DECLARATION BY THE CANDIDATE

I hereby declare that the project report entitled “COMPLEX ADMINISTRTION” submitted by me to NANDI INSTITUTE OF MANAGEMENT AND SCIENCE in partial fulfilment of the requirement for the award of the degree BCA is a record of bonafide project work carried out by me under the guidance of Mrs. DEEPA T .I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any degree in this institute or any other institute or university.

Date signature of the student

ACKNOWLEDGEMENT

I thank this occasion to thank God ,almighty for blessing me with his grace and taking endeavour to a successful culmination.

I extend my sincere and heartfelt thanks to my institution, Nandi Institute of Management and Science for providing me the facilities and inspiration that has made this project success.

I express my heartfelt thanks to Dr. Mansur Baigh of Nandi Institute of Management and Science, Bellary for providing me a congenial atmosphere to work in.

I express my heartfelt thanks to Mr.Jagadeesh Babu , Dean of Nandi Institute of Management and science, Ballari for providing me a congenial atmosphere to work in.

I thank my Head of Department, Mr.Zameer Ahmed for his useful guidance and encouragement throughout the project development period.

I thank my guide, Mrs. Deepa T and Mr. G.Vannurswamy who guided us so that I may have smooth project coordination.

I thank all the teaching and non-teaching staff of the department for being there.

I thank my parents and my friends for all the motivation and encouragement that they showered upon us

ABSTRACT

Complex Administration” is window-based application system .This application is based on the particular complex “SADGURUM COMPLEX” .In this complex office rooms are given for rent. The main objective of this application is to computerize the paper work in the system to automate the work. The Computerization is done so that the storage of all the details regarding tenants will be stored in the system which makes system centralized and the chance of duplication of any data is minimized

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1OBJECTIVE:

“Complex Administration” is an application developed for maintaining the details of tenants. Here the administrator will record the details of the tenants. Administrator has to login first containing username and password if he doesn’t have an account he has to register and then login .This application will be helpful to store the details of the tenants and also the monthly payment of the tenants can be stored easily and can verify whether tenant has paid his monthly payment or not

CHAPTER 2

SYSTEM ANALYSIS

INTRODUCTION

Analysis can be defined as breaking up of any whole so as to find out their nature, function etc. It defines design as to make preliminary sketches of; to sketch a pattern or outline for plan. To plan and carry out especially by artistic arrangement or in a skilful wall. System analysis and design can be characterized as a set of techniques and processes, a community of interests, a culture and an intellectual orientation.

The various tasks in the system analysis include the following.

□ Understanding application.

□ Planning.

□ Scheduling.

□ Developing candidate solution.

□ Performing trade studies.

□ Performing cost benefit analysis.

□ Recommending alternative solutions.

□ Selling of the system.

□ Supervising, installing and maintaining the system.

This system manages to the analysis of the report creation and develops manual entry of the tenant details. First design the register and login form and then design the forms to enter the details of the tenants and to store the details of payment made by the tenants.

2.1EXISTING SYSTEM

The Existing system is a manual entry of the tenants. Here the tenant details will be carried out in the hand written registers. It will be a tedious job to maintain the record for the user. The human effort is more here. The retrieval of the information is not as easy as the records are maintained in the hand written registers. This application requires correct feed on input into the respective field. Suppose the wrong inputs are entered, the application resist to work. so the user find it difficult to use.

2.2PROPOSED SYSTEM

To overcome the drawbacks of the existing system, the proposed system has been evolved. This project aims to reduce the paper work and to save time. The system provides with the best user interface. The efficient reports can be generated by using this proposed system.

Advantages of Proposed System

□ It is trouble-free to use.

□ It is a relatively fast approach to enter tenant details

□ Is highly reliable,

□ st user Interface

2.3 FEASIBILITY STUDY:

Feasibility analysis begins once the goals are defined. It starts by generating broad possible solutions, which are possible to give an indication of what the new system should look lime. This is where creativity and imagination are used. Analysts must think up new ways of doing things-generate new ideas. There is no need to go into the detailed system operation yet. The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization. It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need significantly change the original goal.

Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined. They are,

2.3.1 Economically Feasibility:

Development of this application is highly economically feasible. The only thing to be done is making an environment with an effective supervision.

It is cost effective in the sense that has eliminated the paper work completely. The system is also time effective because the calculations are automated which are made at the end of the month or as per the user requirement.

2.3.2Technical feasibility:

The technical requirement for the system is economic and it does not use any other additional Hardware and software. Technical evaluation must also assess whether the existing systems can be upgraded to use the new technology and whether the organization has the expertise to use it.

Install all upgrades framework into the .Net package supported widows based application. this application depends on Microsoft office and intranet service ,database. Enter their attendance and generate report to excel sheet.

2.3.2 Operational Feasibility:

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. Technical performance include issues such as determining whether the system can provide the right information for the Department personnel student details, and whether the system can be organized so that it always delivers this information at the right place and on time using intranet services. Acceptance revolves around the current system and its personnel.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 HARDWARE REQUIREMENTS (Minimum Requirement)

- Minimum RAM:-1GB
- Hard Disk:-128 GB
- Processor :-Intel Pentium 4(1.50 GHZ) or above

3.2 SOFTWARE REQUIREMENTS (minimum Requirement)

- Operating system :Windows10
- Front-Design: java swings
- Front-End Language :programming language
- Back-End : java DB
- Back-End Connectivity: java Derby

CHAPTER 4

SOFTWARE DESCRIPTION

4.1 Java

What is Java?

Java is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centres, game consoles, scientific supercomputers, cell phones, etc.

What is Java Platform?

Java Platform is a collection of programs that help programmers to develop and run Java programming applications efficiently. It includes an execution engine, a compiler, and a set of libraries in it. It is a set of computer software and specifications. James Gosling developed the Java platform at Sun Microsystems, and the Oracle Corporation later acquired it.

Java Definition and Meaning

Java is a multi-platform, object-oriented, and network-centric language. It is among the most used programming language. Java is also used as a computing platform.

It is considered as one of the fast, secure, and reliable programming languages preferred by most organizations to build their projects.

What is Java used for?

Here are some important Java applications:

- It is used for developing Android Apps
- Helps you to create Enterprise Software
- Wide range of Mobile java Applications
- Scientific Computing Applications
- Use for Big Data Analytics
- Java Programming of Hardware devices
- Used for Server-Side Technologies like Apache, JBoss, GlassFish, etc.

History of Java Programming Language

Here are important landmarks from the history of the Java language:

- The Java language was initially called OAK.
- Originally, it was developed for handling portable devices and set-top boxes. Oak was a massive failure.
- In 1995, Sun changed the name to "Java" and modified the language to take advantage of the burgeoning www (World Wide Web) development business.
- Later, in 2009, Oracle Corporation acquired Sun Microsystems and took ownership of th
- e key Sun software assets: Java, MySQL, and Solaris.

Java Features

Here are some important Java features:

- It is one of the easy-to-use programming languages to learn.
- Write code once and run it on almost any computing platform.

- Java is platform-independent. Some programs developed in one machine can be executed in another machine.
- It is designed for building object-oriented applications.
- It is a multithreaded language with automatic memory management.
- It is created for the distributed environment of the Internet.
- Facilitates distributed computing as its network-centric.

Components of Java Programming Language

A Java Programmer writes a program in a human-readable language called Source Code. Therefore, the CPU or Chips never understand the source code written in any programming language.

These computers or chips understand only one thing, which is called machine language or code. These machine codes run at the CPU level. Therefore, it would be different machine codes for other models of CPU.

However, you need to worry about the machine code, as programming is all about the source code. The machine understands this source code and translates them into machine understandable code, which is an executable code.

All these functionalities happen inside the following 3 Java platform components:

Java Development kit (JDK)

JDK is a software development environment used for making applets and Java applications. The full form of JDK is Java Development Kit. Java developers can use it on Windows, macOS, Solaris, and Linux. JDK helps them to code and run Java programs. It is possible to install more than one JDK version on the same computer.

Why use JDK?

Here are the main reasons for using JDK:

- JDK contains tools required to write Java programs and JRE to execute them.
- It includes a compiler, Java application launcher, Applet viewer, etc.
- Compiler converts code written in Java into byte code.
- Java application launcher opens a JRE, loads the necessary class, and executes its main method.

Java Virtual Machine (JVM):

Java Virtual Machine (JVM) is an engine that provides a runtime environment to drive the Java Code or applications. It converts Java byte code into machine language. JVM is a part of the Java Run Environment (JRE). In other programming languages, the compiler produces machine code for a particular system. However, the Java compiler produces code for a Virtual Machine known as Java Virtual Machine.

Why JVM?

Here are the important reasons of using JVM:

- JVM provides a platform-independent way of executing Java source code.
- It has numerous libraries, tools, and frameworks.
- Once you run a Java program, you can run on any platform and save lots of time.
- JVM comes with JIT (Just-in-Time) compiler that converts Java source code into low-level machine language. Hence, it runs faster than a regular application.

Java Runtime Environment (JRE)

JRE is a piece of software that is designed to run other software. It contains the class libraries, loader class, and JVM. In simple terms, if you want to run a Java program, you need JRE. If you are not a programmer, you don't need to install JDK, but just JRE to run Java programs.

Why use JRE?

Here are the main reasons of using JRE:

- JRE contains class libraries, JVM, and other supporting files. It does not include any tool for Java development like a debugger, compiler, etc.
- It uses important package classes like math, swing, util, lang, awt, and runtime libraries.
- If you have to run Java applets, then JRE must be installed in your system.

Different Types of Java Platforms

There are four different types of Java programming language platforms:

1. Java Platform, Standard Edition (Java SE):

Java SE's API offers the Java programming language's core functionality. It defines all the basis of type and object to high-level classes. It is used for networking, security, database access, graphical user interface (GUI) development, and XML parsing.

2. Java Platform, Enterprise Edition (Java EE):

The Java EE platform offers an API and runtime environment for developing and running highly scalable, large-scale, multi-tiered, reliable, and secure network applications.

3. Java Programming Language Platform, Micro Edition (Java ME):

The Java ME platform offers an API and a small-footprint virtual machine running Java programming language applications on small devices, like mobile phones.

4. Java FX:

JavaFX is a platform for developing rich internet applications using a lightweight user-interface API. It user hardware-accelerated graphics and media engines that help Java take advantage of higher-performance clients and a modern look-and-feel and high-level APIs for connecting to networked data sources.

To understand Java programming language, we need to understand some basic concept of how a computer program can run a command and execute the action.

What is Assembly Language?

The computer is an electronic device, and it can only understand electronic signals or binary signals. For example, the 5-volt electronic signal may represent binary number 1, while 0 volts may represent binary number 0. So your PC is continuously bombarded with these signals.

Eight bits of such signals are grouped together to interpret Text, numerical, and symbols.

For example, the # symbol is identified by the computer as 10101010. Similarly, the pattern for adding a function is represented by 10000011.

This is known as 8-bit computing. Current day processor is capable of decoding 64-bit time. But what is the relation of this concept with the programming language JAVA? Let understand these as an example.

Suppose if you want to tell the computer to add two numbers (1+2) represented by some binary numbers (10000011), how are you going to tell this to your computer? Yes, we are going to use assembly language to get our code executed.

> "Assembly Language is the most elementary form of software development languages."

What are Assembler and compiler?

With the advancement in technology, i/o devices were invented. You could directly type your program into the PC using ASSEMBLER. It converts it into the corresponding machine code (110001..) and feeds it to your processor. Coming back to our example addition of (1+2), the assembler will convert this code into machine code and output.

That apart, you will also have to make calls to create Operating System provided functions to display the code's output.

But alone the assembler is not involved in this process; it also requires the compiler to compile the long code into a small chunk of codes. With the advancement in software development languages, this entire assembly code could shrink into just one line print f 1+2 A with the software called COMPILER. It is used to convert your c language code into assembly code. The assembler converts it into corresponding machine code. This machine code will be transmitted to the processor. The most common processor used in PC or Computers are the Intel processor.

Though present-day compilers come bundled with assembler can directly convert your higher language code into machine code.

Now, suppose the Windows operating system runs on this Intel processor, a combination of Operating System plus the processor is called the PLATFORM. The most common platform in the world is Windows, and Intel is called the Wintel Platform. The other popular platforms are

Now, with a change in processor, the assembly instructions will also change. For example:

- Add instruction in Intel may be called ADDITION for AMD
- OR Math ADD for Power PC

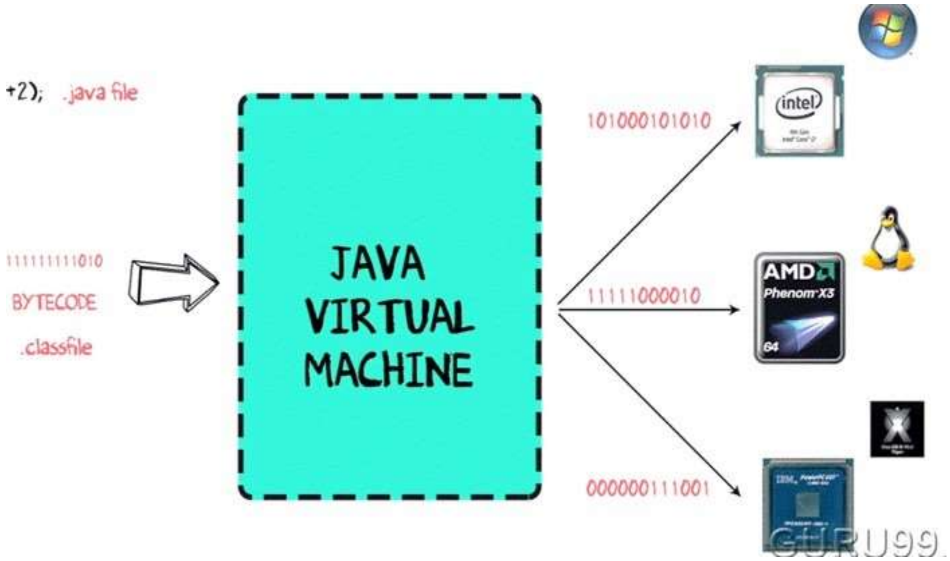
And, with a change in Operating System, OS-level calls' level and nature' will also change.

As a developer, I want my software program to work on all platforms to maximize my revenues. So I would have to buy separate compilers that convert my print f command into the native machine code.

But compilers come expensive, and there is a chance of compatibility issues. So buying and installing a separate compiler for different OS and processor is not feasible. So, what can be an alternative solution? Enter Java language.

How Java Virtual Machine works?

By using Java Virtual Machine, this problem can be solved. But how it works on different processors and O.S. Let's understand this process step by step.



Step 1) The code to display the addition of two numbers is System.out.println(1+2), and saved as a .java file.

Step 2) Using the java compiler the code is converted into an intermediate code called the byte code. The output is a .class file.

Step 3) This code is not understood by any platform, but only a virtual platform called the Java Virtual Machine.

Step 4) This Virtual Machine resides in the RAM of your operating system. When the Virtual Machine is fed with this byte code, it identifies the platform it is working on and converts the byte code into the native machine code.

While working on your PC or browsing the web, whenever you see either of these icons, be assured the java virtual machine is loaded into your RAM. But what makes Java lucrative is that code, once compiled, can run not only on all PC platforms but also on mobiles or other electronic gadgets supporting Java.

Hence, "Java is a programming language as well as a Platform"

How is Java Platform Independent?

Like the C compiler, the Java compiler does not produce native executable code for a particular machine. Instead, Java produces a unique format called bytecode. It executes according to the rules laid out in the virtual machine specification. Therefore, Java is a platform-independent language.

Byte code is understandable to any JVM installed on any OS. In short, the java source code can run on all operating systems.

Summary:

- Java is a multi-platform, object-oriented, and network-centric programming language Java is a general-purpose, class-based, object-oriented programming language.
- Java Platform is a collection of programs that help programmers to develop and run Java applications efficiently.
- Meaning of Java: Java is a multi-platform and network-centric programming language.
- It is mainly used for developing Android Apps and Enterprise Software.
- 2009, Oracle Corporation acquired Sun Microsystems and took ownership of three key Sun software assets: Java, Solaris, and MySQL.
- The latest version of Java released on September 15th, 2020
- The best feature of the Java is that it is one of the easiest programming languages to learn.
- Four types of Java Programming language platforms are: 1) Java Platform, Standard Edition (Java SE) 2) Java Platform, Enterprise Edition (Java EE) 3) Java Platform, Micro Edition (Java ME) 4) JavaFX
- A computer is an electronic device capable of performing computations.
- The computer only understands electronic signals or binary signals.
- Assembler is an advanced technology that converts source core to corresponding machine code (110001..) and feeds to your processor

4.2 Swings

What do you mean by Swings in java?

- An element of The JFC □ Swing Java consists of: oLook and feel o Ease of access o Java 2D o Drag and Drop, etc
- Compiling & executing programs
- 'Javac<program.java>' && 'java<program>'
- If you do not clearly add a GUI constituent to a container, the GUI part will not be put

on view when the container comes into view on the screen

Swing, wing documentation to the AWT, comprise of novel and better mechanism that improve the look and functionality of GUIs. Swing can be deployed to come up with an altogether a Standalone swing Graphical User Interface software or applications and also the Servlets and Applets. It makes use of a model/view design structural design. AWT is not as portable and flexible as Swings are.

Swing Representation/outlook Plan:

Component object and the UI one are being used to put into the practice the "view part" of the MV plan or design. On the other hand, Model object and Change listener help is taken to implement the "model view" of the MV design.

Built on the platform of AWT, Swing is basically written in Java language. The support of an AWT's lightweight component assistance is taken so as to come up with the Swings.

Overall, the design or the architecture of the components or elements of Swings make it much simpler to come up with both the outlook/appearance and behaviour as well. The mixture of AWT and Swing elements or constituents permits it to provide Swing support to the current programs based on AWT.

Example includes, swing mechanism such as JSlider, JButton and JCheckbox could be deployed in the equivalent code with pattern AWT tags, text fields and scrollbars. You could subclass the active or current Swing UI, model, or change listener classes with no need to have rediscovered the whole functioning. Swing also has the skill to put back these matter on-the-tracks.

- 100% Java performance of machinery
- Pluggable Appear & sense
- Lightweight mechanism and procedures
- Bring into play MVC design

Model corresponds to the data

Outlook as a visual demonstration of the data

Controller takes input and decode it to alterations in data

Three components

Constituent set (subclasses of JComponent)

Sustain classes Interface

Classes that symbolize Graphical User Interface elements in Swings have names initiating with the alphabet J. Few of the illustrations include JButton, JLabel, and JSlider. Overall, 250 new classes and 75 interfaces in Swing are there and even more than that and this is double as many as AWT contains.

Ladder for java Swing Class

The class know by the name of JComponent, when moved down straight from Container, is the core class for majority of Swing's user boundary constituents.

Figure 1: Graphical User Interface

Swing consists of elements or constituents that a developer will be making use of to come up with a Graphical User Interface. Let us take a look at some of the most commonly used elements or components of the Swing. The programming know-how of AWT is not at all required to equip with these swing programs.

Illustrations displaying Java Swings

The conventional and the most famous Hello World program is our example for Java Swing as well.

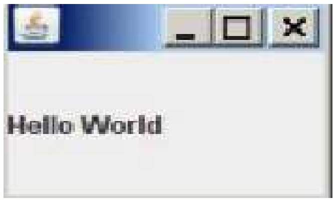
The overall objective to bring this Hello World Program to you all is to teach on how a java program can be created, and then how to compile it and then execute or run the same.

Any Editor such as Text pad/Edit plus or the favourite of all, an IDElike Eclipse can be made use of to come up with a java source code

Listining 1:Full source code

```
import javax.swing.JFrame;
import javax.swing.JLabel;
 //import declaration
 //Verify if the dialog window turns off of its own. Else, introduce
appropriate code
 public class HelloWorldFrame extends JFrame
{
public static void main(String args[])
{
     new HelloWorldFrame();


}
 HelloWorldFrame()
{
       JLabel jlbHelloWorld = new JLabel("Hello World");
      add(jlbHelloWorld);
      this.setSize(100, 100);
      // pack();
      setVisible(true);
   }
}
```

Output Figure



```
Hello World
```

2: Output Hello World

Java Swing Elements

- thereby referring to the existing thread that is jammed till the time interaction with the user has taken place completely.
- JLabel- Talking about the JLabel now, this is used to come up with the text labels. AbstractButton, the abstract provides a sort of platform to a family of elements consisting of buttons, classes as well as the Jbutton.
- JTextField- the elements I popular or rather made use to permit the modification of a single line of text. Some of the new attributes compromise of the skill to prove the text left, right, center etc.
- JPasswordField- Thinking to suppress the input display? Then this is the right element for you to do so. Each of the data or the character entered can be be restored by an echo character. This

actually permits to enter the secret or confidential passwords, example includes the echo character as asterisk, *.

JTextArea- The component is used to edit multiple text lines. It can be used in parallelwith JScrollPane so as to accomplish scrolling which according to the need can

requested to have or even disallow the vertical or horizontal scrollbar.

4.3 Database

What is Data?

In simple words, data can be facts related to any object in consideration. For example, your name, age, height, weight, etc. are some data related to you. A picture, image, file, pdf, etc. can also be considered data.

What is Database?

A database is a systematic collection of data. They support electronic storage and manipulation of data. Databases make data management easy.

Let us discuss a database example: An online telephone directory uses a database to store data of people, phone numbers, and other contact details. Your electricity service provider uses a database to manage billing, client-related issues, handle fault data, etc.

Let us also consider Facebook. It needs to store, manipulate, and present data related to members, their friends, member activities, messages, advertisements, and a lot more. We can

provide a countless number of examples for the usage of databases.

Types of Databases

Here are some popular types of databases.

Distributed databases:

A distributed database is a type of database that has contributions from the common database and information captured by local computers. In this type of database system, the data

is not in one place and is distributed at various organizations.

Relational databases:

This type of database defines database relationships in the form of tables. It is also called Relational DBMS, which is the most popular DBMS type in the market. Database example of the RDBMS system include MySQL, Oracle, and Microsoft SQL Server database.

Object-oriented databases:

This type of computers database supports the storage of all data types. The data is stored in the form of objects. The objects to be held in the database have attributes and methods that define what to do with the data. PostgreSQL is an example of an objectoriented relational DBMS.

Centralized database:

It is a centralized location, and users from different backgrounds can access this data. This type of computers databases store application procedures that help users access the data even from a remote location.

Open-source databases:

This kind of database stored information related to operations. It is mainly used in the field of marketing, employee relations, customer service, of databases.

Cloud databases:

A cloud database is a database which is optimized or built for such a virtualized environment. There are so many advantages of a cloud database, some of which can pay for storage capacity and bandwidth. It also offers scalability on-demand, along with high availability.

Data warehouses:

Data Warehouse is to facilitate a single version of truth for a company for decision making and forecasting. A Data warehouse is an information system that contains historical and commutative data from single or multiple sources. Data Warehouse concept simplifies the reporting and analysis process of the organization.

NoSQL databases:

NoSQL database is used for large sets of distributed data. There are a few big data performance problems that are effectively handled by relational databases. This type of computers database is very efficient in analyzing large-size unstructured data.

Graph databases:

A graph-oriented database uses graph theory to store, map, and query relationships. These kinds of computers databases are mostly used for analyzing interconnections. For example, an organization can use a graph database to mine data about customers from social media.

OLTP databases:

OLTP another database type which able to perform fast query processing and maintaining data integrity in multi-access environments.

Personal database:

A personal database is used to store data stored on personal computers that are smaller and easily manageable. The data is mostly used by the same department of the company and is accessed by a small group of people.

Multimodal database:

The multimodal database is a type of data processing platform that supports multiple data models that define how the certain knowledge and information in a database should be organized and arranged.

Document/JSON database:

In a document-oriented database, the data is kept in document collections, usually using the XML, JSON, BSON formats. One record can store as much data as you want, in any data type (or types) you prefer.

Hierarchical:

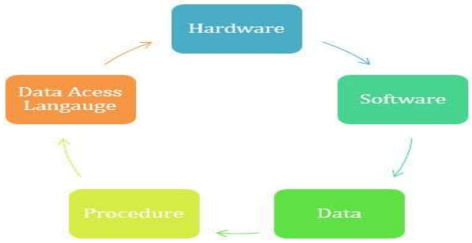
This type of DBMS employs the "parent-child" relationship of storing data. Its structure is like a tree with nodes representing records and branches representing fields.

The windows registry used in Windows XP is a hierarchical database example.

Network DBMS:

This type of DBMS supports many-to-many relations. It usually results in complex database structures. RDM Server is an example of database management system that implements the network model.

Database Components:



There are five main components of a database:

Hardware:

The hardware consists of physical, electronic devices like computers, I/O devices, storage devices, etc. This offers the interface between computers and real-world systems.

Software:

This is a set of programs used to manage and control the overall database. This includes the database software itself, the Operating System, the network software used to share the data among users, and the application programs for accessing data in the database.

Data:

Data is a raw and unorganized fact that is required to be processed to make it meaningful. Data can be simple at the same time unorganized unless it is organized. Generally, data comprises facts, observations, perceptions, numbers, characters, symbols, images, etc.

Procedure:

Procedure are a set of instructions and rules that help you to use the DBMS. It is designing and running the database using documented methods, which allows you to guide the users who operates and manage it.

What does Java Database Connectivity (JDBC) mean?

□ Java Database Connectivity (JDBC) is an application programming interface (API) which allows the programmer to connect and interact with databases. It provides methods to query and update data in the database through update statements like SQL's CREATE, UPDATE, DELETE and INSERT and query statements such as SELECT. Additionally, JDBC can run stored procedures.

□ Like Java, JDBC is compatible with many platforms like Unix and MAC OS.

Techopedia explains Java Database Connectivity (JDBC)

The JDBC API uses Java standard classes and interfaces to connect to databases. In order to use JDBC to connect Java applications to a specific database server, a JDBC driver that supports the JDBC API for that database server is required.

To establish a connection between the Java application and the database, JDBC follows certain steps:

1. Loading the driver: The driver provides a connection to the database.
2. Creating the connection: Once the driver is loaded, the next step is to create a connection. The connection object uses a URL in the specified format, which includes the machine name, port number and database name. It communicates with the database object.
3. Executing SQL statements: Requires an object for building the SQL statement.
4. Returning the resultset: Retrieves and manipulates the database queries. Records can be accessed from the first row to the last row of the database.

Insert query

Syntax

```
INSERT INTO TABLE_NAME (column1, column2, column3,...columnN)
VALUES (value1, value2, value3,...valueN);
```

Example source code

Given that MySQL database table design, let's assume that we just want to insert one record into this MySQL table. To do so, we just need to follow these steps:

1. Create a Java Connection to our example MySQL database.
2. Create a SQL INSERT statement, using the Java PreparedStatement syntax.
3. Set the fields on our Java PreparedStatement object.
4. Execute a Java PreparedStatement.
5. Close our Java MYSQL database connection.
6. Catch any SQL exceptions that may come up during the process.

VIEW Syntax

```
CREATEVIEWview_nameASSELECT column1, column2, ...
FROMtable_nameWHERE condition;
```

DELETE Syntax

```
DELETEFROMtable_nameWHERE condition;
```

CHAPTER 5

PROJECT DESCRIPTION

5.1 PROBLEM DEFINITION:

There is now no doubt that the information revolution is the largest and most innovative that marked the life of mankind in this centuary.Indeed far from being a passing fad or a passing trend,the computer just to bring us many advantages to our way of life.No area was a stranger to this strategy that offers many services both for business and administration.

Today the business is done online worldwide, the management of institutions is done through network technology,all the systems of information management has been digitalized. The problem is to store the details of tenants in paper and payment information according to month writing them manually is more hectic work. It is difficult to maintain records manually.

5.2PROJECT OVERVIEW

Complex administration basically has two main modules for proper functioning

□ Admin module has rights for creating his account and login.

□ User store the details of tenants and payment information of tenants.

5.3MODULE DESCRIPTION

□ The system should be designed in such a way that only authorized people should be allowed to access some particular modules. The records should be modified by only administrators and no one else. The user should always be in control of the application and not the vice versa.

□ The user interface should be consistent so that the user can handle the application with ease and speed. The application should be visually, conceptually clear.

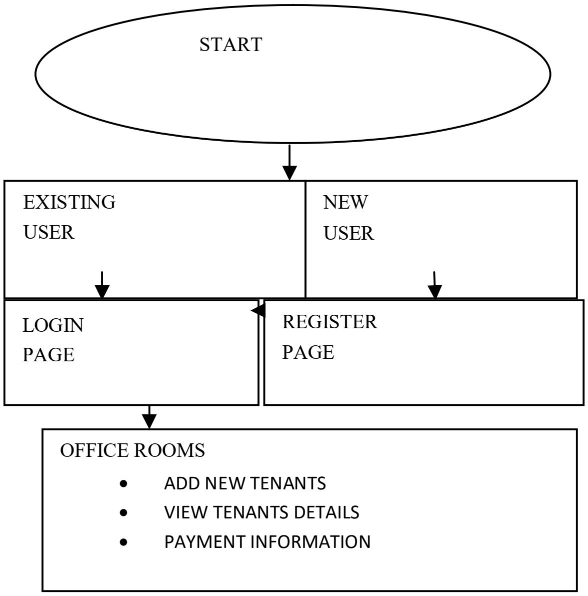
ADMINISTRATOR MODULE:

□ Tenant details:

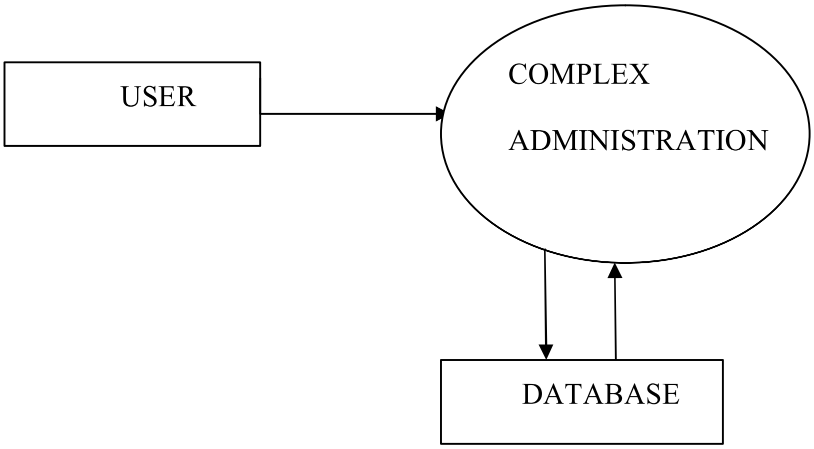
In this module deals with the allocation of personal details of the tenant details like name, adharno, contact no and other details

□ It provides the facility to have a user name and password to the administrator.

5.4 SYSTEMFLOW DIAGRAM:



5.5DATA FLOW DIAGRAM :



5.6 Database Design:

REGISTER TABLE:

- □ To create a login details for the table.

FIELDS	DATA TYPES	DESCRYPTION
USER NAME	VARCHAR(50)	ADMIN USERNAME
PASSWORD	VARCHAR(50)	ADMIN PASSWORD

ADD NEW TENANT TABLE:

- □ To create table to store tenant details.

FIELDS	DATA TYPES	DESCRYPTION
ROOMNUM	INTEGER	ROOM ALLOTED TO TENANTS
NAME	VARCHAR(100)	NAME OF TENANTS
DOB	DATE	DATE OF BIRTH OF TENANTS
CONTACTNUM	VARCHAR(50)	CONTACT NUMBER OF TENANTS
GENDER	VARCHAR(50)	GENDER OF TENANTS
ADDRESS	VARCHAR(50)	NATIVE ADDRESS OF TENANTS
ADHARNUM	VARCHAR(50)	ADHAR NUMBER OF TENANTS
PANNUM	VARCHAR(50)	PANNUMBER OF TENANTS
JOINEDDATE	DATE	JOINEDDATE OF TENANTS
ELECTRICTYNUM	VARCHAR(50)	ELECTRICITY NUMBER OF TENANTS
ADVANCE	VARCHAR(50)	ADVANCE GIVEN BY THE TENANTS

PAYMENT DETAILS OF TENANTS:

- To create table to store payments information of tenants

FIELDS	DATATYPES	DESCRYPTION
NAME	VARCHAR	NAME OF THE TENANT
ROOM NUM	INTEGER	ROOM ALLOTED TO THE TENANT
CONTACT NUM	VARCHAR	CONTACTNO OF THE TENANT
RENT	VARCHAR	AMOUNT PAID BY THE TENANT
MONTH	DATE	DATE WHERE AMOUNT TO BE PAID
PAID	VARCHAR	PAID OR NOT
ADHAR NUM	VARCHAR	ADHARNUM OF THE TENANT

5.6.1 INPUT DESIGN

Input design is part of overall system design that requires special attention designing input data is to make the data entered easy and free from errors. The input forms are designed using the controls available in .NET framework. Validation is made for each and every data that is entered. Help information is provided for the users during when the customer feels difficult.

Input design is the process of converting the user originated inputs to a computer based format. A system user interacting through a workstation must be able to tell the system whether to accept the input to produce reports. The collection of input data is considered to be most expensive part of the system design. Since the input has to be planned in such a manner so as to get relevant information, extreme care is taken to obtain pertinent information

This project first page input is to see whether the admin is new user or existing user, if admin is new user he has to register by giving inputs as username and password, if the admin is existing user then he has to login by giving the inputs like user name and password.

After login the admin can do his work like adding new tenant details ,or storing the payment information of the tenant details

5.6.2 OUTPUT DESIGN

Output design this application "COMPLEX ADMINISTRATION" generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.

The output is designed in such a way that it is attractive, convenient and informative.

Forms are designed with various features, which make the console output more pleasing.

As the outputs are the most important sources of information to the users, better design should improve the system's relationships with us and also will help in decision making. Form design elaborates the way output is presented and the layout available for capturing information.

One of the most important factors of the system is the output it produces. This system refers to the results and information generated. Basically the output from a computer system is used to communicate the result of processing to the user. Complex administration maintains all the information of the tenants. The output produced is, it gives a information of tenants who are currently living in complex and the payment information of the tenants.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Purpose:

System implementation is the important stage of project when the theoretical design is tuned into practical system. The main stages in the implementation are as follows:

□ Planning

□ Training

□ System testing and

□ Changeover Planning

Planning is the first task in the system implementation. At the time of implementation of any system people from different departments and system analysis involve. They are confirmed to practical problem of controlling various activities of people outside their own data processing departments.

The line managers controlled through an implementation coordinating committee. The committee considers ideas, problems and complaints of user department, it must also consider:

□ The implication of system environment

□ Self selection and allocation for implementation tasks

□ Consultation with unions and resources available

□ Standby facilities and channels of communication

□ Student Attendance management system will implement student details ,staff handle subjects details, separate login details ,time table details. It will used to entered subject wise attendance .This application elaborate attendance table generate weekly, consolidate report provide to the End user. Mostly this application will calculate date wise attendance .To select starting date to end date generate reports at the time of activities.

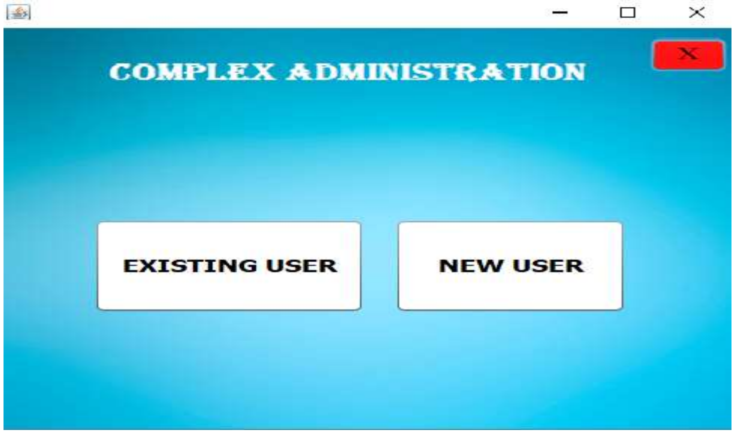
6.2 SYSTEM MAINTENANCE:

Software maintenance is far more than finding mistakes. Provision must be made for environment changes, which may affect either the computer, or other parts of the computer based systems. Such activity is normally called maintenance. It includes both the improvement of the system functions and the corrections of faults, which arise during the operation of a new system.

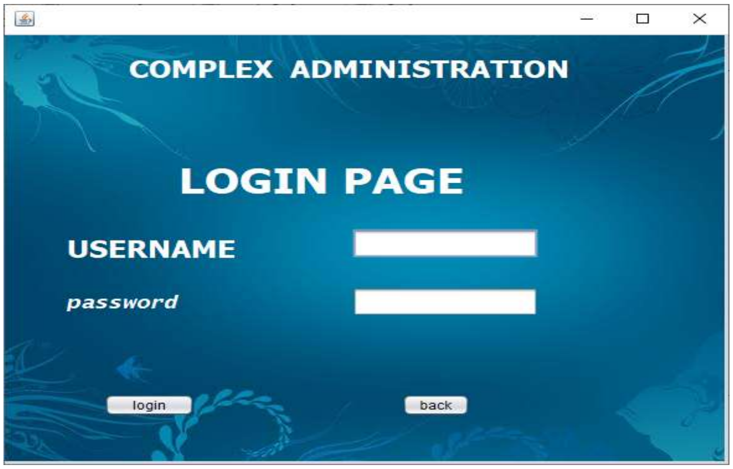
It may involve the continuing involvement of a large proportion of computer department recourses. The main task may be to adapt existing systems in a changing environment.

Back up for the entire database files are taken and stored in storage devices like flash drives, pen drives and disks so that it is possible to restore the system at the earliest. If there is a breakdown or collapse, then the system gives provision to restore database files. Storing data in a separate secondary device leads to an effective and efficient maintains of the system. The nominated person has sufficient knowledge of the organization’s computer passed based system to be able to judge the relevance of each proposed change.

OUTPUT DESIGNS

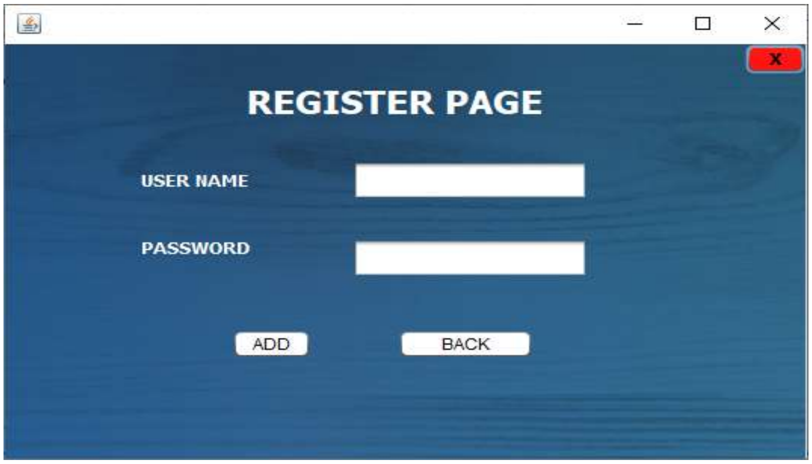


```
private void btncloseActionPerformed(java.awt.event.ActionEvent evt) {

        int a=JOptionPane.showConfirmDialog(null,"Do u really want to exit?","
select",JOptionPane.YES_NO_OPTION);
        if(a==0)
        {
         System.exit(0);
        }
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        login l = new login();
        l.setVisible(true);
       this.dispose();
 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        register r = new register();
        r.setVisible(true);
        this.dispose();
    }
```

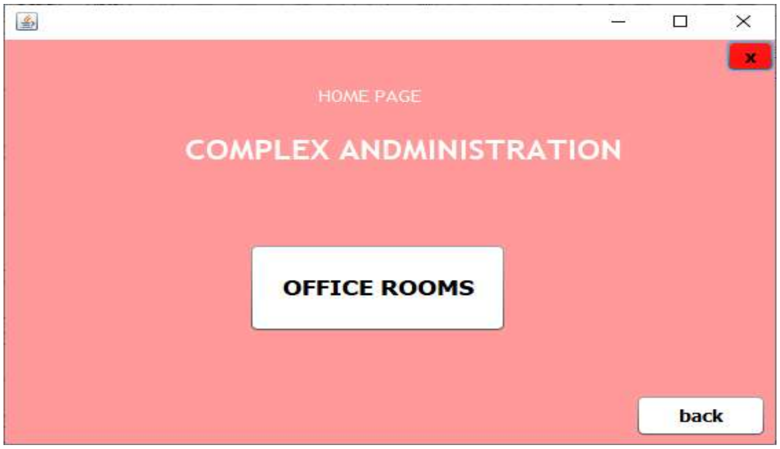


```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        //String user = jTextField1.getText();
        //String pass =  new String(jPasswordField1.getPassword());
        try{
            Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/rachana","app","APP");
            Statement st =con.createStatement();
            ResultSet rs = st.executeQuery("select* from register");
            while(rs.next())
            {

if(rs.getString(1).equals(jTextField1.getText())&&rs.getString(2).equals(jPasswordField1.getText()
))
                {
                    homepage h = new homepage();
                    h.setVisible(true);
                    this.dispose();
                    JOptionPane.showMessageDialog(rootPane,"welcome");
                }
               else
               {
                    JOptionPane.showMessageDialog(rootPane, "invalid password");
               }
            }
        }
        catch(SQLException ex)
        {
      Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, ex);
```

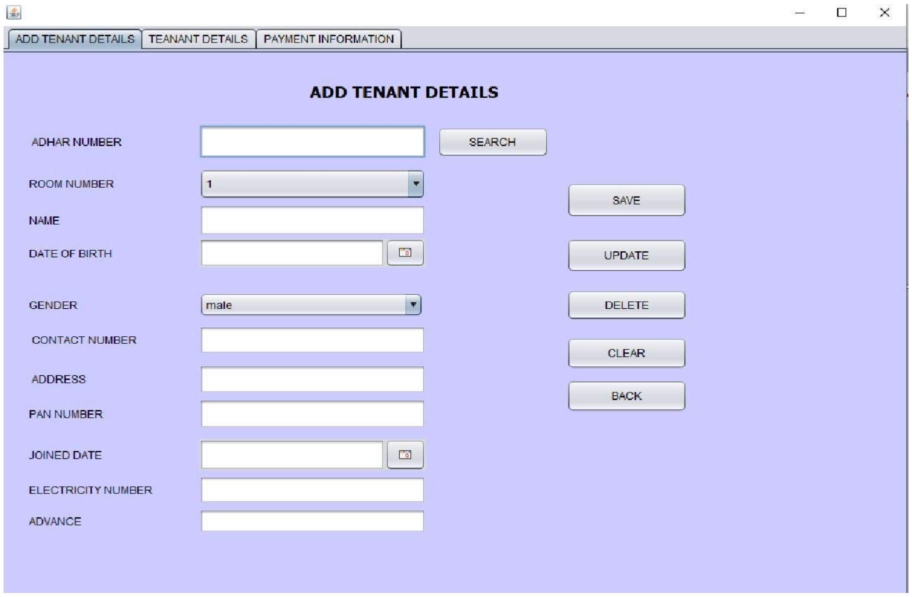
```
        }
}
```



```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

        try {
            // TODO add your handling code here:
            Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/rachana");
            PreparedStatement p = con.prepareStatement("insert into register values(?,?)");
            p.setString(1,regtxtusername.getText() );
            p.setString(2, regtxtpassword.getText());
            p.executeUpdate();

 } catch (SQLException ex) {
            Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, ex);
        }

    }
```



```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        officerooms o = new officerooms();
        o.setVisible(true);
        this.dispose();
```



ADHAR NUMBER

ROOM NUMBER 1

NAME

DATE OF BIRTH

GENDER male

CONTACT NUMBER

ADDRESS

PAN NUMBER

JOINED DATE

ELECTRICITY NUMBER

ADVANCE

PACKAGES

```
package complexadministration;
import java.awt.HeadlessException;
import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import java.util.Date;

import javax.swing.table.DefaultTableModel;
```

FORM CLASS

```
public class officerooms extends javax.swing.JFrame {
    java.util.Date dob;
     java.sql.Date sqldate;
     java.util.Date joineddate;
    java.sql.Date sqldate1;
    java.util.Date month;
    java.sql.Date sqldate2;
```

SAVE BUTTON IN ADD NEW TENANTS

```
private void btnsaveActionPerformed(java.awt.event.ActionEvent evt) {
        int roomnum=Integer.parseInt((String) comborn.getSelectedItem());
        String name=txtname.getText();
        dob=txtdob.getDate();
        sqldate=new java.sql.Date(dob.getTime());
        String gender =(String) combogn.getSelectedItem();
        String contactnum=txtcn.getText();
        String address=txtadd.getText();
        String adharnum=txtan.getText();
        String pannum=txtpn.getText();
        joineddate=txtjd.getDate();
```

```
sqldate1=new java.sql.Date(joineddate.getTime());
 String electricitynum=txten.getText();
       Double advance=Double.parseDouble(txtadvance.getText());
         try
       {
           Connection                                    con                                =
DriverManager.getConnection("jdbc:derby://localhost:1527/rachana","app","APP");
           String                              query="Insert                                into
offaddnewten(ROOMNUM,NAME,DOB,GENDER,CONTACTNUM,ADDRESS,ADHARNUM,P
ANNU
M,JOINEDDATE,ELECTRICITYNUM,ADVANCE)values(?,?,?,?,?,?,?,?,?,?,?)";
    PreparedStatement p;
           p=con.prepareStatement(query);
           java.util.Date d=new java .util.Date();

      SimpleDateFormat df=new SimpleDateFormat("dd-MM-YYYY");

if(comborn.getSelectedItem().equals("")||txtname.getText().equals("")||txtdob.getDate().equals("")||t
xtcn.getText().equals("")||txtadd.getText().equals("")||txtan.getText().equals("")||txtpn.getText().equ
als("")||txtjd.getDate().equals("")||txtadvance.getText().equals(""))
           {
               JOptionPane.showMessageDialog(null,"Fill all the details");
           }
             p.setInt(1,roomnum );//1
           p.setString(2,name );//2
           p.setDate(3,sqldate);//3
           p.setString(4,gender);//4
           p.setString(5,contactnum);//5
           p.setString(6,address );//6

            p.setString(7,adharnum );//7
           p.setString(8,pannum );//8
           p.setDate(9,sqldate1);//9
           p.setString(10,electricitynum);//10
           p.setDouble(11,advance );//11
           p.executeUpdate();
           JOptionPane.showMessageDialog(null," Data inserted sucessfully");

       }
       catch(HeadlessException | SQLException ex)
```

```
            {
               Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, ex);

            JOptionPane.showMessageDialog(this, ex);
            }
 }
```

SEARCH BUTTON IN ADD NEW TENANTS

```
private void btnsearchActionPerformed(java.awt.event.ActionEvent evt) {
      try
      {
         Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/rachana","app","APP");
         Statement st=con.createStatement();
         ResultSet rs=st.executeQuery("select * from offaddnewten WHERE
adharnum='"+txtan.getText()+"'");

 if(txtan.getText().equals(""))
         {
            JOptionPane.showMessageDialog(null,"ENTER ADHARNUMBER");
         }
         else
         {
            if(rs.next())

     {
               comborn.setSelectedItem(rs.getString(1));
               txtname.setText(rs.getString(2));
               txtdob.setDate(rs.getDate(3));
               combogn.setSelectedItem(rs.getString(4));
               txtcn.setText(rs.getString(5));

               txtadd.setText(rs.getString(6));
               //txtan.setText(rs.getString(7));
               txtpn.setText(rs.getString(8));

 txtjd.setDate(rs.getDate(9));
               txten.setText(rs.getString(10));
               txtadvance.setText(rs.getString(11));
            }
            else{
              JOptionPane.showMessageDialog(null,"Does not exist");
            }
  }
      }
      catch(Exception ex)
      {
          Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, ex);
      }
```

UPDATE BUTTON IN ADD NEW TENANT

```
     private void btnupdateActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      int roomnum=Integer.parseInt((String) comborn.getSelectedItem());
      String name=txtname.getText();
      dob=txtdob.getDate();
      sqldate=new java.sql.Date(dob.getTime());

      String gender =(String) combogn.getSelectedItem();
      String contactnum=txtcn.getText();
      String address=txtadd.getText();
      String adharnum=txtan.getText();
      String pannum=txtpn.getText();
      joineddate=txtjd.getDate();
      sqldate1=new java.sql.Date(joineddate.getTime());
      String electricitynum=txten.getText();
      Double advance=Double.parseDouble(txtadvance.getText());
     try

  {
          Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/rachana","app","APP");
          Statement st=con.createStatement();
          PreparedStatement p = con.prepareStatement("update offaddnewten set
roomnum=?,name=?,dob=?,gender=?,contactnum=?,"

  + "address=?,pannum=?,joineddate=?,electricitynum=?,advance=? where adharnum=?");

          java.util.Date d=new java .util.Date();
          SimpleDateFormat df=new SimpleDateFormat("dd-MM-YYYY");
```

```
            p.setInt(1, roomnum);//1
            p.setString(2,name );//2
            p.setDate(3,sqldate);//3
            p.setString(4,gender);//4
            p.setString(5,contactnum);//5

            p.setString(6,address );//6
            p.setString(11,adharnum );//7
            p.setString(7,pannum );//8
            p.setDate(8,sqldate1 );//9
            p.setString(9,electricitynum);//10
            p.setDouble(10,advance );//11
            p.executeUpdate();
            JOptionPane.showMessageDialog(null,"sucessfully Updated");
        }

catch(SQLException ex)
                {
                    Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, ex);
                }

}
```

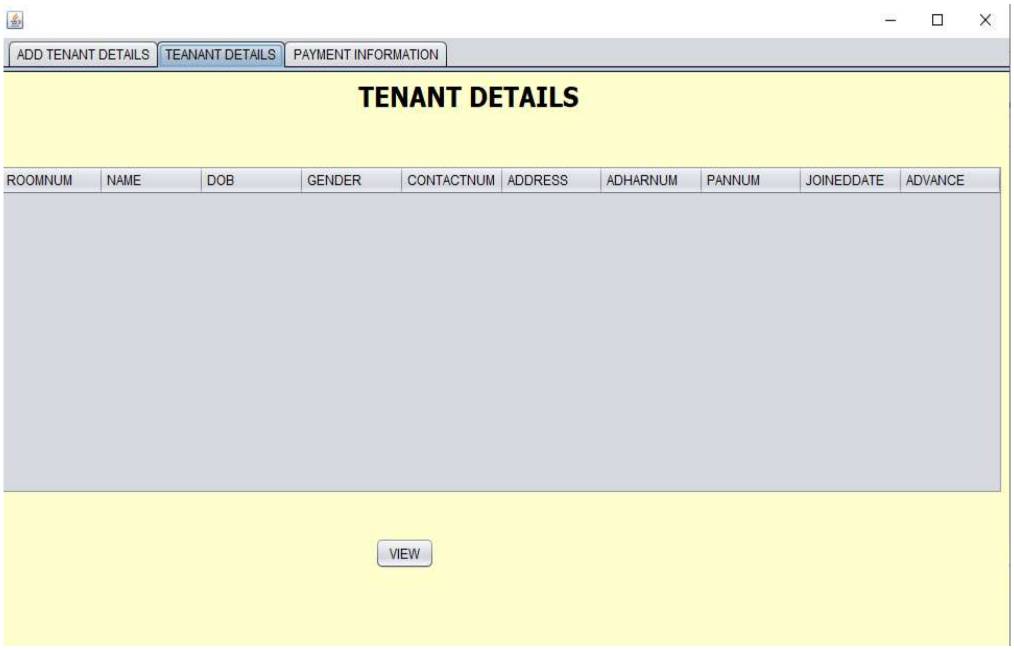
DELETE BUTTON IN ADD NEW TENANT

```
private void btndeleteActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String adharnum=txtan.getText();
        try
        {
            Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/rachana","app","APP");
            PreparedStatement p = con.prepareStatement("delete from offaddnewten where
adharnum='"+adharnum+"'");
            p.executeUpdate();

           JOptionPane.showMessageDialog(null,"sucessfuly deleted");
        }
        catch(SQLException ex)
        {
            Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, ex);
        }
```

```
 }
```

CLEAR BUTTON IN ADD NEW TENANT

```
private void btnclearActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Date date=null;
        comborn.setSelectedItem("");

  txtname.setText("");
        txtdob.setDate(date);
        combogn.setSelectedItem("");
        txtcn.setText("");
        txtadd.setText("");
        txtan.setText("");
        txtpn.setText("");
        txtjd.setDate(date);
        txten.setText("");
        txtadvance.setText("");
 }
```

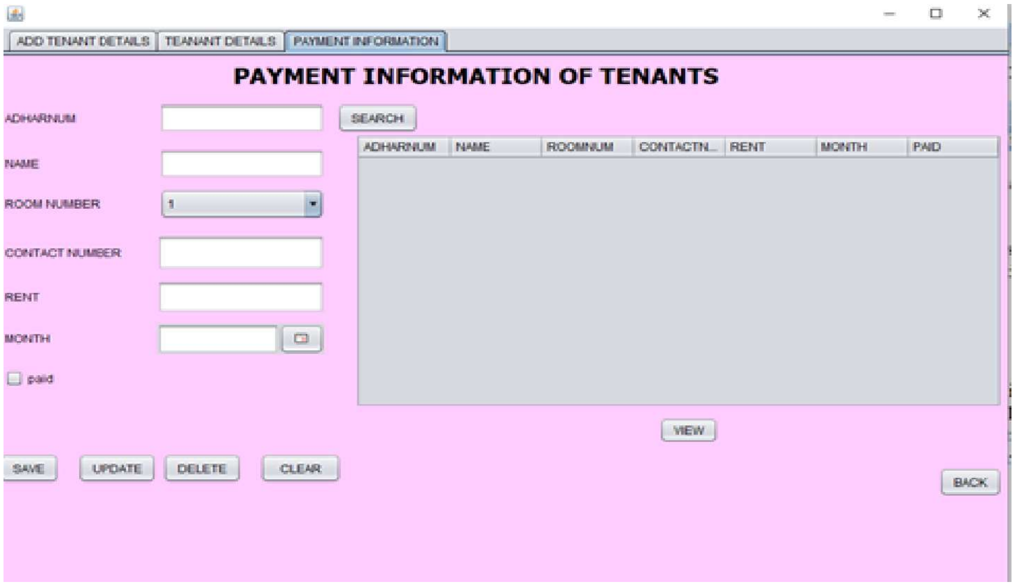


TENANT DETAILS

ROOMNUM	NAME	DOB	GENDER	CONTACTNUM	ADDRESS	ADHARNUM	PANNUM	JOINEDDATE	ADVANCE

TENANT DETAILS (To show in table)

```
private void btntendetActionPerformed(java.awt.event.ActionEvent evt) {

  try
      {
          Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/rachana","app","APP");
          DefaultTableModel model =new DefaultTableModel(new
String[]{"roomnum","name","dob","gender","contactnum","address","adharnum","pannum","joine
ddate","electricitynum","advance"},0);
          String sql="select * from offaddnewten";
          Statement st= con.createStatement();
          ResultSet rs=st.executeQuery(sql);
          while (rs.next())
          {
              String r =rs.getString("roomnum");

       String n =rs.getString("name");

             String d =rs.getString("dob");
              String g=rs.getString("gender");
              String c =rs.getString("contactnum");
              String add =rs.getString("address");
              String adh =rs.getString("adharnum");
              String pan =rs.getString("pannum");
              String jd =rs.getString("joineddate");

  String en=rs.getString("electricitynum");
              String a =rs.getString("advance");
              model.addRow(new Object[]{r,n,d,g,c,add,adh,pan,jd,en,a});
          }
          tendet.setModel(model);
      }
      catch(Exception ex)
      {
          ex.printStackTrace();
          System.out.println(ex);
}
```



SAVE BUTTON IN PAYMENT INFO

```
private void btnsavepayActionPerformed(java.awt.event.ActionEvent evt) {
        String adharnum=txtadh.getText();
        String name=txtn.getText();
        String roomnum=(String) comboroom.getSelectedItem();
        String contactnum=txtcno.getText();
        String rent=txtrnt.getText();
        month=txtmonth.getDate();
        sqldate2=new java.sql.Date(month.getTime());

    String paid;
        if(checkpaid.isSelected())
        {
            paid="yes";
        }
        else

 {
            paid="no";
        }
        java.util.Date d=new java .util.Date();
        SimpleDateFormat df=new SimpleDateFormat("dd-MM-YYYY");
        try
        {
            Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/rachana","app","APP");

        String query="insert into
offpay(NAME,ROOMNUM,CONTACTNUM,RENT,MONTH,PAID,ADHARNUM)values(?,?,?,?
,?,?,?)";
            PreparedStatement p;
            p=con.prepareStatement(query);
            p.setString(1,name);
            p.setString(2,roomnum);
            p.setString(3,contactnum);
            p.setString(4,rent);

 p.setDate(5,sqldate2);
            p.setString(6,paid);
            p.setString(7, adharnum);
            p.executeUpdate();
            JOptionPane.showMessageDialog(null," Data inserted sucessfully");
            }
        catch(Exception ex)
               {

 Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, ex);
               }
}
```

UPDATE BUTTON IN PAYMENT INFO

```
private void btnuppayActionPerformed(java.awt.event.ActionEvent evt) {
       String adharnum=txtadh.getText();
       String name=txtn.getText();

  String roomnum =(String) comboroom.getSelectedItem();
       String contactnum=txtcno.getText();
       String rent=txtrnt.getText();
       month=txtmonth.getDate();
       sqldate2=new java.sql.Date(month.getTime());
       String paid=checkpaid.getText();
      try

    {
            Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/rachana","app","APP");
            Statement st=con.createStatement();
            PreparedStatement p = con.prepareStatement("update offpay set
name=?,roomnum=?,contactnum=?,rent=?,month=?,paid=?WHERE adharnum=?");

            java.util.Date d=new java .util.Date();
            SimpleDateFormat df=new SimpleDateFormat("dd-MM-YYYY");

            p.setString(7,adharnum);
            p.setString(1,name);

           p.setString(2,roomnum);
            p.setString(3,contactnum);
            p.setString(4,rent);
            p.setDate(5,sqldate2);
            p.setString(6,paid);
            p.executeUpdate();

   JOptionPane.showMessageDialog(null,"sucessfully Updated");
        }
        catch(SQLException ex)
                {
                    Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, ex);
                }

}
```

DELETE BUTTON IN PAYMENT INFO

```
private void btndelpayActionPerformed(java.awt.event.ActionEvent evt) {
        try
        {
            Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/rachana","app","APP");
            PreparedStatement p = con.prepareStatement("delete from offpay where
adharnum='"+txtadh.getText()+"'");
            p.executeUpdate();


JOptionPane.showMessageDialog(null,"sucessfuly deleted");
}
        catch(SQLException ex)
        {
            Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, ex);
        }
}
```

SEARCH BUTTON IN PAYMENT INFO

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
         try
        {

Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/rachana","app","APP");
            Statement st=con.createStatement();
            ResultSet rs=st.executeQuery("select * from offpay WHERE
adharnum='"+txtadh.getText()+"'");
            if(rs.next())
            {
                comboroom.setSelectedItem(rs.getString(2));
                txtn.setText(rs.getString(1));
                txtcno.setText(rs.getString(3));
                txtrnt.setText(rs.getString(4));
                txtmonth.setDate(rs.getDate(5));

    if (rs.getString(6).equals("yes"))
                {
                    checkpaid.setSelected(true);

          }
                 txtadh.setText(rs.getString(7));
            }
            else
            {

  JOptionPane.showMessageDialog(null,"Does not exist");
            }
        }
        catch(Exception ex)
        {
```

```
        Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
```

CLEAR BUTTON IN PAYMENT INFO

```
  private void btnclearpayActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Date date=null;
        comboroom.setSelectedItem("");

    txtcno.setText("");
        txtadh.setText("");
        txtn.setText("");
        txtrnt.setText("");
        txtadh.setText("");



        txtmonth.setDate(date);
        checkpaid.setSelected(false);
        }
```

VIEW BUTTON IN PAYMENT INFO

```
private void paytabActionPerformed(java.awt.event.ActionEvent evt) {
        try
        {

Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/rachana","app","APP");
            DefaultTableModel m =new DefaultTableModel(new
String[]{"name","roomnum","contactnum","rent","month","paid","adharnum"},0);
            String sql ="select * from offpay";
            Statement st=con.createStatement();
            ResultSet rs=st.executeQuery(sql);
            while(rs.next())
            {
                String n =rs.getString("name");
                String r =rs.getString("roomnum");

               String rnt =rs.getString("contactnum");
                String cn=rs.getString("rent");
```

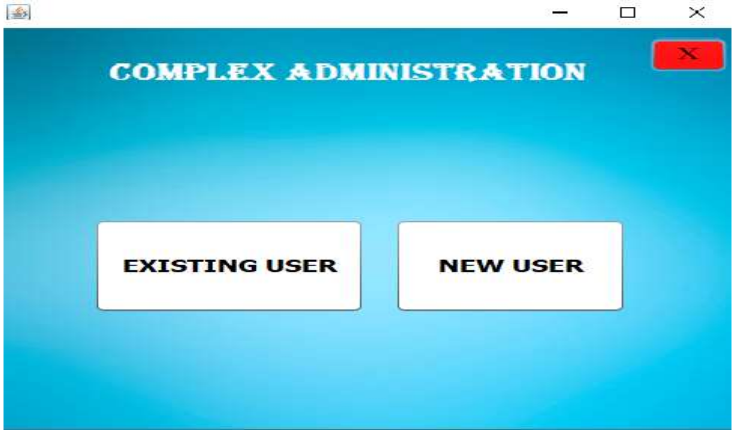
```
             String mn =rs.getString("month");
             String p =rs.getString("paid");
             String a=rs.getString("adharnum");

             m.addRow(new Object[]{n,r,rnt,cn,mn,p,a});
             }
          txtpd.setModel(m);

 }
       catch(Exception ex)
       {
          ex.printStackTrace();
          System.out.println(ex);
          System.out.println("StudentRegistration.BtnInsertActionPerformed()");
       }
   }
```

BACK BUTTON

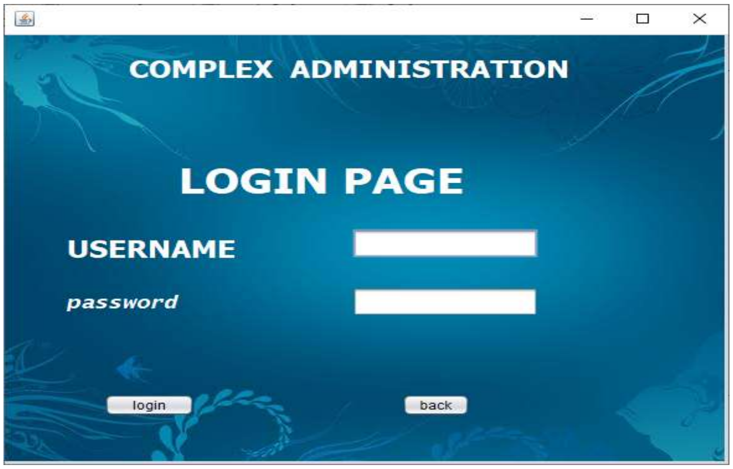
```
private void btnbackpayActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
         homepage h =new homepage();
          h.setVisible(true);

  this.dispose();

    }
```

OUTPUT SCREEN

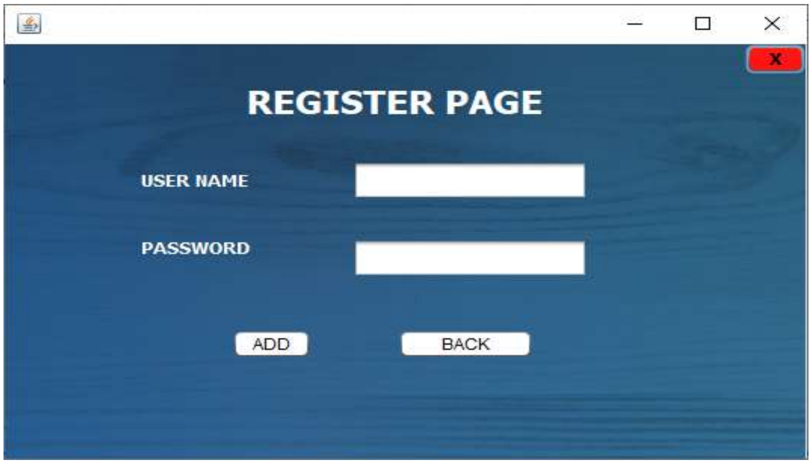
FIRST PAGE



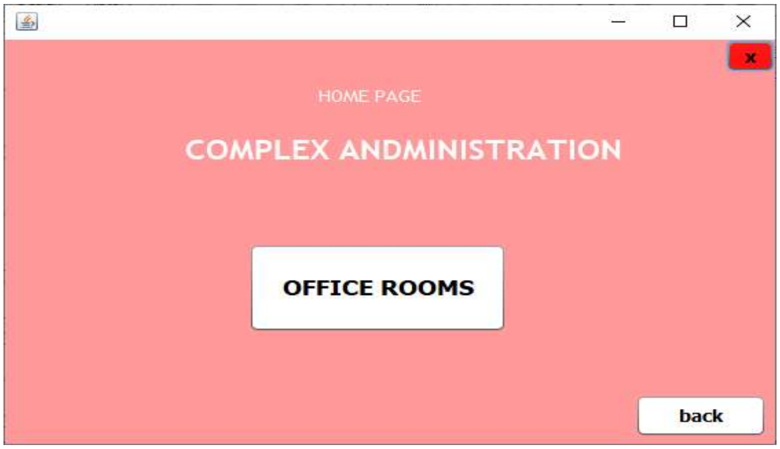
LOGIN PAGE



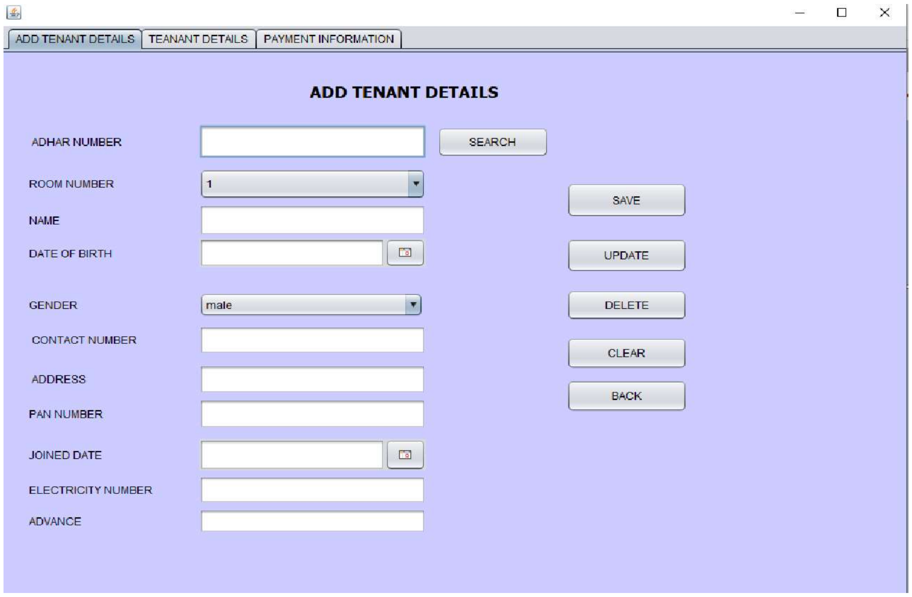
REGISTER PAGE



HOMEPAGE



ADD NEW TENANT DETAILS



ADD TENANT DETAILS

ADHAR NUMBER

ROOM NUMBER: 1

NAME

DATE OF BIRTH

GENDER: male

CONTACT NUMBER

ADDRESS

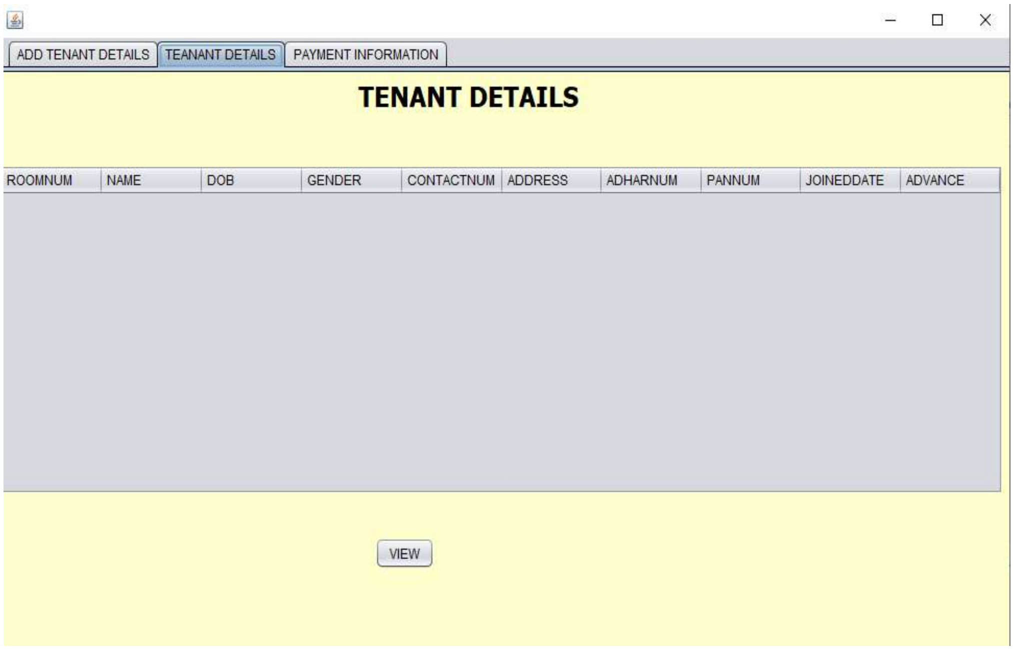
PAN NUMBER

JOINED DATE

ELECTRICITY NUMBER

ADVANCE

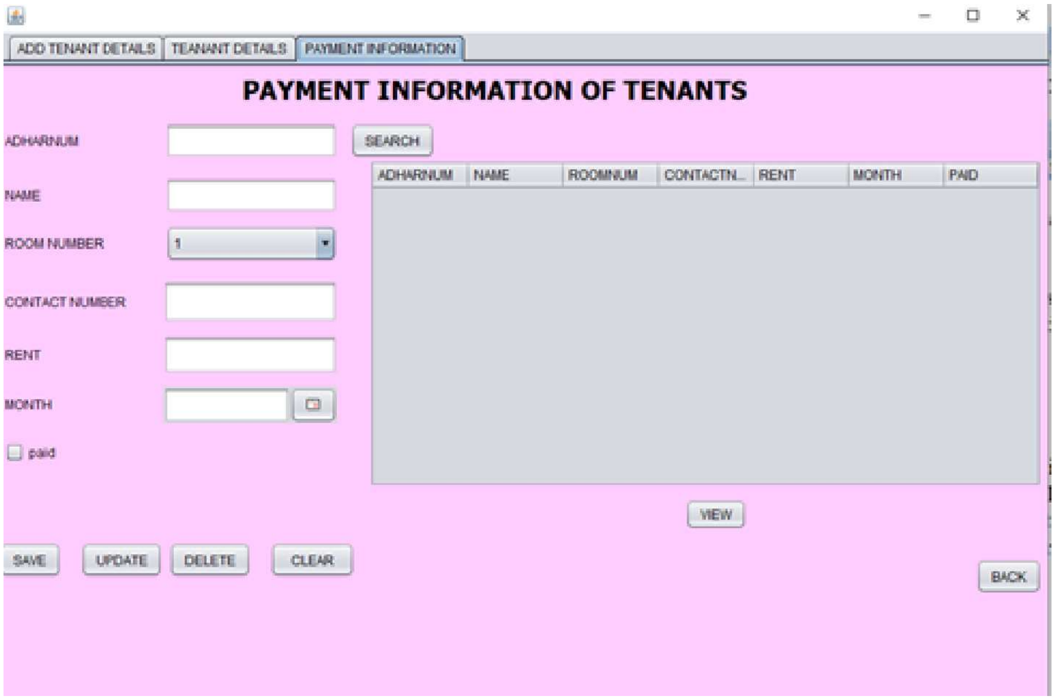
TENANT DETAILS



TENANT DETAILS

ROOMNUM	NAME	DOB	GENDER	CONTACTNUM	ADDRESS	ADHARNUM	PANNUM	JOINEDDATE	ADVANCE

PAYMENT INFORMATION



CHAPTER 7

SYSTEM TESTING

7.1 Introduction

Once source code has been generated, software must be tested to uncover (and correct) as many errors as possible before delivery to customer. Our goal is to design a series of test cases that have a high likelihood of finding errors. To uncover the errors software techniques are used. These techniques provide systematic guidance for designing test that

(1) Exercise the internal logic of software components, and

(2) Exercise the input and output domains of the program to uncover errors In program function, behavior and performance.

Steps: Software is tested from two different perspectives:

(1) Internal program logic is exercised using ―White box test case design Techniques

(2) Software requirements are exercised using ―block box test case design techniques.

In both cases, the intent is to find the maximum number of errors with the Minimum amount of effort and time.

7.2 Testing Methodologies:

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as highlevel tests that validate major system functions against customer requirements. A strategy must provide guidance for the practitioner and a set of milestones for the manager. Because the steps of the test strategy occur at a time when deadline pressure begins to rise, progress must be measurable and problems must surface as early as possible. Following testing techniques are well known and the same strategy is adopted during this project testing.

7.2.1 Unit Testing:

Unit testing focuses verification effort on the smallest unit of software design- the software component or module. The unit test is white-box oriented. The unit testing implemented in every module of student attendance management System. by giving correct manual input to the system ,the data are stored in database and retrieved. If you want required module to access input or get the output from the End user. any error will accrued the time will provide handler to show what type of error will accrued .

7.2.2 System Testing:

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Below we have described the two types of testing which have been take place for this project. it is to check all modules worked on input basis .if you want change any values or inputs will change all information. so specified input is must.

7.2.3 Performance Testing:

Performance testing is designed to test the run-time performance of software within the context of an integrated system. Performance testing occurs throughout all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as white-box tests are conducted.

This project reduce attendance table, codes. it will generate report fast.no have extra time or waiting of results .entered correct data will show result few millisecond. just used only low memory of our system. Automatically do not getting access at another software. Get user permission and access to other applications.

7.3 Test cases

Test case is an object for execution for other modules in the architecture does not represent any interaction by itself. A test case is a set of sequential steps to execute a test operating on a set of predefined inputs to produce certain expected outputs. There are two types of test cases:-manual and automated. A manual test case is executed manually while an automated test case is executed using automation.

In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same.

Ideally, test cases that check error conditions are written separately from the functional test cases and should have steps to verify the error messages and logs. Realistically, if functional test cases are not yet written, it is ok for testers to check for error conditions when performing normal functional test cases. It should be clear which test data, if any is expected to trigger errors.

CONCLUSION

To conclude, Project Data Grid works like a component which can access all the databases and picks up different functions. It overcomes the many limitations incorporated in the attendance.

Easy implementation Environment

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner.

REFERENCES

Programming with java - A PRIMER by E.BalaguruSwamy.

DataBase Book name – Elmasri and Navathe :Fundementals of Database systems ,5th Edtion, Pearson Education,McGraw-Hill.

Software engineering - Ian Sommerville – 9th Edition

https://www.youtube.com/channel/UCA1UW__WEiBd1bmzyKlBUyw