**TRAINING REPORT**

**On**

**NAKAL PROJECT**

**Submitted to** :MAHARAJA RANJIT SINGH PUNJAB TECHNICAL UNIVERSITY

in partial fulfilment of the requirement for the award of the degree of

**B.TECH**

**in**

**COMPUTER SCIENCE ENGINEERING**

**Submitted By**

**KANISHKA DHIR**

**Roll. No.15110169**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**GIANI ZAIL SINGH COLLEGE CAMPUS OF ENGINEERING &TECHNOLOGY, BATHINDA-151001**

**AUGUST 2017**

**TRAINING REPORT**

**On**

**NAKAL PROJECT**

**Submitted to** : MAHARAJA RANJIT SINGH PUNJAB TECHNICAL UNIVERSITY in partial fulfilment of the requirement for the award of the degree of

**B.TECH**

**in**

**COMPUTER SCIENCE ENGINEERING**

**Submitted By**

**KANISHKA DHIR**

**Roll. No. 15110169**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**GIANI ZAIL SINGH COLLEGE CAMPUS OF ENGINEERING &TECHNOLOGY, BATHINDA-151001**

**AUGUST 2017**

**CERTIFICATE FROM COMPANY**

****

**PREFACE**

Training is an integral part of B.Tech and each and every student has to undergo the training for 6 weeks in a company.

This record is concerned about our practical training during the Summer Vacations after the 2nd year. We have taken our Practical training in  **J2SE(Core Java)** During this training, we got to learn many new things about the industry and the current requirements of companies. This training proved to be a milestone in our knowledge of present industry. Every say and every moment was an experience in itself, an experience which theoretical study can’t provide.

**ACKNOWLEDGEMENT**

It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behaviour and acts during the course of study.

I express my sincere gratitude to ***Dr.Naresh Garg*** worthy HOD and ***Dr.Gurpreet Singh***, Training & Placement Incharge for providing me an opportunity to undergo summer training at ***Netmax Technologies Pvt. Ltd.***

I am thankful to ***Mr. Deepak Sharma*** for his support, cooperation, and motivation provided to me during the training for constant inspiration, presence and blessings. He provided his valuable suggestions and precious time in accomplishing my training report.

Lastly, I would like to thank the almighty and my parents for their moral support and my friends with whom I shared my day-to day experience and received lots of suggestions that my quality of work.

**KANISHKA DHIR**

**CANDIDATE’S DECLARATION**

I, Kanishka Dhir , Roll No. 15110169 , B.Tech (Semester- V) of the **Gaini Zail Singh Campus College of Engineering & Technology, Bathinda** hereby declare that the Training Report entitled “ **Nakal Project ”** is an original work and data provided in the study is authentic to the best of my knowledge.This report has not been submitted to any other Institute for the award of any other degree.

**Kanishka Dhir**

**(**Roll No. 15110169**)**

**Place: BATHINDA**

**Date: 13-11-2017**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | ***CONTENT PAGE***  **TOPIC** | **PAGE** |
| **1** | **INTRODUCTION**  1.1 Company Profile  1.2 Java   * Libraries * JRE * JVM * Features   1.3Framework and technologies used   * JavaFX * JDBC * Eclipse   1.4 Database: MySQL   * Queries * Software : Navicat premium | **1-13**  **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **11**  **12** |
| **2** | **INTRODUCTION TO PROJECT**   * 1. Project definition   2.2 Description of project  2.3 Modules Description  2.4Authentication  2.5Block Diagram  2.6 Limitations | **15-18**  **15**  **15**  **16**  **17**  **18**  **18** |
| **3** | **SCREENSHOTS** | **20-31** |
| **4** | **CONCLUSION& FUTURE SCOPE** | **33** |
| **5** | **BIBLIOGRAPHY AND REFERENCES** | **35** |

**LIST OF TABLES :**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **TABLE** | **PAGE** |
| **1.** | TABLE3.1 Registration Form | **28** |

**LIST OF FIGURES :**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **FIGURE** | **PAGE** |
| **1.** | FIG. 1.1:Logo of Java  FIG. 1.2: Compilation and Interpretation of Java  FIG. 1.3: Platform Independence of Java  FIG. 1.4: Eclipse  FIG. 1.5 : Eclipse Neon Workspace | **2**  **5**  **6**  **9**  **11** |
| **2.** | FIG. 2.1 : Project Layout | **17** |
| **3.** | FIG. 3.1:Login  FIG. 3.2:Menu  FIG. 3.3 :New Applicant  FIG. 3.4 :Status Report  FIG. 3.5 :Balance Report  FIG. 3.6 :Pending Report  FIG. 3.7 :Backup of Database  FIG. 3.8 :Recovery of Database  FIG. 3.9:Print | **19**  **19**  **20**  **20**  **21**  **21**  **22**  **23**  **23** |

**CHAPTER-1**

**INTRODUCTION**

***INTRODUCTION***

**1.1 Company Profile :**

Netmax is an emerging software development company. With over 5 years of experience in the industry, we understand what it takes to produce successful corporate image. The right image embodies the company’s fundamental essence and at the same time increases the customer’s identification with the brand.

The Netmax is an image development company that comprises of business, marketing and graphics experts who deliver innovative and eye-catching designs to globally positioned firms.

We use the most up to date visual communication and technology in order to combine the company culture, values and vision in an innovative and outstanding image.

TEAM: We have a committed team of professionals with experience in variety of tools and platforms. Our assistants keep themselves updated with the latest technology trends. Our skilled people share a common vision of growth. We do what we believe in, and we believe in what we do.

BELIEF: We believe in utilizing technology to make things simple and easy to use. We also believe in thinking ahead and delivering training about technology that can fit well with upcoming future technologies.

VALUES: We believe that the rich learning experiences our programs offer can effect powerful and positive transformation in individuals, in their communities, and in their institutions.  We work with our partners and sponsors worldwide to design and implement programs in which participants reach across – and live across – boundaries and borders.  And we do this in an atmosphere of mutual respect, organized collaboration, and fun.

**FACILITIES:**Some of the facilities offered in NETMAX are as follows:

* Trainers at the centre have 5 years of experience in all languages.
* To ensure the quality, all the trainers are certified in their own domain .
* For learning to be comprehensive, the requirement is to automate complex scenarios. Project assignments are given to students to help them realize real time experience.

**1.2 JAVA:**

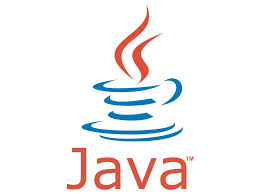
****

FIG 1.1 LOGO OF JAVA

Java is an object-oriented programming language with a built-in application programming interface (API) that can handle graphics and user interfaces and that can be used to create applications or applets. Because of its rich set of API's, similar to Windows, and its platform independence, Java can also be thought of as a platform in itself. Much of the syntax of Java is the same as C and C++. One major difference is that Java does not have pointers.

Java applications are typically [compiled](http://en.wikipedia.org/wiki/Compiler) to [byte code](http://en.wikipedia.org/wiki/Java_bytecode) that can run only on [Java virtual machine](http://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of [computer architecture](http://en.wikipedia.org/wiki/Computer_architecture).

In Java we distinguish between applications, which are programs that perform the same functions as those written in other programming languages, and applets, which are programs that can be embedded in a Web page and accessed over the Internet. Our initial focus will be on writing applications. When a program is compiled, a byte code is produced that can be read and executed by any platform that can run Java.

There were five primary goals in the creation of the java language:

* It should be "simple, object-oriented and familiar"
* It should be "robust and secure"
* It should be "architecture-neutral and portable"
* It should execute with "high performance"
* It should be "interpreted, threaded, and dynamic"

**1.2.1. LIBRARIES:**

* java.lang: Provides classes that are fundamental to the design of the Java programming language. The most important classes are Object, which is the root of the class hierarchy, and Class, instances of which represent classes at run time.
* java.io**:** Provides for system input and output through data streams, serialization and the file system. Unless otherwise noted, passing a null argument to a constructor or method in any class or interface in this package will cause a NullPointerException to be thrown.
* java.applet: Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
* java.sql**:** Provides the API for accessing and processing data stored in a data source (usually a relational database) using the Java programming language. This API includes a framework whereby different drivers can be installed dynamically to access different data sources.

## java.awt: Contains all of the classes for creating user interfaces and for painting graphics and images. A user interface object such as a button or a scrollbar is called, in AWT terminology, a component. The Component class is the root of all AWT components. See Component for a detailed description of properties that all AWT components share.

* java.security**:** Java security technology includes a large set of APIs, tools, and implementations of commonly used security algorithms, mechanisms, and protocols. The Java security APIs span a wide range of areas, including cryptography, public key infrastructure, secure communication, authentication, and access control. Java security technology provides the developer with a comprehensive security framework for writing applications, and also provides the user or administrator with a set of tools to securely manage applications.
* Java.imageio:A class containing static convenience methods for locating ImageReaders and ImageWriters, and performing simple encoding and decoding.
* Java.util Contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes (a string tokenizer, a random-number generator, and a bit array)
* Java.nioThe APIs of NIO were designed to provide access to the low-level I/O operations of modern operating systems. Although the APIs are themselves relatively high-level, the intent is to facilitate an implementation that can directly use the most efficient operations of the underlying platform. It contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes.
* Org.apache.poi:It provides pure Java libraries for reading and writing files in Microsoft Office formats, such as Word, PowerPoint and Excel.

# 1.2.2. JAVA RUNTIME ENVIRONMENT (JRE) :

The Java Runtime Environment (JRE) is a set of software tools for development of Java applications. It combines the Java Virtual Machine (JVM), platform core classes and libraries.  
  
JRE is part of the Java Development Kit (JDK), but can be downloaded separately. JRE was originally developed by Sun Microsystems Inc., a wholly owned subsidiary of Oracle Corporation. Also known as Java runtime.

JRE consists of the following components:

1. Deployment technologies, including deployment, Java Web Start and Java Plug-in.
2. User interface toolkits, including Abstract Window Toolkit (AWT), Swing, Java 2D, Accessibility, Image I/O, Print Service, Sound, drag and drop (DnD) and input methods.
3. Integration libraries, including Interface Definition Language (IDL), Java Database Connectivity (JDBC), Java Naming and Directory Interface (JNDI), Remote Method Invocation (RMI), Remote Method Invocation Over Internet Inter-Orb Protocol (RMI-IIOP) and scripting.
4. Other base libraries, including international support, input/output (I/O), extension mechanism, Beans, Java Automation Extensions (JMX), Java Native Interface (JNI), Math, Networking, Override Mechanism, Security, Serialization and Java for XML Processing (XML JAXP).

**1.2.3. JAVA VIRTUAL MACHINE (JVM) :**

A Java virtual machine (JVM) is [an abstract computing machine](http://en.wikipedia.org/wiki/Virtual_machine#Process_virtual_machines). There are three notions of the JVM: specification, implementation, and instance. An instance of the JVM can execute any executable computer program compiled into [Java bytecode](http://en.wikipedia.org/wiki/Java_bytecode). It is the code execution component of the [Java platform](http://en.wikipedia.org/wiki/Java_platform).

In the Java programming language, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into .class files by the javaccompiler. A .class file does not contain code that is native to your processor; it instead contains bytecodes — the machine language of the Java Virtual Machine (Java VM). The java launcher tool then runs your application with an instance of the Java Virtual Machine.

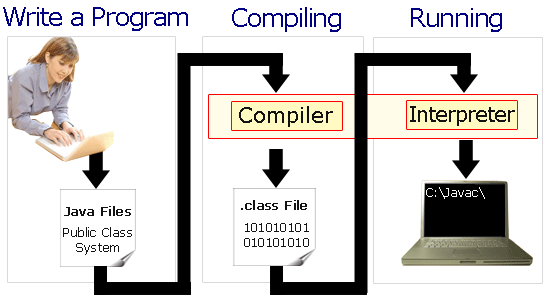


FIG.1.2: Compilation and Interpretation of Java

Because the Java VM is available on many different operating systems, the same .class files are capable of running on Microsoft Windows, the Solaris™ Operating System (Solaris OS), Linux, or Mac OS. Some virtual machines, such as the [Java SE HotSpot at a Glance](http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136373.html), perform additional steps at runtime to give your application a performance boost. This include various tasks such as finding performance bottlenecks and recompiling (to native code) frequently used sections of cod

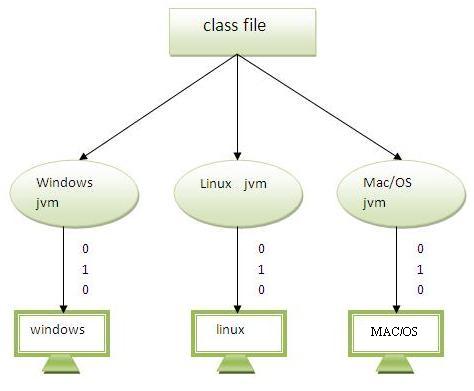


Fig 1.3: Platform Independence of java

**1.2.4. FEATURES OF JAVA :**

* **Platform Independent**.:

A platform is the hardware or software environment in which a program runs. There are two types of platforms software-based and hardware-based. Java provides software-based platform. The Java platform differs from most other platforms in the sense that it's a software-based platform that runs on top of other hardware-based platforms. It has two components:

1. Runtime Environment
2. API(Application Programming Interface)

Java code can be run on multiple platforms e.g. Windows, Linux, Sun Solaris, Mac/OS etc. Java code is compiled by the compiler and converted into bytecode . This bytecode is a platform independent code because it can be run on multiple platforms i.e. Write Once and Run Anywhere (WORA).

* Robust and Secure :

Robust simply means strong. Java uses strong memory management. There are lack of pointers that avoids security problem. There is automatic garbage collection in java. There is exception handling and type checking mechanism in java. All these points makes java robust.

* Distributed :

We can create distributed applications in java. RMI and EJB are used for creating distributed applications. We may access files by calling the methods from any machine on the internet.

* Multithreaded and Interactive :

A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it shares the same memory. Threads are important for multi-media, Web applications etc.

* Dynamic and Extensible Code :

Java is faster than traditional interpretation since byte code is "close" to native code still somewhat slower than a compiled language (e.g., C++).

* Architectural Neutral :

There is no implementation dependent features e.g. size of primitive types is set.

**1.3 FRAMEWORK AND SOFTWARE USED:**

**1.3.1 JAVAFX :**

JavaFX is a Java library used to build Rich Internet Applications. The applications written using this library can run consistently across multiple platforms. JavaFX provides a rich set of graphics and media API’s and it leverages the modern **Graphical Processing Unit** through hardware accelerated graphics.

JavaFX features include:

* Written in Java.
* JavaFX features a language known as FXML, which is a HTML like declarative markup language. The sole purpose of this language is to define a user Interface.
* provides an application Scene Builder.
* embed Swing content using the **Swing Node** class.
* **Integrated Graphics library** − JavaFX provides classes for **2d** and **3d** graphics.
* **Rich set of API’s** − JavaFX library provides a rich set of API’s to develop GUI applications, 2D and 3D graphics, etc. This set of API’s also includes capabilities of Java platform.

## 1.3.2 JDBC :

The JDBC is a set of the database access classes. The term JDBC stands for "Java database Connectivity" and it was developed by Javasoft.

JDBC technology is an API (application program interface) that allows visual access to any tabular data source from java programming languages by mean of some connecting software called drives. it provide cross-DBMS connectivity to wide range of SQL database JDBC allows java applets, servlets and applications to access data in famous database automation system.

It also provides access to other tabular data source such as spread sheets or flat files. The JDBC API allows developer to take advantage of the java platforms “write once, run anywhere" capabilities for industrial strings cross platforms applications that require access to enterprise data. The JDBC API is the industry standard for data base-independent connectivity’s between the java programming language and a wide range of the database. The JDBC API makes it possible to do these things

* Establish a connection with a database or access tabular data source.
* Send SQL statements and process the results.

Because JDBC is a standard specification, a Java program that uses the JDBC API can connect to any database automation system (DBMS) for which there is a JDBC driver.

**1.3.3 ECLIPSE :**

****

FIG 1.4 ECLIPSE

Eclipse is a multi-language [software development environment](http://en.wikipedia.org/wiki/Software_development_environment) comprising an [integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) and an extensible [plug-in](http://en.wikipedia.org/wiki/Plug-in_(computing)) system. It is written mostly in [Java](http://en.wikipedia.org/wiki/Java_(programming_language)). It can be used to develop applications in Java and, by means of various plug-ins, other [programming languages](http://en.wikipedia.org/wiki/Programming_language) including [Ada](http://en.wikipedia.org/wiki/Ada_(programming_language)), [C](http://en.wikipedia.org/wiki/C_(programming_language)), [C++](http://en.wikipedia.org/wiki/C%2B%2B), [COBOL](http://en.wikipedia.org/wiki/COBOL), [Haskell](http://en.wikipedia.org/wiki/Haskell_(programming_language)), [Perl](http://en.wikipedia.org/wiki/Perl), [PHP](http://en.wikipedia.org/wiki/PHP), [Python](http://en.wikipedia.org/wiki/Python_(programming_language)), [R](http://en.wikipedia.org/wiki/R_(programming_language)), [Ruby](http://en.wikipedia.org/wiki/Ruby_(programming_language)) (including [Ruby on Rails](http://en.wikipedia.org/wiki/Ruby_on_Rails) framework), [Scala](http://en.wikipedia.org/wiki/Scala_(programming_language)), [Clojure](http://en.wikipedia.org/wiki/Clojure), [Groovy](http://en.wikipedia.org/wiki/Groovy_(programming_language)) and [Scheme](http://en.wikipedia.org/wiki/Scheme_(programming_language)). It can also be used to develop packages for the software [Mathematica](http://en.wikipedia.org/wiki/Mathematica). Development environments include the Eclipse Java development tools (JDT) for Java, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

The initial [codebase](http://en.wikipedia.org/wiki/Codebase) originated from [VisualAge](http://en.wikipedia.org/wiki/VisualAge). The Eclipse SDK (which includes the Java development tools) is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

ARCHITECHTURE:

The Eclipse Platform subproject provides the core frameworks and services upon which all plug-in extensions are created. It also provides the runtime in which plug-ins are loaded, integrated, and executed. The primary purpose of the Platform subproject is to enable other tool developers to easily build and deliver integrated tools.

It can deal with any type of resource (Java files, C files, Word files, HTML files, JSP files, etc) in a generic manner but doesn't know how to do anything that is specific to a particular file type. The Eclipse platform, by itself, doesn't provide a great deal of end-user functionality - it is what it enables that is interesting. The real value comes from tool plug-ins for eclipse that "teach" the platform how to work with these different kinds of resources. This pluggable architecture allows a more seamless experience for the end user when moving between different tools than ever before possible.

The Eclipse platform defines a set of frameworks and common services that collectively make up "integration-ware" required to support a comprehensive tool integration platform. These services and frameworks represent the common facilities required by most tool builders including a standard workbench user interface and project model for managing resources, portable native widget and user interface libraries, automatic resource delta automation for incremental compilers and builders, language-independent debug infrastructure, and infrastructure for distributed multi-user versioned resource automation.

In addition, the Eclipse platform defines a workbench user interface and a set of common domain-independent user interaction paradigms that tool builders plug into to add new capabilities. The platform comes with a set of standard views which can be extended by tool builders. Tool builders can both add new views, and plug new domain-specific capability into existing views.

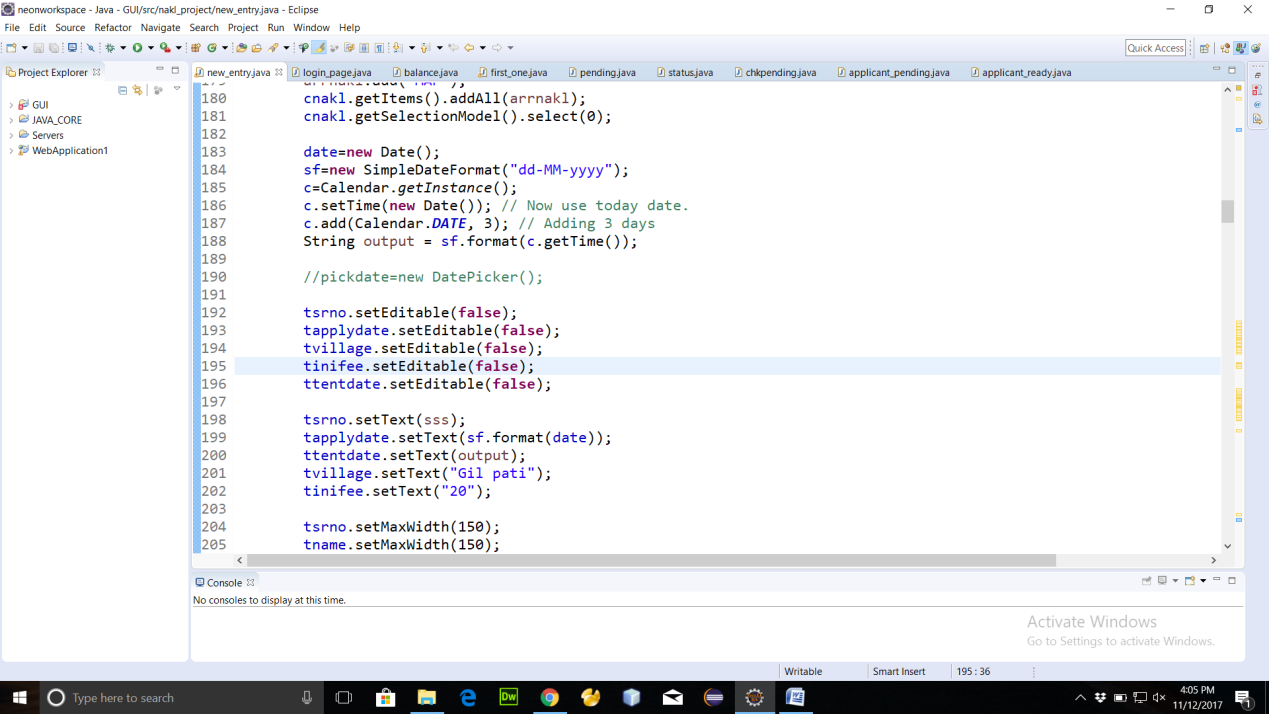


Fig. 1.5 Eclipse Neon Workspace

**1.4 DATABASE :MySQL :**

MySQL is a [relational database automation system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS), and ships with no [GUI](http://en.wikipedia.org/wiki/Graphical_user_interface) tools to administer MySQL databases or manage data contained within the databases. Users may use the included [command line](http://en.wikipedia.org/wiki/Command_line) tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. It is named after co-founder [Michael Widenius](http://en.wikipedia.org/wiki/Michael_Widenius)'s daughter, My, SQL stands for structured query language, better known as "sequel".

It is used for:

1) Querying a database by editing the SQL statement

2) Querying a database with a program.

3) Defining the data organization.

4) Administrating data.

5) Accessing multiple data server.

6) Managing transaction.

**1.4.1 QUERIES:**

These are used to retrieve the data based on specific criteria. This is the most important element of sql.

Some basic queries in sql:

* Creating a table:

Syntax: create table table\_name(column\_name1 datatype(size),column\_name2 datatype(size),……,column\_namedatatype(size));

* Inserting into a table:

Syntax: insert into table\_name values(x,y,z);

* Updating a table:

Syntax: update table\_name set column\_name=’value’ where column\_name=’value’;

* Deleting a column of table:

Syntax: delete from table\_name where column\_name=value;

* Selecting columns of table

Syntax: select \* from table\_name;

**1.4.2SOFTWARE USED: NAVICAT LITE :**

Navicat Lite is a multi-connections Database Administration tool allowing you to connect to MySQL, PostgreSQL and Oracle databases simultaneously within a single application, making database administration to multiple kinds of database so easy. It supports most of the features in MySQL, Oracle and PostgreSQL including Event, Trigger, Function/Procedure, View, etc. NavicatLite is available on three platforms - Microsoft Windows, Mac OS X and Linux. It can connect users to local/remote server, providing several utility tools such as Data/Structure Synchronization, Import/Export, Backup and Report to facilitate the process for data maintenance.Navicat Features:

Latest MySQL versions support

* Compatible with any MySQL server version.
* Support of Events.
* Support of Triggers.
* Support of Procedures and Functions.
* Support of Views.
* VARCHAR Data Type support. Value from 0 to 65,535 in 3.0.3 and later versions.
* BIT Data Type support.
* BINARY/VARBINARY Data Type support
* Blob data type.

**CHAPTER-2**

**INTRODUCTION**

**TO PROJECT**

***INTRODUCTION TO PROJECT***

**2.1 PROJECT DEFINITION:**

My project named **“NAKAL PROJECT”** aims at automating all the work regarding handling the nakals generated in DC office .This software creates a system to automate the services which would be more flexible than the earlier system. This project is meant to reduce the workload and complexities that the owners have to face during the handling the people and the nakals from pattwari. If any modification is required that is to be also done manually.

**2.2 DESCRIPTION OF PROJECT:**

The purpose of this project is provide the owners an automated system so that they can manage the nakals generated at DC office bathinda by providing a very secure environment to the customer as well as to the operator for the nakal they require.

**2.2.1 Existing System:**

**The existing system in which when a person comes to for nakal to be generated, he just provides his details to the person incharge and he notes in his register manually, then passes the details to pattwari who after the work is done hands over the nakal to the operator who charges the customer according the number of pages. Sounds pretty simple.**

**2.2.2 Limitations in Existing System:**

1)  There is no fixed number of days for the customer to receive his delivery.

2)  If there is some enquiry operator has to search whole register .

3)  The complete current system is manual system and it will not provide any kind of security to the data.

4) The calculation of fare is done manually takes a lot of time and error.

5) Operator has full flexibility to charge more from any customer which is cheating to the authority.

6) Operator has full flexibility over the entire balance of the record and can change it according to his needs .

**2.2.3 Proposed System:**

In the proposed system we have the following new implementations: There will be specific login id and password for the operator. The operator will generate a receipt when a customer from gill patti comes to generate nakal, two copies will be printed one for the customer and the other for operator. The receipt contains a unique id number for every customer, the tentative delivery date and an initial fee which will be charged from customer. When the nakal is ready, the operator makes the status as ready to deliver and when the customer comes he pays the remaining amount and final delivery date is stored. The operator can even check his balance and take a print of the excel report. The pending nakals can also be found plus a backup of whole database in the form of an excel sheet can also be taken and recovered.

**2.2.4 Advantages over Existing System :**

The Advantages of the proposed system are:

1) Feel of security for customer as no one else can tale his nakal.

2) Operator can have the balance of his daily record-how many applied today for nakal, how many nakals status is ready to deliver and how many are delivered today.

3) There is transparency for the income either as in initial fee or the total fee.

4) Helps authority manage his employees properly as they cannot change anything in this software.

5) Allows the operator to have a backup for his data in case his system is crashed.

**2.3 Modules Description :**

1. **Customer registration**
2. **Set Status**
3. **Check Balance**
4. **Check Pending**
5. **Take Backup**
6. **Recover**

**DESCRIPTION:-**

1. **Customer registration:** Operator will store the necessary information of every customer in their database which can be used for further communication with the customer. Each customer is given an auto-generated id.
2. **Set Status: Operator can change the status of the nakal received for a particular person as the nakal comes from pattwari and when the person collects the nakal in hand from the operator.**
3. **Check Balance: Operator can check the balance from a particular date to another to check the number of nakals applied, ready to deliver and those delivered with confirming the amount he has deposit.**
4. **Check Pending:** Operator can check the nakals pending with the pattwari,so he can have a record of nakals whose delivery date has been expired.
5. **Backup:** The operator can transfer all the records to an excel sheet created where his exe file is kept so that if any time his system goes down, his data is kept secure.
6. **Recovery: In this module, operator can retrieve all the records of previous customers from the backup file to his database when his system becomes active again.**
7. **Print :**When a new applica**nt registers, two copies are printed, one for customer and another for operator(Printout is not taken until the record is saved).One can take print out of the status report, or the balance report as well as of the pending report.**

**2.4 AUTHENTICATION**:

1) Login to the system through the first page of the application.

2) In case if operator enters the wrong password, an alert is displayed and access is denied.

3) There is no way to retrieve the old password.

**2.5 BLOCK DIAGRAM**

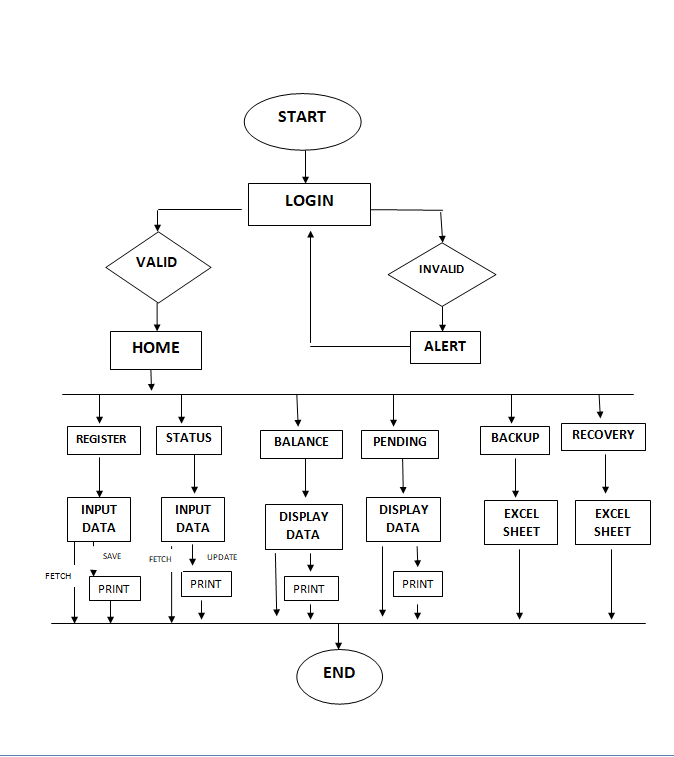


FIG. 2.1 PROJECT LAYOUT

**2.6LIMITATIONS**

The following are the limitations of the project:

* This app works only on computer.
* The employees must be enough educated so that they can use computer properly.
* Login sessions of operators can’t be recorded.

**CHAPTER-3**

**SCREENSHOTS**

***SCREEN SHOTS***

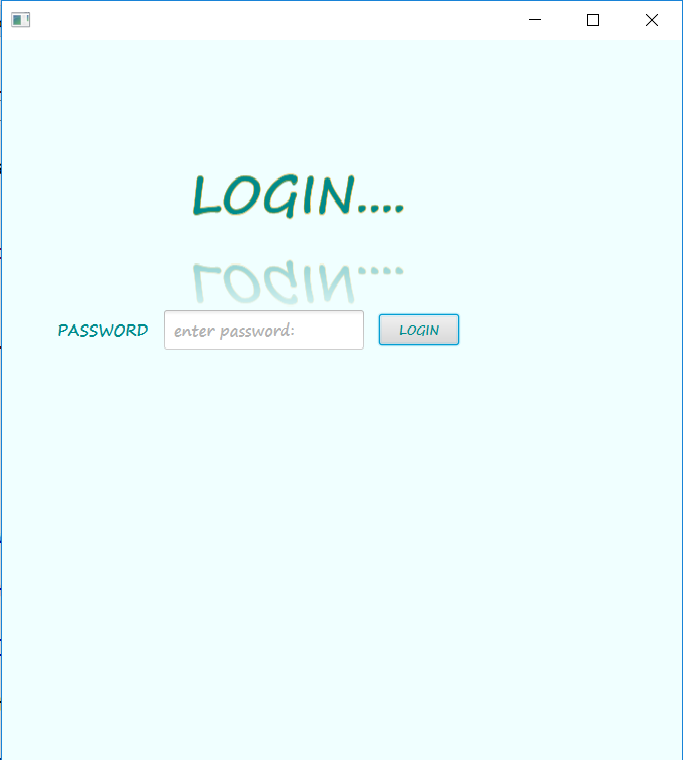
****

FIG 3.1 LOGIN

****

FIG 3.2 MENU

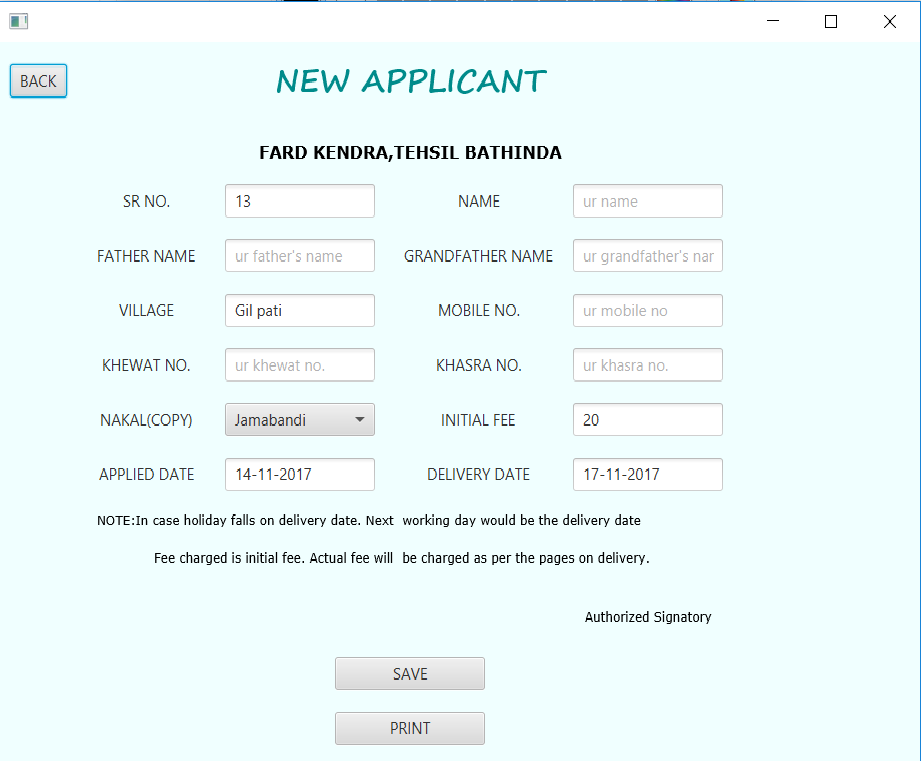


FIG 3.3 NEW APPLICANT

**REGISTRATION:** He will store the necessary information of every customer in their database which can be used for further communication with the customer.

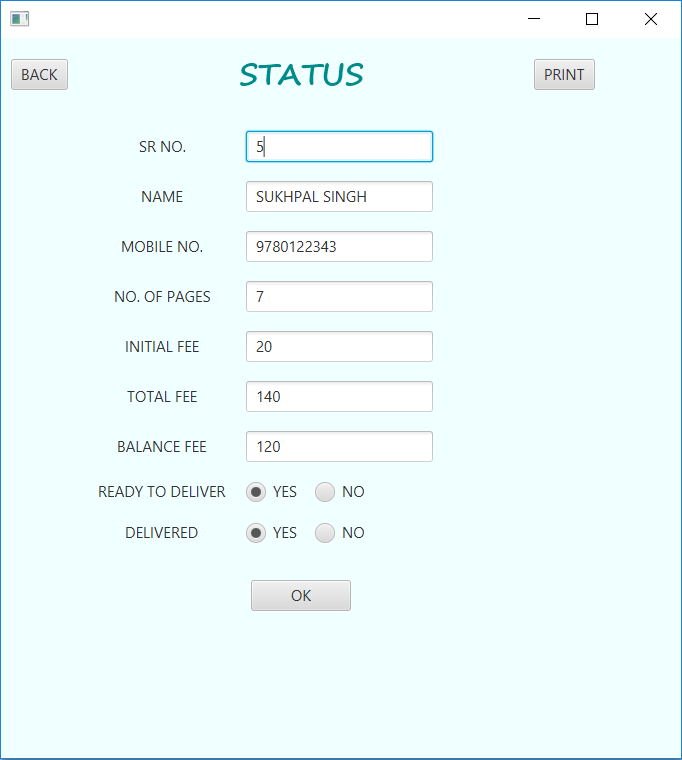
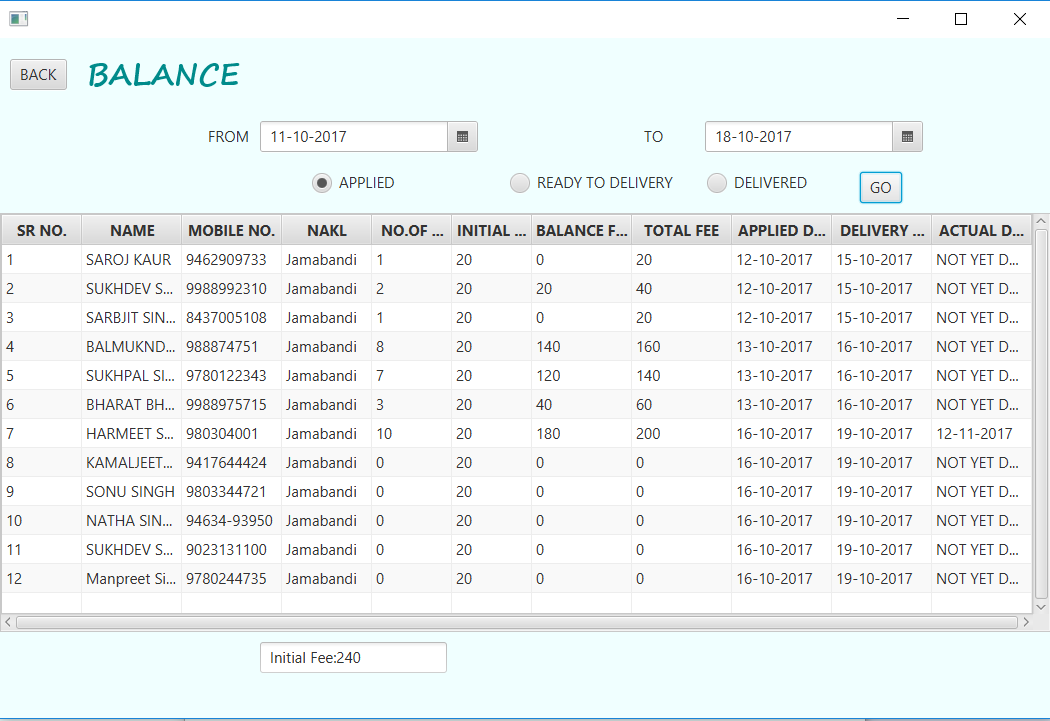
****

FIG 3.4 STATUS REPORT

**STATUS:** He can change the status of report from ready to deliver to delivered.

****FIG 3.5 BALANCE REPORT

**BALANCE:** He can check the list of applicants according to those who applied and check their total initial fee and those nakals which are delivered along with there total fee.

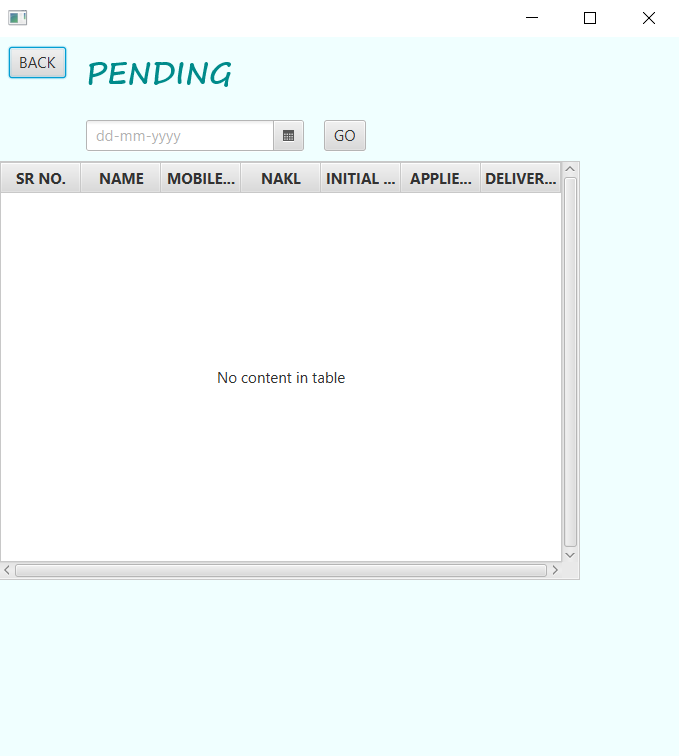
****

FIG 3.6 PENDING REPORT

**PENDING:** He can check the list of applicants according to those who applied according to there tentative delivery date and have yet not received nakals from the pattwari.

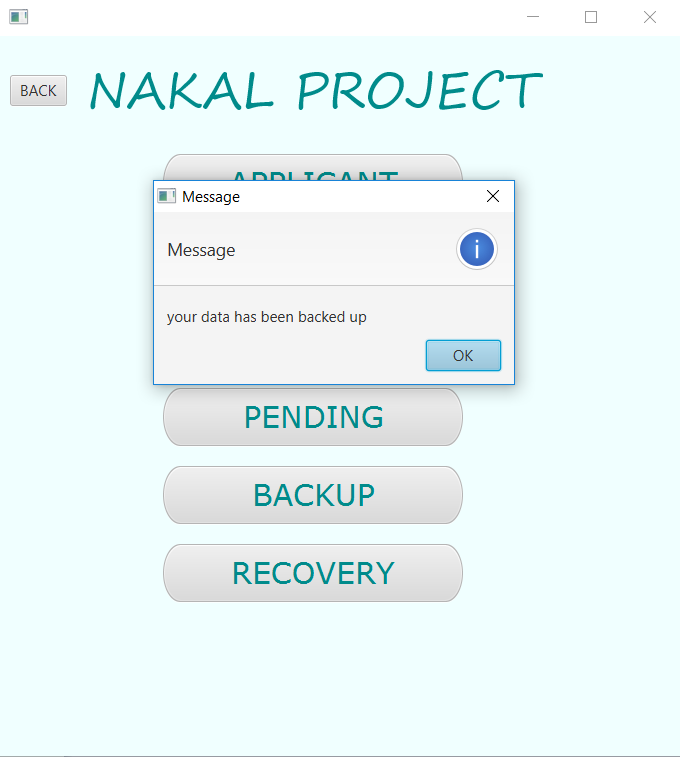
****

FIG 3.7 BACKUP OF DATABASE

**BACKUP:** He can take backup of his data in case his system stops working. Backup is taken in form of an excel sheet.

****

FIG 3.8 RECOVERY OF DATABASE

**RECOVERY:** He can take recovery of his data in case his system had stopped working and now wants to resume the working again. Data is recovered from the backup excel sheet.Even his table is deleted that can be recovered too.

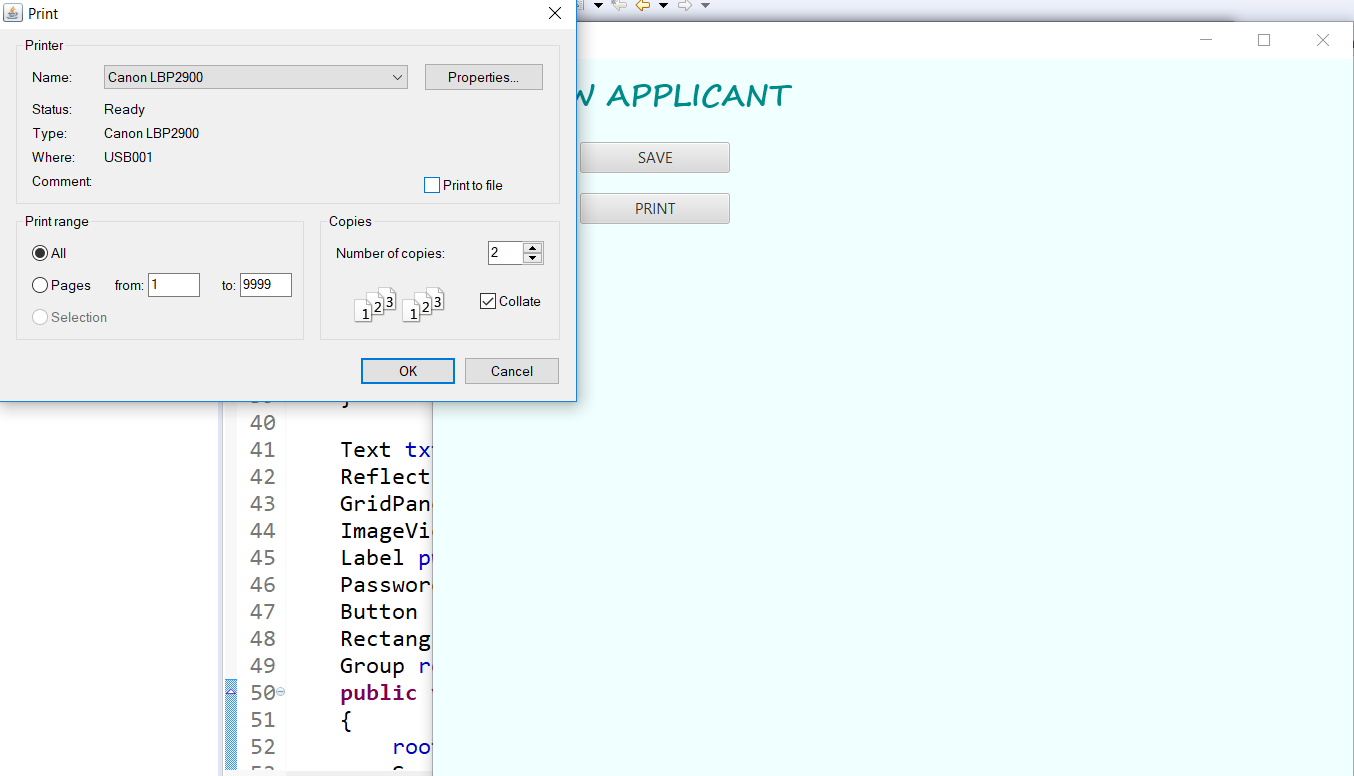


FIG 3.9 PRINT

**PRINT :**In every frame there is a print button , which will open a dialog box in which if there is any printer attached with system the it will detect it and execute the print command.

**3.1 TABLES**

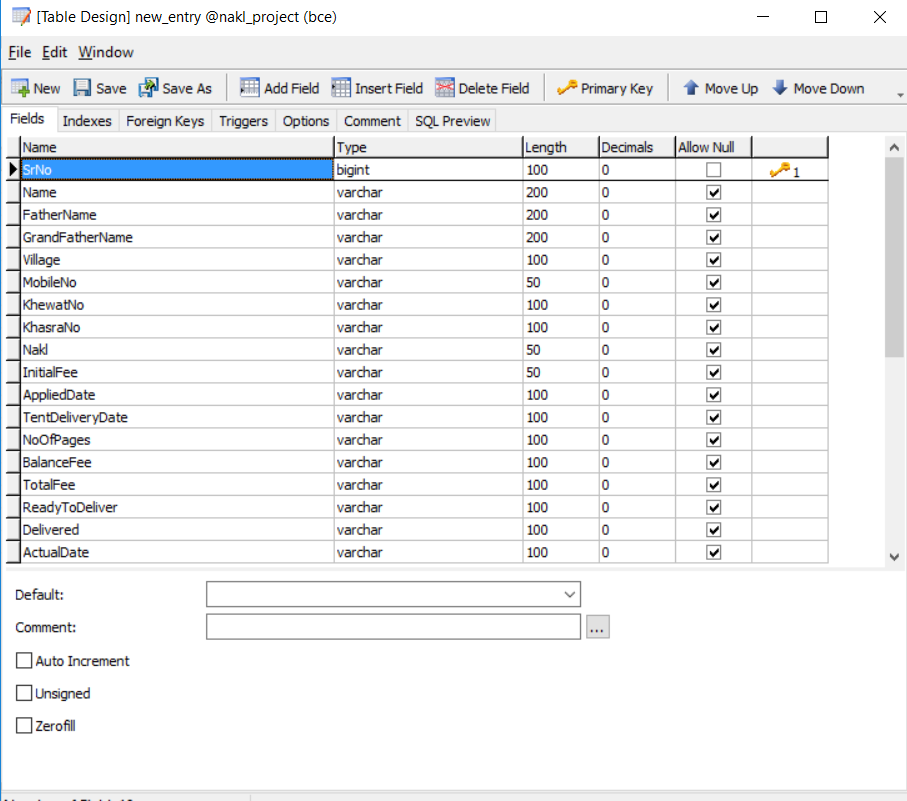
****

TABLE 3.1 REGISTRATION TABLE

**REGISTRATION :** There is only one table used for this whole project named new\_entry with the fields specified above. During applying a new customer, some of the fields are marked as null or not yet specified, later on in the status report they are marked as per given data.

**CHAPTER-4**

**CONCLUSION &**

**FUTURE SCOPE**

***CONCLUSION AND FUTURE SCOPE***

The system according to the expectations of users is easy to use and take less time. It is consistent and unambiguous. The system produced is user friendly, understandable and easy to use so that the users of the system can easily learn to use the system. The response time of the system is very less and takes less time to execute. The system will be able to handle undesirable situations and errors encountered at various levels e.g. if the password is incorrect user will not be allowed to login , if the person forgets to take save the record and tries to take print first.

Presently there are no special rights available to operators. In today’s time with increase in number of people at dc office automated generation of nakals are being the basic necessity. Further the project can be extended to include better customer interaction system, in which customers can also register online.If given a chance to do it again the project could have been better, much safe, and confidential and prevented from unauthorized access, more flexible in nature so that likely changes and alterations can be easily made. High Security checks can be added so that its privacy and integrity can be made.

**CHAPTER-5**

**BIBLIOGRAPHY & REFERENCES**

***BIBLIOGRAPHY AND REFERENCES***

The following books are used to fulfil the requirements of project and are helpful in understanding, development and the maintenance of the project.

* REAL JAVA by Rajesh.K.Bansal

REFERENCES:

* http://www.w3schools.com
* http://docs.oracle.com
* http://www.programmerinterview.com
* http://javatpoint.com
* www.google.co.in
* www.wikipedia.com
* www.eclipse.org
* www.studytonight.com