PROJECT REPORT ON

COMPUTERIZATION OF THE EXAMINATION SECTION OF THE VJTI MAIN OFFICE

SUBMITTED BY

BADRISH CHANDRAMOULI
AMOL GHOTING
RAKESH BILANEY
MANISH CHABLANI

GUIDE
DR. R. D. DARUWALA

DEPARTMENT OF COMPUTER TECHNOLOGY

V.J.T.I., MUMBAI.
1999-2000

A

PROJECT REPORT ON

COMPUTERIZATION OF THE EXAMINATION SECTION OF THE VJTI MAIN OFFICE

Developed by

Badrish Chandramouli	GH - 96413
Amol Ghoting	GH - 96425
Rakesh Bilaney	GH - 96409
Manish Chablani	GH – 96412

Under the guidance of
Dr. R. D. Daruwala

Submitted in partial fulfillment of the requirements of PROJECT II for the award of the degree of Bachelor of Engineering in Computer Engineering

To
University of Mumbai
Through
VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE

In May 2000

Certificate

This is to certify that the project titled '*Computerization of the Examination Section of the Main office of VJTI'* is a bonafide record of the project work done by

Badrish Chandramouli	GH - 96413
Amol Ghoting	GH - 96425
Rakesh Bilaney	GH - 96409
Manish Chablani	GH - 96412

during the year 1999-2000 under the guidance of Dr. R. D. Daruwala, VJTI, University of Mumbai

Head,
Department of
Computer Technology

Guide

Certificate

This is to certify that the project titled '*Computerization of the Examination Section of the Main office of VJTI*' is a bonafide record of the project work done by

Badrish Chandramouli	GH - 96413
Amol Ghoting	GH - 96425
Rakesh Bilaney	GH - 96409
Manish Chablani	GH - 96412

during the year 1999-2000 and they have satisfactorily completed the requirements of PROJECT II as prescribed by the University of Mumbai

Examiner	Examiner
(Internal)	(External)

Acknowledgements

We take this opportunity to express our sincere gratitude towards our guide Dr. R. D. Daruwala for his guidance and whole hearted support towards the completion of our project ‘Computerization of the Examination Section of the VJTI Main office’. The trust he placed in us by giving us round the clock access to the excellent facilities of the Microprocessors Laboratory gave us further encouragement.

We owe a debt of gratitude to Mr. Kulkarni, the Exam Section head, for entertaining our queries at all times of the day.

We thank our Head of Department for the support.

Badrish Chandramouli
Amol Ghoting
Rakesh Bilaney
Manish Chablani

Table of contents

CHAPTER 1
INTRODUCTION

A system is an organized set of functional units or components. VJTI, by this definition is a system with various objectives. It consists of inter-related subsystems such as Accounts, Office, Stores and other various branches. None of these sub-systems can perform independent of the others. This means that proper co-ordination is required between the various departments for the smooth and effective functioning of VJTI as a whole.

The Examination section of VJTI plays in important role of maintaining all student records as well as conducting and managing the Examinations. It needs to allot student seat numbers, send their answer sheets for correction, allot internal and external examiners, process the marks obtained and provide the students with the results and mark-sheets.

Thus in a nutshell, the examination section is a very important part of the institute and its efficient functioning is a must for the smooth working of the institute.

CHAPTER 2
ANALYSIS OF THE SYSTEM

2.1. The problem and its statement in detail

The aim of the project is to analyze the Examination Section within the VJTI Main Office and to design an automated system to improve the same. The first task towards this goal involved a thorough understanding of the existing system, in order to accurately project the system requirements. For this purpose, Structured System Analysis and Design theory was employed in order to streamline the entire task and to possibly avoid errors in understanding the existing system.

Initially, preliminary investigations were undertaken as follows:

a. Request Clarification
The current system of operation was thoroughly scanned by talking to and interviewing the employees and examining the method of operation. The exact request was properly understood in order to ensure that the system would ultimately satisfy the necessity.

b. Feasibility Study
The task of determining whether the system requested is feasible or not, was next considered. We considered three aspects of feasibility:

i. Technical Feasibility
Currently, the Examinations Section does not utilize computers except for clerical purposes. The work performed by the examination section can be summarized as follows:
- Maintaining a master ledger in order to ensure that every student who is appearing for an exam is eligible to do so.
- To allot exam seat numbers and exam halls to students.

- To maintain a file of student seat numbers and subjects
- To verify the number of papers received from the exam supervisors.
- To appoint internal and external examiners and to store the answer sheets and dispatch them to professors for correction.
- To collect the corrected answer sheets, compile the results and allot grace marks as required.
- To generate the results and issue marksheets
- To take care of the payment of bills to the examiners for paper correction.

It was concluded that the maintenance of records would be greatly facilitated if computers were inducted into the system. So it is definitely technically feasible to computerize the Exam Section

ii) Economic Feasibility

Time is a major problem in the current system. Due the requirement of manual entries, the results are often delayed. This causes an imbalance in the entire examination system. Moreover, there are chances of errors creeping in due to manual data entry. If computers were introduced, there would be no such imbalances or errors. It was verified that the college has enough funds to sustain the implementation of the system after it is designed.

iii) Operational Feasibility

Due to a possible lack of computer skills among the employees, there is a requirement to impart extensive training to the personnel in order that they handle the transition to an automated system, gracefully.

Once the above is done, it is felt that the automation of the system is definitely operationally feasible.

c. Request Approval

Once the preliminary investigations are completed, the analysis was submitted to the project guide for approval. Changes suggested by the guide were taken into consideration and the process was repeated. Once the analysis is approved, its priority, completion time and personnel requirements will be estimated and finally the project can be scheduled to launch.

2.2. The existing manual system

The present system functions as follows:

1) **<u>Admission</u>**: When a student is admitted to the institute his record is inserted into the master ledger. Master ledger is a file, which contains record of all the students of the institute. Each record of the student has the following format:
   - i) Roll number
   - ii) Name of the student
   - iii) Name and address of Father/Guardian
   - iv) Age, date of birth
   - v) Previous college and university attended
   - vi) Last exam passed
   - vii) Place of permanent residence
   - viii) Date of admission
   - ix) Category (blind, etc.)
   - x) Class of year to which admitted
   - xi) Results for individual semesters(semester I,II,II,…,VIII)
   - xii) Transference certificate (if any)
   - xiii) Date of leaving
   - xiv) Reason of leaving
   - xv) Remarks

In this record we also specify direct admission of student to semesters III, V or VII

Updating of master ledger:

After any semester results are out, the master ledger is updated. The result of the student is reflected in his/her record in the master ledger.

2) **Eligibility criteria**:

Following are the eligibility criteria for taking any semester exams:

i. For odd semesters (III, V, VII) a maximum of four A.T.K.T's in the latest two semesters are permissible.

ii. For even semesters (IV, VI, VII) a maximum of six A.T.K.T's in the latest two semesters are permissible.

Only students who satisfy the above mentioned criteria are allowed to appear for the examinations. In case a student wishes to appear for the VII semester again to improve his grades, he can appear for theory and practicals only, the rest being carried over.

The allotment of the examination number is carried out after the eligible students pay the examination fees.

3) **Supervisors report**:

On the day of the examinations the supervisor submits the attendance record of the students. This is known as 'Supervisor's Report'. Its format is attached in the appendix.

This report is verified by the office staff and is stored for future reference.

4) **Appointment of Examiners**:

A table containing records of the internal and external examiners is maintained. Using this table, the person in charge of the examination section along with the H.O.D, appoints internal and external examiners for that department.

A preformatted letter is sent to the appointed examiners with a provision for declining the appointment. In case the examiner declines the appointment, another examiner is selected as above.

A sample of the preformatted letter sent to the examiner is attached in the appendix.

5) **Dispatch**:

The following matter is sent to the appointed examiner:

i) Covering letter

ii) Answer books

iii) A blank mark sheet

iv) Resolution form

v) Bill

The covering letter specifies subject, semester or course, date, and number of answer books.

The examiner will fill the blank mark sheet and the bill after correction of the answer papers.

A sample of the covering letter, mark sheet, resolution form and bill is attached in the appendix.

6) **Receiving the mark sheet**:

The examiners return the corrected answer books and the mark sheet. The office staff combines marks of the two sections. After receiving the marks of all the subjects of a semester, processing for grace marks is carried out.

Grace marks are calculated as follows:

i) For whole semester examination: If the student has 50% or more over all, a maximum of 10 marks are awarded as grace independent of the number of subjects.

ii) For part semester examination: A maximum of 2 marks grace per head of passing is awarded, but total grace should not exceed 5 marks.

On receiving the marks of all the subjects of a semester, the marks are printed and the master ledger is updated.

2.3. Disadvantages of the existing system and proposals to overcome them

After analyzing the current system we noted the following disadvantages:

i) Difficulty in updating the master ledger.

The mark sheets bear the examination number of the student whereas the master ledger is arranged as per the roll number of the student. Hence, the process of updating the master ledger is time-consuming and error prone.

In the proposed system, the master ledger as well as the results will be stored in electronic form and will be linked together by a table. Hence the process of updating the master ledger can be automated.

ii) Manual determination of the eligibility criteria.

The master file record of each student has to be checked manually for eligibility. This makes the process lengthy.

In the proposed system, the whole process can be automated.

iii) Manual calculation of grace marks

All the calculations for grace marks have to be performed manually for each student.

This can also be automated in the new system.

iv) No provision for determining the status

Difficulty in determining the status (whether the papers have been sent for correction, or the practical or viva are pending, etc) of a particular semester.

In the proposed system, provision can be made to provide the status. The staff should be able to get information about the pending items of every semester of each branch.

v) No supplementary information on examiner

No information regarding an examiner's past behavior is stored in the current system.
In the proposed system, extra information about the examiner's past record is maintained such as delay in correction, refusing an appointment, etc. This information can be used while appointing an examiner.

vi) No provision for automated reminders
In the current system sometimes the delay in correction of the papers goes unnoticed.
This can be prevented in the proposed system by generating an automatic reminder after a specified number of days.

vii) Delay in results
Due to the various drawbacks mentioned above there is a delay in the declaration of the results.
As the above drawbacks are eliminated in the proposed system the results can be expedited.

2.4. Requirement analysis

2.4.1. Fact finding techniques to collect data

1. Interview

We interviewed the head of Examination Section, and collected data about the functioning of the existing system. After collecting all the necessary information, several questions were raised to clear existing doubts. We have included some of the questions:

1. Explain the format of Master File.
2. Explain the format of letter.
3. Explain the format of database of internal and external examiners.

4. How is the allotment of examiners done?
5. Do you verify the attendance sheet before sending for corrections?
6. Do you maintain a record of which examiners, what subject, what date? If yes, how is it kept?
7. What if an appointed examiner refuses the work?
8. Do you want to send late reminder?
9. Does the examiner return the corrected papers only, or also the list of seat numbers and marks?
10. What are the rules for eligibility of a student to attend each examination?

The answers to these questions were resolved and have been included in the explanation of the existing system.

2. Record Review

The files maintained by the examination section were examined in order to understand the working of the manual system. The format of all the forms used in the system was analyzed. The forms have been attached in the synopsis.

2.4.2. Software and hardware requirements

System requirements

The following questions were asked to accurately project the system requirements:

1. What is being done?

The Examination section of VJTI performs the following tasks:

i) Maintaining academic record of every student.
ii) Allotment of exam numbers to eligible students.
iii) Appointment of examiners and distribution of the student answer sheets to examiners.

iv) Gathers the marks from the examiners and compiles them considering allotment of grace marks.
v) Displays the results.

2. How is it being done?
This information was collected from the concerned staff members, and the information is documented above.

How frequently is this done?
The above procedure is carried out 2 times a year.

Volume:
The Examination sections need to process results for all the branches and all years, about 2500 students. Each student on an average carries 6 subjects.

How well is it being done?
The examination section needs to process a large volume of results and hence it takes them a lot of time to release the results.
The consequence is that student's land up getting their results very late.
A computer is used in only one phase of the entire process. After the results are compiled manually, it is entered into the computer using Excel, and the printouts are taken.

What is the problem and how serious is this problem?
The disadvantages were described in the previous section.

Underlying cause: Manually processing the entries takes a lot of time and is error-prone due to the large volume of the papers.

Software requirements

- Operating System - Microsoft Windows 98
- Visual Basic Version 6.0
- Microsoft Access 97
- Database Engine – DAO 3.5

Hardware requirements

The Examination Section is currently in possession of a single computer, using which marks are manually entered using Excel 97.

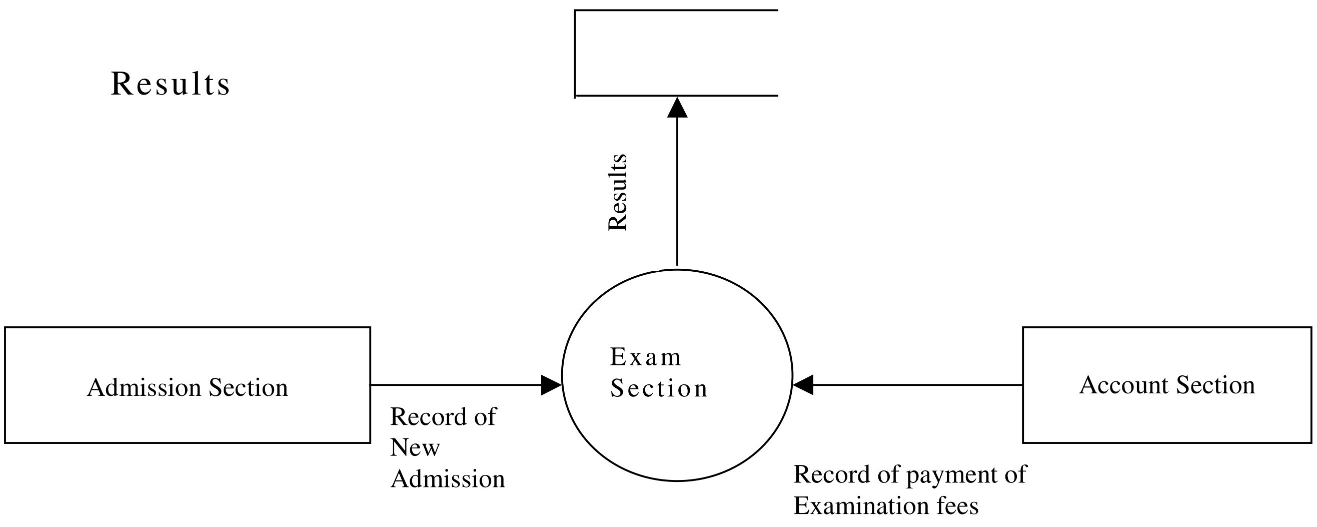
The additional hardware required for the implementation of the proposed system is estimated below.

Minimum Requirements	Recommended System
Pentium @100MHz	Pentium-II @266MHz (2)
32MB RAM	64MB RAM
4.1GB HDD	8 GB HDD
1.44MB FDD	1.44MB FDD
	Flatbed Scanner

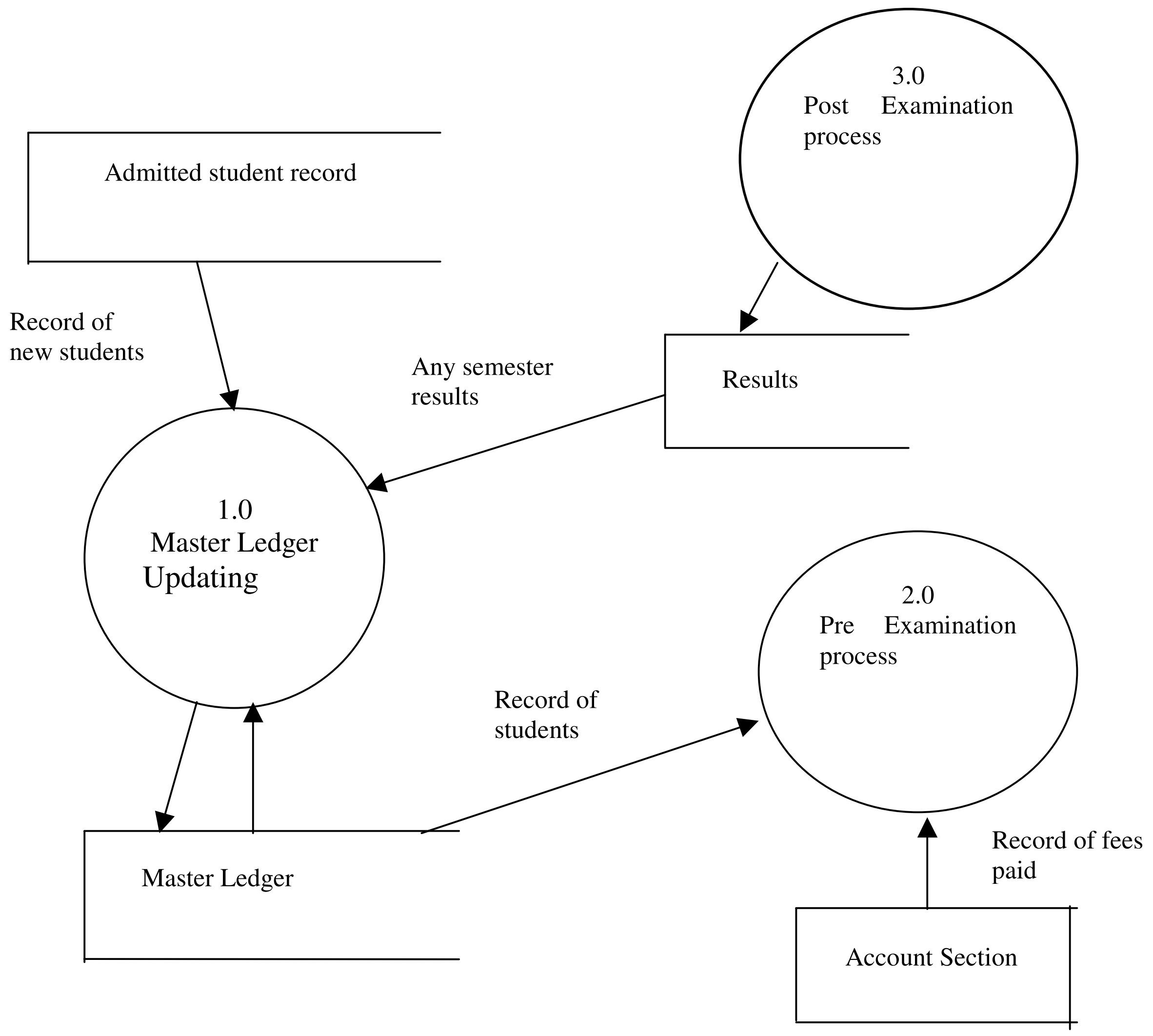
2.5. DFD and ER diagrams

2.5.1. Data Flow Diagrams

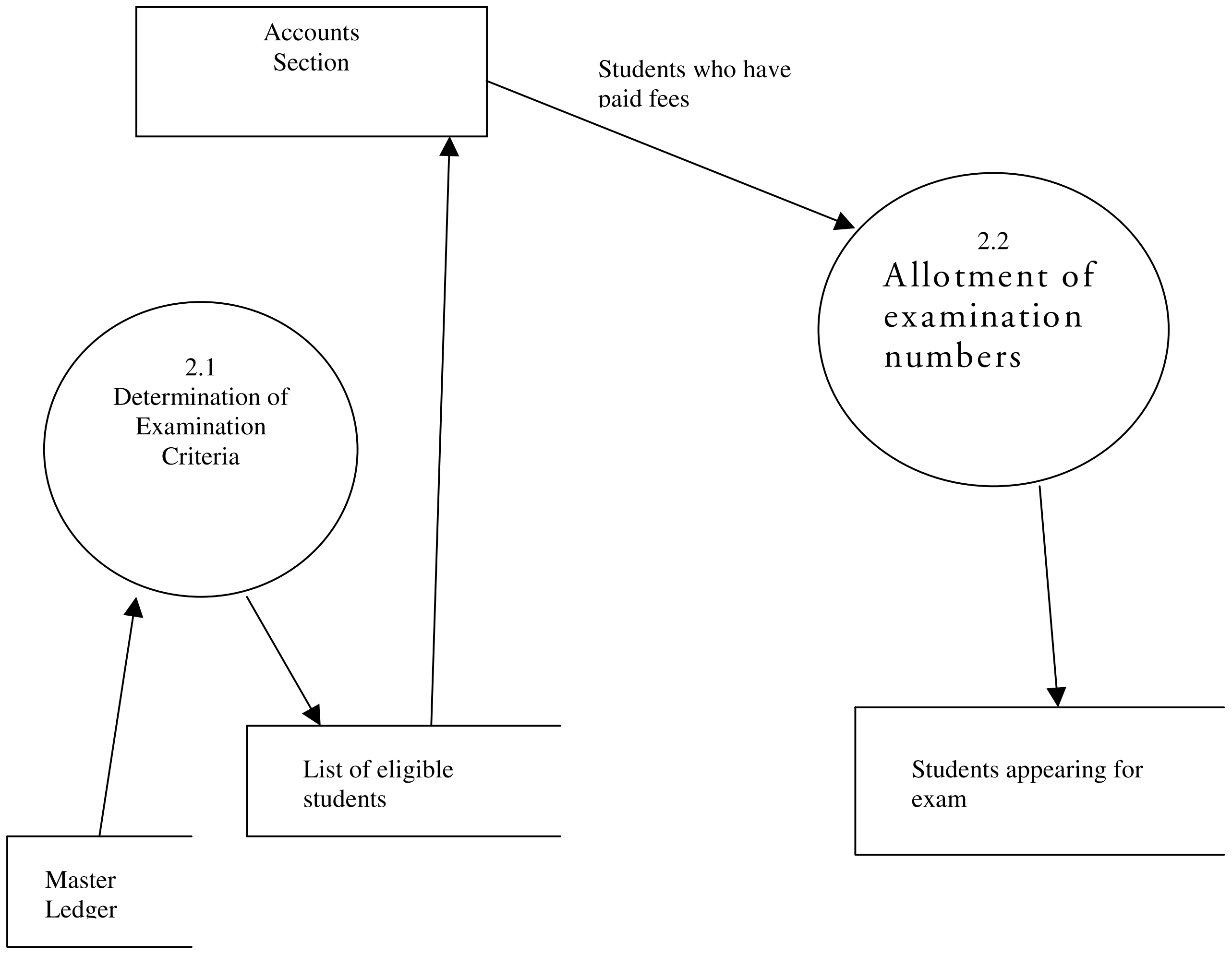
Level 0 DFD



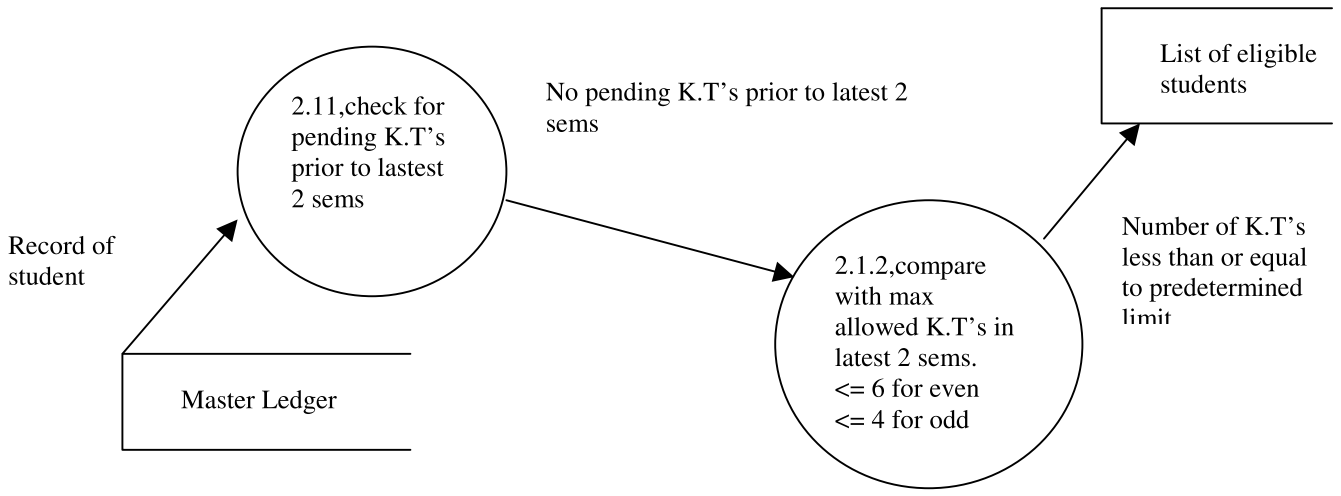
Level 1 DFD


3.0
Post Examination process
Admitted student record
Record of new students
Any semester results
Results
1.0
Master Ledger Updating
2.0
Pre Examination process
Record of students
Record of fees paid
Master Ledger
Account Section

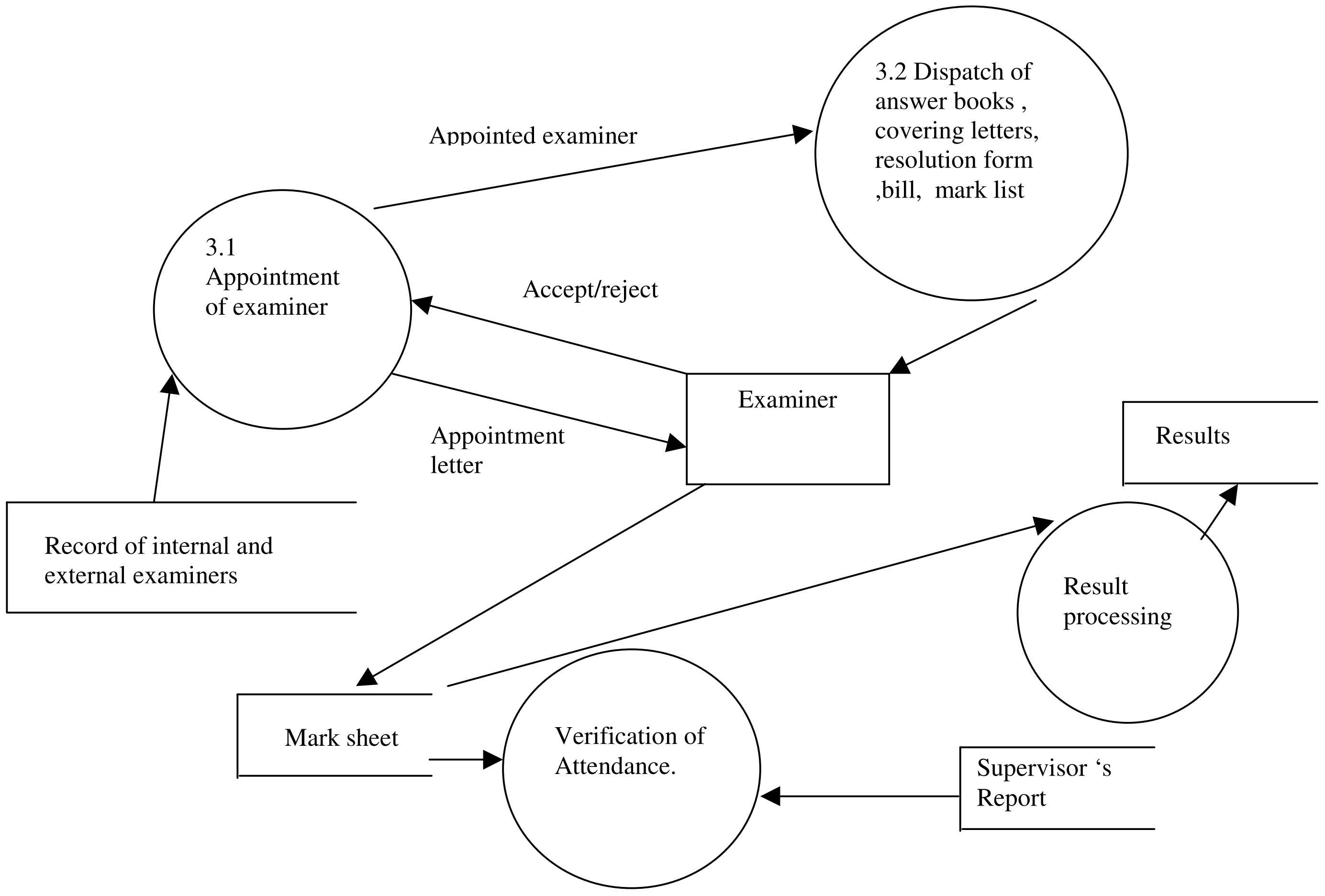
Level 2 DFD for pre Examination process


Accounts Section
Students who have paid fees
2.2 Allotment of examination numbers
2.1 Determination of Examination Criteria
List of eligible students
Students appearing for exam
Master Ledger

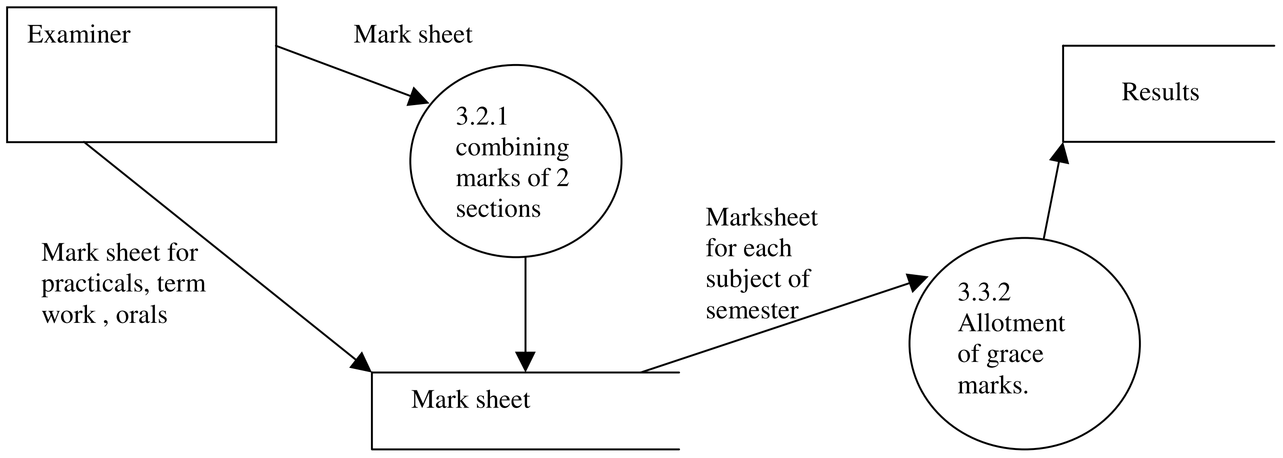
Level 3 DFD for pre Examination process


2.11,check for pending K.T's prior to lastest 2 sems
No pending K.T's prior to latest 2 sems
List of eligible students
Record of student
Number of K.T's less than or equal to predetermined limit
2.1.2,compare with max allowed K.T's in latest 2 sems. <= 6 for even <= 4 for odd
Master Ledger

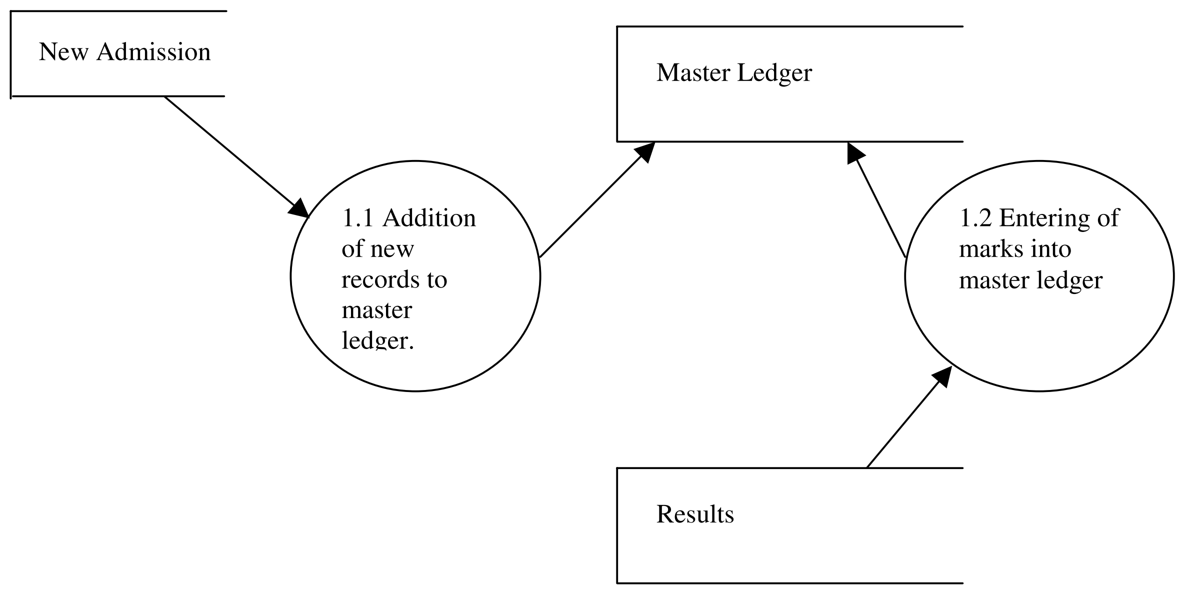
Level 2 DFD for post examination section.


3.2 Dispatch of answer books , covering letters, resolution form ,bill, mark list
Appointed examiner
3.1 Appointment of examiner
Accept/reject
Examiner
Appointment letter
Results
Record of internal and external examiners
Result processing
Mark sheet
Verification of Attendance.
Supervisor 's Report

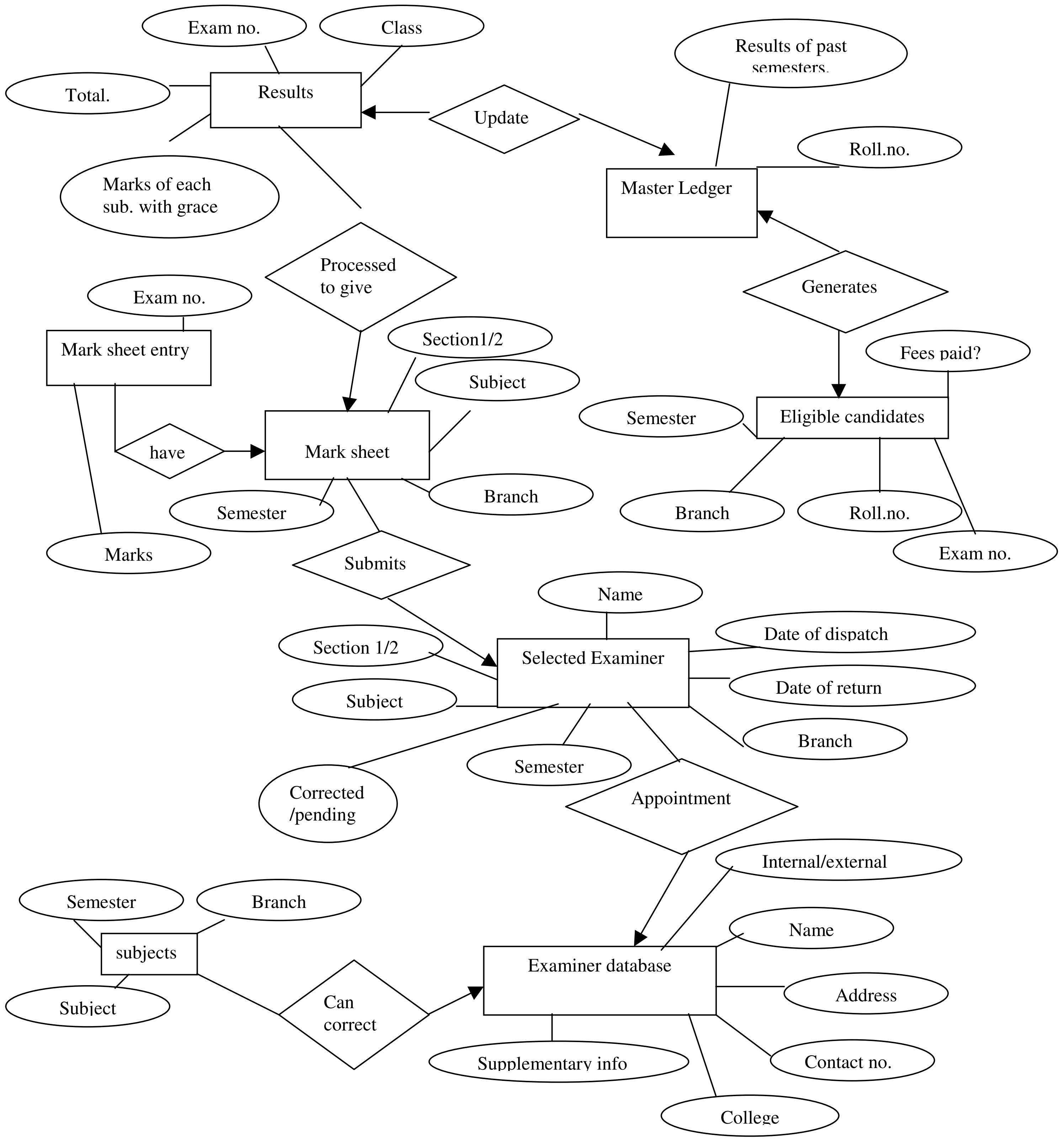
Level 3 DFD for result processing


Examiner
Mark sheet
3.2.1 combining marks of 2 sections
Results
Marksheet for each subject of semester
Mark sheet for practicals, term work , orals
3.3.2 Allotment of grace marks.
Mark sheet

Level 2 DFD for updating Master Ledger


New Admission
Master Ledger
1.1 Addition of new records to master ledger.
1.2 Entering of marks into master ledger
Results

2.5.2. ER Diagram



ER Diagram

2.6. System Design

2.6.1. Architecture of system – Proposals for new system

The systems study reveals a distinct need for an automated system to reduce the time delay caused by the various activities of the system. From the study it is obvious that the system should provide computer support for the following operations which are manual in the existing system: -

1) Master ledger is maintained as a table in Access using student roll-number as the primary key.
2) A procedure is written to display the list of students eligible for a particular semester examination. This procedure takes into consideration all the eligibility criteria mentioned previously.
3) Examination numbers are allotted to the eligible candidates and a table containing examination number and roll number is maintained. This table is used while updating the master ledger.
4) The supervisor report (attendance record) is entered. This record is compared with the mark sheet returned by the examiner and discrepancies (like marks awarded to an absentee or missing marks for a student present for the examination) are signaled.
5) Marks received from the examiner are entered. A procedure is written to combine the marks received for the two sections.
6) Procedure is written to handle the calculation of grace marks.
7) The master ledger is automatically updated.

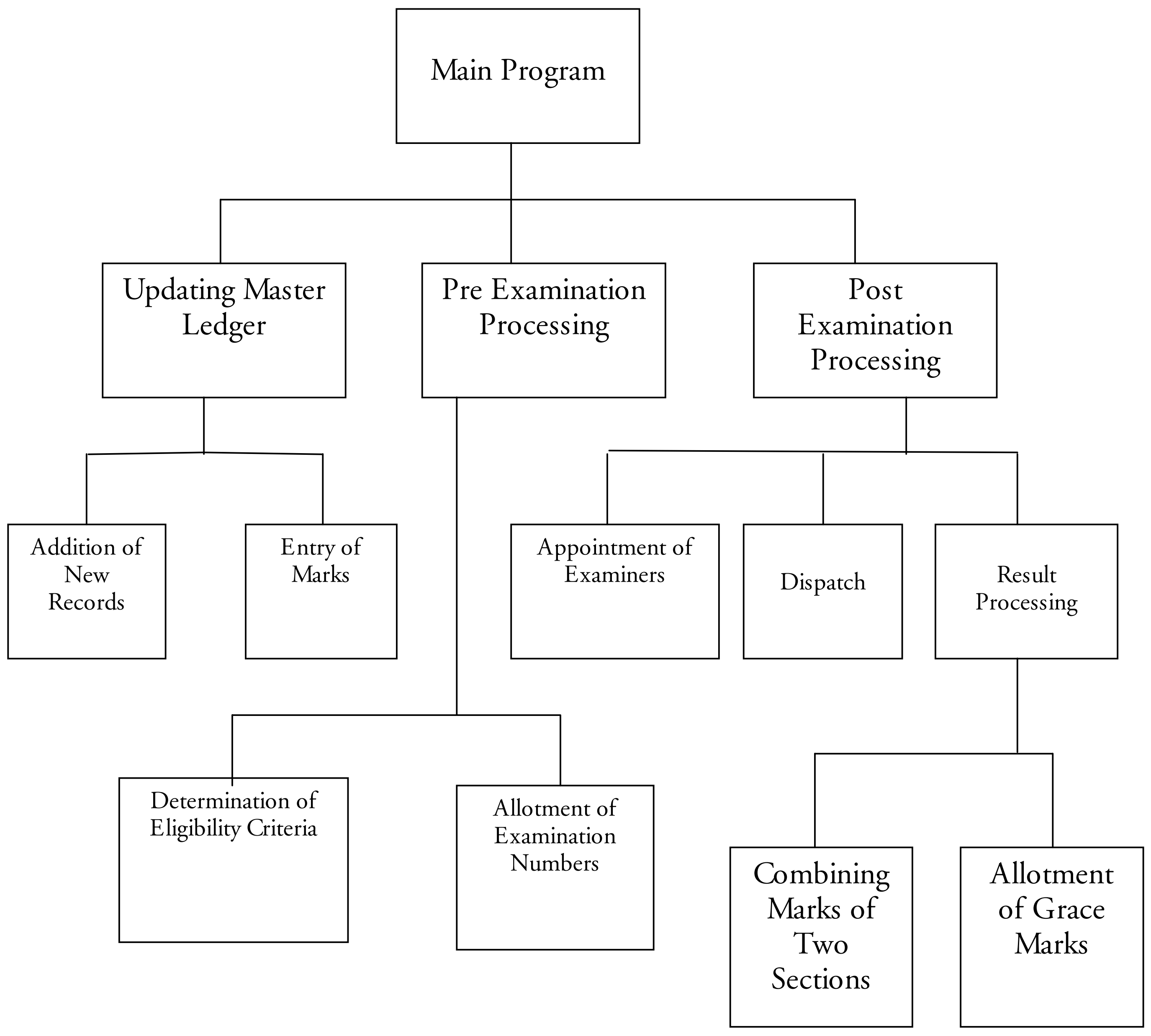
In addition to the above automation following changes are introduced into the new system:

1) Table containing the record of the internal and external examiners is maintained. This table contains personal information (like name, contact number, address), subject information (like subjects taught, college) and supplementary information.

The supplementary information contains past record of the examiner like whether he had earlier refused an appointment, or delayed the correction of answer papers, etc. This information is quite helpful in appointing an examiner.

2) A record is kept to indicate the date of dispatch of answer books and related matter. Also the date by which the corrected answer papers are expected is specified. This information is used to display automatic reminders if the papers are not received on time.

2.6.2. Modules


Main Program
Updating Master Ledger
Pre Examination Processing
Post Examination Processing
Addition of New Records
Entry of Marks
Appointment of Examiners
Dispatch
Result Processing
Determination of Eligibility Criteria
Allotment of Examination Numbers
Combining Marks of Two Sections
Allotment of Grace Marks

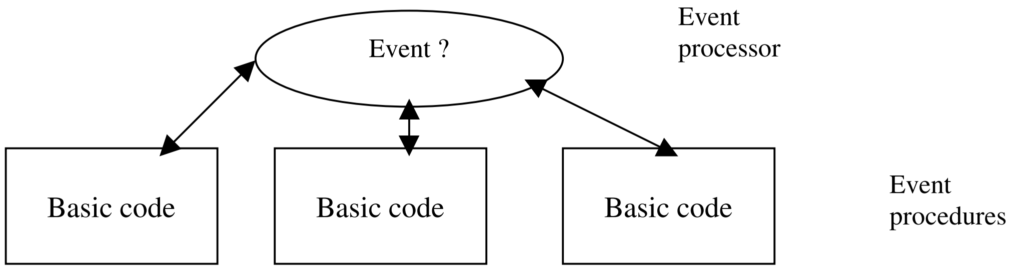
Chapter 3
Development Environment

3.1. Overview Of The Environment

What is Visual Basic?

- **Visual Basic** is a tool that allows us to develop Windows (Graphic User Interface GUI) applications. The applications have a familiar appearance to the user.

- Visual Basic is **event-driven,** meaning code remains idle until called upon to respond to some event (button pressing, menu selection, etc.). Visual Basic is governed by an event processor. Nothing happens until an event is detected. Once an event is detected, the code corresponding to that event (event procedure) is executed. Program control is then returned to the event processor.



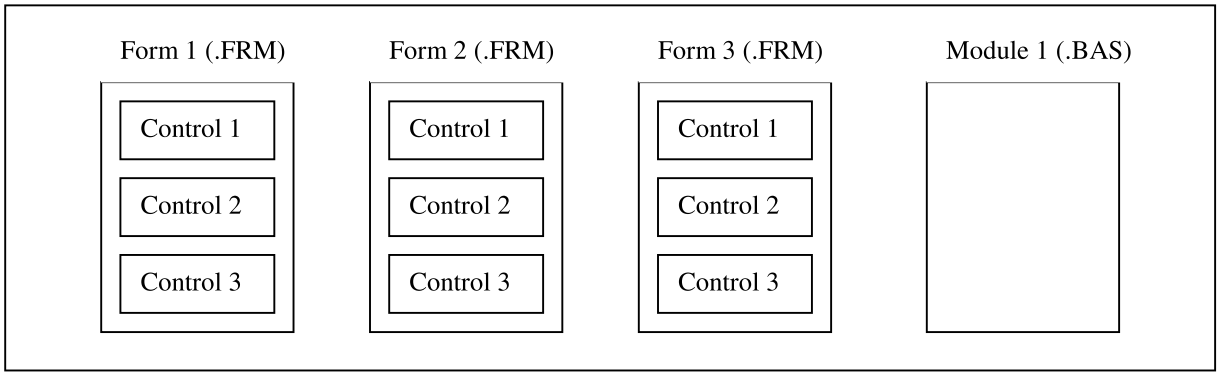
Features of Visual Basic

- Full set of objects - the user 'draws' the application
- Lots of icons and pictures for use
- Response to mouse and keyboard actions
- Clipboard and printer access

- Full array of mathematical, string handling, and graphics functions
- Can handle fixed and dynamic variable and control arrays
- Sequential and random access file support
- Useful debugger and error-handling facilities
- Powerful database access tools
- ActiveX support
- Setup Wizard makes distributing applications simple

Structure Of A Visual Basic Application

Project (.VBP , .MAK)



Application (Project) is made up of:

- **Forms** - Windows that the user creates for user interface
- Controls - Graphical features drawn on forms to allow user interaction (text boxes, labels, scroll bars, command buttons, etc.) (Forms and Controls are **objects.)**
- **Properties** - Every characteristic of a form or control is specified by a property. Example properties include names, captions, size, color, position, and contents. Visual Basic

applies default properties. The user can change properties at design time or run time.

- **Methods** - Built-in procedure that can be invoked to impart some action to a particular object.
- **Event Procedures** - Code related to some object. This is the code that is executed when a certain event occurs.
- **General Procedures** - Code not related to objects. This code must be invoked by the application.
- **Modules** - Collection of general procedures, variable declarations, and constant definitions used by application.

3.2. Database Related Concepts

What is a Relational Database?

A relational database is defined as a collection of data that indicates relation among data elements; or, to put it even more directly, a relational database is a collection of related data.

In order to build a collection of related data, three key building blocks are needed.

These building blocks are (from smallest to largest):

- Data fields (sometimes called data columns)
- Data records (also known as data rows)
- Data tables

Visual Basic Database Field Types

Visual Basic stores values in the data table in data fields. Visual Basic recognizes 14 different data field types that can be used

to store values. Each data field type has unique qualities that make it especially suitable for storing different types of data. Some are used to store images, the results of checkbox fields, currency amounts, calendar dates and various sizes of numeric values. The following table lists the 14 database field types recognized by Visual Basic.
The first column contains the Visual Basic data field type name. This is the name we use when creating data tables using the Visual Data Manager.
The second column shows the number of bytes of storage taken by the various data field types. If the size column is set to "V," the length is variable and is determined by the programmer at design time or by the program at runtime.

The third column in the table shows the equivalent Visual Basic data type fur the associated database field type. This column gives what Visual Basic data type one can use to update the database field.

Table : Visual Basic Data Field Types

Data Field Type	***Size***	***VBType***	***Comments***
BINARY	V	(none)	Limited to 255 bytes
BOOLEAN	I	Boolean	Stores 0 or -1 only
BYTE	I	Integer	Stores 0 to 255 only
COUNTER	8	Long	Auto-incrementing Long type
CURRENCY	8	Currency	15 places to left of decimal, 4 to right
DATETIME	8	Date/Time	Date stored on the left of decimal point, time stored on the right
DOUBLE	8	Double	
GUID	16	(none)	Used to store Globally Unique

			Identifiers
INTEGER	2	Integer	
LONG	8	Long	
LONGBINARY	V	(none)	Used for OLE objects
MEMO	V	String	Length varies up to 1.2 gigabytes
SINGLE	4	Single	
TEXT	V	String	Length limited to 255

Visual Basic Database Objects

Data objects are used within a Visual Basic program to manipulate databases as well as the data tables and indexes within the database. The data objects are the representations (in program code) of the physical database, data tables, fields, indexes, and so on. Every Visual Basic program that accesses data tables uses data objects. Even if one is using the data-aware controls (for example, the data control and bound input controls) and are not writing programming code, one is still using Visual Basic data objects.

The primary data object used in Visual Basic programs is the Recordset object. This is the object that holds the collection of data records used in Visual Basic programs. There are three different types of Recordset objects. They are:

- Dynaset – type recordset object
- Table – type recordset object
- Snapshot – type recordset object

Any of these recordset objects can be used to gain access to an existing data table in a database. However, they each have unique properties and behave differently at times.

Dataset-Oriented Versus Data Record-Oriented

The database model behind the Microsoft Access database and other SQL oriented databases is quite different from the database model behind traditional PC databases such as FoxPro, dBASE, and Paradox. Traditional PC databases are record-oriented database systems. Structured Query Language (SQL) databases are dataset oriented systems.

In record-oriented systems, database operations are done one record at a time. The most common programming construct in record-oriented systems is the loop. Processing in record-oriented systems usually involves creating a routine that reads a single data record, processes it, and returns to read another record until the job is completed. PC databases use indexes to speed the process of locating records in data tables. Indexes also help speed processing by allowing PC databases to access the data in sorted order (by LastName, by AccountBalance, and so on).

In data-oriented systems, such as Microsoft Access, database operations are performed one set at a time, not one record at a time. The most common programming construct in set-oriented systems is the SQL statement. Instead of using program code to loop through single records, SQL databases can perform operations on entire tables from just one SQL statement.
In dataset-oriented databases, single statements are written that selects only the records needed to perform a database operation. Alter identifying the dataset. we apply, the operation to all records in the set. In dataset systems, indexes are used to maintain database integrity more than to speed the location of specific records.

Visual Basic and Data Objects

Visual Basic database objects are dataset-oriented. Visual Basic programs generally perform better when data operations are done with a dataset than when data operations are done on single records. Some Visual Basic objects work well when performing record-oriented operations; most do not. The Visual Basic table-type Recordset object is very good at performing record-oriented processing. The Visual Basic Dynaset- and snapshot-type Recordset objects do not perform well on record-oriented processes.

Dataset Size Affects Program Performance

Unlike record-oriented systems, the size of the dataset created affects the speed at which Visual Basic programs operate. As a data table grows, the program's processing speed can deteriorate. If working in a network environment where the **machine requesting** data and the machine storing the data are separated, sending large datasets over the wire can affect not only the application, but also all applications running on the network. For this reason, it is important to keep the size of the datasets as small as possible. This does not mean one has to limit the number of records in the data tables. Visual Basic data objects give one the power to create datasets that are the proper size for one's needs.

3.3. Types of Data Objects

1. The Dynaset-Type Recordset Data Object

The Visual Basic Dynaset-type Recordset data object is the most frequently used data object in Visual Basic programs. It is used to dynamically gain access to part or all of an existing data table in a database, hence the name Dynaset. When one sets the DatabaseName and RecordSource properties of a Visual Basic data control. one is actually creating a Visual Basic Dynaset-type Recordset.

Dynaset-type Recordset can also be created by using the CreateDynaset method of the Database object.

When a Visual Basic Dynaset-type Recordset is created, no new physical table is created in the database. A Dynaset exists as a virtual data table. This virtual table usually contains a subset of the records in a real data table, but it can contain the complete set. Because creating a Dynaset does not create a new physical table, Dynasets do not add to the size of the database. However, creating Dynasets does take up space in RAM on the machine that creates the set (the one that is running the program). Depending on the number of records in the Dynaset, temporary disk space can also be used on the machine requesting the dataset.

Strengths of the Dynaset-Type Recordset Object

There are several reasons to use Dynasets when accessing data

- In general, Dynasets require less memory than other data objects and provide the most update options, including the capability to create additional data objects from existing Dynasets. Dynasets are the default data objects for the Visual Basic data control, and they are the only updatable data object that can be used for databases connected through Microsoft's Open Database Connectivity (ODBC) model. Dynasets Are Really Key Sets. Visual Basic Dynasets use relatively little workstation memory, even for large datasets.
- The set of keys is stored in RAM and--if the set is too large to store in RAM alone--in a temporary file on a local disk drive. As one scrolls through the dataset, Visual Basic retrieves actual records as needed from the physical table used to create the Dvnaset. If a single text box is present on the form, Visual Basic retrieves the data from the table one record at a time. If a grid of data or a loop that collects several records from the table in succession is present, a small set of the records in the dataset is retrieved by Visual Basic.

Visual Basic also caches records at the workstation to reduce requests to the physical data table, which speeds performance.

- Dynasets Are Dynamic Even though Dynasets are virtual tables in memory created from physical tables, they are not static copies of the data table. After a Dwaset is created, if anyone else altos the underlying data table by nwditj~in~~, addin~~, or deleting records, changes in the Dynaset can be seen as soon as the Dvnaset is refreshed. Refreshing the Dynaset can be done using the *Refresh* method. Dynasets can also be refreshed by moving the record pointer using the arrow keys of the data control or using the *MoveFirst, MoveNext, MovePrevious,* and MoveLast methods. Moving the pointer refreshes only the records to be read, not the entire Dynaset.
- Dynasets Can Be Created from More than one Table. A single. view that contains selected records can be created from several tables, the view can be updated, and therefore all the underlying tables of the data can be updated at one time. This is a very powerful aspect of a Visual Basic Dynaset data object. Using Visual Basic, Dynasets, one can create virtual tables that make it easy to create simple data entry screens and display graphs and reports that show specialized selections of data. The Dynaset data object is the only data object from which we can create another Dynaset. Additional Dynasets can be created by using the Clone method or the CreateDynaset method. When a Dynaset is cloned, an exact duplicate of the Dynaset is created. One can use this duplicate to perform look-ups or to reorder the records for a display. Cloned Dynasets take up slightly less room than the original Dynaset.

Limitations of the Dynaset-Type Recordset Data Object

Although the Dynaset is an excellent data object, it has a few drawbacks that must be considered. Chief among these is that Dynasets do not, allow one to specify an existing index, and also one cannot use the Visual Basic Seek method to quickly locate a single record in the Dynaset. Also, errors can occur when displaying records in a Dvnaset if the records in the underlying data table have been altered or deleted by another user. Dynaset Access and Seek Limitations

Dynasets cannot make use of Index objects that exist in a database because the Index is built to control the entire data table and not just a subset of the data. Because Dynasets could be subsets of the data table, the Index is useless.

These are only minor limitations. If one has defined an Index in the underlying table with the Primary flag turned on, the Visual Basic data engine uses the primary key index when creating the Dynaset. This usually puts the Dynaset in optimal order. Even though one cannot use the Seek method on a Dynaset, one can use the FindFirst, FindNext, FindPrevious, and FindLast methods. Even though they are not true index searches, they are fast enough for operations on small- to medium-sized Dynasets.

If a programmer opens a database and creates a Dynaset from an underlying` table while another user has also opened the same database and created a Dynaset based on the same underlying data table, it is possible that both users will attempt to edit the same data record. If both users edit the same record and both attempt to save the record buck to the underlying table, the second person who attempts to save the record receives a Visual Basic error. When the second person tries to save the record. Visual Basic discovers that the original record in the underlying data table has been altered. In order to maintain database integrity, Visual Basic does not allow the second person to update the table.

2. The Table-Type Recordset Data Object

The Visual Basic Table-type Recordset data object is the data object that gives the programmer access to the physical data table, sometimes referred to as the base table. The programmer can use the Table object to directly open the table defined by Data Manager (or some other database definition tool).

Visual Basic Table objects have their drawbacks, too. Select statements cannot he used to initialize a Table object. One cannot use Bookmarks, create

Filters. or sort the table. Furthermore, the Table data object cannot be used to access ODBC data sources. Only Dynasets and Snapshots can be used with ODBC data sources.

Strengths of the Table-type Recordset Data Object

The real strength of Table objects is that the programmer can Specify Index objects to use when searching for specific records in the table and can use the Seek method. Table objects also use limited workstation memory. Table objects use limited workstation memory because Visual Basic caches pointers to the actual records at the workstation instead of loading all the records into workstation memory. This gives the programs the fastest access speed of all the data objects when searching for a single record.

Table-type Recordset data objects also give the programmer instant information on the state of the data table. This is important in a multi-user environment. As soon as a user adds or deletes a record from the table, all other users who have the data table opens as a Visual Basic Table object also see the changes. Unlike Dynasets and Snapshots, table objects are not subsets of the data table. They contain all the records in the table at all times.

Limitations of the Table-Type Recordset Data Object

Even though the Visual Basic table type recordset object provides the fastest search speed of any of the data objects, it also has certain drawbacks. One cannot sort a table; also one cannot use the Table object when accessing ODBC data sources: and one cannot use the Visual Basic data control to access a Table object. Tables Cannot Use Bookmarks, Sorts, or Filters Unlike Dynasets and Snapshots, Visual Basic Table objects cannot be sorted, filtered, or have Bookmarks set.

Table objects can't use Bookmarks, so the user can't mark his place in a table, move around, and then return to the location using Visual Basic Bookmarks. He can’t however, save the table index value instead.

3. The Snapshot-Type Recordset Data Object

Visual Basic Snapshot-type Recordset objects are almost identical to Dynasetype Recordsets in behavior and properties. However, there are two major differences between Snapshot objects and Dynaset objects. These two differences are the most important aspects of Snapshots.

- Snapshots are stored entirely in workstation memory.
- Snapshots are read-only and non-updateable objects.

Snapshot objects are stored entirely at the workstation, for example, if a programmer creates a Snapshot that contains 500 data records, all 500 records are sent from the data table directly to the workstation and loaded into RAM memory. If the workstation does not have enough RAM available, the records are stored in a temporary file on a local disk drive.

Because all the requested records are loaded on the local machine, initial requests for data can take longer with snapshots that with Dynasets. However when the data records are retrieved and stored locally subsequent access to records within the snapshot object is faster than with the dynaset object. Also because all records must be stored locally, one must be careful not to request too large a dataset to avoid running out of local RAM or disk space.

Snapshots are static views of the underlying data tables. If a user requests a set of data records in a Snapshot object, and then someone deletes several records from the underlying data table, the Snapshot dataset does not reflect the changes in the Underlying table. The only way the user can learn about the changes in the underlying data tables is to create a new Snapshot by making a new request. Snapshot-Type Recordsets Are ReadOnly Data Objects. Snapshots cannot be used to update data tables. They can only be used to view data. This is because Snapshots are actually a copy of the data records created at the local workstation.

3.4. Selecting Data With SQL

SQL is a powerful manipulation language used by Visual Basic and the Microsoft Access Jet database engine as the primary method for accessino the data in databases. SQL statements fall into two broad categories:

• Data manipulation language statements (DML) and .

• Data definition language statements (DDL)

DML statements are used to select, sort, summarize, and calculate the inturmation stored in the data tables. The DDL statements enable us to define data tables, indexes. and database relationships.

What Is SQL?

SQL stands for Structured Query Language. It was developed in the 1970s at IBM as a way to provide computer users with a standardized method for selecting data from various database formats. The intent was to build a language that was not based on any existing programming language, but could be used within any programming language as a way to update and query information in databases.

SQI, statements are just that -- statements. Each statement can perform operations on one or more database objects (tables, columns, indexes, and so on). Most SQI_ statements return results in the form of a set of data records commonly referred to as a view. SQL is not a particularly friendly language. Many programs that use SQL statements hide these statements behind point-and-click dialogs, query-by-example `;rids. and other user-friendly interfaces. In fact, even when one accesses data stored in a relational database SQL is used.

SQL Statements

The SELECT FROM Statement

The SELECT FROM statement is used to pick records from one or more tables in a database. The results of a SELECT-FROM statement are returned as a view. This view is a subset of the source data. In Visual Basic, the view can be returned as a Reeordset, Table, Dynaset, or Snapshot. In its simplest form, a SELECT FROM statement contains two parts:

- A list of one or more table columns to select
- A list of one or more tables that contain the requested columns.

To return all the columns from a table, list each column in the SELECT statement. This works if there are only a few columns in the table. However, if there are many columns, it can become quite tedious. There is a shortcut. To automatically list all columns in the table in the result set, instead of typing column names, type an asterisk (*) The asterisk tells SQL to return all columns in the requested table.

The ORDER BY Clause

By using the SELECT FROM statement, the records returned in the result set are returned in the order in which they were found in the underlying table. To display the results of the SELECT FROM statement in a specialized sorted order use the ORDER BY clause. Placing ASC or DESC after each field in the ORDER BY clause indicates the order, in which the column is sorted, ascending or descending. If no order is supplied. SQL assumes ascending order.

The WHERE Clause

One of the most powerful aspects of the SELECT-FROM statement is its capability to control the content of the result set using the WHERE clause. There are two ways to use the WHERE clause to control the content of the result set:

- Use WHERE to limit the contents of a result set.
- Use WHERE to link two or more tables in a single result set.

Using WHERE to Limit the Result Set : The WHERE clause enables one to perform logical comparisons on data in any column in the data table. In its simplest form the WHERE clause consists of the following: . WHERE column = value

In this line, column represents the name of the column in the requested data table, and value represents a literal value such as NY or Smith. The WHERE clause is always preceded by a SELECT FROM statement.

Visual Basic SQL also supports the use of BETWEEN AND. IN. and LIKE' comparisons. The following SQL statement illustrates the use of BETWEEN AND in a WHERE clause.

SQL Aggregate Functions

The SQL standards define a core set oh functions that are present in all SQL compliant systems. These functions are known as aggregate functions. Aggregate functions are used to quickly return computed results of numeric data stored in a column. The SQL aggregate functions available through the Microsoft Access Jet database engine are

- AVG: Returns the average value of all the values in a column
- COUNT: Returns the number of columns and is usually used to determine the total rows in a view. COUNT is the only standard SQL aggregate function that can be applied to a non-numeric column.

- SUM: Returns the total of all the values in a column.
- MAX: Returns the highest of all the values in a column.
- MIN: Returns the lowest of all the values in a column.

3.5. Visual Basic and the Microsoft Jet Engine

Microsoft Jet is the heart of the Visual Basic database system. It is the part of Visual Basic that handles all database operations. Whether one is reading a Microsoft Access-format database, accessing a FoxPro file, or connecting to a back-end database server using ODBC, Microsoft .let is there. Visual Basic can also be used to create a link between an existing Microsoft Jet database and data in non-Microsoft Jet databases. This process of attaching external data sources provides an excellent way to gain the advantages of the Microsoft Jet data access object layer without havino to convert existing data to Microsoft Jet format.

The several object collections that exist in Visual Basic Microsoft .let databases. including those objects available in the Microsoft Jet 3.5 data engine are

- The DBEngine object
- The Workspace object
- The Database object
- The TabIeDef object
- The Field object
- The Index object
- The Relation object
- The Connection object
- The Recordset object

What Is the Microsoft Jet Database Engine?

The idea behind Microsoft Jet is that the user can use one interface to access multiple types of data. Microsoft designed Microsoft Jet to present a consistent interface to the user regardless of the type of data the user is working with. Consequently the user can use the same Microsoft Jet functions that he uses to access an ASCII text tile or Microsoft Excel spreadsheet to also perform data operations on Microsoft Access databases.

The Microsoft Jet engine is not a single program; it is a set of routines that work together. The Microsoft Jet engine talks to a set of translation routines. These routines, convert the Microsoft Jet request into a request that the target database can understand.

Advantages of Microsoft Jet over the Data Control Object

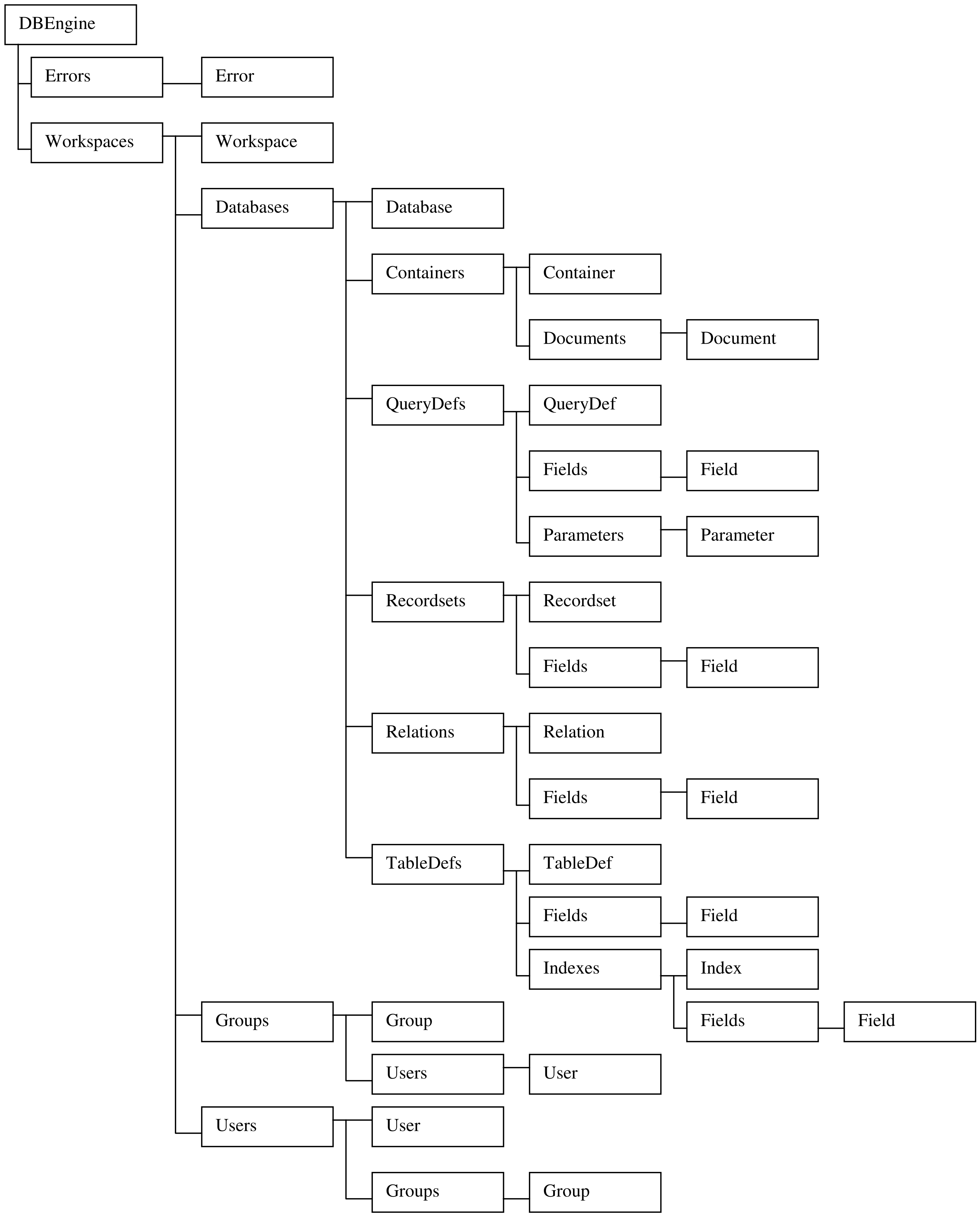
The data control object is used to perform database administrative tasks. The dataaccess objects (DAOs) perform all of the services that the data control does, as well as many more. The data-access objects give complete control over database management.

However, if possible, one must use the data control object to manage the data. This is because it is much easier to use as many of the administrative functions are handled automatically.

Microsoft Jet Data Objects

Microsoft Jet is organized into a set of data-access objects. Each of the objects has collections, properties, and methods:

- Collections: Data-access objects that contain the same type of objects.
- Properties: The data contained within an object (control button, form, and so on) that defines its characteristics. An object's properties can be set.
- Methods: The procedures that can be performed on an object. The user invokes a method.


DBEngine
Errors
Error
Workspaces
Workspace
Databases
Database
Containers
Container
Documents
Document
QueryDefs
QueryDef
Fields
Field
Parameters
Parameter
Recordsets
Recordset
Fields
Field
Relations
Relation
Fields
Field
TableDefs
TableDef
Fields
Field
Indexes
Index
Fields
Field
Groups
Group
Users
User
Users
User
Groups
Group

3.6. The Visual Basic Toolbox

Command Button

Command button is probably the most widely used control. It is used to hcoin, interrupt, or end a particular process

Label Boxes

A **label box** is a control used to display text that a user can't edit directl%. We've seen, though, in previous examples, that the text of a label box can be changed at runtime in response to events.

Text Boxes

A **text box** is used to display information entered at desion time, by a user at runtime, or assigned within code. The displayed text may be edited.

Check Boxes

Check boxes provide a way to make choices from a list of potential candidates. Some, all, or none of the choices in a group may be selected.

Option Buttons

Option buttons provide the capability to make a mutually exclusive choice among a group of potential candidate choices. Hence, option buttons work as a group, only one of which can have a True (or selected) value.

Frames

Frames provide a way of grouping related controls on a form. .-end. in the case of option buttons, frames affect how such buttons operate.
To group controls in a frame, one must first draw the frame. Then, the associated controls must be drawn in the frame. This allows the user to move the frame and controls together

List Boxes

A **list box** displays a list of items from which the user can select one or more items. If the number of items exceeds the number that can be displayed, a scroll bar is automatically added.

Combo Boxes

The combo box is similar to the list box. The differences are a combo box includes a text box on top of a list box and only allows selection of one item. In some cases, the user can type in an alternate response.

Line Tool

The line tool creates simple straight line segments of various width and color. Together with the shape tool discussed next, one can use this tool to 'dress up' applications.

Shape tool

The shape tool can create circles, ovals, squares, rectangles, and rounded squares and rectangles. Colors can be used and various fill patterns are available.

Horizontal and Vertical Scroll Bars

Horizontal and vertical scroll bars are widely used in Windows applications. Scroll bars provide an intuitive way to move through a list of information and make great input devices.

Picture Boxes

The **picture box** allows us to place graphics information on a form. It is best suited for dynamic environments - for example, when doing animation.

Image Boxes

An **image box is very** similar to a picture box in that it allows us to place graphics information on a form. Image boxes are more suited for static situations - that is, cases where no modifications will be done to the displayed graphics.

Drive List Box

The **drive list box** control allows a user to select a valid disk drive at run-time. It displays the available drives in a drop-down combo box. No code is needed to load a drive list box; Visual Basic does this for us. We use the box to get the current drive identification.

Directory List Box.

The **directory list box** displays an ordered, hierarchical list of the user's disk directories and subdirectories. The directory structure is displayed in a list box. Like. the drive list box, little coding is needed to use the directory list box - Visual Basic does most of the work for us.

File List Box

The **file list box** locates and lists files in the directory specified by its Path property at run-time. The user may select the types of tiles he wants to display in the tile list box.

Chapter 4
Design Environment

4.1. About Microsoft Access

Microsoft Access manages all the information from a single database tile. Within the tile, data is divided into separate storage containers called tables; online forms can be used to view, add, and update table data; queries can be used to find and retrieve just the data one wants; and data can be analyzed or printed in a specific layout using reports.

Tables

A table is a collection of data about a specific topic such as products or suppliers.' Using a separate table for each topic, stores the data only once, which crakes the database more maintainable and reduces data-entry errors. Tables organize data into columns (called fields) and rows (called records).

In table **Datasheet view,** it is possible to add, edit, or view the data in a table. One can also check the spelling and print the table's data, filter or sort records. change the datasheet's appearance, or change the table's structure by adding or deleting columns.

1n **table Design view,** an entire table can be created from scratch, or add, delete. or customize an existing table's fields.

Forms

Forms can be used for a variety of purposes. Most of the information in a form comes from an underlying record source. Other information in the form is stored in the form's design.

Graphical objects called controls are used to create the link between a form and its record source. The most common type of control used to display and enter data is a text box.

Queries

A query in Access is a way of getting at the information stored in a database. Once the criteria for the information wanted is specified Access retrieves it. Most queries return a datasheet called a *recordset* containing the appropriate information. But one can also use *action queries* to perform actions on data.

There are four types of action queries y Make-table queries create a new table from data in one or more tables. Delete queries delete a group of records from one or more tables. Append queries append a group of records from one table to another. Y Update queries update a group of records.

Reports

In Access, reports are used to control the appearance of the data when it is printed. One can choose what information to include in a report, and can also bind a report to either a table or to a query that draws information from several tables.

In the Design view, one can create WYSIWYG reports that present information effectively. In the design of a report any tittle, headings, and so on, can be included that can then appear in each instance of the report. Along with data, reports can include pictures and graphic elements. As well as the design view, reports offer Print Preview and Layout Preview.

Print preview gives an idea of how the report shall look when it is printed. Layout preview is a preview that one can use when working in Design view to get an idea of how the report will look with sample data in it.

Macros

A macro is a way of automating routine tasks, so that the computer can perform them for us as a batch, instead of we doing them each time. Access uses Visual Basic for Applications as the language for its macros. Access's macro builder can be used to choose the actions the macro should perform.

Modules

Modules are containers for macro code. Each form or report includes a module to hold any code associated with it, so when one moves the form or report from place to place, it takes the code along with it in the module.

4.2. About designing a database

Before Microsoft Access is used to actually build the tables. forms. and other objects that will make up the database, it is important to take time to design the database. A good database design is the keystone to creating a database that does what one wants it to do effectively, accurately, and efficiently.

4.3. Steps in designing a database

The basic steps in designing a database are

1. One should determine the purpose of the database.
2. The tables needed in the database should be determined.
3. The fields needed in the tables should be identified.
4. Fields with unique values should be identified.
5. The relationships between tables should be determined
6. The design should be refined.
7. Data is added and other database objects are created.
8. Microsoft Access analysis tools should be used.

Creating a database

Microsoft Access provides two methods to create a database. One can create a blank database and then add the tables, forms, reports, and other objects later -- this is the most flexible method, but it requires each database element to be defined separately. Or one can use a Database Wizard to create in one operation the required tables. forms, and reports for the type of database he chooses --this is the easiest way to start creating a database. Either way, a database can modified and extended at any time afer it has been created.

Creating a new blank table

Microsoft Access provides four ways to create a blank (empty) table:

1. The Database Wizard can be used to create in one operation all the tables, forms. and reports required for an entire database. The Database Wizard creates a new database: it can't be used to add new tables, forms, or reports to an existing database.
2. The Table Wizard can be used to choose the fields for the table from a variety of predefined tables such as business contacts, household inventory, or medical records.
3. One can enter data directly into a blank datasheet. When the new datasheet is saved. Microsoft Access will analyze the data and automatically assign the appropriate data type and format for each field.
4. The Design view can be used to specify all of the table details from scratch.

Regardless of which method one uses to create a table. one can use table Design view at any time to customize the table further, such as adding new fields, setting default values, or creating input masks.

4.4. Use of Primary Keys

The power of a relational database system such as Microsoft Access comes from its ability to quickly find and bring together information stored in separate tables using queries, forms, and reports. In order to do this, each table should include a field or set of fields that uniquely identify each record stored in the table. This information is called the primary key of the table. Once a primary key for a table is designated, to ensure uniqueness, Microsoft Access will prevent any duplicate or Null values from being entered in the primary key fields.

There are three kinds of primary keys that can be defined in Microsoft Access: AutoNumber Single-field and Multiple-field.

AutoNumber primary keys

An AutoNumber field can be set to automatically enter a sequential number as each record is added to the table. Designating such a field as the primary key for a table is the simplest way to create a primary key. If a primary key has not been defined for a table then while saving Microsoft Access will ask if it should create a primary key. If the answer is Yes, Microsoft Access will create an AutoNumber primary key.

Single-field primary keys

If there are fields that Contain Unique Values Such as ID numbers or part numbers, one can designate any of those fields as the primary key. If the field selected as primary key does have duplicate or Null values, Microsoft Access won't set the primary key. .A Find Duplicates query can be run to determine which records contain duplicate data.

Multiple-field primary keys

In situations where one cannot guarantee the uniqueness of any single field, two or more fields can be designated as the primary key. The most common situation where this arises is in the table used to relate two other tables in a many-to-many relationship.

4.5. Indexes

Create an index to find and sort records faster:

An index helps Microsoft Access find and sort records faster. Microsoft Access uses indexes in a table as an index is used in a book: to find data, it looks up the location of the data in the index. Indexes can be created based on a single field or on multiple fields. Multiple-field indexes enable us to distinguish between records in which the first field may have the same value.

Multiple-field indexes

If a search is to be done often a good feature to use is to search or sort by two or more fields at a time, an index for that combination of fields can be created. For example, if a criterion is set often for the LastName and FirstName fields in the same query, it makes sense to create a multiple-field index on both fields.

When a sort on a table is by a multiple-field index, Microsoft Access sorts first by the first field defined for the index. If there are records with duplicate values in the first field, Microsoft Access sorts next by the second field defined for the index. and so on.

4.6. Relationships

After tables have been set up for each subject in the database. a way of telling Microsoft Access how to bring that information back together again is needed. The first step in this process is to define relationships between the tables. After that is done, queries, farms, and reports to display information from several tables at once can be created.

4.7. Referential Integrity

Referential integrity is a system of rules that Microsoft Access uses to ensure that relationships between records in related tables are valid, and that related data are not accidentally deleted or changed. Referential integrity can be set when all of the following conditions are met: 1. The matching field from the primary table is a primary key or has a unique index. 2. The related fields have the same data type. There are two exceptions. An AutoNumber field can be related to a Number field with a FieldSize property setting of Long Integer, and an AutaNumber field with a FieldSize property setting of Replication ID can be related to a Number field with a FieldSize property setting of Replication ID. 3. Both tables belong to the same Microsoft Access database. if the tables are linked tables, they must be tables in Microsoft Access format, and the database in which they are stored must be opened to set referential integrity. Referential integrity can't be enforced for linked tables from databases in other formats.

4.8. Data Types In Access

Data type	Storage size	Range
Byte	I byte	0 to 255
Boolean	2 bytes	True or False.
Integer	2 bytes	-32,768 to 32,767.
Long	4 bytes	-2,147,483,648 to 2,147,483,647.
Single	4 bytes	-3.4028231338 to -1.401298E-45 for negative values; 1.401298E-45 to 3.4028231338 for positive values.
Double	8 bytes	-1.797693 133213308 to -•1.940656424713-324 tier negative values; 4.94065641247E-324 to 1.7976931623213308 for positive values.
Currency	8 bytes	-922,337,203,685,477.5808 to 922,337,203,685,-177.5807.
Date	8 bytes	January I, 100 to December 31, 9999.
Object	4 bytes	Any Object reference.
String	10 bytes + string length	0 to approximately 2 billion (approximately 65.-100 for Microsoft Windows version 3.1 and earlier).
String	Length of string	I to approximately 65,400.
Variant (with numbers)	16 bytes	Any numeric value up to the range of a Double.

Variant (with characters)	22 bytes + string length	Same range as for variable-length String.
User-defined	Number required by elements	The range of each element is the same as the range of its data type.

Chapter 5
Database Design

This chapter focusses on the design of the database used in the final application. The exam section database consists of a number of tables that have been designed using Microsoft Access, a relational database management system (RDBMS).

A description of the tables implemented in the project follows.

5.1. The alloted_examiners table

This table stores the details of alloted examiners.

subject_id - Long Integer
internal_examiner_id - Long Integer
external examiner id - Long Integer

5.2. The eligible_candidates table

This table holds the list of candidates who are eligible to attend the examinations.

semester - Long Integer
roll_number - Long Integer
fees_ paid - Long Integer
eligibile_candidate_id - Long Integer (Auto Number)

5.3. The examination_number_allotment table

This table is used to hold the exam numbers of students who have paid the fees.

examination_number_id - Long Integer (Auto Number)
examination_number - Long Integer

eligible_candidate_ id - Long Integer

5.4. The external_examiner_can_correct table

This table holds information about the various subjects which external examiners can correct.

external_examiner_ id - Long Integer
subject_id - Long Integer

5.5. The external_examiner_database table

This table holds the external examiner information.

external_examiner_id - Long Integer (Auto Number)
name - Text
address - Memo
contact_number - Text
college - Text
supplementary_information - Memo

5.6. The internal_examiner_can_correct table

This table holds information about the various subjects which internal examiners can correct.

internal_examiner_ id - Long Integer
subject_id - Long Integer

5.7. The internal_examiner_database table

This table holds the internal examiner information.

internal_examiner_id - Long Integer (Auto Number)
name - Text
address - Memo
contact_number - Text
supplementary_information - Memo

5.8. The marks_database table

This table holds the raw marks data.

marks_database_id - Long Integer (Auto Number)
eligibile_candidate_id - Long Integer
subject_id - Long Integer
marks - Long Integer
section_id - Long Integer

5.9. The master_ledger table

This table contains the entire master ledger, with student-wise information.

roll_number - Long Integer
student_name_first - Text
student_name_middle - Text
student_name_surname - Text
name_address_of_father_or_guardian - Memo
date_of_birth - Date/Time
previous_college_and_university_attended - Text
last_exam_passed - Text
permanent_residence - Memo
month_of_admission - Text
year_of_admission - Long Integer

category - Text
semester_to_which_admitted - Long Integer
kts_in_sem1 - Long Integer
kts_in_sem2 - Long Integer
kts_in_sem3 - Long Integer
kts_in_sem4 - Long Integer
kts_in_sem5 - Long Integer
kts_in_sem6 - Long Integer
kts_in_sem7 - Long Integer
kts_in_sem8 - Long Integer
sem1_remarks - Text
sem2_remarks - Text
sem3_remarks - Text
sem4_remarks - Text
sem5_remarks - Text
sem6_remarks - Text
sem7_remarks - Text
sem8_remarks - Text
transference_certificate - Text
month_of_leaving - Text
year_of_leaving - Long Integer
reason_of_leaving - Memo
remarks - Memo

5.10. selected_external_examiners_after_acceptance table

This table contains the shortlisted external examiners.

external_examiner_id - Long Integer
date_of_dispatch - Date/Time
date_of_return - Date/Time

subject_id - Long Integer

section_id - Long Integer

correction_status - Long Integer

5.11. selected_internal_examiners_after_acceptance table

This table contains the shortlisted internal examiners.

internal_examiner_id - Long Integer

date_of_dispatch - Date/Time

date_of_return - Date/Time

subject_id - Long Integer

section_id - Long Integer

correction_status - Long Integer

5.12. The subjects table

This table holds all subjects of all branches and semesters.

subject_id - Long Integer (Auto Number)

semester - Long Integer

branch - Text

subject_name - Text

head_of_passing - Long Integer

maximum_marks - Long Integer

passing_marks - Long Integer

5.13. The branch_table table

This table holds the branch identification numbers of various branches.

branch_name – Text

branch_id – Integer

5.14. The eligibility_criteria table

This table holds the eligibility criteria for appearing for exams.

oddkts – Number

oddsems – Number

evenkts – Number

evensems - Number

5.15. The head_of_passing table

This table hold information for various heads of passing.

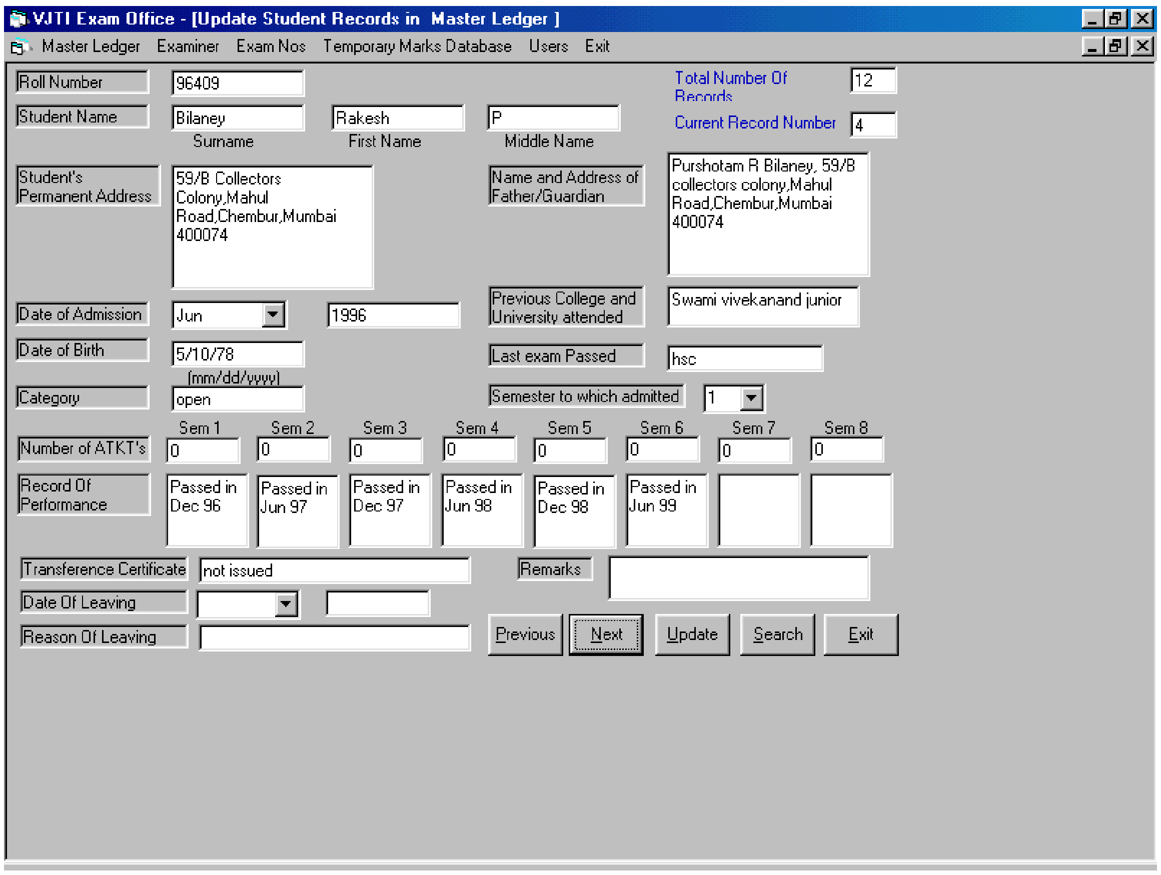
hop_id - Number

hop_name - Text

Chapter 6
Forms Design

Forms are the primary means of accepting user input in most database intensive applications, and the Computerization of the examination section is no exception. Forms are also the best means for displaying the stored data to the user. In this chapter we describe in detail the design of the forms implemented in the application.

6.1. The Master Ledger Forms



Roll Number	96409		Total Number Of Records	12
			Current Record Number	4
Student Name	Bilaney (Surname)	Rakesh (First Name)	P (Middle Name)	
Student's Permanent Address	59/B Collectors Colony,Mahul Road,Chembur,Mumbai 400074		Name and Address of Father/Guardian	Purshotam R Bilaney, 59/B collectors colony,Mahul Road,Chembur,Mumbai 400074
Date of Admission	Jun	1996	Previous College and University attended	Swami vivekanand junior
Date of Birth	5/10/78 (mm/dd/yyyy)		Last exam Passed	hsc
Category	open		Semester to which admitted	1

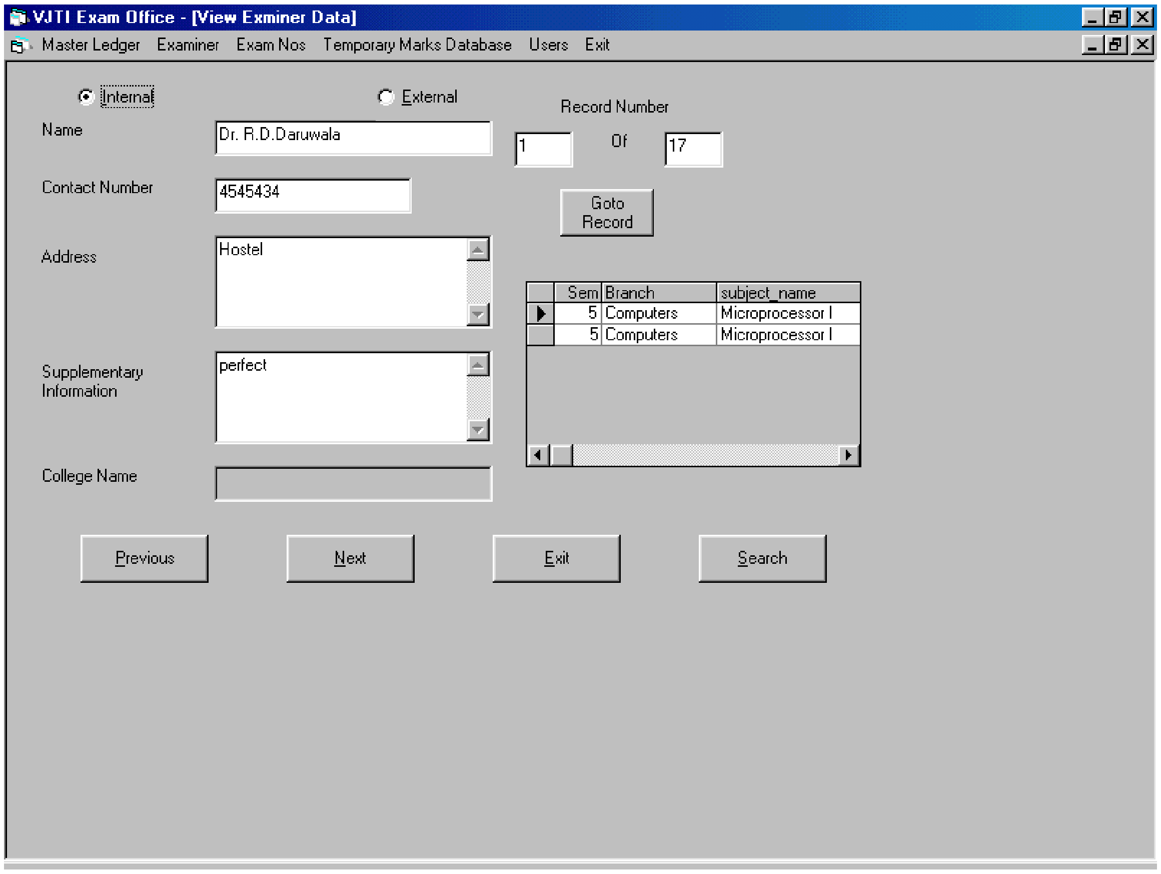
	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5	Sem 6	Sem 7	Sem 8
Number of ATKT's	0	0	0	0	0	0	0	0
Record Of Performance	Passed in Dec 96	Passed in Jun 97	Passed in Dec 97	Passed in Jun 98	Passed in Dec 98	Passed in Jun 99		

Transference Certificate	not issued	Remarks	
Date Of Leaving			
Reason Of Leaving			

This form is varies slightly depending on the operation to be performed. Various operations are Add/View/Update/Delete. These forms are used to display and update the student records for the duration for which they are in the institute.

This form uses the “master_ledger” table. All accesses are made to this table. All updates are also made to this table.

6.2. The Examiner forms



(● Internal) ○ External

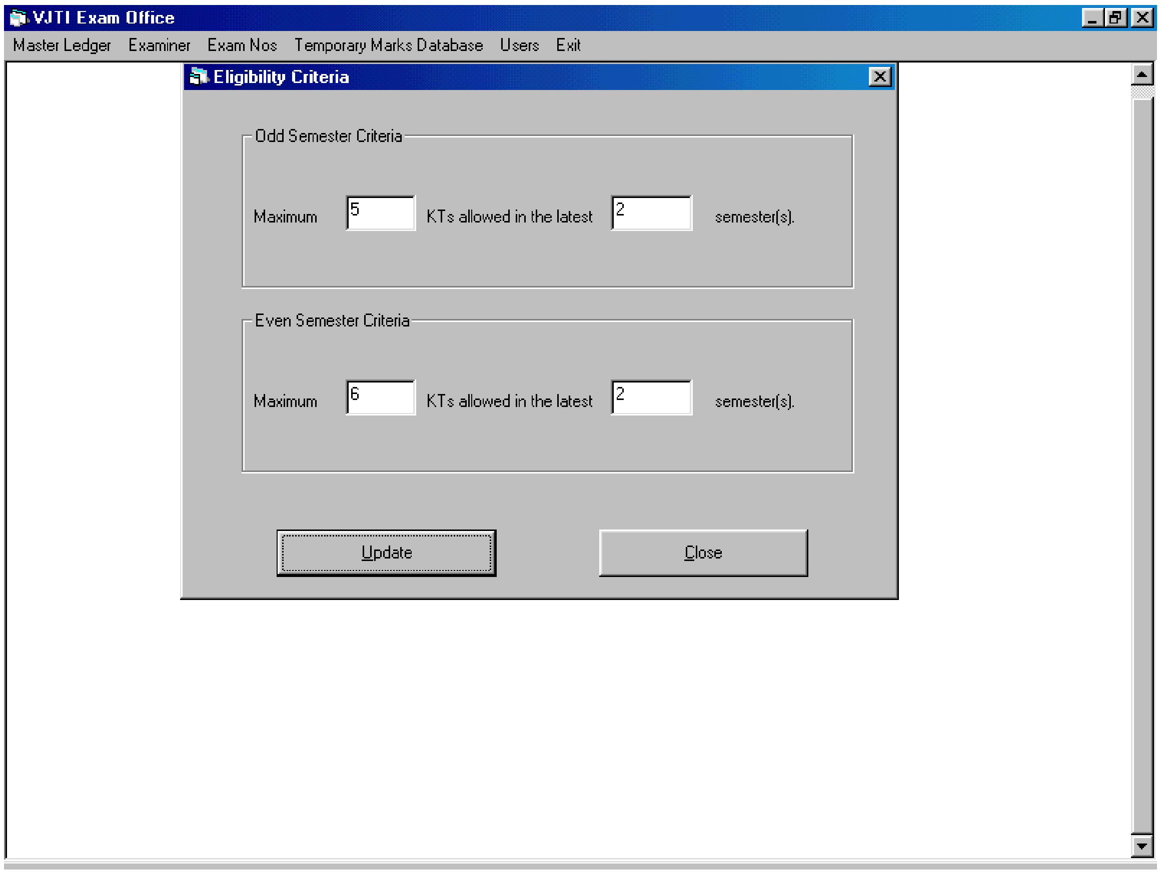
Name	Dr. R.D.Daruwala
Contact Number	4545434
Address	Hostel
Supplementary Information	perfect
College Name	

Record Number 1 Of 17

Sem	Branch	subject_name
5	Computers	Microprocessor I
5	Computers	Microprocessor I

Examiner forms are used to maintain an examiner database. These forms are used to aid the examination section staff in assigning the internal and external examiner for a particular subject. All data about the examiner including his past behavior is stored using these forms. The form also displays a list of all subjects the examiner can correct.

6.3. The Examination number allotment forms



Odd Semester Criteria

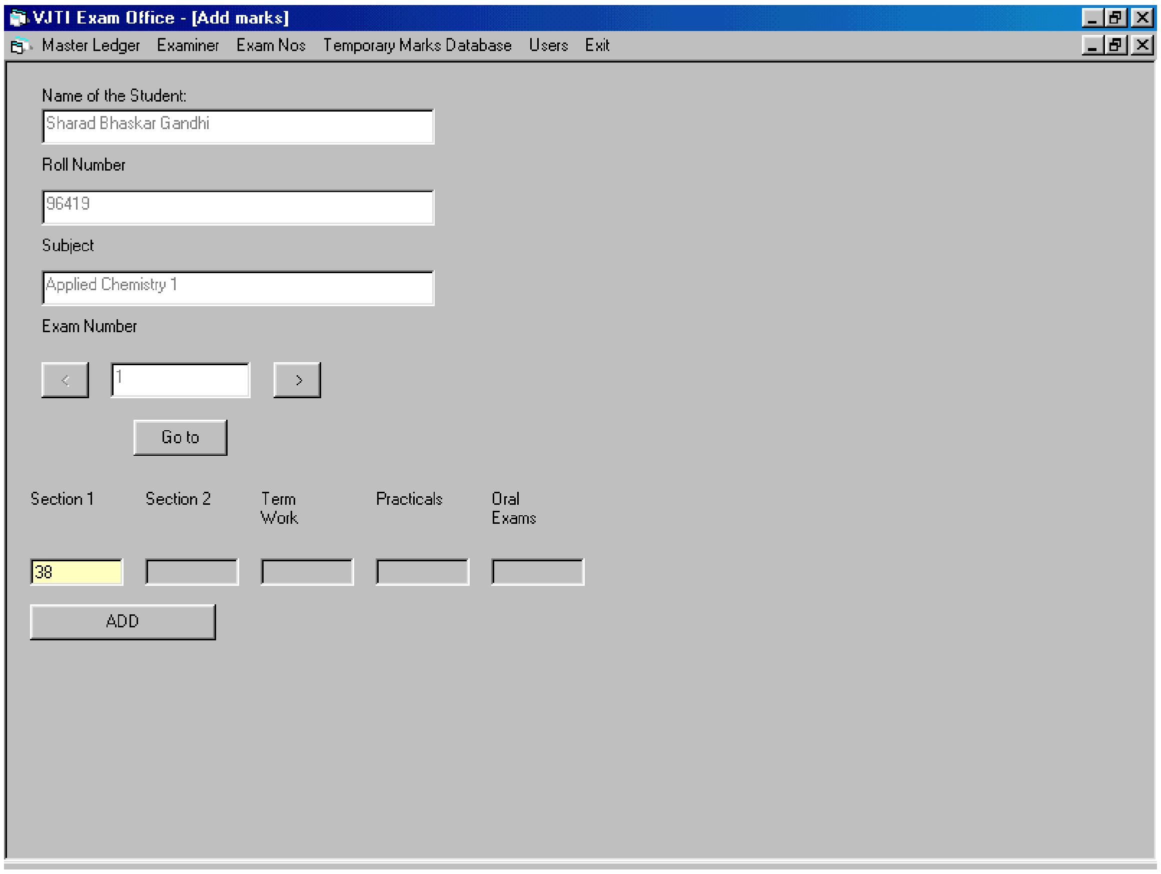
Maximum 5 KTs allowed in the latest 2 semester(s).

Even Semester Criteria

Maximum 6 KTs allowed in the latest 2 semester(s).

These forms are used to allot examination numbers for students appearing for all semesters other than Semester 2, Semester 7 and Semester 8. Examination numbers for the above are allotted by the University of Mumbai. We first generate a list of eligible candidates. This list is generated using eligibility criteria specified by the University of Mumbai. Using the list of eligible candidates and the status of fee payment, a list of examination numbers is generated. The criteria can be changed at any time giving maximal flexibility to the staff.

6.4. The Marks Database forms:



Name of the Student: Sharad Bhaskar Gandhi

Roll Number: 96419

Subject: Applied Chemistry 1

Exam Number: 1

Section 1	Section 2	Term Work	Practicals	Oral Exams
38				

These forms are used to manage the marks database. The user needs to specify the branch, semester, subject and section for which he wants to enter marks. The user enters the marks on the basis of examination numbers, which makes the process quick and easy. These forms also provide an updating feature.

Chapter 7
Application Overview

The user interface is one of the most important parts of any GUI-based system. The exam section application makes use of an MDI (Multiple Document Interface) form from which the user can select any of a number of forms to work with. The options are provided in the form of pull-down menus on the main form.

The main menu items are:

- Master ledger
- Examiner
- Exam Nos
- Marks database
- Exit

The Master Ledger menu consists of the following items:

1. Add – Add a record to the master ledger
2. View – View the existing records
3. Update – Update the master ledger records
4. Delete – Delete an existing record

After selecting any of these options, a form will be displayed to perform the required functions.

The Examiner menu consists of the following items:

1. Add – Add a new examiner into the database
2. View – View the existing examiners
3. Update – Update the records of examiners

The Exam No menu consists of the following items:

1. Modify eligibility criteria – This module updates the eligibility criteria
2. Generate eligible list – Generate the list of eligible candidates

3. Fee payment – User can modify the fee payment status
4. Allot exam numbers – Eligible students who have paid fees are allotted exam numbers

The temporary marks database menu consists of only one entry:
Manage – This module is used to add, view and update sectionwise marks.

7.1. Working with the application

The forms will typically be used in the following order:

1. The master ledger forms will be used for initial entry of student records into the database.
2. The examiner details will be entered through the examiner menus.
3. For every semester, exam numbers are generated using the Exam No menu
4. The marks are entered into the temporary marks database

Chapter 8
Conclusion

In the preceding chapters, we analyzed the working of the VJTI Examination section and identified some of its drawbacks. Along with this, we also proposed a computerized system for the working of the examination section. MS Access was chosen as the platform for modeling the exam section database, while Visual Basic was selected for creating the user interface.

All the forms and most of the registers have been computerized and a menu-based application has been developed. We anticipate that the use of this application will cut down the work of the exam section staff significantly.

Appendix

Supervisor's report

Preformatted letter

Covering letter

Mark Sheet

Branch and semester:

Exam held in _______ 19____

Subject   Max: 100

Min: 40

Seat No.	Question No.												Section		Grand Total Out of 100
	1	2	3	4	5	6	7	8	9	10	11	12	I	II	

Total number of examinees ________

Examiner's Sign

____

Date __________

Resolution form

Bill

Bibliography

[1] James A. Senn, Analysis and System Design, McGraw Hill Publication

[2] Henry Korth and Abraham Silberschatz, Database System Concepts, McGraw Hill Publication (Third Edition), 1998.

[3] Roger S. Pressman, Software Engineering: A Practitioner's Approach, McGraw Hill Publication (Fourth Edition), 1992.

[4] Peter Norton, Peter Norton's Guide to programming in Visual Basic

[5] Michael Amundsen and Curtis Smith, Database Applications using Visual Basic, Sams Publication (First Indian Edition), 1997.