Project Report

On

Online Pharmacy Management System

In

JAVA and MYSQL

Abstract

Our project title is ‘Online Pharmacy Management System’. Here we are discussing about medicine and medical products. Nowadays people get uncomfortable to go on a crowded place. In the present time, it's very difficult to buy medicine when you are sick and not able to go outside to buy medicine. It’s an online-based application where you can buy medicine through online. In this website, patients can search all type medicine. This web applications developed based on the requirements of the client. People nowadays are used to buy products online. It saves time save time and human efforts, our web-based application can be helpful where we have some manual procedure. This web application provides a friendly interface to the customers. Actually this system is developed to reduce the complexity of human life. Anyone can order their product easily. It makes a person's life easier.

Acknowledgment

First we express our heartiest thanks and gratitude to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We would like to express our heartiest gratitude to Professor, and Head**,** Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parent

Table of Contents

Chapter-1. Introduction

1.1 Introduction to Organization

The main aim of the project is the management of the database of the pharmaceutical shop. This is done by creating a database of the available medicines in the shop. The database is then connected to the main program by using interconnection of the Visual Basic program and the database already created.

The purpose of this Pharmacy Management System project is to improve the maintenance and manipulation of the drugs in the medicals. The pharmacy management system will be used to minimize the time and resource by maintaining the details of the drug systemically so that the data can be used in possible quickest time. While the resource which is minimized are workforce, money, papers, etc. The system is user-friendly and will help the pharmacist. This Pharmacy Management System will reduce the burden on pharmacist and will make the system efficient by providing the more accurate details about drugs in the medical.

1.2 Project Description

The web-based pharmacy system that is designed to improve accuracy and to enhance safety and efficiently pharmaceutical store. It helps to customer for time saving and less costing .It helps patients to manage drugs easily by using online order approach the system will contain an interactive user interface so that the user can easily access all the features of the system. The traditional pharmacy is take more time consuming and traditional paper based system. On this system the user always find the medicine which has manufacture date and expiry date correctly. Different pharmacies and recognized that most of them they recording their data manually like registered books. This type recording the take more time of consumer and very incorrect because it can be miss the name of medicine which one fined the customer.

1.3 Objective

Here are some objective that we are looking for.

1. Comprehensively low price than other manual shop and less time to find the medicine

2. Accurately dispense medications

3. Appropriately utilize pharmaceutical and pharmacokinetics mathematics to perform accurate medication calculations.

4. To identify health care-related factors associated with online pharmacy use by considering patients.

1.4 Problem Statement

Managing a very large pharmacy with records on papers will be tedious and difficult to keep track of inventories with regards to the drugs in the store, expiry date, quantity of drugs available based on the categories and their functions.

1.5 Proposed System

The design of the pharmacy management system is based on the computer which will simplify the maintenance of the information, accessible and efficient. The Pharmacy Management System will provide the information about the end of the drugs in the medical so that the physician can order them drugs before the end. The pharmacist and nurses will get more accurate results at the time sell, about the details of the use of medicines and the dosages so that the system will become more reliable to use than the present system. The records of each work will be secure as to access the information the user must have to provide the ID and password in the system.

1.6 Existing System

Currently, the medical works are based on the manual process, and each work is maintained in the paper. The details of purchasing drugs, audits, sell reports maintained on the paper while anyone can enter into the system and can make changes in these reports, so it is not a safe method to keep the information on the paper. The pharmacist faces problem in searching the products from the self as it is not an easy method to remember about the place of each medicine. There no Pharmacy Management System which can alert the pharmacist about the end of the drugs.

Chapter-2. Requirement, Analysis and System Specification

2.1 Feasibility Study

The feasibility study of any system is mainly intended to study and analyze the proposed system and to decide whether the system under consideration will be viable or not after implementation. That is it determines the usability of the project after deployment. To come to a result, a set of queries is answered keeping the efficiency of the software and its impact on the domain for which it was developed. It mainly emphasis on the following three questions elucidated below as:

- What are the user's requirements and how does a candidate system meet them?
- What resources are available for the proposed systems? Is it worth solving the problem?
- What is the likely impact of the proposed system on the organization? i.e. how does the proposed system fit within the organization?
- Operational feasibility
- Technical feasibility
- Economical feasibility

TECHNICAL FEASIBILITY:

- In technical feasibility, we study all technical issues regarding the proposed system. It is mainly concerned with the specifications of the equipment and the software, which successfully satisfies the end-user's requirement. The technical needs of the system may vary accordingly but include:

- The feasibility to produce outputs in a given time.
- Response time under certain conditions.
- Facility to communicate data.
- Under this analysis process, questions like :
- Does the compatible platform exist within our domain or can we procure it?
- Would the recommended technology guarantee the reliability, accuracy and data security?
- Whether the current technology is sufficient for the development of the system?
- Whether the identified user needs may be satisfied using current software and hardware technologies.

Outcome - For this feasibility study, a thorough study of the complete functionality to be provided in the system, as described in the System Requirement Specification (SRS) was done, ensuring if everything was possible using different types of frontend and backend platforms. Existing technology supports the application completely.

B. ECONOMIC FEASIBILITY:

This is a very important aspect to be considered while developing a project. This project is based on the minimum possible cost factor.

- All hardware and software costs have to be borne by the organization.

- The proposed System is computer based. It requires average computing capabilities and access to internet, which are very basic requirements and can be afforded by any organization hence it doesn’t incur additional economic overheads, which renders the system economically feasible

OPERATIONAL FEASIBILITY:

The system will be used by users if it is developed well who undermine the possible application benefits. Its purpose is to find out whether the application will be functional after its development and installation?

- **Clients Supports:** Client and user support for the proposed system is there. The proposed system is fully GUI based and is very user friendly and all inputs to be taken are all self-explanatory even to a layman. No major training and new skills are required. Besides, a user guide has been attached in the applications to let the users know the essence of the system so that they feel comfortable with the new system. As far as my study is concerned, the clients are comfortable and happy as the system has cut down their loads and doing.

- **User Support:** User involvement in the building of the present system is sought to keep in mind the user specific requirement and needs. Users will have control over their own information. Important information such as Test results can be generated at the click of a button.

2.2 Software Requirement Specification

2.2.1 Functional Requirements

There are functions done by the system such as: store the necessary information of medicine,

prepare bill for the medicine, easily searching of medicine, Update, delete and save data of medicine.

Store the necessary information of the drugs:

The PMS (pharmacy management system) stores the detail information about each medicine including Actual name, formula of medicine and how it is importance and for which diseases is required. Since the information for each drug were required in some cases like the use of drugs, when use drugs and for whom is given.

Searching Medicine and other Data’s:

The PS has easily searching of medicine which shows in which shelf is put and the behavior of the medicine. The searching process is based on the name of the give data or the identification of the item. Here when the user searches the item on search bar the related things were displayed in the screen and can select the actual item that the user needs. [2]

Alerting pharmacy Data’s in the system:

Changing medicines to another because of medicine outdated, modifying the saved medicine data for incorrect data, deleting of data of the pharmacy can be done on the system.

2.2.2 Non-Functional Requirements

This pharmacy management system is able to operate in the following characteristics.

Usability: Any familiar in using windows operation can operate the system since it has user friendly user interface. Which have the instruction menu’s how to use it which self-directive application then can be used the system without ambiguity.

Reliability: The pharmacy system is available based on the user needs, can work properly, and do transactions efficiently including safe data management of the pharmacy. For invalid and malfunctioned operation, the system will restart in order to prevent data loose as well as safe operation within 5 seconds. The pharmacy system is password protected to change things on the system. Here the pharmacist manager control over the system by login to the pharmacy system. Any user can’t use the system but the guest user can see on general properties of the pharmacy and medicines without password. As result data is protected and controlled by only the administrator. [4]

Performance: The pharmacy management system operates its function in small amount of time which is less than two seconds and can be accessed by one user at a time or concurrently. To access the user must first login to the system which must have the pharmacy system privileged and also the system can store data up to 40 GB data. When the system may be busy due to malfunction operation it may wait up to one minute other ways the pharmacy system restarts. [4]

User interface: the user interface is friendly which is easy to use. And having attractive frame structure which is prepared in assumption with other related systems. Also the user can change him/her user favorite interfaces that Is available in the system. [4]

Operation: the pharmacy management system is operated and controlled by the pharmacy manager for safe work. [4]

Supportability: This pharmacy management system operates in any version of windows operating system. Such as windows xp, windows 2003, windows 7, windows 8 and other related versions. The system can be easily maintained by the manager of the pharmacy system by using the prepared documents of the system for easy maintenance. Other ways it is maintained by the system developers for corrective and other heavy problems.

Implementation: The system is implemented in Intel(R) Core(TM) i5 processor with 4 GB RAM,64-bit computer. And it is implemented through testing on both Black and White testing. The language we use implement the system is NetBeans IDE, JDK, MYSQL database.

Chapter-3. System Design

3.1 Design Approach

System Design is the solution to the creation of a new system. This is the important aspect made up of several steps. The complete, efficient and successful system should provide the following in succession:

- From where should we start
- Where we have to go
- Where should we stop

If the project is to be successful, we will need to answer this question. The answer of these questions is schema manner and is known as system design.

A systematic manner will be followed so as to achieve beneficial results at the end. It involves starting with a vague idea and ultimately developing it up into a useful system. The design phase is transition from a user oriented to a document oriented to the programmers.

The steps for the successful project are as follows:

We should define the problem completely and the goals should be known before our destination. In the next step, we should specify inputs and outputs of our interest.

Then the structure of various databases should be designed which will be used during the programming. Next, we should design our programs of user-friendly nature and always provide a way for the user to read back the origin if he/she finds any complex problem at any stage. We should know the function of each and every program which will lead us to or helps us to read at the specified goal.

Then we write these individual programs which later on joining solve our problem. Next step involves the testing of these programs and correction – if necessary.

The design approach used in this project is **Object Oriented**. In the object-oriented design approach, the system is viewed as a collection of objects (i.e., entities). The state is decentralized among the objects and each object manages its own state information. For example, in a Library

Automation Software, each library member may be a separate object with its own data and functions to operate on these data. In fact, the functions defined for one object cannot refer to or change data of other objects. Objects have their own internal data which define their state. Similar objects constitute a class. In other words, each object is a member of some class. Classes may inherit features from super classes. Conceptually, objects communicate by message passing.

3.2 Flowchart

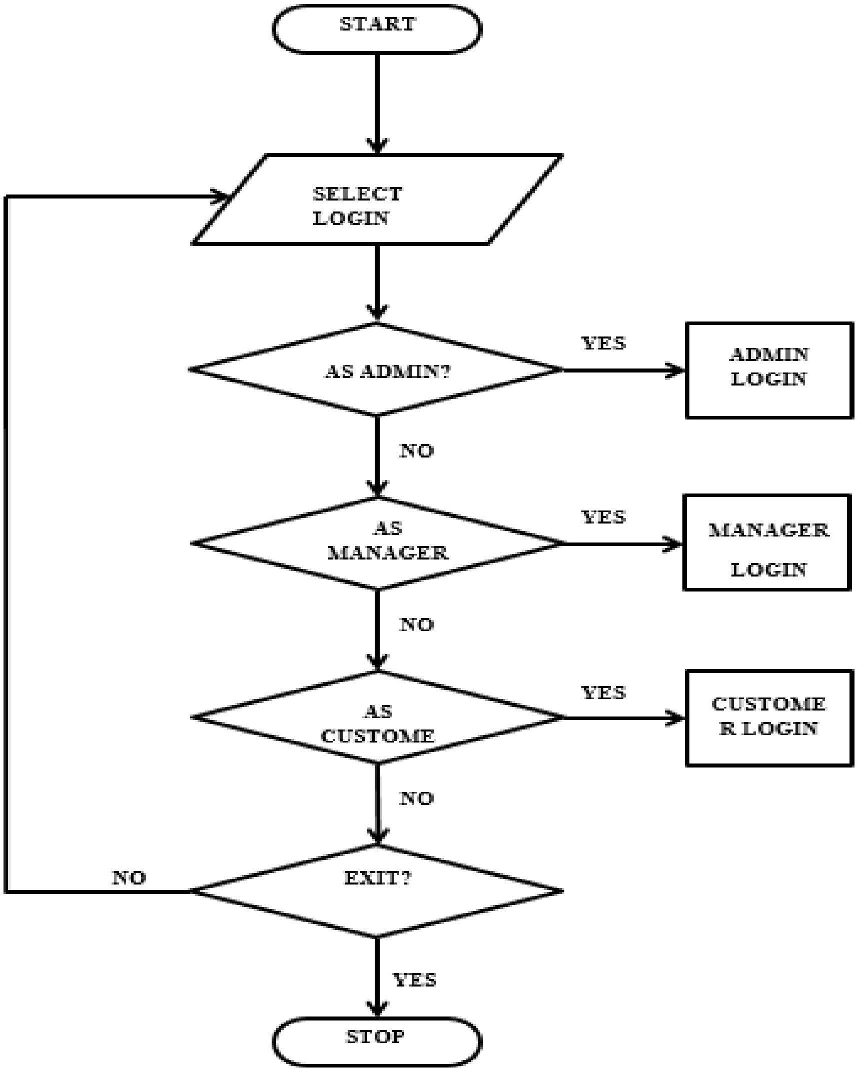
A **flowchart** is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analysing, designing, documenting or managing a process or program in various fields.

Flowcharts are used in designing and documenting simple processes or programs. Like other types of diagrams, they help visualize what is going on and thereby help understand a process, and perhaps also find less-obvious features within the process, like flaws and bottlenecks. There are different types of flowcharts: each type has its own set of boxes and notations. The two most common types of boxes in a flowchart are:

- a processing step, usually called activity, and denoted as a rectangular box.
- a decision, usually denoted as a diamond.

A flowchart is described as "cross-functional" when the chart is divided into different vertical or horizontal parts, to describe the control of different organizational units. A symbol appearing in a particular part is within the control of that organizational unit. A cross-functional flowchart allows the author to correctly locate the responsibility for performing an action or making a decision, and to show the responsibility of each organizational unit for different parts of a single process.



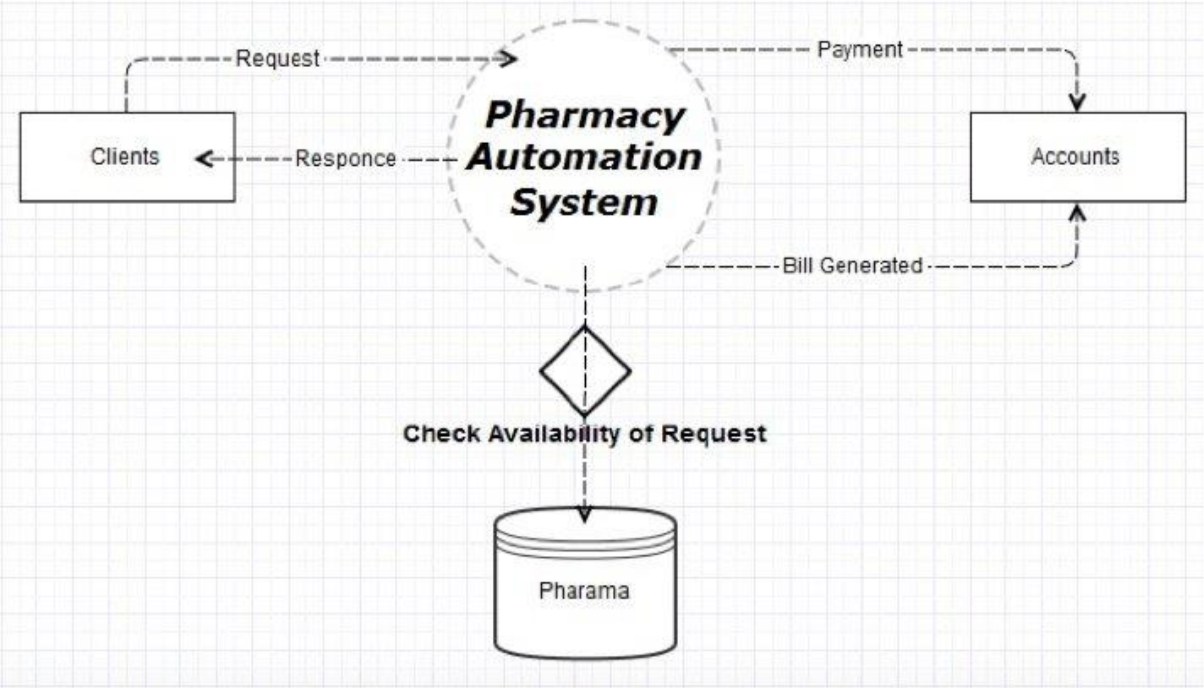
3.3 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole.

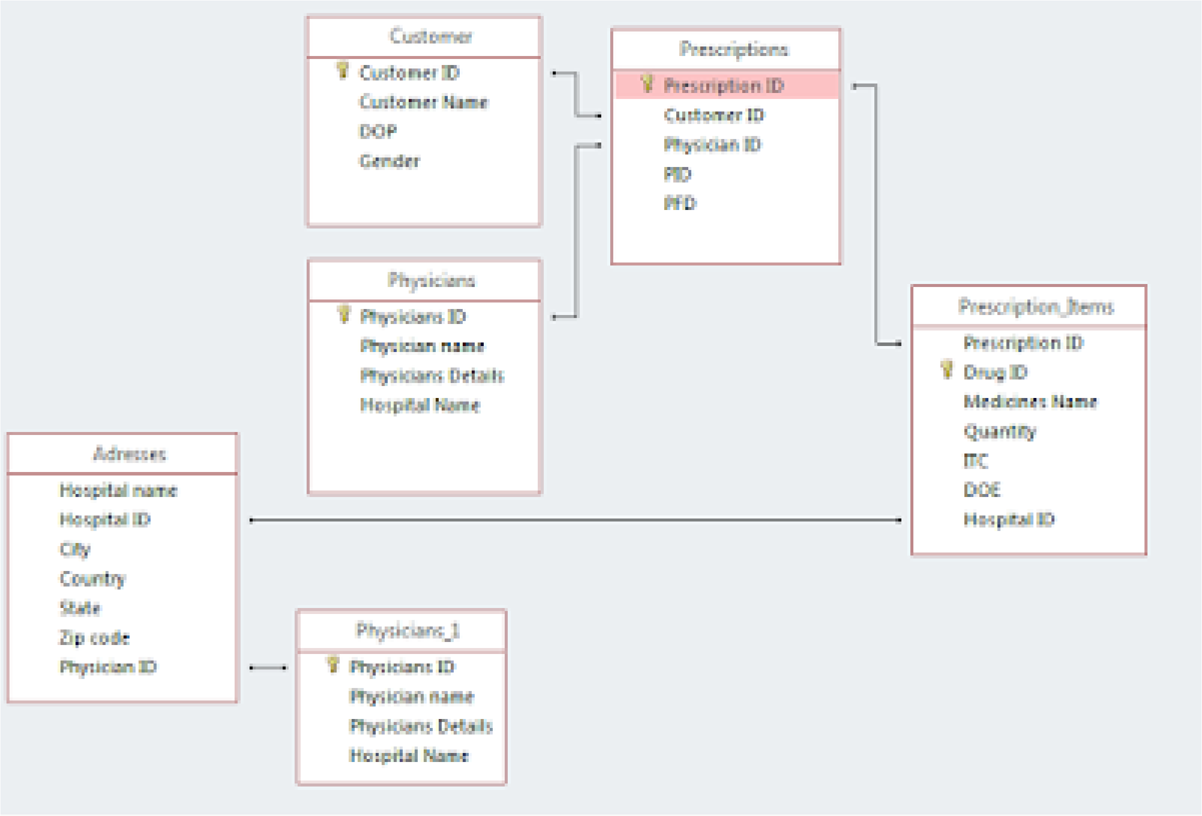
It usually begins with a context diagram as level 0 of the DFD diagram, a simple representation of the whole system. To elaborate further from that, we drill down to a level 1 diagram with lower-level functions decomposed from the major functions of the system. This could continue to evolve to become a level 2 diagram when further analysis is required.

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. They can be used to analyse an existing system or model a new one.



3.4 Database Design/Schema Diagram

The database schema will show the relations between all the tables and will also justify the use of indexes in the tables because without creating indexes relations weren’t possible. Also, the advantages of relationships are **ON DELETE CASCADE AND ON UPDATE CASCADE.** We don’t have to change or delete the details from every table. If the main key is deleted or updated then the indexed key will reflect the change respectively.



Login Details

Field Name	Field Type	Field Length	Description
Admin Username	Varchar	(15)	Admin Username
Password	Varchar	(15)	Admin password

Medicine Information

Field Name	**Field Type**	**Field Length**	**Description**
Medicine Name	Varchar	(15)	Medicine Name
Medicine ID	Varchar	(7)	Medicine Number in stock
Manufacturer	Varchar	( )	Manufacturer
Batch No	Varchar	( )	Batch Number
Production Date	Date/Time	( )	Production Date
Expiry Date	Date/Time	( )	Expiry Date
Dosage	Text	( )	Dosage
Reg_Date	Date/Time	( )	Registration Date
Quantity	Int	( )	Quantity
Cost Price	Int	( )	Cost Price
Interest Rate	Int	( )	Interest Rate
Expected Sale	Int	( )	Expected Sale

3.5 Methodology

Collection of fact is the act of getting and gathering information from various sources in order to be able to compose the project. Data used for designing of the system were gathered through several means. Therefore, the method used in the design and collections of information from various sources are as follows:

1. Collecting and analyzing existing materials on the project topic, written by different expert.
2. Studying the present system in detail and the organizational style.
3. Knowing and understanding the input and output processes of the existing system.
4. Knowing and understanding the input and output processes of the existing system.

5. Interviews: A qualitative form of interview was conducted in the pharmacy to know the equipment needed, and the mode of operation of the old system.
6. Primary data: This source has to do with the text book contacted for the development of this project.

Chapter-4. Implementation, Testing and Maintenance

4.1 Tools & Technology Used

4.1.1 Back-end Technologies

JAVA

Java is a widely used object-oriented programming language and software platform that runs on billions of devices, including notebook computers, mobile devices, gaming consoles, medical devices and many others. The rules and syntax of Java are based on the C and C++ languages.

One major advantage of developing software with Java is its portability. Once you have written code for a Java program on a notebook computer, it is very easy to move the code to a mobile device. When the language was invented in 1991 by James Gosling of Sun Microsystems (later acquired by Oracle), the primary goal was to be able to "write once, run anywhere."

It's also important to understand that Java is much different from JavaScript. Javascript does not need to be compiled, while Java code does need to be compiled. Also, Javascript only runs on web browsers while Java can be run anywhere.

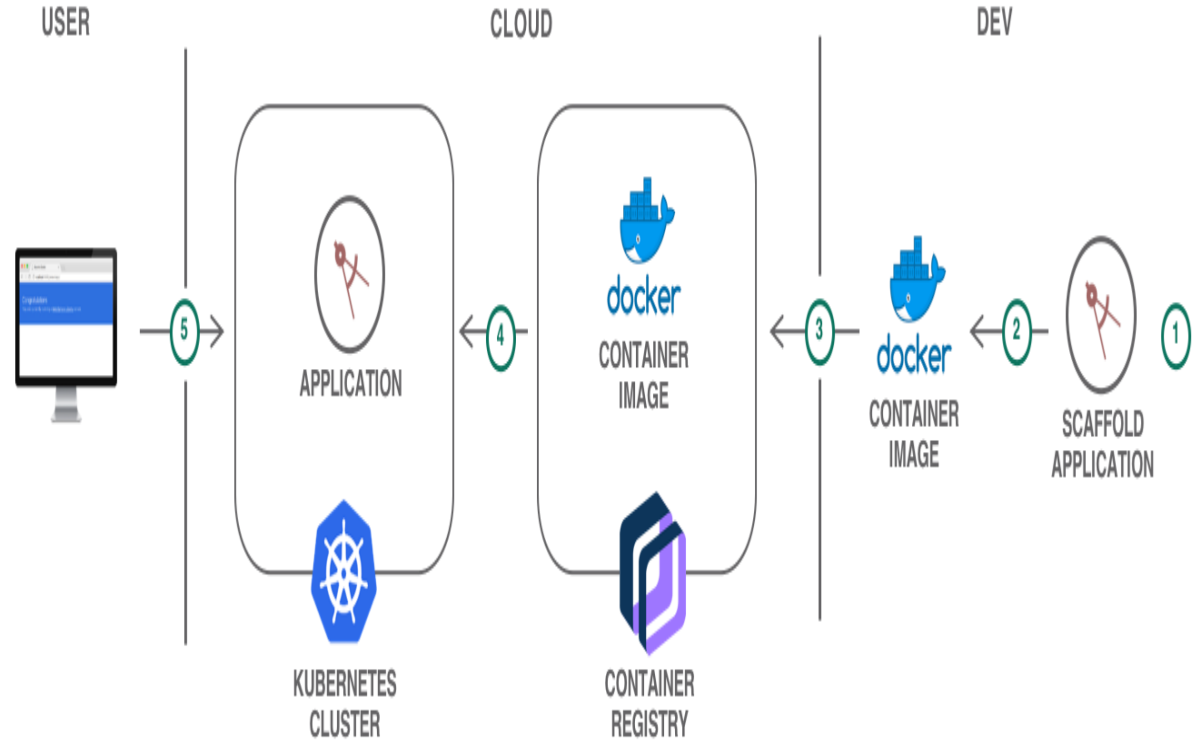
New and improved software development tools are coming to market at a remarkable pace, displacing incumbent products once thought to be indispensable. In light of this continual turnover, Java's longevity is impressive; more than two decades after its creation, Java is still the most popular language for application software development—developers continue to choose it over languages such as Python, Ruby, PHP, Swift, C++, and others. As a result, Java remains an important requirement for competing in the job market.

Benefits

When it comes to choosing a programming language and environment for your next enterprise application, there are solid technical reasons to consider Java, including interoperability, scalability, and adaptability.

The core philosophy behind its creation—interoperability across disparate devices—remains the strongest argument for favoring Java for new enterprise applications. Java's object-oriented architecture allows you to create modular programs and reusable code, shortening development cycles and extending the longevity of enterprise applications.

Platform scalability is a key attribute of Java. With Java, you can use one single system across a broad range of use cases. Existing desktop applications can be easily adapted to run on smaller devices that have limited resources. You can also migrate applications from mobile to desktop, developing business apps for the Android platform and then integrating them into your current desktop software, bypassing lengthy and expensive development cycles.



Java also wins points with strategic planners for its ability to adapt to new use cases. For example, Java is widely considered to be an ideal platform for the Internet of Things (IoT). The typical IoT application interconnects a large number of disparate devices, a task that is greatly simplified by the fact that billions of devices run Java. Furthermore, Java’s extensive ecosystem of developers is constantly developing and sharing new libraries with functionality specifically targeted at IoT application development.

Fortune 500 companies answer the question "What is Java?" by saying that it is a staple in the world of enterprise application development. The vast majority of these companies depend on Java for production applications. Many of these applications are migrating to mobile platforms, a process made much easier because of Java’s portability and scalability.

Despite the ever-widening choice of software technologies, Java remains popular with software managers due to the ease of learning and its ubiquity. Seasoned developers take advantage of extensive Java online resources and communities to streamline application development and simplify troubleshooting, reducing time to market and increasing quality. For both business and technical reasons, Java will continue to be an essential technology in software groups developing enterprise applications.

MYSQL

MySQL is one of the most recognizable technologies in the modern big data ecosystem. Often called the most popular database and currently enjoying widespread, effective use regardless of industry, it’s clear that anyone involved with enterprise data or general IT should at least aim for a basic familiarity of MySQL.

With MySQL, even those new to relational systems can immediately build fast, powerful, and secure data storage systems. MySQL’s programmatic syntax and interfaces are also perfect gateways into the wide world of other popular query languages and structured data stores.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or a place to hold the vast amounts of information in a corporate network. In particular, a relational database is a digital store collecting data and organizing it according to the relational model. In this model, tables consist of rows and columns, and relationships between data elements all follow a strict logical structure. An RDBMS is simply the set of software tools used to actually implement, manage, and query such a database.

Because MySQL enjoys the most widespread use in many industries, business users from new webmasters to experienced managers should strive to understand its main characteristics. Deciding whether to use this technology, and communicating about it effectively, starts with a review of MySQL’s basic availability, structure, philosophy, and usability.

MySQL is widely compatible

Though often associated with internet applications or web services, MySQL was designed to be extensively compatible with other technologies and architectures. The RDBMS runs on all major computing platforms, including Unix-based operating systems, such as the myriad Linux distributions or Mac OS, and Windows.

MySQL's client-server architecture means it can support a variety of backends, as well as different programming interfaces. Data can be directly migrated from MySQL to its forks (e.g. MariaDB), as well as most other RDBMSs thanks to architectural and language similarities.

Established Oracle and third-party migration tools further allow MySQL to move data to and from a vast set of general storage systems, whether these are designed to be on-premises or cloud-based. MySQL can be deployed in virtualized environments, distributed or centralized, and even exists as portable standalone libraries for learning purposes, testing, or small applications.

MySQL's wide compatibility with all these other systems and software makes it a particularly practical choice of RDBMS in most situations.

MySQL databases are relational

The primary factor differentiating relational databases from other digital storage lies in how data is organized at a high level. Databases like MySQL contain records in multiple, separate, and highly codified tables, as opposed to a single all-encompassing repository, or collections of semi- or unstructured documents.

This allows RDBMSs to better optimize actions like data retrieval, updating information, or more complex actions like aggregations. A logical model is defined over all of the contents of the database, describing for example the values allowed in individual columns, characteristics of tables and views, or how indices from two tables are related.

Relational models have remained popular for several reasons. They empower users with intuitive, declarative programming languages — essentially telling the database what result is wanted in language akin to, or at least comprehensible as, written english, instead of meticulously coding up each step of the procedure leading to that result. This moves a lot of the work into the RDBMS and SQL engines, better enforcing logical rules and saving valuable resources and manpower.

MySQL is open-source

Any individual or enterprise may freely use, modify, publish, and expand on Oracle's open-source MySQL code base. The software is released under the GNU General Public License (GPL).

For MySQL code needing to be integrated or included in a commercial application (or if open-source software is not a priority), enterprises can purchase a commercially licensed version from Oracle.

Again, these options provide organizations with additional flexibility if deciding to work with MySQL. The public and community-based nature of open-source releases enriches MySQL's documentation and online support culture, while also ensuring that sustained or newly-developed capabilities never stray too far from current user needs.

MySQL is easy to use

Though MySQL's relational nature and the ensuing rigid storage structures might seem restrictive, the tabular paradigm is perhaps the most intuitive, and ultimately allows for greater usability.

In fact, MySQL makes many concessions to supporting the widest possible variety of data structures, from the standard but rich logical, numeric, alphanumeric, date, and time types, to more advanced JSON or geospatial data. Beyond mere data types and an expansive built-in feature set, the MySQL ecosystem also includes a variety of tools, easing everything from server management to reporting and data analysis.

4.1.2 Front-end Technologies

- **HTML5:-** HTML5 is the fifth revision and newest version of the HTML standard. It offers new features that provide not only rich media support but also enhance support for creating web applications that can interact with users, their local data, and servers more easily and effectively than was previously possible.

- **CSS3:-** CSS i.e., Cascading Style Sheets enables you to define rules on how an element should appear. CSS styles are extremely expressive and detailed and allows you to have a very high degree of control over the styling of HTML elements with very little code. CSS3 together with HTML5 has been at the forefront of advanced Web Design and Web Development.
- **BOOTSTRAP:-** Bootstrap is an HTML, CSS and JavaScript Framework and it is useful to develop responsive websites without rewriting the code for each device or screen. Bootstrap has not only HTML/CSS components but also JS components. With Bootstrap you get to use common HTML elements but with a beautiful representation of items.
- **JAVASCRIPT:-** JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

  Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. Over 97% of websites use it client-side for web page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on the user's device.

  As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

4.1.3 Tools

- **VISUAL STUDIO CODE:-** Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.

NETBEANS

NetBeans IDE is a free, open source, integrated development environment (IDE) that enables you to develop desktop, mobile and web applications. The IDE supports application development in various languages, including Java, HTML5, PHP and C++. The IDE provides integrated support for the complete development cycle, from project creation through debugging, profiling and deployment. The IDE runs on Windows, Linux, Mac OS X, and other UNIX-based systems.

The IDE provides comprehensive support for JDK 7 technologies and the most recent Java enhancements. It is the first IDE that provides support for JDK 7, Java EE 7, and JavaFX 2. The IDE fully supports Java EE using the latest standards for Java, XML, Web services, and SQL and fully supports the GlassFish Server, the reference implementation of Java EE.

4.2 Coding Standard of Languages used

4.2.1 Back-end Coding Guidelines

Step 1: Verify that it is already installed or not. Check whether Java is already installed on the system or not. ...

Step 2: Download JDK. Click the below link to download jdk 1.8 for you windows 64 bit system. ...

Step 3: Install JDK. ...

Step 4 : Set the Permanent Path.

4.2.2 Front-end Coding Guidelines

Don’t use CDN links.. Always download files and load from the local directory

- **HTML**

1. Variable Name:

   Use Dash case for all classes or id’s in HTML

   Eg: <div class=”graph-card”></div>

2. Don’t use inline styling as they can’t be overridden by the style sheet.

   Styles will follow the following priority.

   - Inline style (Higher Priority)
   - Internal styles (Medium Priority)
   - External styles (Lower Priority)

3. Every html element should be inside in container or container fluid

4. Wrap all the columns inside the row and each its corresponding device width. Eg: col-lg-3 col-md-4 col-sm-6

5. Don’t override the default classes. If you want to do change the style, write a separate class and add it to the default class

6. Wrap the icons in span or div and give a classname. Apply styles to that classname

- **CSS**

1. Imports all the fonts in the stylesheet.
2. Write media queries at the bottom of the script.
    - Follow ascending order for max-width (Recommended)
    - Follow descending order for min-width
    - Important breakpoints 1024px, 767px,420px
3. Don’t write !important to override the properties instead create a custom class and use it.

- **Script**

1. Variable name :
    a. Use camelcase for declaring variables in javascript/jquery.

2. Function name :
    a. Use meaningful function names and in camel cases.

3. Use // for inline comments and /* */ for multi line comments

4. Keep all the functions at the bottom of your script.

4.3 Testing Techniques and Test Plans

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

4.3.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive.

4.3.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, consistent. Integration testing is specifically aimed at ensuring the combination of components is correct and exposing the problems.

4.3.3 Functional Test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

4.3.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test.

4.3.5 White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It has a purpose. It is used to test areas that cannot be reached from a black box level.

4.3.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. It is a test in which the software under test is treated as a black box . You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

4.3.7 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

4.4 Maintenance

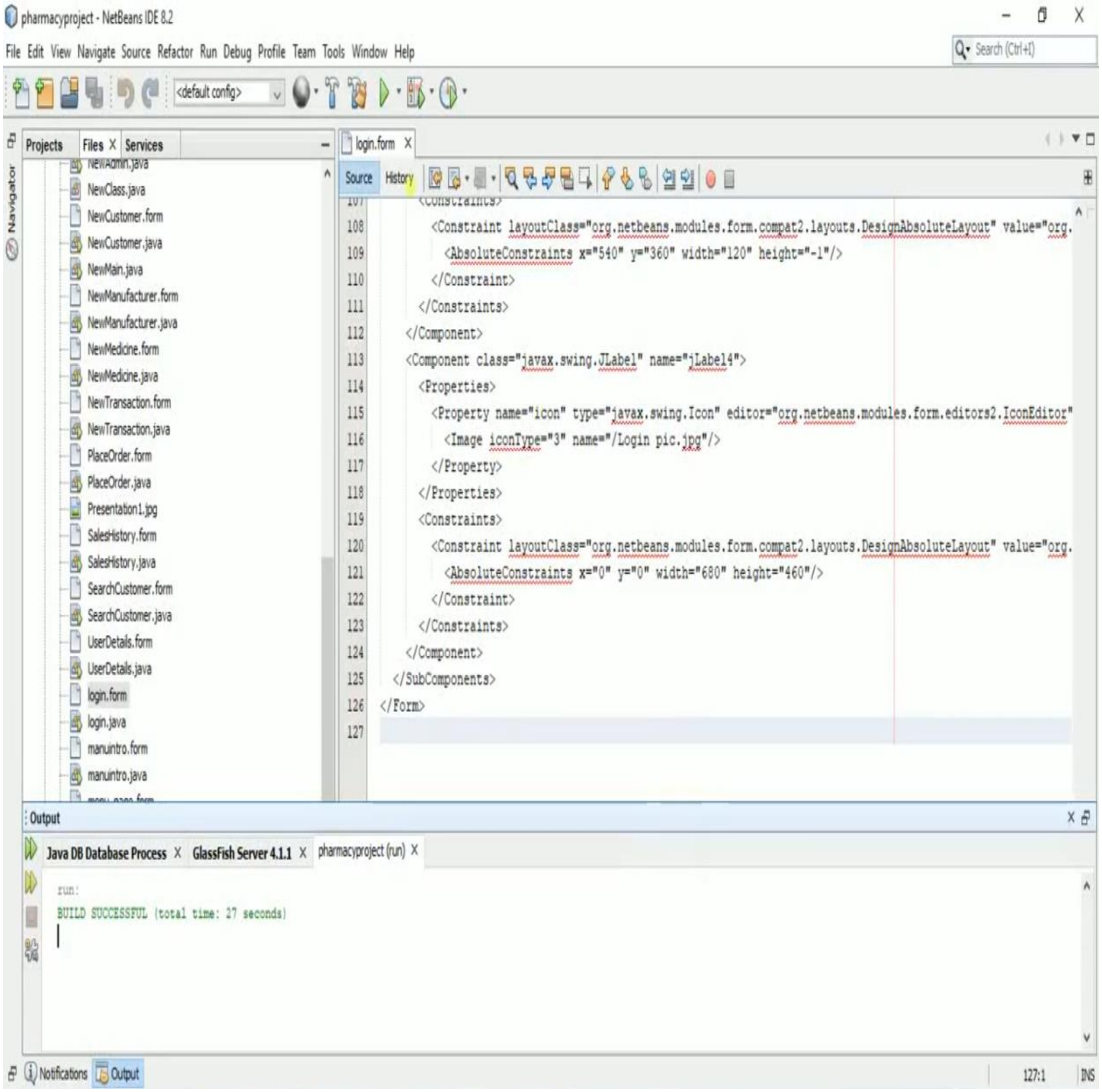
Once the code runs properly, it enters the maintenance phase. All systems need maintenance. Maintenance is required because there are often some residual errors remaining in the system that must be removed as they are discovered. Maintenance involves understanding the effects of the change, making the changes to both the code and the documents, testing the new parts and retesting the old parts that were not changed. Maintenance is mainly of two types:

- ★ **Corrective Maintenance:** Almost all software that is developed has residual errors or bugs in them. Many of these surfaces only after the system has been in operation, sometimes for a long time. These errors once discovered need to be removed, leading to the software to be changed. This is called Corrective Maintenance.

- ★ **Adaptive Maintenance:** Even without bugs, software frequently undergoes change. The software often must be upgraded and enhanced to include more features and provide more services. This requires modification of the software. This type of maintenance is known as Adaptive Maintenance.

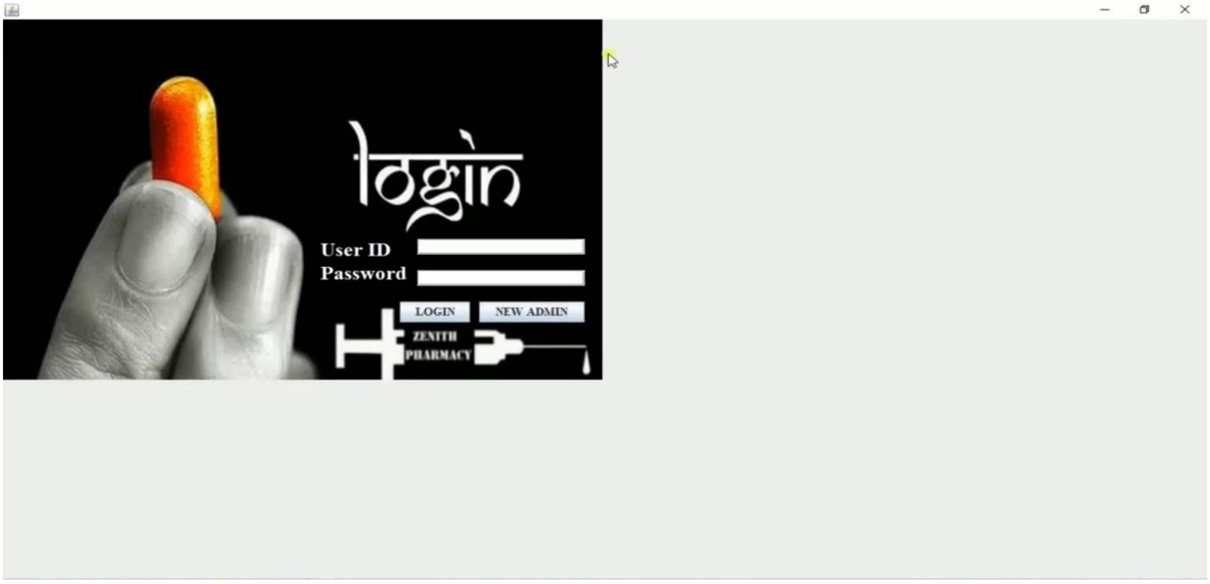
Chapter-5. Result and Discussion

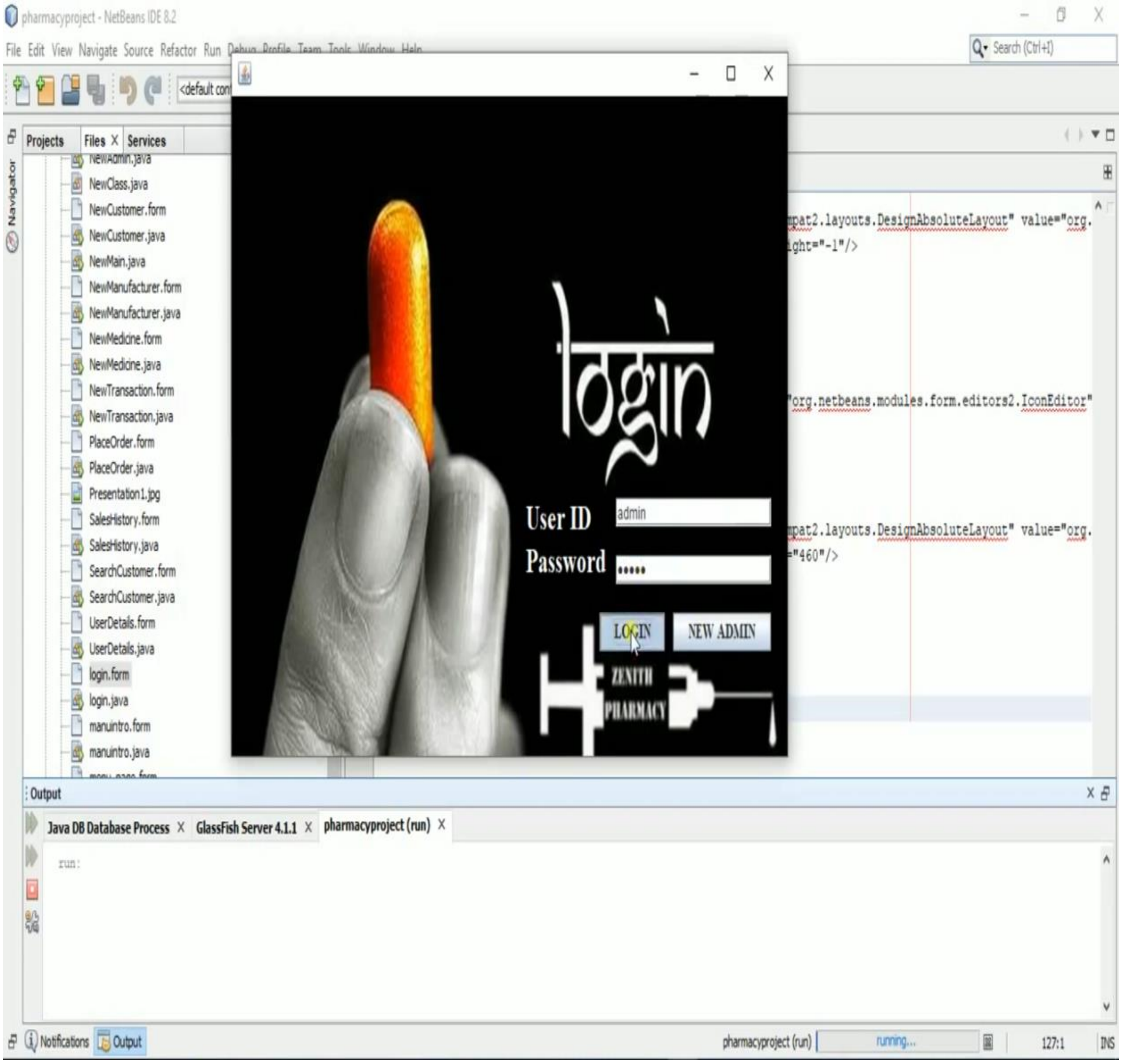
5.1 Snapshots



```
      <Constraints>
        <Constraint layoutClass="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
          <AbsoluteConstraints x="540" y="360" width="120" height="-1"/>
        </Constraint>
      </Constraints>
    </Component>
    <Component class="javax.swing.JLabel" name="jLabel4">
      <Properties>
        <Property name="icon" type="javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor"
          <Image iconType="3" name="/Login pic.jpg"/>
        </Property>
      </Properties>
      <Constraints>
        <Constraint layoutClass="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
          <AbsoluteConstraints x="0" y="0" width="680" height="460"/>
        </Constraint>
      </Constraints>
    </Component>
  </SubComponents>
</Form>
```

```
run:
BUILD SUCCESSFUL (total time: 27 seconds)
```



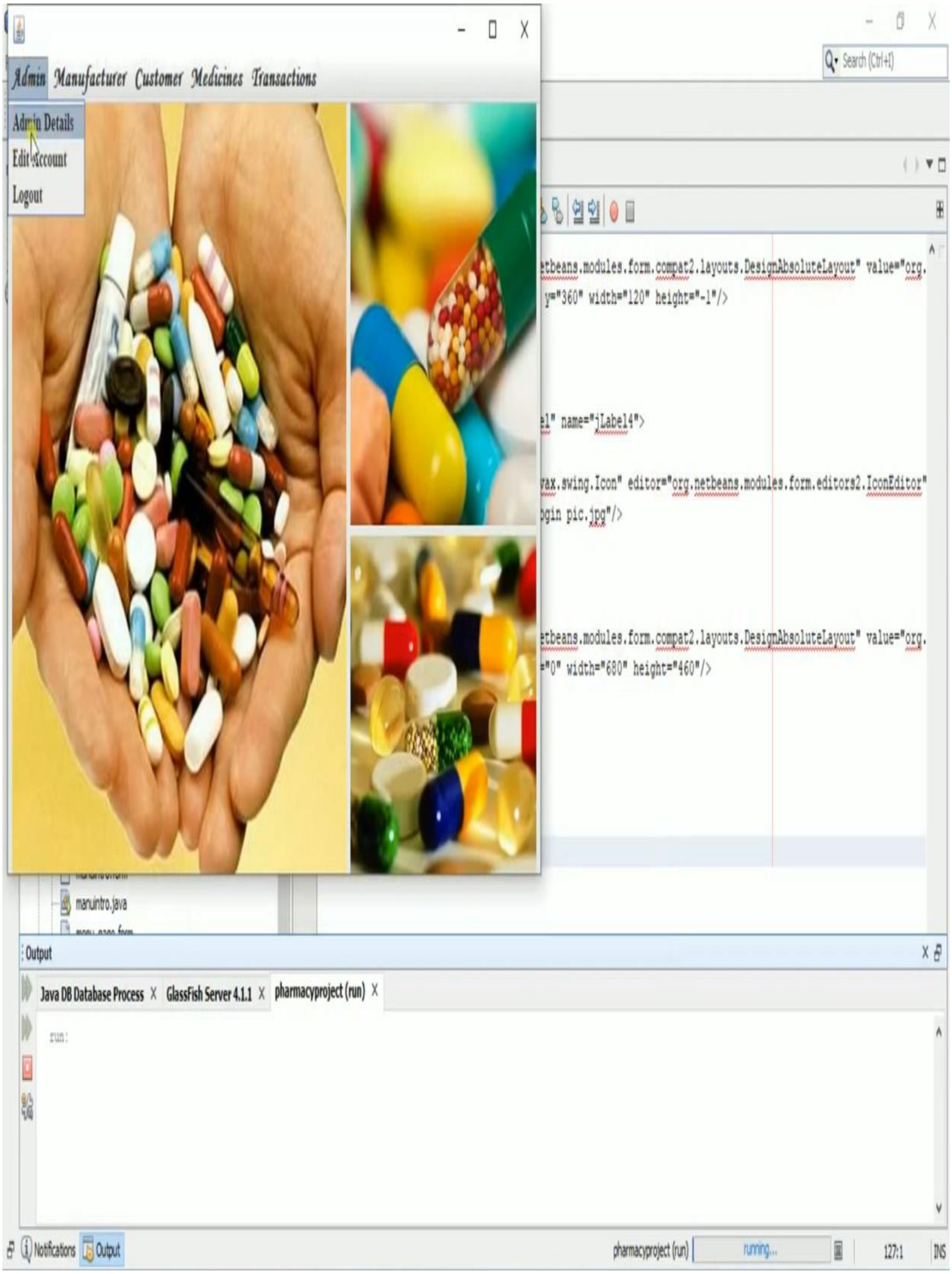


User ID admin

Password •••••

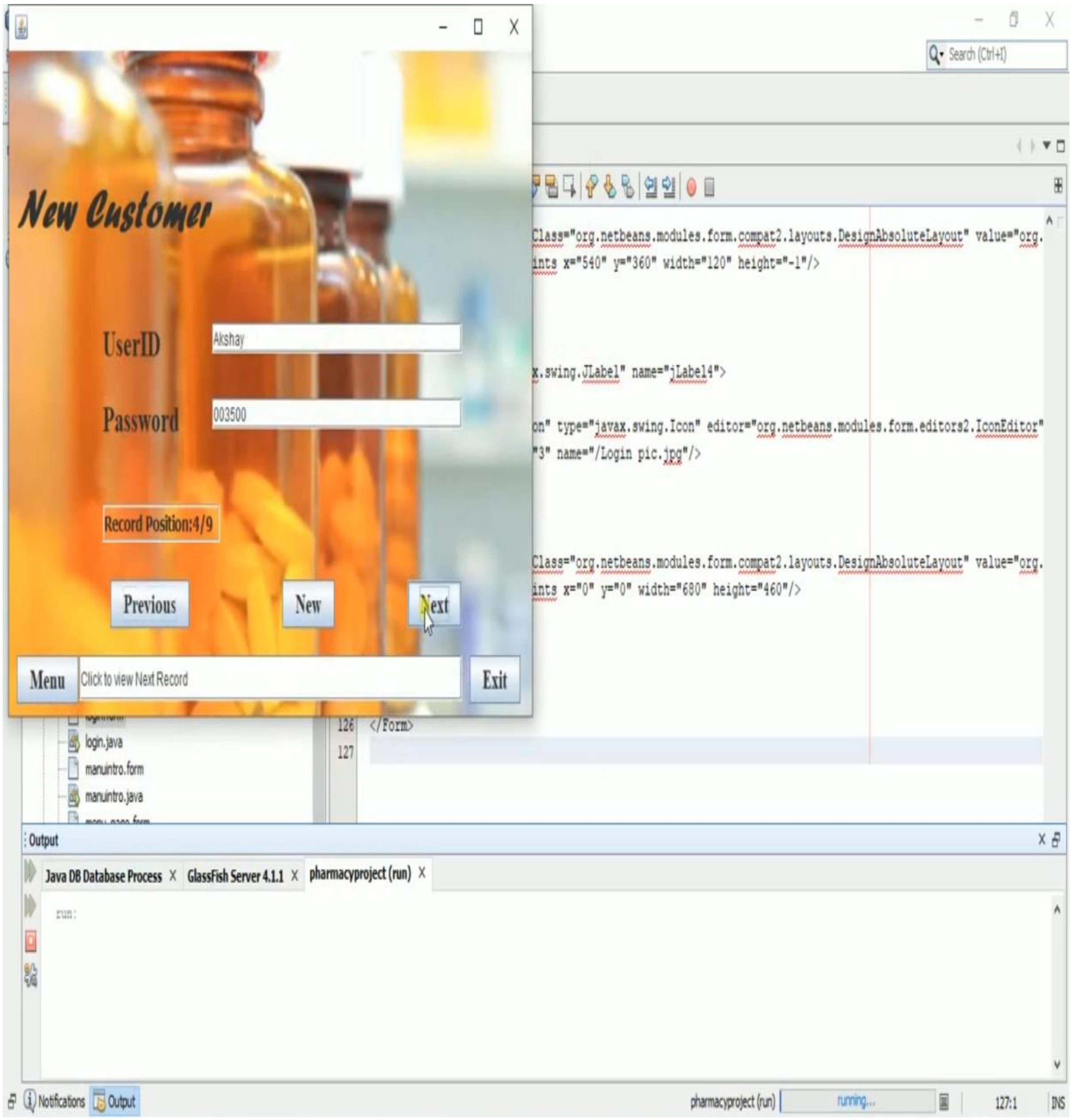
```
pat2.layouts.DesignAbsoluteLayout" value="org.
ight="-1"/>

"org.netbeans.modules.form.editors2.IconEditor"

pat2.layouts.DesignAbsoluteLayout" value="org.
="460"/>
```

```
run:
```



```
etbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
 y="360" width="120" height="-1"/>

el" name="jLabel4">

vax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor"
ogin pic.jpg"/>

etbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
="0" width="680" height="460"/>
```

```
run:
```



UserID: Akshay

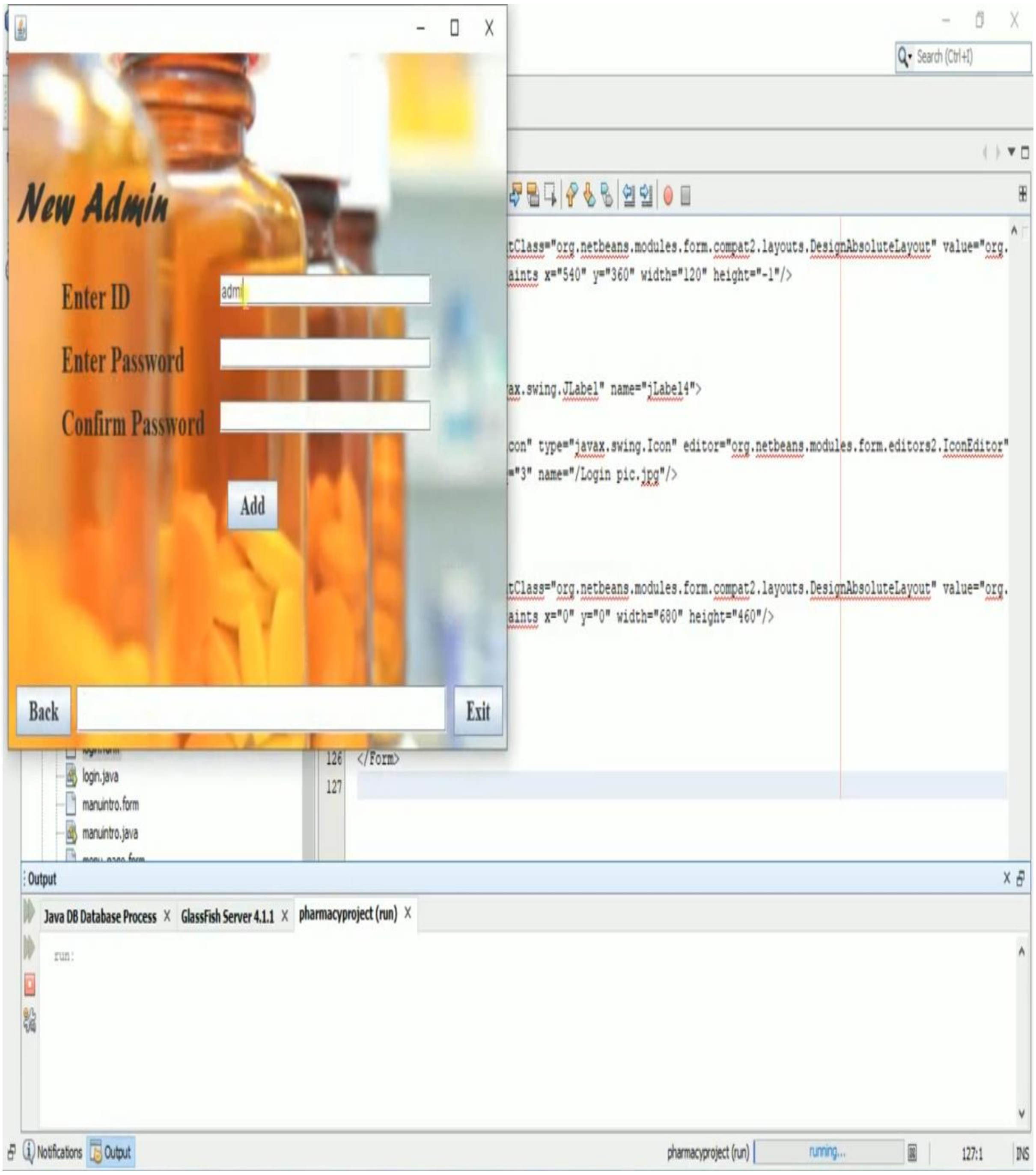
Password: 003500

Record Position:4/9

Click to view Next Record

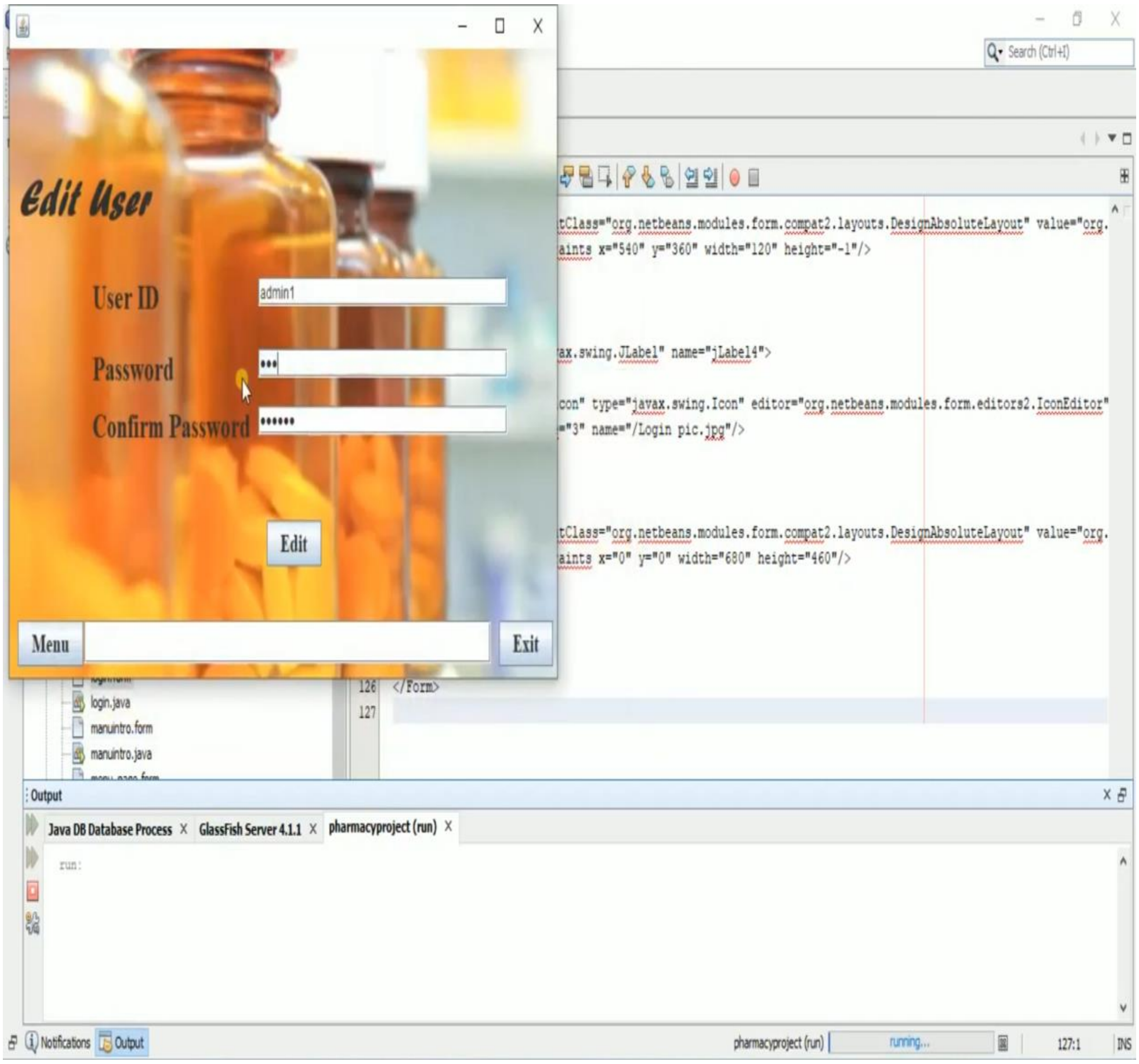
```
Class="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
ints x="540" y="360" width="120" height="-1"/>

x.swing.JLabel" name="jLabel4">

on" type="javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor"
"3" name="/Login pic.jpg"/>

Class="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
ints x="0" y="0" width="680" height="460"/>

</Form>
```

```
run:
```



```
tClass="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
aints x="540" y="360" width="120" height="-1"/>

ax.swing.JLabel" name="jLabel4">

con" type="javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor"
="3" name="/Login pic.jpg"/>

tClass="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
aints x="0" y="0" width="680" height="460"/>

</Form>
```

```
run:
```



Edit User

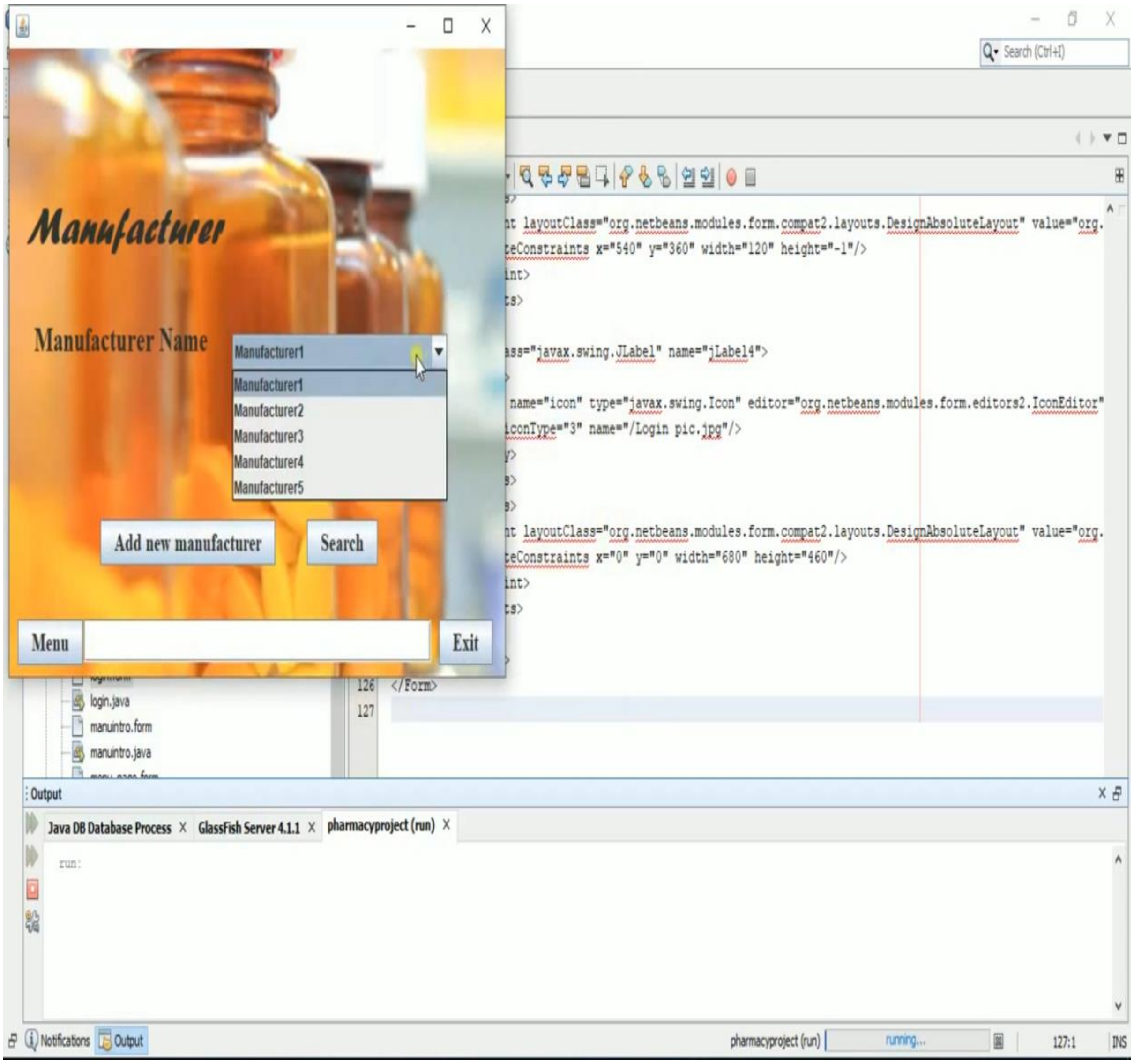
User ID: admin1

Password: •••

Confirm Password: ••••••

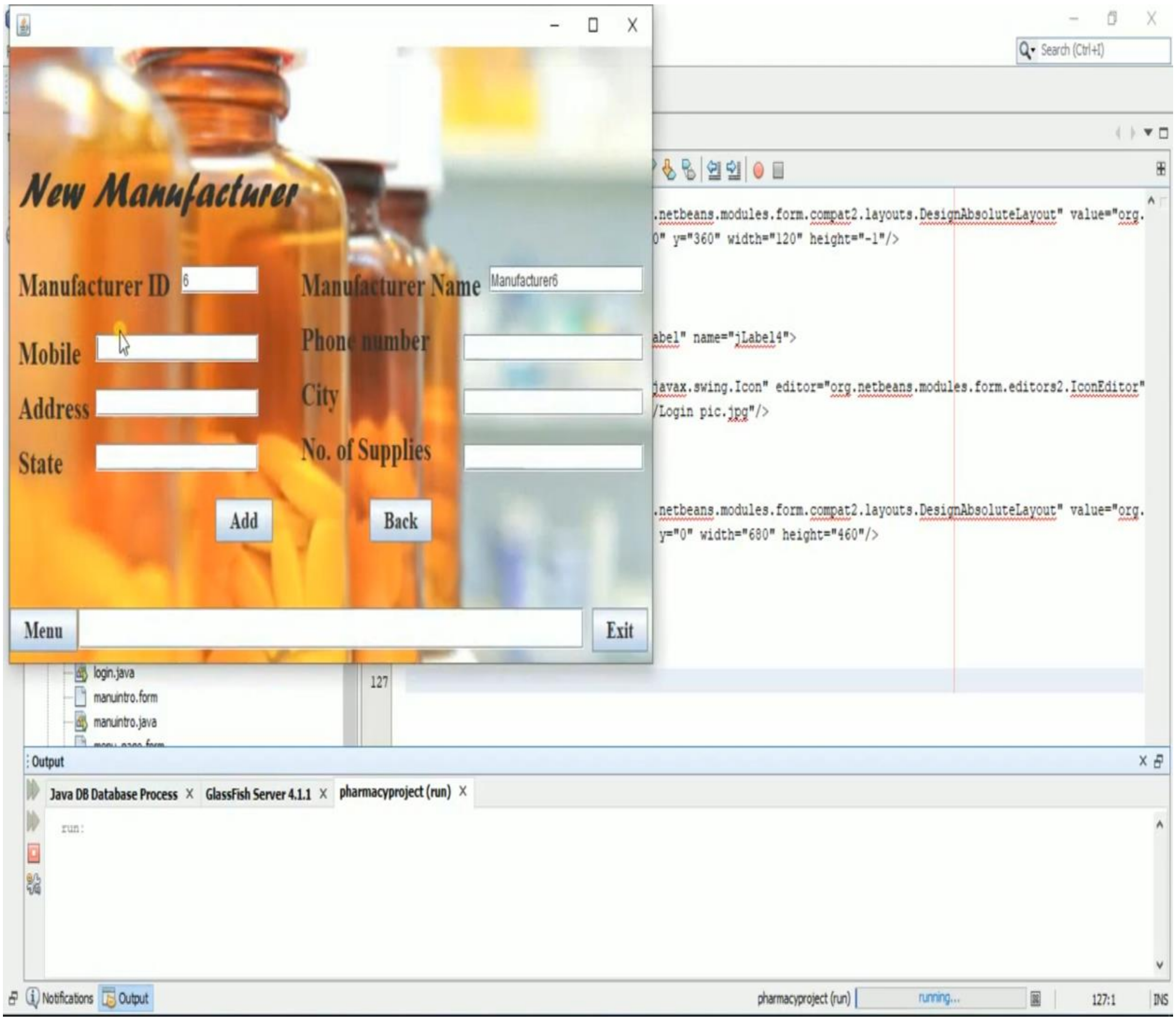
```
tClass="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
aints x="540" y="360" width="120" height="-1"/>

ax.swing.JLabel" name="jLabel4">

con" type="javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor"
="3" name="/Login pic.jpg"/>

tClass="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
aints x="0" y="0" width="680" height="460"/>

</Form>
```

```
run:
```



```
s>
nt layoutClass="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
teConstraints x="540" y="360" width="120" height="-1"/>
int>
ts>

ass="javax.swing.JLabel" name="jLabel4">
>
 name="icon" type="javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor"
iconType="3" name="/Login pic.jpg"/>
y>
s>
s>
nt layoutClass="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
teConstraints x="0" y="0" width="680" height="460"/>
int>
ts>

>
</Form>
```

```
run:
```

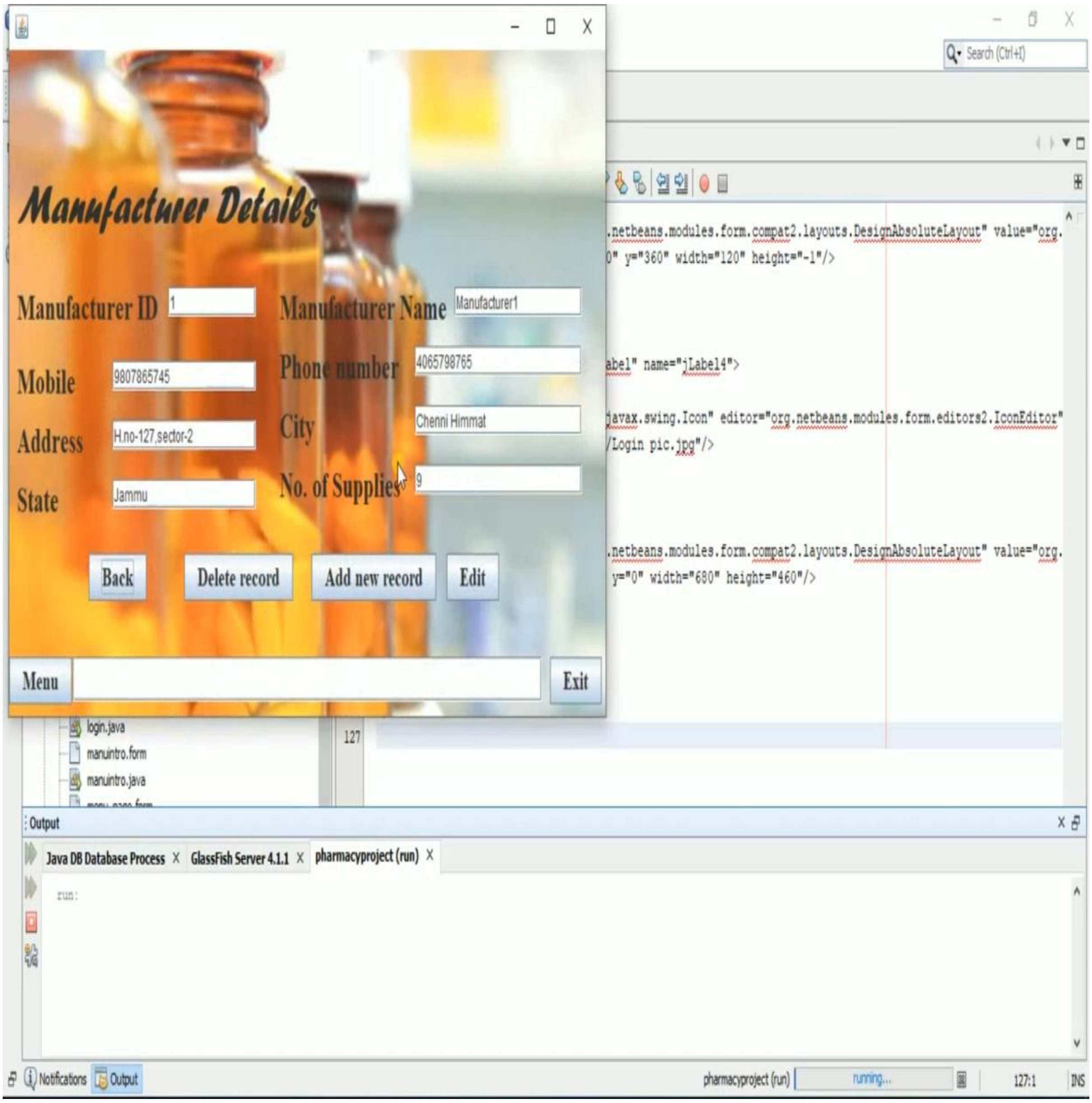


New Manufacturer

Manufacturer ID	6	Manufacturer Name	Manufacturer6
Mobile		Phone number	
Address		City	
State		No. of Supplies	

```
.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
0" y="360" width="120" height="-1"/>

abel" name="jLabel4">

javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor"
/Login pic.jpg"/>

.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
 y="0" width="680" height="460"/>
```

```
run:
```



Manufacturer Details

Manufacturer ID: 1

Manufacturer Name: Manufacturer1

Mobile: 9807865745

Phone number: 4065798765

Address: H.no-127,sector-2

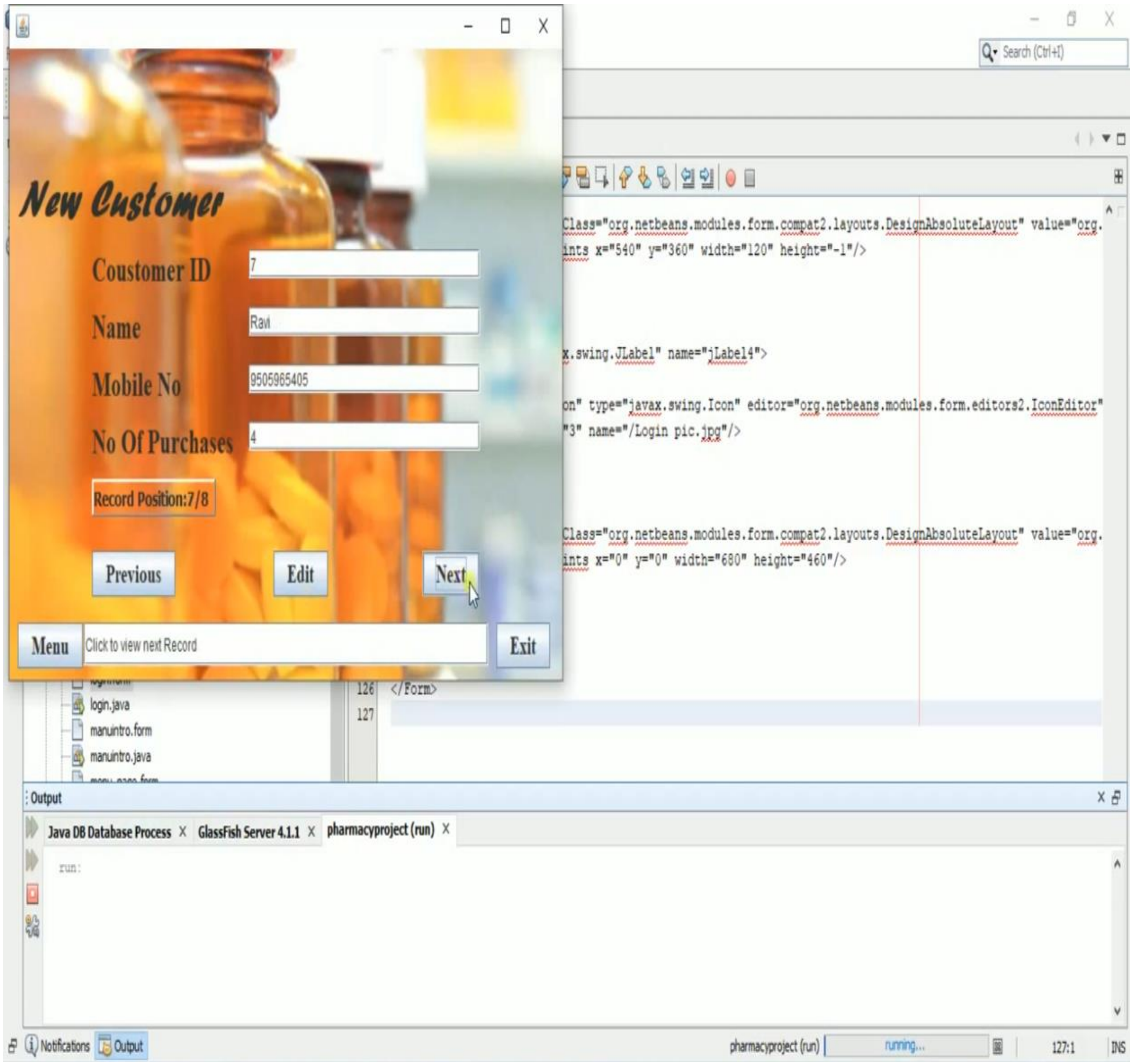
City: Chenni Himmat

State: Jammu

No. of Supplies: 9

```
.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
0" y="360" width="120" height="-1"/>

abel" name="jLabel4">

javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor"
/Login pic.jpg"/>

.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
 y="0" width="680" height="460"/>
```

```
run:
```



New Customer

Coustomer ID: 7

Name: Ravi

Mobile No: 9505965405

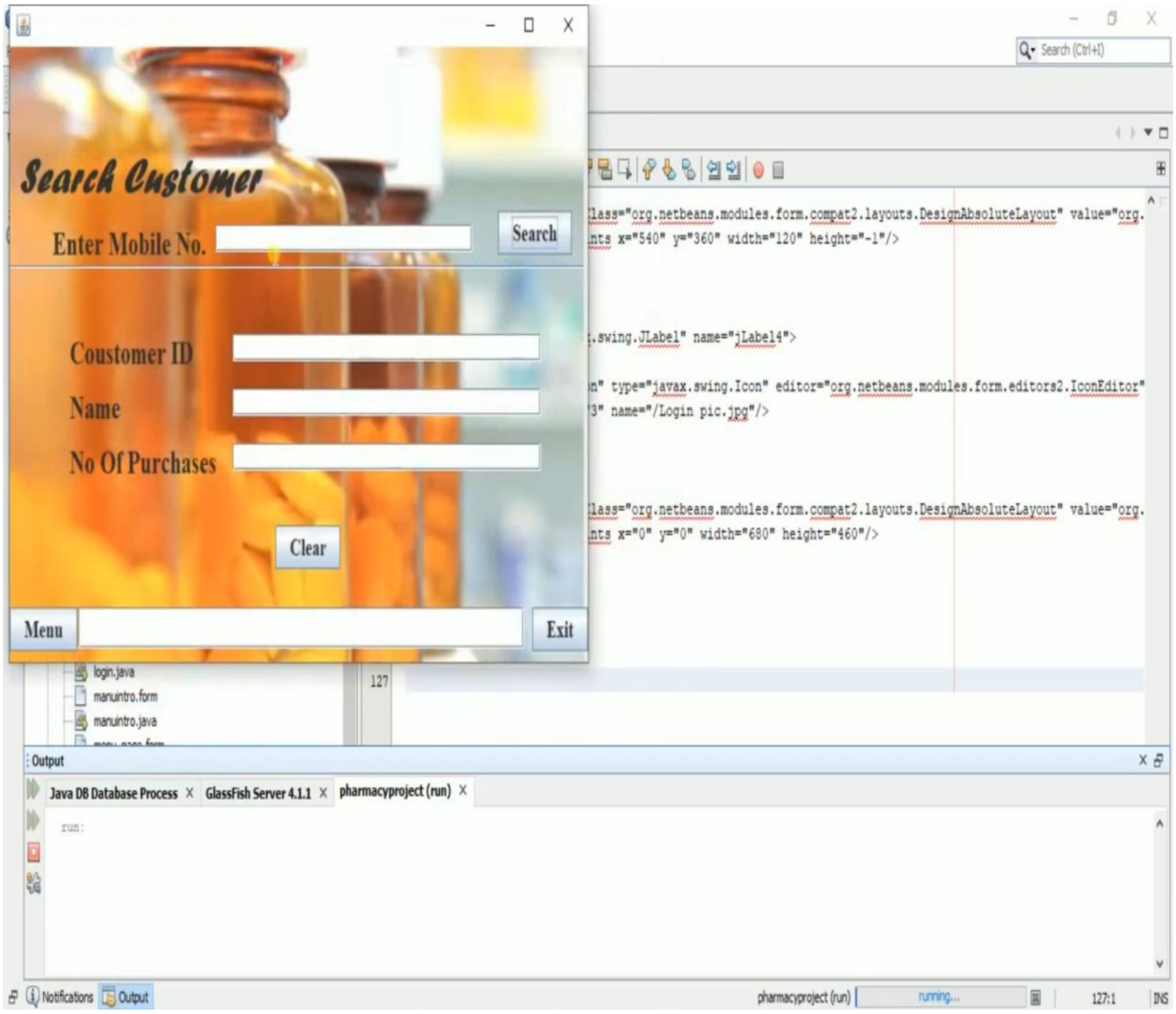
No Of Purchases: 4

Record Position:7/8

Click to view next Record

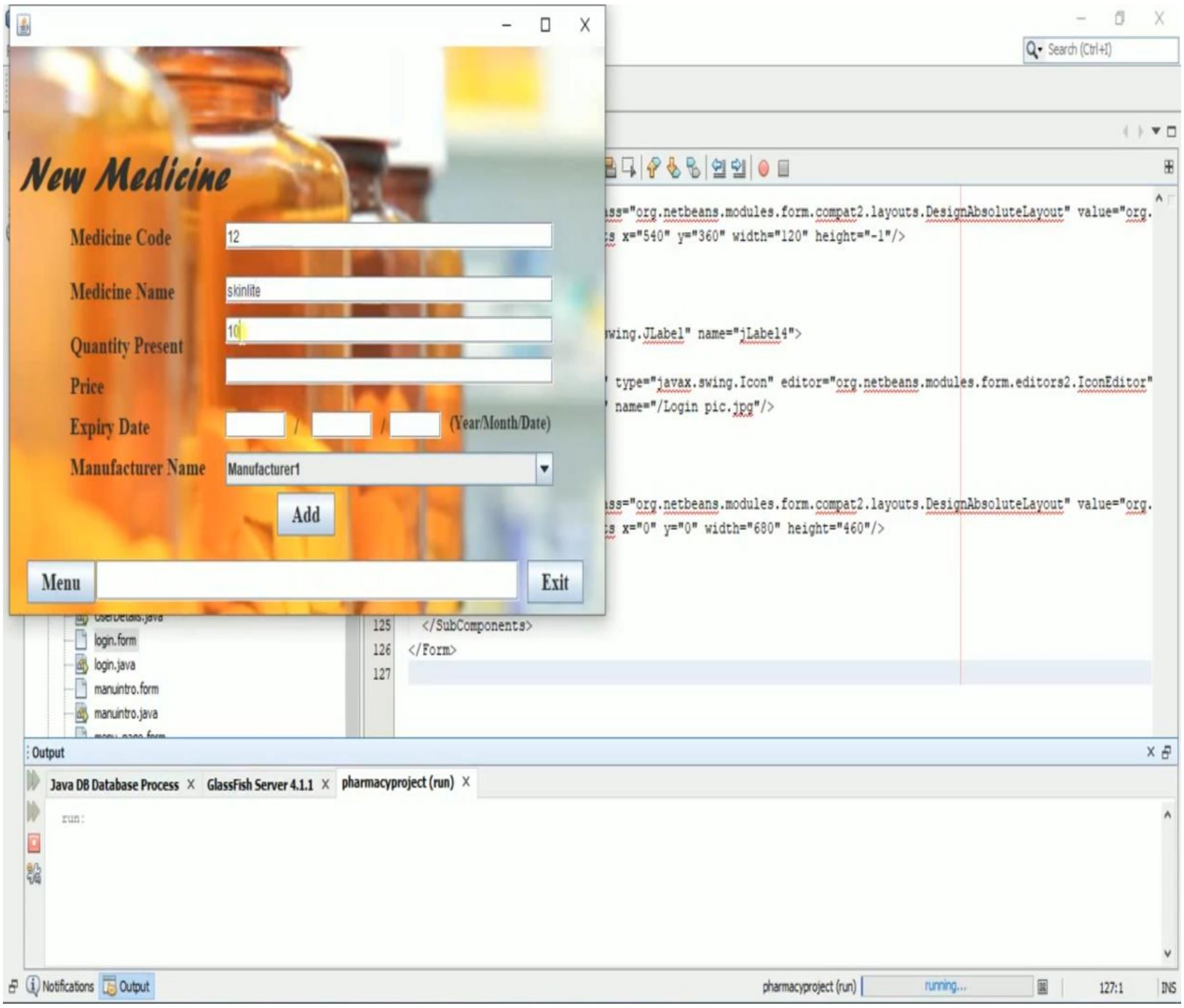
```
Class="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
ints x="540" y="360" width="120" height="-1"/>



x.swing.JLabel" name="jLabel4">

on" type="javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor"
"3" name="/Login pic.jpg"/>



Class="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
ints x="0" y="0" width="680" height="460"/>



</Form>
```

```
run:
```



```
lass="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
nts x="540" y="360" width="120" height="-1"/>

.swing.JLabel" name="jLabel4">

n" type="javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor"
3" name="/Login pic.jpg"/>

lass="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
nts x="0" y="0" width="680" height="460"/>
```

```
run:
```



New Medicine

Medicine Code: 12

Medicine Name: skinlite

Quantity Present: 10

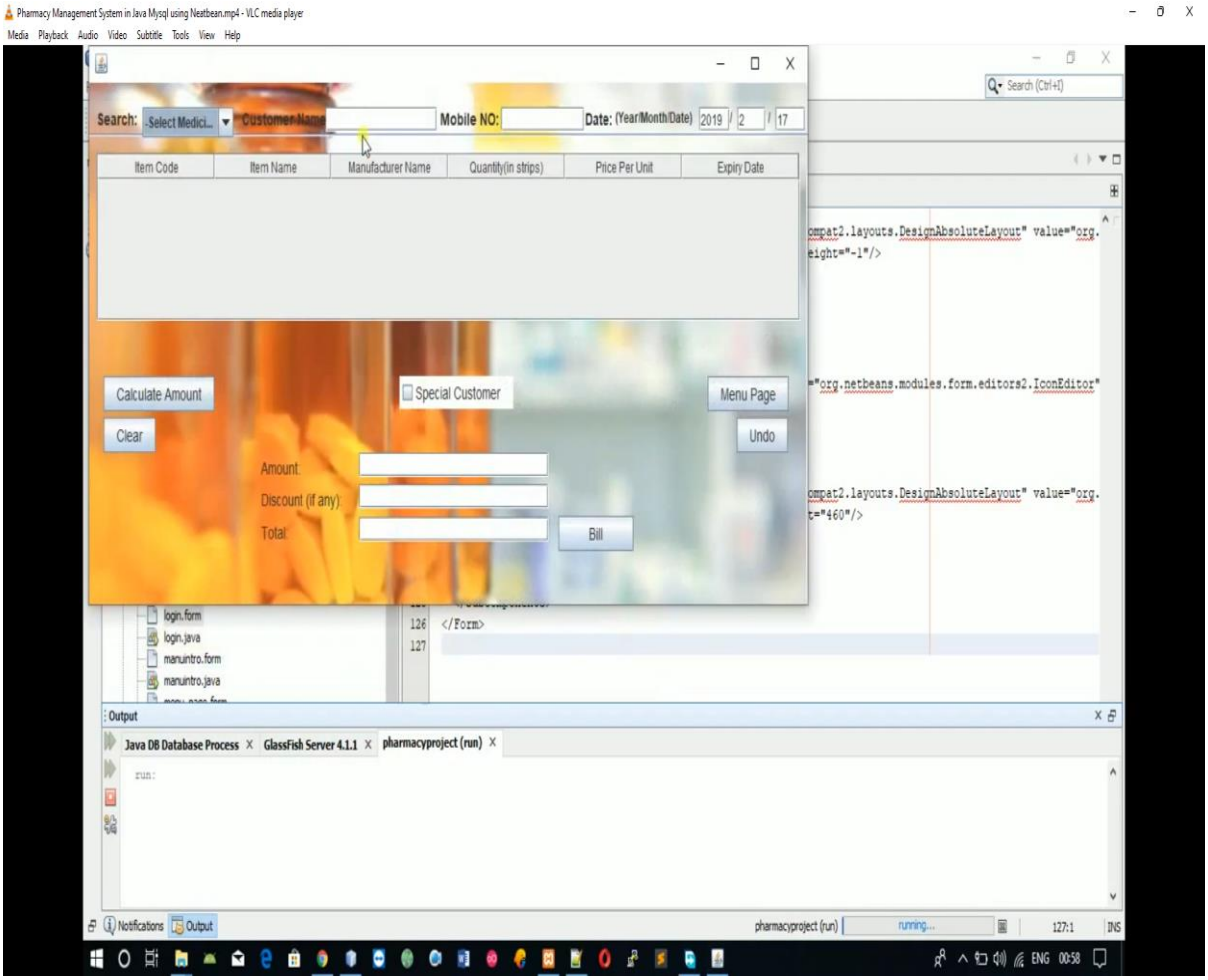
Price:

Expiry Date: / / (Year/Month/Date)

Manufacturer Name: Manufacturer1

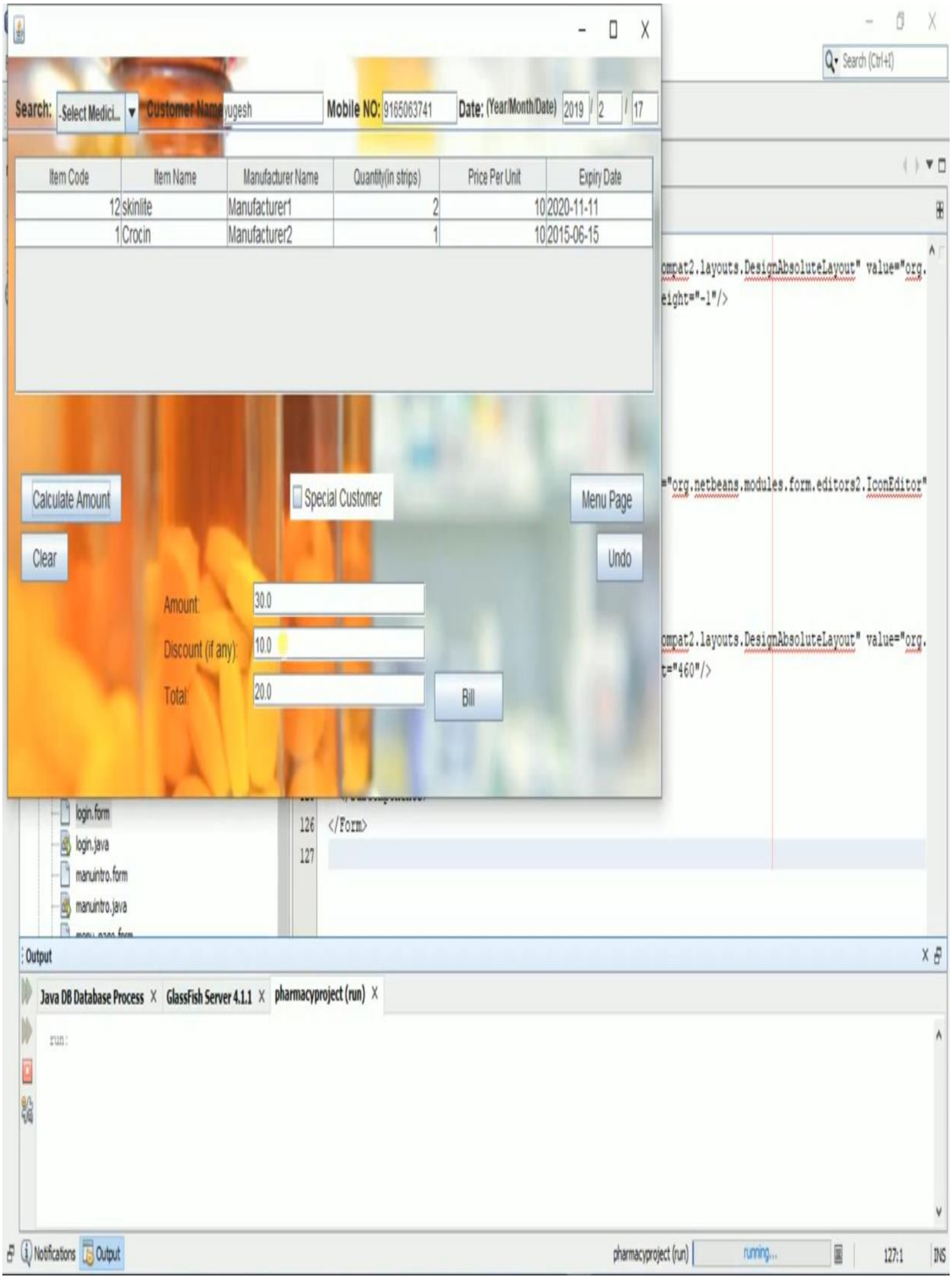
```
ass="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
ts x="540" y="360" width="120" height="-1"/>

wing.JLabel" name="jLabel4">

' type="javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor"
' name="/Login pic.jpg"/>

ass="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
ts x="0" y="0" width="680" height="460"/>

  </SubComponents>
</Form>
```

```
run:
```



```
ompat2.layouts.DesignAbsoluteLayout" value="org.
eight="-1"/>
="org.netbeans.modules.form.editors2.IconEditor"
ompat2.layouts.DesignAbsoluteLayout" value="org.
t="460"/>
</Form>
```

```
run:
```



Search: -Select Medici... ▾ Customer Name yugesh Mobile NO: 9165063741 Date: (Year/Month/Date) 2019 / 2 / 17

Item Code	Item Name	Manufacturer Name	Quantity(in strips)	Price Per Unit	Expiry Date
12	skinlite	Manufacturer1	2	10	2020-11-11
1	Crocin	Manufacturer2	1	10	2015-06-15

□ Special Customer

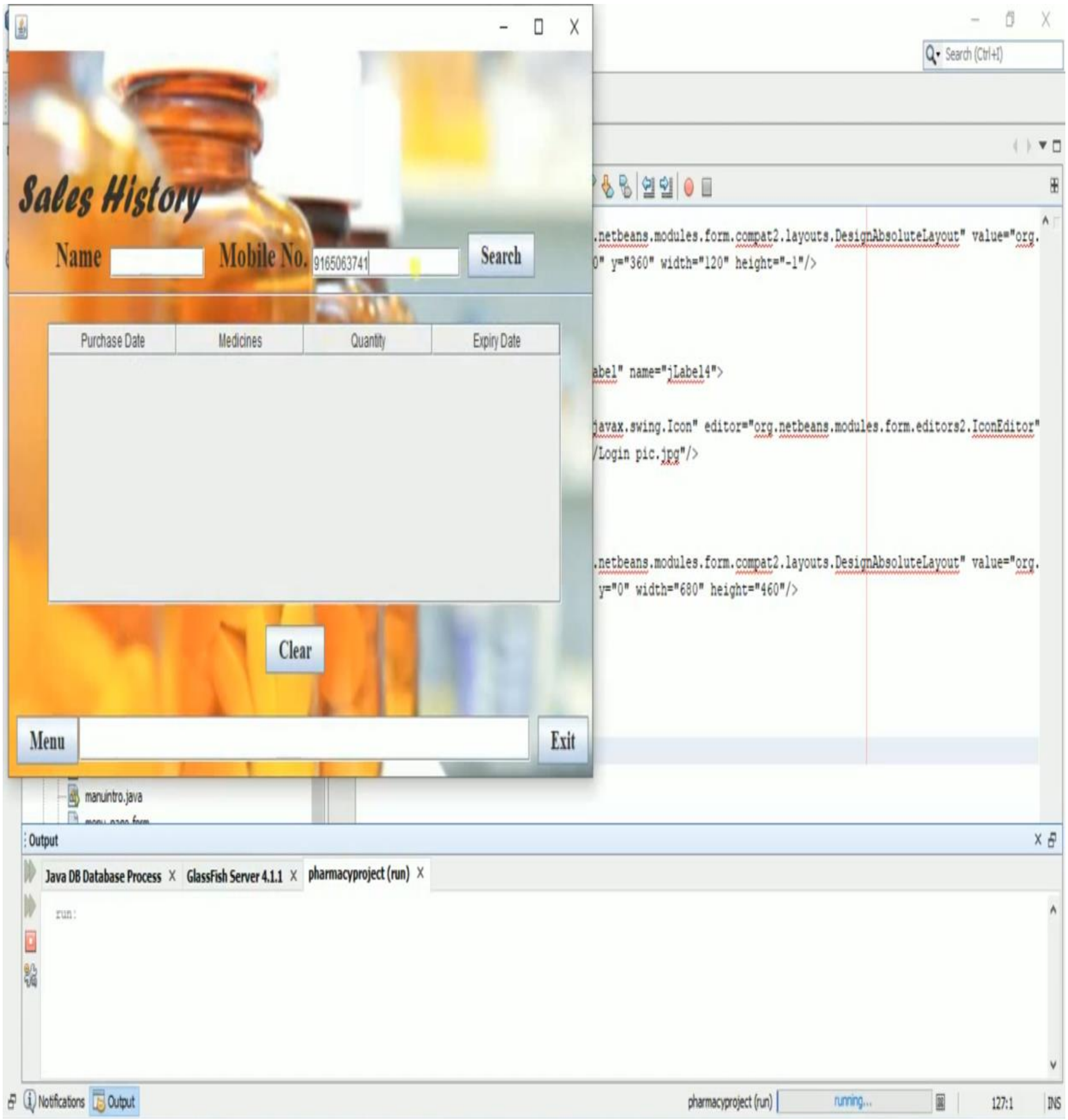
Amount: 30.0

Discount (if any): 10.0

Total: 20.0

```
ompat2.layouts.DesignAbsoluteLayout" value="org.
eight="-1"/>
="org.netbeans.modules.form.editors2.IconEditor"
ompat2.layouts.DesignAbsoluteLayout" value="org.
t="460"/>
</Form>
```

```
run:
```



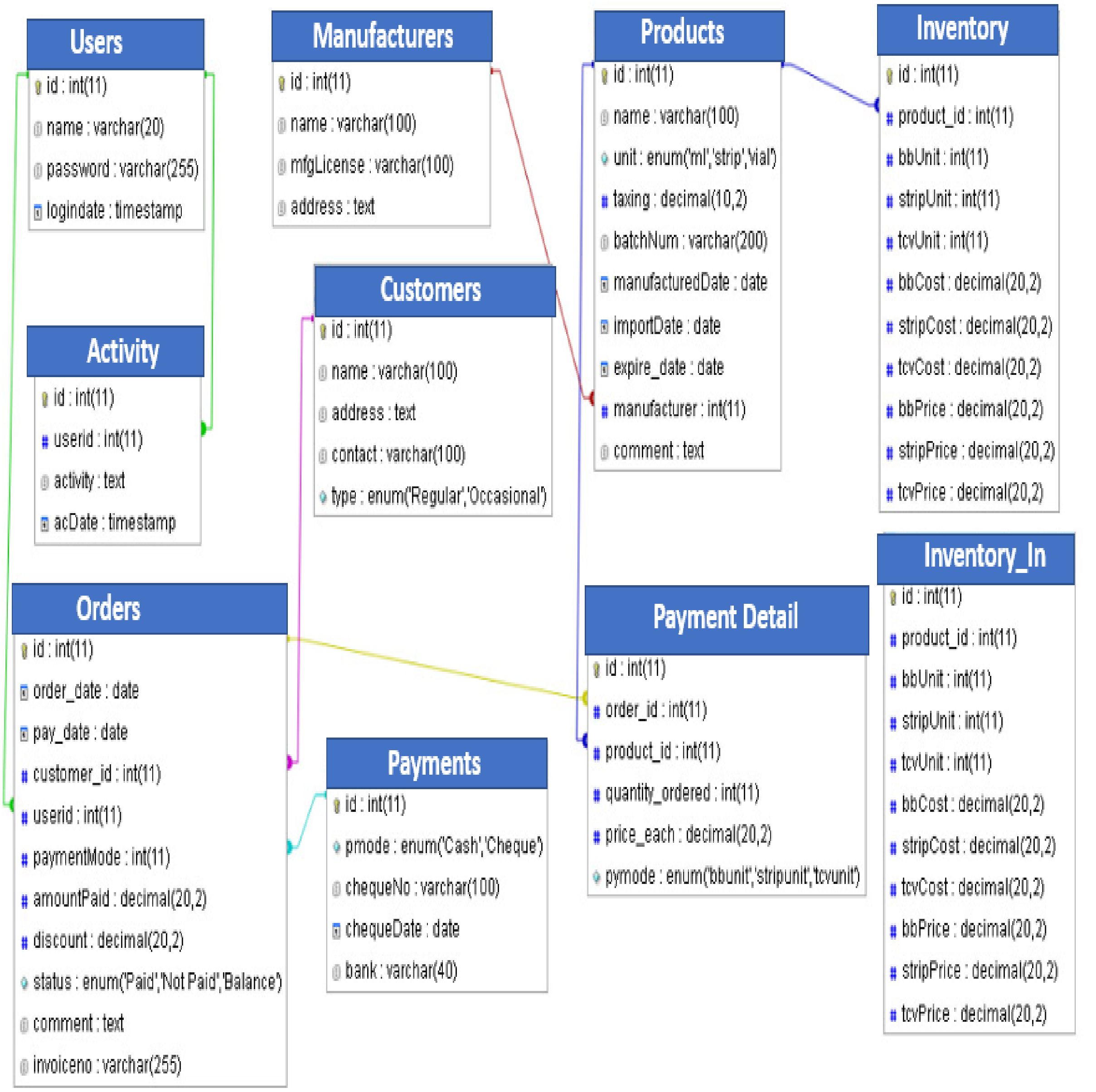
Mobile No. 9165063741

```
.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
0" y="360" width="120" height="-1"/>

abel" name="jLabel4">

javax.swing.Icon" editor="org.netbeans.modules.form.editors2.IconEditor"
/Login pic.jpg"/>

.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout" value="org.
 y="0" width="680" height="460"/>
```

```
run:
```

5.2 Database Collection

Databases are sets of collections. Collections store records, which are referred to as documents. Collections are the equivalent of tables in RDBMS, and documents can be thought of as rows in a table. The difference is that you don’t define what columns (or rather attributes) there will be in advance. Every document in any collection can have arbitrary attribute keys and values. Documents in a single collection will likely have a similar structure in practice however, but the database system itself does not impose it and will operate stable and fast no matter how your data looks like.



Chapter-6. Conclusion and Future Scope

6.1 Conclusion

Effective implementation of this software will take care of the basic requirements of the pharmacy management system because it is capable of providing easy and effective storage of information related to activities happening in the stipulated area. With these, the objectives of the system design will be achieved.

In order to allow for future expansion, the system has been designed in such a way that will allow possible modification as it may deem necessary by the pharmacy management, whenever the idea arises.

6.2 Future Scope

Designing this application (Pharmacy management system) is not an easy task. It all started from the requirement gathering and passes through so many other stages before completion.

Based on the benefits of this system and tremendous value it will add to customer-user satisfaction, the below recommendation will be considered.

It is recommended that the new system should be used with the necessary specifications of the system requirements and provision for an uninterrupted power supply should be made available throughout the hours of operation of the pharmacy to avoid power outage. There should also be basic computer knowledge for the users of the software.

It is recommended that the software be improved especially in areas of accounting as it will be of great impact to the development of retail pharmacy.

Chapter-7. References

1. https://www.talend.com/resources/what-is-mysql/
2. https://www.w3schools.com/java/
3. https://www.javatpoint.com/
4. https://www.tutorialspoint.com/java/index.htm
5. https://www.geeksforgeeks.org/java-how-to-start-learning-java/
6. https://projectworlds.in/java-projects-with-source-code/pharmacy-management-system-in-java-mysql-using-netbeans/