# Palestine Polytechnic University

College of Engineering and Technology
Electrical & computer Engineering Department

Software Project
Geotechnical Software

Project Team:
Raef Mashaaleh
Naser Shalaldah
Bassam Al-Kawasmeh

Project Supervisor
Dr. Salman Talahmeh
Co-Advisor
Dr. Nabil AL- Joulani

Hebron – Palestine

January, 2005

# CONTENTS

## Chapter Four

## Design



## Chapter Five

## Coding and Implementation



## Chapter Six

## Testing

## Chapter Seven

## Maintenance



## References

## Appendices

# List of Figures

# List of Tables

## Problem Initiation

Geotechnical lab is one of the main labs in the Civil Engineering Department at Palestine Polytechnic University (PPU). This lab contains 13 experiments; each experiment has a specific problem.

The Supervisor of this lab faces a lot of problems in solving these problems in that it takes a lot of time doing calculations, they are error prone, some of them are very difficult to solve and tedious, and don't give the required degree of accuracy.

The idea of this project was put on as a graduation project by Dr. Nabil Al-Julani many times during the last three years.

Thus, there is a need for computer based system to solve these problems. This system should have the capability of reading data measured practically, and follow a specific algorithm to solve the problem and gives the results. The system also should be able of giving the results as a report including plotting when required, and saving these results then printing them when needed.

Based on the previous information, in that there is a need for computer based system to solve these problems. This idea was raised after two meetings with Dr. Talahmeh and Dr. Julani were a number of ideas discussed and evaluated, and after extensive search through the Geotechnical experiment books we decided to adopt this idea and apply it on PPU.

# Chapter One

# Introduction

## Introduction

Computers are used in various engineering fields. For example they are used in Civil Engineering for mechanical drawing, Structural and Transportation Engineering two types of Civil Engineers, one of them uses computers for analyzing and the other uses it to design highways and buildings.

Nowadays, as the development of the life goes on, the computer has enters all fields of life, including the civil engineering. Due to this reason, we determined to design this project to facilitate the work of technicians work instead of traditional calculation.

Further, development of computer techniques is used in solving problems in all fields of engineering technology. Students will make computerized solutions to problems with which they are familiar. They also Use data base management, spreadsheets, etc.

They use such programs as CAD, Computer Aided Drafting, and STAAD,' Structural Analysis and Design.' CAD is used to aid in the oftentimes-tedious task of mechanical drawings by hand. STAAD does the same, but also helps to minimize the many equations and calculations of analyzing a structure.

Civil technicians use the principles and theories of science, engineering, and mathematics to solve technical problems in lab and research, development, manufacturing, sales, construction, inspection, and maintenance. Their work is more limited in scope and more practical. So they can write one program that can solve these problems quite easily, instead of going through each step over and over again by hand. Matlab is program that can do these operations, but it can also easily do matrix operations and make graphs.

The idea of this project was accidentally suggested by the Civil Engineering Department Supervisor when he talked about the problem that faces the civil engineering students and employees in carrying out the geotechnical experiments calculations. After that we adopted the idea and studied it in details, then we searched the internet and found that were many geotechnical software available in the market with cost exceeding 5000$ for some of them.

The idea of this project was brought to our supervisor, and after we discussed it with him, he asked us to search again and give him the final result of search, and finally after searching again and again and illustrating the idea to the instructor he accepted the idea and asked us to continue working on it.

In this Chapter we will present an abstract for all subjects covered in the project. Our main concern is to focus on the benefits of computer usage and give a quick view of the name of experiments that will create the system.

The geotechnical software idea was worked on by a lot of software companies, such as the GeoSoft, Geotechnical Software Services (GSS) and Geotechnical and Geoenvironmental Software (GGS); which brought to the market a lot of programs that could perform one or more geotechnical experiments.

The Geotechnical and Geoenvironmental Software Directory (GGSD) catalogues 1 589 programs in the fields of Geotechnical Engineering, Soil Mechanics, Rock Mechanics, Engineering Geology, Foundation Engineering, Hydrogeology, Geoenvironmental Engineering, Environmental Engineering, Data Analysis and Data Visualization and lists 805 worldwide suppliers and publishers of these programs.

Programs are indexed by program name, program category, operating system and program status. The Directory entries are listed by category. Program publishers and suppliers are indexed by organization name and by country. The Directories of programs, publishers and suppliers are cross-referenced. The Directory also gives 28 links to other WWW sites featuring geotechnical, Geoenvironmental or a related software.

The following table shows some software that perform one experiment on the basis of geotechnical and soil analysis:

| Program Name | Description |
| --- | --- |
| DCSIEVE | Sieve and sedimentation analysis |
| DCCONE | Cone penetration tests |
| DCPRESS | Compression test |
| DCSHEAR | Shear strength test |
| DCPERM | Permeability test |
| Axial & Direct Shear | Axial & Direct Shear evaluate one of laboratory testing |
| Construction Quality Control for Cement Products | Software for reporting the results of compressive and flexural tests on cylinders, cubes and beams |
| Drilling Logs plus Field/Laboratory Testing Software | Includes the drilling log software and all of the field and laboratory testing programs listed above |
| Drilling Logs | Software for producing drilling, test pit and monitoring well installation logs, and two-dimensional profiles |
| Field and Laboratory Testing Software | Grain Size, Atterberg Limits, Soil Classifications, Moisture-Density, Consolidation, California Bearing Ratio , R-Value, Axial & Direct Shear, Field Density Test |

Table 1.1 Geotechnical Software that perform one experiment

These Software's are very expensive to buy; some of them reach $800 for some of the experiments as shown in table 1.2 below. Thus the civil Engineering Department Supervisor had suggested the idea to be worked on by the Computer Systems Engineering students to create software that can perform the geotechnical lab experiments at Palestine Polytechnic University (PPU); the table below shows the price of some geotechnical software's:

| Program Name | Description | Price |
|---|---|---|
| GEOPRO | Geotechnical Design | $795 |
| GEOLOG | Boring log Drafting | $795 |
| GEOCAL | Soil Lab Testing | $595 |
| GEOROK | Rock Mechanics Design | $795 |

Table 1.2 Geotechnical Software's Prices

The Geotechnical software we are working on will include 13 experiments from the geotechnical lab adopted at Palestine Polytechnic University (PPU). These experiments are difficult to manipulate manually take a lot of time, error prone and don't give the required degree of accuracy; These experiments are shown in table 1.3 below:

| Experiment Number | Experiment name |
|---|---|
| Exp no.1 | Water or Moisture Content |
| Exp no.2 | Specific gravity determination using Pycnometer |
| Exp no.3 | Bulk Density determination Using Core Cutter |
| Exp no.4 | Bulk Density determination Using Immersion in water |
| Exp no.5 | Bulk Density determination Using Direct Measurement |
| Exp no.6 | Tri Axial Compression and Shear Test |
| Exp no.7 | Direct Shear Test of Soil |
| Exp no.8 | Liquid Limit Determination Using Casagrand Method |
| Exp no.9 | Field Density Test Using Sand Replacement Method |
| Exp no.10 | Falling head Soil Permeability test |
| Exp no.11 | Constant Head Soil Permeability Test |
| Exp no.12 | Laboratory Compaction Test |
| Exp no.13 | Unconfined Compression Test Of Soil |

Table 1.3 Experiments table

In chapter two, the System Specification was determined including the objectives, Project benefits for users, development team and the society, functional and Nonfunctional Requirements, Project constrains, allocation and trade off, development requirement and cost for hardware, software, humans and others. After that, we analyze the cost benefits, feasibility study, risk evaluation, Project Plan and Scheduling. Finally, it summarizes and gives some recommendations to be continued in the next chapter.

In the Third Chapter, we focus on the software requirements specification. Firstly, the requirement definition explains the nonfunctional, functional, constrains details, System of Data Flow Diagram, Data Dictionary, Database requirements and other requirements.

The last part of this chapter is the requirement analysis, which explained by the viewpoints oriented analysis and system contexts.

Chapter Four describes the Output Design, Input Design, Database Design. Functional Design, validation for each function system structure as architectural design, abstract specification, interface design and data structure design. Here we determine-as required-out SQL and structured design in borrowing system more than the remaining parts in the project.

In Chapter five, Coding and Implementation which include Coding Programming Language, Database System ,Establishment of Development Environment ,Database Creation and Configuration, Coding and Unit Testing .

Testing and Maintenance were discussed in chapters six and seven. We first put the testing plan and then tested the system by taking some cases and applying them to see if it is valid or not. In the Maintenance chapter, we established production environment then ,by migration and deployment, we put the Maintenance Plan.

We hope that we will get our goals in this work and if so, it will be the help of the Allah, otherwise it will be our own mistake.

# Chapter Two
# System Specification

2.1 Introduction

2.2 Objectives

2.3 Benefits

2.4 Functional Requirements

2.5 Non-Functional Description

2.6 Project Constrains

2.7 Allocation and Trade-off:

2.8 Development Requirements and Cost

2.9 Cost Benefits Analysis:

2.10 Feasibility Study

2.11 Risk Evaluation

2.12 Plan and Scheduling

2.13 Summery and Recommendations

# System Specification

## 2.1 Introduction

Geotechnical lab has a number of experiments that need a lot of time to do calculations and plot graphs, which are difficult to do manually.The project aims to create a software that would enable the user to enter values measured practically and get the results needed directly.

The system specification was determined after two meeting with the Civil Engineering Departement Supervisor, which included a quick overview of the experiments that performed in the geotechnical lab, These meetings illustrate our vision about the system specification.

In addition to the previous meeting we read many books about the specified topic, and made a search over the internet that provided us with a lot of information about the geotechnical lab and software's available in the market that encouraged us to adopt the idea.

Finally, this provided us an overview of the general specification of this project which can be summarized by many points about the main objectives,benifts, functional and non-functional requirment, project constrains, development requirment and cost, cost benefit analysis, feasibility study, risk evaluation and project plan and scheduling.

## 2.2 Objectives

These are the main objectives of the project:

To provide windows application for 13 Geotechnical experiments mentioned in table 1.1

- To enable students and supervisors to do experimental calculations vary fast way compared to that did manually.
- To enable plotting results of experiments graphically to and show it on screen
- To save experimental results in Data Base (DB) and retrieve later.
- To Provide the steps that are required to recover from an incomplete or broken lab solution
- To provide help screen for each experiment.

## 2.3 Benefits

The project has a number of benifts

- **For users**
  1. more easy to do calculations
  2. more accurate
  3. more time saving
  4. more safety to save results than papers archeifing
  5. more easy to retrieve information later
- **For the development team**
  1. It enhances team experience in programing
  2. It increases team information's about project topics
  3. It complete the Software project requirment
- **For the Society**
  1. It provides a Software for Geotechnical Lab whith low price relative to that in markets ($ 2500 )
  2. It saves analysis results for some areas that can be used later on

## 2.4 Functional Requirments

This project contains many functions. In this section the main functions and a description for each will be provided:

- **Login**

  This function enables the user to enter the program

- **Create User:**

  This function enables the administrator to create new program user for the new students.

- **Retrieve results**

  Allows retrieving information about some experiments done previously.

- **Help**

  Gives the user main informations about the program

- **Save**:

  Allow the user to save the results of experiments.

- **Exit**:

  Allows the user to exit from the program.

- **Perform exp1 (Water or moisture Content):**

`Allows the user to enter values that are required for experiment1 (the weight of empty container, the weight of container and sample of soil, weight of container and dry soil) and then gives the user the opportunity to view the results (water or moisture content (w %)), save them and print them. Also it gives the user some help in performing experiment 1

- **Perform exp2 (Specific Gravity of Soil Using the Pycnometer)**

Allows the user to enter values that are required for experiment2 (weight of empty container, weight of container and sample of soil, weight of container and soil sample and water, weight of container filled just with water) and then gives the user the opportunity to view the results(Specific Gravity of Soil), save them and print them. Also it offers the user some help concerning performing experiment 2.

- **Perform exp3 (Bulk Density determination using the Core Cutter)**

Allow the user to enter values that are required for experiment 3 (diameter of ring or cylinder, height of ring or cylinder, the weight of empty ring or cylinder, the weight of empty ring or cylinder and the soil and two pieces of glass, weight of one piece glass, weight of the other piece glass) and then give the user the opertunity to view the results (Bulk D ensity), save them and p rint t hem. A lso it o ffers the user s om h elp concerning performing experiment 3.

- **Perform exp4 (Bulk Density Determination Using Immersion in water)**

Allow the user to enter values that are required for experiment 4(sample weight, sample weight covered with paraffin, sample weight tipped in water density of paraffin.) and then give the user the opportunity to view the results (Bulk Density), save them and print them. Also it gives the user some help regarding performing experiment 4

- **Perform exp5 (Bulk Density determination by direct measurement)**

Allow the user to enter values that are required for experiment 5 (sample weight, length of container, container width, container height) and then gives the user the opertunity to view the results(Bulk Density), save them and print them .When it gives the user some help regarding performing experiment 5

- **Perform exp6 (Triaxial Shear Test)**

Allow the user to enter values that are required for experiment 6 (Initial length, Sample diameter, Initial Sample Area, Vertical Shear at any time, Vertical Pressure at failure) and then give the user the opertunity to view the results (percentage of Stress, cohesion, angle of internal friction, shear stress), save them and print them .Then it gives the user som help concerning performing experiment 6

- **Perform exp7 (Direct Shear Test of Soil)**

Allow the user to enter values that are required for experiment 7 (vertical stress on the sample, Surface Area, time) and then give the user the opertunity to view the results (Vertical stress, Shear Stress, cohesion, angle of internal friction), save them and print them. Besides it gives the user som help in performing experiment 7

- **Perform exp8 (Liquid Limit Determination Using Casagrand Method)**

Allow the user to enter values that are required for experiment 8 (Number of Hits, Number of Sample (Trial), Number of container, Weight of wet sample and container, Weight of dry sample and container, Weight of container) and then give the user the opertunity to view the results (Average Plastic Limit, Liquid Limit), save them and print them. Moreover it gives the user some help in performing experiment 8.

- **Perform exp9 (Field Density Test Using Sand Replacement Method)**

Allow the user to enter values that are required for experiment 9 (Sample Number, weight of wet soil taken from the hole, weight of dry soil taken from the hole, weight of empty cone, weight of sand that fills the hole and cone, weight of sand that fills the cone, Sand density) and then give the user the opertunity to view the results (wet soil density, dry soil density), save them and print them .It gives the user som help in performing experiment 9.

- **Perform exp10 (Falling Head Soil Permeability Test)**

Allow the user to enter values that are requird for experiment 10 (Sample Number, Sample Length, Time, Height before, Height 2, Area of Container, Area of glass cross section) and then give the user the opertunity to view the results (Permeability coefficient), save them and print them. It gives the user som help in performing experiment 10.

- **Perform exp11 (Constant Head Soil Permeability Test)**

Allow the user to enter values that are required for experiment 11 (Sample Number, amount of filtered water from the sample, Sample Length, cross sectional area, the time required for leak, difference between water height in both tubes) and then give the user the opertunity to view the results (Permeability coefficient), save them and print them. Also it gives the user some help in performing experiment 11

- **Perform exp12 (Laboratory Compaction Test)**

Allow the user to enter values that are required for experiment 12 (Sample Number, weight of cylinder container and base, weight of cylinder container and base and the soil inside it, volume of container, moisture content.) and then give the user the opertunity to view the results (wet soil density dry soil density), saves and print them. Also it gives the user some help in performing experiment 12

- **Perform exp13 (Unconfined Compression Test of Soil)**

Allow the user to enter values that are required for experiment 13 (Sample Number, Compression Power, Initial cross sectional area, Initial Height , difference in height) and then give the user the opertunity to view the results (Stress coefficient, Strain), save them and print them .Also it gives the user som help about performing experiment 13.

## 2.5 Non-Functional Description

Non-functional requirements define system properties and constraints.

- Easiness of use: the system must provide the user a friendly interface to make it easy to deal with.
- Integrity: the system must be integrated with the existing systems and databases.
- Accuracy: the system must provide a high level of accuracy in applying the suitable algorithm for each experiment.
- Speed: The system must follow the best algorithms to get the best complexity which leads to faster program execution.
- High efficiency: the system should be as efficient as possible.
- High reliability: the system means time to failure and probability of unavailability must be low.
- Easiness of use: the system should be easy to use and provide help frames for users.
- High robustness: the system should not be halted when an error occurs.
- High portability: the percentage of target dependent statements number of target system must be low.
- The system and its documentation must be delivered by January, 5, 2005.
- Ethical requirements: the system must follow the esthetical transactions in Palestine.
- Security: the system must maintain its data as secure as possible, so no unprivileged user can access it.

## 2.6 Project Constrains

This section lists the constraints that exist in the project as well as a description of each:

- UserID: there must be six characters at least and less than 20 with no special characters.
- Password: there must be six characters at least and less than nine with no special characters.
- No user can create other user or change his password except through the administrator.
- Create User: No one can create a new user except the administrator who has the valid administrator's user name and password.
- Print: it requires that an experiment has already been done and is currently viewed on the screen.
- Save: an experiment must be previously done.
- Retrieve (search) results: the Data types of the experiment the user looks for must be the same as they are in the Database.
- Each experiment should provide help frames on how to implement the experiment.
- The program shouldn't exit directly when pressing Exit button, it should ask the user again if he wants to exit.
- Each experiment should provide the capability to save and print results.
- The Direct Shear Test of Soil experiment should have the capability to plot a graph between Vertical stress and Shear Stress
- The Liquid Limit Determination Using Casagrand Method experiment should have the capability to plot a graph between Numbers of hits and Water Content.
- The Laboratory Compaction Test experiment should have the capability to plot a graph between dry soil density and moisture content.

- Each experiment should have a button to go to main screen.
- These experiments must be implemented by a high level visual programming language such as VB.6, VB.NET, Java … etc.

## 2.7 Allocation and Trade-off:

The idea under investigation is so tedius, time consuming, error proned, so it is very hard to manipulate it manually.

The problem we are to solve needs a software to implement it rather than hardware tools.

| Function | Hardware | Software | Manual |
|---|---|---|---|
| Login | | SW | |
| Create User | | SW | |
| Print Experiment | | SW | |
| Save Experiment | | SW | |
| Help | | SW | |
| View Result | | SW | |
| Retrieve | | SW | |
| Perform Experiment | | SW | |
| Enter Values | | | Manual |

Table 2.1 Allocation and Trade-off

## 2.8 Development Requirments and Cost

This section lists all costs needed to implement this project:

**Hardware**

| Item | No. of units | Unit Cost | Subtotal |
|---|---|---|---|
| Desktop computers P4, 2 GHz, 512 MB RAM, 20G HD | 1 | $500 | $500 |
| Total | | | $500 |

Table 2.2: Development Hardware Cost

**Software**

| Item | Number of units | Unit Cost | Subtotal |
|---|---|---|---|
| VB6.0 | 1 | $ | 75 |
| Microsoft Word and PowerPoint | 1 | $ | 50 |
| SQL Server 2000 | 1 | $ | 50 |
| Total | | | $175 |

Table 2.3: Development Software Cost

**Humans**

| Member | Number | Cost $ | Subtotal |
|---|---|---|---|
| Engineering Students | 3 | 400 | 1200 |
| Supervisor | 1 | 800 | 800 |
| Total | | | $2000 |

Table 2.4: Development Human Cost

**Books**

| Book Name | Cost $ | Number | Subtotal $ |
|---|---|---|---|
| VB6.0 | 0 | 1 | 20 |
| Database fundamentals | 0 | 1 | 30 |
| Total | | | $50 |

Table 2.5: Cost Benefits Analysis

**Others:**

There is another $200 to cover other areas (transportation, papers, pens…etc).

The following table summarizes the development cost.

| | Total |
|---|---|
| HW | Available |
| SW | Available |
| Humans | Available |
| Books | Available |
| Others | 200 |
| Total | $200 |

Table 2.6: Development Cost Summary:

## 2.9 Cost Benefits Analysis:

The project has a number of benefits from the cost aspect

- **Non tangible benefits:**

1. It will provide the civil engineering departement with software that performs the geotechnical lab calculations and plotting graphs, saving and printing results.
2. It will approve our work team capabilites
3. It is a request of software project course requirements

## 2.10 Feasibility Study

This section will discuss some alternatives, resources and risk evolution that may arise during the project development process.

1. Technical Feasibility:

   This project requires good programming capabilities and methods, an experience in designing windows applications and in dealing with the database (build the tables, interact through the application, security issue ...etc).

   All these capabilities are available in the work team, who has considerable experience in many programming languages such as C, Visual basic 6.0, Java, and VB.NET ...etc, and in database applications such as access and oracle.

2. Legal Feasibility:

   In the regular situation there are no limitations or policies in building such technical projects, so we do not need to take a license to implement this project, and there are no illegal consequences.

3. Economic Feasibility

   The cost of this project discussed in the development requirement and cost section can be coverd by the work team. So, the work team is capable to cover the project cost.

## 2.11 Risk Evaluation

This section contains the risks that may appear in the project and the possible solutions:

- Civil engineering dissatisfaction:

  The project should be built according to civil engineering departement requirements, through there is a fear that the civil engineering departement administrator's dissatisfaction in the production.

  The work team will treat the requirement one by one to accomplish them; in spite of that, and because this project is a graduation project, it is not a big problem if it is not adopted.

- Shortage of development time:

  The time that specified to develop such a project is not enough as we hope to.

  The project team divided the course time in a schedule to finish this project at the specified time.

- Hardware failure:

  To avoid any failure in the hardware and to keep the project, a regular base backup will be made each two days on a different hard disk.

- In case of university closure:

  The work team live in Hebron city in a closed area so that we can meet regardless of any close of University or a siege on the city, an urgent plan is put down to face such a situation. Also, we will communicate with the supervisor directly at his home.

- In case of illness :

  In such a condition other members of the team have to increase their efforts so that they can substitute that member.

## 2.12 Plan and Scheduling:

The work team has scheduled the work in this project to accomplish it within this semister (before Jan. 5th 2005) as shown in figure 2.1 below

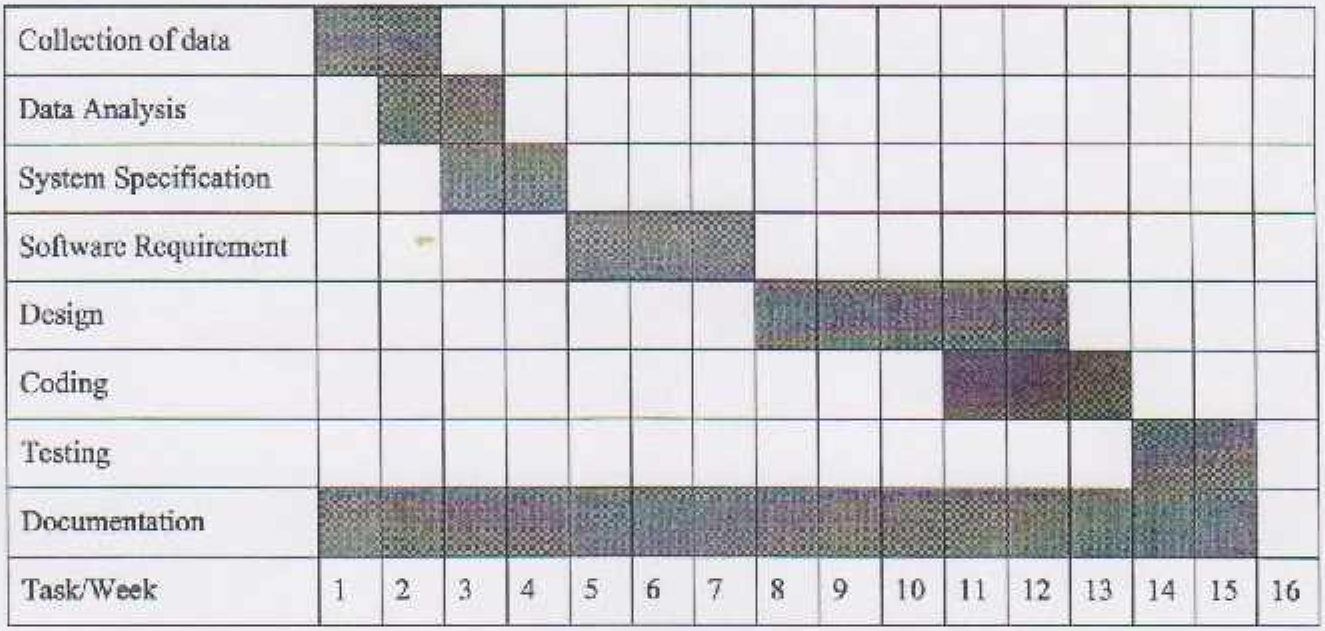


Figure 2.1 Scheduling

## 2.13 Summery:

In this chapter, we have determined the system specifications including objectives, benefits, functional and non-functional requirments, project constrains, allocation and trade off, development requirments and cost, cost benefit analysis, feasibility study, risk evaluation and project plan and scheduling.

These specifications were discussed with the advisor Dr. Salman Al-Talahma and advisor Dr. Nabil Al-Julani, who both agreed on the top level specification of the proposed system.

The next chapter will discuss the Software Specification of the system including the requirement definition that explains the functional, nonfunctional, constrains details, System Data Flow Diagram, Data Dictionary, Database requirements and additional requirements depending on the information collected in this chapter.

# Chapter Three

# Software Requirement Specification

## 3.1 Introduction

In this section the software specifications will be addressed and identified in more technical terms, so this stage will be the portal to system design.

In this section we will cover:

- Functional description of our system, in which all the supported functions and services will be identified and modeled.
- Validation criteria: a set of standards to which our system will be referenced for evaluation.
- Behavioral models in which the behavior of the system will be modeled using a data flow diagram.
- Data dictionary in which a complete description of each system entity will be provided, its name, type, input and output.
- Database requirements in which the required features of the database will be identified.
- Database model which is an abstract modeling of the system database.

## 3.2 Functional Description:

This section lists the major functions in the project and a description for each.

**Function:** Login

**Description:** This Function allow the user to enter and use the Program

**Input:** UserId and Password

**Source:** Windows Form

**Output:** Open the application program

**Destination:** Database

**Require:** Nothing

**Precondition:** valid information provided by the user

**Post condition:** user enter input to get the output

**Procedure:** This Program will check the user name and password in the Database if not valid an error message will be given; else the program will open the main page.

**Validation:** UserID should be between 6 and 20 characters with no special character Password should be between 6 and 8 characters with no special character.

**Function:** Create User

**Description:** This Function allow the Administrator to Create a new user

**Input:** Administrator user name and password

**Source:** Windows Form

**Output:** Create new user

**Destination:** Database

**Require:** Nothing

**Precondition:** valid Administrator user name and password

**Post condition:** administrator enter UserID and password to create new user

**Procedure:** This Program will check the Administrator user name and password in the Database if not valid an error message will be given; else the program will open the Create User Form

**Validation:** UserID should be between 6 and 20 characters with no special character Password should be between 6 and 8 characters with no special character.

**Function:** Exit

**Description:** This Function allow the user to end the Program

**Input:** Nothing

**Source:** Windows Form

**Output:** End program

**Destination:** Nothing

**Require:** Nothing

**Precondition:** Nothing

**Post condition:** user closes the program.

**Procedure:** This function will give the user message box whether he want to exit or not. if yes then exit, else no action.

**Validation:** No validation required

**Function:** Save

**Description:** this function enables the user to save results of Exp.1 on the Database.

**Input:** Results of current experiment.
The location and number of the sample used in this experiment.

**Source:** Windows Form (from the user).

**Output:** message box (The Result had been Saved)

**Destination:** Data Base

**Require:** Experimental results.

**Precondition:** Perform Exp1. and valid information provided by the user.

**Post condition:** Print and search for experiment results.

**Procedure:** this function will take the results of exp.1, and then it will ask the user for the location and number of the sample used in this experiment, after that a validation test will be applied on the inputs, if not valid it gives the user an error message, otherwise data will be saved in the data base, and a notification message to the user to show him that the result had been saved successfully

**Validation:** The location name should be between 6 and 20 characters with no special character.

**Function:** Help

**Description:** this function gives the user helpful information on how the experiment should be performed.

**Input:** Nothing.

**Source:** Data Base

**Output:** Message Box contains the help message

**Destination:** Screen

**Require:** Nothing..

**Precondition:** Nothing

**Post condition:** Nothing

**Procedure:** this function will take stored hints and help statements on exp.1 and display them to the user in a message box.

**Validation:** No validations are required.

**Function:** Print

**Description:** this function enables the user to print results of a specified experiment on the a spreadsheet of paper.

**Input:** Result of exp.1

**Source:** Data Base

**Output:** printing information of exp.1

**Destination:** Printer

**Require:** Perform Exp1, and saved in the Data Base.

**Precondition:** Nothing

**Post condition:** Nothing

**Procedure:** this function will take the results of exp.1, where they are stored in the data base, and print them on a paper.

**Validation:** The print should be after processing.

**Function:** Perform Exp.1 (Water or moisture Content)

**Description:** this function enables the user to calculate the water or moisture content

**Input:** m1: the weight of empty container
m2: the weight of container and sample of soil
m3: weight of container and dry soil

**Source:** Windows Form (from the user).

**Output:** water or moisture content **(w %)**

**Destination:** Variables and arrays

**Require:** nothing.

**Precondition:** valid information provided by the user.

**Post condition:** user enter input to get the output

**Procedure:** this function will first request three inputs (m1, m2, m3) , then it will perform a validation test on this inputs, if the validation doesn't succeeded, an error message will be displayed, else the function will be implemented to give the water or moisture content. The algorithm of this experiment can be found on page 68 in the Soil Testing for Construction Purposes.

**Validation:** m1: the weight of empty container must be float in grams
m2: the weight of container and sample of soil must be float in grams
m3: weight of container and dry soil must be float in grams

**Footer:** Display Results, Help, Save, Print, And Main Form.

**Function:** Perform Exp.2 (Specific Gravity of Soil Using the Pycnometer)

**Description:** this function enables the user to determine Specific Gravity of Soil Using the Pycnometer

**Input:**
m1: the weight of empty container
m2: the weight of container and sample of soil
m3: weight of container and soil sample and water
m4: weight of container filled just with water

**Source:** Windows Form (from the user).

**Output:** Specific Gravity of Soil

**Destination:** Variables and arrays

**Require:** Nothing.

**Precondition:** valid information provided by the user.

**Post condition:** user enter input to get the output

**Procedure:** this function will take the value of m1, m2, m3, m4 and checks if it is valid or not, if not valid then it will notify the user about the error, and then it will return to the form to allow the user re-enter the data again. If the inputs are good and met the validation criteria then this function will apply a specific algorithm on these data and give the user the output. The algorithm of this experiment can be found on page 87 in the Soil Testing for Construction Purposes.

**Validation**: m1: the weight of empty container must be float in grams
m2: the weight of container and sample of soil must be float in grams
m3: weight of container and soil sample and water must be float in grams
m4: weight of container filled just with water must be float in grams

**Footer:** Display Results, Help, Save, Print, Main Form.

**Function:** Perform Exp.3 (Bulk Density determination using the Core Cutter)

**Description:** this function enables the user to determine Bulk Density determination using the Core Cutter

**Input:** D: diameter of ring or cylinder

H: height of ring or cylinder.

m1: the weight of empty ring or cylinder

m2: the weight of empty ring or cylinder and the soil and two pieces of Glass.

m3: weight of one piece glass

m4: weight of the other piece glass

**Source:** Windows Form (from the user).

**Output:** Bulk Density1

**Destination:** Variables and arrays

**Require:** Nothing.

**Precondition:** valid information provided by the user.

**Post condition:** user enter input to get the output

**Procedure:** this function will take the values of D, H and perform a validation on them, if valid calculate the volume and continue performing exp.4, else give the user error message.

After that the function request the user to enter the values of m1, m2, m3, m4 and checks if it is valid or not, if not valid then it will notify the user about the error, and then it will return to the form to allow the user re-enter the data again. If the inputs are good and met the validation criteria then this function will calculate the Bulk Density. The algorithm of this experiment can be found on page 91 in the Soil Testing for Construction Purposes.

**Validation:** D: diameter of ring or cylinder must be float in cm

H: height of ring or cylinder: must be float in cm

m1: the weight of empty ring or cylinder must be float in grams

m2: the weight of empty ring or cylinder and the soil and two pieces of Glass: must be float in grams

m3: weight of one piece glass must be float in grams

m4: weight of the other piece glass must be float in grams

**Footer:** Display Results, Help, Save, Print, Main Form.

**Function:** Perform Exp.4 (Bulk Density determination using Immersion in water)

**Description:** this function enables the user to determine Bulk Density determination using Immersion in water

**Input:** m: sample weight.
m1: the sample weight covered with paraffin.
m2: sample weight tipped in water
Pp: density of paraffin.

**Source:** Windows Form (from the user).

**Output:** Bulk Density2

**Destination:** Variables and arrays

**Require:** Nothing.

**Precondition:** valid information provided by the user.

**Post condition:** user enter input to get the output

**Procedure:** this function will take the values of m, m1, m2, Pp and checks if it is valid or not, if not valid then it will notify the user about the error, and then it will return to the form to allow the user re-enter the data again. If the inputs are good and met the validation criteria then this function will calculate the Bulk Density and print it out on the screen. The algorithm of this experiment can be found on page 96 in the Soil Testing for Construction Purposes.

**Validation:** m: sample weight: must be float in grams
m1: the sample weight covered with paraffin: must be float in grams
m2: sample weight tipped in water must be float in grams
Pp: density of paraffin: must be float in grams/cm

**Footer:** Display Results, Help, Save Print, And Main Form.

**Function:** Perform Exp.5 (Bulk Density determination by direct measurement)

**Description:** this function enables the user to determine Bulk Density by direct measurement

**Input:** m: sample weight.
L: length of container
W: container width
H: container height.

**Source:** Windows Form (from the user).

**Output:** Bulk Density3

**Destination:** Variables and arrays

**Require:** Nothing.

**Precondition:** valid information provided by the user.

**Post condition:** user enter input to get the output

**Procedure:** this function will take the values of the three dimensions H, L and W and then it will perform a validation test on them whether they are valid or not, if not valid then notify the user by an error message, otherwise the function will continue and calculate the volume V and then calculate the bulk density from m and V. The algorithm of this experiment can be found on page 99 in the Soil Testing for Construction Purposes.

**Validation:** m: sample weight. Must be float in grams
L: length of container must be float in mm
W: container width must be float in mm
H: container height. Must be flout in mm

**Footer:** Display Results, Help, Save Print, And Main Form.

**Function:** Perform Exp.6 (Triaxial Shear Test)

**Description:** this function enables the user to determine Triaxial Shear .

**Input:** L0: Initial length.

D: Sample diameter.

A0: Initial Sample Area.

X: Vertical Shear at any time.

P: Vertical Pressure at failure

**Source:** Windows Form (from the user).

**Output:** ε: percentage of Stress

C: cohesion

Φ: angle of internal friction

τ: shear stress

**Destination:** Variables and arrays

**Require:** Nothing.

**Precondition:** valid information provided by the user.

**Post condition:** user enter input to get the output

**Procedure:** this function will take the values of L0 and D and then it will perform a validation test on them whether they are valid or not, if not valid then notify the user by an error message, otherwise the function will continue and calculate A0 and then calculate ε from X and L0 also it will calculate the output vertical pressure (σ1- σ3) , the function will also give a graph between σ and τ and from the curve the user have to find C and Φ. The algorithm of this experiment can be found on page 158 in the Soil Testing for Construction Purposes.

**Validation:** L0: Initial Length: must be float in mm

D: Sample Diameter: must be float in mm

A0: Initial Sample Area.: must be float in mm

X: Vertical Shear at any time: must be float in mm

P: Vertical Pressure at failure must be float in mm

**Function:** Perform Exp.7 (Direct Shear Test of Soil)

**Description:** this function enables the user to determine the Direct Shear of Soil .

**Input:** Ni: vertical stress on the sample

A: Surface Area

τ: shear stress

**Source:** Windows Form (from the user).

**Output:** σi: Vertical stress

τ: Shear Stress

C: cohesion

Φ: angle of internal friction

graph (σi: X τ)

**Destination:** Variables and arrays

**Require:** Nothing.

**Precondition:** valid information provided by the user.

**Post condition:** user enter input to get the output

**Procedure:** this function will take the values Ni, A and τ and then it will perform a validation test on them whether they are valid or not, if not valid then notify the user by an error message, otherwise the function will continue and calculate σi and τ .also give a graph between σ and τ and from the curve the user have to find C and Φ. The algorithm of this experiment can be found on page 154 in the Soil Testing for Construction Purposes.

**Validation:** Ni: vertical stress on the sample must be float in KNewton

A: Surface Area must be float in meter square

T: must be float in KNewton

**Footer:** Display Results, Help, Save Print, Graph, And Main Form.

**Function:** Perform Exp.8 (Liquid Limit Determination Using Casagrand Method)

**Description:** this function enables the user to determine Liquid Limit Using Casagrand Method.

**Input:** **Nh:** Number of Hits

**Ns:** Number of Sample (Trial)

Nc: Number of container

m1: Weight of wet sample and container

m2: Weight of dry sample and container

m3: Weight of container

**Source:** Windows Form (from the user).

**Output:** Average Plastic Limit and Liquid Limit

Graph (Number of hits X Water Content)

**Destination:** Variables and arrays

**Require:** Nothing.

**Precondition:** valid information provided by the user.

**Post condition:** user enter input to get the output

**Procedure:** this function will take the Number of Hits, Number of Sample (Trial), Number of container, Weight of wet sample and container, Weight of dry sample and container and finally the Weight of container and then it will perform a validation test on them whether they are valid or not, if not valid then notify the user by an error message, otherwise the function will continue and calculate Average Plastic Limit and Liquid Limit also give a graph (Number of hits X Water Content). The algorithm of this experiment can be found on page 74 in the Soil Testing for Construction Purposes.

**Validation:** Number of Hits must be Integer

Number of Sample (Trial) must be Integer

Number of container must be Integer

Weight of wet sample and container must be float in gram

Weight of dry sample and container must be float in gram

Weight of container must be float in gram

**Footer:** Display Results, Help, Save, Print, Graph, Main Form.

**Function:** Perform Exp.9 (Field Density Test Sand Replacement Method)

**Description:** this function enables the user to determine Field Density Test Using Sand Replacement Method

**Input:** i: Sample Number
W1: weight of wet soil taken from the hole
W2: weight of dry soil taken from the hole
W3: weight of empty cone
*W4: weight of sand that fill the hole and cone*
*Ps: Sand density*

**Source:** Windows Form (from the user).

**Output:** P: wet soil density, Pd: dry soil density

**Destination:** Variables and arrays

**Require:** Nothing.

**Precondition:** valid information provided by the user.

**Post condition:** user enter input to get the output

**Procedure:** this function will take the weight of wet soil taken from the hole, weigh of dry soil taken from the hole, weight of empty cone, weight of sand that fill the hole and cone, weight of sand that fill the cone and Sand density and then it will perform a validation test on them whether they are valid or not, if not valid then notify the user by an error message, otherwise the function will continue and calculate wet soil density dry soil density. The algorithm of this experiment can be found on page 233 in the Soil Testing for Construction Purposes.

**Validation:** i: Sample Number Hits must be Integer
W1: weight of wet soil taken from the hole container must be float in gram
W2: weight of dry soil taken from the hole container must be float in gram
W3: weight of empty cone container must be float in gram
W4: weight of sand that fill the hole and cone container must be float in gram

**Footer:** Display Results, Help, Save, Print, Graph, Main Form.

**Function:** Perform Exp.10 (Falling Head Soil Permeability Test)

**Description:** this function enables the user to determine Soil Permeability using Falling Head test.

**Input:**
i: Sample Number
L: Sample Length
T: Time
H1: Height before
H2: Height 2
A: Area of Container
a: Area of glass cross section

**Source:** Windows Form (from the user).

**Output:** K: Permeability coefficient

**Destination:** Variables and arrays

**Require:** Nothing.

**Precondition:** valid information provided by the user.

**Post condition:** user enter input to get the output

**Procedure:** this function will take the Sample Length, Time, Height before stress, Height after stress, Area of Container and the Area of glass cross section area, and then it will perform a validation test on them whether they are valid or not, if not valid then notify the user by an error message, otherwise the function will continue and calculate Permeability coefficient. The algorithm of this experiment can be found on page 197 in the Soil Testing for Construction Purposes.

**Validation:** i: Sample Number must be integer
L: Sample Length must be float in cm
T: Time must be float in second
H1: Height before must be float in cm
H2: Height 2 must be float in cm
A: Area of Container must be float in cm square
a: Area of glass cross section must be float in cm square

**Footer:** Display Results, Help, Save Print, Graph, And Main Form.

**Function:** Perform Exp.11 (Constant Head Soil Permeability Test)

**Description:** this function enables the user to determine Soil Permeability using Constant Head test.

**Input:** i: Sample Number

Q: amount of filtered water from the sample

L: Sample Length

A: cross sectional area

t: the time required for leak

h: difference between water height in both tubes.

**Source:** Windows Form (from the user).

**Output:** K: Permeability coefficient

**Destination:** Variables and arrays

**Require:** Nothing.

**Precondition:** valid information provided by the user.

**Post condition:** user enter input to get the output

**Procedure:** this function will take the i: Sample Number, amount of filtered water from the sample, Sample Length, cross sectional area, the time required for leak, difference between water height in both tubes. And then it will perform a validation test on them whether they are valid or not, if not valid then notify the user by an error message, otherwise the function will continue and calculate Permeability coefficient. The algorithm of this experiment can be found on page 200 in the Soil Testing for Construction Purposes.

**Validation:** i: Sample Number must be Integer

Q: amount of filtered water from the sample: must be flout in cm cube

L: Sample Length must be flout in cm

A: cross sectional area must be flout in cm square

t: the time required for leak must be flout in second

h: difference between water height in both tubes: must be flout in cm

**Footer:** Display Results, Help, Save Print, Graph, And Main Form.

**Function:** Perform Exp.12 (Laboratory Compaction Test)

**Description:** this function enables the user to determine Compaction in the laboratory

**Input:** i: Sample Number

m1: weight of cylinder container and base

m2: weight of cylinder container, base and the soil inside it.

V: volume of container

w: moisture content.

**Source:** Windows Form (from the user).

**Output:** P: wet soil density

Pd: dry soil density

graph(Pd X Wc)

**Destination:** Variables and arrays

**Require:** Nothing.

**Precondition:** valid information provided by the user.

**Post condition:** user enter input to get the output

**Procedure:** this function will take the weight of cylinder container and base, weight of cylinder container and base and the soil inside it, volume of container finally it will takes the moisture content. and then it will perform a validation test on them whether they are valid or not, if not valid then notify the user by an error message, otherwise the function will continue and calculate wet soil density and dry soil density. The algorithm of this experiment can be found on page 210 in the Soil Testing for Construction Purposes.

**Validation:** i: Sample Number must be Integer

m1: weight of cylinder container and base must be float in gram

m2: weight of cylinder container, base and the soil inside it. must be float in gram

V: volume of container must be float in $cm^3$

W: moisture content must be float

**Footer:** Display Results, Help, Save Print, Graph, And Main Form.

**Function:** Perform Exp.13 (Unconfined Compression Test of Soil)

**Description:** this function enables the user to determine the shape of the sample at failure

**Input:** i: Sample Number
Pi: Compression Power
A0: Initial cross sectional area
H: Initial Height
ΔHi: difference in height

**Source:** Windows Form (from the user).

**Output:** σ1: Stress coefficient
εi: Strain

**Destination:** Variables and arrays

**Require:** Nothing.

**Precondition:** valid information provided by the user.

**Post condition:** user enter input to get the output

**Procedure:** this function will take the weight of cylinder container and base, weight of cylinder container and base and the soil inside it, volume of container finally it will takes the moisture content. And then it will perform a validation test on them whether they are valid or not, if not valid then notify the user by an error message, otherwise the function will continue and calculate wet soil density and dry soil density. The algorithm of this experiment can be found on page 163 in the Soil Testing for Construction Purposes.

**Validation:** i: Sample Number must be Integer
Pi: Compression Power: must be float in KNewton
A0: Initial cross sectional area: must be float in cm square
H: Initial Height: must be float in mm
ΔHi: difference in height must be float in mm

**Footer:** Display Results, Help, Save, Print, Graph, Main Form.

### 3.3 Project Constraints Details

This section will explain in details all project constraints:

A. Login:

- UserID: must be provided by the administrator so that it should contain the student name up to 20 characters with no special character.
- Password: must be provided by the administrator so that it should contain the student name up to 8 characters with no special character

B. Create User: the users are constrained to the UserID and password given by the administrator, so that no user can create other users or change his password only through the administrator.

C. Print: it requires that an experiment is previously done and is currently viewed on the screen; this function must be disabled while the results are not ready.

D. Save: an experiment must be previously done and results are viewed, also the user should give valid parameters for experiment name, location of the experiment.

E. The program shouldn't exit directly when pressing Exit button, it should ask the user again if he want to exit.

F. Retrieve (search) results: the Data types of the experiment the user looks for must be the same as they are in the Database, this should provide the user with the required information about that experiment.

G. Help frames: must be provided for all experiments, and must contain a clear steps on how to implement each one; also it give the user all the input parameters and data type for each one.

H. The Direct Shear Test of Soil experiment should have the capability to plot a graph between Vertical stress and Shear Stress, the critical point on the graph must be obviously displayed.

I. The Liquid Limit Determination Using Casagrand Method experiment should have the capability to plot a graph between Numbers of hits and Water Content.

Then the critical point on the graph must be obviously displayed.

J. The Laboratory Compaction Test experiment should have the capability to plot a graph between dry soil density and moisture content. Then the critical point on the graph must be obviously displayed.

K. Each experiment should have a button to go the user to main screen when he clicks on it, also exit.

L. These experiments must be implemented by using a high level visual programming language such as VB.6, VB.NET, Java ... etc. and the program should have a good interface.

## 3.5 System Data Flow Diagram

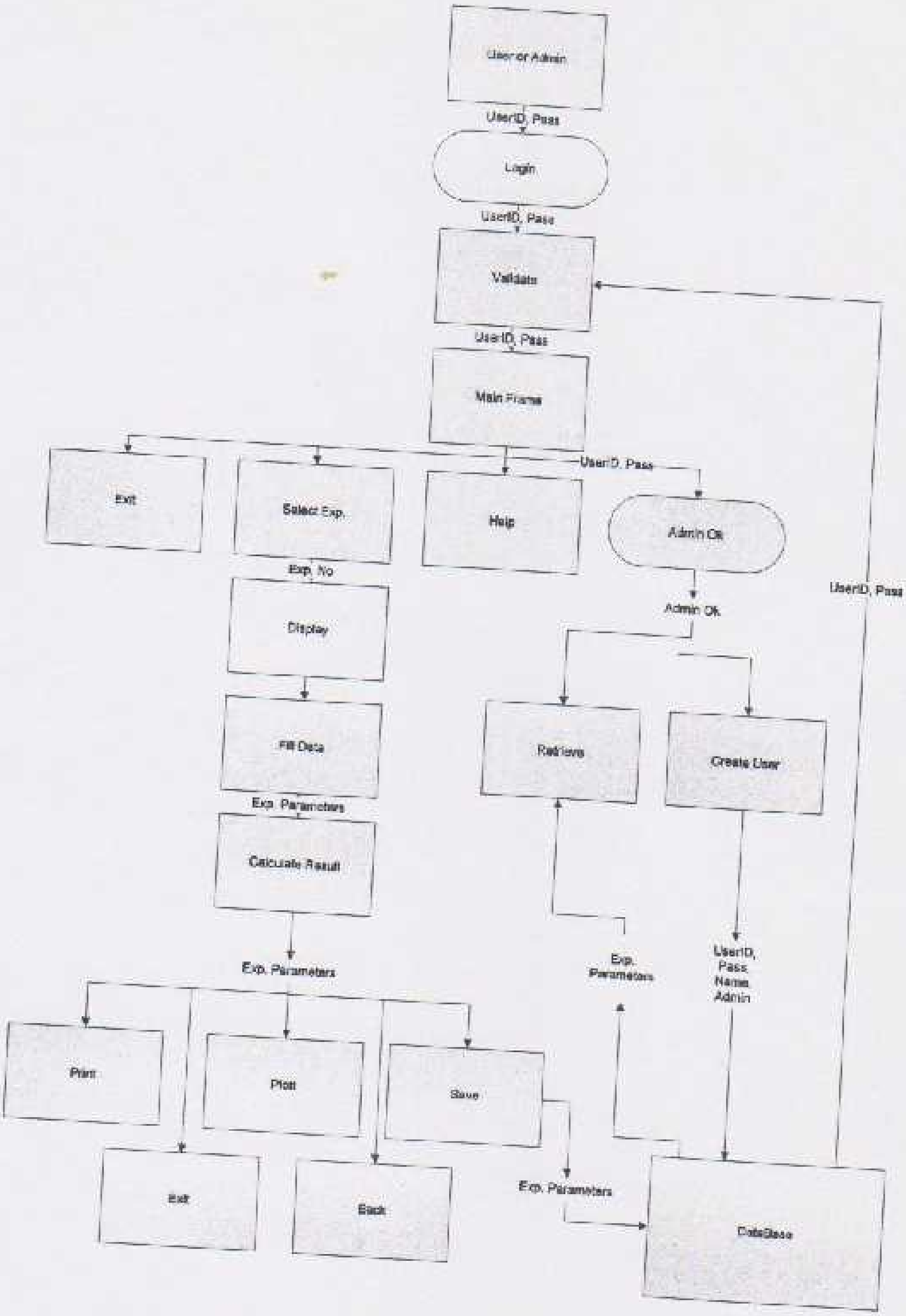


Figure 3.1 System Data Flow Diagram

## 3.5 Data Dictionary:

Here we will show the entity name (function) with description of what this function will do, as shown in the following table:

| Entity name | Type | Description |
|---|---|---|
| Login | Function | This function enables the user to enter the program |
| Help | Function | Gives the user main informations about the program |
| Save | Function | This function enables the user to Save the experimental results on the Data Base |
| Retrieve results | Function | Allow retrieving information about some experiments done previously |
| Print | Function | This function enables the user to Print the experimental results on paper |
| Exit | Function | This function enables the user to exit from the programe |
| Perform Exp.1 (Water or moisture Content | Function | this function enables the user to calculate the water or moisture content |
| Perform Exp.2 (Specific Gravity of Soil Using the Pycnometer | Function | this function enables the user to determine Specific Gravity of Soil Using the Pycnometer |
| Perform Exp.4 (Bulk Density determination using Immersion in water) | Function | this function enables the user to determine Bulk Density determination using Immersion in water |
| Perform Exp.5 (Bulk Density determination by direct | Function | this function enables the user to determine Bulk Density by direct measurement |

| | | |
|---|---|---|
| measurement) | | |
| Perform Exp.6 (Triaxial Shear Test) | Function | this function enables the user to determine Triaxial Shear |
| Perform Exp.7 (Direct Shear Test of Soil) | Function | this function enables the user to determine the Direct Shear of Soil |
| Perform Exp.8 (Liquid Limit Determination Using Casagrand Method) | Function | this function enables the user to determine Liquid Limit Using |
| Perform Exp.9 (Field Density Test Using Sand Replacement Method) | Function | this function enables the user to determine Field Density Test Using Sand Replacement Method |
| Perform Exp.10 (Falling Head Soil Permeability Test) | Function | this function enables the user to determine Soil Permeability using |
| Perform Exp.11 (Constant Head Soil Permeability Test) | Function | this function enables the user to determine Soil Permeability using Constant Head test. |
| Perform Exp.12 (Laboratory Compaction Test) | Function | this function enables the user to determine Compaction in the laboratory |
| Perform Exp.13 (Unconfined Compression Test of Soil) | Function | this function enables the user to determine the shape of the sample at failure |

Table 3.1 Data Dictionary

## 3.6 Database Requirement

This section will determine data to be stored in the data Base:

**Exp_Names:**

- Experiment ID
- Experiment name
- Description

**Users**:

- User ID
- User Name
- User password
- Admin

**Perform Exp.1 (Water or moisture Content)**

- Exp_ID
- Sequential Number
- User ID
- Date
- Notes
- Sample Number
- The weight of empty container
- The weight of container and sample of soil
- weight of container and dry soil
- Water or moisture content (w %)

**Perform Exp.2 (Specific Gravity of Soil Using the Pycnometer**

- Exp_ID
- Sequential Number
- User ID
- Date

- Notes
- Sample Number
- The weight of empty container
- The weight of container and sample of soil
- weight of container and soil sample and water
- weight of container filled just with water
- Specific Gravity of Soil

**Perform Exp.3 (Bulk Density determination using the Core Cutter)**

- Exp_ID
- Sequential Number
- User ID
- Date
- Notes
- Sample Number
- Diameter of ring or cylinder
- Height of ring or cylinder.
- The weight of empty ring or cylinder
- The weight of empty ring or cylinder and the soil and two pieces of Glass.
- weight of one piece glass
- weight of the other piece glass
- Bulk Density

**Perform Exp.4 (Bulk Density determination using Immersion in water)**

- Exp_ID
- Sequential Number
- User ID
- Date
- Notes
- Sample Number
- Sample weight.

- The sample weight covered with paraffin.
- Sample weight tipped in water
- Density of paraffin.
- Bulk Density

**Perform Exp.5 (Bulk Density determination by direct measurement**

- Exp_ID
- Sequential Number
- User ID
- Date
- Notes
- Sample Number
- Sample weight.
- length of container
- Container width
- Container height.
- Bulk Density

**Perform Exp.6 (Triaxial Shear Test)**

- Exp_ID
- Sequential Number
- User ID
- Date
- Notes
- Initial length.
- Sample diameter.
- Initial Sample Area.
- Vertical Shear at any time.
- Vertical Pressure at failure.
- Sample Number
- ε: percentage of Stress
- C: cohesion
- Φ: angle of internal friction
- τ: shear test

**Perform Exp.7 (Direct Shear Test of Soil)**

- Exp_ID
- Sequential Number
- User ID
- Date
- Notes
- Vertical stress on the sample
- Surface Area
- Time
- Sample Number
- σi: Vertical stress
- τ: Shear Stress
- C: cohesion
- Φ: angle of internal friction
- graph (σi: X τ)

**Perform Exp.8 (Liquid Limit Determination Using Casagrand Method)**

- Exp_ID
- Sequential Number
- User ID
- Date
- Notes
- Number of Hits
- Number of Sample (Trial)
- Number of container
- Weight of wet sample and container
- Weight of dry sample and container
- Weight of container
- Sample Number
- Average Plastic Limit
- Liquid Limit

- Graph ( Number of hits X Water Content

**Perform Exp.9 (Field Density Test Using Sand Replacement Method**

- Exp_ID
- Sequential Number
- User ID
- Date
- Notes
- Sample Number
- weight of wet soil taken from the hole
- weight of dry soil taken from the hole
- weight of empty cone
- weight of sand that fill the hole and cone
- weight of sand that fill the cone
- Sand density
- Sample Number
- P: wet soil density
- Pd: dry soil density

**Perform Exp.10 (Falling Head Soil Permeability Test)**

- Exp_ID
- Sequential Number
- User ID
- Date
- Notes
- Sample Number
- Sample Length
- Time
- Height before
- Height 2
- Area of Container
- Area of glass cross section

- K: Permeability coefficient

**Perform Exp.11 (Constant Head Soil Permeability Test)**

- Exp_ID
- Sequential Number
- User ID
- Date
- Notes
- Sample Number
- Amount of filtered water from the sample
- Sample Length
- Cross sectional area
- The time required for leak
- Difference between water height in both tubes.
- K: Permeability coefficient

**Perform Exp.12 (Laboratory Compaction Test)**

- Exp_ID
- Sequential Number
- User ID
- Date
- Notes
- Sample Number
- Weight of cylinder container, base and the soil inside it.
- Weight of cylinder container and base
- Volume of container.
- Moisture content.
- P: wet soil density
- Pd: dry soil density

**Perform Exp.13 (Unconfined Compression Test of Soil)**

- Exp_ID
- Sequential Number
- User ID
- Date
- Notes
- Sample Number
- Compression Power
- Initial cross sectional area
- Initial Height
- Difference in height
- $\sigma 1$: Stress coefficient
- $\varepsilon i$: Strain

## 3.7 Summery

In this chapter we had described in details the functional requirements including function name , describtion , input , source , output , destination , require , precondition ,postcondition ,procedure and validation for each experiment.

The system detailed specifications have been determined and approved by both the supervisor Dr. Salman Talahmeh and Civil Engineering supervisor Dr. Nabeel Jolani.

The general database requirements view was written so that it enables us to continue to the next stage of programming which is the design stage.

# Chapter Four

# Design

# Design

## 4.1 Introduction

In this section the functions design will be implemented using functional o riented methodology, where each function will be designed accordingly.

The following will be covered in this section

- Input output design: a design for the input output screens.
- Database design: a complete design for the database tables, all tables and their fields.
- Objects design: where each function will be designed using a flow chart, its interface and constraints will be identified.
- Validation for each objects using decision tables.
- Summary and recommendations.

## 4.2 Input /Output Design

In this section the input and output will be designed including the interface, constrains and flowchart.

### 4.2.1 Screens:

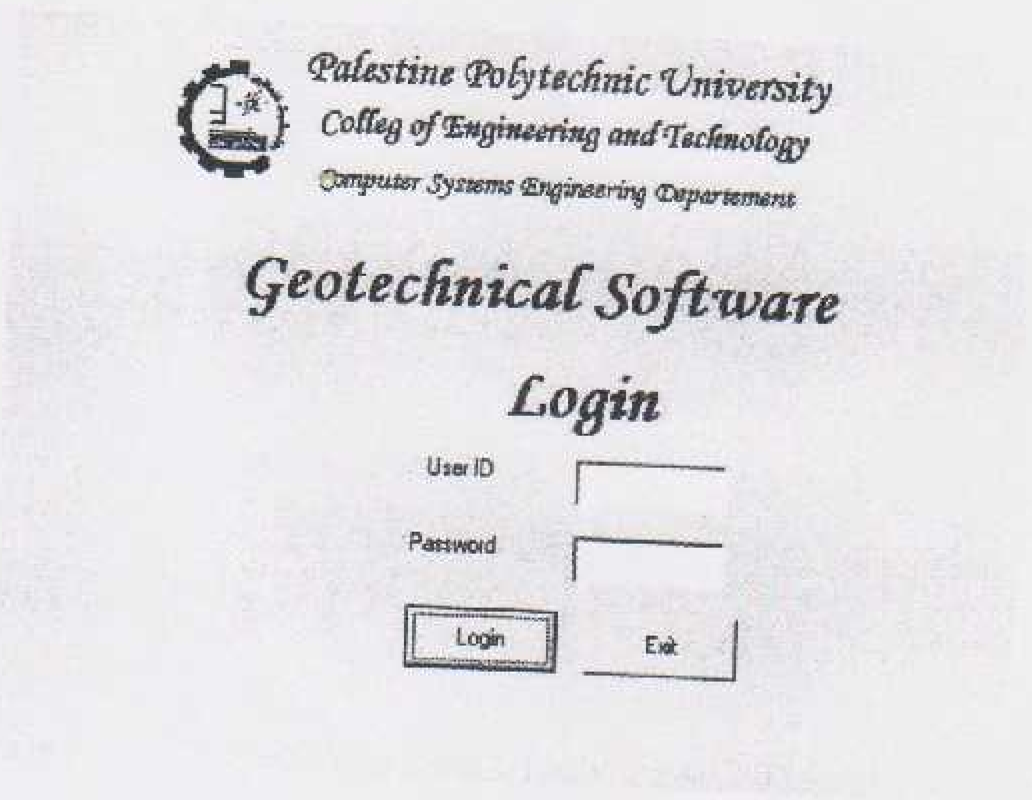


Figure 4.1 Login Screen

Login: This Function allow the user to enter and use the Program

UserID: should be between 6 and 20 characters with no special character

Password: should be between 6 and 8 characters with no special character

Exit: This Function allow the user to end the Program

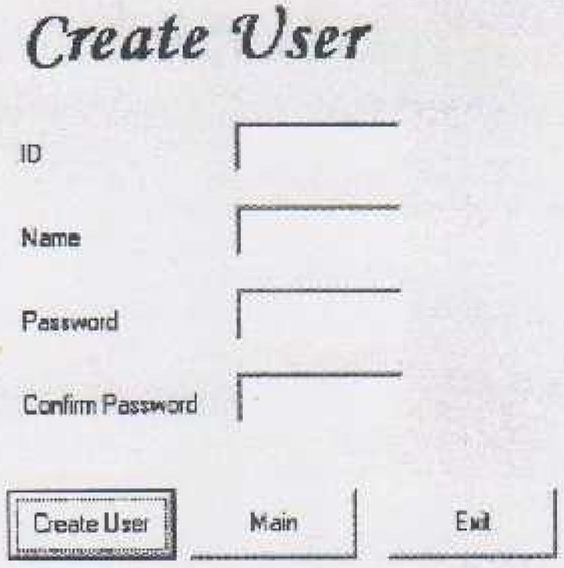


Figure 4.2 Create user Screen

Create User: This Function allow the Administrator to create a new user
ID: should be between 6 and 20 characters with no special character
Password: should be between 6 and 8 characters with no special character.
Exit: This Function allow the user to end the Program

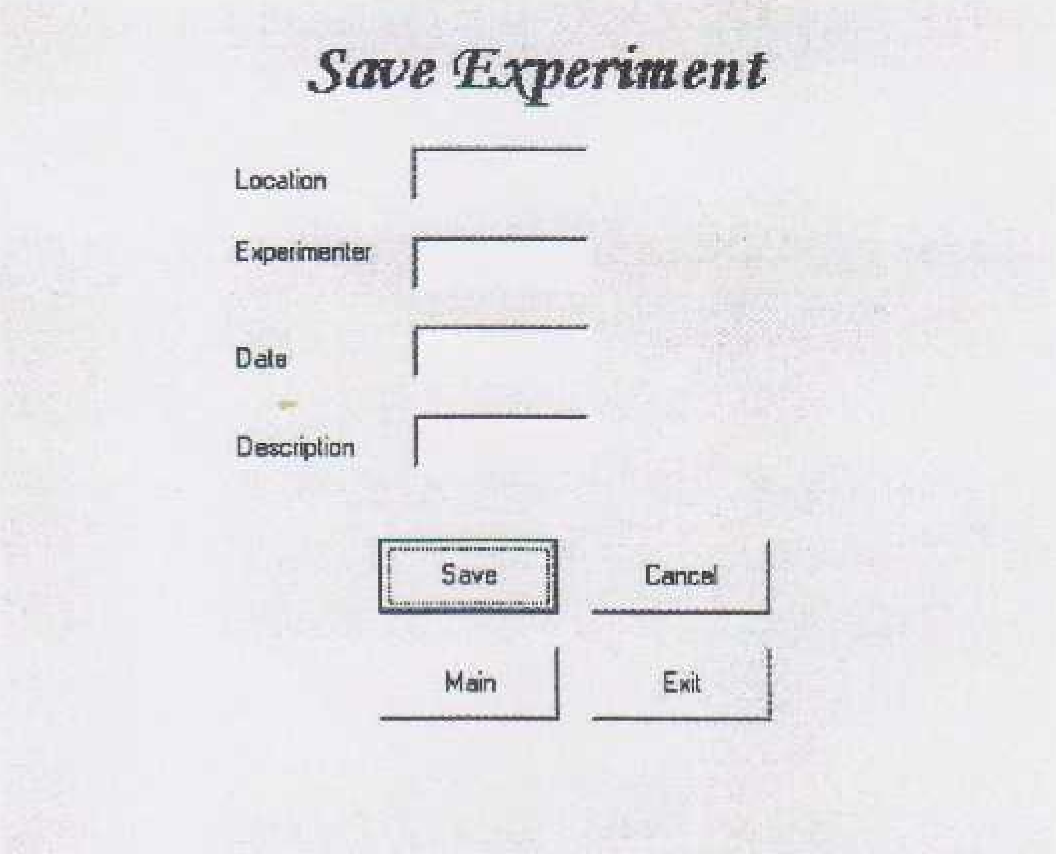


Figure 4.3 Save Experiment Screen

Save: this function enables the user to save results of Exp. on the Database

The location name: should be between 6 and 20 characters with no special character

Exit: This Function allow the user to end the Program

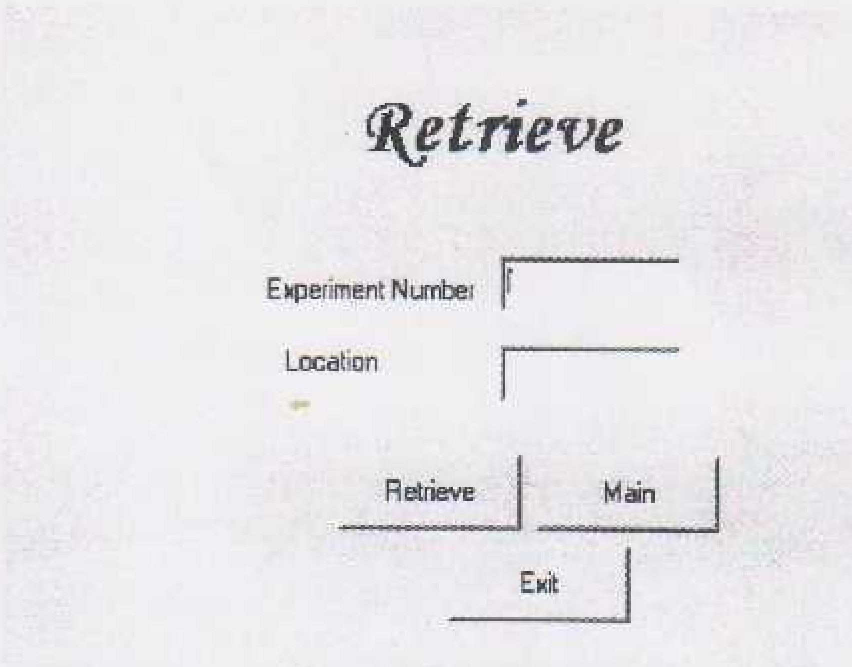


Figure 4.4 Retrieve Experiment Screen

Retrieve: allow the administrator to retrieve result

Experiment Number: should be numerical

Exit: This Function allow the user to end the Program

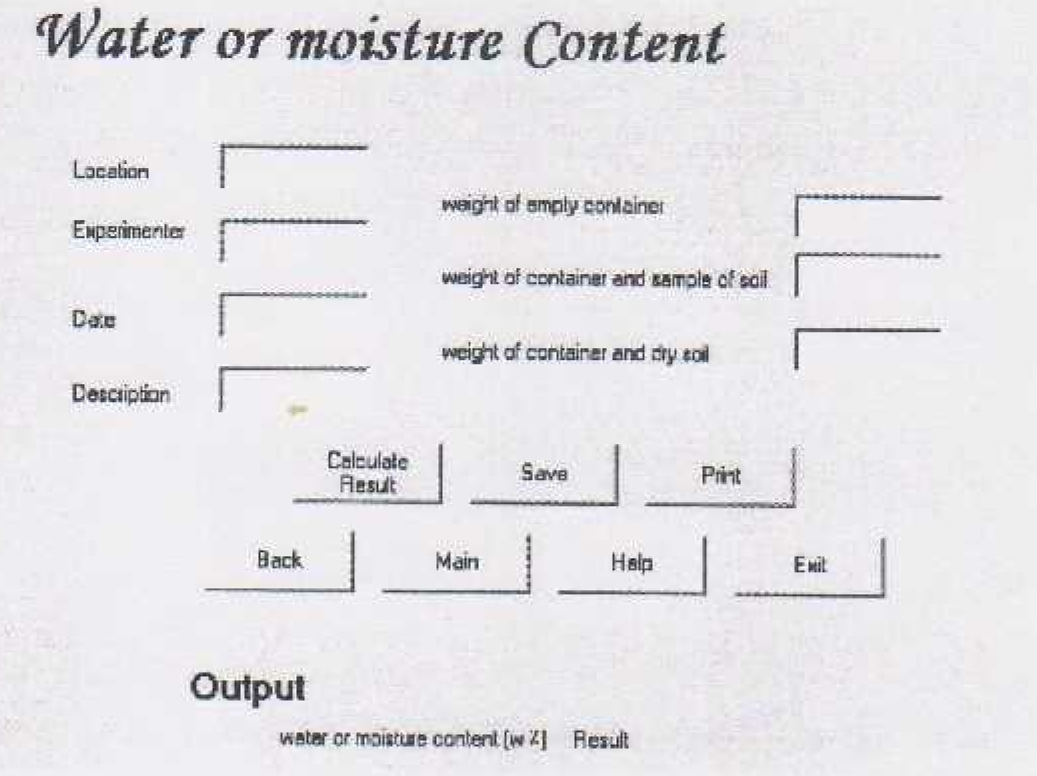


Figure 4.5 Water or moisture Content Screen

Water or moisture Content: this function enables the user to calculate the water or moisture content

The weight of empty container must be float in grams
The weight of container and sample of soil must be float in grams
Weight of container and dry soil must be float in grams

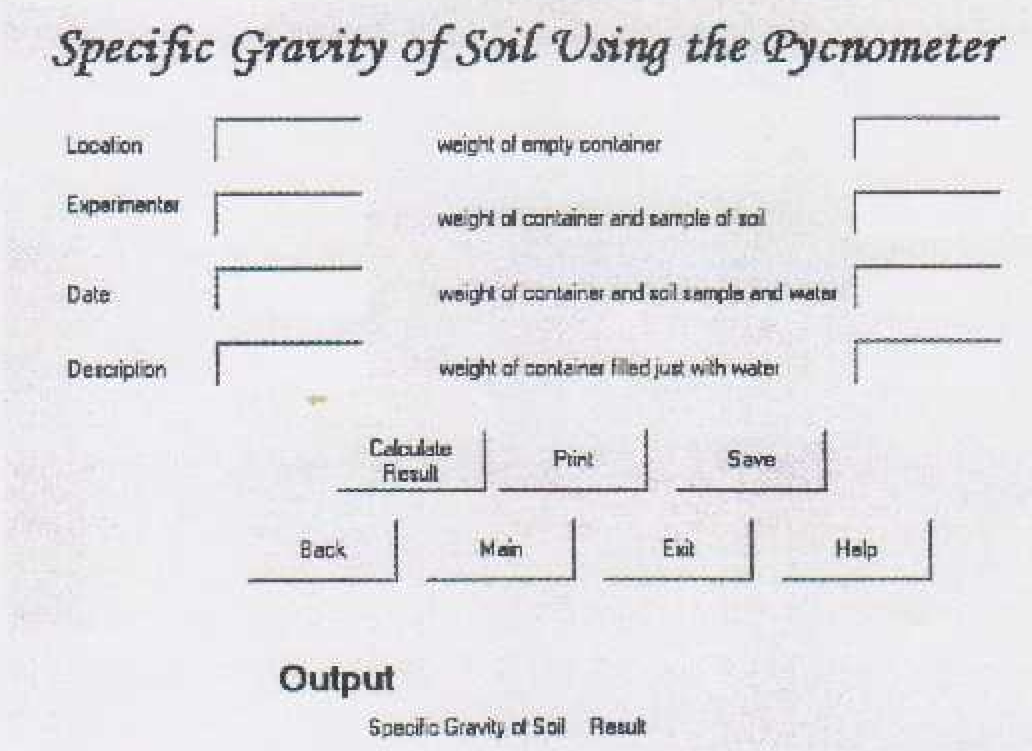


Figure 4.6 Specific Gravity of Soil Using the Pycnometer Screen

Specific Gravity of Soil Using the Pycnometer: this function enables the user to determine Specific Gravity of Soil Using the Pycnomete

The weight of empty container must be float in grams

The weight of container and sample of soil must be float in grams

Weight of container and soil sample and water must be float in grams

Weight of container filled just with water must be float in grams

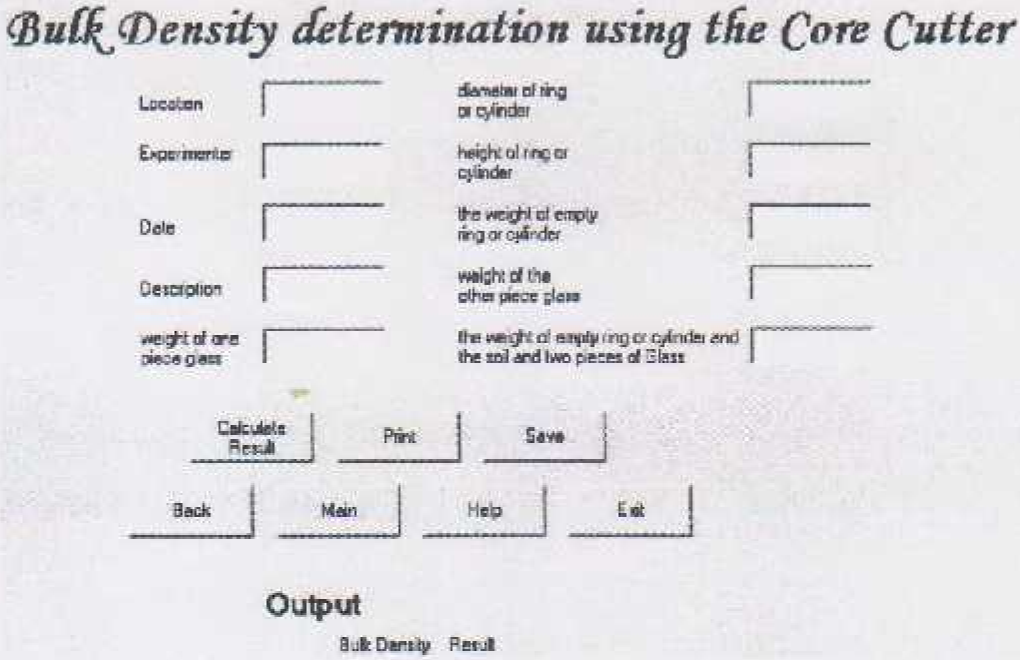


Figure 4.7 Bulk Density determinations using the Core Cutter Screen

Bulk Density determination using the Core Cutter: this function enables the user to determine Bulk Density determination using the Core Cutter

Diameter of ring or cylinder must be float in cm

Height of ring or cylinder: must be float in cm

The weight of empty ring or cylinder must be float in grams

The weight of empty ring or cylinder and the soil and two pieces of Glass: must be float in grams

Weight of one piece glass must be float in grams

Weight of the other piece glass must be float in grams

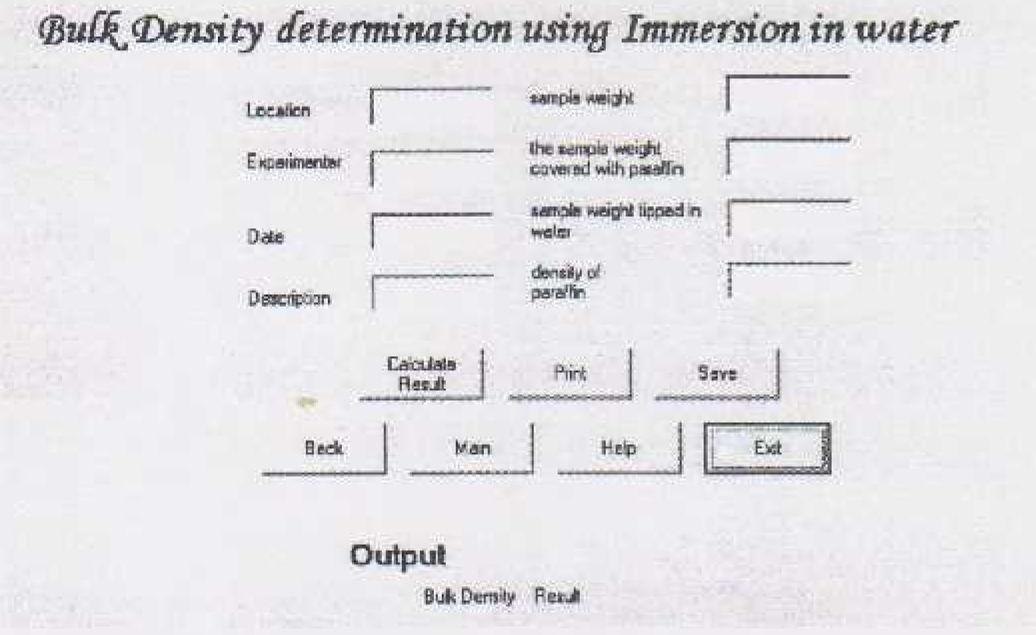


Figure 4.8 Bulk Density determination using Immersion in water Screen

Bulk Density determination using Immersion in water: this function enables the user to determine Bulk Density determination using Immersion in water

Sample weight: must be float in grams

The sample weight covered with paraffin: must be float in grams

Sample weight tipped in water must be float in grams

Density of paraffin: must be float in grams/cm

# *Bulk Density determination by direct measurement*

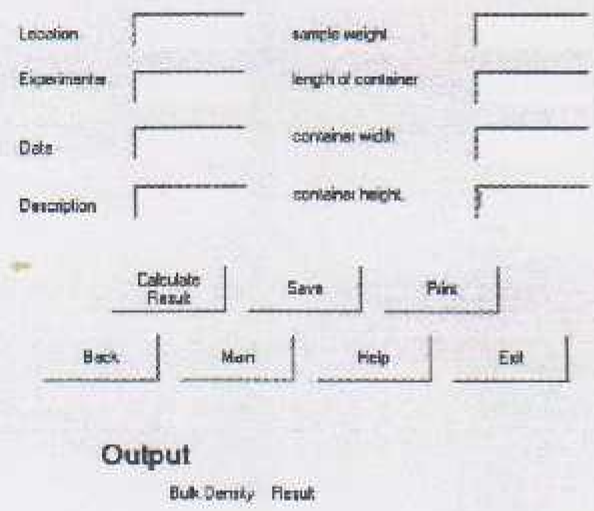


Figure 4.9 Bulk Density determinations by direct measurement Screen

Bulk Density determination by direct measurement: this function enables the user to determine Bulk Density by direct measurement

Sample weight. Must be float in grams

Length of container must be float in mm

Container width must be float in mm

Container height. Must be flout in mm

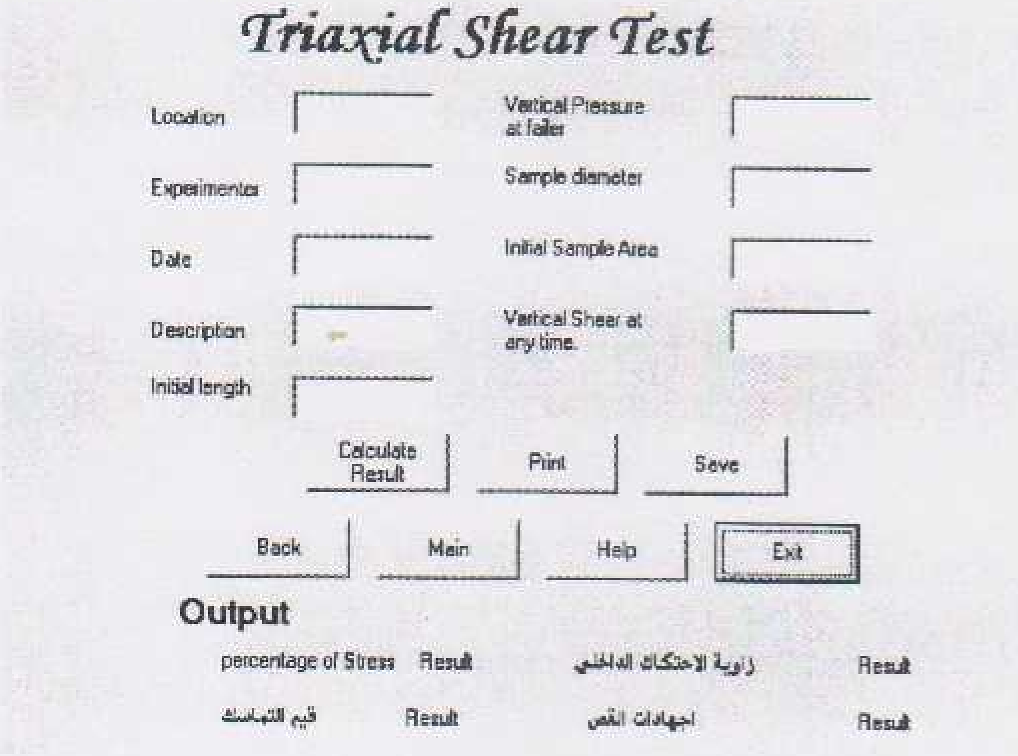


Figure 4.10 Triaxial Shear Test Screen

Triaxial Shear Test: this function enables the user to determine Triaxial Shear

Initial Length: must be float in mm

Sample Diameter: must be float in mm

Initial Sample Area.: must be float in mm

Vertical Shear at any time: must be float in mm

Vertical Pressure at failure must be float in mm

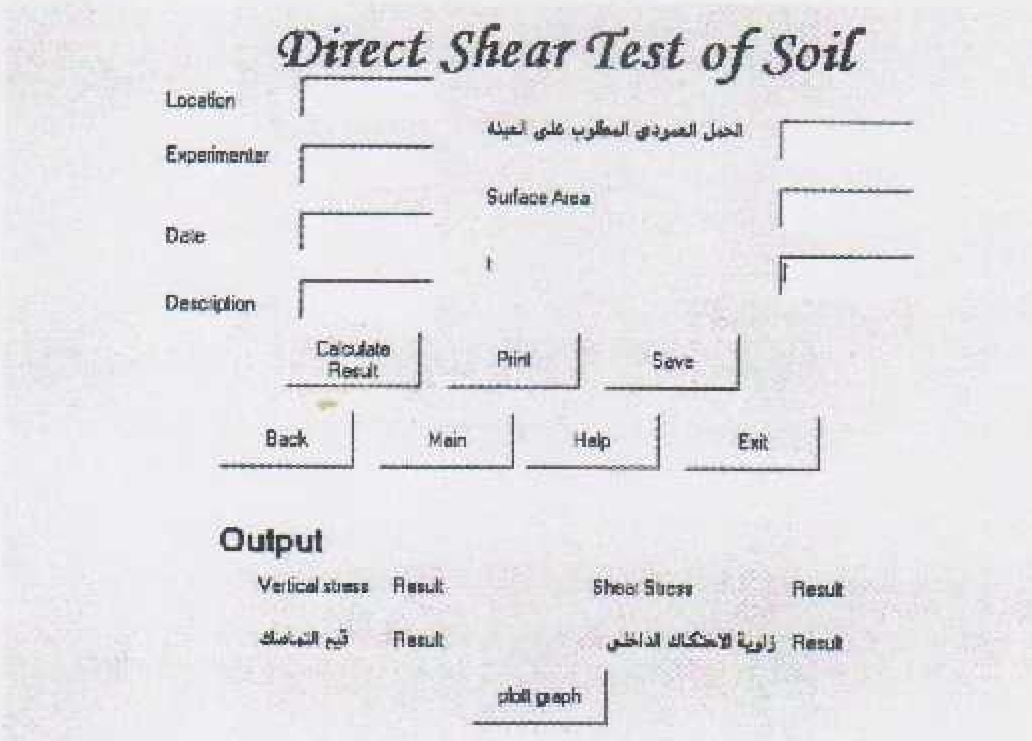


Figure 4.11 Direct Shear Test of Soil Screen

Direct Shear Test of Soil: this function enables the user to determine the Direct Shear of Soil

vertical stress on the sample must be float in KNewton

Surface Area must be float in meter square

Must be float in KNewton

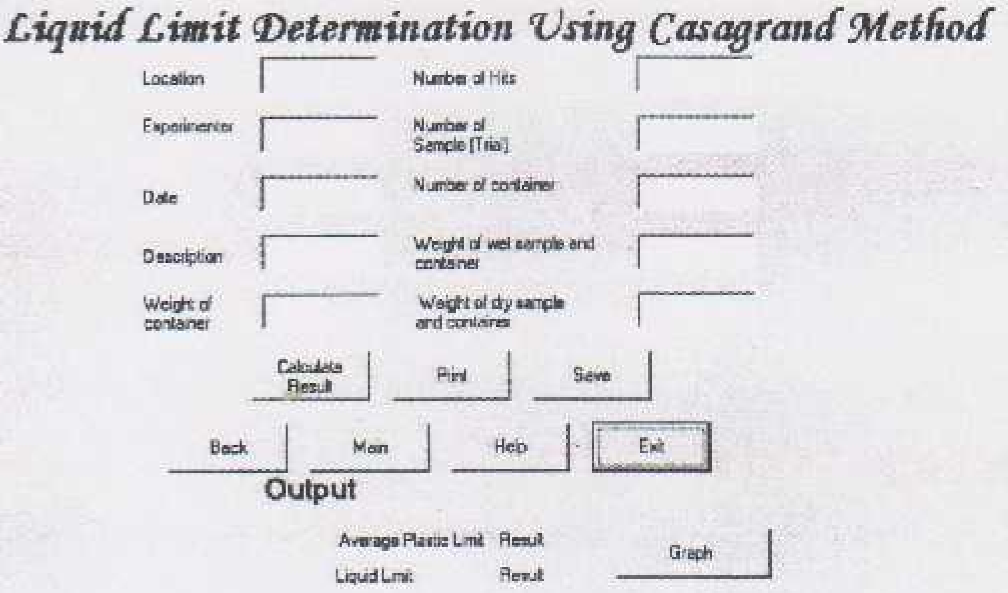


Figure 4.12 Liquid Limit Determination Using Casagrand Method Screen

Liquid Limit Determination Using Casagrand Method: this function enables the user to determine Liquid Limit Using

Number of Hits must be Integer

Number of Sample (Trial) must be Integer

Number of container must be Integer

Weight of wet sample and container must be float in gram

Weight of dry sample and container must be float in gram

Weight of container must be float in gram

# *Field Density Test Using Sand Replacement Method*

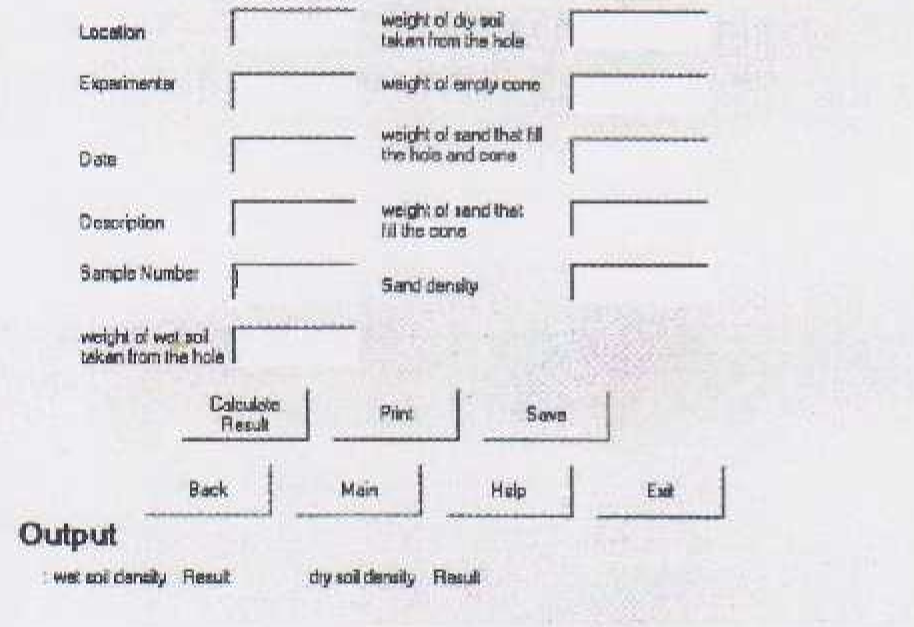


Figure 4.13 Field Density Test Using Sand Replacement Method Screen

Field Density Test Sand Replacement Method: this function enables the user to determine Field Density Test Using Sand Replacement Method

Sample Number Hits must be Integer

Weight of wet soil taken from the whole container must be float in gram

Weight of dry soil taken from the whole container must be float in gram

Weight of empty cone container must be float in gram

Weight of sand that fill the hole and cone container must be float in gram

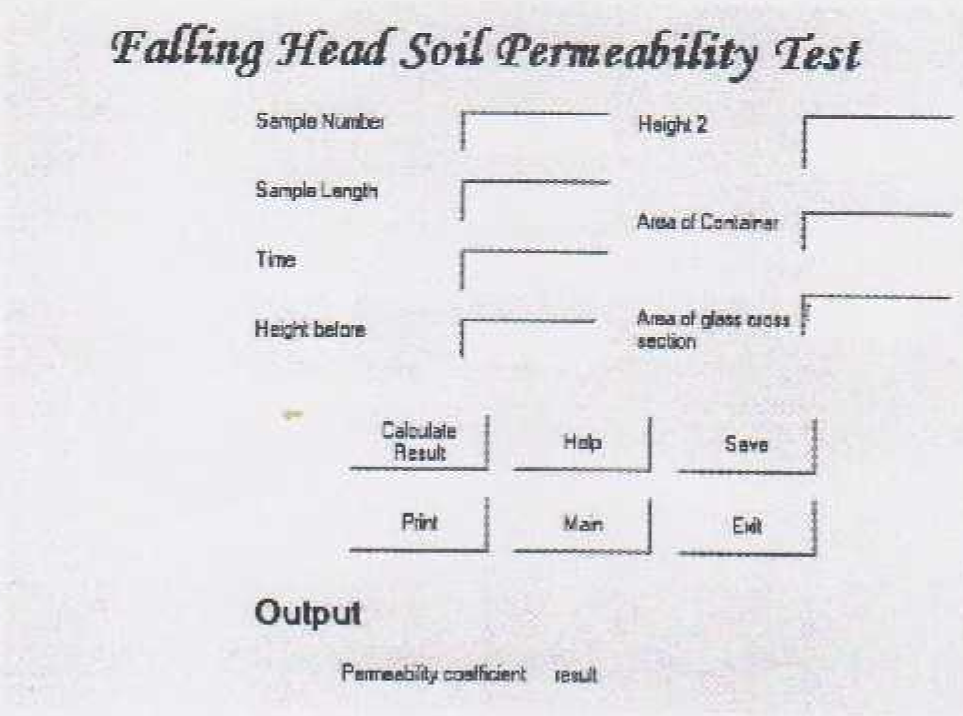


Figure 4.14 Falling Head Soil Permeability Test Screen

Falling Head Soil Permeability Test: this function enables the user to determine Soil Permeability using Falling Head test.

Sample Number must be integer

Sample Length must be float in cm

Time must be float in second

Height before must be float in cm

Height 2 must be float in cm

Area of Container must be float in cm square

Area of glass cross section must be float in cm square

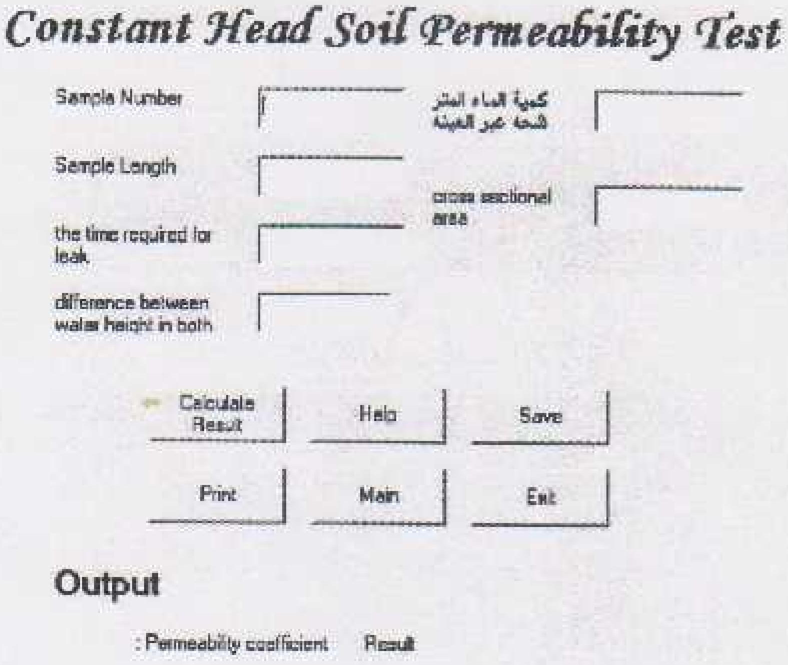


Figure 4.15 Constant Head Soil Permeability Test Screen

Constant Head Soil Permeability Test: this function enables the user to determine Soil Permeability using Constant Head test.

Sample Number must be Integer

Amount of filtered water from the sample: must be flout in cm cube

Sample Length must be flout in cm

Cross sectional area must be flout in cm square

The time required for leak must be flout in second

Difference between water heights in both tubes: must be flout in cm

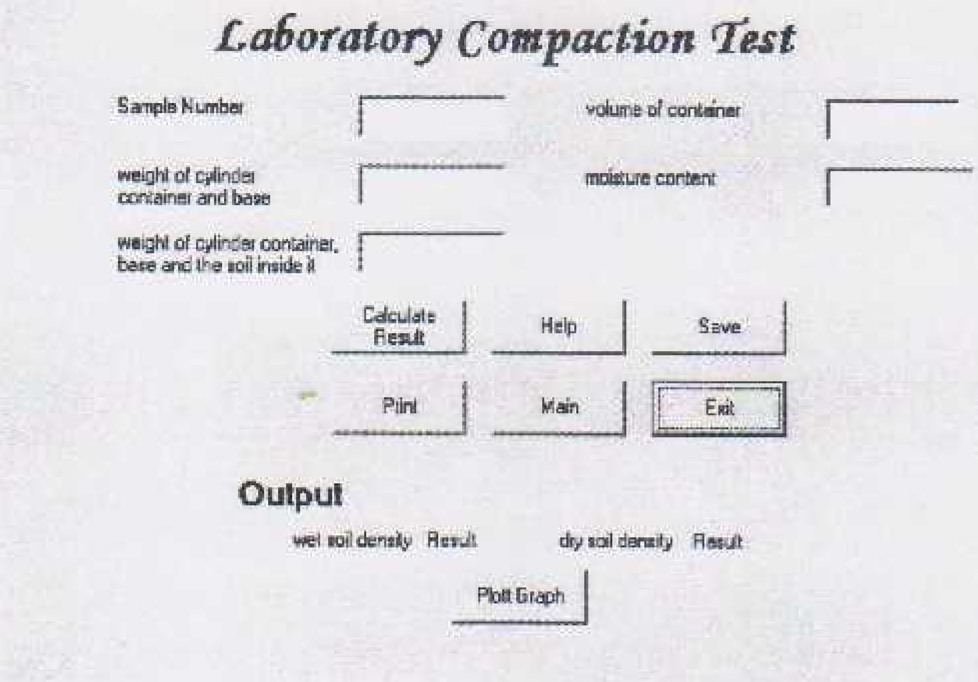


Figure 4.16 Laboratory Compaction Test Screen

Laboratory Compaction Test: this function enables the user to determine Compaction in the laboratory

Sample Number must be Integer

Weight of cylinder container and base must be float in gram

Weight of cylinder container, base and the soil inside it must be float in gram

Volume of container must be float in $cm^3$

Moisture content must be float

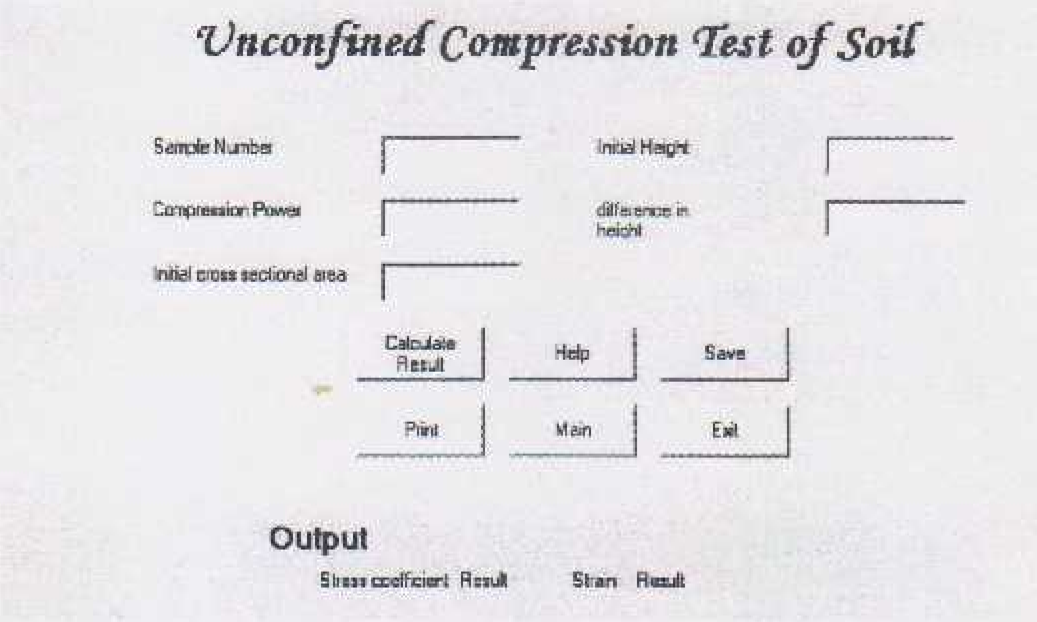


Figure 4.17 Unconfined Compression Test of Soil Screen

Unconfined Compression Test of Soil: this function enables the user to determine the shape of the sample at failure

Sample Number must be Integer

Compression Power: must be float in KNewton

Initial cross sectional area: must be float in cm square

Initial Height: must be float in mm

Difference in height must be float in mm

### 4.2.2 Flow charts:

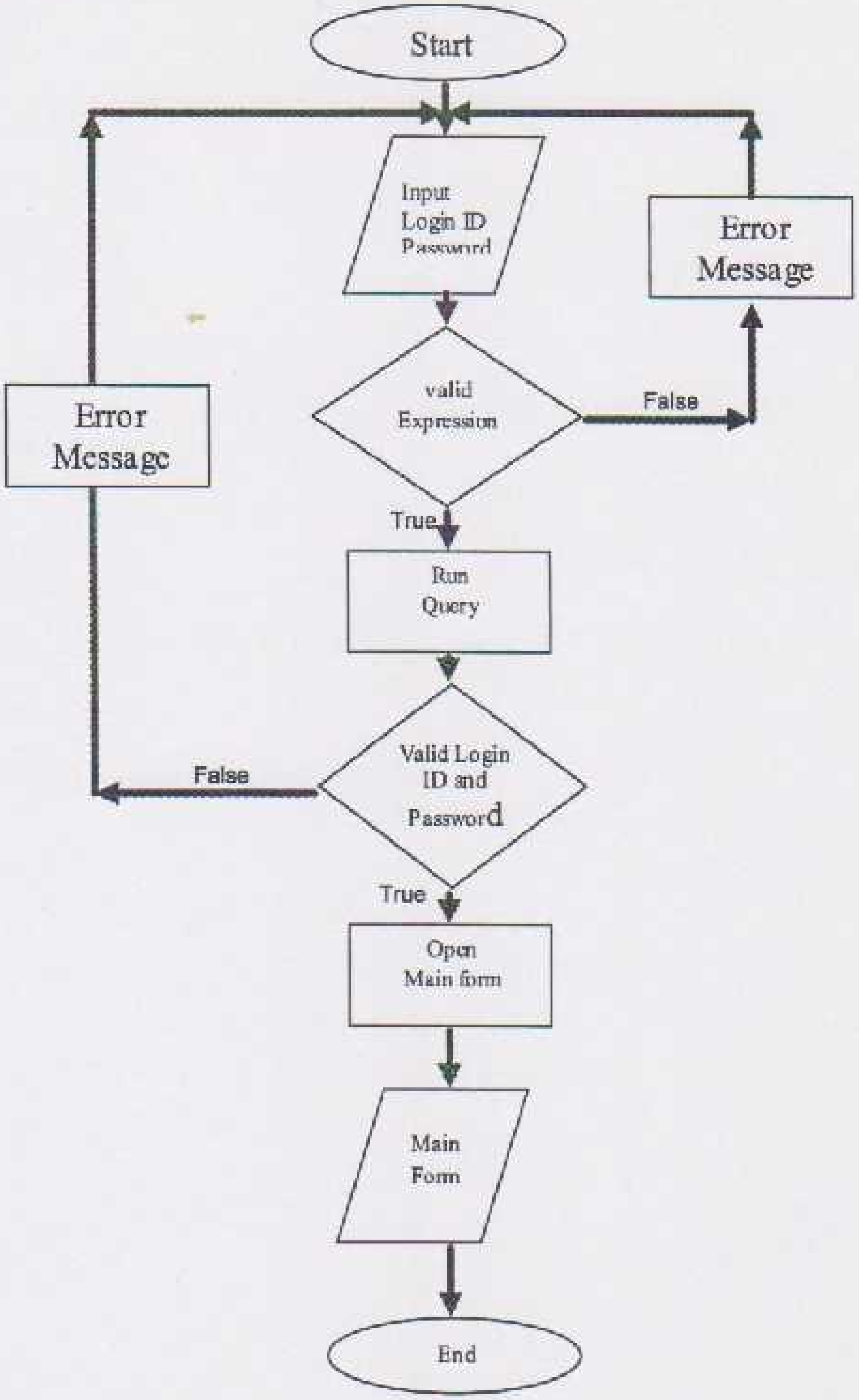


Figure 4.18 Login Flowchart

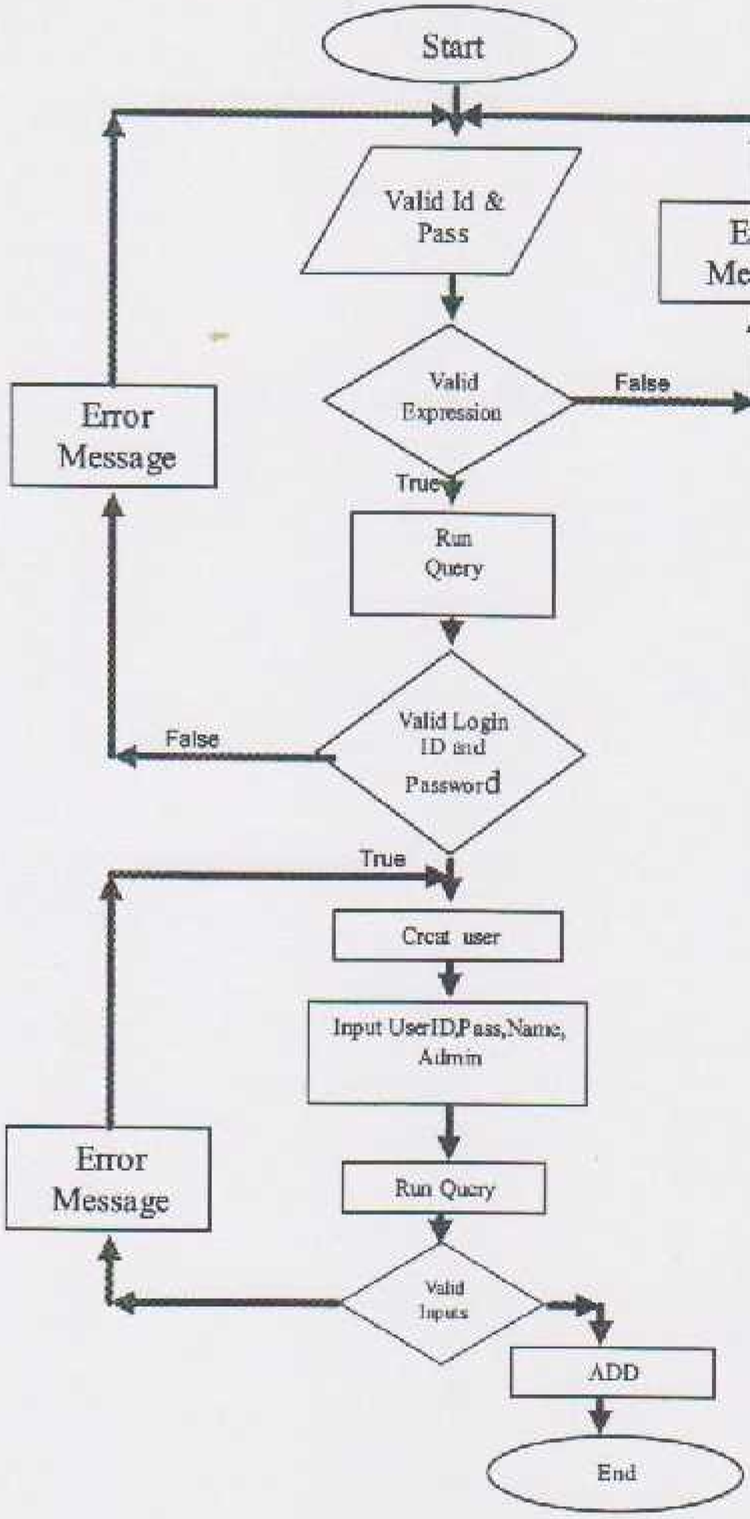


Figure 4.19 Create user Flowchart

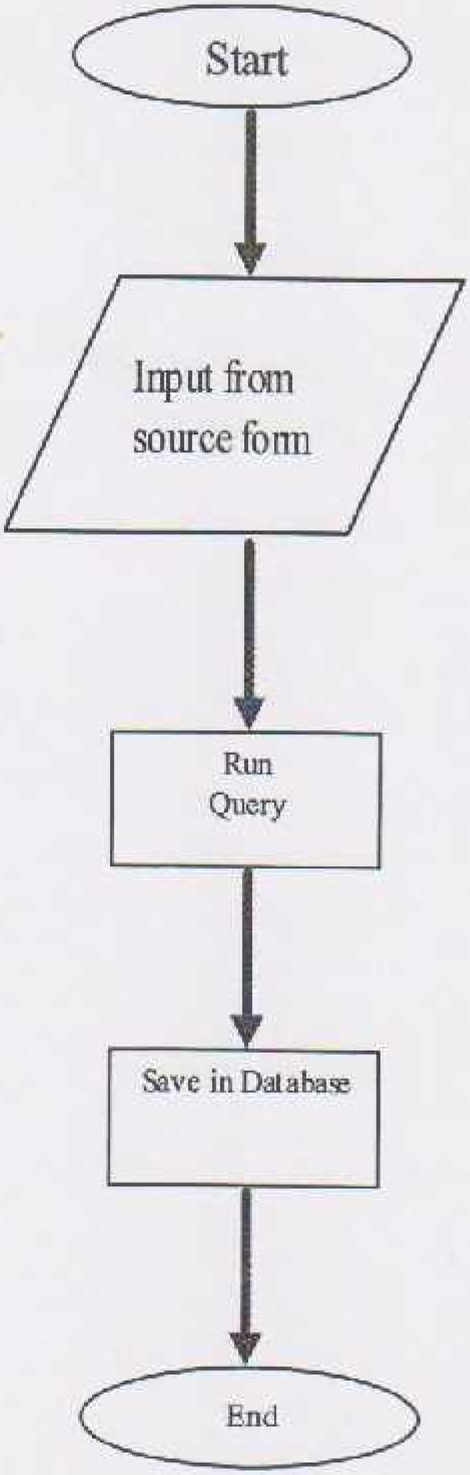


Figure 4.20 Save Exp. Flowchart

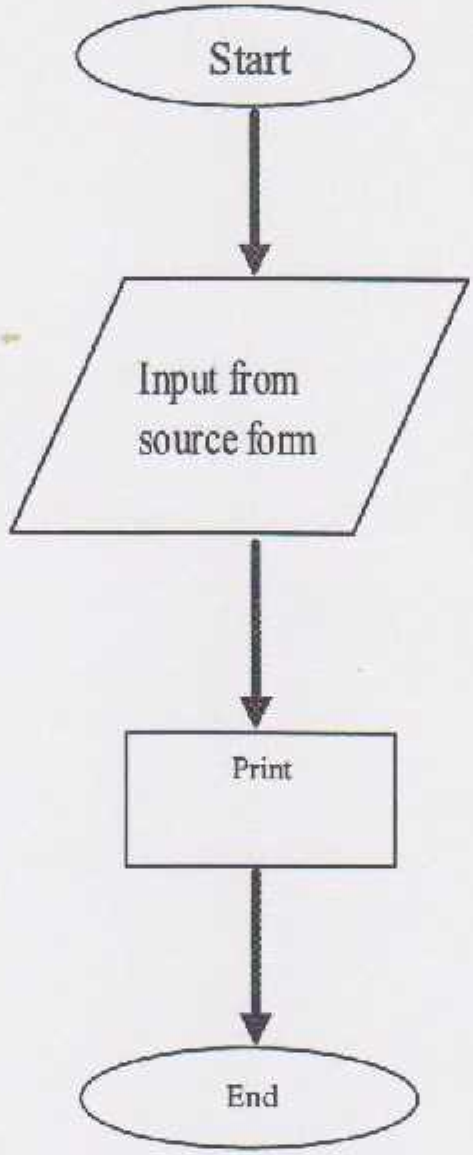


Figure 4.21 Print Exp. Flowchart

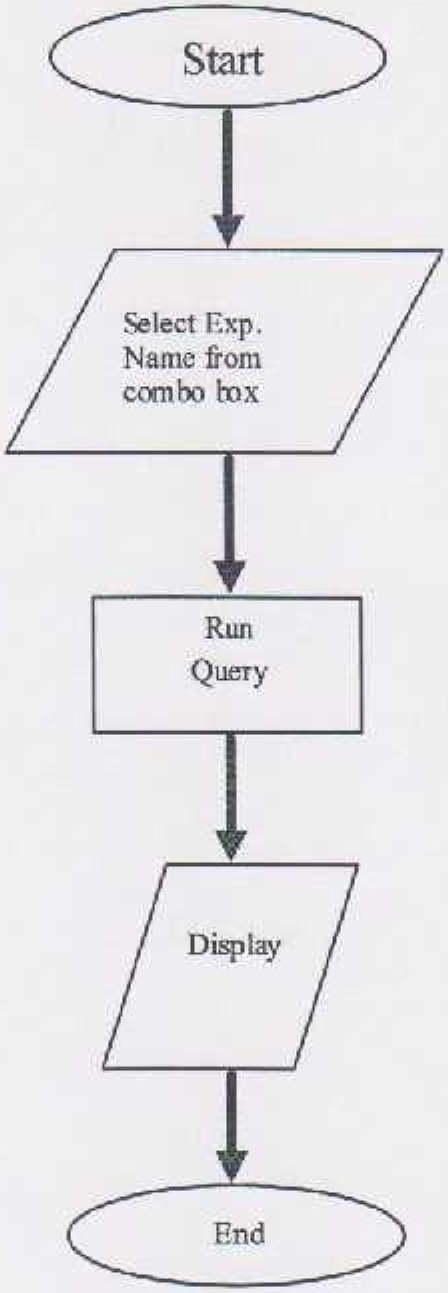


Figure 4.22 Retrieve Exp. Flowchart

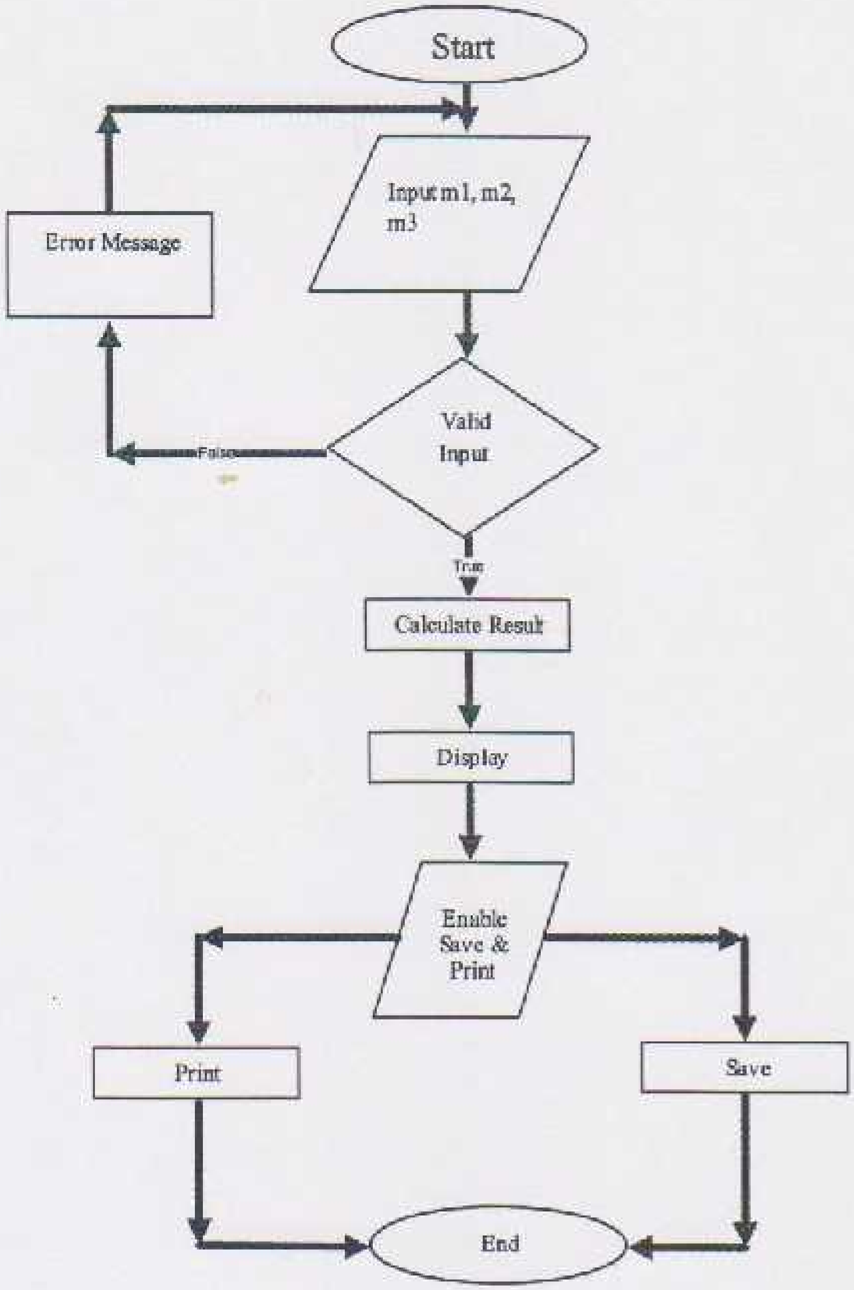


Figure 4.23 Water or moisture Content Flowchart

**Where:**

m1: the weight of empty container

m2: the weight of container and sample of soil

m3: weight of container and dry soil

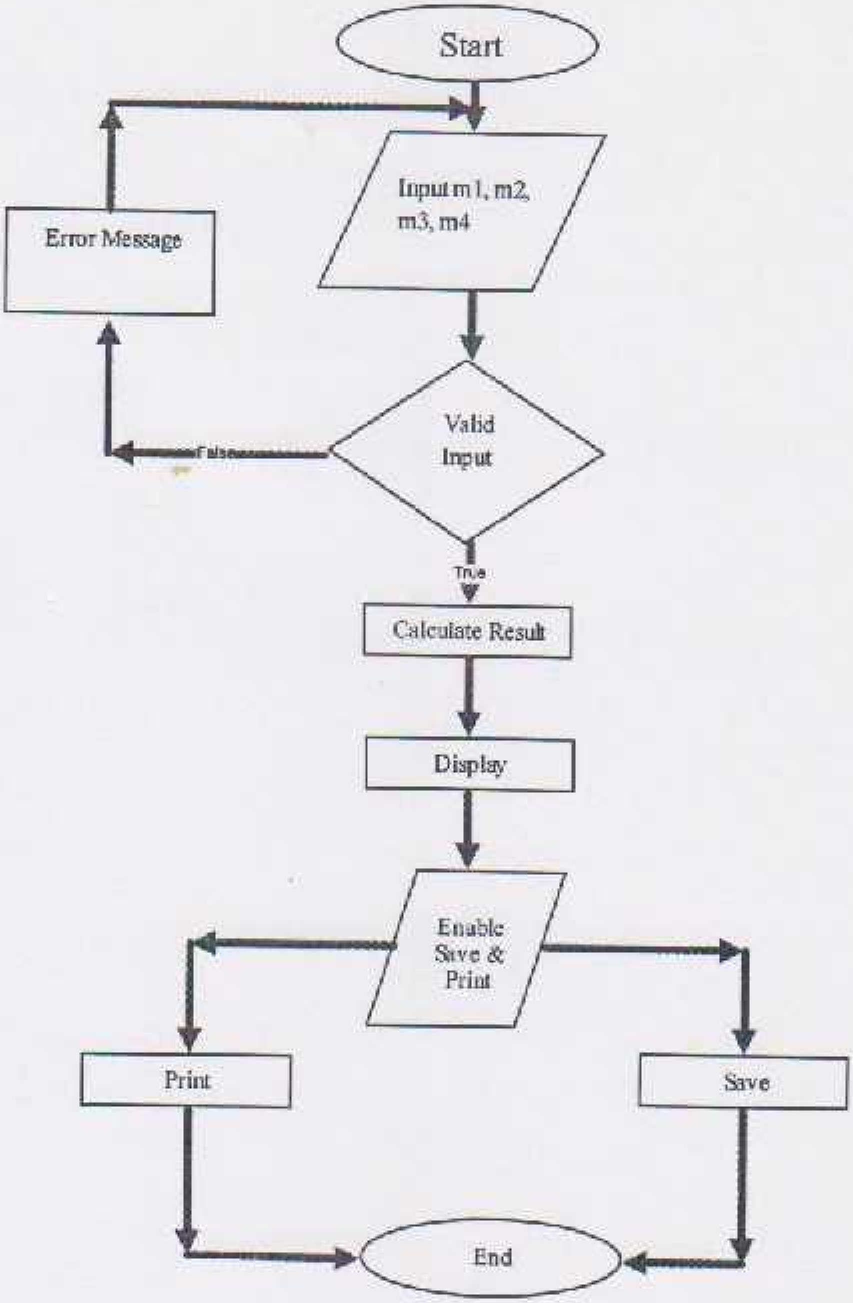


Figure 4.24 Specific Gravity of Soil Using the Pycnometer Flowchart

**Where:**

m1: the weight of empty container

m2: the weight of container and sample of soil

m3: weight of container and soil sample and water

m4: weight of container filled just with water

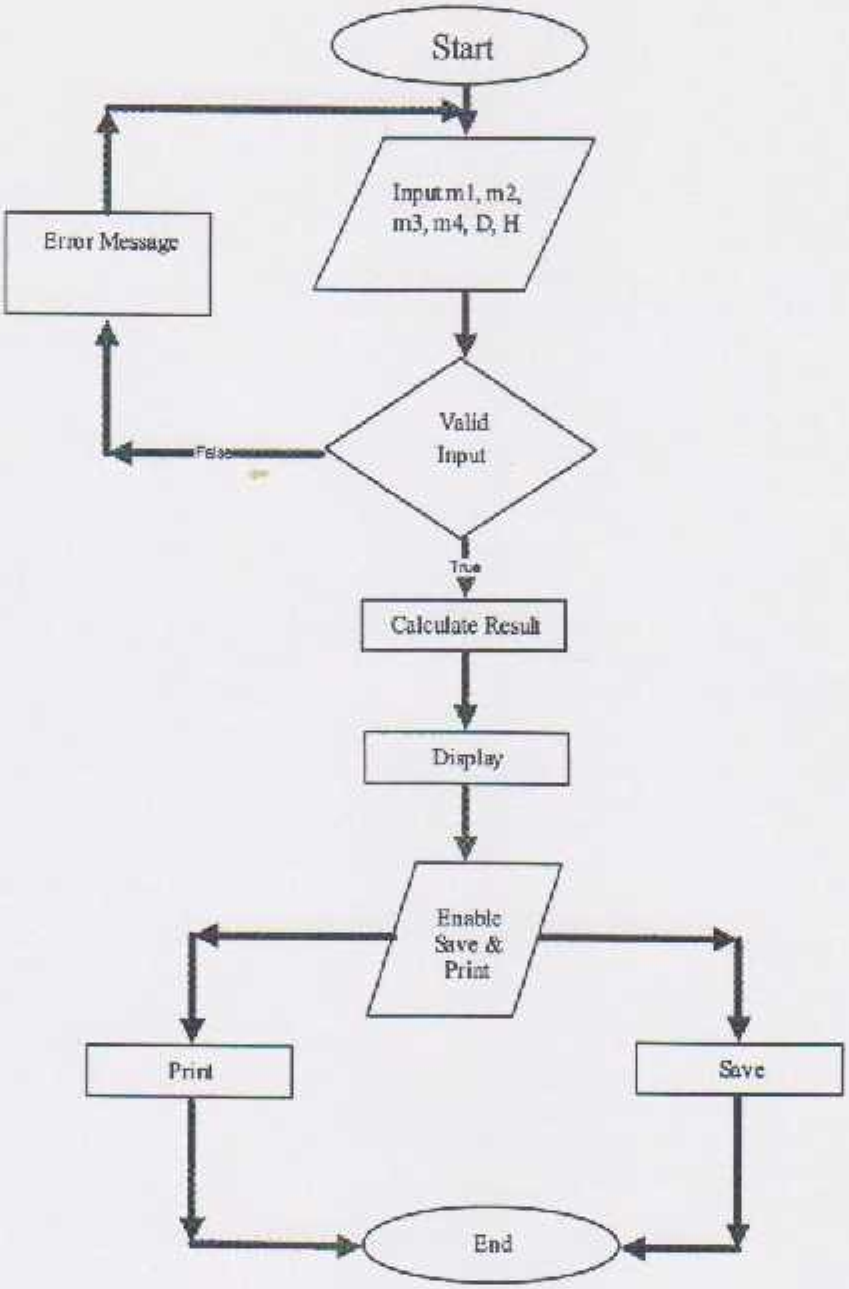


Figure 4.25 Bulk Density determinations using the Core Cutter Flowchart

**Where:** D: diameter of ring or cylinder

H: height of ring or cylinder.

m1: the weight of empty ring or cylinder

m2: the weight of empty ring or cylinder and the soil and two pieces of Glass.

m3: weight of one piece glass

m4: weight of the other piece glass

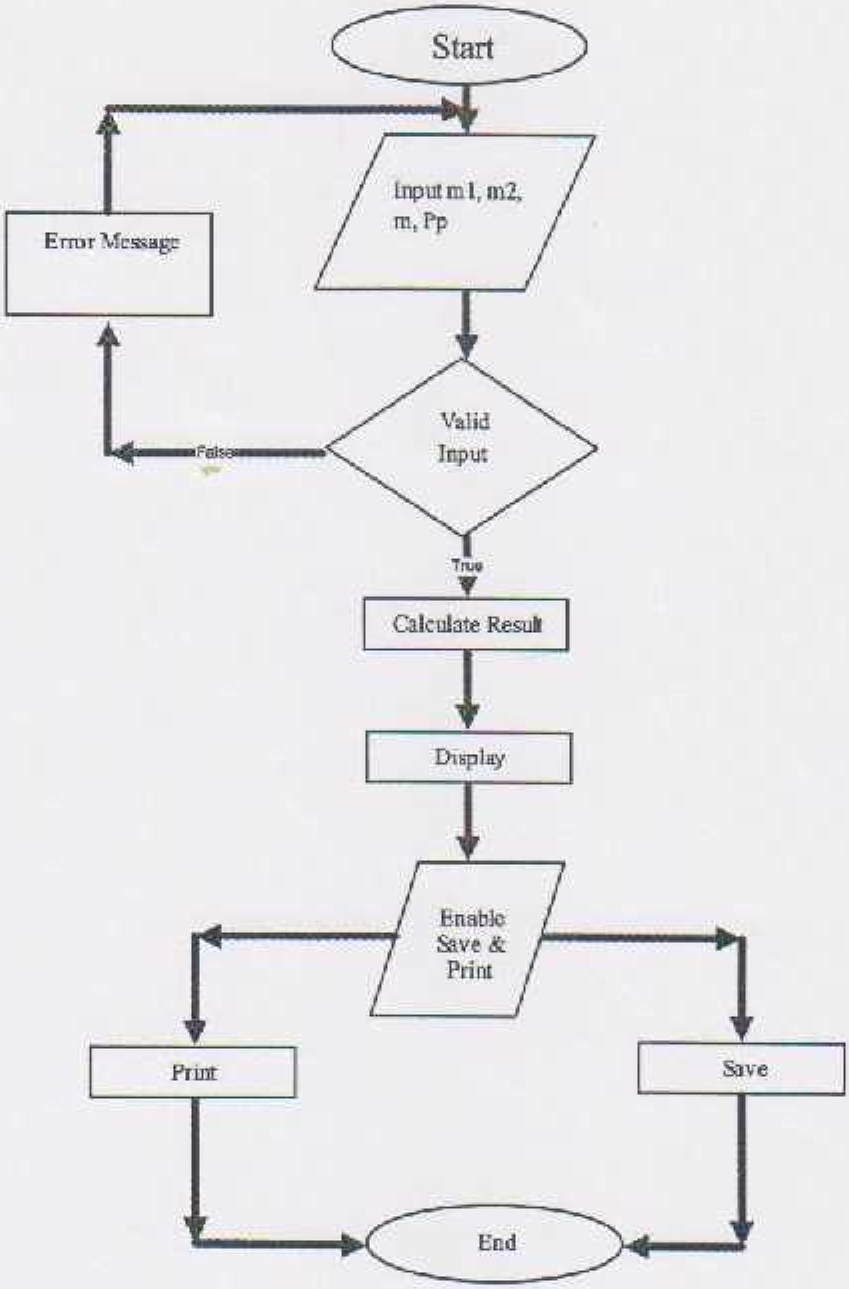


Figure 4.26 Bulk Density determination using Immersion in water Flowchart

Where:m: sample weight.

m1: the sample weight covered with paraffin.

m2: sample weight tipped in water

Pp: density of paraffin.

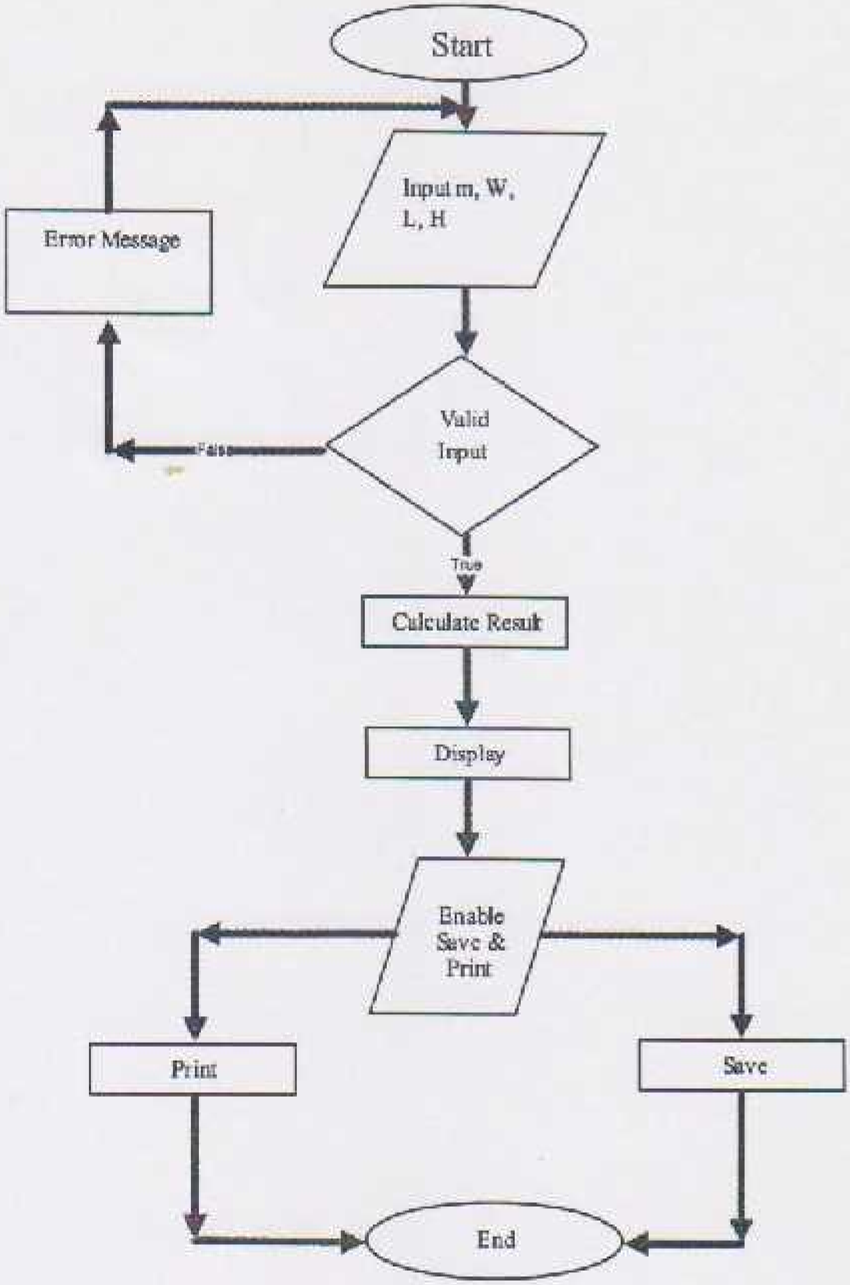


Figure 4.27 Bulk Density determination by direct measurement Flowchart

**Where:** m: sample weight.

L: length of container

W: container width

H: container height.

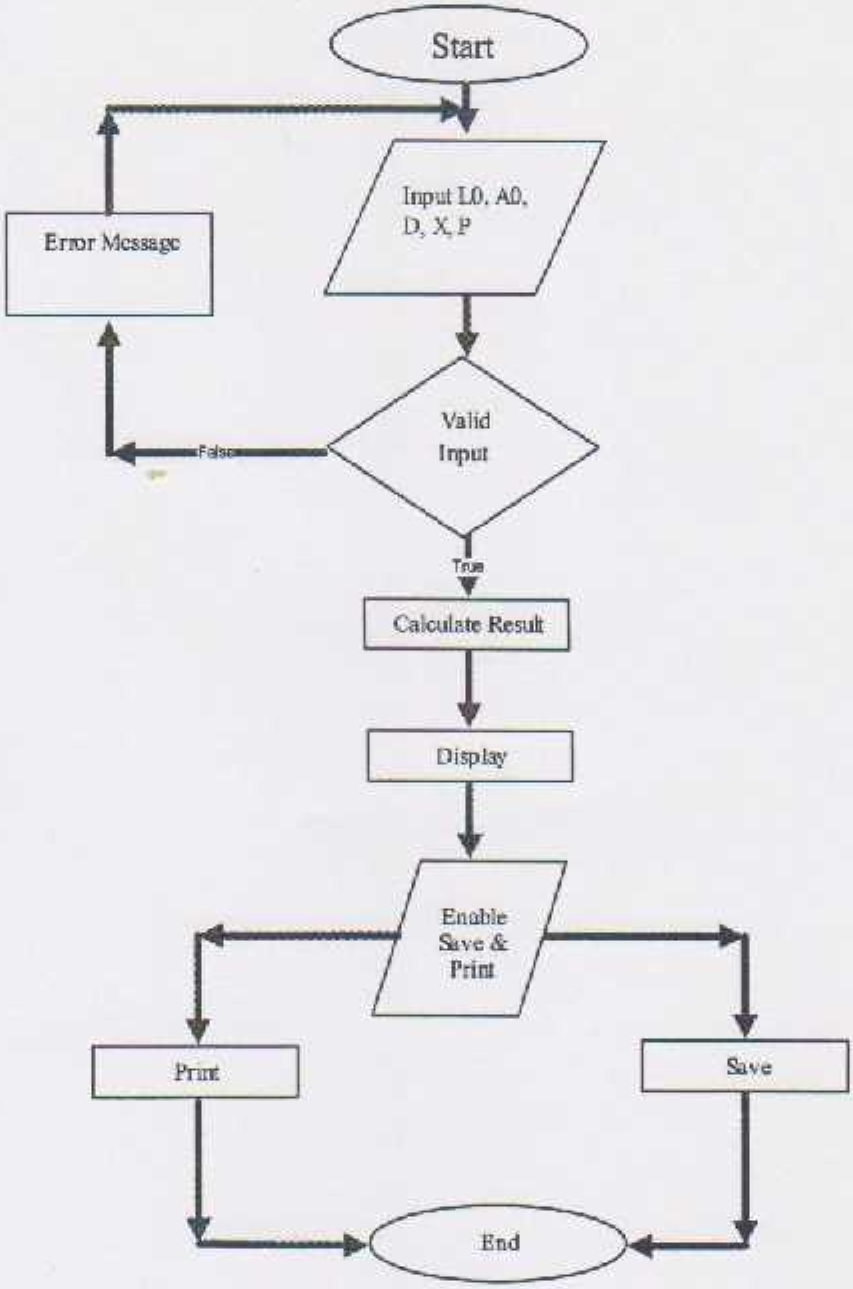


Figure 4.28 Triaxial Shear Test Flowchart

**Where:** L0: Initial length.

D: Sample diameter.

A0: Initial Sample Area.

X: Vertical Shear at any time.

P: Vertical Pressure at failure

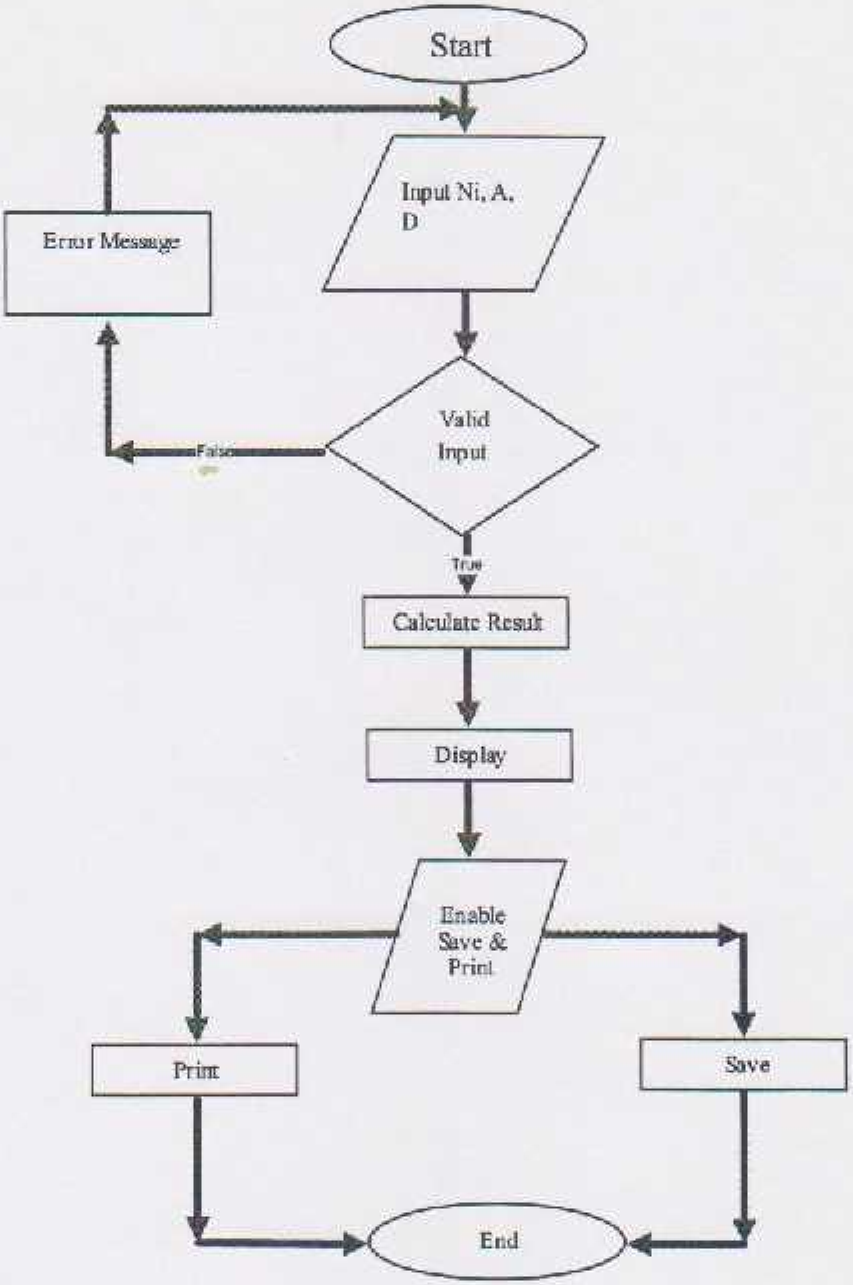


Figure 4.29 Direct Shear Test of Soil Flowchart

Where: Ni: Vertical Stress On the Sample

**A:** Surface Area

T: Shear Stress

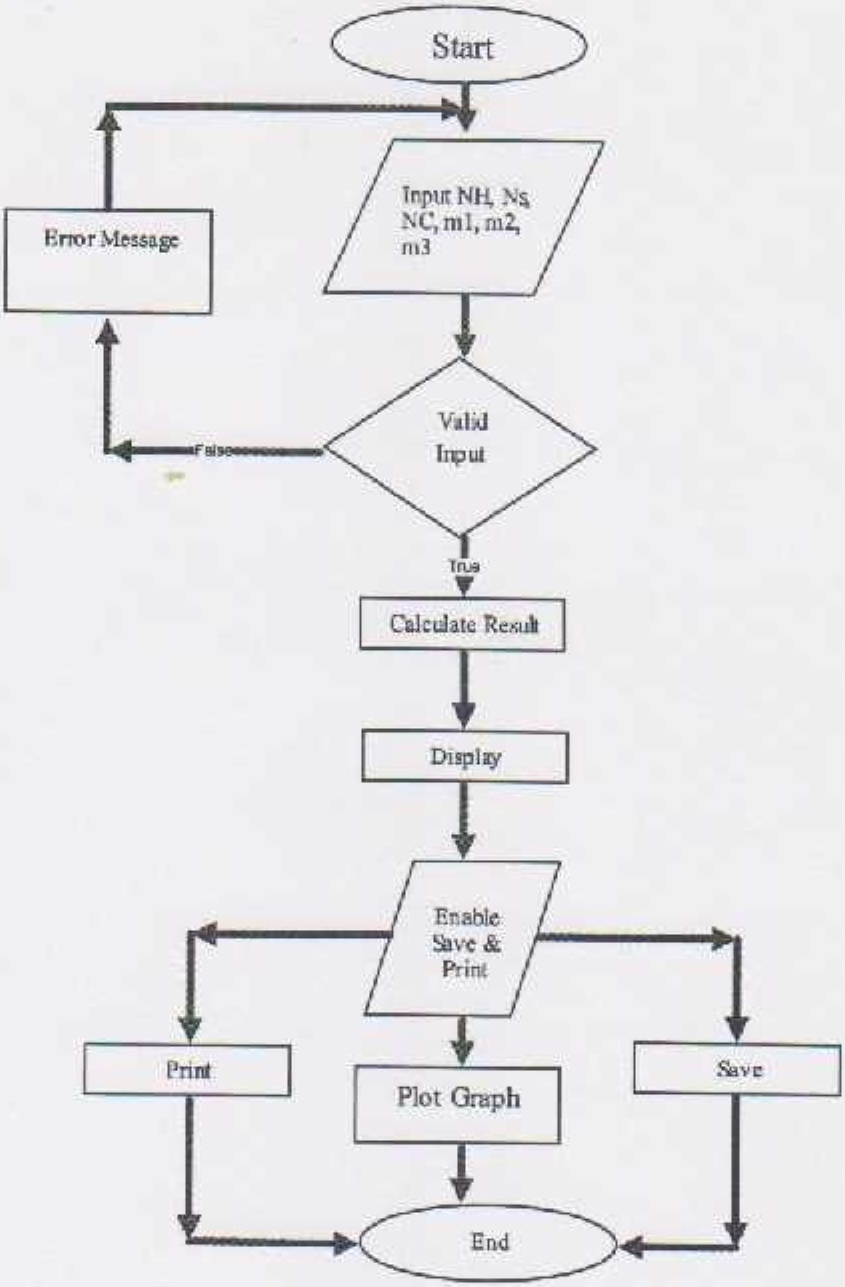


Figure 4.30 Liquid Limit Determination Using Casagrand Method Flowchart

Where: Nh: Number of Hits

Ns: Number of Sample (Trial)

Nc: Number of container

m1: Weight of wet sample and container

m2: Weight of dry sample and container

m3: Weight of container

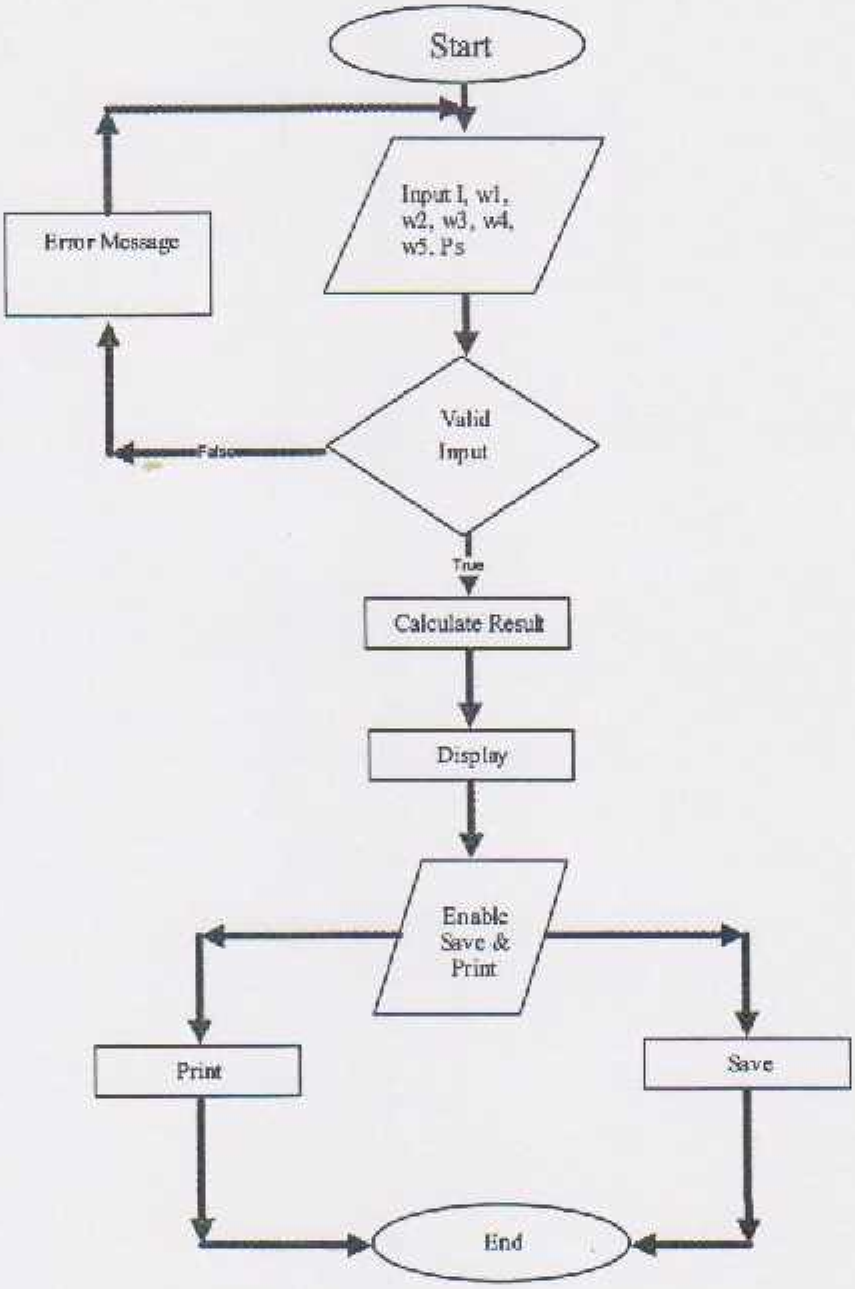


Figure 4.31 Field Density Test Using Sand Replacement Method Flowchart

Where: I: Sample Number

W1: weight of wet soil taken from the hole

W2: weight of dry soil taken from the hole

W3: weight of empty cone

W4: weight of sand that fill the hole and cone

W5: weight of sand that fill the cone

Ps: Sand density

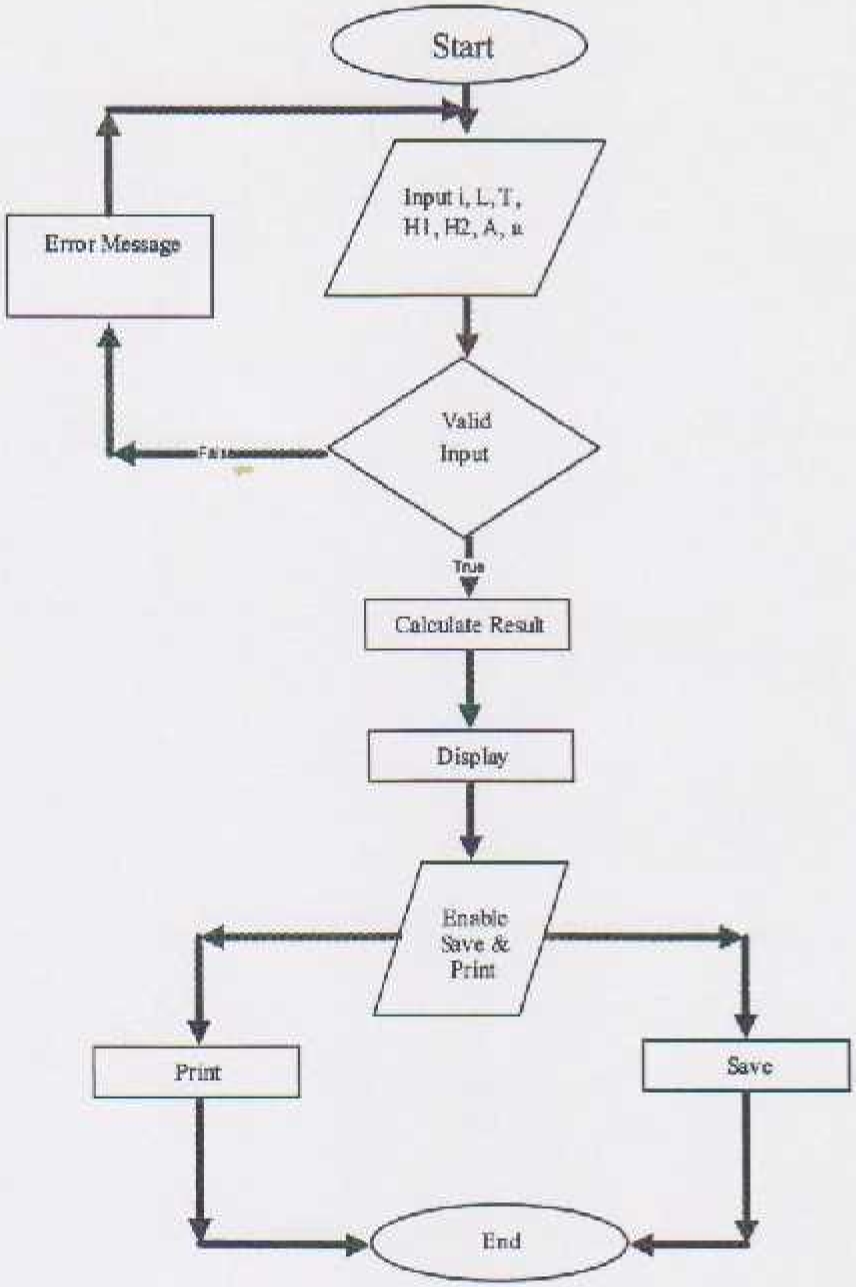


Figure 4.32 Falling Head Soil Permeability Test Flowchart

**Where:** I: Sample Number

L: Sample Length

T: Time

H1: Height before

H2: Height 2

A: Area of Container

a: Area of glass cross section

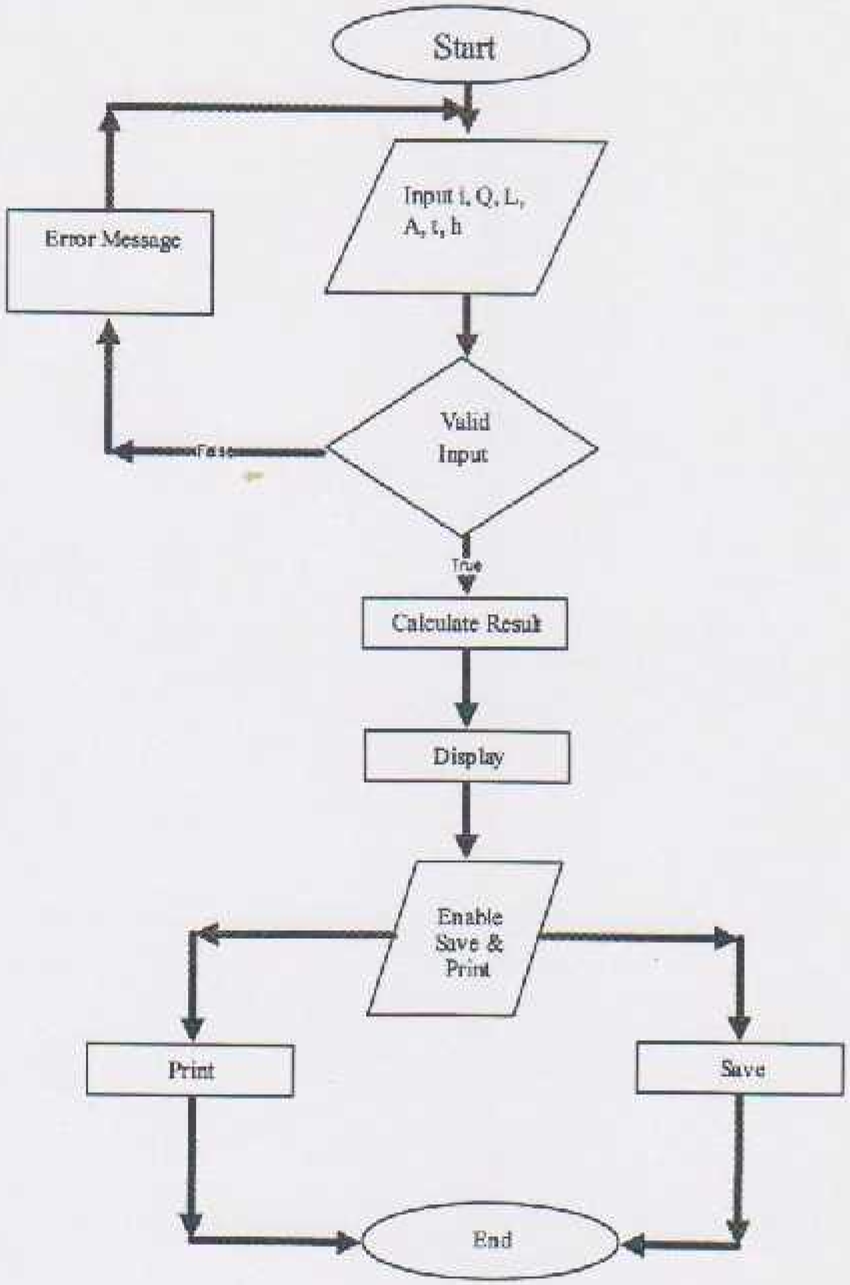


Figure 4.33 Constant Head Soil Permeability Test Flowchart

Where: i: Sample Number

Q: Amount of filtered water from the sample

L: Sample Length

A: cross sectional area

t: the time required for leak

h: difference between water height in both tubes

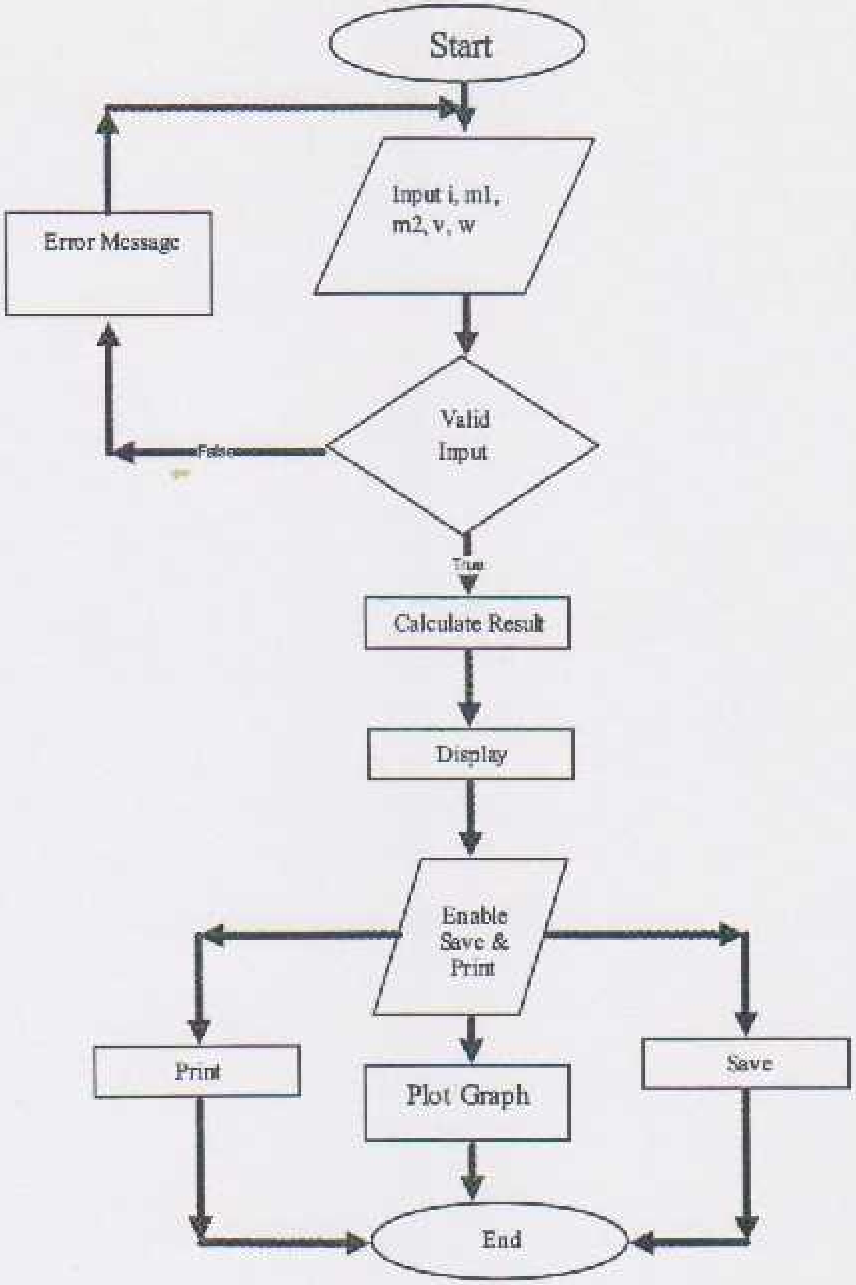


Figure 4.34 Laboratory Compaction Test Flowchart

**Where:** i: Sample Number

m1: weight of cylinder container and base

m2: weight of cylinder container, base and the soil inside it.

V: volume of container

W: moisture content.

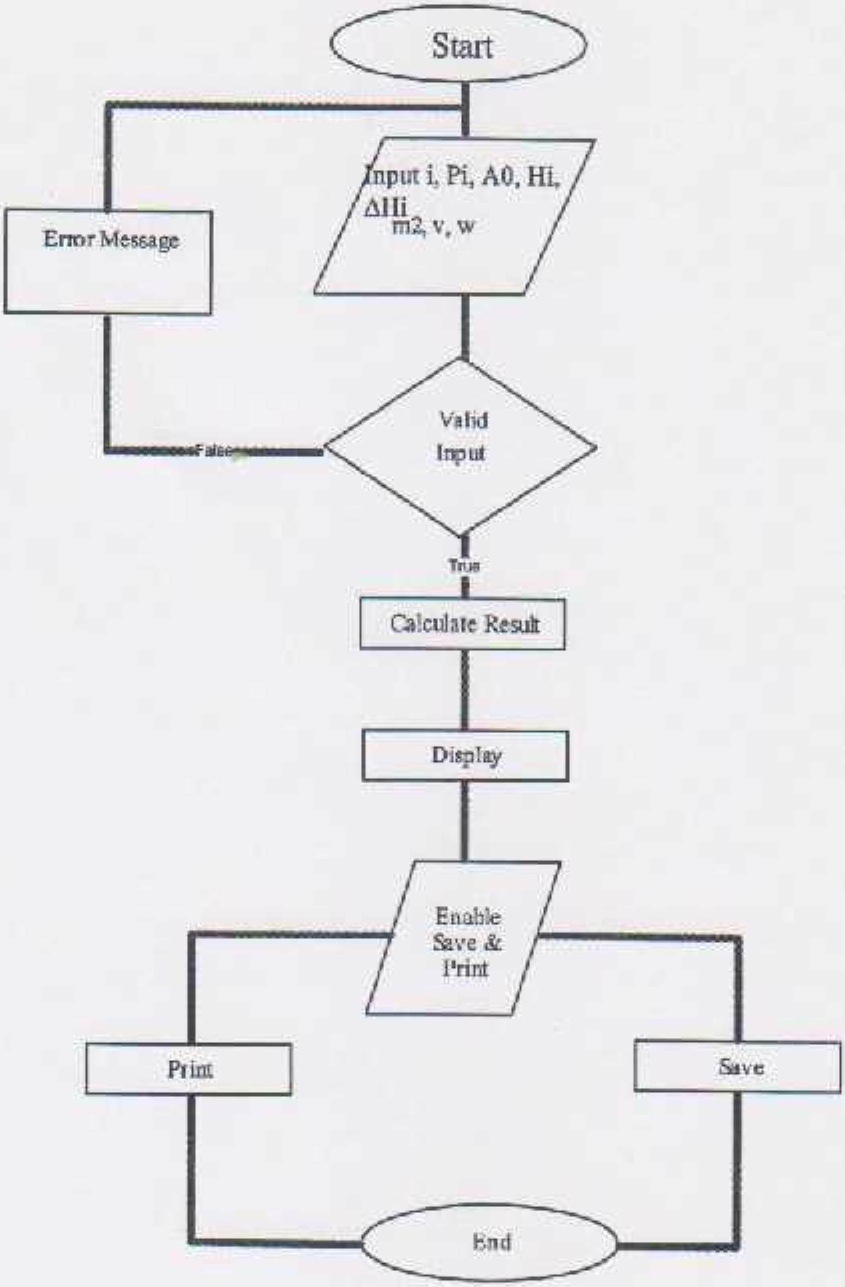


Figure 4.35 Unconfined Compression Test of Soil Flowchart

**Where:** i: Sample Number

Pi: Compression Power

A0: Initial cross sectional area

H: Initial Height

ΔHi: difference in height

### 4.2.3 Interface:

**Login**

Identify the user using login ID and password

- Input: Login ID, login Password.
- Output: main form depending on the user type

**Create User**

This function enables the administrator to create a new user to be able to use the program

- Input: User ID, user password, password_ confirmation, Admin
- Output: update user list

**Save Experiment**

This function enables the administrator to save experiment information

- Input: Source experiment Inputs and calculations
- Output: Store in Database.

**Retrieve Experiment**

This function enables the user to be retrieve the information of experiment

- Input: experiment number and location
- Output: experiment information

**Water or moisture Content**

This function enables the user to perform this experiment

- Input: : the weight of empty container
  The weight of container and sample of soil
  Weight of container and dry soil
- Output: water or moisture content **(w %)**

**Specific Gravity of Soil Using the Pycnometer**

This function enables the user to perform this experiment

- Input: : the weight of empty container
  The weight of container and sample of soil
  Weight of container and soil sample and water
  Weight of container filled just with water
- Output: Specific Gravity of Soil

**Bulk Density determination using the Core Cutter**

This function enables the user to perform this experiment

- Input: : diameter of ring or cylinder
  Height of ring or cylinder.
  The weight of empty ring or cylinder
  The weight of empty ring or cylinder and the soil and two pieces of Glass.
  Weight of one piece glass
  Weight of the other piece glass
- Output: Bulk Density1

**Bulk Density determination using Immersion in water**

This function enables the user to perform this experiment

- Input: sample weight.
  The sample weight covered with paraffin.
  Sample weight tipped in water
  Density of paraffin.
- Output: Bulk Density2

**Bulk Density determination by direct measurement**

This function enables the user to perform this experiment

- Input: sample weight.
  Length of container
  Container width
  Container height.

- Output: Bulk Density3

**Triaxial Shear Test**

This function enables the user to perform this experiment

- Input: Initial length
  Sample diameter
  Initial Sample Area
  Vertical Shear at any time
  Vertical Pressure at failure
- Output: : percentage of Stress
  Cohesion
  Angle of the internal friction
  Stress test

**Direct Shear Test of Soil**

This function enables the user to perform this experiment

- Input: vertical
  Surface Area
- Output: Vertical stress
  Shear Stress
  Cohesion
  Angle of friction stress
  Graph (σi: X τ)

**Liquid Limit Determination Using Casagrand Method**

This function enables the user to perform this experiment

- Input: Number of Hits
  Number of Sample (Trial)
  Number of container
  Weight of wet sample and container
  Weight of dry sample and container
  Weight of container

- Output: Average Plastic Limit
  Liquid Limit
  Graph (Number of hits X Water Content)

**Field Density Test Using Sand Replacement Method**

This function enables the user to perform this experiment

- Input: Sample Number
  Weight of wet soil taken from the hole
  Weight of dry soil taken from the hole
  Weight of empty cone
  Weight of sand that fill the hole and cone
  Weight of sand that fill the cone
  Sand density
- Output: wet soil density
  Dry soil density

**Falling Head Soil Permeability Test**

This function enables the user to perform this experiment

- Input: Sample Number
  Sample Length
  Time
  Height before
  Height 2
  Area of Container
  Area of glass cross section
- Output: Permeability coefficient

**Constant Head Soil Permeability Test**

This function enables the user to perform this experiment

- Input: Sample Number
  Amount of filtered water from the sasmple
  Sample Length
  Cross sectional area
  The time required for leak
  Difference between water height in both tubes.

- Output: Permeability coefficient

**Laboratory Compaction Test**

This function enables the user to perform this experiment

- Input: Sample Number
  Weight of cylinder container and base
  Weight of cylinder container, base and the soil inside it.
  Volume of container
  Moisture content.
- Output: wet soil density
  Dry soil density
  Graph (Pd X Wc)

**Unconfined Compression Test of Soil**

This function enables the user to perform this experiment

- Input: Sample Number
  Compression Power
  Initial cross sectional area
  Initial Height
  Difference in height
- Output: Stress coefficient
  Strain

### 4.2.4 Constrains

**Login:**

- Login ID should contain at least six characters.
- Password should contain at least six characters.
- Password must be different from the login ID.
- Only alphanumeric characters will be accepted

**Create User:**

- Login ID should contain at least six characters.
- Password should contain at least six characters.
- Password must be different from the login ID.
- Login ID and Password accepts only alphanumeric characters

**Save Experiment**

- Location: text within 20 character
- Experimenter: text within 20 character
- Date : should be in date format
- Description: text.

**Retrieve Experiment**

- Experiment number should be within 1-13.
- Location should be alphabetical.

**Water or moisture Content**

- the weight of empty container must be float in grams
- the weight of container and sample of soil must be float in grams
- weight of container and dry soil must be float in grams

**Specific Gravity of Soil Using the Pycnometer**

- the weight of empty container must be float in grams
- the weight of container and sample of soil must be float in grams
- weight of container and soil sample and water must be float in grams
- weight of container filled just with water must be float in grams

**Bulk Density determination using the Core Cutter**

- diameter of ring or cylinder must be float in cm
- height of ring or cylinder: must be float in cm
- the weight of empty ring or cylinder must be float in grams
- the weight of empty ring or cylinder and the soil and two pieces of
- Glass: must be float in grams
- weight of one piece glass must be float in grams
- weight of the other piece glass must be float in grams.

**Bulk Density determination using Immersion in water**

- sample weight: must be float in grams
- the sample weight covered with paraffin: must be float in grams
- sample weight tipped in water must be float in grams
- density of paraffin: must be float in grams/cm.

**Bulk Density determination by direct measurement**

- sample weight. Must be float in grams
- length of container must be float in mm
- container width must be float in mm
- container height. Must be flout in mm

**Triaxial Shear Test**

- Initial Length: must be float in mm
- Sample Diameter: must be float in mm
- Initial Sample Area.: must be float in mm
- Vertical Shear at any time: must be float in mm
- Vertical Pressure at failure must be float in mm

**Direct Shear Test of Soil**

- Vertical stress on the sample must be float in KNewton
- Surface Area must be float in meter square
- must be float in KNewton

**Liquid Limit Determination Using Casagrand Method**

- Number of Hits must be Integer
- Number of Sample (Trial) must be Integer
- Number of container must be Integer
- Weight of wet sample and container must be float in gram
- Weight of dry sample and container must be float in gram
- Weight of container must be float in gram.

**Field Density Test Using Sand Replacement Method**

- Sample Number Hits must be Integer
- weight of wet soil taken from the hole container must be float in gram
- weight of dry soil taken from the hole container must be float in gram
- weight of empty cone container must be float in gram
- weight of sand that fill the hole and cone container must be float in gram

- weight of sand that fills the cone container must be float in gram

**Falling Head Soil Permeability Test**

- Sample Number must be integer
- Sample Length must be float in cm
- Time must be float in second
- Height before must be float in cm
- Height 2 must be float in cm
- Area of Container must be float in cm square
- Area of glass cross section must be float in cm square

**Constant Head Soil Permeability Test**

- Sample Number must be Integer
- Amount of filtered of water from the sample: must be flout in cm cube
- Sample Length must be flout in cm
- cross sectional area must be flout in cm square
- the time required for leak must be flout in second
- difference between water height in both tubes: must be flout in cm

**Laboratory Compaction Test**

- Sample Number must be Integer
- weight of cylinder container and base must be float in gram
- weight of cylinder container, base and the soil inside it. must be float in gram
- volume of container must be float in $cm^3$
- moisture content must be float

**Unconfined Compression Test of Soil**

- Sample Number must be Integer

- Compression Power: must be float in KNewton
- Initial cross sectional area: must be float in cm square
- Initial Height: must be float in mm
- difference in height must be float in mm

## 4.4 Database Design:

The following ER diagram shows the relationship between the tables and the cardinality ratio for each relation:

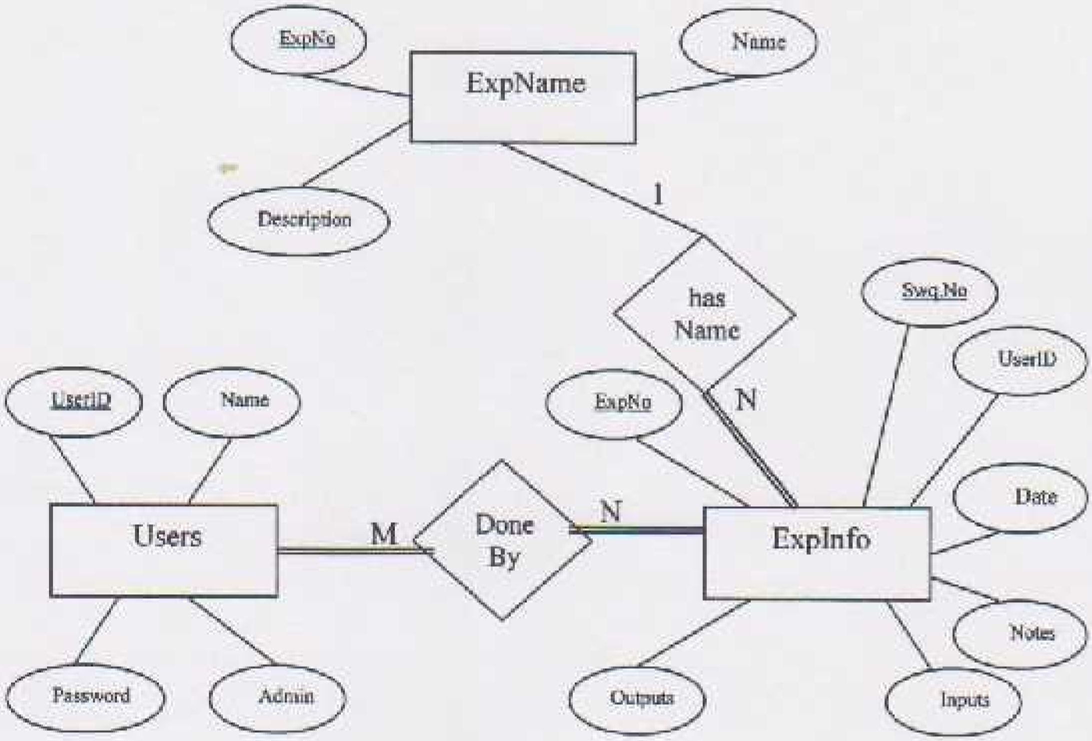


Figure 4.36 Database Design (ER-Diagram)

The ExpInfo represents 13 different tables that relate with the other two tables (Users and ExpName) in the same way as shown above in the conceptual entity relation diagram.

## 4.5 Functional Design:

This section analyzes the Database that we will in our software in order to store results of experiments and retrieve them later when needed, also it includes the ER Diagram that shows the entity relationship of the tables.

the data base will consist of 15 tables as shown below:

**Exp_Name table**

| Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|
| [Experiment ID] | numeric | 9 | |
| [Experiment name] | varchar | 100 | |
| Description | varchar | 1000 | ✓ |

Table 4.1 Exp_Name

**Users table**

| Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|
| userid | numeric | 9 | |
| username | varchar | 100 | |
| userpass | char | 8 | |
| admin | binary | 2 | |

Table 4.2 Users

**Water or Moisture Content table**

| Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|
| Exp_ID | numeric | 9 | |
| [Sequential Number] | numeric | 9 | |
| [User ID] | numeric | 9 | |
| [Date] | datetime | 8 | |
| Notes | varchar | 1000 | ✓ |
| [The weight of empty containe | decimal | 9 | |
| [The weight of container and s | decimal | 9 | |
| [weight of container and dry s | decimal | 9 | |
| [Water or moisture content] | decimal | 9 | |

Table 4.3 Water or Moisture Content

**Specific gravity determination using Pycnometer table**

| Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|
| Exp_ID | numeric | 9 | |
| [Sequential Number] | numeric | 9 | |
| [User ID] | numeric | 9 | |
| [Date] | datetime | 8 | |
| Notes | varchar | 1000 | ✓ |
| [The weight of empty | decimal | 9 | |
| [The weight of contair | decimal | 9 | |
| [weight of container a | decimal | 9 | |
| [weight of container fi | decimal | 9 | |
| [Specific Gravity of So | decimal | 9 | |

Table 4.4 Specific gravity determination using Pycnometer

**Bulk Density determination Using Core Cutter table**

| | Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|---|
| | Exp_ID | numeric | 9 | |
| | [Sequential Number] | numeric | 9 | |
| | [User ID] | numeric | 9 | |
| | [Date] | datetime | 8 | |
| | Notes | varchar | 1000 | ✓ |
| | [Diameter of ring or c) | decimal | 9 | |
| | [Height of ring or cylin | decimal | 9 | |
| | [The weight of empty | decimal | 9 | |
| | [weight of soil sample] | decimal | 9 | |
| | [weight of one piece ς | decimal | 9 | |
| | [weight of the other p | decimal | 9 | |
| | [Bulk Density] | decimal | 9 | |

Table 4.5 Bulk Density determination Using Core Cutter

**Bulk Density determination Using Immersion in water table**

| | Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|---|
| | Exp_ID | numeric | 9 | |
| | [Sequential Number] | numeric | 9 | |
| | [User ID] | numeric | 9 | |
| | [Date] | datetime | 8 | |
| | Notes | varchar | 1000 | ✓ |
| | [Sample weight] | decimal | 9 | |
| | [The sample weight cc | decimal | 9 | |
| | [Sample weight tipped | decimal | 9 | |
| | [Density of paraffin.] | decimal | 9 | |
| | [Bulk Density] | decimal | 9 | |

Table 4.6 Bulk Density determination Using Immersion in water

**Bulk Density determination Using Direct Measurement table**

| | Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|---|
| | Exp_ID | numeric | 9 | |
| | [Sequential Number] | numeric | 9 | |
| | [User ID] | numeric | 9 | |
| | [Date] | datetime | 8 | |
| | Notes | varchar | 1000 | ✓ |
| | [Sample weight.] | decimal | 9 | |
| | [length of container] | decimal | 9 | |
| | [Container width] | decimal | 9 | |
| | [Container height.] | decimal | 9 | |
| | [Bulk Density] | decimal | 9 | |

Table 4.7 Bulk Density determination Using Direct Measurement

**Tri Axial Compression and Shear Test table**

| | Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|---|
| | Exp_ID | numeric | 9 | |
| | [Sequential Number] | numeric | 9 | |
| | [User ID] | numeric | 9 | |
| | [Date] | datetime | 8 | |
| | Notes | varchar | 1000 | ✓ |
| | [Initial length.] | decimal | 9 | |
| | [Sample diameter.] | decimal | 9 | |
| | [Initial Sample Area.] | decimal | 9 | |
| | [Vertical Shear at any | decimal | 9 | |
| | [Vertical Pressure at f. | decimal | 9 | |
| | [Sample Number] | decimal | 9 | |
| | cohesion | decimal | 9 | |
| | [angle of internal fricti | decimal | 9 | |
| | [shear test] | decimal | 9 | |

Table 4.8 Tri Axial Compression and Shear Test

**Direct Shear Test of Soil table**

| | Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|---|
| | Exp_ID | numeric | 9 | |
| | [Sequential Number] | numeric | 9 | |
| | [User ID] | numeric | 9 | |
| | [Date] | datetime | 8 | |
| | Notes | varchar | 1000 | ✓ |
| | [Vertical stress on the | decimal | 9 | |
| | [Surface Area] | decimal | 9 | |
| | [Time] | decimal | 9 | |
| | [Sample Number] | decimal | 9 | |
| | [Vertical stress] | decimal | 9 | |
| | [Shear Stress] | decimal | 9 | |
| | cohesion | decimal | 9 | |

Table 4.9 Direct Shear Test of Soil

**Liquid Limit Determination Using Casagrand Method table**

| | Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|---|
| | Exp_ID | numeric | 9 | |
| | [Sequential Number] | numeric | 9 | |
| | [User ID] | numeric | 9 | |
| | [Date] | datetime | 8 | |
| | Notes | varchar | 1000 | ✓ |
| | [Number of Hits] | numeric | 9 | |
| | [Number of Sample] | numeric | 9 | |
| | [Number of container] | numeric | 9 | |
| | [Weight of wet sample | decimal | 9 | |
| | [Weight of dry sample | decimal | 9 | |
| | [Weight of container] | decimal | 9 | |
| | [Average Plastic Limit] | decimal | 9 | |
| | [Liquid Limit] | decimal | 9 | |

Table 4.10 Liquid Limit Determination Using Casagrand Method

**Field Density Test Using Sand Replacement Method table**

| | Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|---|
| | Exp_ID | numeric | 9 | |
| | [Sequential Number] | numeric | 9 | |
| | [User ID] | numeric | 9 | |
| | [Date] | datetime | 8 | |
| | Notes | varchar | 1000 | ✓ |
| | [Sample Number] | numeric | 9 | |
| | [weight of wet soil tak | decimal | 9 | |
| | [weight of dry soil tak | decimal | 9 | |
| | [weight of empty cone | decimal | 9 | |
| | [weight of sand that f | decimal | 9 | |
| | [weight of sand that f | decimal | 9 | |
| | [Sand density] | decimal | 9 | |
| | [wet soil density] | decimal | 9 | |
| | [dry soil density] | decimal | 9 | |

Table 4.11 Field Density Test Using Sand Replacement Method

**Falling head Soil Permeability test table**

| | Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|---|
| | Exp_ID | numeric | 9 | |
| | [Sequential Number] | numeric | 9 | |
| | [User ID] | numeric | 9 | |
| | [Date] | datetime | 8 | |
| | Notes | varchar | 1000 | ✓ |
| | [Sample Number] | numeric | 9 | |
| | [Sample Length] | decimal | 9 | |
| | [Time] | decimal | 9 | |
| | [Height before] | decimal | 9 | |
| | [Height 2] | decimal | 9 | |
| | [Area of Container] | decimal | 9 | |
| | [Area of glass cross s | decimal | 9 | |
| | [Permeability coefficie | decimal | 9 | |

Table 4.12 Falling head Soil Permeability test

**Constant Head Soil Permeability Test table**

| Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|
| Exp_ID | numeric | 9 | |
| [Sequential Number] | numeric | 9 | |
| [User ID] | numeric | 9 | |
| [Date] | datetime | 8 | |
| Notes | varchar | 50 | ✓ |
| [Sample Number] | numeric | 9 | |
| [Amount of filtered wa | decimal | 9 | |
| [Sample Length] | decimal | 9 | |
| [Cross sectional area] | decimal | 9 | |
| [The time required for | decimal | 9 | |
| [Difference between | decimal | 9 | |
| [Permeability coefficie | decimal | 9 | |

Table 4.13 Constant Head Soil Permeability Test

**Laboratory Compaction Test table**

| Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|
| Exp_ID | numeric | 9 | |
| [Sequential Number] | numeric | 9 | |
| [User ID] | numeric | 9 | |
| [Date] | datetime | 8 | |
| Notes | varchar | 1000 | ✓ |
| [Sample Number] | numeric | 9 | |
| [Weight of cylinder co | decimal | 9 | |
| [Weight of cylinder co | decimal | 9 | |
| [Volume of container. | decimal | 9 | |
| [Moisture content.] | decimal | 9 | |
| [wet soil density] | decimal | 9 | |
| [dry soil density] | decimal | 9 | |

Table 4.14 Laboratory Compaction Test

**Unconfined Compression Test Of Soil table**

| Column Name | Data Type | Length | Allow Nulls |
|---|---|---|---|
| Exp_ID | numeric | 9 | |
| [Sequential Number] | numeric | 9 | |
| [User ID] | numeric | 9 | |
| [Date] | datetime | 8 | |
| Notes | varchar | 1000 | ✓ |
| [Sample Number] | numeric | 9 | |
| [Compression Power] | decimal | 9 | |
| [Initial cross sectional | decimal | 9 | |
| [Initial Height] | decimal | 9 | |
| [Difference in height] | decimal | 9 | |
| [Stress coefficient] | decimal | 9 | |
| Strain | decimal | 9 | |

Table 4.15 Unconfined Compression Test Of Soil

## 4.6 Summery:

In this chapter we had designed the input and output screens needed for the Geotechnical Software, and illustrates on each one the input and output variables and the validation required for each input, also we explain this as flowcharts.

Also we did the Database Design, Showing the tables needed for the software and by using the ER diagram we show the primary key for each table and the relationship between these tables.

As a result of these analyses, and after the approval of our instructor and Civil Engineering Department Supervisor, we recommend going on working on the next chapter which is the Coding and Implementation.

# Chapter Five

# Coding and Implementation

**5.1 Introduction**

**5.2 Coding Programming Language:**

**5.3 Database System**

**5.4 Establishment of Development Environment.**

**5.5 Database Creation and Configuration**

**5.6 Coding and Unit Testing**

**5.7 Summary and Recommendations**

# Coding and Implementation

## 5.1 Introduction

This chapter illustrates the coding and implementation of the system, where the environment of the coding process and implementation will be determined, each function will be implemented and coded, the database and its relations, constraints and keys will be implemented.

## 5.2 Coding Programming Language:

There are different types of programming languages that can be used in coding the Geotechnical software, the work team decided to use the Visual Basic 6.0 programming language as front end and the SQL Server 2000 as back end to save the results of the performed experiment.

### 5.2.1 Visual Basic 6.0:

Why Visual Basic 6.0 Designer?

Because of:

- Reduced Development Time.
- Reduced Development Cost.
- Ease of use.
- High Compatibility with Windows Environment.
- VB needs less resources than VB.Net
- Work team experience
- Shortage of time and big project requirements did not allow learning new language.
- VB.6.0 contents convenient for the project.
- We do not need the features of .Net technology.

## 5.3 Database System

-Structured Query Language (SQL)

SQL is a database sublanguage used in queering, updating, and margining relational database. Moreover, it is not like the same sense as other programming language such as C or Pascal, SQL can either be use in formulating interactive queries or be embedded in an application as instruction for handling data. SQL also contains component for defining, controlling, altering and securing data. SQL provide easy and flexibility way for using it, so both technical and non-technical users can use it.

SQL server 2000 which used in database implementation have more thane one features which strongly support the database configurations management, this feature conceder the reason for which the SQL server 2000 conceder as the most popular and reliable database application, some of this feature which use in this system is:

### Enterprise manger:

It is an administrative application that provides split-screens representation of SQL server management capability, it is display the console tree and tools, and it is used to create new SQL server group, databases, options and property for each SQL server component.

### 1. Database Diagram:

Diagram are used to constructing actual database, it is concise and understandable representation of ER data model.

The figure below shows the database diagram for the TriAxial Shear Test Experiment.

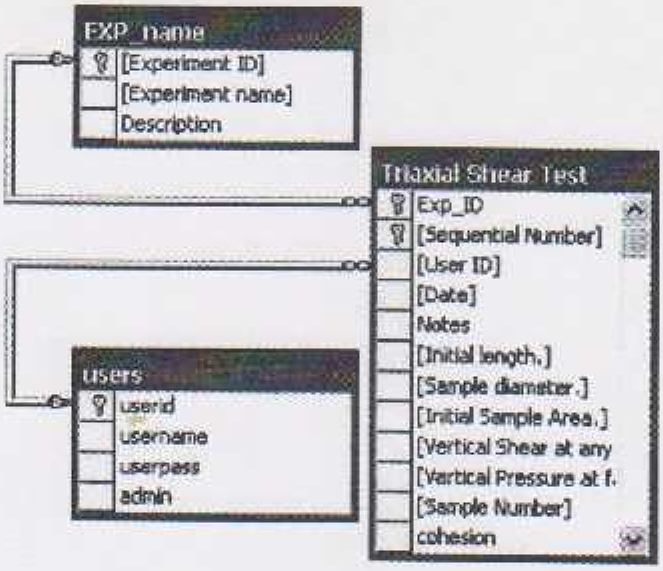


Figure 5.1 Database Diagram

### 2. Security:

SQL server 2000 allows making user and granting them a specific or data basing administrator privileges in order to control the access of the database resource according to the nature of migration.

### 3. Migration:

It is the process of moving one database information to another in the same application or from one database application to another.

## The implementation of the SQL server 2000

1. Installation of SQL server 2000

2. Change password of the default user account.

3. Create the user account to use it instead of default sa account.

**Establish of user account**

- Administrator user account.
- Public user account
- Create database

## 5.4 Establishment of Development Environment.

### 5.4.1 Hardware Needed:

The minimum Hardware component to help in the implementation:

- Pentium4 PC of these features:
  - Speed 1800MHz, 256MB RAM, 40GB H.D.
- Two flash memories.
- Printer.
- Input and output devices (keyboard, mouse, screen).

### 5.4.2 Software Environment.

The software development environment consists of the following:

- **Windows Operating System (Windows XP professional):**

  The application which is used to develop this system, is Visual Basic 6.0 require an operating system to be usable, the XP operating system is one of the best operating systems which could be used beside this, XP provide the following features which consider as important issue to run the ASP.net:

-Operating System Implementation

Starting the implementation of the software by installing windows XP professional, because it have capabilities to provide many service to client that is important to implementation the other software needed.

-Windows XP professional installing:

Steps for installing:

1. Run setup program.
2. Run setup wizard.
3. Installing windows XP professional networking.
4. Complete the setup program.

- **Office XP:**

  After installing windows XP it was chosen suitable office which was installed, that help to write documentation in word and drown some figures on power point.

- **VB 6.0**

  This software was used to program all screens which used in this project that include the screens for the experiments inputs, outputs, help, print and save screens.

  Also this software was done very important part of this project which is classification and compare.

- **SQL Server 2000**

Why Microsoft SQL Server?

Because:

- High Compatibility with Microsoft Visual Basic.

- Special Support for Microsoft SQL Server in ASP.NET.

- Fit the size of the project since huge database is not required.

SQL Server Configurations: SQL Server was configured with the default configurations except security and authentication which becomes windows authentication only and ASPNET user account was added.

Changing authentication method.

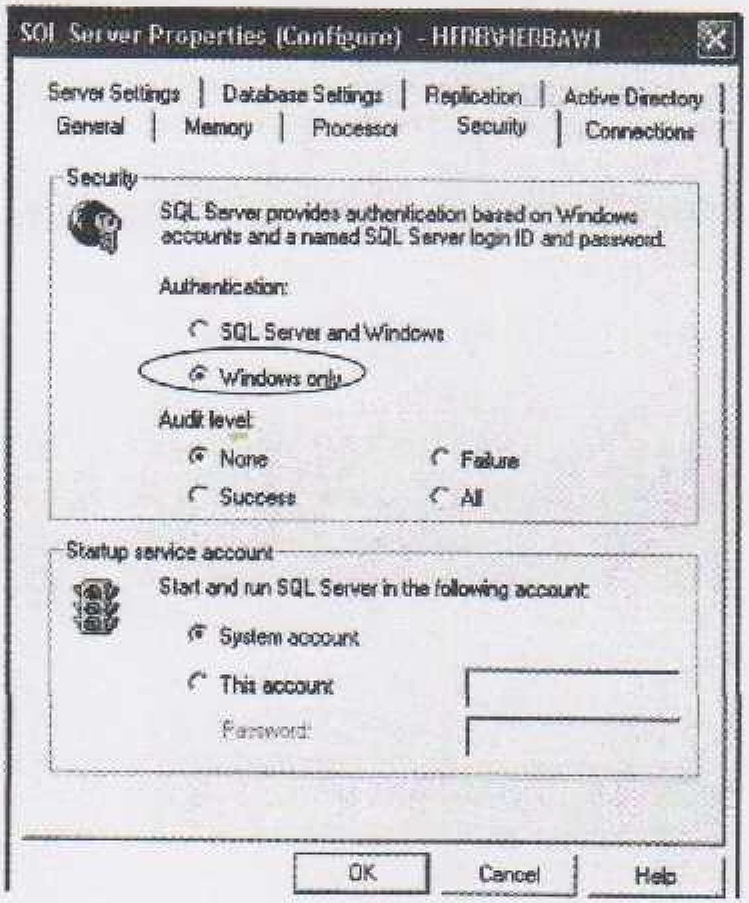


Figure 5.2 Changing authentication method

## 5.5 Database Creation and Configuration

Data base in this project are stored in these tables:

- **Login:** This table contains user_id, User_Name, Password, Logged_in, Admin.
- **Experiments:** This table contains the following fields:

  Exp_Number, Exp_Name, Description.
- **Exp_Info:** This Table contains the following fields:

  Exp_Number, User_ID, date, Notes, Inputs, outputs.

These tables were created by the SQL Server 2000 according to the following steps:

1. open the SQL Server 2000 by Clicking the Enterprise Manager as shown in the Figure below:

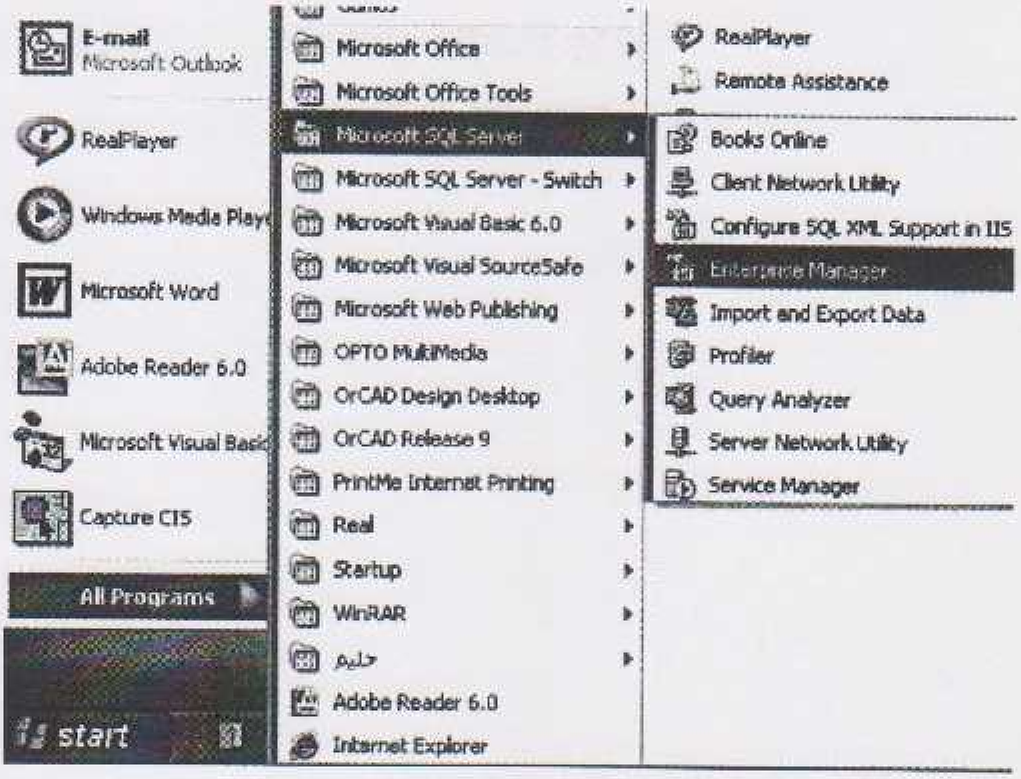


figure 5.3 Opening the SQL Server 2000

2. Open the Database folder

3. Create a new Database by right click on the data base folder and choose New Database, and fill in the Database name as follows:

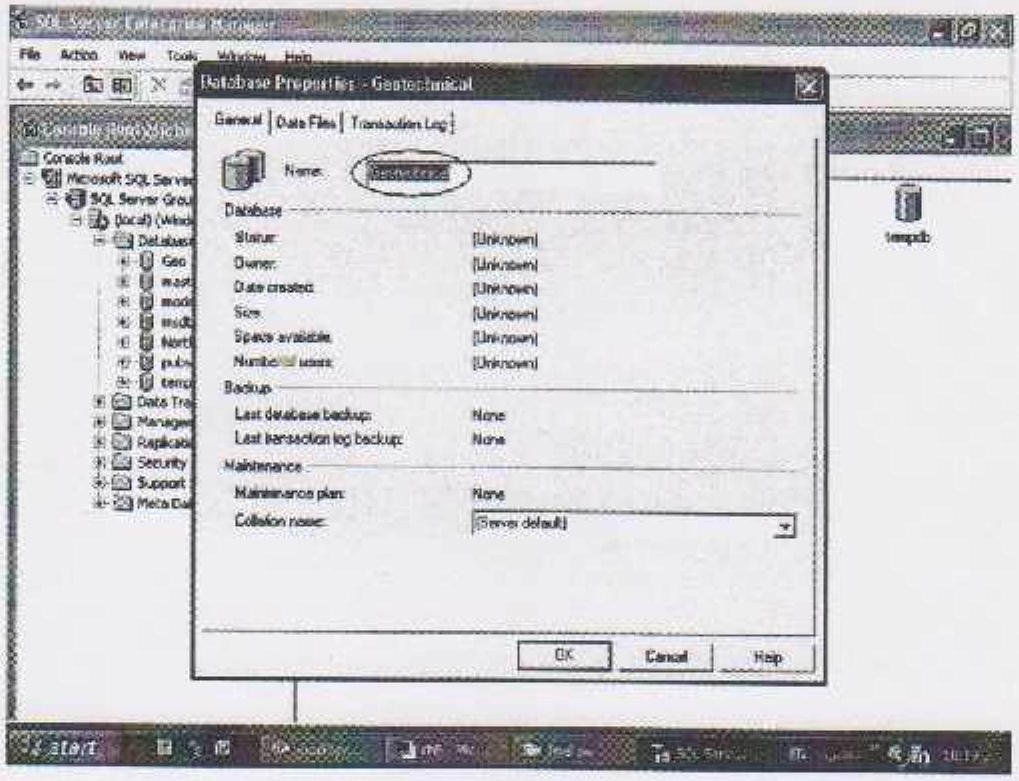

Figure 5.4 Creating New Database

4.Then we create the tables we need inside the Database as shown below:

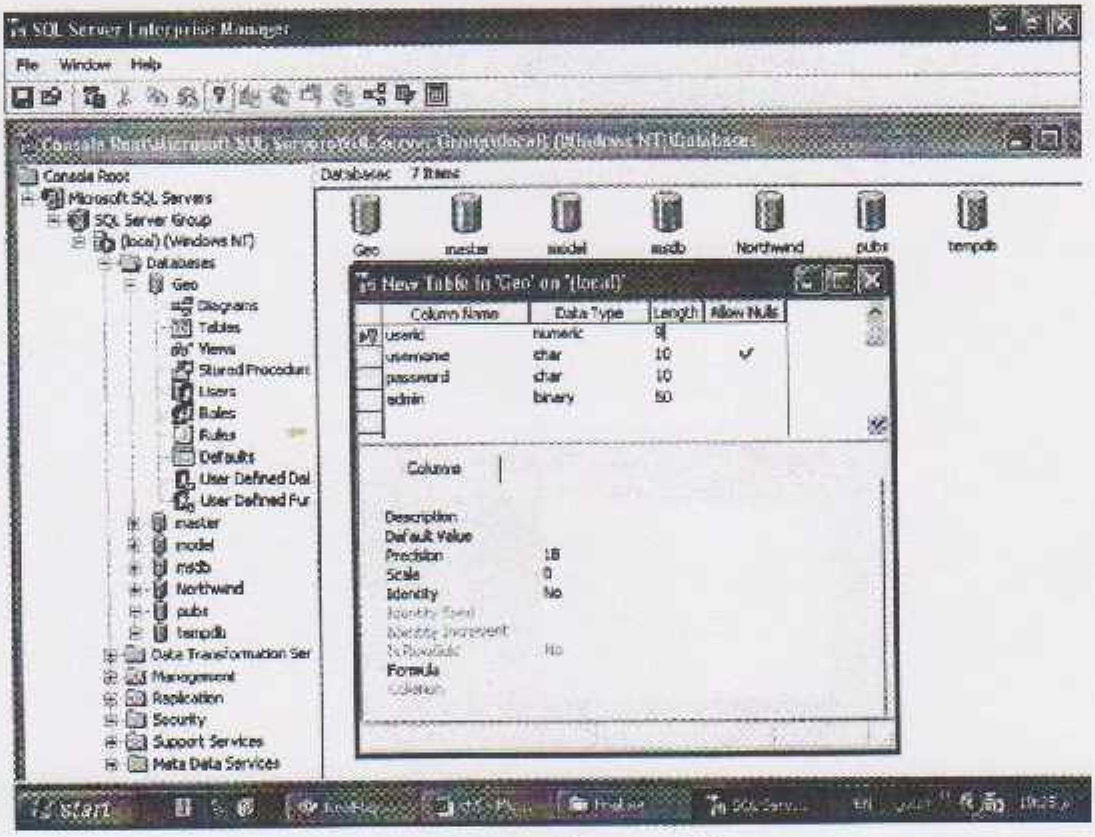


Figure 5.5 Creating a new Table

3. Finally, we fill the table fields and save it with the name we want.

## 5.6 Coding and Unit Testing

The screens of the software were designed successfully, and the following figures show the final view of these screens:

We design the screens in VB.6.0 that is shown below:


Figure5.6 Login screen

Function allow the user to enter and use the Program

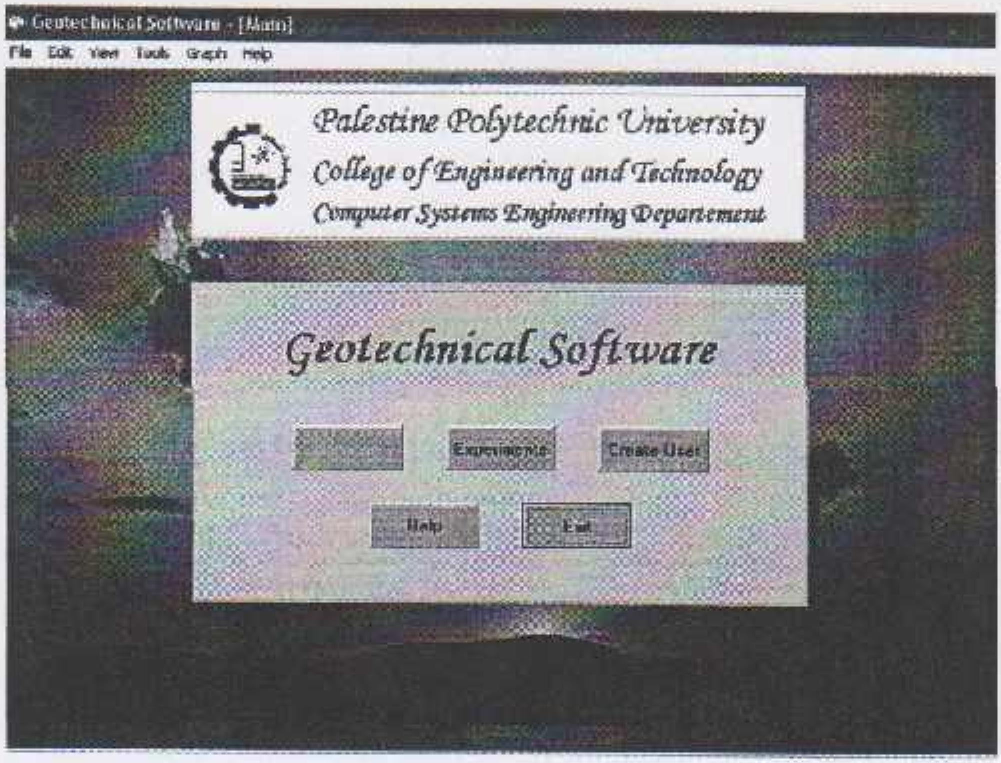


Figure5.7Main screen

This Screen Gives the Main Options for the user as ordinal and administrator user

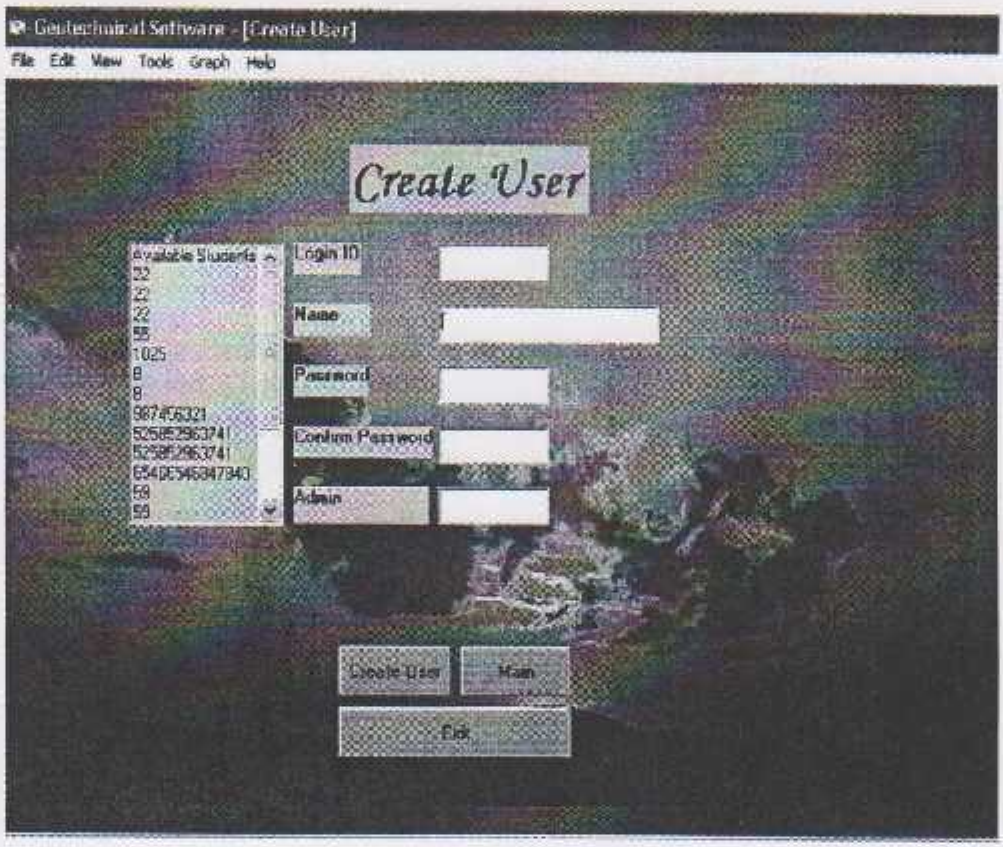


Figure5.8 create user screen

This Function allow the Administrator to create a new user

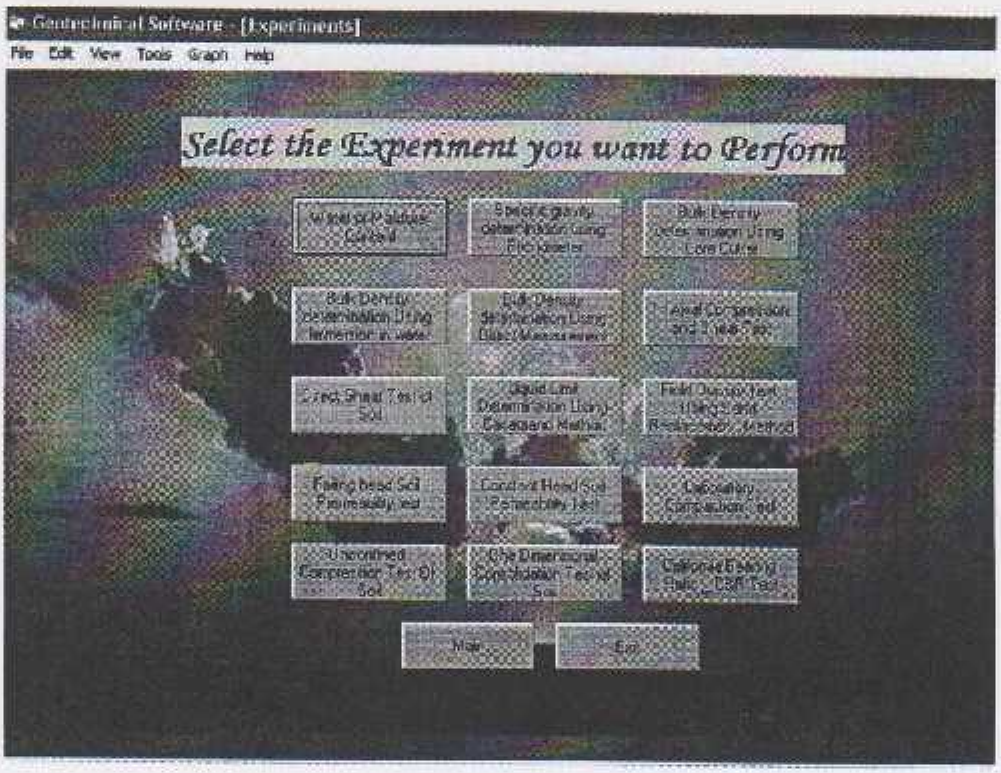


Figure5.9 Experiments screen

This Screen Gives the Experiments that are supported by the Geotechnical Software.


Figure5.10 Water or moisture Content screen

Water or moisture Content: this function enables the user to calculate the water or moisture content

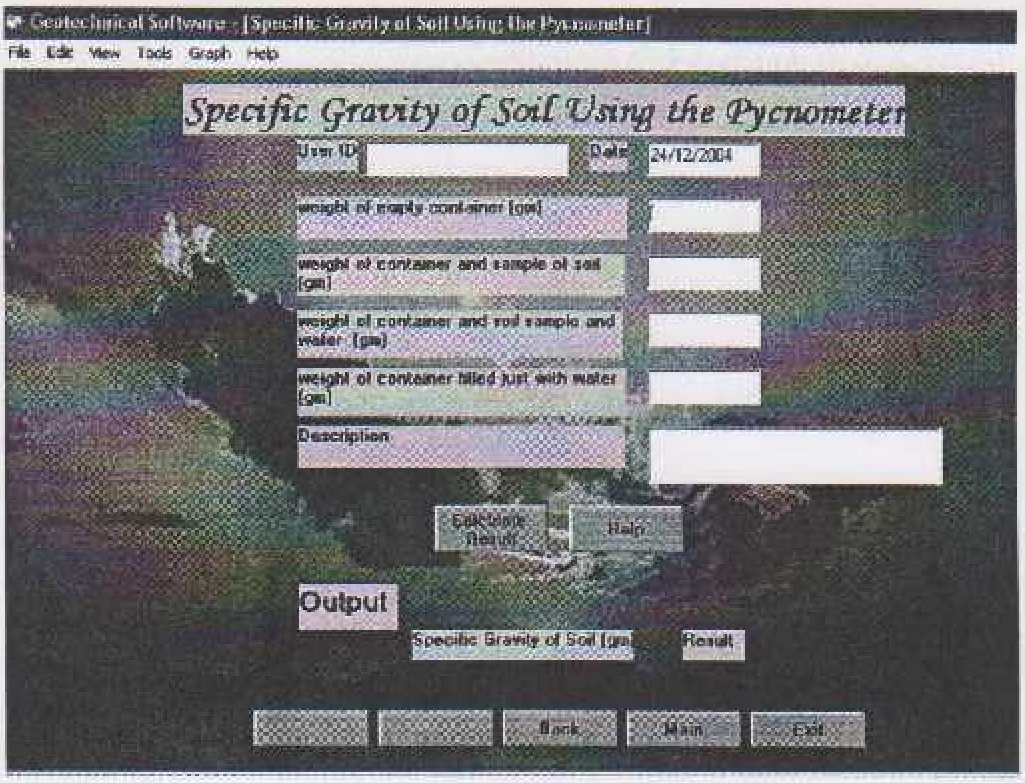


Figure5.11 Specific gravity determination using Pycnometer screen

Specific Gravity of Soil Using the Pycnometer: this function enables the user to determine Specific Gravity of Soil Using the Pycnomete

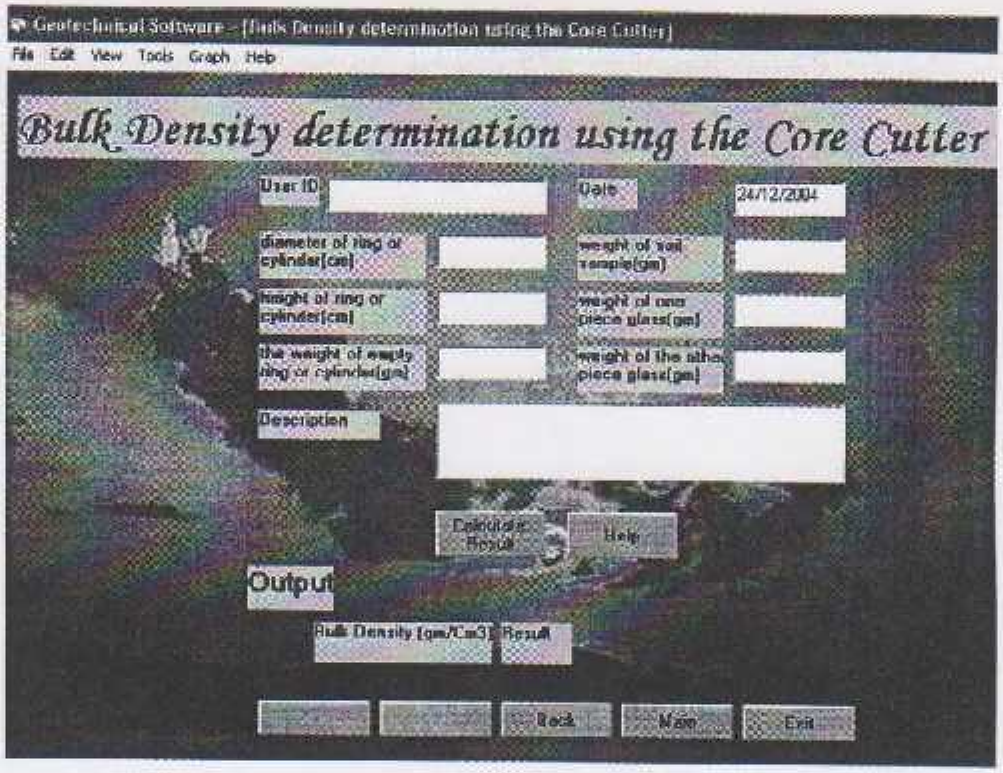


Figure5.12 Bulk Density determination Using Core Cutter screen

Bulk Density determination using the Core Cutter: this function enables the user to determine Bulk Density determination using the Core Cutter

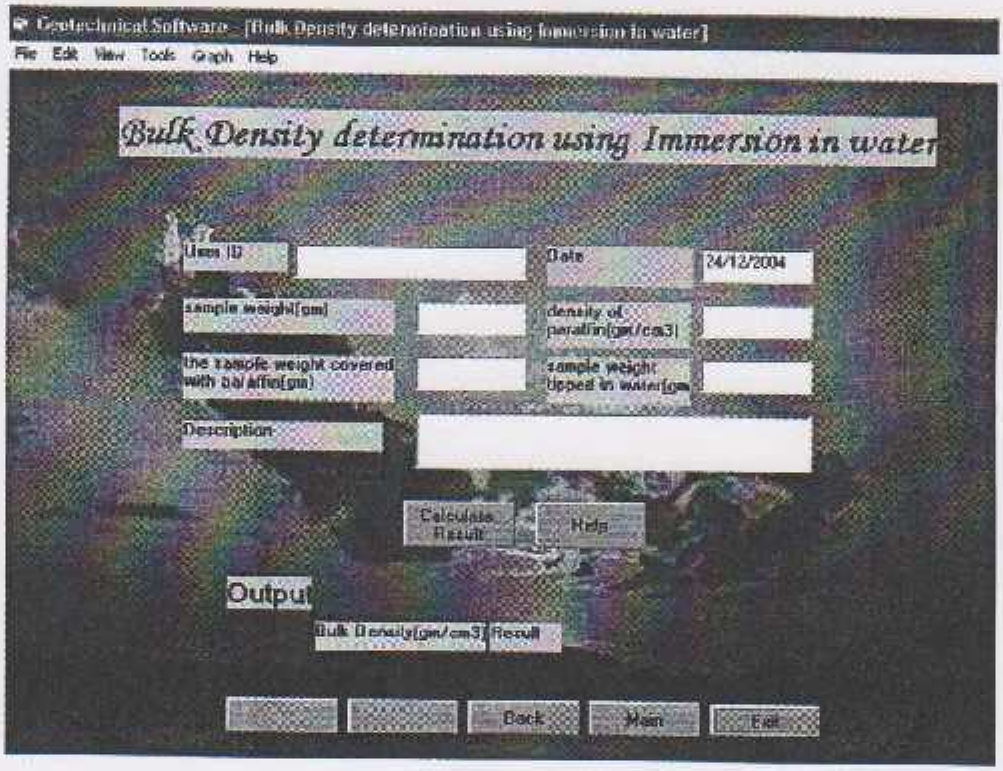


Figure5.13 Bulk Density determination Using Immersion in water screen

Bulk Density determination using Immersion in water: this function enables the user to determine Bulk Density determination using Immersion in water

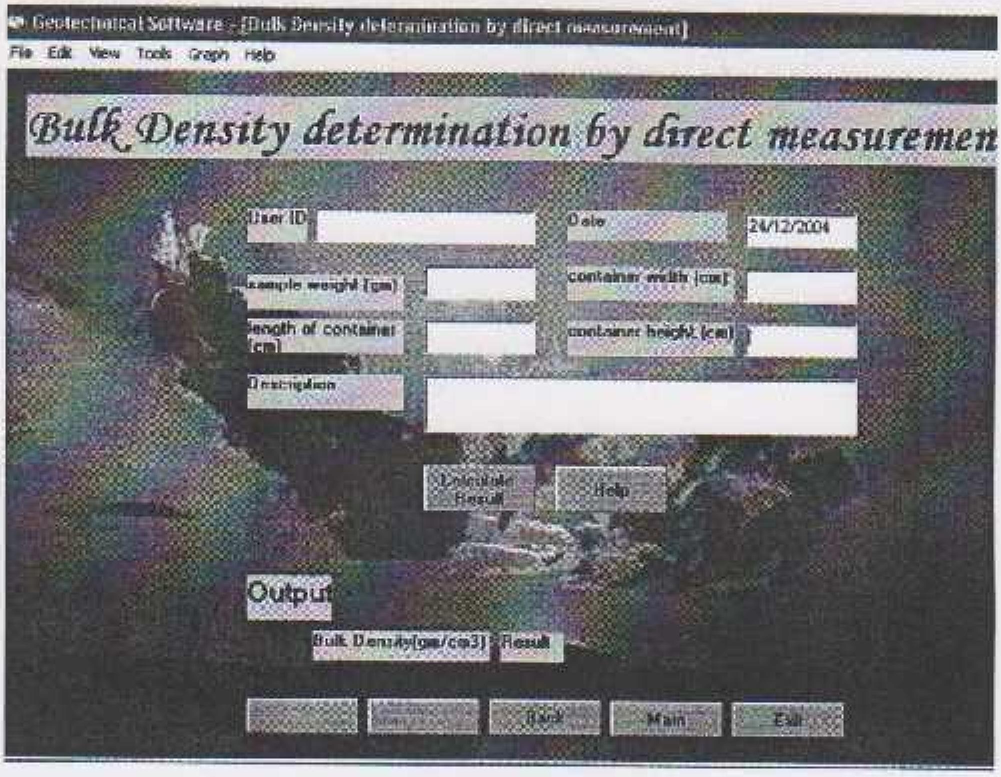


Figure5.14 Bulk Density determination Using Direct Measurement screen

Bulk Density determination by direct measurement: this function enables the user to determine Bulk Density by direct measurement

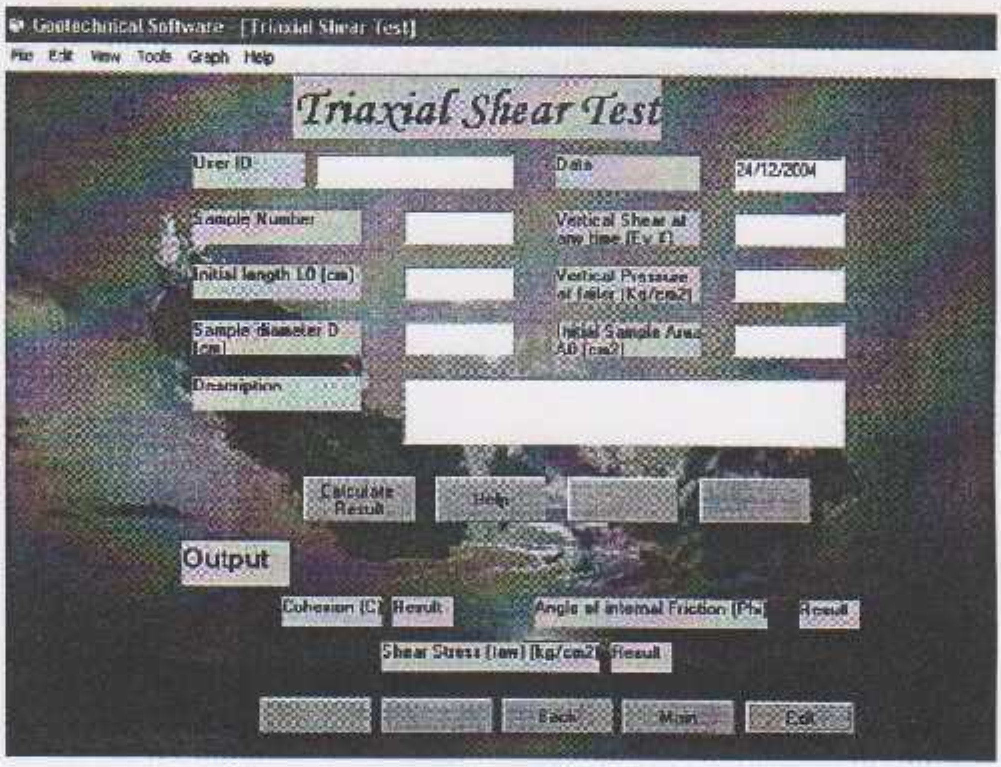


Figure5.15 Tri Axial Compression and Shear Test screen

Triaxial Shear Test: this function enables the user to determine Triaxial Shear
Initial Length: must be float in mm

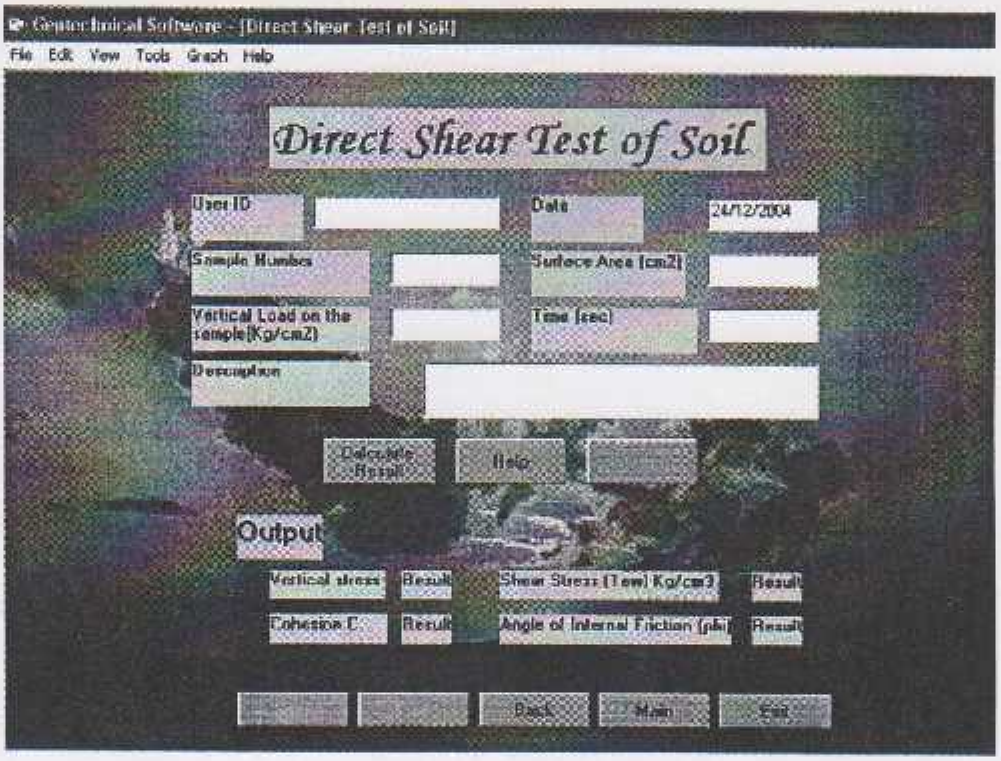


Figure5.16 Direct Shear Test of Soil screen

Direct Shear Test of Soil: this function enables the user to determine the Direct Shear of Soil

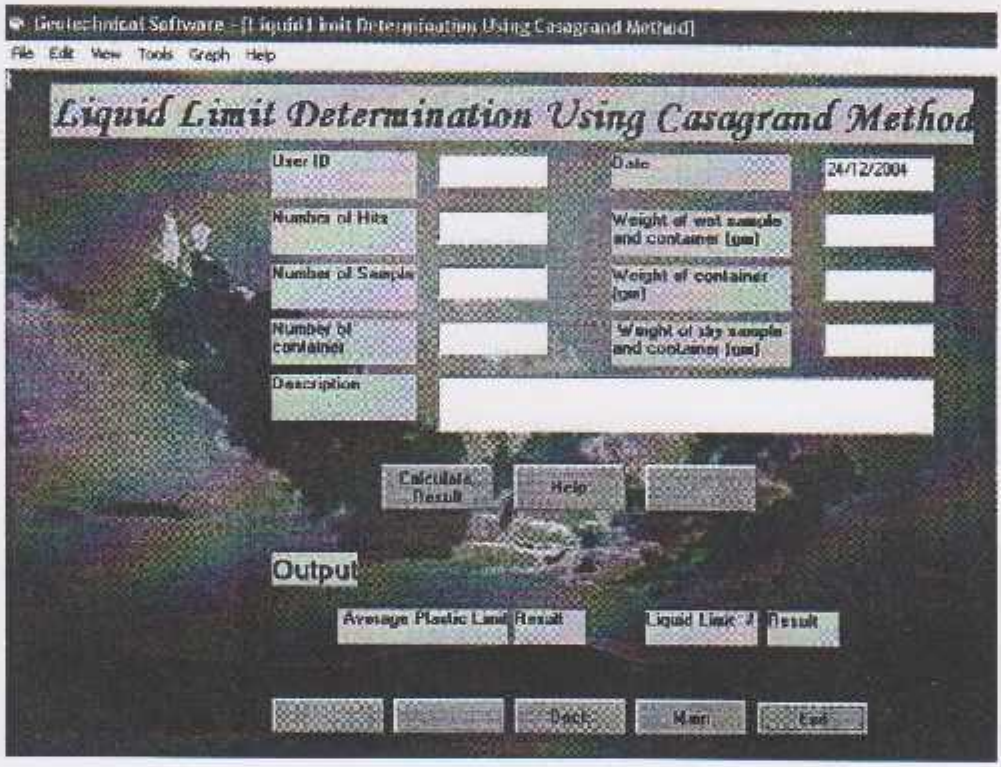


Figure5.17 Liquid Limit Determination Using Casagrand Method screen

Liquid Limit Determination Using Casagrand Method: this function enables the user to determine Liquid Limit Using Casagrand Method

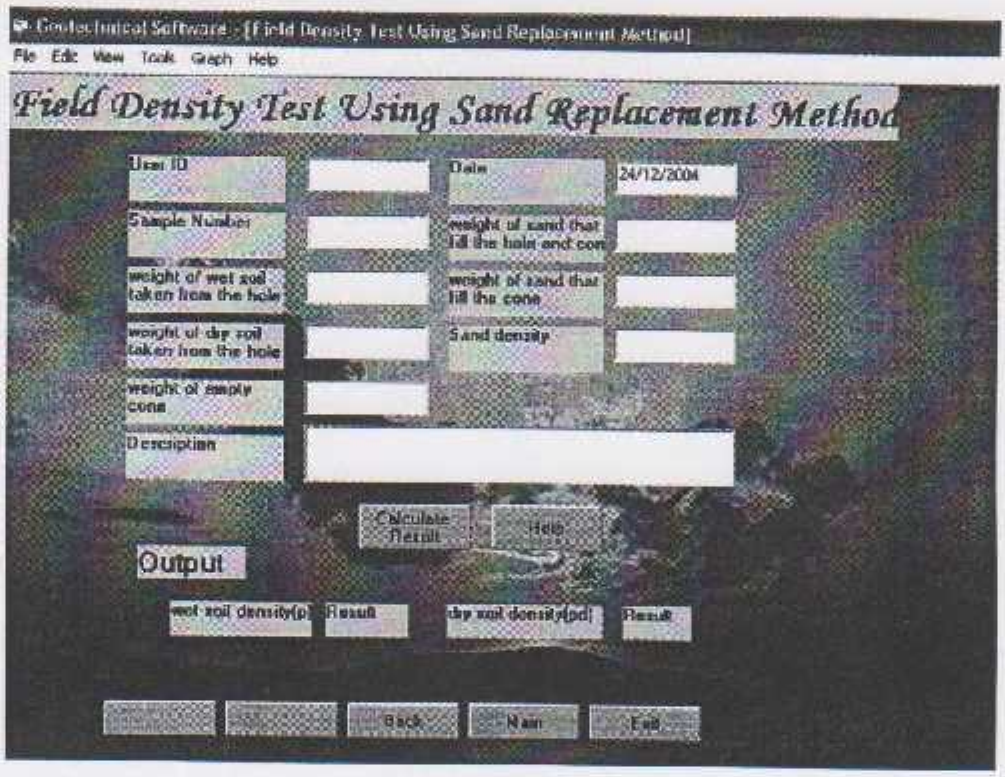


Figure5.18 Field Density Test Using Sand Replacement Method screen

Field Density Test Sand Replacement Method: this function enables the user to determine Field Density Test Using Sand Replacement Method

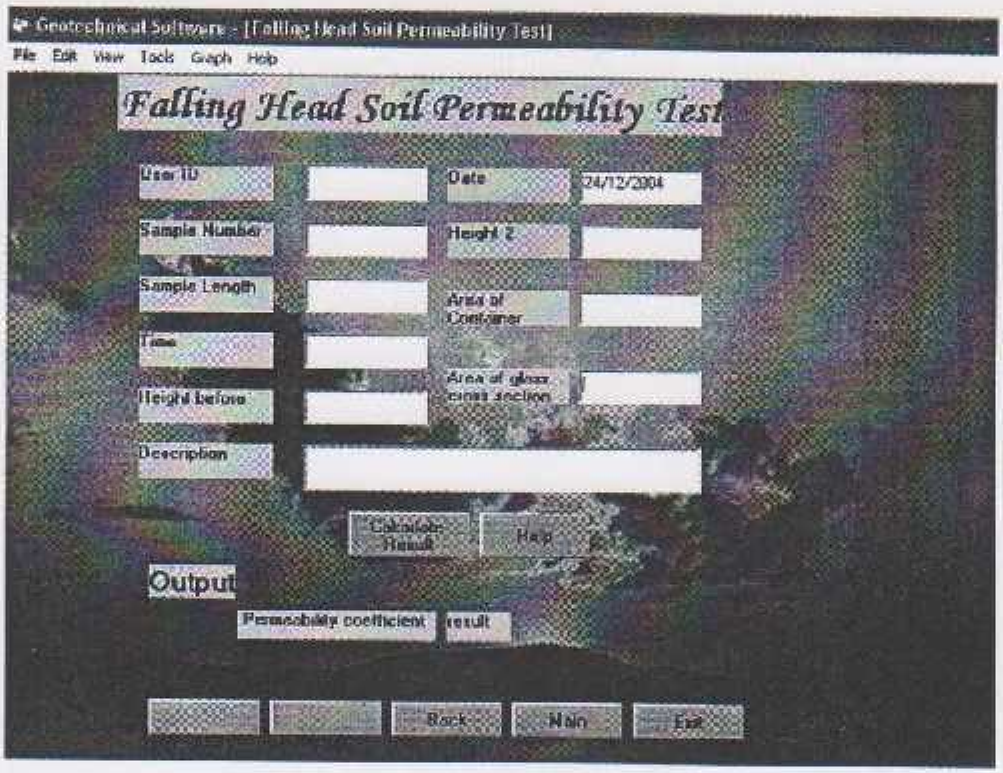


Figure5.19 Falling head Soil Permeability test screen

Falling Head Soil Permeability Test: this function enables the user to determine Soil Permeability using Falling Head test.

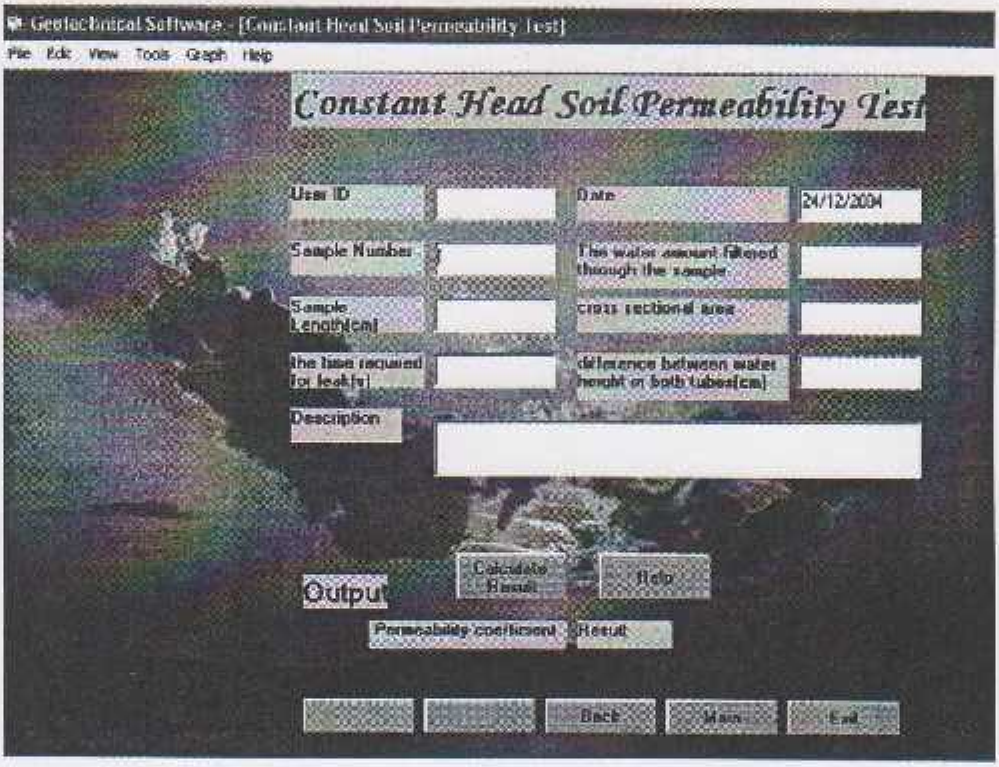


Figure5.20 Constant Head Soil Permeability Test screen

Constant Head Soil Permeability Test: this function enables the user to determine Soil Permeability using Constant Head test.

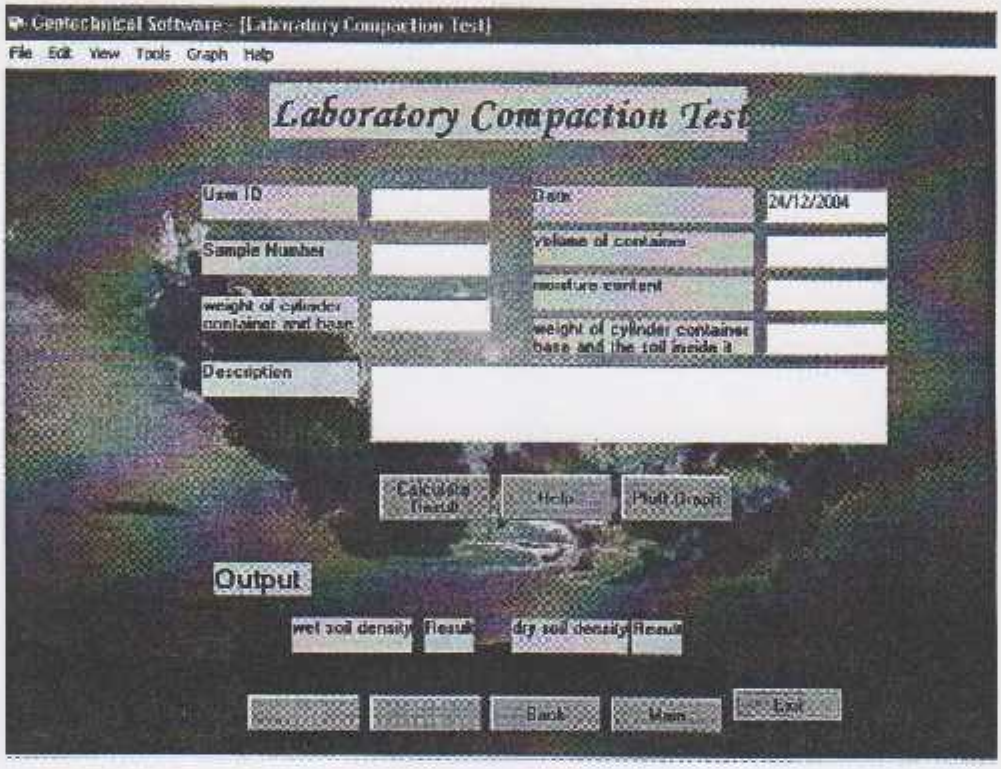


Figure5.21 Laboratory Compaction Test screen

Laboratory Compaction Test: this function enables the user to determine Compaction in the laboratory

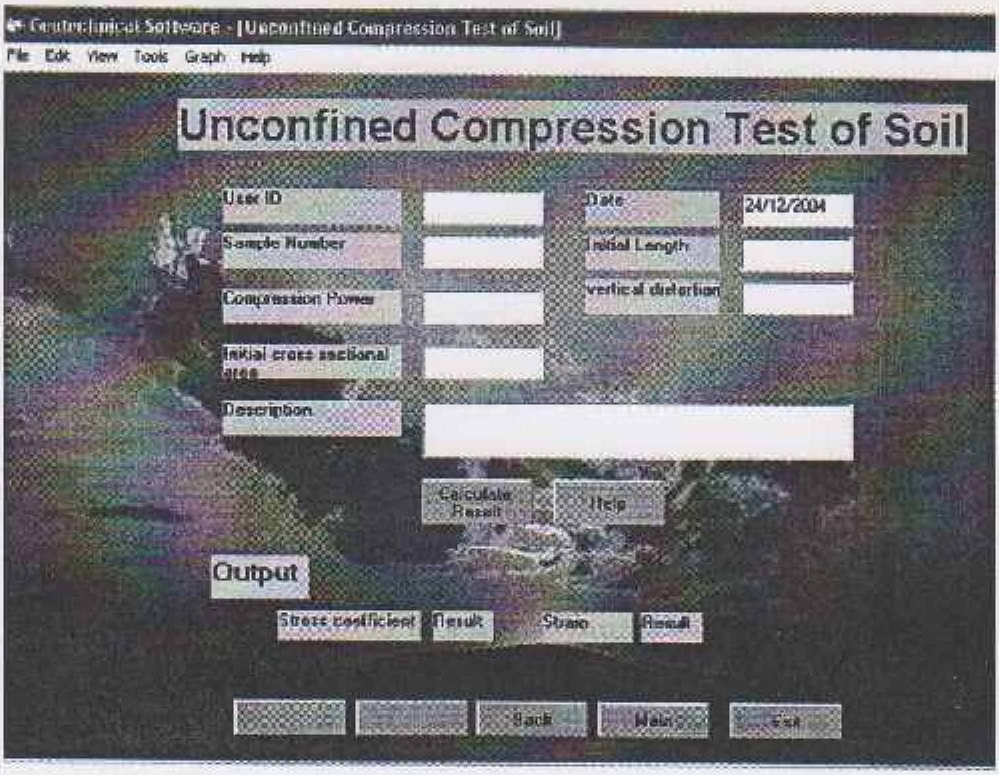


Figure5.22 Unconfined Compression Test of Soil screen

Unconfined Compression Test of Soil: this function enables the user to determine the shape of the sample at failure

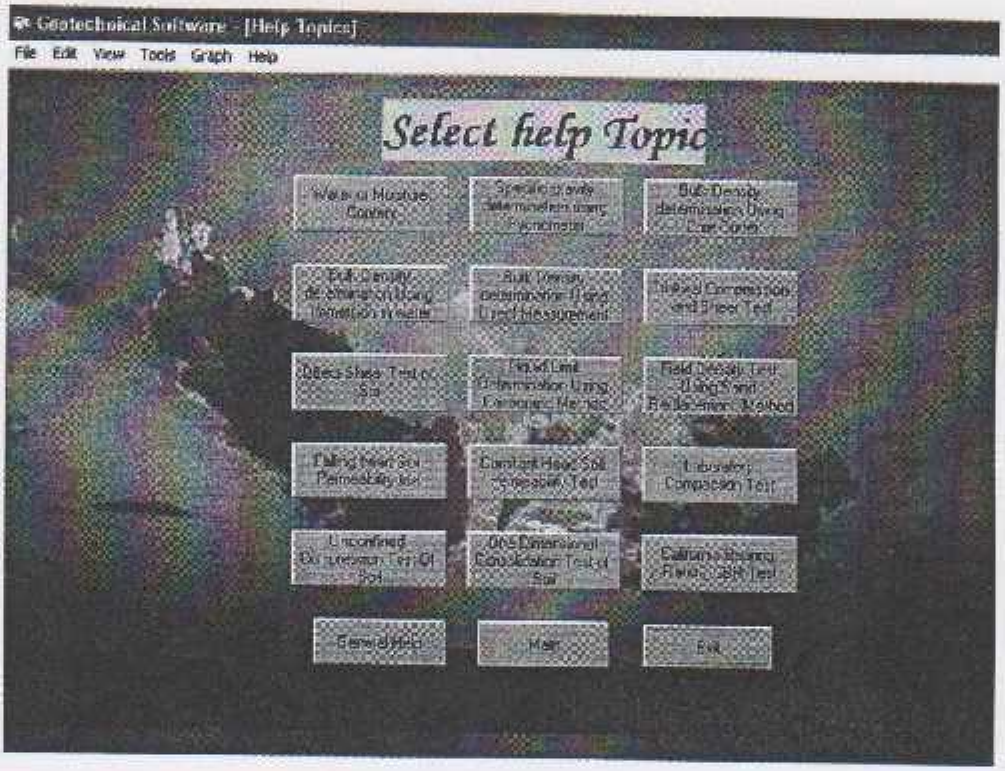


Figure5.23 Help screen

This Screen Gives Help Topics on the Experiments that are supported by the Geotechnical Software.

## 5.7 Summary:

In this chapter the program is coded by the Visual Basic 6.0 Programming language, and by using the SQL Server 2000 to store the results of the experiments.

Now the system is ready to be tested before the use, which will be done by the Civil Engineering Department.

# Chapter Six

# Testing

## 6.1 Introduction

After the software has been programmed and a unit testing applied on the program, it is now ready to be tested before it can be used by the user (Civil Engineering Department), by using the program in order to check it for errors.

## 6.2 Testing Plan:

The testing will be done by checking all the system screens, validations and functions. The testing plan was put on to be performed in several steps, as follow:

- Testing the login Screen.
- Testing the Create user Screen
- Testing the Experiments one by one.
- Testing the Graph Screens.
- Testing the Retrieve Experiment.

The Testing firstly started with the Login screen, by trying to login as an administrator and as a user and observing how the program deal with these cases. After that they create a new user and observe if the program will login or not, and what privileges will be given to that user.

The experiments were checked by filling textboxes with virtual data to observe the response of the program for valid and invalid data.

Save and print results is then checked to determine whether they work properly or not. The save is checked again by retrieving the results that have been saved previously. After that Civil Engineering gives us some notes on the program that we must make.

## 6.3 Testing Plan Results:

The following table shows particular inputs for the system and the expected behavior of the system for these inputs, for each form:

Login Form:

| Function | Inputs | Valid/Invalid |
|---|---|---|
| Login | UserID= poikjh<br>Password = 1236 | Invalid user |
| Login | UserID= 015247<br>Password = 1236 | Valid |

Table 6.1 Login test

Create User Form:

| Function | Inputs | Valid/Invalid |
|---|---|---|
| Create user | UserID= poikjh<br>Name=raef<br>Password = 1236<br>Confirm Password=4568<br>Admin=0 | Invalid UserID<br>Incorrect confirmation of password |
| Create user | UserID= 5456<br>Name=raef<br>Password = 1236<br>Confirm Password=1236<br>Admin=0 | Valid, user created |
| Create user | UserID= 5456<br>Name=raef<br>Password = 1236<br>Confirm Password=1236<br>Admin=10 | Invalid , admin 0, 1 |

Table 6.2 Create User test

Water or moisture Content form:

| Function | Inputs | Valid/Invalid |
|---|---|---|
| Calculate result | the weight of empty container=10<br>the weight of container and sample of soil=20<br>weight of container and dry soil=15 | Valid, well calculate result |
| Calculate result | the weight of empty container=kj<br>the weight of container and sample of soil=20<br>weight of container and dry soil=15 | Invalid, weight must be float |
| Calculate result | the weight of empty container=<br>the weight of container and sample of soil=20<br>weight of container and dry soil=15 | Invalid, fill in data |

Table 6.3 Water or moisture Content test

Specific Gravity of Soil Using the Pycnometer

| Function | Inputs | Valid/Invalid |
|---|---|---|
| Calculate result | the weight of empty container=10<br>the weight of container and sample of soil=20<br>weight of container and soil sample and water=30<br>weight of container filled just with water=20 | Valid, well calculate result |
| Calculate result | the weight of empty container=10<br>the weight of container and sample of soil=ax<br>weight of container and soil sample and water=30<br>weight of container filled just with water=20 | Invalid, weight must be float |
| Calculate result | the weight of empty container=10<br>the weight of container and sample of soil=<br>weight of container and soil sample and water=30<br>weight of container filled just with water=20 | Invalid, fill in data |

Table 6.4 Specific Gravity of Soil Using the Pycnometer test

Bulk Density determination using the Core Cutter:

| Function | Inputs | Valid/Invalid |
|---|---|---|
| Calculate result | *diameter of ring or cylinder=5<br>*height of ring or cylinder=10<br>*the weight of empty ring or cylinder=20<br>*the weight of empty ring or cylinder and the soil and two pieces of Glass=30<br>*weight of one piece glass=3<br>*weight of the other piece glass=2 | Valid, well calculate result |
| Calculate result | *diameter of ring or cylinder=5<br>*height of ring or cylinder=10<br>*the weight of empty ring or cylinder=xy<br>*the weight of empty ring or cylinder and the soil and two pieces of Glass=30<br>*weight of one piece glass=3<br>*weight of the other piece glass=2 | Invalid, weight must be float |
| Calculate result | *diameter of ring or cylinder=5<br>*height of ring or cylinder=10<br>*the weight of empty ring or cylinder=20<br>*the weight of empty ring or cylinder and the soil and two pieces of Glass=<br>*weight of one piece glass=3<br>*weight of the other piece glass=2 | Invalid, fill in data |

Table 6.5 Bulk Density determination using the Core Cutter test

Bulk Density determination using Immersion in water :

| Function | Inputs | Valid/Invalid |
|---|---|---|
| Calculate result | *sample weight=4<br>*the sample weight covered with paraffin=10<br>*sample weight tipped in water=15<br>*density of paraffin=5 | Valid, well calculate result |
| Calculate result | *sample weight=4<br>*the sample weight covered with paraffin=ds<br>*sample weight tipped in water=15<br>*density of paraffin=5 | Invalid, weight must be float |
| Calculate result | *sample weight=4<br>*the sample weight covered with paraffin=10<br>*sample weight tipped in water=<br>*density of paraffin=5 | Invalid, fill in data |

Table 6.6 Bulk Density determination using Immersion in water test

Bulk Density determination by direct measurement:

| Function | Inputs | Valid/Invalid |
|---|---|---|
| Calculate result | * sample weight=4<br>*length of container=7<br>*container width=4<br>*container height=9 | Valid, well calculate result |
| Calculate result | * sample weight=4<br>*length of container=qw<br>*container width=4<br>*container height=9 | Invalid, weight must be float |
| Calculate result | * sample weight=4<br>*length of container=<br>*container width=4<br>*container height=9 | Invalid, fill in data |

Table 6.7 Bulk Density determination by direct measurement test

Triaxial Shear Test

| Function | Inputs | Valid/Invalid |
|---|---|---|
| Calculate result | * Initial length=5<br>*Sample diameter=10<br>*Initial Sample Area=15<br>*Vertical Shear at any time.=11<br>*Vertical Pressure at failure=13 | Valid, well calculate result |
| Calculate result | * Initial length=5<br>*Sample diameter=10<br>*Initial Sample Area=gh<br>*Vertical Shear at any time.=11<br>*Vertical Pressure at failure=13 | Invalid, weight must be float |
| Calculate result | * Initial length=5<br>*Sample diameter=10<br>*Initial Sample Area=<br>*Vertical Shear at any time.=11<br>*Vertical Pressure at failure=13 | Invalid, fill in data |

Table 6.8 Triaxial Shear Test

Direct Shear Test of Soil

| Function | Inputs | Valid/Invalid |
|---|---|---|
| Calculate result | *vertical stress on the sample=140<br>*Surface Area=60<br>*shear stress=120 | Valid, well calculate result |
| Calculate result | *vertical stress on the sample=140<br>*Surface Area=60<br>*shear stress=rt | Invalid, weight must be float |
| Calculate result | *vertical stress on the sample=140<br>*Surface Area=60<br>*shear stress= | Invalid, fill in data |

Table 6.9 Direct Shear Test of Soil Test

Liquid Limit Determination Using Casagrand Method

| Function | Inputs | Valid/Invalid |
|---|---|---|
| Calculate result | * Number of Hits=3<br>*Number of Sample (Trial)=1<br>*Number of container=1<br>*Weight of wet sample and container=50<br>*Weight of dry sample and container=36<br>*Weight of container=12 | Valid, well calculate result |
| Calculate result | * Number of Hits=3<br>*Number of Sample (Trial)=1<br>*Number of container=1<br>*Weight of wet sample and container=poi<br>*Weight of dry sample and container=36<br>*Weight of container=12 | Invalid, weight must be float |
| Calculate result | * Number of Hits=3<br>*Number of Sample (Trial)=1<br>*Number of container=1<br>*Weight of wet sample and container=<br>*Weight of dry sample and container=36<br>*Weight of container=12 | Invalid, fill in data |

Table 6.10 Liquid Limit Determination Using Casagrand Method Test

Field Density Test Using Sand Replacement Method

| Function | Inputs | Valid/Invalid |
|---|---|---|
| Calculate result | *Sample Number=1<br>*weight of wet soil taken from the hole=4<br>*weight of dry soil taken from the hole=3<br>*weight of empty cone=8<br>*weight of sand that fill the hole and cone=6<br>*weight of sand that fill the cone=10<br>*Sand density=10 | Valid, well calculate result |
| Calculate result | *Sample Number=1<br>*weight of wet soil taken from the hole=4<br>*weight of dry soil taken from the hole=3<br>*weight of empty cone=8<br>*weight of sand that fill the hole and cone=6<br>*weight of sand that fill the cone=10<br>*Sand density=10 | Invalid, weight must be float |
| Calculate result | * Number of Hits=3<br>*Number of Sample (Trial)=1<br>*Number of container=1<br>*Weight of wet sample and container=<br>*Weight of dry sample and container=36<br>*Weight of container=12 | Invalid, fill in data |

Table 6.11 Field Density Test Using Sand Replacement Method Test

Falling Head Soil Permeability Test

| Function | Inputs | Valid/Invalid |
| --- | --- | --- |
| Calculate result | *Sample Number=1<br>*Sample Length=4<br>*Time=5<br>*Height before =8<br>*Height 2=8<br>*Area of Container=45<br>*Area of glass cross section=30 | Valid, well calculate result |
| Calculate result | *Sample Number=1<br>*Sample Length=4<br>*Time=jhn<br>*Height before =8<br>*Height 2=8<br>*Area of Container=45<br>*Area of glass cross section=30 | Invalid, weight must be float |
| Calculate result | *Sample Number=1<br>*Sample Length=<br>*Time=5<br>*Height before =8<br>*Height 2=8<br>*Area of Container=45<br>*Area of glass cross section=30 | Invalid, fill in data |

Table 6.12 Falling Head Soil Permeability Test

Constant Head Soil Permeability Test

| Function | Inputs | Valid/Invalid |
|---|---|---|
| Calculate result | *Sample Number=2<br>*amount of filtered water from the sample=10<br>*Sample Length=12<br>*cross sectional area=15<br>*the time required for leak=60<br>*difference between water height in both tubes =4 | Valid, well calculate result |
| Calculate result | *Sample Number=2<br>*amount of filtered water from the sample=tr<br>*Sample Length=12<br>*cross sectional area=15<br>*the time required for leak=60<br>*difference between water height in both tubes =4 | Invalid, weight must be float |
| Calculate result | *Sample Number=2<br>*amount of filtered water from the sample=10<br>*Sample Length=<br>*cross sectional area=15<br>*the time required for leak=60<br>*difference between water height in both tubes =4 | Invalid, fill in data |

Table 6.13 Constant Head Soil Permeability Test

Laboratory Compaction Test

| Function | Inputs | Valid/Invalid |
|---|---|---|
| Calculate result | *Sample Number=1<br>*weight of cylinder container and base=10<br>*weight of cylinder container, base and the soil inside it=20<br>*volume of container=14<br>*moisture content=5 | Valid, well calculate result |
| Calculate result | *Sample Number=1<br>*weight of cylinder container and base=uiy<br>*weight of cylinder container, base and the soil inside it=20<br>*volume of container=14<br>*moisture content=5 | Invalid, weight must be float |
| Calculate result | *Sample Number=1<br>*weight of cylinder container and base=10<br>*weight of cylinder container, base and the soil inside it=20<br>*volume of container=14<br>*moisture content= | Invalid, fill in data |

Table 6.14 Laboratory Compaction Test

Unconfined Compression Test of Soil

| Function | Inputs | Valid/Invalid |
|---|---|---|
| Calculate result | *Sample Number=1<br>*Compression Power=50<br>*Initial cross sectional area=10<br>*Initial Height =5<br>*difference in height=14 | Valid, well calculate result |
| Calculate result | *Sample Number=1<br>*Compression Power=50<br>*Initial cross sectional area=saeg<br>*Initial Height =5<br>*difference in height=14 | Invalid, weight must be float |
| Calculate result | *Sample Number=1<br>*Compression Power=50<br>*Initial cross sectional area=<br>*Initial Height =5<br>*difference in height=14 | Invalid, fill in data |

Table 6.15 Unconfined Compression Test of Soil

## 6.4 Summery

In this chapter a testing plan was applied on the system, and it gives us a successful results.

Now the system is ready to be used by the user (Civil Engineering Department), and we will follow the unexpected problems that may appear during the use of the program.

# Chapter Seven

# Maintenance

## 7.1 Introduction

During the operation and working period, errors and omissions in the original requirements are discovered. Program and design errors emerge and the need for new functionality is identified, problem resolution (corrective), enhancements, and interface modifications. The system must therefore evolve to remain useful.

In this section we illustrate how to establish the production environment, migration and deployment plan and the maintenance plan.

## 7.2 Establishment of production environment

In this section we will determine how to establish the production environments which are: hardware requirement, setup database and the installation process.

### Hardware Requirements:

The Hardware components that had been used in the project are:

- Pentium4 PC of these features:
  - Speed 1800MHz, 256MB RAM, 40GB H.D.
- Printer.
- Input and output devices (keyboard, mouse, screen).

### Setup Database:

The database had been taken as follows:

- Open SQL Server Service manager
- Stop the Service
- Go to C:\programfiles\ Microsoft SQL Server\MSSQL\Data
- Then copy the database (GEO).

## Installation:

After setting up the database on the user environment, the user can run the software directly without need to install it, since the software is .EXE file which is compatible with the available windows operating systems( 98,millinuim, 2000,XP).

## 7.3 Migration and deployment plan

This section will include the way in which the system will be in the production environment. There are three ways for that:
Parallel, direct and pilot.

**Direct way:**

Since there is no previous software for the geotechnical lab, the software will be putted directly in the production environment.

## 7.4 Maintenance plan

The maintenance plan is concerned with the error handling and how the user can report these errors.

**Error handling:**

We had handled the predicted errors that may occur, caused by the invalid inputs or the semantic inputs and make our best to avoid these errors. In case of other errors, the user can contact us so that we will repair those errors.

## 7.5 Summery:

In this chapter a maintenance and setup plan on the system done and it have a acceptable results.

Now the system is ready to be setup by the user (Civil Engineering Department), and we will follow the unexpected problems that may appear during the setup of the program.

References:


- Software Engineering , Sixth Edition2001, ISBN 0 201 39815 X, Ian Sommerville
- Fundamentals of Database System , Third Edition2000 , ISBN 0 201 54263 3, Ramez AL_Masri and Shamkant Navathe
- فحوصات التربة للأغراض الإنشائية, الطبعة الأولى2003م , د. سامي أحمد حجاوي
- Internet

# Appendix A

# User Manual

## About Geotechnical Software:

Geotechnical lab is one of the main labs in the Civil Engineering Department at Palestine Polytechnic University (PPU). This lab contains 13 experiments; each experiment has a specific problem.

The Supervisor of this lab faces a lot of problems in solving these problems in that it takes a lot of time doing calculations, they are error prone, some of them are very difficult to solve and tedious, and don't give the required degree of accuracy.

The idea of this project was put on as a graduation project by Dr. Nabil Al-Julani many times during the last three years.

Thus, there is a need for computer based system to solve these problems. This system should have the capability of reading data measured practically, and follow a specific algorithm to solve the problem and gives the results. The system also should be able of giving the results as a report including plotting when required, and saving these results then printing them when needed.

Based on the previous information, in that there is a need for computer based system to solve these problems. This idea was raised after two meetings with Dr. Talahmeh and Dr. Julani were a number of ideas discussed and evaluated, and after extensive search through the Geotechnical experiment books we decided to adopt this idea and apply it on PPU.

## Experiments Included in the Software

The following table includes the experiments programmed in this software:

| Experiment Number | Experiment name |
| --- | --- |
| Exp no.1 | Water or Moisture Content |
| Exp no.2 | Specific gravity determination using Pycnometer |
| Exp no.3 | Bulk Density determination Using Core Cutter |
| Exp no.4 | Bulk Density determination Using Immersion in water |
| Exp no.5 | Bulk Density determination Using Direct Measurement |
| Exp no.6 | Tri Axial Compression and Shear Test |
| Exp no.7 | Direct Shear Test of Soil |
| Exp no.8 | Liquid Limit Determination Using Casagrand Method |
| Exp no.9 | Field Density Test Using Sand Replacement Method |
| Exp no.10 | Falling head Soil Permeability test |
| Exp no.11 | Constant Head Soil Permeability Test |
| Exp no.12 | Laboratory Compaction Test |
| Exp no.13 | Unconfined Compression Test Of Soil |

## How to use:

The user of the program should have knowledge of the use of the Geotechnical lab in order to use it in an efficient way.

The Main steps that the user should follow in using the Software:

1. He should have a valid user name and password so that he can login and use it
2. After you login you have 5 main choices as administrator and three as an ordinal user, the choices for the user are:
    a. Retrieve for administrator only
    b. Experiments
    c. Create user for administrator only
    d. Print
    e. Save
    f. Help
    g. Exit

**Retrieve Choice:**

Allows the administrator to see the old exp. Saved in the database and all of that experiments information:

The user firstly should select the experiment he wants to see, after that the program will display all that experiment information in a new form such as:

- User Name
- Experimenter name
- Date of performing the Experiment
- Notes of the Experimenter about the experiment
- Experiment Parameters

After that the user has other choices to do:

- Print these Values
- Perform it Again by Selecting New
- Help on that Experiment
- To go Back to main menu or to the Experiments Menu

.Retrieve (search) results: the Data types of the experiment the user looks for must be the same as they are in the Database, this should provide the user with the required information about that experiment.

**Experiments:**

Allows the administrator or user to select one experiment to perform among 13 different experiments.

Inside the experiments the user can fill in data and then calculate results, after that he has the following choices:

- Save
- Retrieve
- Perform the experiment again
- Go back to main menu or experiments menu
- Exit from the program

**Create user:**

Allows the administrator only to create a new

**Print:**

Allows the administrator only to create a new user or administrator
the users are constrained to the UserID and password given by the administrator, so that no user can create other users or change his password only through the administrator
Print: it requires that an experiment is previously done and is currently viewed on the screen; this function must be disabled while the results are not ready

**Save:**

Save an experiment must be previously done and results are viewed, also the user should give valid parameters for experiment name, location of the experiment

**Help:**

Given briefly explanation to the administrator or user for each experiment.

Help frames must be provided for all experiments, and must contain clear steps on how to implement each one; also it give the user all the input parameters and data type for each one.

**Exit from the program:**
Allows the user to exit from the program (Ending the Program)

**How to contact us:**

The User of the program can contact us directly through the Computer Engineering Department or by the following E-mail:

Raef86@yahoo.com

Eng_BKawasmeh@yahoo.com

Naser_17_12@yahoo.com

# Appendix B

## Source Code

For Bulk Density determination Using Immersion in water

```
Public conn As New ADODB.Connection
Public RS As New ADODB.Recordset
Public com As New ADODB.Command
Private Sub Command1_Click()

Dim v As Double

Dim m1 As Double
Dim m2 As Double
Dim m As Double
Dim p As Double
Dim pp As Double

On Error GoTo Err_handling



If Text1.Text = "" Then
MsgBox ("please fill in the data"), (vbOKOnly), ("Geotechnical Software")
Exit Sub
End If

If Text2.Text = "" Then
MsgBox ("please fill in the data"), (vbOKOnly), ("Geotechnical Software")
Exit Sub
End If

If Text3.Text = "" Then
MsgBox ("please fill in the data"), (vbOKOnly), ("Geotechnical Software")
Exit Sub
End If

If Text4.Text = "" Then
MsgBox ("please fill in the data"), (vbOKOnly), ("Geotechnical Software")
```

```
Exit Sub
End If

m = Text1.Text
m1 = Text2.Text
m2 = Text3.Text
pp = Text4.Text

v = (m1 - m2) - (m1 - m) / pp
p = m / v
Label7(1).Caption = p
Command3.Enabled = True
Command4.Enabled = True

Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Text6.Enabled = False
Text7.Enabled = False
Text8.Enabled = False
Exit Sub
Err_handling:
   MsgBox "Error " & Err.Number & " | " & Err.Description, vbOKOnly,
"Geotechnical Software"


End Sub

Private Sub Command2_Click()
BulkDensitydeterminationusingImmersioninwater.Hide
help4.Show
End Sub
```

```
Private Sub Command3_Click()
conn.ConnectionString = "Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist
Security Info=False;Initial Catalog=Geo;Data Source=RAEFCOMP"
conn.CursorLocation = adUseServer
conn.Open

com.CommandText = "select [Sequential Number] from [Bulk Density determination
using Immersion in water] "
com.ActiveConnection = conn
com.CommandType = adCmdText
Set RS = com.Execute
While Not RS.EOF
seq = RS("sequential number")
RS.MoveNext
Wend

seq = seq + 1
Trim ("seq")
inscom = "insert into [Bulk Density determination using Immersion in water] values
("
inscom = inscom & 4
inscom = inscom & "," & seq
inscom = inscom & "," & Main.idnumber
inscom = inscom & "," & "'" & Text6.Text
inscom = inscom & "'" & "," & "'" & Text7.Text
inscom = inscom & "'" & "," & Text1.Text
inscom = inscom & "," & Text2.Text
inscom = inscom & "," & Text3.Text
inscom = inscom & "," & Text4.Text
inscom = inscom & "," & Label7(1).Caption
inscom = inscom & ")"

com.CommandText = inscom
com.ActiveConnection = conn
```

```
com.CommandType = adCmdText
Set RS = com.Execute
conn.Close
MsgBox ("Data Successfully Added to the DataBase"), vbOKCancel, ("Geotechnical
Software")
Adodc1.Refresh
DataList1.Refresh
Command3.Enabled = False
End Sub

Private Sub Command4_Click()
On Error GoTo Err_handling

Err_handling:
   MsgBox "Error " & Err.Number & " | " & Err.Description, vbOKOnly,
"Geotechnical Software"



End Sub

Private Sub Command5_Click()
BulkDensitydeterminationusingImmersioninwater.Hide
Main.Show

End Sub

Private Sub Command6_Click()
Dim answer As Integer
Select Case MsgBox("Are You Sure You Want To Exit", vbOKCancel, "EXIT")
Case 1
End
Case 2
Case Else
End Select
```

```
End Sub

Private Sub Command7_Click()
BulkDensitydeterminationusingImmersioninwater.Hide
Experiments.Show
End Sub

Private Sub Command8_Click()
Text1.Enabled = True
Text2.Enabled = True
Text3.Enabled = True
Text4.Enabled = True
Text6.Enabled = True
Text7.Enabled = True
Text8.Enabled = True
Text1.Text = ""
Text2.Text = ""
Text4.Text = ""
Text3.Text = ""
Text7.Text = ""
Text6.Text = Date
Label7(1).Caption = "Result"
Label12.Visible = False
Text8.Visible = False
Command4.Enabled = False
Command3.Enabled = False
Command1.Enabled = True
Label6(0).Visible = False
DataList1.Visible = False
Text1.SetFocus
End Sub

Private Sub DataList1_Click()
Label12.Visible = True
```

```
Text8.Visible = True

x = DataList1.SelectedItem
id = DataList1.BoundText

conn.ConnectionString = "Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist
Security Info=False;Initial Catalog=Geo;Data Source=RAEFCOMP"
conn.CursorLocation = adUseServer
conn.Open

Namecom = "select * from [Bulk Density determination using Immersion in water]
where [Sequential Number] = "
Namecom = Namecom & Trim(id)
Namecom = Namecom
com.CommandText = Namecom
com.ActiveConnection = conn
com.CommandType = adCmdText
Set RS = com.Execute

Text1.Text = RS("sample weight")
Text2.Text = RS("The sample weight covered with paraffin")
Text4.Text = RS("Density of paraffin.")
Text3.Text = RS("Sample weight tipped in water")
Text7.Text = RS("Notes")
Text6.Text = RS("date")
Label7(1).Caption = RS("Bulk Density")
useriddid = Trim(RS("User ID"))
conn.Close

conn.ConnectionString = "Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist
Security Info=False;Initial Catalog=Geo;Data Source=RAEFCOMP"
conn.CursorLocation = adUseServer
conn.Open
```

```
Namecom = "select username from users where userid = (select userid from [Bulk
Density determination using Immersion in water] where [Bulk Density determination
using Immersion in water].[Sequential Number]= "
Namecom = Namecom & x
Namecom = Namecom & ")"
com.CommandText = Namecom
com.ActiveConnection = conn
com.CommandType = adCmdText
Set RS = com.Execute
Text8.Text = RS("username")
conn.Close

Command4.Enabled = False
Command3.Enabled = False
Command1.Enabled = False

End Sub

Private Sub Form_GotFocus()
WindowState = 2
Text5.Text = Main.uin
Text1.SetFocus
End Sub

Private Sub form_load()
WindowState = 2
Text5.Text = Main.uin
Text6.Text = Date

End Sub

Private Sub Text1_LostFocus()
If Text1.Text = "" Then
Exit Sub
```

```
End If
If IsNumeric(Text1.Text) Then
Else
Text1.Text = ""
MsgBox ("ERROR DATA, Data must be float"), (vbOKOnly), ("Geotechnical
Software")
Exit Sub
End If
End Sub

Private Sub Text2_LostFocus()
If Text2.Text = "" Then
Exit Sub
End If
If IsNumeric(Text2.Text) Then
Else
Text2.Text = ""
MsgBox ("ERROR DATA, Data must be float"), (vbOKOnly), ("Geotechnical
Software")
Exit Sub
End If
End Sub

Private Sub Text3_LostFocus()
If Text3.Text = "" Then
Exit Sub
End If
If IsNumeric(Text3.Text) Then
Else
Text3.Text = ""
MsgBox ("ERROR DATA, Data must be float"), (vbOKOnly), ("Geotechnical
Software")
Exit Sub
End If
```

```
End Sub

Private Sub Text4_LostFocus()
If Text4.Text = "" Then
Exit Sub
End If
If IsNumeric(Text4.Text) Then
Else
Text4.Text = ""
MsgBox ("ERROR DATA, Data must be float"), (vbOKOnly), ("Geotechnical
Software")
Exit Sub
End If
End Sub
```