VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590014

A DBMS Mini-Project Report
On

“DEFENCE-X”

Submitted in partial fulfilment of the requirements for the 5th semester of ***Bachelor of Engineering in Computer Science and Engineering*** *of*
Visvesvaraya Technological University, Belagavi

Submitted by:

ANKIT DWIVEDI	**1RN17CS014**
ARK PANDEY	**1RN17CS018**

Under the Guidance of:

Mr Bhavani Shankar K	**Mr. H.R. Shasidhar**	**Mrs S Mamatha Jajur**
Assistant Professor	**Assistant Professor**	**Assistant Professor**
Dept. of CSE	**Dept. of CSE**	**Dept. of CSE**

Department of Computer Science and Engineering
RNS Institute of Technology
Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560 098

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road,
Bengaluru-560 098

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that the DBMS mini-project work entitled **“Defence-X”** has been successfully carried out by **Ankit Dwivedi** bearing USN **1RN17CS014** , and **Ark Pandey** bearing USN **1RN17 CS018**, bonafide students of **RNS Institute of Technology** in partial fulfillment of the requirements for the **5th semester** **Bachelor of Engineering** in **Computer Science and Engineering** of **Visvesvaraya Technological University**, Belagavi, during the academic year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the mini-project requirements of DBMS lab of 5th semester BE in CSE.

Mr. Bhavani Shankar K	**Mr. H.R. Shasidhar**	**Mrs S Mamatha Jajur**
Assistant Professor	**Assistant Professor**	**Assistant Professor**
Dept. of CSE	**Dept. of CSE**	**Dept of CSE**

<u>External Viva:</u>

Name of the Examiners	**Signature with Date**
1.	
2.	

ACKNOWLEDGMENTS

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

We would like to thank **Dr. H N Shivashankar**, Director, RNSIT, Bangalore, for his moral support towards completing our project.

We are grateful to **Dr. M K Venkatesha,** Principal, RNSIT, Bangalore, for his support towards completing this mini project.

We would like to thank **Dr. G T Raju**, Dean of Engg., Prof. & Head , Department of Computer Science & Engineering, RNSIT, Bangalore, for his valuable suggestions and expert advice.

We deeply express my sincere gratitude to my guide **Mr Bhavani Shankar K** and **Mrs S Mamatha Jajur,** Asst Prof, Department of CSE, RNSIT, Bangalore, for their able guidance, regular source of encouragement and assistance throughout this project.

We would like to thank all the teaching and non-teaching staff of department of Computer Science & Engineering, RNSIT, Bengaluru for their constant support and encouragement.

Date	:	ANKIT DWIVEDI	1RN17CS014
Place	: Bengaluru	ARK PANDEY	1RN17CS018

ABSTRACT

It is necessary for various government and defence forces agencies to keep track and maintain an efficient database for the soldiers of the Military,Navy and Air Force of India.The challenges in the manual system are to keep an accurate track of all the soldiers which is difficult to maintain and also very much time consuming.Also,data may not be perfect.

This is a software which can be used by the various government agencies as well as the Indian Defence Forces agencies in order to maintain and keep track of all the soldiers which are posted at different locations. It helps to keep a record of all the soldier details and login credentials are created for security.This software also allows to search for data such as soldier postings ,soldier weapons ,location and various other details.

In comparison to the existing systems, the proposed system will be better in maintaining soldier records as well as their posting locations and will also give the end user a simple and efficient platform to track and record the defence forces of India.

This MiniProject has been implemented using Netbeans as FrontEnd and Mysql as Backend.

CONTENT

CHAPTER 1

INTRODUCTION

1.1 DATABASE TECHNOLOGIES

The essential feature of database technology is that it provides an internal representation (model) of the external world of interest. Examples are the representation of a particular date/time/flight/aircraft in airline reservation or of item code/item description/quantity on hand/reorder level/reorder quantity in a stock control system.The technology involved is concerned primarily with maintaining the internal representation consistent with external reality; this involves the results of extensive R&D over the past 30 years in areas such as user requirements analysis, data modelling, process modelling, data integrity, concurrency, transactions, file organisation, indexing, rollback and recovery, persistent programming, object-orientation, logic programming, deductive database systems, active database systems... and in all these (and other) areas there remains much to be done.

The essential point is that database technology is a CORE TECHNOLOGY with links to:

- Information management / processing
- Data analysis / statistics
- Data visualization / presentation
- Multimedia and hypermedia
- Office and document systems
- Business processes, workflow, CSCW (computer-supported cooperative work)

Relational DBMS is the modern base technology for many business applications. It offers flexibility and easy-to-use tools at the expense of ultimate performance. More recently relational systems have started to extend their facilities in the directions of information retrieval, object-orientation and deductive/active systems leading to the so-called 'Extended Relational Systems'.

Information Retrieval Systems started with handling library catalogues and extended to full free-text utilizing inverted index technology with a lexicon or thesaurus. Modern systems utilize some KBS (knowledge-based systems) techniques to improve retrieval.

Object-Oriented DBMS started for engineering applications where objects are complex, have versions and need to be treated as a complete entity. OODBMSs share many of the OOPL features such as identity, inheritance, late binding, overloading and overriding. OODBMSs have found favour in engineering and office systems but have not yet been successful in traditional application areas.

Deductive / Active DBMS have emerged over the last 20 years and combine logic programming technology with database technology. This allows the database itself to react to external events an to maintain dynamically its integrity with respect to the real world.

1.2 CHARACTERISTICS OF DATABASE APPROACH

Traditionally, data was organized in file formats. DBMS was a new concept then, and all the research was done to make it overcome the deficiencies in traditional style of data management. A modern DBMS has the following characteristics −

- Real-world entity − A modern DBMS is more realistic and uses realworld entities to design its architecture. It uses the behavior and attributes too. For example, a school database may use students as an entity and their age as an attribute.
- Relation-based tables − DBMS allows entities and relations among them to form tables. A user can understand the architecture of a database just by looking at the table names.
- Isolation of data and application − A database system is entirely different than its data. A database is an active entity, whereas data is said

to be passive, on which the database works and organizes. DBMS also stores metadata, which is data about data, to ease its own process.

- Less redundancy − DBMS follows the rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Normalization is a mathematically rich and scientific process that reduces data redundancy.
- Consistency − Consistency is a state where every relation in a database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state. A DBMS can provide greater consistency as compared to earlier forms of data storing applications like file-processing systems.
- Query Language − DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and as different filtering options as required to retrieve a set of data. Traditionally it was not possible where file-processing system was used.
- ACID Properties − DBMS follows the concepts of Atomicity, Consistency, Isolation, and Durability (normally shortened as ACID). These concepts are applied on transactions, which manipulate data in a database. ACID properties help the database stay healthy in multitransactional environments and in case of failure.
- Multiuser and Concurrent Access − DBMS supports multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when users attempt to handle the same data item, but users are always unaware of them. ○ Multiple views − DBMS offers multiple views for different users. A user who is in the Sales department will have a different view of database than a person working in the Production department. This feature enables the users to have a concentrate view of the database according to their requirements. ○ Security − Features like multiple views offer security to some extent where users are unable to access data of other users and departments.

DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage. DBMS offers many different levels of security features, which enables multiple users to have different views with different features. For example, a user in the Sales department cannot see the data that belongs to the Purchase department. Additionally, it can also be managed how much data of the Sales department should be displayed to the user. Since a DBMS is not saved on the disk as traditional file systems, it is very hard for miscreants to break the code.

1.3 APPLICATIONS OF DBMS

Applications where we use Database Management Systems are:

- **Telecom**: There is a database to keeps track of the information regarding calls made, network usage, customer details etc. Without the database systems it is hard to maintain that huge amount of data that keeps updating every millisecond.
- **Industry**: Where it is a manufacturing unit, warehouse or distribution centre, each one needs a database to keep the records of ins and outs. For example distribution centre should keep a track of the product units that supplied into the centre as well as the products that got delivered out from the distribution centre on each day; this is where DBMS comes into picture.
- **Banking System**: For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of Database management systems.
- **Education sector**: Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is a hell lot amount of inter-related data that needs to be stored and retrieved in an efficient manner.

- **Online shopping**: You must be aware of the online shopping websites such as Amazon, Flip kart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

1.4 PROBLEM DESCRIPTION/STATEMENT

Various government agencies and the Indian defence forces department are in a need of a system to manage the details of all the soldiers.In a single software they will need the ability to handle all the postings of soldiers at different locations and keep track of all soldier details including weapons,etc.

This is a software which can be used by Indian defence forces department mainly the Military,Navy and Air Force department for maintaining the soldier records and keeping track of them all and a password and login is created for security.

CHAPTER 2

REQUIREMENT ANALYSIS

2.1 HARDWARE REQUIREMENTS

The Hardware requirements are very minimal and the program can be run on most of the machines.

Processor	:	Pentium4 processor
Processor Speed	:	2.4 GHz
RAM	:	1 GB
Storage Space	:	40 GB
Monitor Resolution	:	1024*768 1280*1024

2.2 SOFTWARE REQUIREMENTS

Operating System	:	Windows 10
IDE	:	MYSQL Workbench, NetBeans

2.3 FUNCTIONAL REQUIREMENTS

2.3.1 Major Entities

The system must have a password protected access system such that only people with authenticated credential are allowed to access the function of the system.

The system must include functions that will allow the user to create soldier details, posting location,weapons etc. and them to the database.

The System must include functions to delete records in the database.The System must provide functions for displaying the current defence database

2.3.2 End User Requirements

The main users of the product would be management staff in the Indian Defence forces agencies.The Staff should have login credential so that only selected staff can access the software.The Staff incharge has the responsibility of creating Soldier forms,adding soldier details and updating the soldier details.

2.3.4 Java

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use,particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer lowlevel facilities than either of them.

One design goal of Java is portability, which means that programs written for the Java platform must run similarly on any combination of hardware and operating system with adequate runtime support. This is achieved by compiling the Java language code to an intermediate representation called Java bytecode, instead of directly to architecture-specific machine code. Java bytecode instructions are analogous to machine code, but they are intended to be executed by a virtual machine (VM) written specifically for the host hardware. End users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a web browser for Java applets.

2.3.5 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by MySQL AB. MySQL AB is a commercial company, founded by the MySQL developers. It is a second generation Open Source company that unites Open Source values and methodology with a successful business model.

- MySQL is a database management system.

  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- MySQL is a relational database management system.

  A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- MySQL software is Open Source.

  Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. The MySQL Database Server is very fast, reliable, and easy to use.

- MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

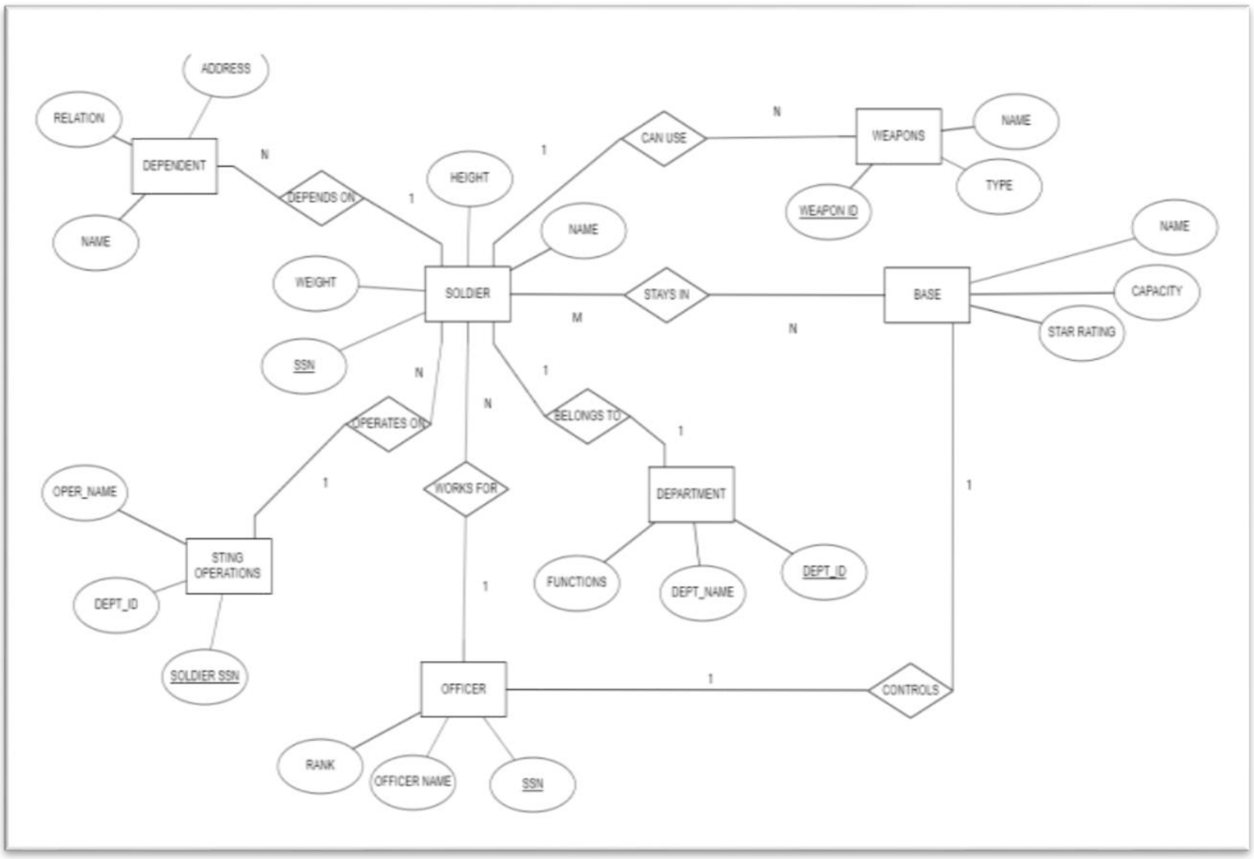
CHAPTER 3

DATABASE DESIGN

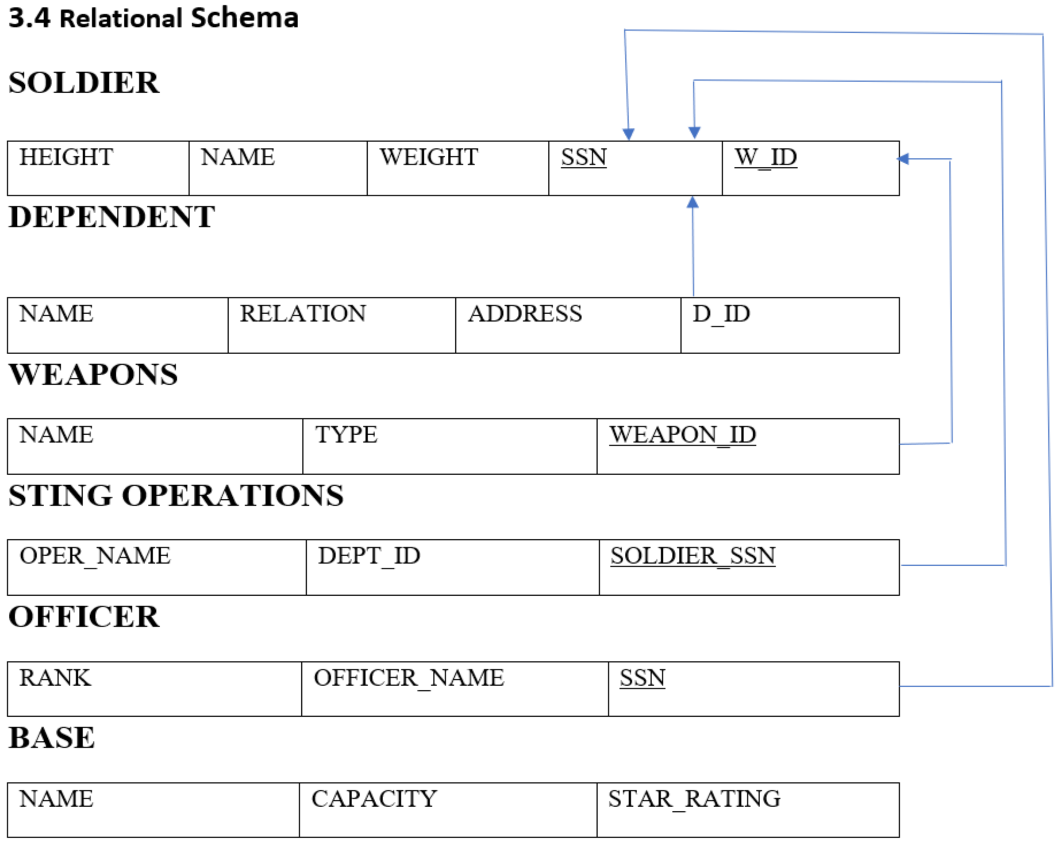
3.1 Entities, Attributes and Relationships

- **SOLDIER:** Height, Name, Weight, <u>SSN, </u>W_ID
- **DEPENDENT:** Name, Relation, Address
- **WEAPONS:** Name, Type, <u>Weapon_ID</u>
- **STING OPERATIONS:** Oper_Name, Dept_ID, Soldier_SSN
- **OFFICER:** Rank, Officer_Name, <u>SSN</u>
- **BASE:** Name, Capacity, Star_Ratings

3.3 ER Schema


ADDRESS
RELATION
DEPENDENT
NAME
N
DEPENDS ON
1
HEIGHT
1
CAN USE
N
WEAPONS
NAME
TYPE
WEAPON ID
NAME
SOLDIER
WEIGHT
SSN
STAYS IN
M
N
BASE
NAME
CAPACITY
STAR RATING
N
1
OPERATES ON
N
BELONGS TO
1
OPER_NAME
1
WORKS FOR
DEPARTMENT
1
STING OPERATIONS
DEPT_ID
SOLDIER SSN
FUNCTIONS
DEPT_NAME
DEPT_ID
1
OFFICER
1
CONTROLS
RANK
OFFICER NAME
SSN

3.4 Relational Schema



SOLDIER

HEIGHT	NAME	WEIGHT	SSN	W_ID

DEPENDENT

NAME	RELATION	ADDRESS	D_ID

WEAPONS

NAME	TYPE	WEAPON_ID

STING OPERATIONS

OPER_NAME	DEPT_ID	SOLDIER_SSN

OFFICER

RANK	OFFICER_NAME	SSN

BASE

NAME	CAPACITY	STAR_RATING

CHAPTER 4

IMPLEMENTATION

4.1 DATABASE CONNECTIVITY

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is Java based data access technology and used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database, and is oriented towards relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the Java virtual machine (JVM) host environment.

FUNCTIONALITY

JDBC allows multiple implementations to exist and be used by the same application. The API provides a mechanism for dynamically loading the correct Java packages and registering them with the JDBC Driver Manager. The Driver Manager is used as a connection factory for creating JDBC connections.

JDBC connections support creating and executing statements. These may be update statements such as SQL's CREATE, INSERT, UPDATE and DELETE, or they may be query statements such as SELECT. Additionally, stored procedures may be invoked through a JDBC connection. JDBC represents statements using one of the following classes:

- Statement – the statement is sent to the database server each and every time.
- PreparedStatement – the statement is cached and then the execution path is predetermined on the database server allowing it to be executed multiple times in an efficient manner.
- CallableStatement – used for executing stored procedures on the database.

Update statements such as INSERT, UPDATE and DELETE return an update count that indicates how many rows were affected in the database. These statements do not return any other information.

Query statements return a JDBC row result set. The row result set is used to walk over the result set. Individual columns in a row are retrieved either by name or by column number. There may be any number of rows in the result set. The row result set has metadata that describes the names of the columns and their types.

There is an extension to the basic JDBC API in the javax.sql.

JDBC connections are often managed via a connection pool rather than obtained directly from the driver.

JDBC DRIVERS

JDBC drivers are client-side adapters (installed on the client machine, not on the server) that convert requests from Java programs to a protocol that the DBMS can understand. Types Commercial and free drivers provide connectivity to most relational-database servers. These drivers fall into one of the following types:

a. Type 1 that calls native code of the locally available ODBC driver.

b. Type 2 that calls database vendor native library on a client side. This code then talks to database over the network.

c. Type 3, the pure-java driver that talks with the server-side middleware that then talks to the database.

d. Type 4, the pure-java driver that uses database native protocol.

Note also a type called an internal JDBC driver - a driver embedded with JRE in Java-enabled SQL databases. It is used for Java stored procedures.

Steps to connect to the database in java

There are 5 steps to connect any java application with the database in java using JDBC.

They are as follows:

1. Register the driver class
2. Creating connection
3. Creating statement
4. Executing queries
5. Closing connection

Register the driver class

The forName() method of Class class is used to register the driver class. This method is used to dynamically load the driver class.

Syntax of forName() method :

public static void forName(String className)throws ClassNotFoundException .

Create the connection object

The getConnection() method of DriverManager class is used to establish connection with the database.

Syntax of getConnection() method:

- public static Connection getConnection(String url)throws SQLException
- public static Connection getConnection(String url,String name,String password) throws SQLException

Create the Statement object

The createStatement() method of Connection interface is used to create statement. The object of statement is responsible to execute queries with the database.

Syntax of createStatement() method:

public Statement createStatement()throws SQLException

Execute the query

The executeQuery() method of Statement interface is used to execute queries to the database. This method returns the object of ResultSet that can be used to get all the records of a table. Syntax of executeQuery() method:

public ResultSet executeQuery(String sql)throws SQLException

Close the connection object

By closing connection object statement and ResultSet will be closed automatically. The close() method of Connection interface is used to close the connection.

Syntax of close() method:

public void close()throws SQLException

4.2 Pseudo Code For Major Functionalities

Pseudocode:

```
////CLASS CONNECTION////

public static Connection ConnecrDb(){

        try{

            Class.forName("org.sqlite.JDBC");

            Connection
conn=DriverManager.getConnection("jdbc:sqlite:C:\\Users\\Ark
Pandey\\Desktop\\defencexdb\\Defence-x.sqlite.sqlite");

            return conn;

        }

        catch(Exception e)

        {

            JOptionPane.showMessageDialog(null, e);

            return null;

        }

    }
```

```
////CLASS SIGNUP////

            String sql="insert into Account (Username,Name,Password,Sec_Q,Answer)
            values (?,?,?,?,?)";
            pst=conn.prepareStatement(sql);
            pst.setString(1,jTextField1.getText());
            pst.setString(2,jTextField2.getText());
            pst.setString(3,jTextField3.getText());

            pst.setString(4,(String) jComboBox1.getSelectedItem());
            pst.setString(5,jTextField4.getText());
            pst.execute();
            JOptionPane.showMessageDialog(null, "New Account Created");
```

////CLASS NEW SOLDIER ENTRY////

```
String sql="insert into Soldier(Soldier_ssn,name,department,rank,age,salary)
values (?,?,?,?,?,?)";

pst=conn.prepareStatement(sql);
pst.setString(1,jTextField1.getText());
pst.setString(2,jTextField2.getText());
pst.setString(3,(String) jComboBox1.getSelectedItem());
pst.setString(4,jTextField3.getText());
pst.setString(5,jTextField4.getText());
pst.setString(6,jTextField5.getText());
pst.execute();
 JOptionPane.showMessageDialog(null, "New Soldier Recruited");
```

////CLASS NEW CAMP ENTRY////

```
String sql="insert intoCamp(camp_id,name,location,camptype,capacity,vicinity)
values (?,?,?,?,?,?)";
pst=conn.prepareStatement(sql);
pst.setString(1,jTextField1.getText());
pst.setString(2,jTextField2.getText());
pst.setString(3,jTextField3.getText());
pst.setString(4,(String) jComboBox1.getSelectedItem());
pst.setString(5,(String) jComboBox2.getSelectedItem());
pst.setString(6,(String) jComboBox3.getSelectedItem());

pst.execute();
 JOptionPane.showMessageDialog(null, "New Camp Assigned");
```

////CLASS STORED PROCEDURE////

```
 String sql3="SELECT MAX(AGE) FROM SOLDIER";
 pst=conn.prepareStatement(sql3);
  rs=pst.executeQuery();
  if(rs.next()){

String avg=rs.getString("MAX(AGE)");
jTextField3.setText(avg);
```

////CLASS ASSIGN SOLDIER////

```
String sql="select * from Soldier where Soldier_SSN=?";

pst=conn.prepareStatement(sql);
pst.setString(1,jTextField1.getText());
```

```
            rs=pst.executeQuery();
            if(rs.next())
            {
                String add1=rs.getString("Name");
                jTextField2.setText(add1);
                String add2=rs.getString("Department");
                jTextField3.setText(add2);
                String add3=rs.getString("Rank");
                jTextField4.setText(add3);
                String add4=rs.getString("Age");
                jTextField5.setText(add4);
                String add5=rs.getString("Salary");
                jTextField6.setText(add5);
                rs.close();
                pst.close();
            }

////CLASS RETURNED////

            String sql="select * from issue where Camp_ID=?";
            pst=conn.prepareStatement(sql);
            pst.setString(1,jTextField7.getText());

            rs=pst.executeQuery();
            if(rs.next())
            {
                String add1=rs.getString("Cname");
                jTextField8.setText(add1);
                 String add2=rs.getString("Location");
                jTextField9.setText(add2);
                 String add3=rs.getString("Camptype");
                jTextField10.setText(add3);
                 String add4=rs.getString("Capacity");
                jTextField11.setText(add4);
                 String add5=rs.getString("vicinity");
                jTextField12.setText(add5);
                 String add6=rs.getString("Soldier_SSN");
                jTextField1.setText(add6);
                 String add7=rs.getString("Name");
                jTextField2.setText(add7);
                String add8=rs.getString("Department");
                jTextField3.setText(add8);
                 String add9=rs.getString("Rank");
                jTextField4.setText(add9);
                 String add10=rs.getString("Age");
                jTextField5.setText(add10);
                 String add11=rs.getString("Salary");
                 jTextField6.setText(add11);
                  String add12=rs.getString("dateofmission");
                 jTextField13.setText(add12);


                rs.close();
                pst.close();
            }

////CLASS STATISTICS////

            public void jTable2()
    {
        try{
            String sql="select Soldier_SSn,Name,Department,Rank,Age,Salary from Issue";
            pst=conn.prepareStatement(sql);
            rs=pst.executeQuery();
            jTable2.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null, e);
        }
    }
     public void jTable3()
     {
          try{
            String sql="select Camp_ID,Name,Location,CampType,Capacity,Vicinity from
Return";
            pst=conn.prepareStatement(sql);
            rs=pst.executeQuery();
            jTable3.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null, e);
        }
     }
```

CHAPTER 5
RESULTS, SNAPSHOTS AND DISCUSSIONS

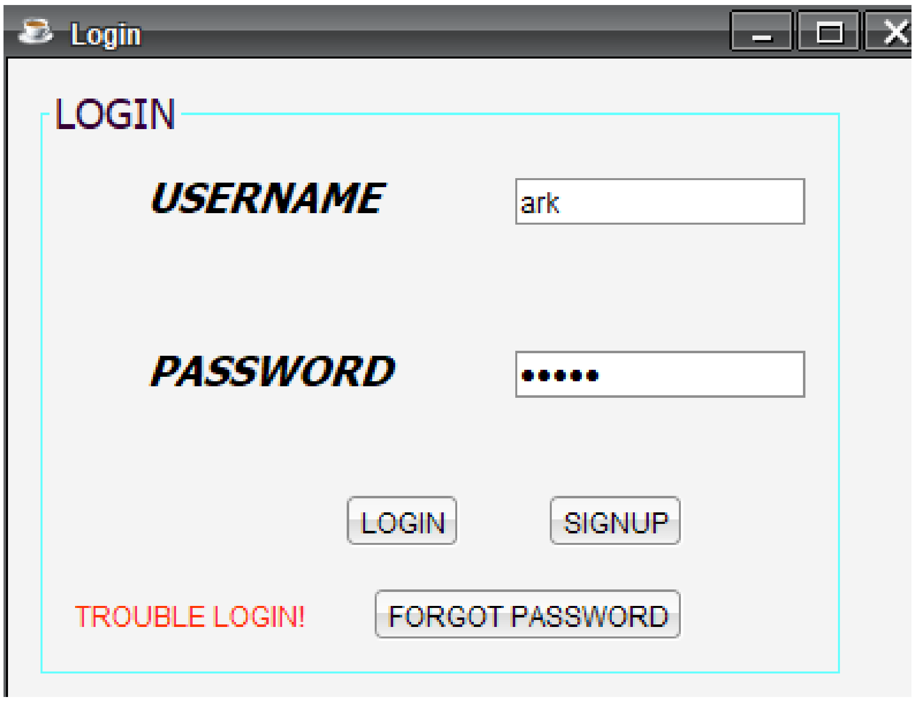


Fig 1: Login Page

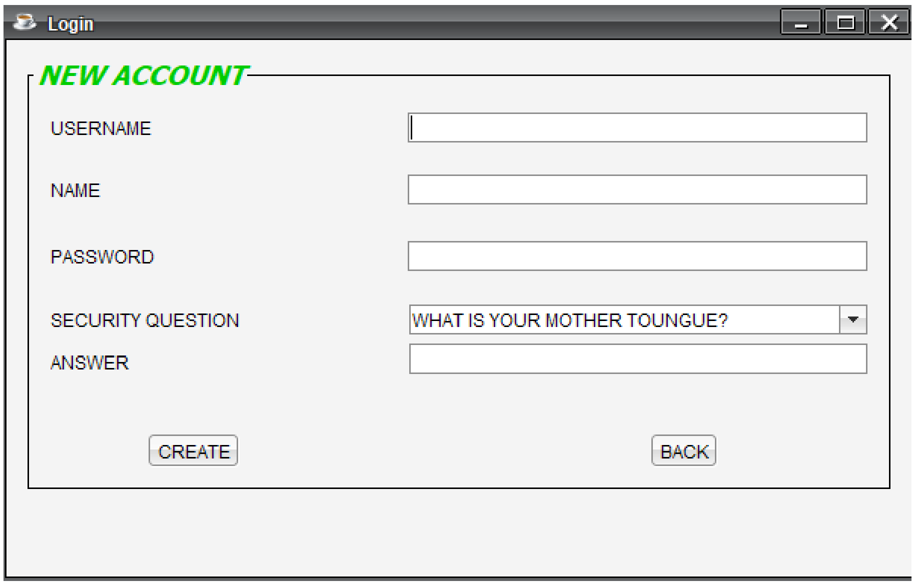
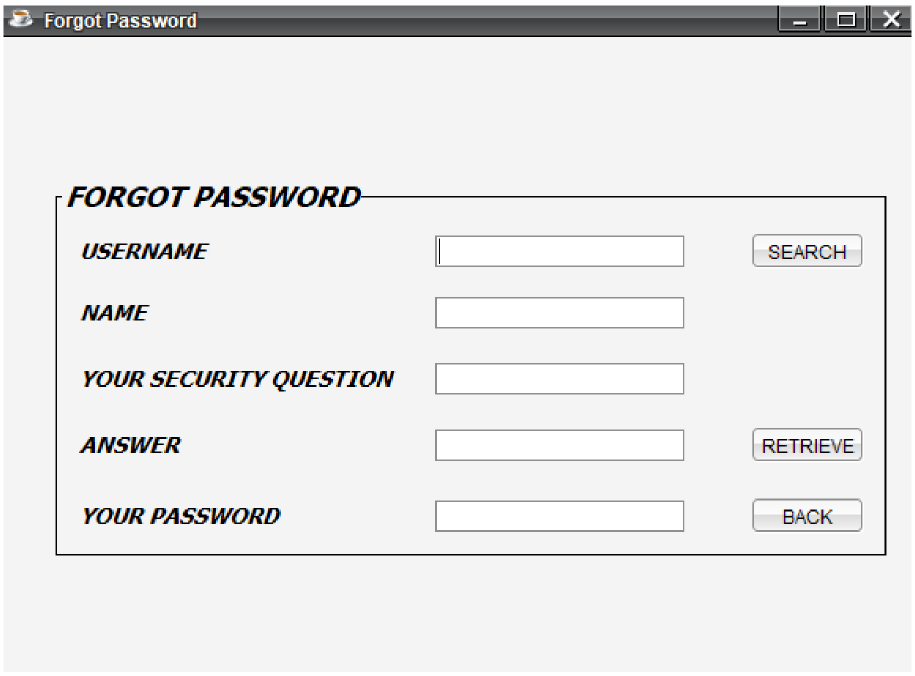


Fig 2: Signup Page



Forgot Password

FORGOT PASSWORD

USERNAME		SEARCH
NAME		
YOUR SECURITY QUESTION		
ANSWER		RETRIEVE
YOUR PASSWORD		BACK

Fig 3: Password Regeneration Page

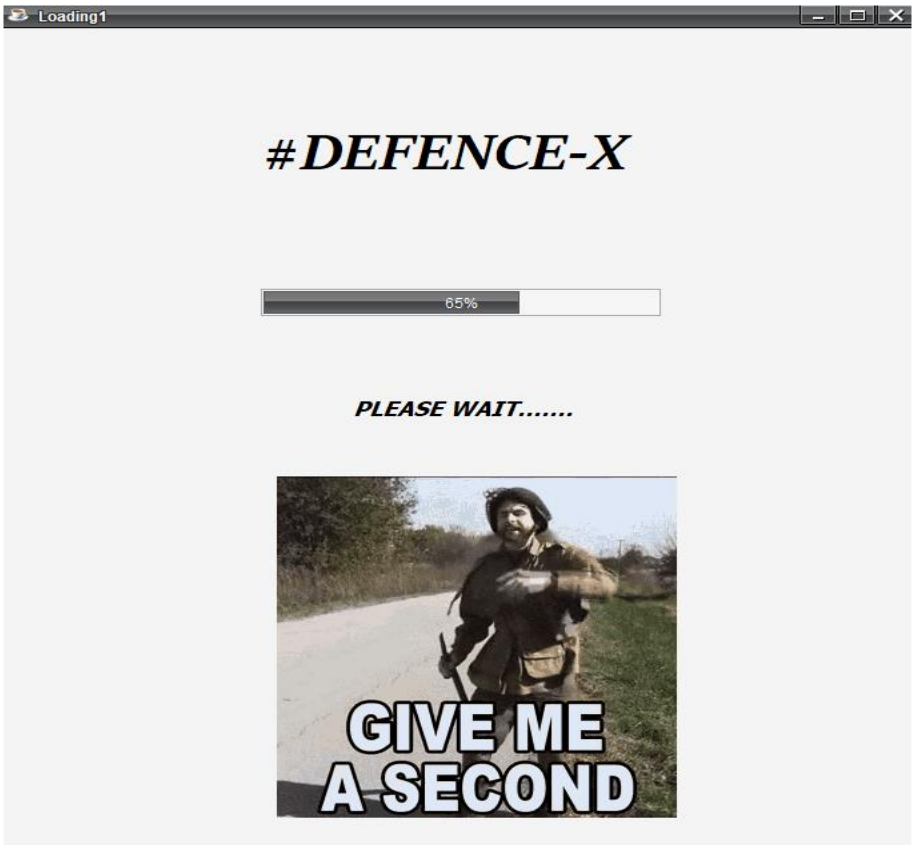


Fig 4: A Fun Loading Page

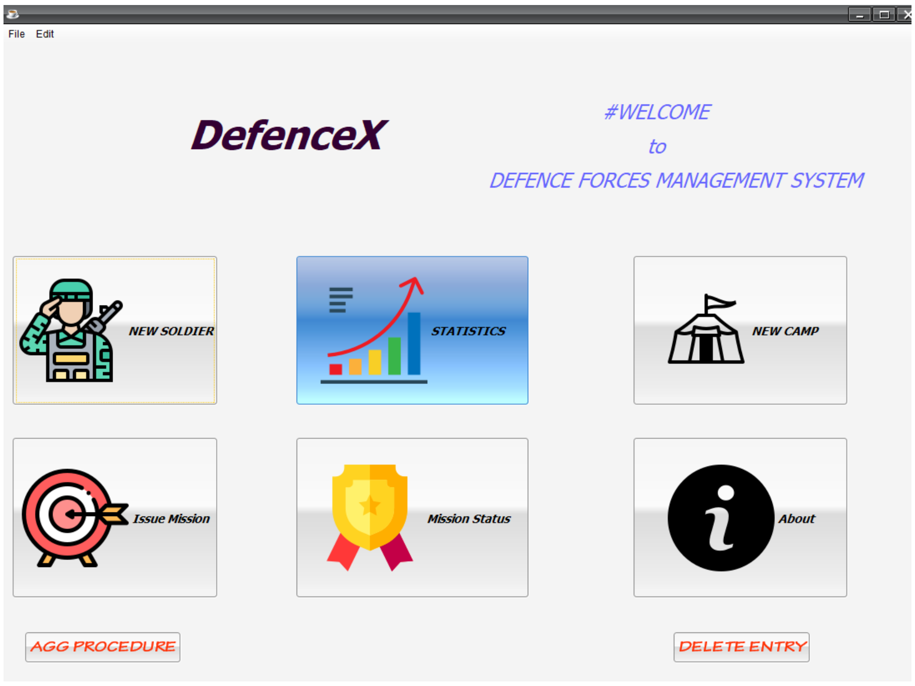
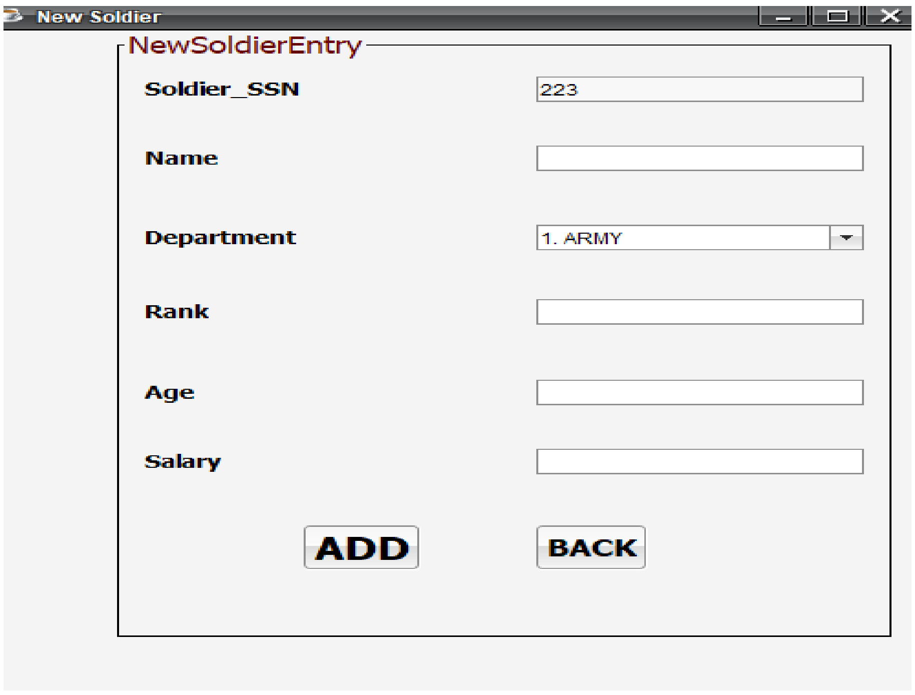


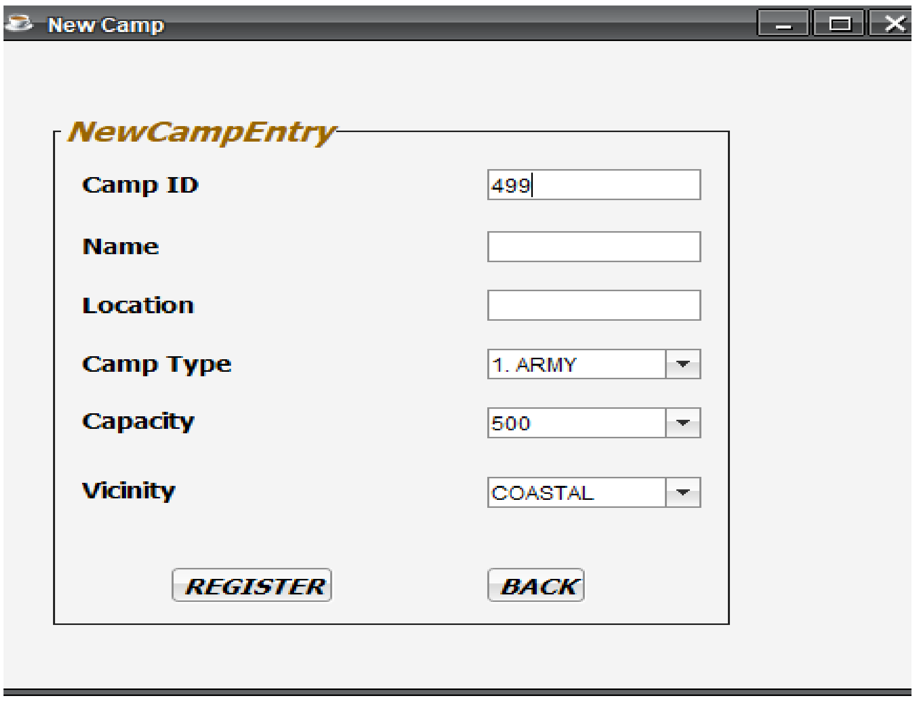
Fig 5: Home Page


New Soldier

NewSoldierEntry

Soldier_SSN	223
Name	
Department	1. ARMY
Rank	
Age	
Salary	

Fig 6: Soldier Entry Page

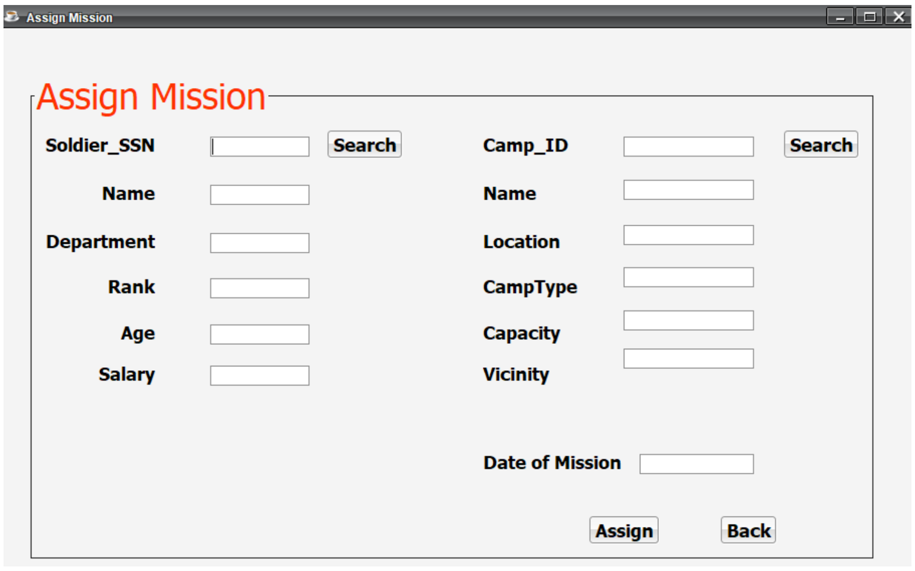


New Camp

NewCampEntry

Camp ID	499
Name	
Location	
Camp Type	1. ARMY
Capacity	500
Vicinity	COASTAL

Fig 7: New Army Camp Page



Assign Mission

Assign Mission

Soldier_SSN	
Name	
Department	
Rank	
Age	
Salary	
Camp_ID	
Name	
Location	
CampType	
Capacity	
Vicinity	
Date of Mission	

Fig 8: Mission Assignment Page

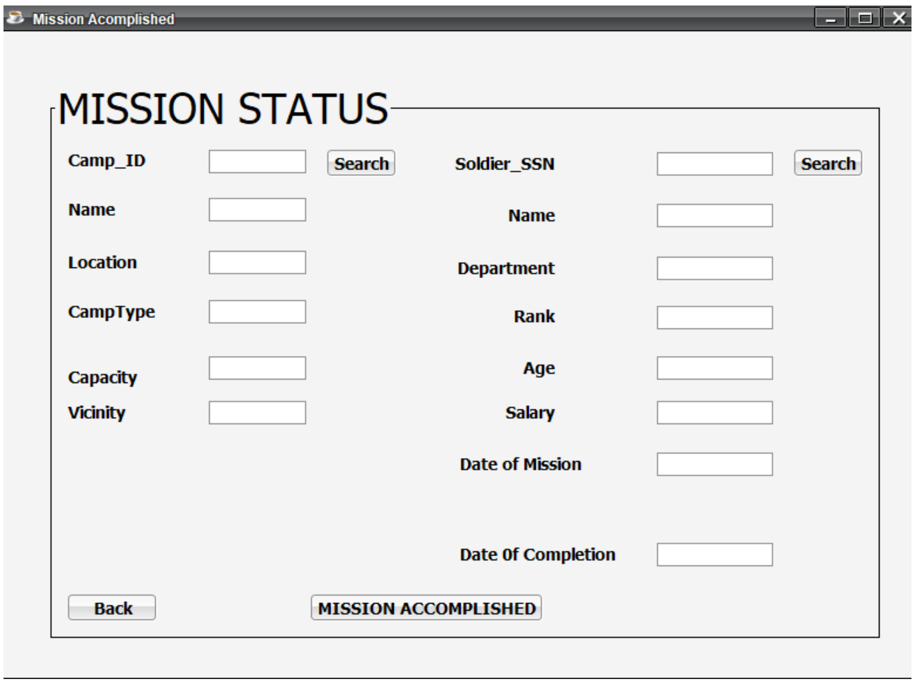


Fig 9: Mission Status Page

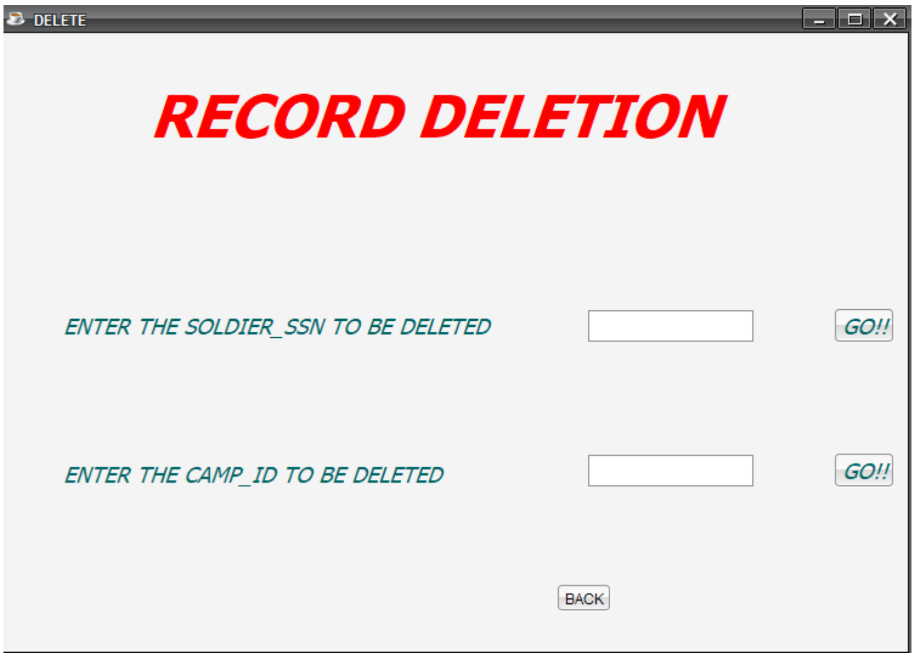

DELETE
RECORD DELETION
ENTER THE SOLDIER_SSN TO BE DELETED
GO!!
ENTER THE CAMP_ID TO BE DELETED
GO!!
BACK

Fig 10: Record Elimination page

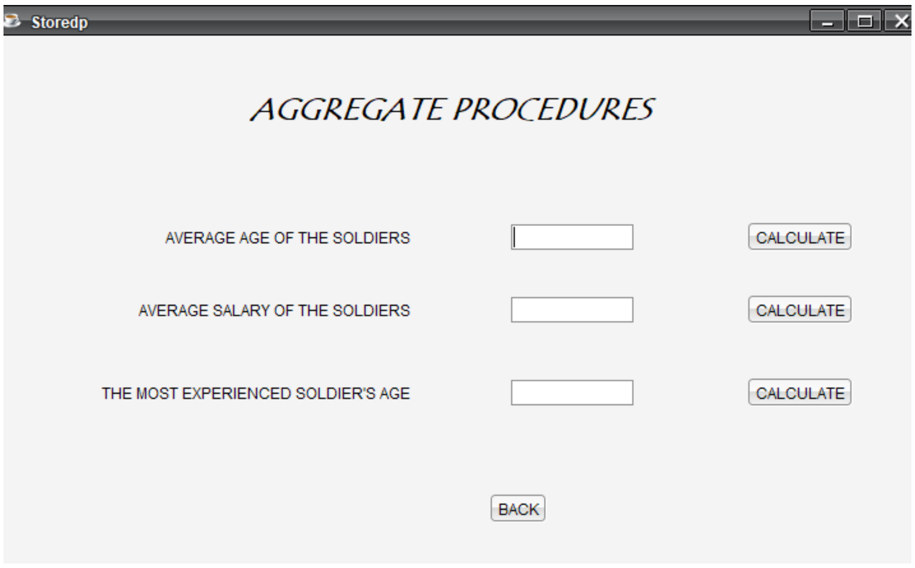
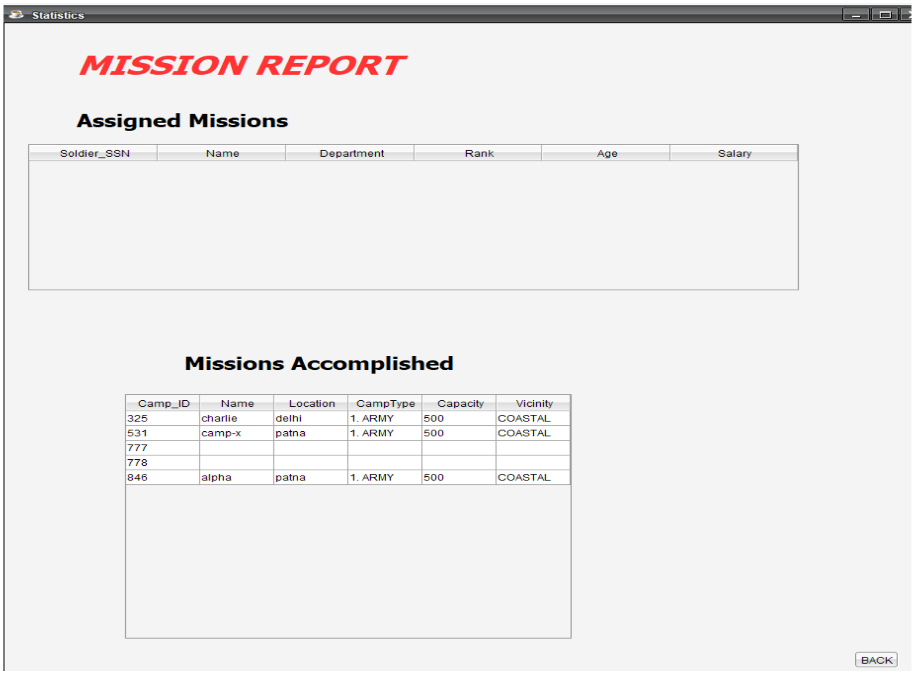

Storedp
AGGREGATE PROCEDURES
AVERAGE AGE OF THE SOLDIERS
CALCULATE
AVERAGE SALARY OF THE SOLDIERS
CALCULATE
THE MOST EXPERIENCED SOLDIER'S AGE
CALCULATE
BACK

Fig :11 Stored Procedures page



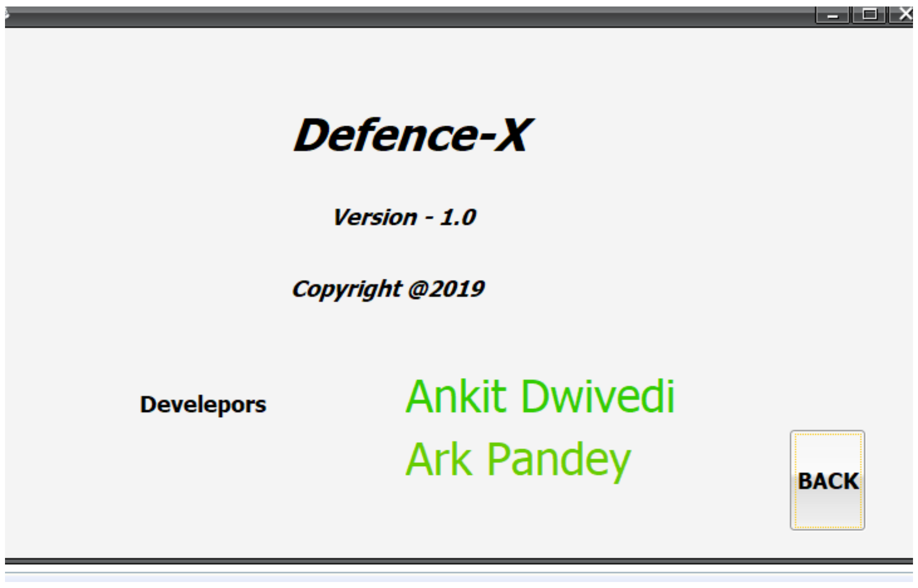
Assigned Missions

Soldier_SSN	Name	Department	Rank	Age	Salary

Missions Accomplished

Camp_ID	Name	Location	CampType	Capacity	Vicinity
325	charlie	delhi	1. ARMY	500	COASTAL
531	camp-x	patna	1. ARMY	500	COASTAL
777					
778					
846	alpha	patna	1. ARMY	500	COASTAL

Fig:12 Final Report Generator


Defence-X

Version - 1.0

Copyright @2019

Develepors

Ankit Dwivedi

Ark Pandey

Fig:13 Credits page

CHAPTER 6
CONCLUSION AND FUTURE ENHANCEMENTS

CONCLUSION

The system was mainly designed to reduce the manual work of updating and tracking and also make it easier for the user.
It also provides flexible and powerful reports regarding soldier details, posting details, weapon details ,etc.
Thus defence forces management system was implemented successfully

FUTURE ENHANCEMENT

- There is always a room for improvement in any software package,however good and efficient it may be.
- But the improvement thing is that the system should be flexible enough for further modifications.
- Considering this important factor, the system is designed in such away that provisions can be given for further enhancement withoutaffecting the system presently developed.

BIBILOGRAPHY

- Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
- Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill
- StackOverFlow website
- TutorialsPoint website