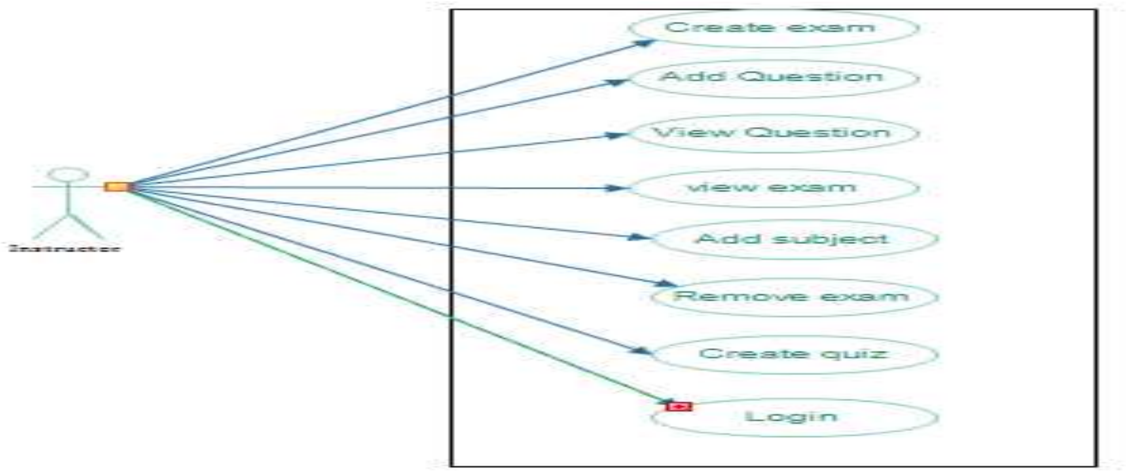
FIGURE 2.20: STUDENT PREVIOUS EXAM VIEW

2.5 Use Case Diagram

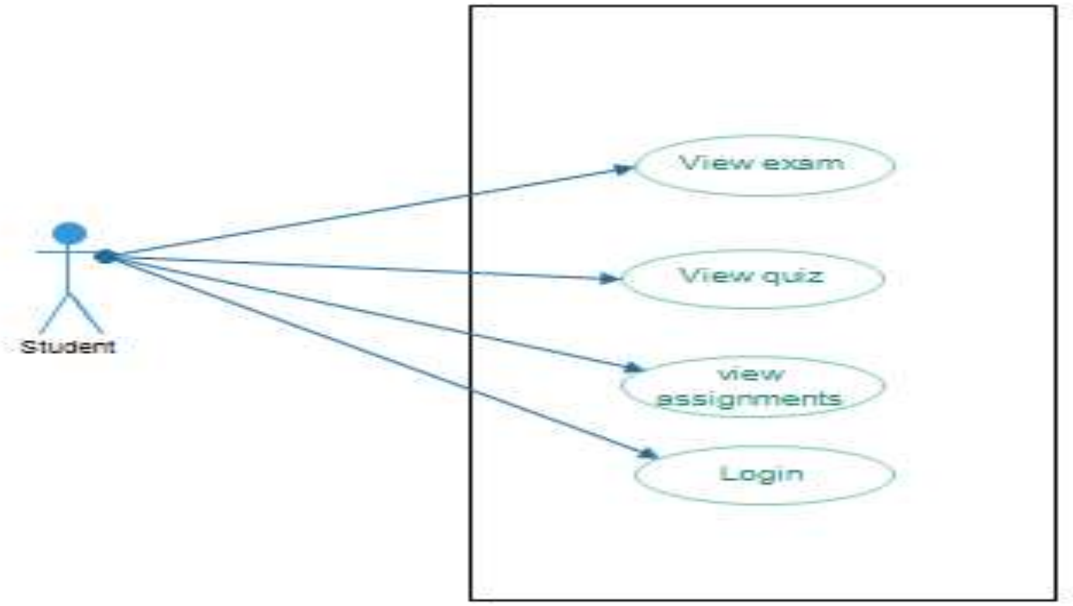
Figure 2.21 ADMIN FUNCTIONS



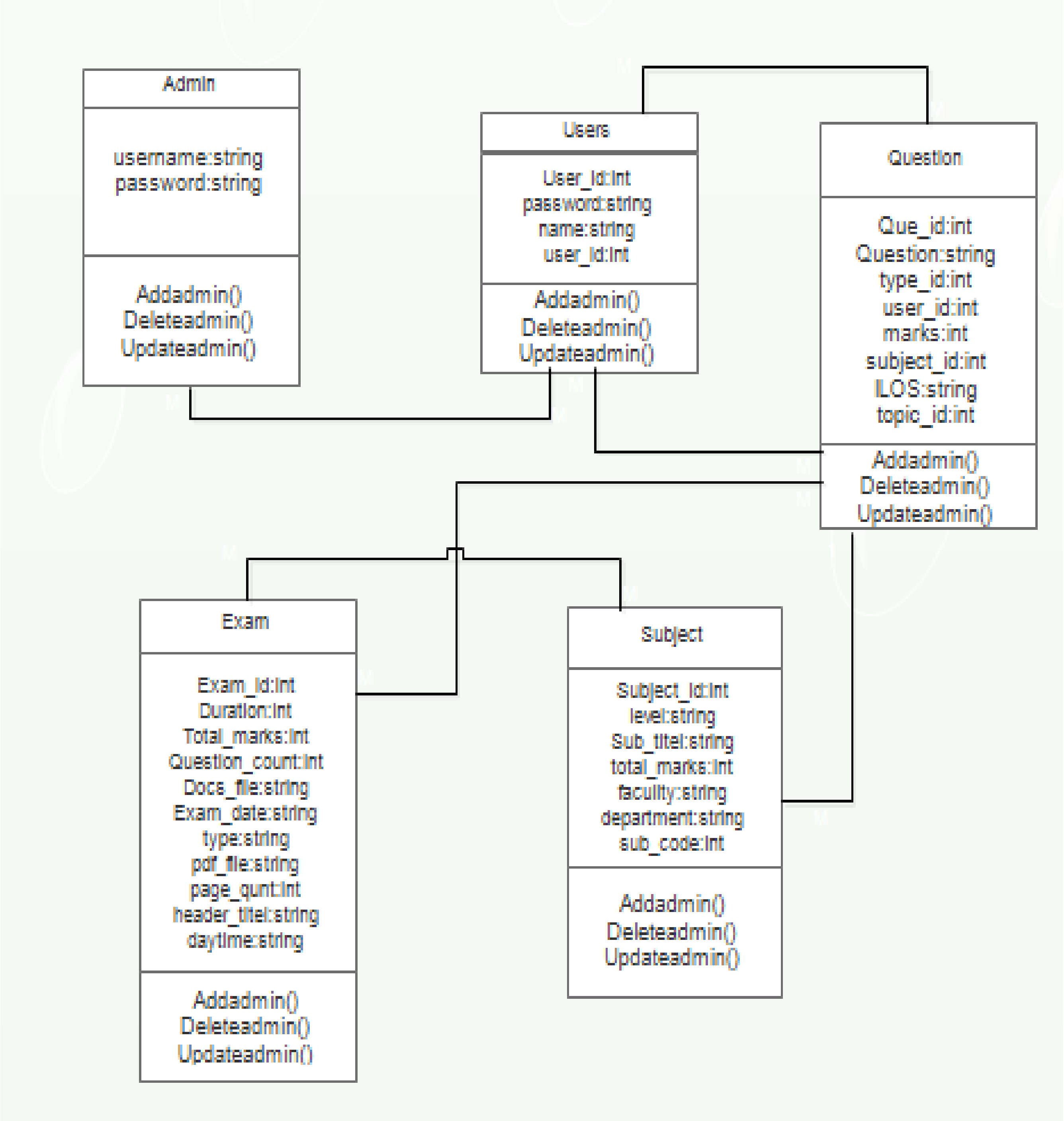
Use Case Diagram
Figure 2.22 INSTRUCTOR FUNCTIONS



Use Case Diagram
Figure 2.23 STUDENT FUNCTIONS



2.6
Figure 2.24 Class Diagram



Chapter 3

3.1 System Design

The next step after system analysis phase is system design, which include designing the diagrams of the system which help the developers in implementation of the system

So, in this chapter we will discuss each diagram in the systems such as:

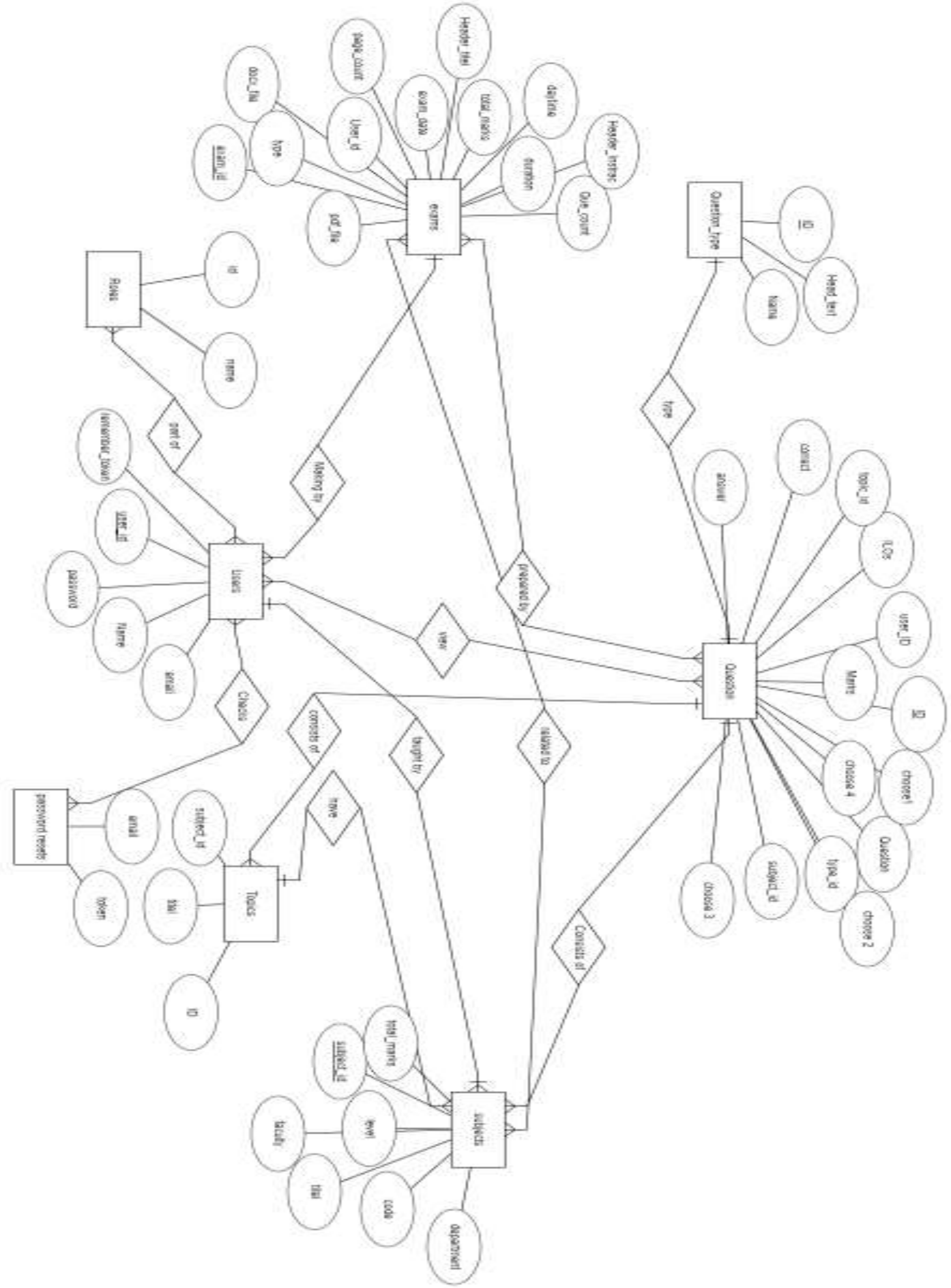
- **Entity relationship diagram:**

  The ER model defines the conceptual view of a database. It works around real-world entities and the associations among them. At view level, the ER model is considered a good option for designing databases.

- **Database Scheme**

  A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

3.2 Figure 3.1 ER Diagram



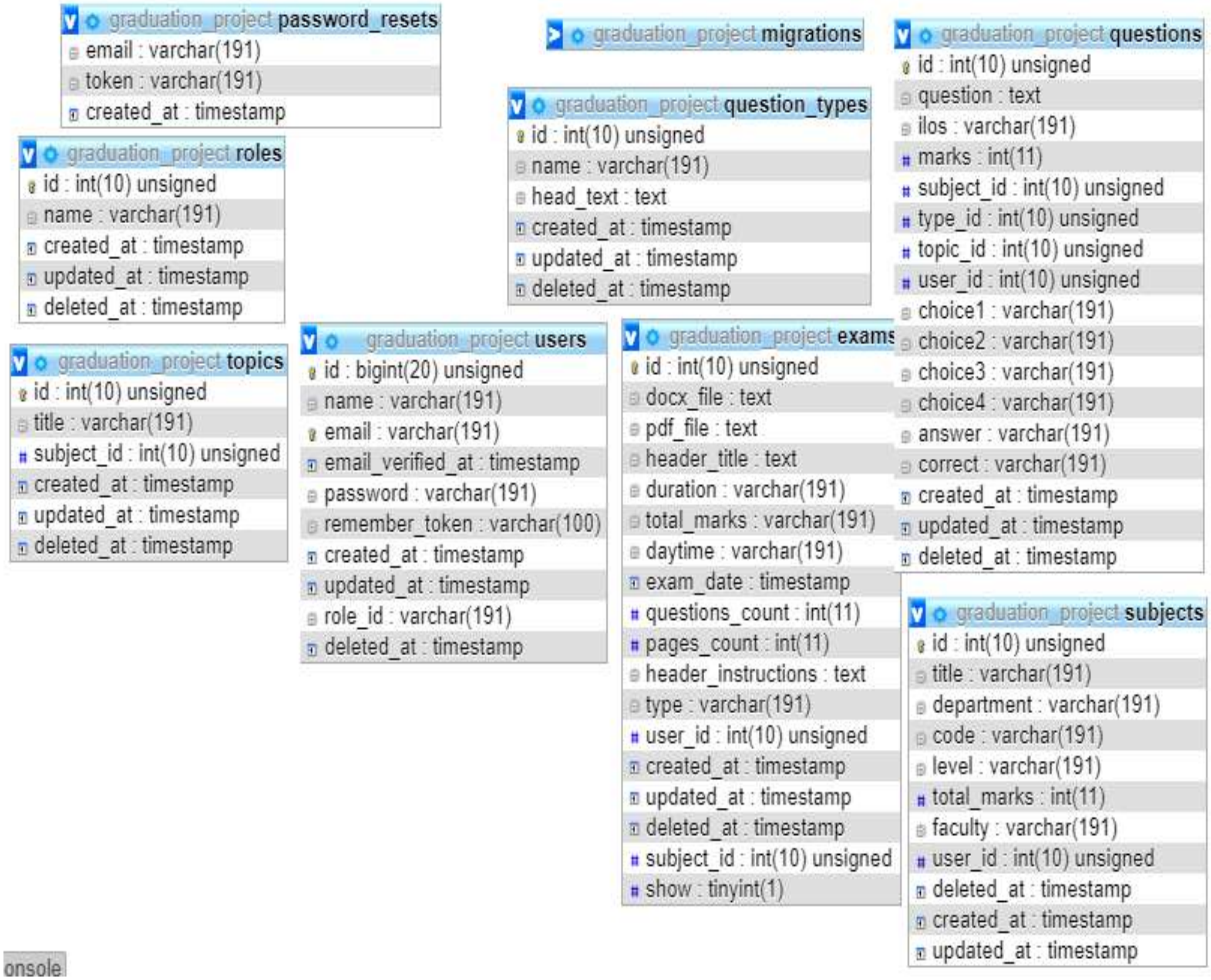
Chapter 4

4.1 Implementation

The implementation phase is divided into 6 sub-phases:

- Implementation of database tables and test the database
- Implementation of Admin Dashboard
- Implementation of Instructor Dashboard
- Implementation of Student Dashboard
- Implementation of Exam (Final, midterm, quiz, assignment)
- Implementation of Ilos (test, Report)

4.2 Figure 4.1 Database Implementation [Tables]



graduation_project password_resets
- email : varchar(191)
- token : varchar(191)
- created_at : timestamp

graduation_project roles
- id : int(10) unsigned
- name : varchar(191)
- created_at : timestamp
- updated_at : timestamp
- deleted_at : timestamp

graduation_project topics
- id : int(10) unsigned
- title : varchar(191)
- subject_id : int(10) unsigned
- created_at : timestamp
- updated_at : timestamp
- deleted_at : timestamp

graduation_project users
- id : bigint(20) unsigned
- name : varchar(191)
- email : varchar(191)
- email_verified_at : timestamp
- password : varchar(191)
- remember_token : varchar(100)
- created_at : timestamp
- updated_at : timestamp
- role_id : varchar(191)
- deleted_at : timestamp

graduation_project migrations

graduation_project question_types
- id : int(10) unsigned
- name : varchar(191)
- head_text : text
- created_at : timestamp
- updated_at : timestamp
- deleted_at : timestamp

graduation_project exams
- id : int(10) unsigned
- docx_file : text
- pdf_file : text
- header_title : text
- duration : varchar(191)
- total_marks : varchar(191)
- daytime : varchar(191)
- exam_date : timestamp
- questions_count : int(11)
- pages_count : int(11)
- header_instructions : text
- type : varchar(191)
- user_id : int(10) unsigned
- created_at : timestamp
- updated_at : timestamp
- deleted_at : timestamp
- subject_id : int(10) unsigned
- show : tinyint(1)

graduation_project questions
- id : int(10) unsigned
- question : text
- ilos : varchar(191)
- marks : int(11)
- subject_id : int(10) unsigned
- type_id : int(10) unsigned
- topic_id : int(10) unsigned
- user_id : int(10) unsigned
- choice1 : varchar(191)
- choice2 : varchar(191)
- choice3 : varchar(191)
- choice4 : varchar(191)
- answer : varchar(191)
- correct : varchar(191)
- created_at : timestamp
- updated_at : timestamp
- deleted_at : timestamp

graduation_project subjects
- id : int(10) unsigned
- title : varchar(191)
- department : varchar(191)
- code : varchar(191)
- level : varchar(191)
- total_marks : int(11)
- faculty : varchar(191)
- user_id : int(10) unsigned
- deleted_at : timestamp
- created_at : timestamp
- updated_at : timestamp

4.3 Code Implementation

```
<?php

namespace App\Http\Controllers;

use Response;

use App\Models\Exam;

use App\Models\User;

use App\Models\Topic;

use App\Models\Subject;

use App\Models\Question;

use Illuminate\Http\File;

use Laracasts\Flash\Flash;

use Illuminate\Http\Request;

use App\Models\Question_type;

use PhpOffice\PhpWord\PhpWord;

use PhpOffice\PhpWord\IOFactory;

use App\DataTables\ExamDataTable;

use App\Repositories\ExamRepository;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Storage;

use App\Http\Requests\CreateExamRequest;

use App\Http\Requests\UpdateExamRequest;

use PhpOffice\PhpWord\TemplateProcessor;

use App\Http\Controllers\AppBaseController;
```

```
class ExamController extends AppBaseController
{
    /** @var  ExamRepository */
    private $examRepository;
    public function __construct(ExamRepository $examRepo)
    {
        $this->examRepository = $examRepo;
    }
    /**
     * Display a listing of the Exam.
     *
     * @param ExamDataTable $examDataTable
     * @return Response
     */
    public function index(ExamDataTable $examDataTable)
    {
        return $examDataTable->render('exams.index');
    }
    /**
     * Show the form for creating a new Exam.
     *
     * @return Response
     */
    public function create()
    {
        return view('exams.create');
```

```
}
/**
 * Store a newly created Exam in storage.
 *
 * @param CreateExamRequest $request
 *
 * @return Response
 */
public function store(CreateExamRequest $request)
{
    $input = $request->all();

    $exam = $this->examRepository->create($input);

    Flash::success('Exam saved successfully.');

    return redirect(route('exams.index'));
}
/**
 * Display the specified Exam.
 *
 * @param  int $id
 *
 * @return Response
 */
public function show($id){
    $exam = $this->examRepository->find($id);
    if (empty($exam)) {
        Flash::error('Exam not found');
        return redirect(route('exams.index')); }
    return view('exams.show')->with('exam', $exam); }  /**
 * Show the form for editing the specified Exam.* * @param  int $id  *
 * @return Response  */
public function edit($id) {
    $exam = $this->examRepository->find($id);
    if (empty($exam)) {
        Flash::error('Exam not found');
        return redirect(route('exams.index'));     }
    return view('exams.edit')->with('exam', $exam);   }
/**  * Update the specified Exam in storage. *
 * @param  int              $id  * @param UpdateExamRequest $request * * @return Response */
public function update($id, UpdateExamRequest $request) {
    $exam = $this->examRepository->find($id);
    if (empty($exam)) {
        Flash::error('Exam not found');
        return redirect(route('exams.index'));  }
    $exam = $this->examRepository->update($request->all(), $id);
    Flash::success('Exam updated successfully.');
    return redirect(route('exams.index'));  }
/*** Remove the specified Exam from storage.   *  * @param  int $id *
 * @return Response */
public function destroy($id){
        $exam = $this->examRepository->find($id);
        if (empty($exam)) {
            Flash::error('Exam not found');
            return redirect(route('exams.index'));    }
        $this->examRepository->delete($id);
        Flash::success('Exam deleted successfully.');
        return redirect(route('exams.index')); }
    public function choseIlos($index = 0)   {
        $ilos = ['Intellectual skills', 'General and transferable skills', 'Professional and practical skills',
'Knowledge and understanding'];
        return $ilos[$index % 4];}
    public function exam_generate_store(Request $r)
    { $outerChosen = [];
        $topics = $r->topics;
        //dd($r);
        $phpWordHandle = new PhpWord();
        $section = $phpWordHandle->addSection();
        //Generate Exam
        for ($i = 1; $i <= count($r->question_num); $i++) {
            $numbering = 1;
            $question_head = Question_type::find($r->question_type[$i])->head_text;
            //Get Questions Algorithm  /**
             * get 10 questions from each ilos category
             * choose random one
             * push into array so it dont repeats */
            $chosen = []; // save choosed questions
    //get 10 random questions ids
    for ($inner = 0; $inner < $r->question_count[$i]; $inner++) {
      $chosen[] = optional(Question::where('type_id', $r->question_type[$i])
          ->where('ilos', $this->choseIlos($inner))
          ->where('subject_id', $r->subject_id)
          ->inRandomOrder()->whereIn('topic_id', $topics)
          ->whereNotIn('id', $chosen)
          ->limit(10)
          ->first())
        ->id;}
   $chosen = array_filter($chosen);
   $outerChosen = array_merge($outerChosen, $chosen);
    //get the questions
    $questions = Question::whereIn('id', $chosen)->get();
    //Start Generating
    $textrun = $section->addTextRun();
    $textrun->addText(htmlspecialchars("Question({$i}): " . $question_head . " \t ({$r->question_marks[$i]} Marks)"), array('bold' => true, 'size' => 14));
    $section->addTextBreak(1);
    $textrun = $section->addTextRun();
    if ($r->question_type[$i] != 5) {
      foreach ($questions as $value) {
        $textrun->addText(htmlspecialchars("{$numbering}. " . $value->question), array('size' => 11));
        $textrun->addTextBreak(1);
        $numbering++; }
        } else {
            foreach ($questions as $value) {
                //dd($value);
                $textrun->addText(htmlspecialchars("{$numbering}. " . $value->question), array('size' => 11));
                $textrun->addTextBreak(1);
                $numbering++;
                $textrun->addText(htmlspecialchars("A- " . $value->choice1), array('size' => 11));
                $textrun->addText(htmlspecialchars("\t"));
                $textrun->addText(htmlspecialchars("B- " . $value->choice2), array('size' => 11));
                $textrun->addText(htmlspecialchars("\t"));
                // $textrun->addTextBreak(1);
                $textrun->addText(htmlspecialchars("C- " . $value->choice3), array('size' => 11));
                $textrun->addText(htmlspecialchars("\t"));
                $textrun->addText(htmlspecialchars("D- " . $value->choice4), array('size' => 11));
                $textrun->addTextBreak(2); } }
        //$section->addTextBreak(1);}
    $objWriter = IOFactory::createWriter($phpWordHandle);
    $fullXml = $objWriter->getWriterPart('Document')->write();
    $TemplateProcessor = new TemplateProcessor(public_path('templates/exam') . '/' . $r->template);
    $TemplateProcessor->replaceBlock('exam', $this->getBodyBlock($fullXml));
    //Set Exam Values
    $TemplateProcessor->setValue('department', htmlspecialchars($r->department));
    $TemplateProcessor->setValue('dr', htmlspecialchars($r->instructor_name));
    $TemplateProcessor->setValue('level', htmlspecialchars($r->level));
    $TemplateProcessor->setValue('title', htmlspecialchars($r->title));
```

```
        $TemplateProcessor->setValue('code', htmlspecialchars($r->code));
        $TemplateProcessor->setValue('date', htmlspecialchars($r->exam_date));
        $TemplateProcessor->setValue('daytime', htmlspecialchars($r->exam_daytime));
        $TemplateProcessor->setValue('dur', htmlspecialchars($r->exam_duration));
        $TemplateProcessor->setValue('q_count', htmlspecialchars(count($r->question_num)));
        $TemplateProcessor->setValue('instructions', htmlspecialchars($r->header_instructions));
        $TemplateProcessor->setValue('exam_header_title', htmlspecialchars($r->exam_type . ' exam for ' . $r->exam_term . ' term, Academic year ' . $r->exam_year));
        $TemplateProcessor->setValue('marks', htmlspecialchars($r->total_marks));
        //save template with table
        $wordDocumentFile = $TemplateProcessor->save();
        $fileName = $r->exam_type . ' Exam - ' . $r->title . ' ' . $r->exam_date . '.docx';
        Storage::putFileAs('public', new File($wordDocumentFile), $fileName);
        $examfile = storage_path('app\\public\\' . $fileName);
        $exam = $this->examRepository->create($r->all());
        $exam->docx_file = $examfile;
        $exam->subject_id = $r->subject_id;
        $exam->type = $r->exam_type;
        $exam->user_id = Auth::id();
        $exam->file_name = $fileName;
        $exam->update();
        //dd($chosen);
        $exam->setMeta('questions', $outerChosen);
       return redirect((route('exam.report', ['exam_id' => $exam->id]))); }
    public function examReport(Request $r){
        $exam = Exam::find($r->exam_id);
        $questions_ids = $exam->getMeta('questions');

        //$ilos = ['Intellectual skills', 'General and transferable skills', 'Professional and practical skills', 'Knowledge and understanding'];

        $ilos = [];

        $topics = [];

        $ilos['is'] = Question::where('ilos', 'Intellectual skills')->whereIn('id', $questions_ids)->count();

        $ilos['gt'] = Question::where('ilos', 'General and transferable skills')->whereIn('id', $questions_ids)->count();

        $ilos['pp'] = Question::where('ilos', 'Professional and practical skills')->whereIn('id', $questions_ids)->count();

        $ilos['ku'] = Question::where('ilos', 'Knowledge and understanding')->whereIn('id', $questions_ids)->count();

        $question_sum = $ilos['is'] + $ilos['gt'] + $ilos['pp'] + $ilos['ku'] ;

        foreach (Topic::where('subject_id', $exam->subject_id)->get() as $key => $value) {

            $topics[$value->title] = Question::where('topic_id', $value->id)->whereIn('id', $questions_ids)->count();  }

        return view('main.report')->with(['exam' => $exam, 'ilos' => $ilos, 'topics' => $topics, 'qs' => $question_sum]);  }

    public function download_exam(Request $r)  {

        $exam = Exam::find($r->exam_id);

        $path = $exam->docx_file;

        $fileName = $exam->file_name;

        //download file

        header('Content-Description: File Transfer');

        header('Content-Type: application/msword');

        header('Content-Disposition: attachment; filename="' . $fileName . '"');

        header('Expires: 0');
        header('Cache-Control: must-revalidate');
        header('Pragma: public');
        header('Content-Length: ' . filesize($path));
        flush(); // Flush system output buffer
        readfile($path); }
    public function generate_store(Request $r){
        switch ($r->exam_type) {
            case 'final':
            case 'midterm':
                //Validation
                foreach ($r->question_count as $value) {
                    if (empty($value)) {
                        Flash::error('Number Of Question is Required');
                        return redirect()->back()->with($r->query_string);  }  }
                if (empty($r->topics)) {
                    Flash::error('You should choose at least one chapter');
                    return redirect()->back()->with($r->query_string);}
                if (empty($r->total_marks) || array_sum($r->question_marks) != $r->total_marks) {
                    Flash::error('Sum of marks must equal total marks');
                    return redirect()->back()->with($r->query_string); }
                return $this->exam_generate_store($r);
                break;
            default:
                # code...
                break;  } }
    protected function getBodyBlock($string)  {
        if (preg_match('%(?i)(?<=<w:body>)[\s|\S]*?(?=</w:body>)%', $string, $regs)) {
            return $regs[0];
        } else {
            return '';   }  }
    public function generate(Request $r)  {
        $s = Subject::where('id', $r->subject)->firstOrFail();
        $qc = $r->noq;
        return view('main.gen_exam')->with([
            's' => $s,
            'type' => $r->type,
            'qc' => $qc,
            'question_types' => Question_type::all(),
        ]);}
    public function other_generate(Request $r)  {
        $s = Subject::where('id', $r->subject)->firstOrFail();
        $qc = $r->noq;
        return view('main.gen_other')->with([
            's' => $s,
            'type' => $r->type,
            'qc' => $qc,
            'question_types' => Question_type::all(),
        ]);  }
    public function generator(Request $r)  {
        switch ($r->type) {
            case 'final':
            case 'midterm':
                $users = User::where('id', Auth::id())->with('subjects')->first();
                return view('main.exam')->with(['type' => $r->type, 'u' => $users]);
                break;
            case 'quiz':
            case 'assignment':
            $users = User::where('id', Auth::id())->with('subjects')->first();
                return view('main.other')->with(['type' => $r->type, 'u' => $users]);
                break;
<?php
namespace App\Http\Controllers;
use App\DataTables\QuestionDataTable;
use App\Http\Requests;
use App\Http\Requests\CreateQuestionRequest;
use App\Http\Requests\UpdateQuestionRequest;
use App\Repositories\QuestionRepository;
use Flash;
use App\Http\Controllers\AppBaseController;
use Response;
class QuestionController extends AppBaseController{
    /** @var  QuestionRepository */
    private $questionRepository;
    public function __construct(QuestionRepository $questionRepo) {
        $this->questionRepository = $questionRepo;  }
    /** * Display a listing of the Question.* * @param QuestionDataTabl$questionDataTable* @return
Response */
    public function index(QuestionDataTable $questionDataTable)  {
        return $questionDataTable->render('questions.index');  }
    /*** Show the form for creating a new Question.** @return Response */
    public function create()
    {
        return view('questions.create'); }
 /**  * Store a newly created Question in storage.* * @param CreateQuestionRequest $request* *
@return Response*/
    public function store(CreateQuestionRequest $request) {
        $input = $request->all();
        $question = $this->questionRepository->create($input);
        Flash::success('Question saved successfully.');
        return redirect(route('questions.index'));  }
    /** * Display the specified Question. * * @param  int $id
     * * @return Response */
    public function show($id)   {
        $question = $this->questionRepository->find($id);
        if (empty($question)) {
            Flash::error('Question not found');
            return redirect(route('questions.index')); }
        return view('questions.show')->with('question', $question); }
    /*** Show the form for editing the specified Question.
     ** @param  int $id * * @return Response  */
    public function edit($id)  {
        $question = $this->questionRepository->find($id);
        if (empty($question)) {
            Flash::error('Question not found');
            return redirect(route('questions.index')); }
        return view('questions.edit')->with('question', $question);  }
    /** * Update the specified Question in storage. *
     * @param  int     $id* @param UpdateQuestionRequest $request*
     * @return Response  */
    public function update($id, UpdateQuestionRequest $request) {
        $question = $this->questionRepository->find($id);
        if (empty($question)) {
            Flash::error('Question not found');
            return redirect(route('questions.index'));   }
        $question = $this->questionRepository->update($request->all(), $id);
        Flash::success('Question updated successfully.');
        return redirect(route('questions.index'));  }
  /** * Remove the specified Question from storage.* * @param  int $id
     * * @return Response */
    public function destroy($id)
    {
        $question = $this->questionRepository->find($id);
        if (empty($question)) {
            Flash::error('Question not found');
            return redirect(route('questions.index'));  }
        $this->questionRepository->delete($id);
        Flash::success('Question deleted successfully.');
        return redirect(route('questions.index'));    }}
<?php
/*
```

```
|--------------------------------------------------------------------------
| Web Routes|--------------------------------------------------------------------------|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|*/
Route::get('generator', 'ExamController@generator')->name('generator');
Route::post('generate_store', 'ExamController@generate_store')->name('generate.store');
Route::get('generate', 'ExamController@generate')->name('generate');
Route::post('other_generate_store', 'ExamController@other_generate_store')-
>name('other.generate.store');
Route::get('other_generate', 'ExamController@other_generate')->name('other.generate');
Route::get('ExamReport', 'ExamController@examReport')->name('exam.report');
Route::get('downloadExam', 'ExamController@download_exam')->name('exam.download');
Auth::routes();
Route::get('/', 'HomeController@index')->name('welcome');
Route::get('/home', 'HomeController@index');
Route::get('generator_builder',
'\InfyOm\GeneratorBuilder\Controllers\GeneratorBuilderController@builder');
Route::get('field_template',
'\InfyOm\GeneratorBuilder\Controllers\GeneratorBuilderController@fieldTemplate');Route::get('relation
_field_template',
'\InfyOm\GeneratorBuilder\Controllers\GeneratorBuilderController@relationFieldTemplate');
Route::post('generator_builder/generate',
'\InfyOm\GeneratorBuilder\Controllers\GeneratorBuilderController@generate');
Route::post('generator_builder/rollback',
'\InfyOm\GeneratorBuilder\Controllers\GeneratorBuilderController@rollback');
Route::post(
    'generator_builder/generate-from-file',
'\InfyOm\GeneratorBuilder\Controllers\GeneratorBuilderController@generateFromFile');

Route::resource('questionTypes', 'Question_typeController');

Route::resource('subjects', 'SubjectsController');

Route::resource('topics', 'TopicController');

Route::resource('exams', 'ExamController');

Route::resource('questions', 'QuestionController');

Route::resource('users', 'UserController');

Route::resource('roles', 'RoleController');

{

    "name": "laravel/laravel",

    "type": "project",

    "description": "The Laravel Framework.",

    "keywords": [

        "framework",

        "laravel"

    ],

    "license": "MIT",

    "require": {

        "php": "^7.1.3",

        "appointer/swaggervel": "dev-master",

        "barryvdh/laravel-ide-helper": "^2.6",

        "doctrine/dbal": "~2.3",

        "fideloper/proxy": "^4.0",

        "infyomlabs/adminlte-templates": "5.8.x-dev",

        "infyomlabs/generator-builder": "dev-master",
        "infyomlabs/laravel-generator": "5.8.x-dev",
        "infyomlabs/swagger-generator": "dev-master",
        "laravel/framework": "5.8.*",
        "laravel/tinker": "^1.0",
        "laravelcollective/html": "^5.8.0",
        "phpoffice/phpword": "^0.16.0",
        "plank/laravel-metable": "^1.1",
        "spatie/laravel-medialibrary": "^7.0.0",
        "yajra/laravel-datatables-buttons": "^4.6",
        "yajra/laravel-datatables-fractal": "^1.5"
    },
    "require-dev": {
        "beyondcode/laravel-dump-server": "^1.0",
        "filp/whoops": "^2.0",
        "fzaninotto/faker": "^1.4",
        "mockery/mockery": "^1.0",
        "nunomaduro/collision": "^2.0",
        "phpunit/phpunit": "^7.5"
    },
    "config": {
        "optimize-autoloader": true,
        "preferred-install": "dist",
        "sort-packages": true
    },
    "extra": {
        "laravel": {
            "dont-discover": []
        }
    },
    "autoload": {
        "psr-4": {
            "App\\": "app/"
        },
        "classmap": [
            "database/seeds",
            "database/factories"
        ]
    },
    "autoload-dev": {
        "psr-4": {
            "Tests\\": "tests/"
        }
    },
    "minimum-stability": "dev",
    "prefer-stable": true,
    "scripts": {
        "post-autoload-dump": [
            "Illuminate\\Foundation\\ComposerScripts::postAutoloadDump",
            "@php artisan package:discover --ansi"
        ],
        "post-root-package-install": [
            "@php -r \"file_exists('.env') || copy('.env.example', '.env');\""
        ],
        "post-create-project-cmd": [
            "@php artisan key:generate --ansi"
        ] }}
<?php
namespace App\Models;
use App\Models\User;
use Eloquent as Model;
use Plank\Metable\Metable;
use Illuminate\Database\Eloquent\SoftDeletes;
/**
 * Class Exam
 * @package App\Models
 * @version June 22, 2019, 11:01 am UTC
 *
 * @property \App\Models\User user
 * @property string docx_file
 * @property string pdf_file
 * @property string header_title
 * @property string duration
 * @property string total_marks
 * @property string daytime
 * @property string exam_date
 * @property integer questions_count
 * @property integer pages_count
 * @property string header_instructions
 * @property string type
 * @property integer user_id
 * @property string created_at
 */
class Exam extends Model
{
    use SoftDeletes;
    use Metable;
    public $table = 'exams';
    protected $dates = ['deleted_at'];
    public $fillable = [
        'docx_file',
        'pdf_file',
        'header_title',
        'duration',
        'total_marks',
        'daytime',
        'exam_date',
        'questions_count',
        'pages_count',
        'header_instructions',
        'type',
        'user_id',
        'subject_id',
        'created_at',
        'show'
```

```
];
/**
 * The attributes that should be casted to native types.  * * @var array*/
protected $casts = [
    'id' => 'integer',
    'docx_file' => 'string',
    'pdf_file' => 'string',
    'header_title' => 'string',
    'duration' => 'string',
    'total_marks' => 'string',
    'daytime' => 'string',
    'questions_count' => 'integer',
    'pages_count' => 'integer',
    'header_instructions' => 'string',
    'type' => 'string',
    'subject_id' => 'integer',
    'user_id' => 'integer'
];
/*** Validation rules* * @var array*/
public static $rules = [
];
public function user()  {
    return $this->BelongsTo(\App\Models\User::class);
}
public function subject()
{
        return $this->BelongsTo(\App\Models\Subject::class);
    }
}
<?php
namespace App\Models;
use App\Models\Topic;
use Eloquent as Model;
use Illuminate\Database\Eloquent\SoftDeletes;
/**
 * Class Subjects
 * @package App\Models
 * @version June 22, 2019, 10:41 am UTC
 *
 * @property \App\Models\User user
 * @property string title
 * @property string department
 * @property string code
 * @property string level
 * @property integer total_marks
 * @property string faculty
 * @property integer user_id
 */
class Subject extends Model{
    use SoftDeletes;
    public $table = 'subjects';
    protected $dates = ['deleted_at'];
    public $fillable = ['title',  'department',  'code', 'level', 'total_marks', 'faculty', 'user_id'];

    /** * The attributes that should be casted to native types.*

     * @var array*/

    protected $casts = [

        'id' => 'integer',

        'title' => 'string',

        'department' => 'string',

        'code' => 'string',

        'level' => 'string',

        'total_marks' => 'integer',

        'faculty' => 'string',

        'user_id' => 'integer'];

 /** * Validation rules ** @var array*/

    public static $rules = [

        'title' => 'required',

        'department' => 'required',

        'code' => 'required',

        'level' => 'required',

        'total_marks' => 'required',

        'faculty' => 'required'  ];

    /*** @return \Illuminate\Database\Eloquent\Relations\HasOne **/

    public function user()

    {return $this->hasOne(\App\Models\User::class, 'id', 'user_id');}

    public function topics() {

        return $this->hasMany(\App\Models\Topic::class); }}

<?php
```

```
namespace App\Repositories;

use App\Models\Exam;

use App\Repositories\BaseRepository;

/*** Class ExamRepository * @package App\Repositories

 * @version June 22, 2019, 11:01 am UTC*/

class ExamRepository extends BaseRepository{

    /*** @var array*/

    protected $fieldSearchable = [ 'docx_file', 'pdf_file', 'header_title',  'duration', 'total_marks',
'daytime','exam_date',  'questions_count','pages_count',

        'header_instructions', 'type', 'user_id',  'created_at'    ];

 /** * Return searchable fields * * @return array */

    public function getFieldsSearchable(   {

        return $this->fieldSearchable;  }

    /*** Configure the Model**/

    public function model(){

        return Exam::class;    }}

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;

use Illuminate\Foundation\Auth\AuthenticatesUsers;

class LoginController extends Controller

{/* |---------------------------------------------------------------------------  | Login Controller

    |--------------------------------------------------------------------------|

    | This controller handles authenticating users for the application and

    | redirecting them to your home screen. The controller uses a trait

    | to conveniently provide its functionality to your applications. |
    */  use AuthenticatesUsers; /** * Where to redirect users after login.*
     * @var string*/
    protected $redirectTo = '/home';/**
     * Create a new controller instance. * * @return void */
    public function __construct()  {
        $this->middleware('guest')->except('logout'); }}
<?php
namespace App\Http\Controllers\Auth;
use App\User;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use Illuminate\Foundation\Auth\RegistersUsers;
class RegisterController extends Controller
{  /* |-----------------------------------------------------------------------
    | Register Controller  |-----------------------------------------------------------------------|
    | This controller handles the registration of new users as well as their
    | validation and creation. By default this controller uses a trait to
    | provide this functionality without requiring any additional code.
    |*/
    use RegistersUsers;

    /**  * Where to redirect users after registration. * * @var string*/
    protected $redirectTo = '/home';
  /** * Create a new controller instance.* * @return void*/
    public function __construct()
    {
        $this->middleware('guest');
    }
    /**
     * Get a validator for an incoming registration request.
     * * @param  array  $data * @return \Illuminate\Contracts\Validation\Validator*/
    protected function validator(array $data)
    {
        return Validator::make($data, [
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
            'password' => ['required', 'string', 'min:8', 'confirmed'],
        ]);
    } /**
     * Create a new user instance after a valid registration. *
     * @param  array  $data
     * @return \App\User */
    protected function create(array $data){
        return User::create([
            'name' => $data['name'],
            'email' => $data['email'],
            'password' => Hash::make($data['password']),
        ]); }}
@extends('layouts.app')
@section('content')
<form action="{{ route('other.generate.store') }}" method="post">
```

```
    {{ csrf_field() }}
    <input type="hidden" name="template" value="{{ $_GET['template'] }}">
{!! Form::hidden('query_string', $_SERVER["QUERY_STRING"]) !!}
{!! Form::hidden('subject_id', $_GET['subject']) !!}
     <div style="width:90%; margin:auto; overflow: hidden; background: #fff;">
        <div style="margin:20px; border: 5px solid #000; height:93%;">
            <div style="width:90%; margin: auto; overflow: hidden;">
                <h3 style="color:red;">This is only A preview form please choose the desaier Template To Generate.</h3>
                <br>
                <div class="clearfix"></div>
                <img src="{{ asset('assets/image/download.jpg') }}" alt="Image" style="width:100px; float: left;">


                <img src="{{ asset('assets/image/ss.png') }}" alt="Image" style="width:120px; float: right">
            </div>
            <div style="width:98%; margin: 10px auto; overflow: hidden; text-align: center; border: 5px solid #000;">
             <div>
                <h2>Octoper 6 University</h2>
                <p style="font-size:20px">{{$s->faculty}}</p>
             </div>
             <div style=" width:100%; margin: auto; overflow: hidden; text-align:left; border-top: 5px solid #000; display: flex; flex-direction: row; justify-content: space-around;">
                <div style="font-size:18px; font-weight: bold; padding:5px; width:45%; border-right:5px solid #000; ">
                    <p>
                    Department Of
                    <select name="department">
                        <option <?php if($s->department == 'Computer Science') echo 'selected' ?> value="Computer Science">Computer Science</option>
                        <option <?php if($s->department == 'Information Systems') echo 'selected' ?> value="Information Systems">Information Systems</option>
                        <option <?php if($s->department == 'CS & IS') echo 'selected' ?> value="CS & IS">CS & IS</option>
                    </select>
                </p>
                <p>
                    Study Year/Level :
                    <select name="level">
                        <option <?php if($s->level == 1) echo 'selected' ?> value="1st">1st</option>
                        <option <?php if($s->level == 2) echo 'selected' ?> value="2nd">2nd</option>
                        <option <?php if($s->level == 3) echo 'selected' ?> value="3rd">3rd</option>
                        <option <?php if($s->level == 4) echo 'selected' ?> value="4th">4th</option>
                    </select>
                    Level
                </p>
                <p>
                    Course Title : <input name="title" value="{{ $s->title }}" type="text">
                </p>
                <p>
                    Course code: <input name="code" value="{{ $s->code }}" type="text">
                </p>
            </div>
            <div style="font-size:18px; font-weight: bold; padding:5px; width:45%; border-left:5px solid
#000; ">
                <p>
                    {{ ucfirst($_GET['type']) }} Number: # <input style="width:20%;" name="ass_num"
type="text" value="{{ $_GET['ass_num'] }}">
                </p>
                <p>
                    Model: {{ $_GET['nom'] }} Model(s)
                    <input type="hidden" name="nom" value="{{ $_GET['nom'] }}">
                </p>
                <p>
                    Due (Dead Line): <input type='text' style="width:50%;" name="due"
id='datetimepicker1' />
                </p>
                <p>
                    Total Marks: <input name="total_marks" type="text"> Marks
                </p>
            </div>
           </div>
          </div>
          <div style=" padding-left: 10px">
                <div>
                    <p style="font-size:20px; font-weight: bold; padding:5px;  text-decoration:
underline;">Choose Chapters to generate from: </p>
                </div>
                <div>
                    <p style="font-size:20px; font-weight: bold; padding:5px;  text-decoration: underline;">
```

```
                            @foreach ($s->topics()->get() as $topic)

                                {{ $topic->title }}: <input type="checkbox" name="topics[]" value="{{ $topic->id
}}">

                            @endforeach


                        </p>

                    </div>



                    <div>

                        <p style="font-size:20px; font-weight: bold; padding:5px;  text-decoration:
underline;">Answer the follwing questions :</p>

                    </div>



@for ($i = 1; $i <= $qc; $i++)



                <div style=" display: flex; flex-direction: column; padding:5px; ">

                    <div >

                        <h4 style="font-size:20px; font-weight: bold; text-decoration: underline; float:
left;">Question {{ $i }}:</h4>

                        <input type="hidden" name="question_num[{{ $i }}]" value="{{ $i }}">

                        <p style="float:right; font-size:20px; font-weight: bold;">

                            (

                                <input type="text" name="question_marks[{{ $i }}]" style="width:40px; ">…. Marks

                            )

                        </p>

                    </div>

                    <div style="font-size:20px;">
```

```
                        <div>
                            <label for="exampleInputFile" style="font-size:20px; margin-left:30px">Question Types</label>
                            <select name="question_type[{{ $i }}]" style="width:300px; height:32px;">
                                @foreach ($question_types as $qt)
                                    <option value="{{ $qt->id }}">{{ $qt->name }}</option>
                                @endforeach
                            </select>
                            <div style="margin:20px">
                                <label for="numberquestion">Number of Questions</label>
                                <input type="text" name="question_count[{{ $i }}]" id="numberquestion">
                            </div>
                        </div>
                    </div>
                </div>
@endfor
                <div style="width:30%; margin: auto; overflow: hidden; padding: 20px 0">
                    <p style="font-size:25px; font-weight: bold; font-style: italic;">With My Best Wiches</p>
                    <p style="font-size:15px; font-weight: bold; font-style: italic;">DR.<input name="instructor_name" type="text"></p>
                </div>
            </div>
        </div>
        <div style="width:15%; margin: auto; overflow: hidden; ">
            <button type="submit" class="btn btn-primary" >Generate Assignment</button>
        </div>
```

```
    </form>
@endsection
@push('scripts')
           <script type="text/javascript">
            $(function () {
                $('#datetimepicker1').datetimepicker(
                {   format: 'DD/mm/YYYY h:mm a'
                } );});
        </script>
@endpush
@extends('layouts.app')
@section('content')
   @include('flash::message')
<form action="{{ route('generate.store') }}" method="post">
    {{ csrf_field() }}
    <input type="hidden" name="template" value="{{ $_GET['template'] }}">
{!! Form::hidden('query_string', $_SERVER["QUERY_STRING"]) !!}
{!! Form::hidden('subject_id', $_GET['subject']) !!}
     <div style="width:90%; margin:auto; overflow: hidden; background: #fff;">
        <div style="margin:20px; border: 5px solid #000; height:93%;">
            <div style="width:90%; margin: auto; overflow: hidden;">
                <h3 style="color:red;">This is only A preview form please choose the desaier Template To
Generate.</h3>
                <br>
                <div class="clearfix"></div>
           <img src="{{ asset('assets/image/download.jpg') }}" alt="Image" style="width:100px; float: left;">
           <img src="{{ asset('assets/image/ss.png') }}" alt="Image" style="width:120px; float: right">
        </div>
        <div style="width:98%; margin: 10px auto; overflow: hidden; text-align: center; border: 5px solid #000;">
         <div>
           <h2>Octoper 6 University</h2>
           <p style="font-size:20px">{{$s->faculty}}</p>
           <p style="font-size:20px">
                {!! Form::select('exam_type', ['midterm' => 'Midterm', 'final' => 'Final'], null ) !!}
             Exam for
             <select name="exam_term">
                <option value="1st">1st</option>
                <option value="2nd">2nd</option>
                <option value="summer course">summer course</option>
             </select> Term,
             Academic year
             <select name="exam_year">
                <option value="2018/2019">2018/2019</option>
                <option value="2019/2020">2019/2020</option>
                <option value="2020/202">2020/2021</option>
                <option value="2021/2022">2021/2022</option>
                <option value="2022/2023">2022/2023</option>
             </select>
           </p>
```

```
    </div>
    <div style=" width:100%; margin: auto; overflow: hidden; text-align:left; border-top: 5px solid #000; display: flex; flex-direction: row; justify-content: space-around;">
     <div style="font-size:18px; font-weight: bold; padding:5px; width:45%; border-right:5px solid #000; ">
        <p>
          Department Of
          <select name="department">
            <option <?php if($s->department == 'Computer Science') echo 'selected' ?> value="Computer Science">Computer Science</option>
            <option <?php if($s->department == 'Information Systems') echo 'selected' ?> value="Information Systems">Information Systems</option>
            <option <?php if($s->department == 'CS & IS') echo 'selected' ?> value="CS & IS">CS & IS</option>
          </select>
        </p>
        <p>
          Study Year/Level :
          <select name="level">
            <option <?php if($s->level == 1) echo 'selected' ?> value="1st">1st</option>
            <option <?php if($s->level == 2) echo 'selected' ?> value="2nd">2nd</option>
            <option <?php if($s->level == 3) echo 'selected' ?> value="3rd">3rd</option>
            <option <?php if($s->level == 4) echo 'selected' ?> value="4th">4th</option>
          </select>
          Level
        </p>
        <p>
```

```
                Course Title : <input name="title" value="{{ $s->title }}" type="text">
            </p>
            <p>
                Course code: <input name="code" value="{{ $s->code }}" type="text">
            </p>
        </div>
        <div style="font-size:18px; font-weight: bold; padding:5px; width:45%; border-left:5px solid #000; ">
            <p>
                Examination Date : <input name="exam_date" type="date">
            </p>
            <p>
                Examination Starts:
                <select name="exam_daytime">
                    <option value="Moming">Morning</option>
                    <option value="Evening">Evening</option>
                </select>
            </p>
            <p>
                Allowed Examination Time:
                <select name="exam_duration">
                    <option value="1">1</option>
                    <option value="2">2</option>
                    <option value="3">3</option>
                    <option value="4">4</option>
                </select> hour</p>
```

```
                    <p>
                        Total Marks: <input id="total_marks" name="total_marks" type="text"> Marks  </p>
                <div>
             </div>
            </div>
            <div style="width:98%; margin: 10px auto; overflow: hidden; text-align: center; border: 5px solid
#000;">
                <p style="font-size:18px; font-weight: bold; padding:5px;">
                    The Examination consist of(x)Questions in (x)page </p>
            </div>
            <div style=" padding-left: 10px">  <div>
                        <p style="font-size:20px;">Instructions of Examination (<input style="width: 50%"
type="text" name="header_instructions" value="Read following Questions Carefully answer all of them,
assume any messing data">)</p>
                    </div>
                    <div>
                        <p style="font-size:20px; font-weight: bold; padding:5px;  text-decoration:
underline;">Choose Chapters to generate from: </p>
                    </div>
                    <div>
                         <div class="row">
                            @foreach ($s->topics()->get() as $topic)
                                <div style="font-size:20px; font-weight: bold; padding:5px;  text-decoration:
underline;" class="col-md-3">{{ $topic->title }} - ({{ $topic->questions->count() }} Qs): <input
type="checkbox" name="topics[]" value="{{ $topic->id }}"></div>
                            @endforeach
                        </div>
```

```
          </div>
          <div>
            <p style="font-size:20px; font-weight: bold; padding:5px;  text-decoration:
underline;">Answer the follwing questions :</p>
          </div>
@for ($i = 1; $i <= $qc; $i++)
        <div style=" display: flex; flex-direction: column; padding:5px; ">
          <div >
            <h4 style="font-size:20px; font-weight: bold; text-decoration: underline; float:
left;">Question {{ $i }}:</h4>
            <input type="hidden" name="question_num[{{ $i }}]" value="{{ $i }}">
            <p style="float:right; font-size:20px; font-weight: bold;">
              (
                <input id="question_marks" type="text" name="question_marks[{{ $i }}]"
style="width:40px; ">.... Marks
              )
            </p>
          </div>
          <div style="font-size:20px;">
            <div>
              <label for="exampleInputFile" style="font-size:20px; margin-left:30px">Question
Types</label>
              <select name="question_type[{{ $i }}]" style="width:300px; height:32px;">
                @foreach ($question_types as $qt)
                  <option value="{{ $qt->id }}">{{ $qt->name }}</option>
                @endforeach
              </select>
              <div style="margin:20px">
                <label for="numberquestion">Number of Questions</label>
                <input type="text" name="question_count[{{ $i }}]" id="numberquestion">
              </div>
            </div>
          </div>
        </div>
@endfor
        <div style="width:30%; margin: auto; overflow: hidden; padding: 20px 0">
          <p style="font-size:25px; font-weight: bold; font-style: italic;">With My Best Wiches</p>
          <p style="font-size:15px; font-weight: bold; font-style: italic;">DR.<input
name="instructor_name" type="text" value="{{ auth()->user()->name }}"></p>
        </div>
      </div>
    </div>
    <div style="width:15%; margin: auto; overflow: hidden; ">
      <button type="submit" id="submit" class="btn btn-primary" >Generate Exam</button>
    </div>
  </form>
@endsection
@extends('layouts.app')
@section('content')
  <section class="content container-fluid" >
   <div>
    <div class="box box-primary">
     <div class="containar">
```

```
        <button class="b btn btn-primary">Generate New {{ ucfirst($type) }}</button>
        <div class="desc">
        <div class="box box-primary">
          <form method="get" action="{{ route('generate') }}">
           {!! Form::hidden('type', $type) !!}
              {{ csrf_field() }}
              <div style=" padding-top:10px; width: 100%;">
                  <div class="form-group" style="width: 40%;  margin: auto; overflow: hidden;">
                      <label  style="font-size:20px;">Choose Subject</label>
                    <select name="subject" class="form-control">
                      <?php $subjects = (auth()->user()->role_id == 1 ? App\Models\Subject::get() : $u-
>subjects)  ?>
                      @foreach ($subjects as $subject)
                        <option value="{{ $subject->id }}">{{ $subject->title }}</option>
                      @endforeach
                    </select>
                  </div>
                  <div class="form-group" style="width: 40%;  margin: auto; overflow: hidden;">
                      <label  style="font-size:20px;">Number of Questions</label>
                    <select name="noq" class="form-control">
                      @for ($i = 1; $i < 9; $i++)
                          <option value="{{ $i }}">{{ $i }}</option>
                      @endfor
                    </select>
                  </div>
                  <div class="form-group" style="width: 40%;  margin: auto; overflow: hidden;">
```

```
                         <label  style="font-size:20px;">Choose Template</label>
                        <select name="template" class="form-control">
                         @foreach (\Illuminate\Support\Facades\File::files(public_path() . '\templates\exam') as
$file)
                           <option value="{{ $file->getFileName() }}">{{ $file->getFileName() }}</option>
                         @endforeach
                        </select>
                     </div>
                 </div>
                   <div style="padding: 30px 20px;margin:0 85%">
                 <button type="submit" class="btn btn-primary">Continue</button>
              </div>
              </form>
          </div>
          </div>
          </div>
        </div>
      </div>
    </div>
  </section>
@endsection@extends('layouts.app')
@section('content')
  <section class="content container-fluid" >
    <div>
      <div class="box-header">
        <h2>Welcome, What you want to do?</h2>
      </div>
      <div class="box">
        <div class="containar">
          <a href="{{ route('generator', ['type'=>'final']) }}" class="b btn btn-primary">Generate Final Exam</a>
        </div>
      </div>
      <div class="box">
        <div class="containar">
          <a href="{{ route('generator', ['type'=>'midterm']) }}" class="b btn btn-primary">Generate Mid-Term Exam</a>
        </div>
      </div>
      <div class="box">
        <div class="containar">
          <a href="{{ route('generator', ['type'=>'quiz']) }}" class="b btn btn-primary">Generate Quiz</a>
        </div>
      </div>
      <div class="box">
        <div class="containar">
          <a href="{{ route('generator', ['type'=>'assignment']) }}" class="b btn btn-primary">Generate Assignment</a>
        </div>
      </div>
    </div>
  </section>
@endsection
```

```
@extends('layouts.app')

@section('content')

  <section class="content container-fluid" >

   <div>

    <div class="box box-primary">

     <div class="containar">

      <button class="b btn btn-primary">{{ ucfirst($exam->subject->title) }} {{ $exam->type }} Exam</button>

      <a href="{{ route('exam.download', ['exam_id'=>$exam->id]) }}" class="b btn btn-success">Download Exam</a>

      <hr>

      <div class="box box-solid">

      <div class="box-header with-border">

       <i class="fa fa-text-width"></i>

 <h3 class="box-title">{{ ucfirst($exam->subject->title) }} {{ $exam->type }} Exam Report</h3>

      </div>

      <!-- /.box-header -->

      <div class="box-body">

       <dl>

        <dt>Subject Name</dt>

        <dd>{{ $exam->subject->title }}</dd>

        <dt>Number Of Question</dt>

        <dd>{{ $qs }} Questions</dd>

       </dl>

      </div>

      <!-- /.box-body -->
```

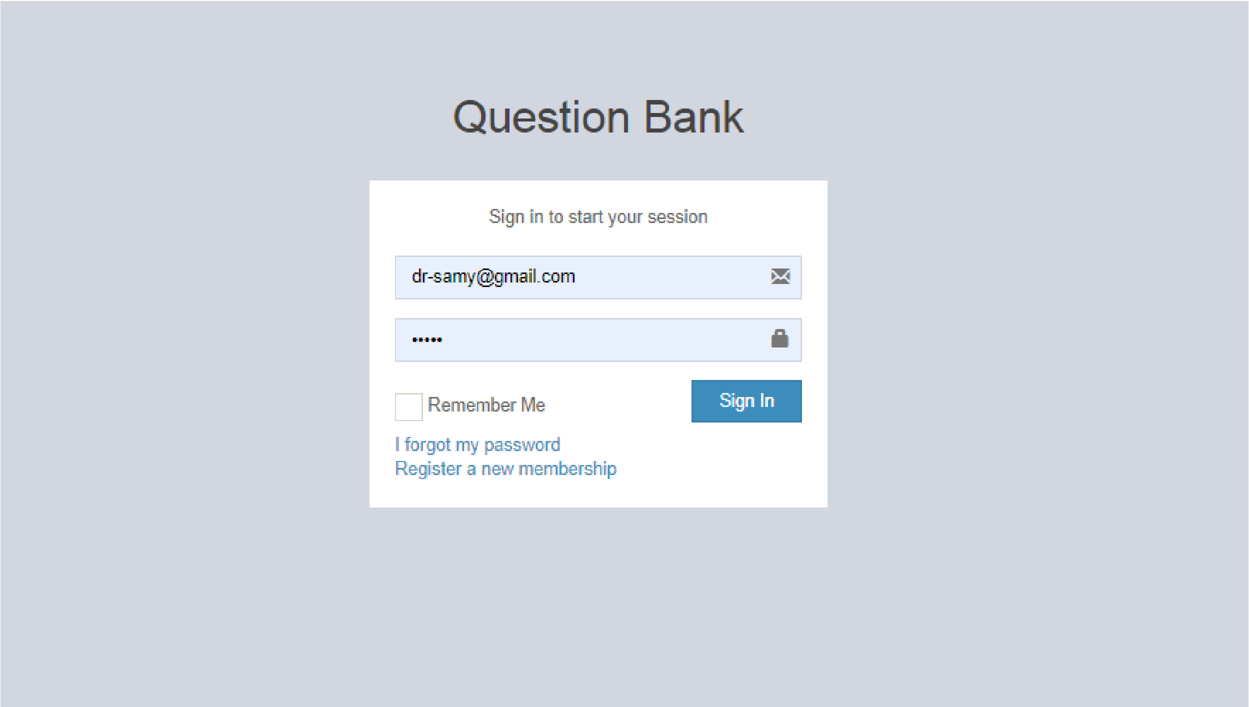
```
          </div>
<hr>
            <div class="desc">
             <div class="box box-primary">
            <div class="box-header with-border">
                Questions ILOs Report</div>
                    <div class="box-body table-responsive no-padding">
                    <table class="table table-hover">  <tr>
                        <th>ILOs</th>
                        <th># Questions</th>
                        <th>Rito</th>  </tr>
                        <tr>
                        <td>Intellectual skills</td>
                        <td>{{ $ilos['is'] }}</td>
                        <td>{{  round(($ilos['is'] / $qs ) * 100, 2) }}%</td>   </tr><tr>
                        <td>General and transferable skills</td>
                        <td>{{ $ilos['gt'] }}</td>
                        <td>{{  round(($ilos['gt'] / $qs ) * 100, 2) }}%</td>
                        </tr>
                        <tr>
                        <td>Professional and practical skills</td>
                        <td>{{ $ilos['pp'] }}</td>
                        <td>{{  round(($ilos['pp'] / $qs ) * 100, 2) }}%</td>
                        </tr>
                        <tr>
                        <td>Knowledge and understanding</td>
```

```
                 <td>{{ $ilos['ku'] }}</td>

                 <td>{{  round(($ilos['ku'] / $qs ) * 100, 2) }}%</td>

                 </tr>

              </table>

              </div>

              <!-- /.box-body -->

         </div>

        </div>

<hr>

                 <div class="desc">

         <div class="box box-primary">

        <div class="box-header with-border">

           Questions Chapters Report

        </div>

              <div class="box-body table-responsive no-padding">

              <table class="table table-hover">

                 <tr>

                 <th>Chapter Name</th>

                 <th># Questions</th>

                 <th>Rito</th>

                 </tr>

                 @foreach ($topics as $topic => $count)

                 <tr>

                 <td>{{ $topic }}</td>

                 <td>{{ $count }}</td>

                 <td>{{  round(($count / $qs ) * 100, 2) }}%</td>
```

```
                    </tr>

                    @endforeach

                  </table>

                  </div>

                  <!-- /.box-body -->

 </div>

           </div>

                      </div>  </div>

                                         </div>

                                                   </div>

                                                                   </section>

@endsection
```

4.4

Figure 4.2 LOGIN



Question Bank

Sign in to start your session

dr-samy@gmail.com

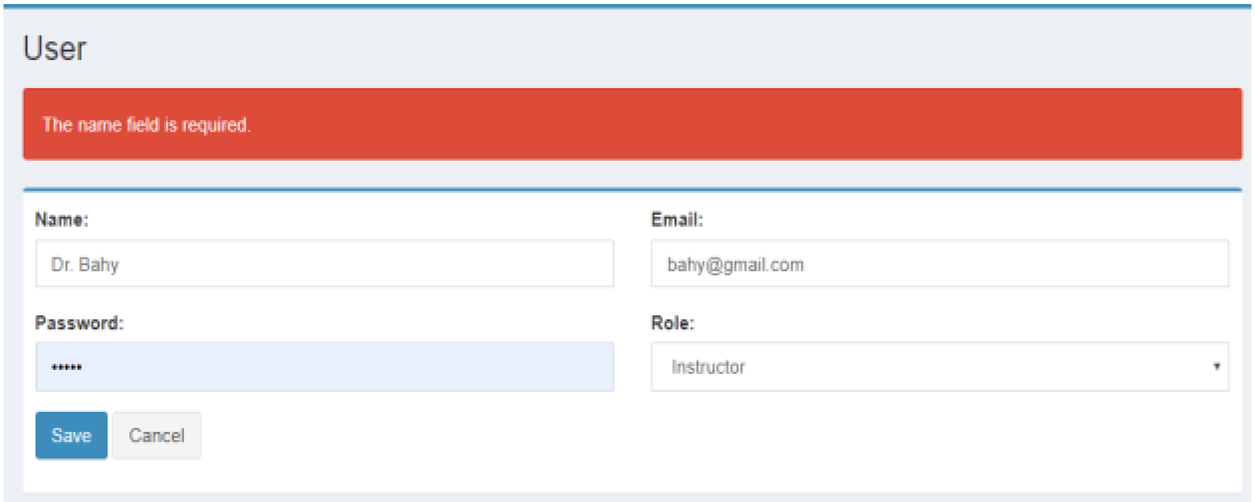
•••••

□ Remember Me

I forgot my password

Register a new membership

Figure 4.3 ADD USER



User

The name field is required.

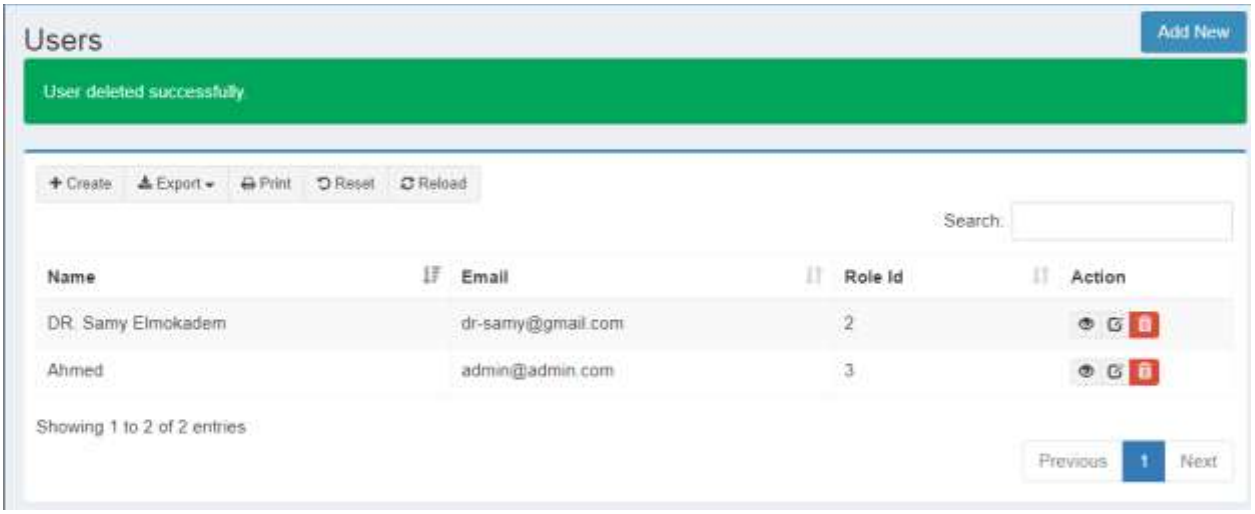
Name: Dr. Bahy

Email: bahy@gmail.com

Password: •••••

Role: Instructor

Figure 4.4 USERS CONTROL



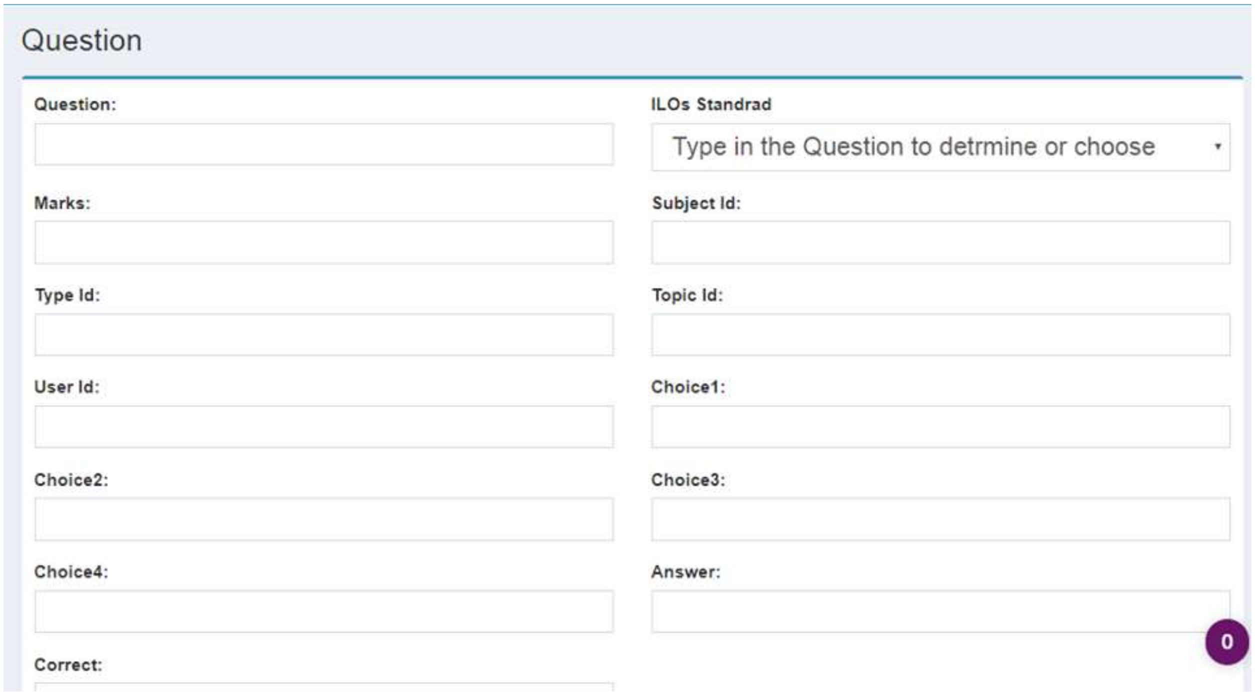
Users

User deleted successfully

Name	Email	Role Id	Action
DR. Samy Elmokadem	dr-samy@gmail.com	2	
Ahmed	admin@admin.com	3	

Showing 1 to 2 of 2 entries

Figure 4.5 ADD QUESTION



Question

Question:

ILOs Standrad: Type in the Question to detrmine or choose

Marks:

Subject Id:

Type Id:

Topic Id:

User Id:

Choice1:

Choice2:

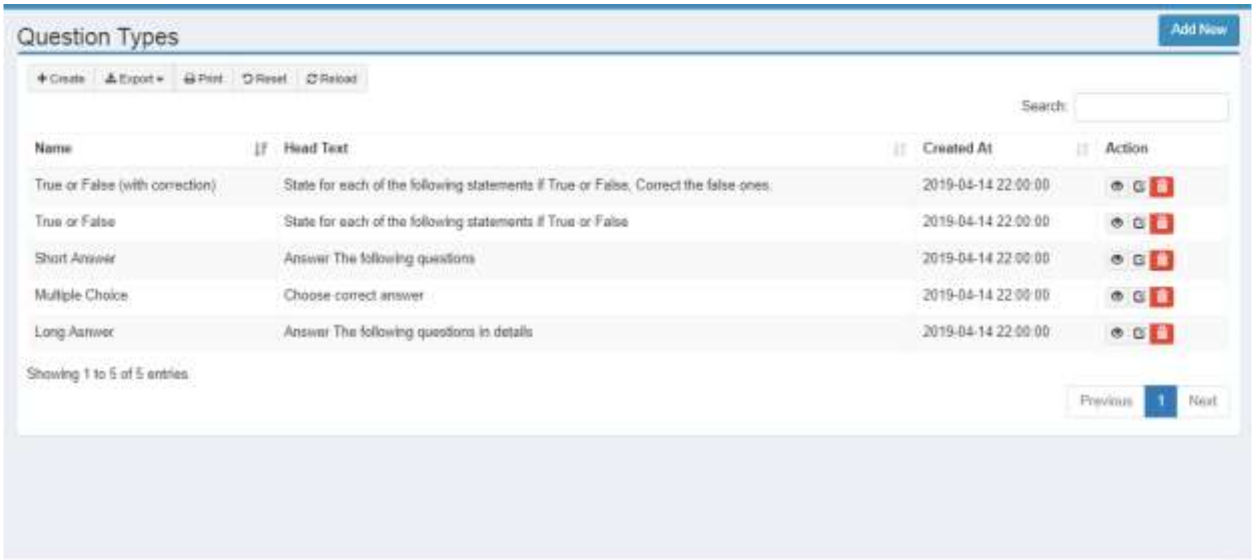
Choice3:

Choice4:

Answer:

Correct:

Figure 4.6 QUESTION TYPE

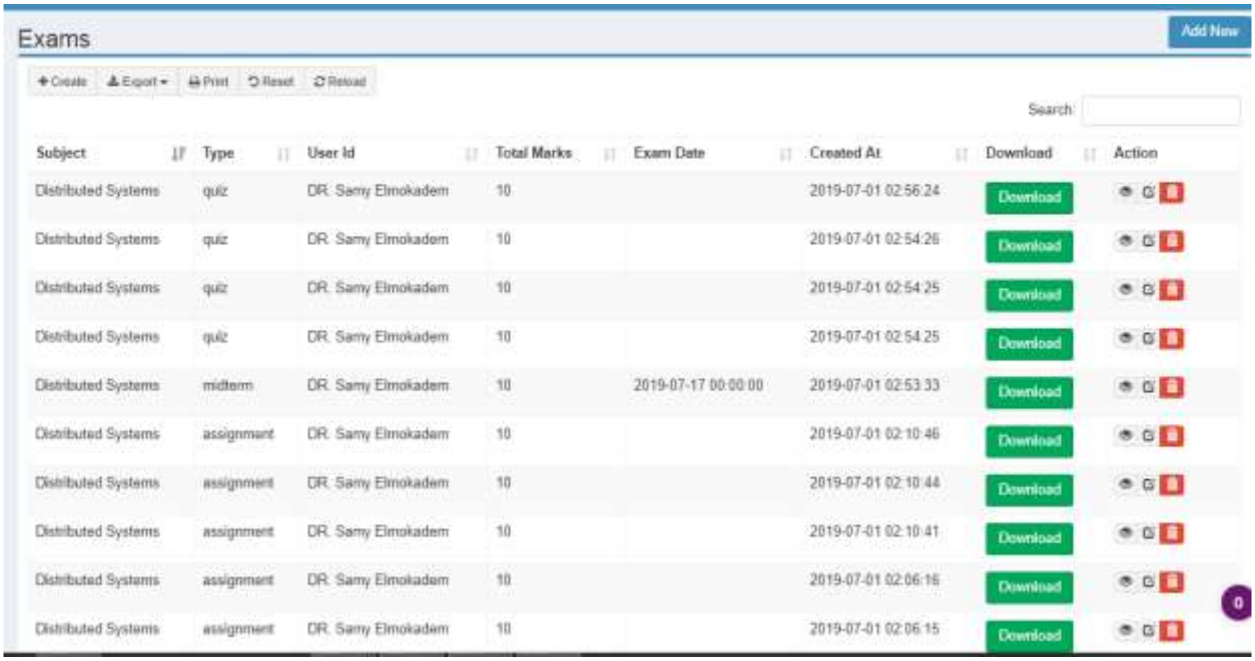


Question Types

Name	Head Text	Created At	Action
True or False (with correction)	State for each of the following statements if True or False, Correct the false ones	2019-04-14 22:00:00	
True or False	State for each of the following statements if True or False	2019-04-14 22:00:00	
Short Answer	Answer The following questions	2019-04-14 22:00:00	
Multiple Choice	Choose correct answer	2019-04-14 22:00:00	
Long Asnwer	Answer The following questions in details	2019-04-14 22:00:00	

Showing 1 to 5 of 5 entries

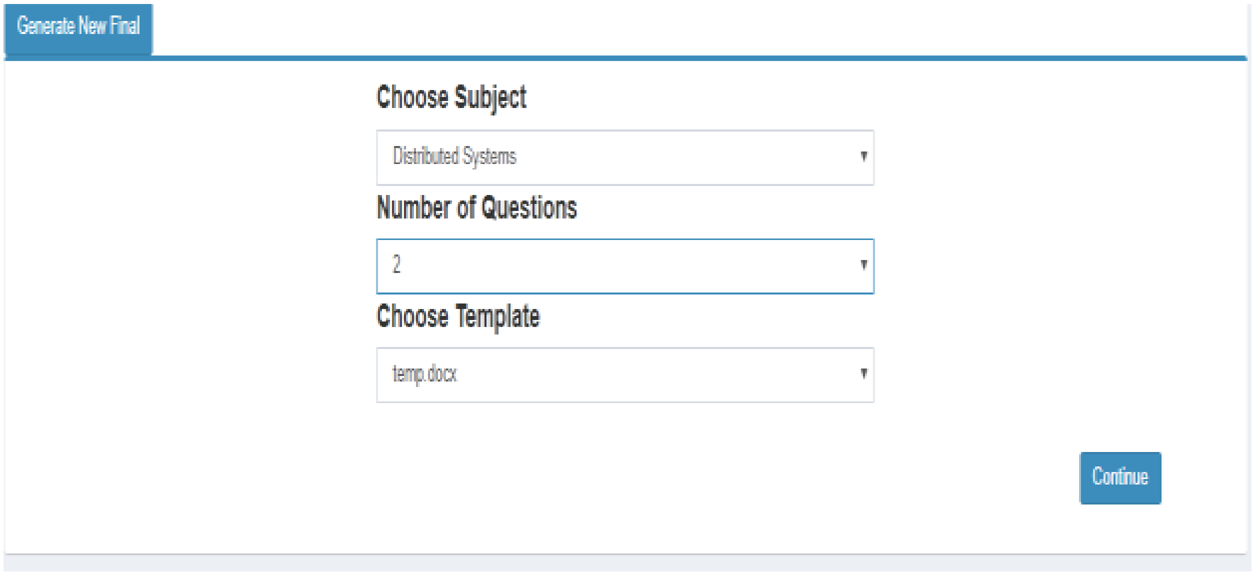
Figure 4.7 PERVIOUS EXAM



Exams

Subject	Type	User Id	Total Marks	Exam Date	Created At	Download	Action
Distributed Systems	quiz	DR. Samy Elmokadem	10		2019-07-01 02:56:24	Download	
Distributed Systems	quiz	DR. Samy Elmokadem	10		2019-07-01 02:54:26	Download	
Distributed Systems	quiz	DR. Samy Elmokadem	10		2019-07-01 02:54:25	Download	
Distributed Systems	quiz	DR. Samy Elmokadem	10		2019-07-01 02:54:25	Download	
Distributed Systems	midterm	DR. Samy Elmokadem	10	2019-07-17 00:00:00	2019-07-01 02:53:33	Download	
Distributed Systems	assignment	DR. Samy Elmokadem	10		2019-07-01 02:10:46	Download	
Distributed Systems	assignment	DR. Samy Elmokadem	10		2019-07-01 02:10:44	Download	
Distributed Systems	assignment	DR. Samy Elmokadem	10		2019-07-01 02:10:41	Download	
Distributed Systems	assignment	DR. Samy Elmokadem	10		2019-07-01 02:06:16	Download	
Distributed Systems	assignment	DR. Samy Elmokadem	10		2019-07-01 02:06:15	Download	

Figure 4.8 FINAL EXAM GENERATOR



Generate New Final

Choose Subject

Distributed Systems ▾

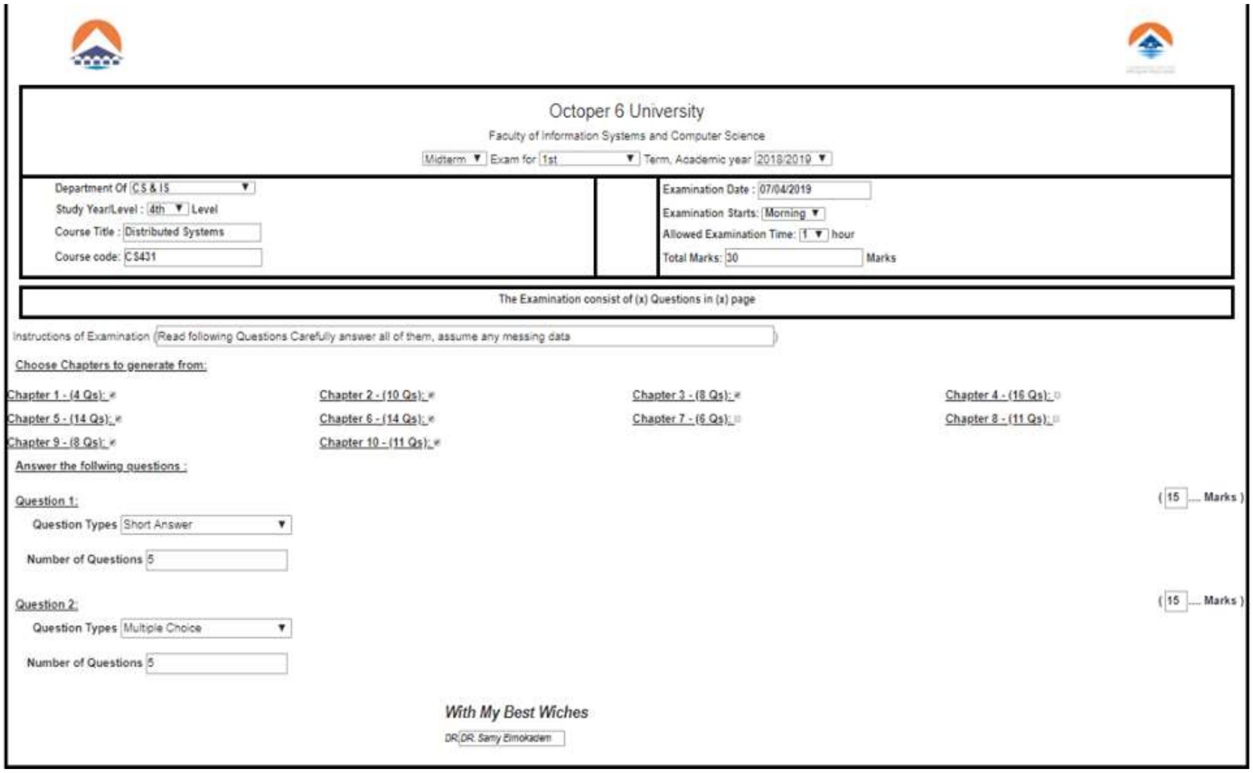
Number of Questions

2 ▾

Choose Template

temp.docx ▾

Figure 4.9 TEMPLET TO GENERATE EXAM



Octoper 6 University

Faculty of Information Systems and Computer Science

Midterm ▼ Exam for 1st ▼ Term, Academic year 2018/2019 ▼

Department Of C S & I S ▼

Study Year\Level : 4th ▼ Level

Course Title : Distributed Systems

Course code: C S431

Examination Date : 07/04/2019

Examination Starts: Morning ▼

Allowed Examination Time: 1 ▼ hour

Total Marks: 30 Marks

The Examination consist of (x) Questions in (x) page

Instructions of Examination (Read following Questions Carefully answer all of them, assume any messing data)

Choose Chapters to generate from:

Chapter 1 - (4 Qs): ☑	Chapter 2 - (10 Qs): ☑	Chapter 3 - (8 Qs): ☑	Chapter 4 - (16 Qs): □
Chapter 5 - (14 Qs): ☑	Chapter 6 - (14 Qs): ☑	Chapter 7 - (6 Qs): □	Chapter 8 - (11 Qs): □
Chapter 9 - (8 Qs): ☑	Chapter 10 - (11 Qs): ☑		

Answer the follwing questions :

Question 1: (15 .... Marks)

Question Types Short Answer ▼

Number of Questions 5

Question 2: (15 .... Marks)

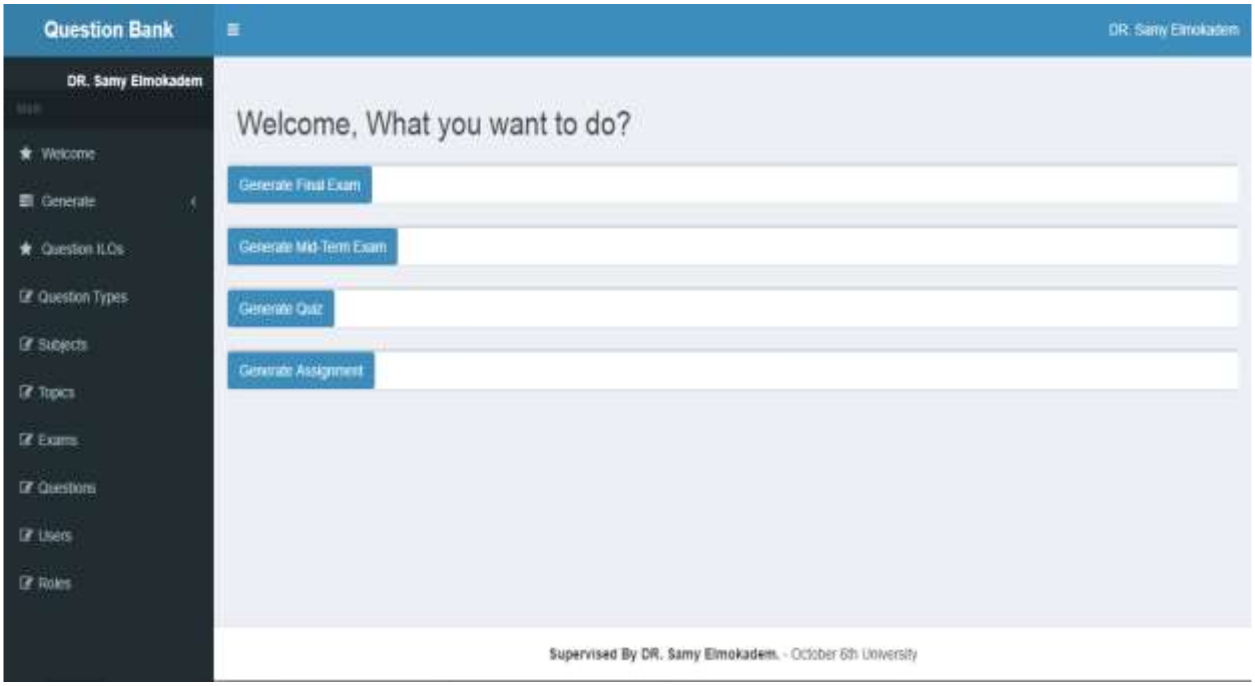
Question Types Multiple Choice ▼

Number of Questions 5

With My Best Wiches

DR DR. Samy Elmokadem

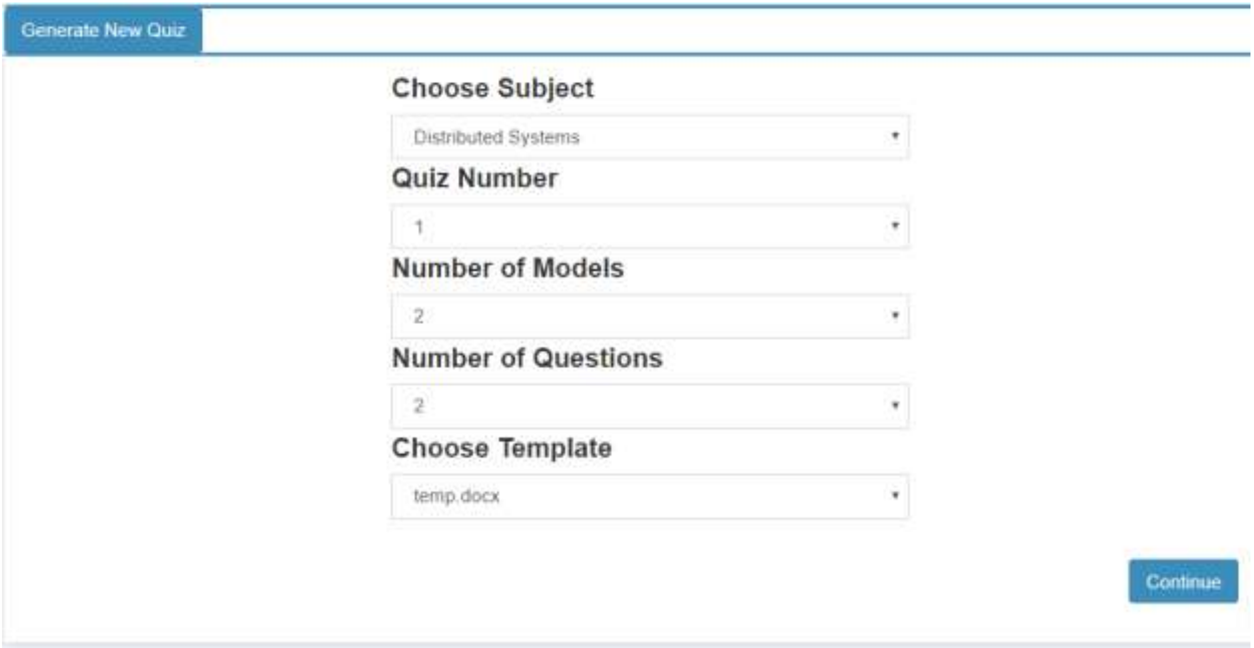
Figure 4.10 MAIN PAGE



Welcome, What you want to do?

Supervised By DR. Samy Elmokadem. - October 6th University

Figure 4.11 GENERATE QUIZ



Choose Subject

Distributed Systems

Quiz Number

1

Number of Models

2

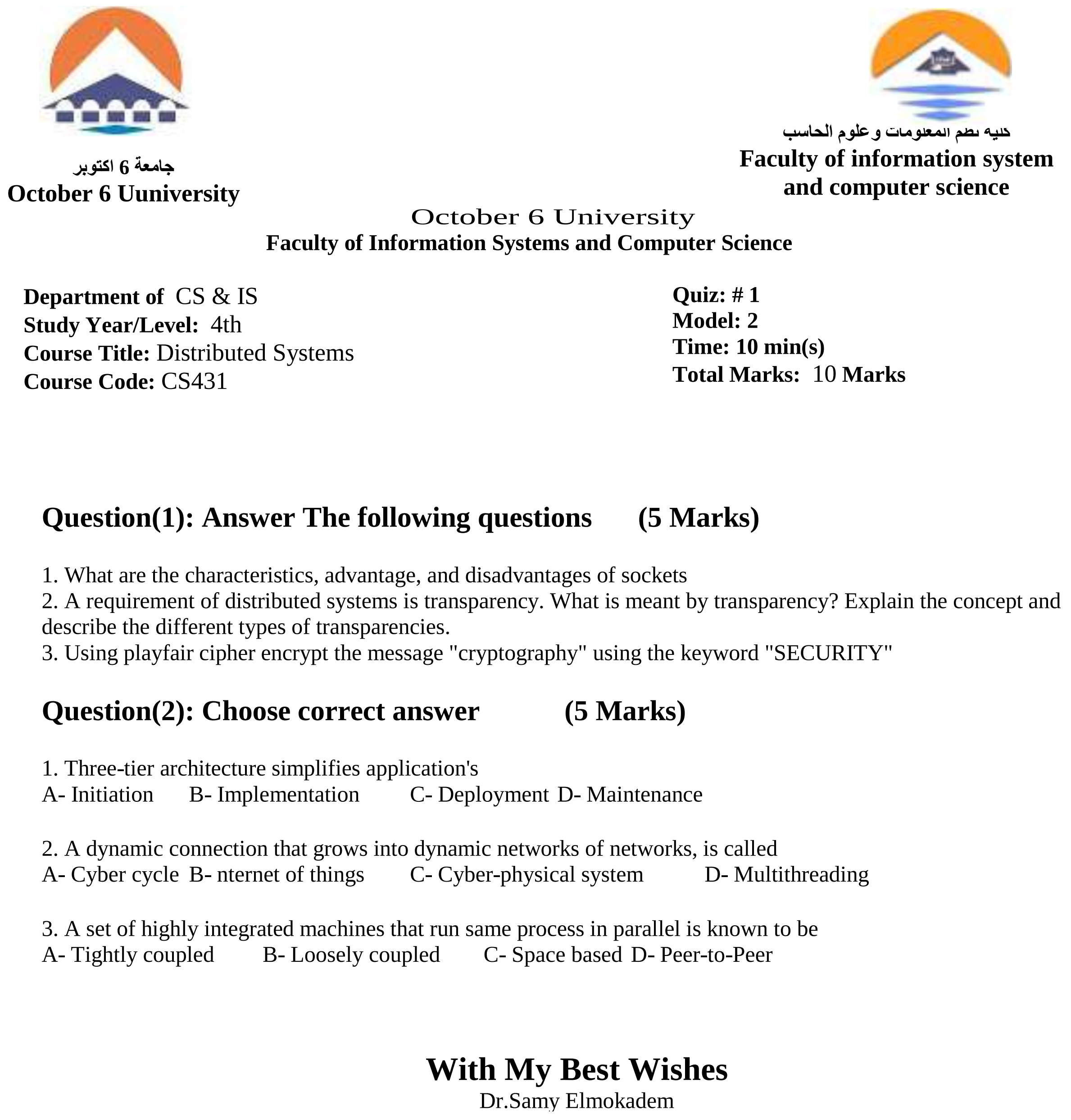
Number of Questions

2

Choose Template

temp.docx

Figure 4.12 QUIZ PAPER



كليه نظم المعلومات وعلوم الحاسب

Faculty of information system
and computer science

جامعة 6 اكتوبر

October 6 Uuniversity

October 6 University

Faculty of Information Systems and Computer Science

Department of CS & IS
Study Year/Level: 4th
Course Title: Distributed Systems
Course Code: CS431

Quiz: # 1
Model: 2
Time: 10 min(s)
Total Marks: 10 **Marks**

Question(1): Answer The following questions (5 Marks)

1. What are the characteristics, advantage, and disadvantages of sockets
2. A requirement of distributed systems is transparency. What is meant by transparency? Explain the concept and describe the different types of transparencies.
3. Using playfair cipher encrypt the message "cryptography" using the keyword "SECURITY"

Question(2): Choose correct answer (5 Marks)

1. Three-tier architecture simplifies application's
A- Initiation B- Implementation C- Deployment D- Maintenance

2. A dynamic connection that grows into dynamic networks of networks, is called
A- Cyber cycle B- nternet of things C- Cyber-physical system D- Multithreading

3. A set of highly integrated machines that run same process in parallel is known to be
A- Tightly coupled B- Loosely coupled C- Space based D- Peer-to-Peer

With My Best Wishes

Dr.Samy Elmokadem

Figure 4.13 EXAM PAPER

جامعة 6 اكتوبر
October 6 Uuniversity

كلية نظم المعلومات وعلوم الحاسب
Faculty of information system and computer science

October 6 University
Faculty of Information Systems and Computer Science
final exam for 1st term, Academic year 2018/2019

Department of CS & IS
Study Year/Level: 4th
Course Title: Distributed Systems
Course Code: CS431

Examination Date: 2019-07-24
Examination Starts: Morning
Allowed Examination time: 2 hour(s)
Total Marks: 60 **Marks**

The Examination consists of 4 Question(s) in 3 Page(s)

Instructions of Examination (Read following Questions Carefully answer all of them, assume any messing data)

Question(1): Answer The following questions (20 Marks)

1. A requirement of distributed systems is transparency. What is meant by transparency? Explain the concept and describe the different types of transparencies.
2. What is meant by process, kernel, and file system persistence IPC
3. What are the ways of implementing message queue attached to link
4. What are the challenges addressed in the design of secure distributed system
5. What is an open distributed system and what benefits does openness provide?

Question(2): State for each of the following statements if True or False (10 Marks)

1. Latency is measured as the processing power after send operation before data starts to arrive at destination host.
2. Resource sharing is not the main motivating factor for constructing distributed systems
3. An Architectural model of a distributed system is concerned with the placement of its parts and relationship between them.
4. Distributed systems are suitable for building global e-business
5. Encryption is the process of encoding a message in such a way as to hide its contents

Question(3): Choose correct answer (20 Marks)

1. DLP stands for
A- Data-level processing B- Degree-level processing C- Data-level parallelism D- Degree-level parallelism

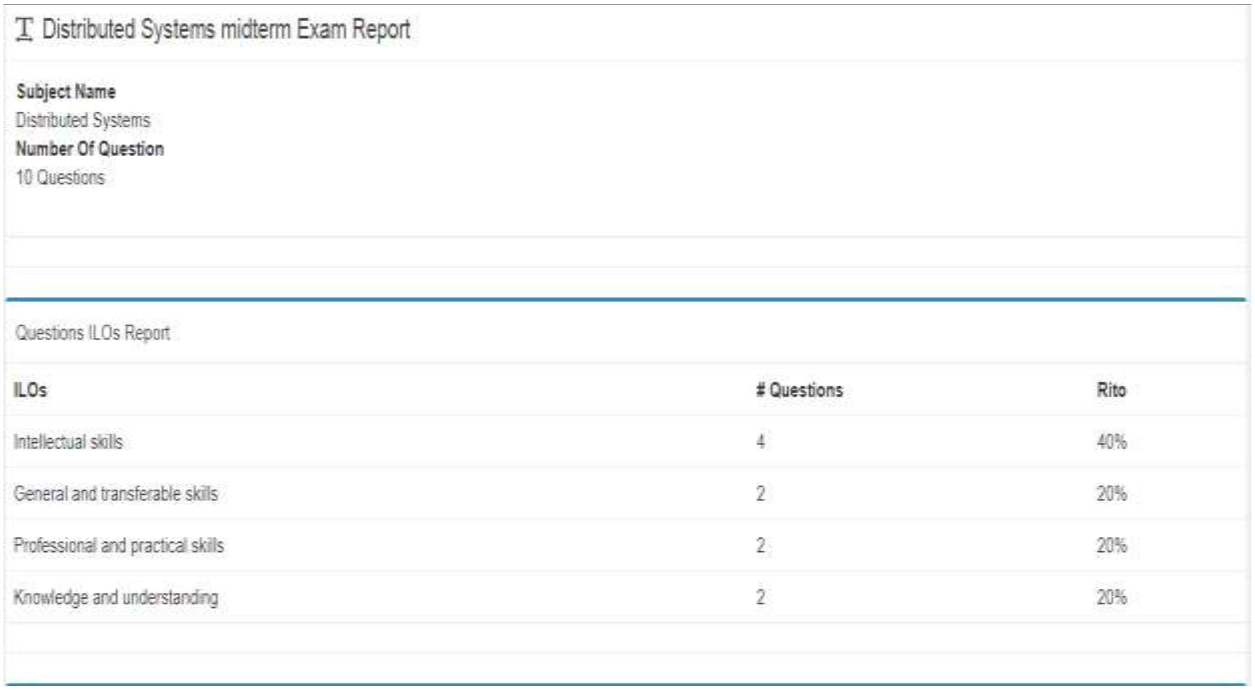
2. To provide high-throughput service is measures taken by
A- Efficiency B- Adaptation C- Dependability D- Flexibility

3. The local operating system on the server machine passes the incoming packets to the:
A- server stub B- client stub C- client operating system D- none of the mentioned

4. Three-tier architecture simplifies application's
A- Initiation B- Implementation C- Deployment D- Maintenance

5. A model in which components of a software system are shared among multiple computers is known as
A- Parallel cloud B- Distributed cloud C- Virtualized cloud D- Centralized cloud

Figure 4.14 ILOS REPORT



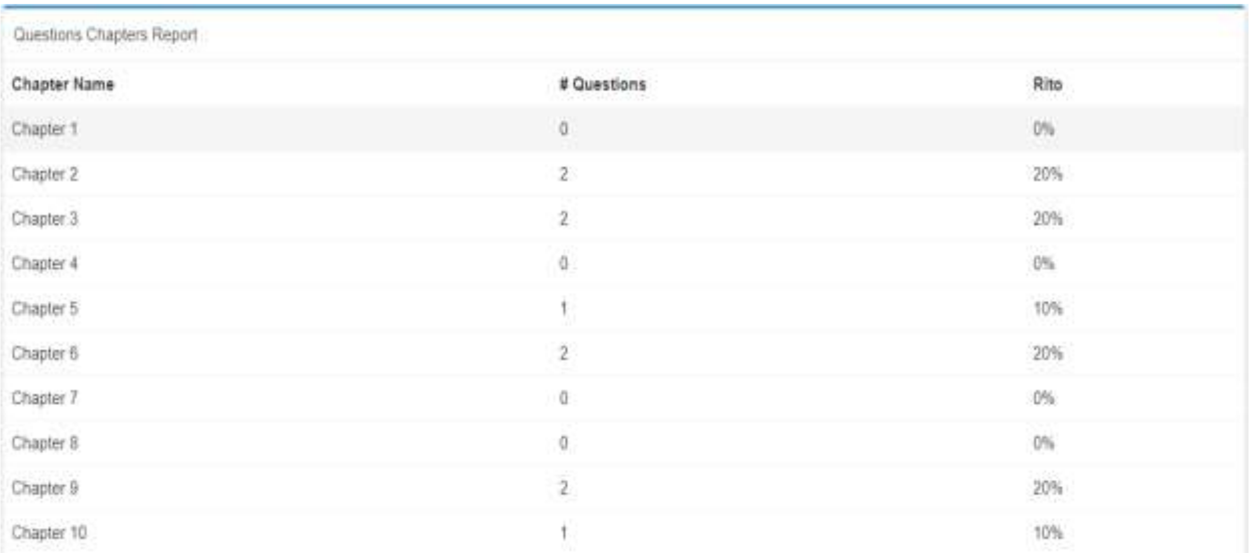
Distributed Systems midterm Exam Report

Subject Name
Distributed Systems
Number Of Question
10 Questions

Questions ILOs Report

ILOs	# Questions	Rito
Intellectual skills	4	40%
General and transferable skills	2	20%
Professional and practical skills	2	20%
Knowledge and understanding	2	20%

Figure 4.15 CHAPTER REPORT



Questions Chapters Report

Chapter Name	# Questions	Rito
Chapter 1	0	0%
Chapter 2	2	20%
Chapter 3	2	20%
Chapter 4	0	0%
Chapter 5	1	10%
Chapter 6	2	20%
Chapter 7	0	0%
Chapter 8	0	0%
Chapter 9	2	20%
Chapter 10	1	10%

Chapter 5

5.1 Literature Survey

1. Existing Paper-based Systems

The existing system for Question Paper Generation requires human staff to chalk out questions that appear in the question paper. professors select the questions according to the syllabus and pattern as prescribed by the curriculum. The question paper then may be referred to a higher authority who has the final say in these matters.

2. Limitation of Paper-based Systems

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, 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: -

a) Lack of storage space

b) Prone to damage

c) Inefficient document transportation

d) Supply costs

e) Poor environmental credentials

f) Limited collaboration

g) Editing problems

3. Existing Question Paper Generation Systems

As the manual generation of a balanced question paper by an individual is quite complex, the blending of technology into teaching and learning process is inevitable. A simple and efficient way for an examination paper generation is provided. A three tier model is provided in this framework Generation of Examination Papers is governed by the Syllabus Engine, Pattern Composer and Question Aggregator. The generated question paper is based on the pattern or skeleton of the course. Another component called Bank Management takes care of User Rights and Privilege assignment. Questions are entered through the Question Aggregator. The attributes related to questions are type, marks and complexity.

All these attributes are efficiently used during Question Paper Generation

The paper generator selects a question according to the pattern and complexity This engine also introduces a marking systems wherein any selected question is marked so that it might not be selected again.

Automatic Question Paper Generation System using Randomization Algorithm‟ describes a system which uses a shuffling algorithm (existing algorithm) as a randomization technique : The system defines several modules such as user administration, subject selection, difficulty level specification, question entry, question management, paper generation, and paper management It supports multiple languages. Also, mathematical formulae and diagrams can be integrated in questions.

However, this system fails to utilize the highly efficient marking system. So, questions once selected may be repeated in subsequent papers. This is a limitation of this system.

The Question Paper Generator System‟ has provided a ready to use built-in question bank, The paper aptly describes CQZ (Cloze Question Generation) putting more emphasis on the actual type of the questions . Another paper on „Automatic Test Paper Generation Based on Ant Colony Algorithm‟ has implemented a complex but highly efficient Ant Colony Algorithm . It requires building of a mathematical model of constraint according to the requirements of the paper.

5.2 Future work

We have addressed only a small part of the problem of automatically generating questions.

However, there are many areas for future work. One particularly interesting area for

future work is the exploration of alternative of Question generation. In (5.2.1), we discuss potential

approaches to generate various types of exams that rely on the questions entered by user in the question bank that we intend to improve in the future to make it easier for the user to submit large amount of questions and determine its ILOs standard in-time.

We also disscused generation of quizzes, we want to extend this feature to include generate and mark quizzes online.

We talked about a new feature (reports) every exam will be generated will generate a report with it but we wanted to extend it more to include mark of every questions and more statistical information about the questions.

Currently we don't support the model answer of the question paper but in future work will add the ability to generate the exam and the model answer.

5.2.1 Alternative Question generation, techniques and submitting

Currently there's one way to enter question and that is enter one by one, but to make it more convenience to submit questions we would like to extend this to support submission by excel sheet and determine the question ilos type while insertion.

5.2.2 Conclusion

This is a web-Based System which is implemented as a real-time application. Our future effort is to employee different types of randomization as well as addition to online quiz mark and generation.

we can enhance the same software by making provision to extend question generation to online quiz. This system can emphasize more percentage of difficulty when question papers are referred from the bank.

by providing controlled access to the resources, we also considered the importance of randomization in the task of paper generation. Our system has deployed an efficient algorithm which is totally randomized and avoids repetition of questions is consequent question papers, making it impossible to derive any pattern in the papers. We distinguish between administrators, instructors and students by their tasks. Therefore, the resultant Question Paper Generation provides improvement in terms of controlled access to the resources, random generation of question papers and a secure platform.

5.3 References

[1] Automated Question Paper Generation System Rohan Bhirangi, Smita Bhoir Computer Engineering Department, Ramrao Adik Institute of Technology Navi Mumbai, Maharashtra, India April 2016 ISSN: 2278 -9359 (Volume-5, Issue-4)

[2] Surbhi Choudhary, Abdul Rais Abdul Waheed, Shrutika Gawandi and Kavita Joshi, "Question Paper Generator System," International Journal of Computer Science Trends and Technology, vol. 3, issue 5, Sept – Oct 2015.

[3] "Towards Automatic Topical Question Generation"
Y llias ,Chali Sadid , A.Hasan University of Lethbridge,
Lethbridge, AB, Canada
[4] https://www.statpac.com/surveys/questionqualities.
htm
[5] "Using Questions to Teach Better"Rick Garlikov -
http://www.garlikov.com/teaching/usingquestions.html

[6] "Automatic Factual Question Generation from Text - Semantic Scholar" issued in 2011.

https://pdfs.semanticscholar.org/ebd9

[7] http://forums.codeguru.com/showthread.php?518588-Question-Generator-Algorithm

[8] http://www.sciencemag.org/content/228/4698/456.short.

[9] https://act.org/research/researchers/reports/pdf/ACT_RR93-09.pdf

[10] http://www.w3schools.com/

[11] https://stackoverflow.com/

[12] https://laravel.com/

[13] https://www.wikipedia.org/