Optical Mark Recognizer for QAU Entry Test Checking.

A report submitted as a partial fulfillment of the

requirements for the award of degree of

M.Sc. in Computer Science

By:

Noorulain

Under the supervision of:
Dr. Onaiza Maqbool.
Assistant Professor

Dedicated to

Dr. Qurratulain.

Project Brief

Project Title:	Optical Mark Recognizer for QAU Entry Test Checking.
Undertaken By:	Noorulain
Supervised By:	Dr. Onaiza Maqbool Assistant Professor, Department of Computer Science, Quaid I Azam University Islamabad.
Tools Used:	MATLAB version 7.10, Microsoft Visual Studio 2010, WAMP Server version 2.2
Operating system:	Microsoft Windows 7
System Used:	Intel(R) Core 2 Duo CPU 2.40 GHz.
System Memory:	2GB RAM

Acknowledgements:

First of all I would like to extend my sincere and humble gratitude to Almighty Allah whose blessings and guidance has been a real source of all my achievements in my life.

I am grateful to my supervisor **Dr. Onaiza Maqbool** for her sincere help and guidance during my studies and project work.

It is my bounden duty to pay tributes to my worthy teachers and staff members of Department of Computer Science Quaid-i-Azam University Islamabad. Greatest thanks go to all my teachers for their kind help and guidance.

I am indebted to my parents for their prayers and moral & financial support during my studies. I am also thankful to my brothers and sisters for their support and prayers.

My sincere thanks to all border and day scholar friends for their sincere help, encouragement and enjoyable company during my student life. They are really great source of inspiration for me in my difficulties.

At the end I would like to thank all of my class fellows, seniors and juniors for their cooperation and help.

Noor ul ain.

Abstract:

Optical Mark Recognition (also called **Optical Mark Reading** and **OMR**) is the automated process of capturing the human-marked data which is in the form of bubbles, squares or tick marks. This technique is widely used in various applications like exam evaluation, automated attendance marking, community surveys etc.

Many traditional OMR (Optical Mark Recognition) devices make use of commercially available dedicated OMR scanners and use especially designed thick sheets.

The present work proposes to automate the test checking on a desktop computer using ordinary scanner. Machine learning algorithm is used to recognize marked answers on scanned images, classification of marked answers as correct or incorrect, compilation and displaying of results.

A standardized sheet is designed for conducting exam.

Table of Contents

Chapter: 1

Introduction

1.1 Optical Mark Recognition:

Optical mark recognition (OMR) is the automated process of capturing the data which is in form of some marks like bubbles, squares or ticks. This is done by contrasting reflectivity at predetermined positions on the sheet. When we shine a beam of light onto the paper, the scanner detects the marked region as it reflects less light than the unmarked area on the paper.

The most familiar application of OMR is multiple choice question examinations, where students mark their answers and personal information by darkening the circles on a pre-printed sheet. This sheet is then evaluated using image scanning machine.

1.2 History:

1.2.1 IBM 805 Test Scoring Machine:

The first test scoring machine was IBM 805 marketed by IBM in early 1930's [1]. The design of this machine was based on electric conductivity.

It was devised by a high school science teacher R. B. Johnson in Michigan USA. R. B. Johnson devised this machine for his own use to record students' answers and to compare them to an answer key set up on his machine. IBM bought the rights to his invention and the machine was soon in market by the name of "IBM 805 Test Scoring Machine".

Tests to be scored by the machine were answered by marking separate answer sheets, which were then dropped into the machine for processing.

Inside the machine was a contact plate with an equal number of contacts corresponding to the number of answers.

A scoring key separated the answers into two groups, the "rights" and the "wrongs".

When an answer sheet was inserted into the machine an amount of current equal to total rights and total wrongs was allowed to flow. When the operator manipulated the controls, the 805 indicated the scores. The IBM 805's speed was limited only by the operator's ability to insert sheets in the machine and record the scores. An experienced operator could record scores on answer sheets at the rate of about 800 sheets per hour.

1.2.2 Optical Mark recognition:

In 1960's IBM replace conductivity method of IBM 805 by optical mark sense system. The first optical mark recognition scanner used a mimeograph paper-transport mechanism directly coupled to a magnetic drum memory. Although it was not a general purpose computer, it made extensive use of computer technology.

IBM marketed optical mark-sense test scoring machine as the IBM 1230 Optical mark scoring reader. This machine allowed IBM to migrate a wide variety of applications developed for its mark sense machines to the new optical technology.

OMR has been used in many situations as mentioned below. OMR is still used extensively for surveys and testing though.

1.3 When to use OMR:

OMR based evaluation is preferred over the manual methods when:

- A large volume of data is to be collected and processed in short period of time.
- Data is to be collected from large number of sources simultaneously
- Questionnaires consist of multiple choice questions or selection of categories.
- Survey collectors are limited.
- Very high accuracy is required.

1.4 Mechanism of OMR:

A typical OMR machine consists of three main units:

1.4.1 Feeding Unit:

The feeding unit is responsible of picking sheets one by one and letting the sheets go through photoelectric conversion unit without any discrepancies.

1.4.2 Photoelectric Conversion Unit:

The photoelectric conversion unit irradiates light to the surface of the sheet by some light source like lamp, and then converts the intensity of reflection of light to an electric signal by lens and sensor and inputs the signal to the image memory. The electric signal is recognized as '0' for the white light and '1' for the dark light according to the strength of the reflected light.

There are two processors in this unit: recognition processor and control processor. The recognition processor reads the mark from the image, recognizes it and sends the corresponding signal to control processor. The control processor produces data, and at the same time controls all units of OMR system.

1.4.3 Recognition Control Unit:

Mark recognition is a kind of pattern recognition technique. This technique has repeatedly been improved and has steadily brought about good results. The recognition process was based on a hardwired logic in the early days. The process is now conducted by software with a recognition-specialized processor or a microcomputer. Recognition by software has brought about more flexibility in the recognition process, increase in the reading methods, advancement in accuracy of reading, and simultaneous input of different types of sheets.

1.5 Problem Definition:

Quaid i Azam University (QAU) has impressively grown in size and activities during the last few decades of its existence. At present, the university has four faculties and nine other teaching and research institutes. These faculties have many departments.[1]

Admissions to these departments are mainly through entrance examination conducted by the university.

Till now, most of the entrance exams have been evaluated manually which proves to be a tedious task considering the large number of candidates appearing for the exams.

In the present work, we propose to automate the evaluation system of these exams that makes this evaluation possible on a desktop computer. An ordinary scanner is used to get the images of marked sheets. Moreover the OMR sheets used by the system are printed on ordinary printing paper. It is expected that this system will not only reduce the burden of checking the bulk of papers but it will also ensure accuracy and impartiality.

1.6 Objectives:

The objectives to be met by the implementation of this project are:

- Exploring the use of machine learning for recognizing the marked answers from scanned images of answer sheets.
- Making such a system work in limited resources i.e. neither special purpose dedicated scanner is used, nor is the thick OMR sheet required.
- Providing the users with a comprehensive solution for the problem of automation of test checking.

1.7 Challenges:

Challenges to be faced in the implementation of such a system are as follows:

- In order to evaluate an OMR sheet, first of all it is required to capture a crisp image of the sheet.
- Since the system is built to work with an ordinary flat-bed scanner with no feeding unit (dedicated scanners used with OMR have specially designed feeding units) so there is a strong possibility of tilted images or images having other discrepancies.
- Therefore a lot of processing on the data (images) is required before using it.
- OMR sheets available in the market are designed in accordance with the special purpose scanners used with OMR. In this project we have to devise a solution that works with an ordinary flat-bed scanner therefore OMR sheet selection and designing are challenging tasks.

1.8 What's New????

In this section we identify the new features of the system i.e. the features not present together in previously implemented systems.

- OMRs available in the market use dedicated scanners with built in micro-controllers for the recognition of marks whereas this project offers a low cost solution for automation of test checking as it does not require any special purpose dedicated scanner.
- OMR machines present in the market are connected to a computer for exporting and saving results. This project is to make OMR possible on a desktop computer and the product will be able to maintain the record of each and every test that gets checked.

- System generates its own OMR sheets printed on ordinary printing paper.
- System maintains record of the previous tests.

1.9 Outline of the Report:

In the following chapters an effort has been put to describe the software based OMR for QAU.

Chapter 2 gives the background of data mining, machine learning and classification. In Chapter 3 the analysis performed over user requirements is discussed. Chapter 4 contains complete project design. Chapter 5 is about implementation phase of the project. Chapter 6 discusses the testing techniques used to test the system and chapter 7 is about future enhancements.

Chapter: 2

BACKGROUND

The amount of data in the world and our lives seems ever increasing.
New hardware and software technologies, inexpensive storage medium and online storage make it easy to save things that we would have trashed.
Everyday everything that happens anywhere in this world gets recorded.
The amount of data in the world doubles every 9 months.
Now the question is…. How to make the best use of this data?

Before going towards the answer of this question we need to know the difference among *data, information* and *knowledge.*
These terms are mostly used for overlapping concepts but the basic difference is "the level of abstraction".
Data is the lowest level of abstraction, information is the next level, and finally, knowledge is the highest level among all three.

2.1 Data:

Known ***raw*** facts and figures that have no meaning are called data. Data are unorganized and unstructured facts.
Data on its own carries no meanings. For data to become information, it must be processed and interpreted.

2.2 Information:

Processed form of data from which useful conclusions can be drawn is called information.

Information is organized and structured data presented in a meaningful way in the form of tables charts etc.
The patterns, associations, or relationships among the *data in a large data set* can provide *information*.

2.3 Knowledge:

According to cognitive science knowledge is symbolic representations stored in the brain (Newell, 1990).
Now this view about knowledge is changing. Knowledge is not something that is stored in the brain. Knowledge is created in a situation by relating the stored information, and is never again used in exactly the same way.

2.4 Data Mining:

Data mining (also called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information.

The goal of the data mining process is to extract knowledge from an existing data set and transform it into a human-understandable structure for further use.

Data mining is about finding relationships and patterns (in databases) that have not previously been discovered.

We need techniques for finding such patterns in data for making predictions. The techniques used for data mining are usually based on machine learning.

2.5 What is Machine Learning??

Machine learning is defined as a process by which machines change their behavior in a way that makes them perform better in future.

In case of machines, learning is more about performance rather than knowledge.

Learning can be tested by observing present behavior and comparing it with the past behavior.

2.6 Why Machine Learning??

Techniques other than machine learning have been around for some time but do not perform well when there is a large amount of data. Machine learning techniques prove to be useful because of their ability to generalize.

Machine learning techniques perform accurately on new unseen examples if trained well on a finite set of data.

2.7 Techniques of Data Mining:

Several data mining techniques have been developed and used in data mining projects. Some of these techniques are listed below. We will briefly define these techniques

2.7.1 Classification:

Classification is a data mining technique used to classify each item in a set of data into one of predefined set of classes or groups.

This data mining technique is discussed with detail in section 2.8 of this chapter.

2.7.2 Association:

Association is one of the best known data mining technique. In association, a pattern is discovered based on a relationship of a particular item with other items in the same transaction.

2.7.3 Clustering:

Clustering is a data mining technique that makes meaningful or useful cluster of objects that have similar characteristic using automatic technique. Different from classification, clustering technique also defines the classes and put objects in them, while in classification objects are assigned into predefined classes.

2.7.4 Prediction:

Prediction as its name indicates, predicts, but unlike classification it predicts continuous valued functions.

2.7.5 Sequential Patterns:

Sequential patterns analysis is one of data mining technique that seeks to discover similar patterns in data transaction over a business period.

2.8 Classification:

Classification is a classic data mining technique based on machine learning. Classification is a data mining function that assigns items in a collection to predefined target categories or classes based on a number of observed attributes related to that object.

The goal of classification is to accurately predict the target class for each case in the data.

Classifications are discrete and do not imply order.

2.9 Classifiers:

Different classification algorithms use different techniques for finding relationships. These relationships are summarized in a model, which can then be applied to a different data set in which the class assignments are unknown.

A classifier is a model that takes objects as input and assigns each one to a predefined target class.

Examples of classifiers:

- Decision Trees
- Support Vector Machines
- Naïve Bayes and Bayesian Belief Networks
- Neural Networks/Artificial Neural Networks

2.9.1 Neural Networks:

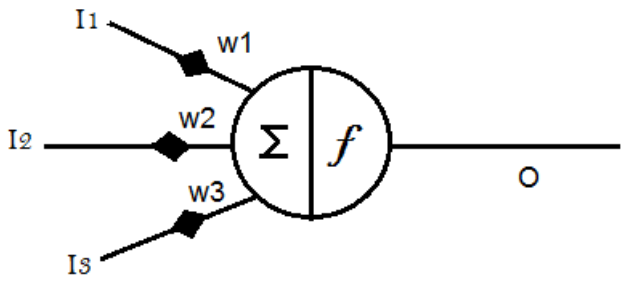
Any processing device, algorithm, or hardware, whose design and functioning is modelled on the design and functioning of human brain is called Artificial Neural Network.

An ANN is a network of many very simple processing units (sometimes also called perceptron), each possibly having a small amount of local memory.

These basic units of an ANN resemble neurons which are the basic structural and functional units of human brain.

Structure:

It receives input from some other units, or from an external source. Each input has a weight 'w' associated with it, which can be modified to model synaptic learning.



The processing unit sums the inputs and then applies an activation function (transfer/threshold function).

Activation function is a way that the output from individual neuron is scaled to appropriate value.

O is the output line.

2.9.2 Types of Neural Network:

On the basis of connection type, neural networks are divided into two types.

Feed forward (Static):

In a feed-forward network a unit transfers its output to all the units on the next layer, but there is no transferring to the previous layers.

Feedback (dynamic):

In a feedback networks, a unit can transfer its output to the units on previous layers, thereby forming directed cycles.

2.9.3 Learning in neural networks:

The automated way of adjusting the weights so that the network can perform a specific task is referred to as learning.

Learning can be of two types.

Supervised Learning:

In supervised learning, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then calculated, causing the system to adjust the weights which control the network. This process occurs over and over as the weights are continually tweaked.

Unsupervised Learning:

In this mode of learning outputs are not provided. i.e. the examples given for the learning are not labeled, network itself finds the correlations between the data.

2.9.4 Why Neural Networks???

Neural networks are a powerful technique to solve many real world problems.

The best quality of neural networks is the ability to train. Neural network learns from experience and improves its performance.

Neural networks are error-tolerant.

Neural networks are able to deal with incomplete information or noisy data.

Neural networks can be very effective in problems like classification, especially when we have a large amount of data and it is not always possible to define the rules or steps that lead to the solution of a problem.

Chapter 3:

SOFTWARE REQUIREMENTS

3.1 INTRODUCTION

3.1.1 Purpose:

The purpose of this document is to present detailed requirements for OPTICAL MARK READER FOR QUAID I AZAM UNIVERSITY. It will explain the purpose and features of the system, the interfaces the constraints under which it must operate and how the system will react to the external stimuli. This document is intended for both the stakeholders and the developers of the system.

3.1.2 Definitions, Acronyms and abbreviations:

- OMR Optical Mark Recognition
- QAU Quaid i Azam University
- GUI Graphical User Interface
- ADF Auto Document Feeder

3.1.3 Scope of the Document:

The specifications pertain to the PC software OPTICAL MARK READER FOR QAU. This title is a software application that makes OMR possible in a desktop machine. It provides a user friendly environment. This document provides an overview of the rule set and the services provided by the software. This document also discusses the "scenarios" that specify graphical layout for this software.

2.1.4 Project scope:

The software will be a standalone application that makes OMR possible on a desktop computer with an ordinary scanner to scan the OMR sheets marked by the students.
The system uses machine learning algorithm to recognize marked answers on the sheets, classification of marked answers as correct or incorrect, and finally compilation of results.
Since the software is built for Quaid I Azam University, therefore, the users of this system are the faculty members of QAU.

3.1.5 Overview:

Section 3.2 covers information applicable to the end users. It describes interaction at the human-to-system level as well any system/hardware interfacing that is pertinent to the user. This section also covers

information applicable to developers. It describes the functionality to be fulfilled by the system.
Section 3.3 mentions the user characteristics.

3.2 OVERALL DESCRIPTION

3.2.1 Product Perspective

OMR for QAU is a software based solution for the automation of checking of tests that are based on multiple choice questions. The term "software based" we mean that most of the work will be done by the software and no specific special purpose hardware would be used. i.e. no special purpose dedicated scanners are required.

3.2.2 Product Functions:

These are the services that the application should provide. Use case analysis is done to capture the functional requirements of the system. A use case is a list of steps defining interactions between an actor and a system to achieve a goal. The actor can be a human or an external system.

Two types of goals are discussed here

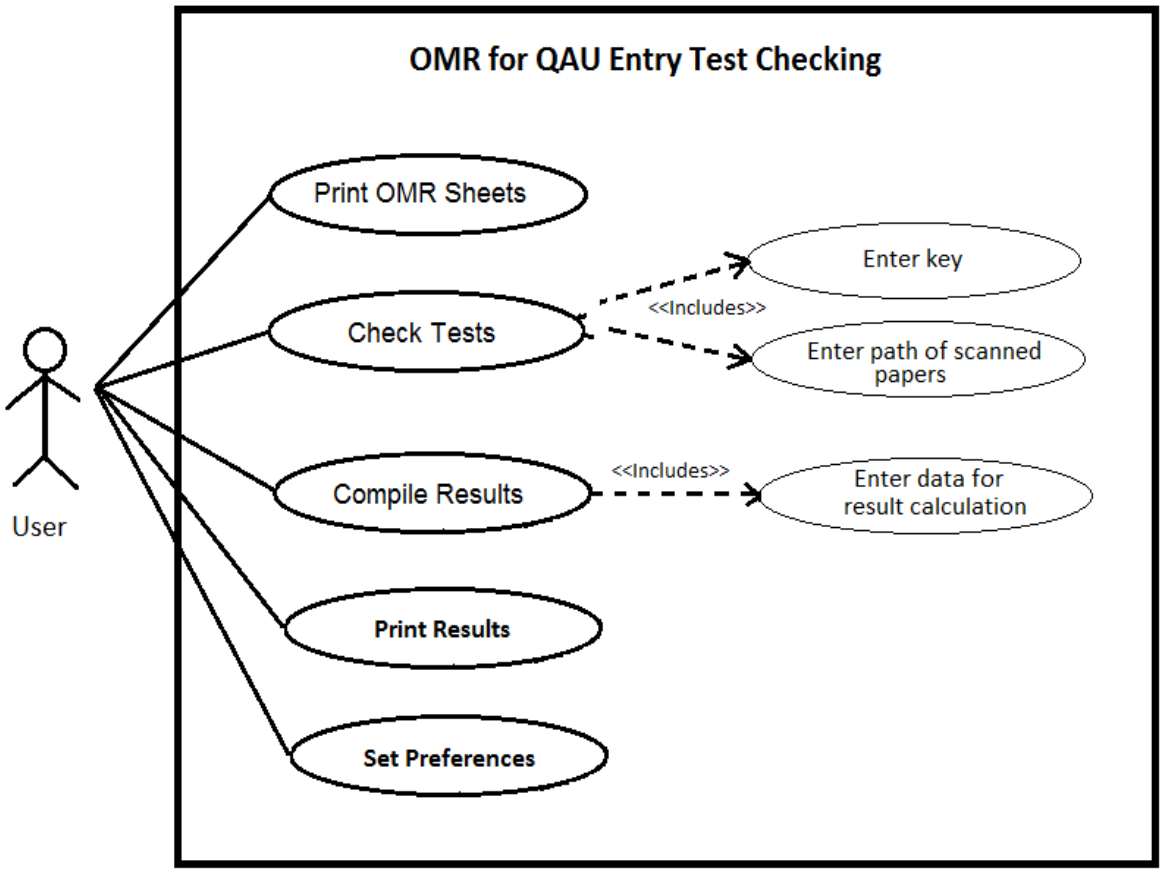
- User goals: The functions that user wants the system to perform.
- Sub functions: these are low level functions that the system must perform to achieve user's goals.

Use cases are written against user goals and functions against sub-functions.

USE CASE IDENTIFICATION:

Use Case ID	Use Case Name	Primary Actor
1	Print OMR Sheets	User
2	Check Tests	User
3	Compile Results	User
4	Print Results	User
5	Set Preferences	User

USE CASE DIAGRAM:



USE CASE DESCRIPTION:

Use Case ID	1
Use Case Name	Print OMR sheets
Actors	User
Description	The system suggests a sheet depending upon the number of questions entered by the user. User can get that sheet printed by giving a print command.
Pre-conditions	Printer should be connected.
Trigger	Pressing the "Print" button.
Basic Flow	1. User selects the "Print OMR Sheets" option. 2. User enters the number of questions he/she wants to include in the test. 3. User select "preview" option. 4. System suggests a sheet based upon the number of questions entered by the user, and shows the print preview of that OMR sheet. 5. User presses the "Print button". 6. System Prints the sheets.
Post conditions	OMR sheets get printed.
Alternate Flow	If printer is not connected, the system will display an error message.

Use Case ID	2
Use Case Name	Check tests
Actors	User
Description	The user after entering the path of folder containing tests, format of scanned images and path of the key can initiate the process of checking.
Pre-conditions	Scanned images of papers to be checked should be present in a folder in one of the drives of the system. A file containing answer key to the questions should be present in one of the drives of the system.
Trigger	Pressing the "Start Checking" button.
Basic Flow	1. User selects the "Check Tests" option. 2. User enters the path of the folder containing scanned tests. 3. User enters the format of the scanned tests. 4. User enters the path of the file containing the key. 5. User presses the "start checking" button. 6. System checks the tests and saves the record.
Post conditions	Tests get checked and record is saved.
Alternate Flow	N/A.

Use Case ID	3
Use Case Name	Compile Results
Actors	User
Description	Once the papers get checked, user can select the option to compile results. After compilation of results user can save the results by pressing the "save" button.
Pre-conditions	Tests should be checked before user compilation of results.
Trigger	Pressing the "Compile" button.
Basic Flow	1. User selects the "Compile Results" option. 2. User enters points given for each correct answer. 3. User enters points deducted for a wrong answer 4. User enters the path of the file containing answer key. 5. User enters the percentage of points required by a student to pass the test. 6. User presses the "Compile" button. 7. System compiles the result and displays it.
Post conditions	Results get compiled and displayed to the user.
Extension conditions	User can save results by selecting "save results" option. User can get the result sheet printed by pressing "print" button. [Use case ID: 4]

Use Case ID	4
Use Case Name	Print Results
Actors	User
Description	After compilation of results user can print the result list.
Pre-conditions	Printer should be connected to the system.
Trigger	Selecting the “Print” option.
Basic Flow	1. User can view result list. 2. User can select to get the result sheet printed by pressing “print” button. 3. System prints results.
Post conditions	Result sheet gets printed.
Alternate Flow	N/A.

Use Case ID	5
Use Case Name	Set Preferences
Actors	User
Description	User can set default input values.
Pre-conditions	------
Trigger	Selecting the “Select Preferences” option.
Basic Flow	1. User sets default values. 2. User presses “save” button. 3. Default values are saved by the system.
Post conditions	System should display values saved by user in all input containers until and unless changed by the user.
Alternate Flow	N/A.

Sub functions:

1	Read Students Records
2	Enter Key
3	Classify
4	Save Records
5	Update Records

Function ID	1
Function Name	Read Students Records
Description	This function reads students records from a file and saves these records.
Pre-conditions	Records should be present in a file.
Post conditions	Records should be saved in database.
Input	Path of the file containing the records.
Source	Use case: 2
Output	------
Destination	Use case: 2

Function ID	2
Function Name	Enter Key
Description	This function reads correct answers to the questions (key) that are saved in a file.
Pre-conditions	Key should be present in a file.
Post conditions	Key should be saved.
Input	Path of the file containing the key.
Source	Use case: 2
Output	---------
Destination	Use case: 2

Function ID	3
Function Name	Classify
Description	This function classifies the answers marked by students into their respective classes.
Pre-conditions	Images of tests should be present in one of the directories of the user
Post conditions	Tests should be checked.
Input	Image of a test.
Source	Use case 2
Output	Confirmation message
Destination	Use case 2

Function ID	4
Function Name	Save Records.
Description	This function saves the records of the checked tests.
Pre-conditions	Tests should be checked.
Post conditions	Test Records should be saved.
Input	Semester code
Source	Use case 2
Output	Confirmation message.
Destination	Use case 2

Function ID	5
Function Name	Update Records.
Description	This function also saves the records.
Pre-conditions	Tests should be checked.
Post conditions	Test Records should be saved.
Input	Semester code
Source	Use Case 3
Output	Confirmation message.
Destination	Use case 3

3.2.3 User Characteristics

Users of the system are

-Faculty Members

The system assumes a basic understanding of graphical user interfaces. The intended user is comfortable with simple menus.

3.3 Specific Requirements:

User Interface

- The system runs in a window.
- The primary window contains direct manipulation menus with buttons for carrying out the required functions.
- Initially user is provided with two options.
  - User can print new OMR sheets.
  - Or he/she can check scanned tests.
- If user selects the “print OMR sheet” option the system should provide a print preview of the OMR sheet.
- If the user selects “check tests” option the system should provide the user with file browser dialogues so that user can easily browse the path of the folder containing scanned images and the path of the file containing key.
- There should be a push button to initiate the process of checking.
- After checking of tests the user should be allowed to enter data for the compilation of results.
- An overview of the result should be displayed which shows
  - Total Students appeared
  - No. of students passed
  - No. of students failed
  - Result Percentage
- User should be provided an option to save the results.
- User should be able to view the result statistics.
- There should be an option to print result sheet.

3.4 Hardware Interfaces

No hardware interfaces are specified beyond the availability of a display screen, storage medium, system memory and a printer for printing OMR sheets and compiled results.

3.6 Identification of Classes: (Abstract Level)

Following classes can be identified at this level.

- **GUI:** This is the interface class that has some buttons, tabs and textboxes as per the requirement of the user. Its attributes will be decided at the design phase.
- **Controller:** It is not a good practice to implement business logic in the GUI and give direct access to the data store in GUI. Therefore a class is needed where the actual control and access to different classes and data store exists.
- **OMR Module:** This module is responsible for the
  - Extraction of data from OMR sheets
  - Preprocessing of data and
  - Classification of digits of roll number.
  - Classification of answers marked by candidates.

Chapter: 4

System Design

4.1 Introduction:

Before describing the architecture design of the OMR it is important to know what software architecture is.

Software architecture is a structure that comprises

- Main structural elements of a software
- Visible properties of these elements and
- Interaction among these elements.

Rest of this chapter is divided into three sections.

In section 4.2 an overview of the system architecture design and the main components of the system are briefly discussed. Section 4.3 discusses each component and its properties and functions in detail. In section 4.4 of this chapter the interaction among the main components of the system is shown with the help of diagrams.

4.2 System Architecture:

Before describing the system architecture, we first have a look at the architecture diagram of the system (figure 4.1) showing main components of the system.

4.2.1 Architecture Diagram:

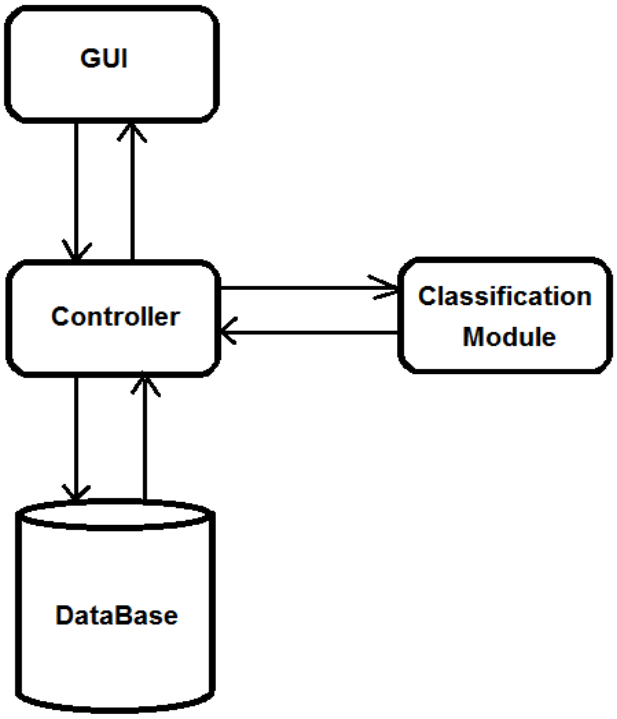


Figure 4.1

4.2.2 Description:

Architecture diagram of the system clearly shows that this system has a layered architecture.

Layered Architecture:

Layered architecture is a type of software architecture in which code is divided into logical layers defined by specific functions i.e. related functionalities within an application are grouped together into layers. The purpose of this logical division is to organize the code. It does not imply that these logical layers might run on different computers.

Layered architecture has at least three separate logical layers:
- Presentation Layer
- Business/Logic Layer
- Data Layer

Presentation layer is responsible for taking input from the user and displaying results, this layer cannot communicate directly with the data layer.
Logic Layer is the middle layer containing all the business logic and prevents direct access to the data. Communication between presentation and data layer occurs through this layer.
Data layer contains the data model of the system.

In our system GUI is the presentation layer.
Controller and classification module together form the logic layer.
The database is the data layer.

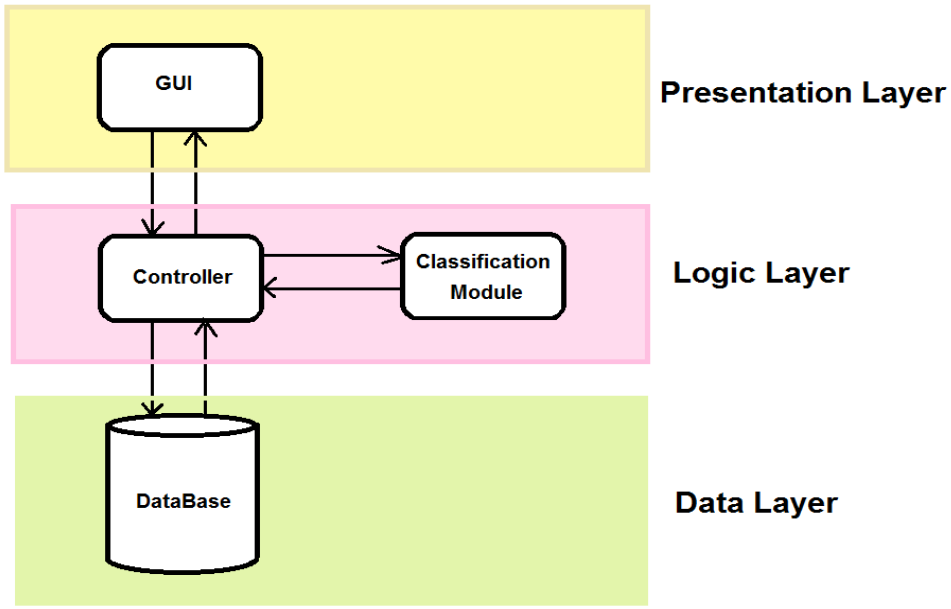


Figure4.2

4.3 System Components:

System is divided into four main components.

- GUI
- Controller
- Classification Module
- Database

4.3.1 The GUI:

GUI of the system needs to provide options for the following four main functionalities on the primary window.

a. Print OMR Sheets
b. Check Tests
c. Compile Results
d. Set Preferences

a. Print OMR Sheets:

If the user selects "Print OMR sheet" option he/she shall be provided with a textbox to enter number of questions the user wants to add in the paper.

User shall be able to see a print preview of the sheet suggested by the system.

User should be provided with a button to initiate the printing process.

b. Check Tests:

The window should contain 2 file browsers to help the user to browse and enter paths of:

- The folder containing scanned images of the tests to be evaluated.
- The file containing answer key to questions in the test to be evaluated.

Window should also contain a combo box to facilitate user to select format of the scanned images.

There should be a textbox to enter the number of questions in the paper. There should be a button to initiate process of checking.

c. Compile Results:

User would have to enter some data for the compilation of results, so he/she must be provided with four textboxes to enter

- Semester code.
- Points allocated for each correct answer.
- Points deducted for each incorrect answer.
- Percentage required by a candidate to pass the exam.

There should be two push buttons, one for generating "compile results" command and the other for saving results.

d. Set Preferences:

This tab provides the users with the facility to set the default values for all the above mentioned inputs. And there should be a button to save these default values.

4.3.2 Controller:

In order to separate the system's logic from the GUI a controller module is required that serves as a bridge between GUI and the Data thereby preventing direct access to the data and ensuring data abstraction.

This module contains all the functions responsible for the transfer of control to other modules and the functions that communicate with the database.

Data entered by the user will be passed to the controller from the GUI.

Controller module will pass control to the classification module and will wait for it to get done with the classification task.

After getting an "okay" signal from the classification module it shall read the answers returned by the classification module.

Controller will also read the answer key from the file present at the path specified by the user.

Match the answers marked by students to those in the answer key to determine whether a question is correct or incorrect and then save this information into the database.

4.3.3 Classification Module:

This module actually performs the task of classification.
This OMR has to read the following two things from the sheet:

- Registration number of a candidate
- Answers marked by the candidate

a. Reading Registration Number:

There are ten categories (0-9) for classification of registration number. Each digit comprising registration number needs to be classified into one of the ten categories.

b. Reading answers by a candidate:

Answers shall be classified into the following categories.

- Category **A:** Only option **A** is marked out of the five given options.
- Category **B:** Only option **B** is marked out of the five given options.
- Category **C:** Only option **C** is marked out of the five given options.
- Category **D:** Only option **D** is marked out of the five given options.
- Category **E:** Only option **E** is marked out of the five given options.
- Category **F:** An option not marked according to the given instructions or more than one options marked.
- Category **N:** No option selected for a given question.

Neural Networks:

Neural network classifiers shall be used for classification.

Two neural networks will be required, one for the classification of digits and the other for classification of answers.

Figure 4.3 shows the image of an answer marked by a candidate that neural network needs to classify.

Figure 4.3

Figure 4.4 shows the image of a digit that neural network has to classify.

Figure 4.4

Locating and Cropping of this input image from the sheet proves to be a challenge since we are not using any special purpose scanners to scan the sheets.

In the next section we will see how this data extraction is done in commercially available OMRs and then we will devise a solution keeping in mind our limited resources.

Extraction of data (that need to be classified) from the OMR sheet:

a. Conventional Process: (Based on OMR Scanning Machine)

- It uses high speed OMR Machine which has a built-in scanner and data processing software.

- OMR sheets used for exams are mostly designed on a standard a4 size paper.
- This a4 size OMR sheet is divided into 40 to 48 fixed sized channels depending upon the type of special purpose scanner (i.e. size of the read head of the scanner) used to scan the sheets.
- The read head of the scanner scans the sheet channel by channel thereby extracting the data.
- OMR sheets available in the market have special type of small size marks printed on them. These marks are used to identify OMR field in an OMR sheet.

b. Based on OMR Software (New Process):

Now the OMR technology is shifting from hardware based solution to software based solution. In this new type of solution regular simple Image scanner is used to scan OMR Sheets and later the processing is done by OMR Software.
This project is to build this type of OMR solution that uses regular scanner and ordinary printing paper for OMR sheets

Issues:
This software based technology is not very common and a variety of designs of OMR sheets available in the market are designed for hardware based OMR solutions.
Since these OMR sheets are not designed for software based OMR solution, therefore making a software based solution work with all the designs of those sheets which are designed in accordance with hardware based OMR solution is a big problem.

Solution: 1 Devise a solution which works with one or two of the available designs of the OMR sheets.
Problem: This solution is not practical because every time we conduct an exam we would have to go to the market to get that specific design of the sheet for which our OMR solution works.

Solution:2 A better solution, however, is to design a sheet keeping in mind our limited resources and providing the user with the facility to get these OMR sheets printed using our software product.
Advantage: Advantage of this solution is that besides solving the problem of data extraction it also provides the user with a comprehensive package and the user will not have to go anywhere for getting the OMR sheets.

OMR Sheet Design:

Following points were considered at the time of designing of OMR sheet for our software.

i. **Form size:** Single or double sided A4 sized sheets are considered enough for capturing exam data, therefore we are also using A4 size sheets for printing OMR sheets.

ii. **Orientation:** In about 90% of the cases OMR sheets are designed in landscape mode so we also selected landscape mode for the OMR sheet design.

iii. **Dropout Colours:** In order to capture data in blue or black ink, dropout colours (e.g. rose 30, harvest 24, midnight 83, teal 90 etc.) are used in printing of OMR sheets.

But this is not a constraint for out OMR solution. This OMR solution can work well with black and white sheets therefore dropout colours are not used and simple black and white OMR sheets are designed.

iv. **Timing tracks:** Conventionally, black coloured marks called timing tracks are used for tracking of OMR Field in an OMR sheet, In our design, we are using margins to track OMR field.

v. **Channels:** An A4 size paper is divided into 40 to 48 channels depending upon the number of heads in an Optical Mark Reader, where each head scans only one channel in a sheet to capture data.

Since we are not using such sophisticated hardware, therefore we apply a technique called template matching to find co-ordinates of the area from where we can start extracting data to feed to the neural network.

vi. **Bubbles:** Bubbles are the most common way of getting someone make a mark on the sheet to signify one's response to a question that is why our OMR sheet is also designed with bubbles.

vii. **Blocks:** OMR form is divided into block to capture different type of information. For instance there is an ID block to capture registration number, Name Block to record name and the most important of all is the answers block used capture response of a candidate.

In our sheet design there are only two block, the ID block and the question/answer block.

viii. **Logos:** often the organizations using OMR sheets ask to print their logo on the sheet. The organization for which this OMR solution has been designed is Quaid iAzam University, therefore the Header of the sheet is designed to has logo of the university.

4.3.4 Database Design:

This system maintains record of candidates appearing for the entrance tests, information about the entrance tests and result of every student, therefore a database needs to be maintained.

The database shall contain four tables.

a. Students
b. Test_info
c. Answers
d. Result

a. Students Table:

This table contains information of the candidates appearing for the tests. Attributes of this table and their description is explained in the following table.

Attributes	Description
Semester	Semester code of the semester in which a candidate is registering.
Reg_no	Registration number assigned to a candidate at the time of submission of admission form.
Name	Name of a candidate appearing for test.
FName	Father's Name of the candidate.
Domicile	Domicile information of the candidate.

b. Test_Info Table:

This table contains important information regarding an entrance test.

Attributes	Description
Semester	Semester code of the semester in which the test was conducted.
Std_appeared	Number of students appeared in the test.
Total_Qs	Total number of questions in the entrance test.
M_correct	Points granted for each correct answer.
D_incorrect	Points deducted for each incorrect answer.
Pass_Percentage	Percentage required by a user to pass the test.

c. Answers Table:

This table contains the record of every question marked by a student.

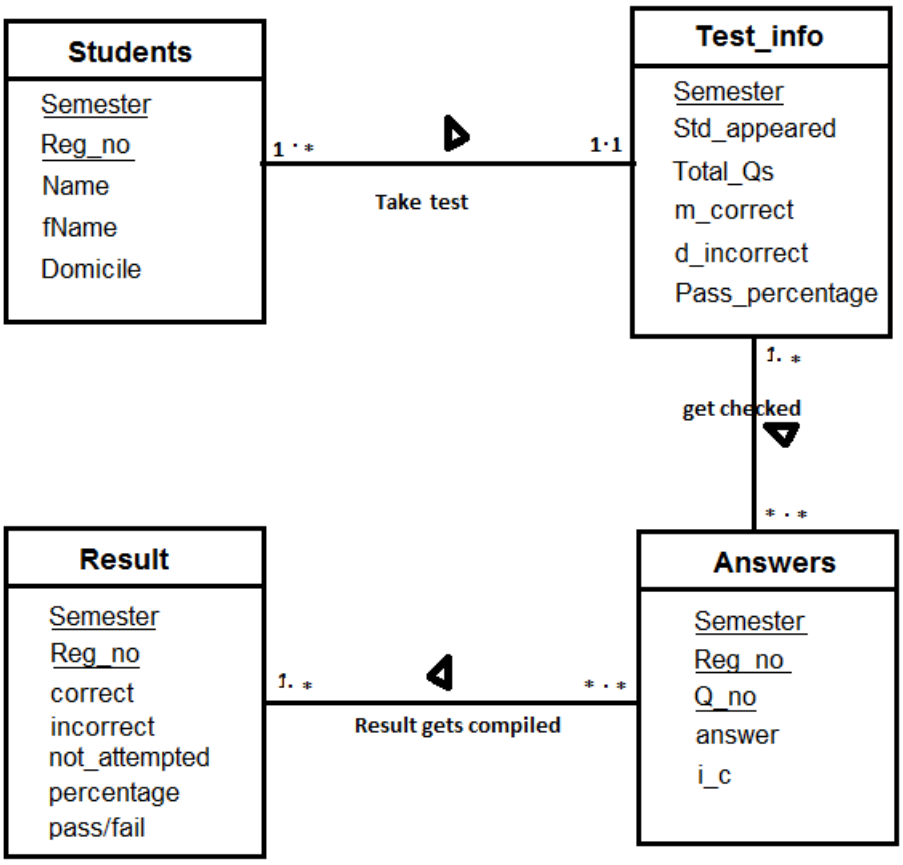
Attributes	Description
Semester	Semester code
Reg_no	Registration number of the candidate.
Q_no	Question number
Answer	The answer selected by the user
Correct/incorrect	Stores whether the answer is correct, incorrect or not attempted.

d. Results Table:

The aim to build this table is to store the number of correct answers, number of incorrect answers, and no of not attempted questions of each candidate.

Attributes	Description
semester	Semester code
Reg_no	Registration number of the candidate
Correct	Number of correct answers by the candidate
Incorrect	Number of incorrect answers by the candidate
Not_attempted	Number of not attempted questions
Percentage	Percentage of the candidate
Pass_fail	Candidate is pass or fail

4.3.5 Entity Relationship Diagram:



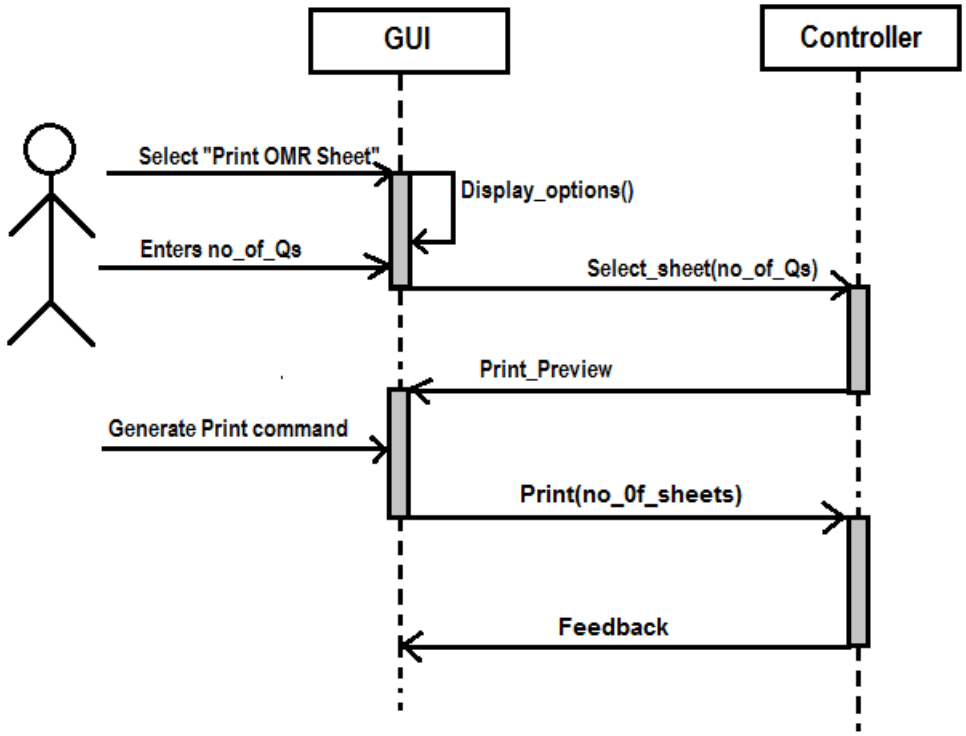
4.4 Interaction Among the components:

Interaction among components of the system is showed with the help of sequence diagrams. In sequence diagrams, usually, the interaction among ***object*** of a system are shown but the interaction shown in the following diagrams is among all the ***components*** of the system not just the objects.

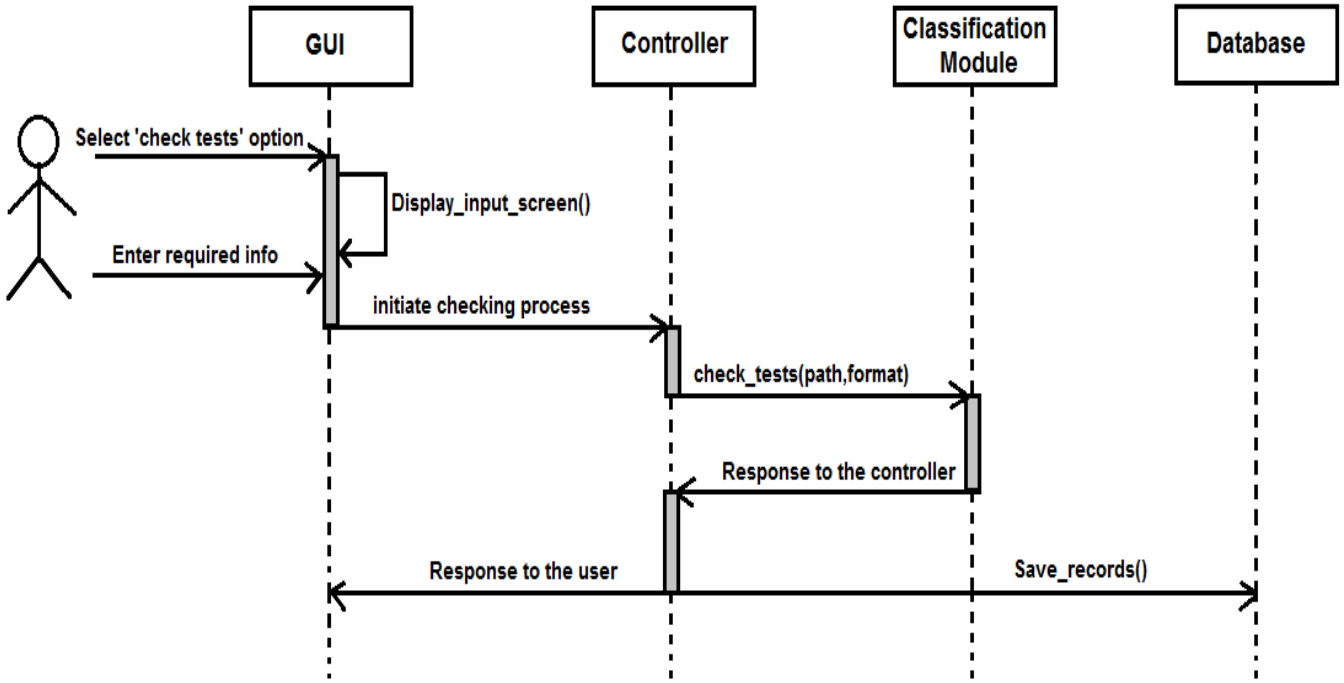
The main purpose of a sequence diagram is to define event sequences that result in some desired outcome. The focus is less on messages and more on the order in which messages occur.

4.4.1 Sequence Diagrams/Interaction Diagrams:

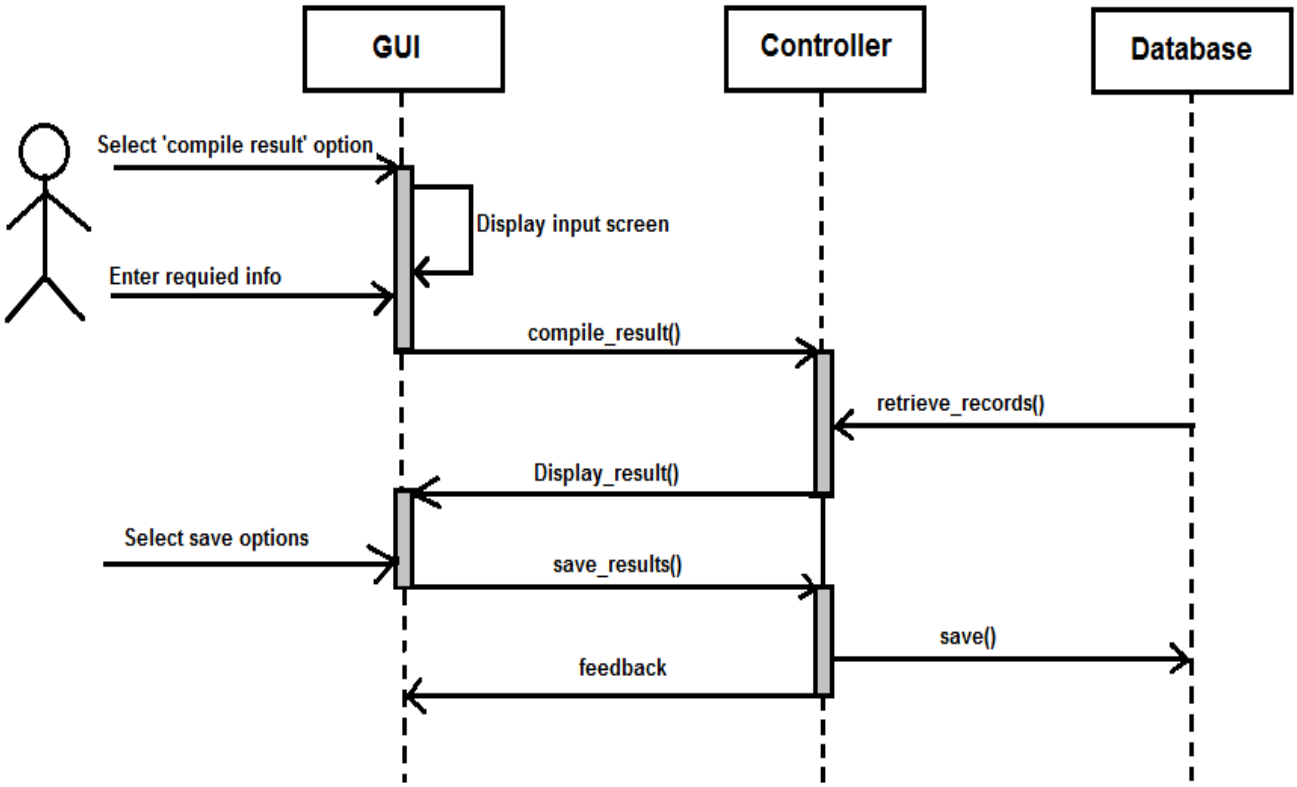
Sequence Diagram "Print OMR Sheets"



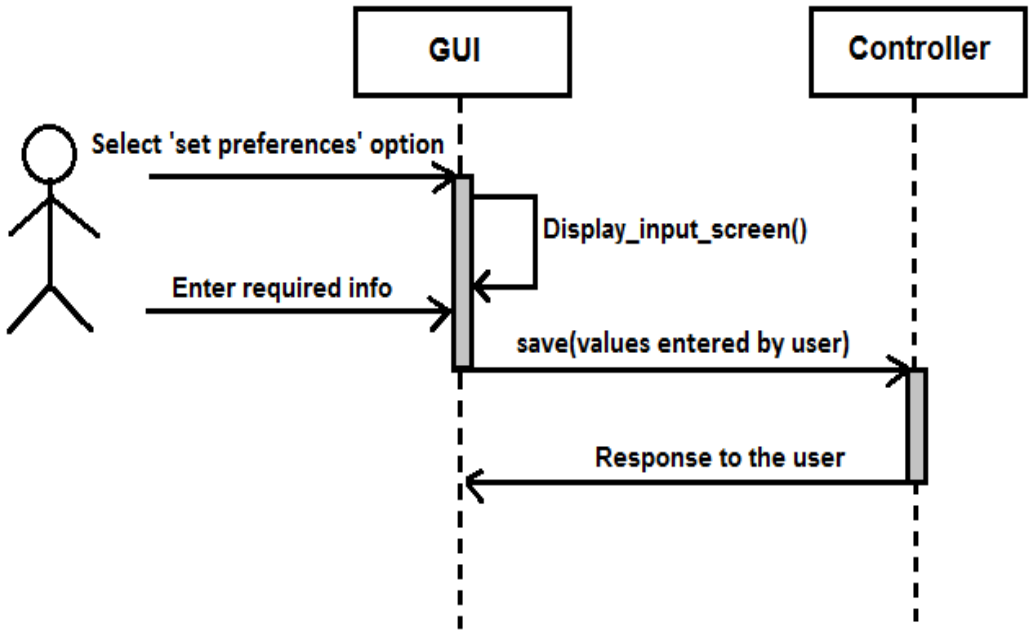
Sequence Diagram "Check Tests".



Sequence Diagram "Compile Results":



Sequence diagram: Set Preferences.



Chapter 5:

System Implementation

5.1 Platforms:

Database: MySQL Version 5.1.30 in the package of WAMP 2.0.

GUI: GUI has been built in Visual C# 2010.

Classification Module: MATLAB 7.10 (R2010a)

5.2 Implementation Description:

This section contains brief description of implementation of different components of the system.

As discussed in chapter 4, the four main components of the system are:

- Controller
- Classification Module
- GUI
- Database

The implementation of each of these components is described in detail.

5.2.1 Controller:

Controller is implemented in Visual C#.

Description:

This module contains all the functions responsible for the transfer of control to other modules and the functions that communicate with the database.

Input/output:

Controller is the central component of the system. Flow of control as well as information among other components is regulated by the controller. This component takes its input from GUI and classification module and its output goes to GUI, classification module and Database.

Functions:

Public functions of this component are:

- **Print_OMR_sheets(number of questions):**

  This method takes number of questions as argument and

  displays a print preview to the user when user envokes print command the OMR sheets get printed

- **Call_classification_module(address of scanned images, format of images):**

This method of controller takes two parameters as an input from the user through GUI and passes these parameters to the MATLAB module that is responsible for classification.

Output of this function is the result returned by classification module i.e. registration numbers and answers recognized by the neural network.

- **Read_key(address):**

Read_key(address) is a method of controller that reads a Microsoft Excel file which contains answer key to the questions in the tests. This method takes address of file to be read as a parameter.

- **Compare ():**

This method compares answer key to the answers marked by the students to determine whether an answer is correct or not. This method also counts the total number of correct answers, total number of incorrect answers, the number of questions not attempted and the number of questions not marked correctly by each candidate.

- **Read_Students_data(address):**

This is a method of controller that reads a Microsoft Excel file which contains students' data. This method takes address of file as a parameter.

- **Save_Results1():**

This function performs following tasks:

- establishes a connection with the database
- populates the Answers table in the database with the values calculated in compare() function.
- Closes the connection.

- **Compile_results(m_correct, d_incorrect, p_percent):**

This function takes three arguments:

m_correct: points allocated for each correct answer.

d_incorrect: points deducted for each incorrect answer.

p_percent: percentage of marks required by a candidate to pass the test.

This function fetches data from the 'Answers' table and calculates the percentage of a candidate and determines whether the candidate passed the exam or not.

- **Save_Results2():**

This function is responsible for

- Setting up a connection with the database
- populating 'Results' table in the database

  values used to populate the 'Results' table are the students record read by 'read_students-record function' and percentage and pass-fail status of a candidate determined in the compile_resuls table.
- Terminating the connection.

- **Save_test_info(semester):**

This method is used to populate 'test_info' table in the database.

This method takes semester code as a parameter and is responsible for setting up a connection with the MySQL database, saving information about the test and closing the connection.

- **View Results():**

This method displays a list of all the candidates appearing for the test.

This list shows following information of each candidate.

- Semester
- Registration number
- Name
- Father's Name
- Domicile
- Pass/Fail Status

- **Print_results():**

This method prints the result sheet on users demand.

5.2.1 Classification Module:

Description:

Classification module is built in MATLAB.

The main function of this module is to classify the data, using neural network classifier, into predefined categories.

These categories are discussed in detail in section 4.3.3.

The data that needs to be classified is shown in the figure below.

figure 5.1

Image of answer to be classified

figure 5.2

Image of registration number to be classified.

Figure 5.1 and 5.2 show that the data (answer marked by a student) which needs to be classified is a very small portion of the scanned image of the test.

So before performing classification we have to extract this data from the OMR sheet.

In order to crop this portion of image we need to locate the co-ordinates for cropping. Scanner used to scan the sheets is not an ADF (Auto Document Feeder) scanner therefore co-ordinates are different in every scanned image. Moreover, scanned images can have a lot of other discrepancies, for instance the possibility of tilted images is very high.

So before cropping the image, tilt correction must be done.

Input/output:

- Path of the folder containing scanned images of the tests.
- Format of the scanned images.
- No of questions.

Methods in the module:

This module has to perform the following tasks

- **Tilt correction:**

  The scanned image of an OMR sheet scanned by an ordinary scanner is tilted most of the times. This method measures the angle between border of the sheet and first line (of pixels) of the image, if angle is greater than zero, sheet is given positive or negative rotation accordingly to fix the tilt.

- **Crop OMR field:**

  OMR field is a specific area on the OMR sheet where user has to enter his/her response. To reduce the size of the image and to speed up the process of location of co-ordinates OMR field is extracted.

- **Template matching for co-ordinate location:**

  In order to crop an object, some co-ordinates (pixel locations) are needed. These co-ordinates are located by using a technique called 'template matching'. The system performs this template matching to find the location to start the extraction of classification data.

- **Crop answer grid and registration number grid:**

  Location to start extraction of data is determined by template matching and then data is extracted and processed before giving as an input to the neural network.

- **Classification:**

  For the classification of answers and registration number artificial neural network classifiers are used.

  Two neural networks are used, one for registration number and other for classifying answers.

  Type of Neural Networks used is Multi-layer Feed forward perceptron networks and backpropagation method is used to train the networks.

Number of neurons in the input layer is equal to the size of the image to be classified, shown in figure 5.01.

Number of outputs is equal to the number of categories in which we want to classify an image.

The number of output lines for the neural network used for reading registration number is ten because we are classifying each digit into one of the ten categories discussed in chapter 4.

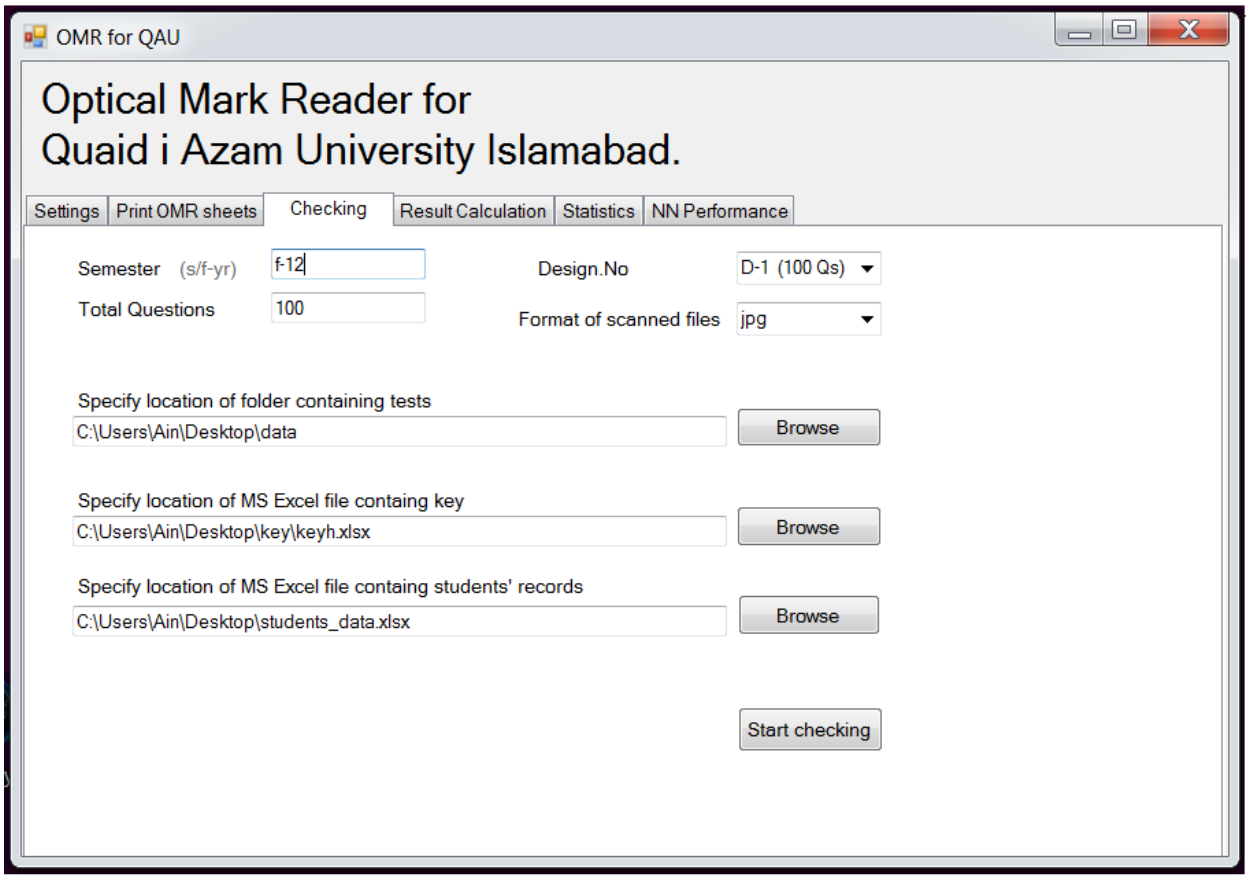
Neural network used for answers is having seven output lines.

- **Training:** Training of the neural networks has not yet been completed because the size of training data is kept open and user has been provided the facility to train the network to improve its performance.

  40 images of each category are used for training of the networks.

5.2.3 GUI:

A snapshot of GUI:



5.2.4 Database:

The database has been implemented using MySQL version 5.1.30.

Database needs to maintain the record of the:

- Test conducted
- Candidates appearing in the test
- Answers marked by candidates.
- Results of candidates.

Chapter 6:

System Testing

6.1 Introduction:

Software testing is an important element of software quality assurance. If done carefully and successfully, it can uncover many errors in software made during design and implementation phase.

6.2 Testing Strategies Used:

I have applied unit and integration testing. The overall number of test cases is enormous and cannot be included in this document, so I have selected a small sample from these test cases to show how the testing process is carried out.

6.2.1 Unit Testing: Unit testing focuses on verification of individual units of software design. A unit is a smallest testable part of an application.

6.2.2 Integration Testing: Integration testing is done to check connectivity of components. Integration testing identifies the problems that occur when the individual units are combined i.e it detects the problem in interface of the two units. Integration testing is done after unit testing.

6.2.3 Testing Neural Networks:

For evaluation of neural network's training state and performance, training and validation datasets are distributed for different experiments. K-fold or Jackknife mechanism is used for training and validation. Jackknife method is used with 4 folds, where k = 4. That's why original data set is organized in 4 combinations to evaluate the neural networks thoroughly. Dataset organization for 4 experiments is explained in following table.

Dataset organization: (Answers identification):

Data set	Total no. of instances of category							Training Instances (each category)	Validation Instances
	A	B	C	D	E	F	N		
DS 1	100	100	100	100	100	100	100	26-100	1-25
DS 2	100	100	100	100	100	100	100	1-25, 51-100	26-50
DS 3	100	100	100	100	100	100	100	1-50, 76-100	51-75
DS 4	100	100	100	100	100	100	100	1-75	76-100

Experimental results of neural network for classification of answers

Category A

Experiment	Total test instances	Mis-classified	Correct Classified	Sample Error	Average Error
1	25	13	12	0.52	0.44
2	25	10	15	0.4	
3	25	12	13	0.48	
4	25	9	16	0.36	

Category B

Experiment	Total test instances	Mis-classified	Correct Classified	Sample Error	Average Error
1	25	10	15	0.6	0.415
2	25	12	13	0.52	
3	25	9	16	0.36	
4	25	7	18	0.38	

Category C

Experiment	Total test instances	Mis-classified	Correct Classified	Sample Error	Average Error
1	25	10	15	0.4	0.35
2	25	6	19	0.31	
3	25	6	19	0.31	
4	25	7	18	0.38	

Category D

Experiment	Total test instances	Mis-classified	Correct Classified	Sample Error	Average Error
1	25	15	10	0.6	0.5
2	25	14	11	0.56	
3	25	11	14	0.44	
4	25	10	15	0.4	

Category E

Experiment	Total test instances	Mis-classified	Correct Classified	Sample Error	Average Error
1	25	15	10	0.6	0.54
2	25	14	11	0.56	
3	25	15	10	0.6	
4	25	10	15	0.4	

Category N

Experiment	Total test instances	Mis-classified	Correct Classified	Sample Error	Average Error
1	25	15	10	0.6	0.52
2	25	13	12	0.52	
3	25	13	12	0.52	
4	25	11	14	0.44	

Category F

Experiment	Total test instances	Mis-classified	Correct Classified	Sample Error	Average Error
1	25	10	15	0.4	0.49
2	25	14	11	0.56	
3	25	10	15	0.6	
4	25	10	15	0.4	

Unit testing:

Functionalities to be tested:

F1: Selection of appropriate OMR sheet design based on number of questions entered by the user.
F2: Correctly read students data from Excel file.

Functionality F1: Selection of appropriate OMR sheet design based on number of questions entered by the user.

Components: GUI and Controller

Test Cases:

T1: Input valid number of questions (1 - 150)
T2: Input invalid number of questions (0)
T3: Input invalid number of questions (151)

Test Case Number	T1
Input	Number of questions entered = 75
Expected output	Preview of a sheet consisting of 100 questions.
Actual Output	Preview of a sheet consisting of 100 questions.
Test Conducted by	Noorulain
Date of test	11th July, 2012.

Test Case Number	T2
Input	Number of questions entered = 0
Expected output	Display message "Please enter a number between 1 and 150."
Actual Output	Message displayed "Please enter a number between 1 and 150."
Test Conducted by	Noorulain
Date of test	11th July, 2012.

Test Case Number	T3
Input	Number of questions entered = 151
Expected output	Display message "Please enter a number between 1 and 150."
Actual Output	Message displayed "Please enter a number between 1 and 150."
Test Conducted by	Noorulain
Date of test	11th July, 2012.

Functionality F2: Correctly read students data from an Excel file.

Component: Controller

Description: We need to check whether the system reads students data from the file (the address of which is given). To check this we display this data on screen to check whether it was correctly read or not.

Test Case Number	T4
Input	"C:\key.xls" (address of file)
Expected output	Display correct data read from file.
Actual Output	Displayed correct data read from file.
Test Conducted by	Noorulain
Date of test	11th July, 2012.

Integration testing:

Functionalities to be tested:

F3: Classification module takes parameters in correct format and returns the results after classification.

Description: : We need to check whether the classification module when integrated to the controller works fine or not.

Components: classification module and controller.

Test Case Number	T5
Input	Path of folder containing tests, format of images.
Expected output	Answers classified by the classification module.
Actual Output	Answers classified by the classification module.
Test Conducted by	Noorulain
Date of test	11th July, 2012.

F4: Correctly saves records in database.

Description: We need to check whether the database gets populated correctly or not after the tests get checked.

Components: Controller module

Test Case Number	T6
Input	Data that we need to save
Expected output	The database would get updated with the new records added correctly.
Actual Output	The database got updated with the new records added correctly.
Test Conducted by	Noorulain
Date of test	11th July, 2012.

Chapter 7:

Future Enhancements.

For future, it is planned to add the following features to the system.

1.1 More OMR sheet designs:

In future more designs of OMR sheets can be introduced thereby providing the user more choice to select the sheet which the user thinks is the most appropriate for test he/she is going to conduct.

1.2 Evaluation of Quizzes:

The system has been designed to automate the checking of entrance tests conducted every semester but the system can be can be further extended to evaluate quizzes especially of courses for which a large number of students are registered.

1.3 Variable sized Registration Number:

I present solution registration number consisting of 6 digits has been used due to some constraints in OMR sheet designing, but in future variable sized registration number can be used, or constants can also be added to the registration number.

1.4 More Queries about the results:

The database is designed to keep record of:

- Every test that gets checked by the software.
- Every candidate appearing in the test.
- Every answer marked by a candidate.

So a lot of useful information can be extracted.
For instance:

- The user can compare results of tests conducted in different semesters.
- User can view how many candidates correctly answered a specific question.
- How many candidates did not answer a specific questions
- And a lot of such queries can be implemented.

References:

[1] K. Chinnasarn Y. Rangsanseri "*An image-processing oriented optical mark reader"* Applications of digital image processing XXII, Denver CO,1999.

[2] Jiawei Han and Micheline Kamber Harcourt "*Data Mining Concepts and Techniques".*

[3] Ian H. Witten, Mark A. Hall *"Data Mining, Practical Machine Learning tools and techniques."*

[4] http://msdn.microsoft.com/en-us/library/ee658098.aspx

[5] *International Journal of Information Technology and Knowledge Management*
July-December 2012.

[6] http://library.thinkquest.org/C007009/introduction/types/types.html

[7] Bass, Len, Paul Clements, and Rick Kazman. *Software Architecture in Practice, 2nd ed*. Addison-Wesley Professional, 2003.

[8] Murraydata OMR form Design.