![GALGOTIAS UNIVERSITY]

ONLINE PHARMACY MANAGEMENT SYSTEM

A Project Report of Capstone Project - 3

Submitted by

SHIVANSH BAJPAI
(1613101696 / 16SCSE101044)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

Mr. G. NAGARAJAN
Professor

APRIL/MAY- 2020

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **"ONLINE PHARMACY MANAGEMENT SYSTEM"** is the bonafide work of "**SHIVANSH BAJPAI (1613101696)"** who carried out the project work under my supervision.

SIGNATURE OF HEAD

Dr. MUNISH SHABARWAL,
PhD (Management), PhD (CS)
Professor & Dean,
School of Computing Science &
Engineering

SIGNATURE OF SUPERVISOR

Mr. G NAGARAJAN M.Tech., Ph.D.,
Professor
School of Computing Science &
Engineering

TABLE OF CONTENTS

ACKNOWLEDGMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Mr. G. Nagarajan** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I am extremely indebted to Mr. G. Nagarajan, Department of Computer Science and Engineering, Galgotias University for his valuable suggestions and constant support throughout my project tenure. I would like to express our thanks to all faculty and Staff members of Department of Computer Science and Engineering, Galgotias University for their support in completing this project on time.

We also express gratitude towards our parents for their kind co-operation and encouragement which help me in completion of this project. My thanks and appreciations also go to our friends in developing the project and people who have willingly helped me out with their abilities.

Shivansh Bajpai
163101696
16SCSE101044

Abstract

The main aim of developing this application is to supply the medicines all over the country by just a single click and to reduce the time consumption. Online pharmacy is a web-based application. The user can post requirement for medicine. The User can purchase medicine online. Medicine is provided at your doorstep by the nearest associate store. The prescription is mandatory for ordering medicine. As per the prescription, the user can search medicine and useful information. This application provides information for daily consumption of medicine. This application provides user login to the customer. And admin can get the all expired medicines information and he can able to see all orders information of clients. It's main aim is just to reduce squandering of time and effort that customer puts into this process of purchasing medicine .

After seeing everyone's common problem regarding the medicine world . Everyone's way of tackling the problem the similar way of throwing away their priceless time and energy in getting medicine for their family members or for themselves was just a cry for help . So I decided to make things easier for everyone by providing the medicine at their doorstep by just one click away

List of Tables

Table Name: customer
Primary Key: email

Field Name	Data Type	Size
name	Varchar	45
age	Integer	22
email	Varchar	45
password	Varchar	45
mobile	Varchar	45
address	Varchar	45
City	Varchar	45
state	Varchar	45
dob	Date	
gender	Varchar	15

Table Name: Employee
Primary Key: email

Field Name	Data Type	Size
password	varchar	50
Name	Varchar	50
age	Integer	
mobile	Varchar	50
Email	Varchar	50
field	Varchar	50
Gender	Varchar	50
college	Varchar	50
Address	Varchar	50
State	Varchar	50
City	Varchar	50
pan	Varchar	50
dob	Date	
aadhar	Varchar	50

Table Name: medicine
Primary Key: name

Field Name	Data Type	Size
name	Varchar	45
Quantity	Integer	22
price	Varchar	45
url	Varchar	100
mfg	date	
exp	date	
Prescription	Varchar	45

Table Name: complains
Primary Key: id(Auto Increment)

Field Name	Data Type	Size
name	Varchar	45
age	Integer	22
email	Varchar	45
description	Varchar	45
mobile	Varchar	45
gender	Varchar	45
id	Integer	

Table Name: jobs
Primary Key: id(Auto Increment)

Field Name	Data Type	Size
id	Integer	11
description	Varchar	100
designation	Varchar	45
experience	Integer	11
salary	Integer	22

Table Name: candidates
Primary Key: id(Auto Increment)
Foreign Key: job_id

Field Name	Data Type	Size
id	Integer	11
first_name	Varchar	50
last_name	Varchar	50
mobile	Varchar	50
gender	Varchar	50
email	Varchar	50
resume	LongBlob	
job_Id	Integer	11
Address	Varchar	50
State	Varchar	50
City	Varchar	50
experience	Integer	11

Chapter 1. Introduction

(i) Overall View – The customer goes to the shop and purchases the medicine required. So a lot of time is wasted and the person gets tired. If he wants to exchange the product, once again he goes to the shop and replaces them. The complete process depends on the physical interactions . This whole process involves lots of effort and loss of time so to make everything easier and comforting our website provides facility to get medicine at your very doorstep .

Our Application provides the following things:

●To provide login facility to the customers.

●To provide a list of all available medicines to the customers.

●To update all the list of expired medicines.

●Medicines are delivered at your door-step

(ii) Purpose – After seeing everyone's common problem regarding the medicine world . Everyone's way of tackling the problem the similar way of throwing away their priceless time and energy in getting medicine for their family members or for themselves was just a cry for help . So I decided to make things easier for everyone by providing the medicine at their doorstep by just one click away. To order medicine from this website a User should follow the some authentication process. He can able to logging into our

application by providing valid user name and password .after that user can able to search the medicines and he will order the required medicines information through online. The main purpose of admin is to check for stock of medicine and for expiry dates of respective medicines . He can able to add the new medicines information which are visible to the clients. And he can view the expired medicines information .and he can able to update the medicines status .he is behaves like the manufacturer of medicines and he will add the supplier details to the client . So to conclude from this the primary purpose of this project is to deliver medicine at customer's doorstep.

(iii) Motivation and Scope – For buying medicine the customer goes to the shop and purchases the medicine required. So a lot of time is wasted and the person gets tired. If he wants to exchange the product, once again he goes to the shop and replaces them. So after considering this process we can say that the the prime concern of this project is to work on customer's ease of achieving the medicine in the most comfy way as possible.. The business goal for the application is to provide the medicines to all the people & admin will provide the supplier details . There are already some existing companies that work on this idea .

Scope of this project is large if we consider the examples of some already built companies like Medlife . Considering what this project has to offer scope is very wide open to build an empire It is better to order online than to go by yourself and purchase medicine that too sometimes under harsh weather .

Chapter 2. Literature Survey

- The Application of IT in the hospital pharmacy management

   Authors -Tie Hong, Man Dong

   Conclusion - The introduction of information technology in hospital makes the management for drugs using in accordance with the permission more simple and reliable especially in the narcotic drugs, psychotropic drugs, and medicare drugs using. The information technology makes us master the whole hospital drug use situation and achieve the goal of effective monitoring and scientific supervision on the clinical medication.

- Mobile Application for checking the status of stock availability

   Authors -N Ambreen Kubra, N Brundha

   Conclusion - The term mobile application is a way by which computer program is developed to be implemented on smart handled devices and andriod phones and so on. To develop an pharmacy application the android platform is used. This application will facilitate the clientele to get access to the medicine without walking through every pharmacy in the tracked location.

- Conversational bot for pharmacy

Authors -Nur Syahirah Ahmad, Man Dong

Conclusion - The chatbot is able to advice the types of medications to be taken based on the information provided by the customer. The development of the application is able to improve the communication process between the customers and the pharmacy and at the same time helps the pharmacy to have a better customer management system.

Chapter 3. System Design
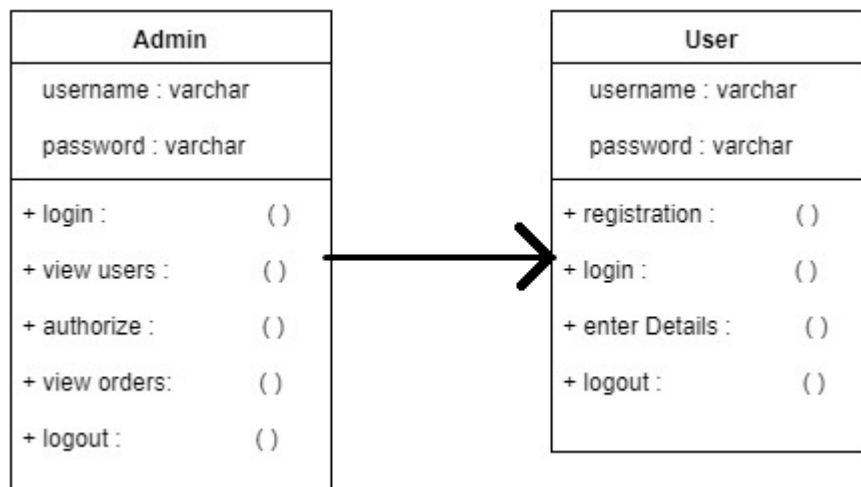
System Design

The most creative and challenging face of the system development is System Design. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Design goes through the logical and physical stages of development.

In designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fulfill. The first step is to determine how the output is to be produced and in what format. Second, input data and master files have to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing.
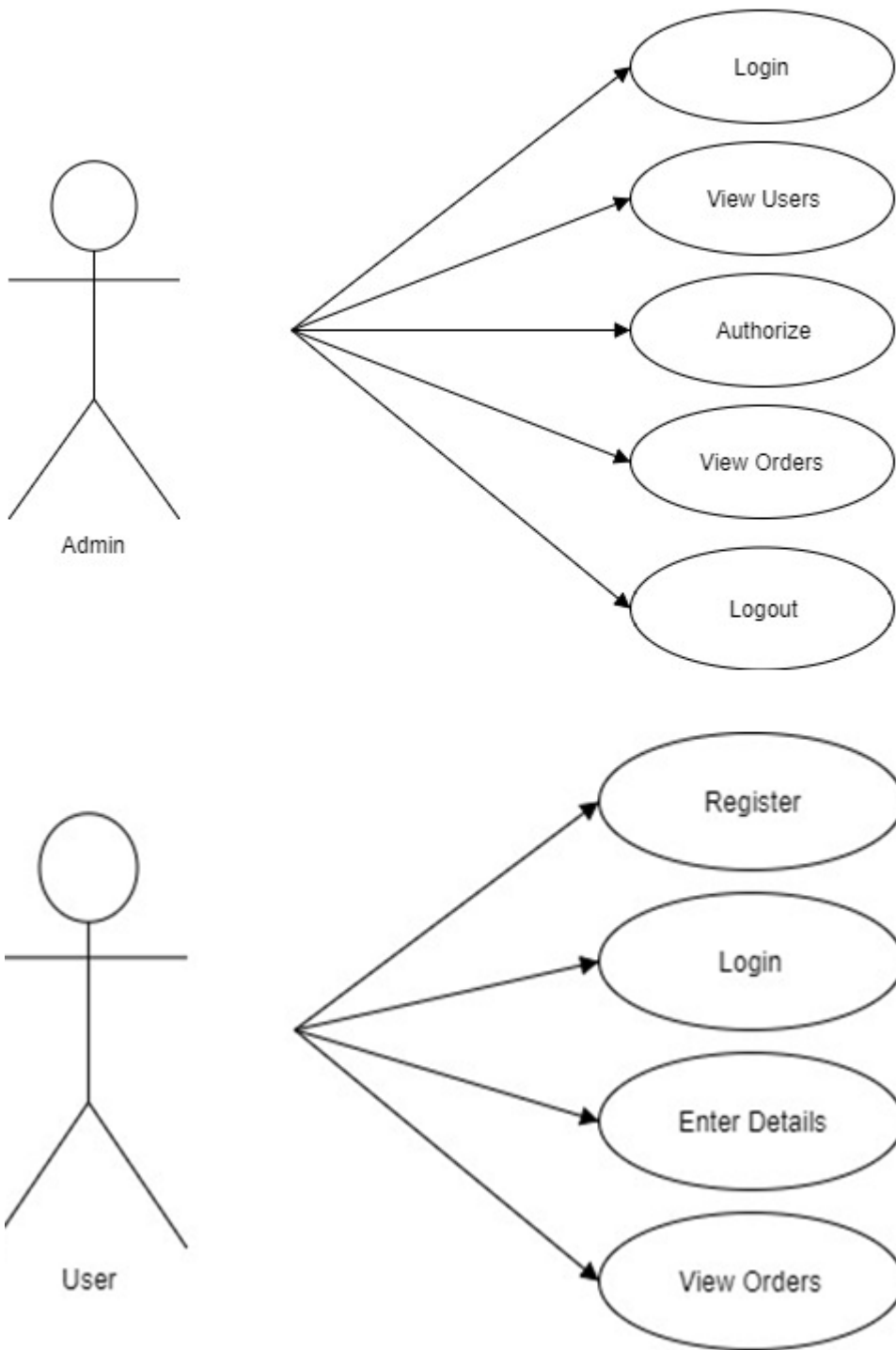
Design of a system can be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Thus system design is a solution to "how to" approach to the creation of a new system. Thus important phase provides the understanding and the procedural details necessary for implementing the system recommended in the feasibility study. The design step provides a data design, architectural design, and a procedural design.
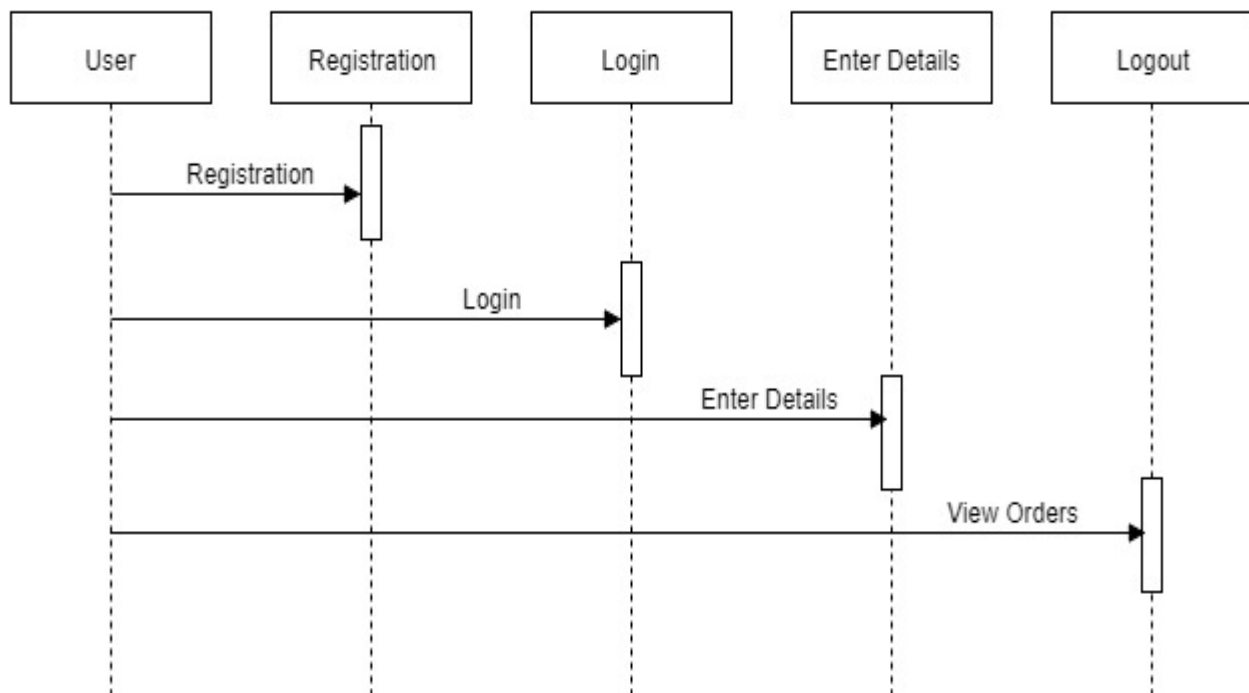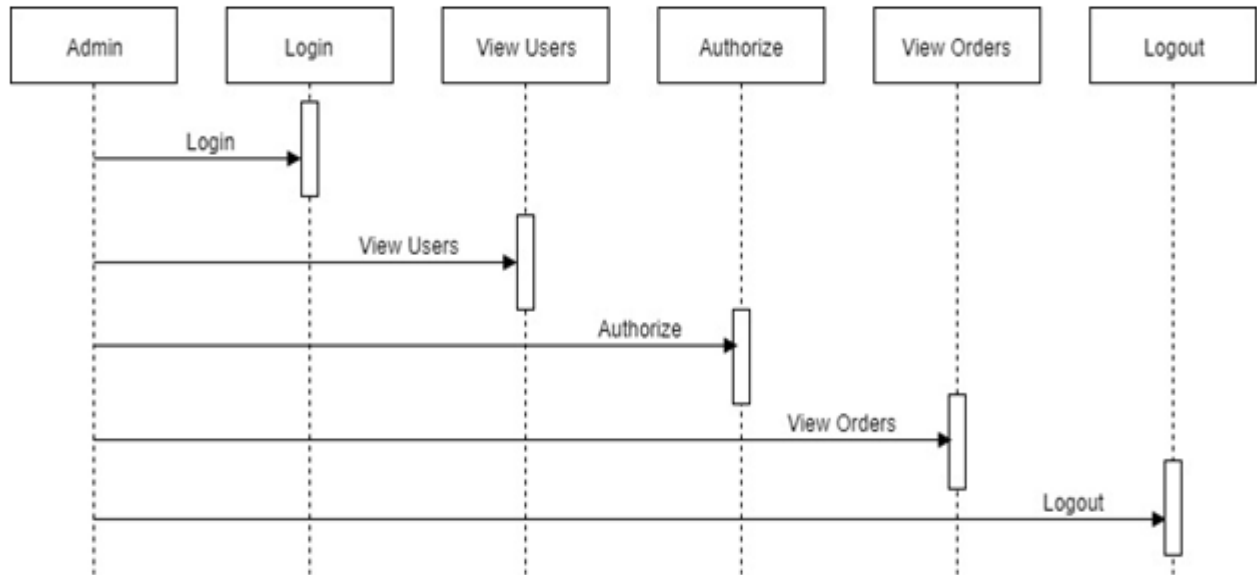
Logical Design

Logical data design is about the logically implied data. Each and every data in the form can be designed in such a manner to understand the meaning. Logical data 5 designing should give a clear understanding and idea about the related data used to construct a form.
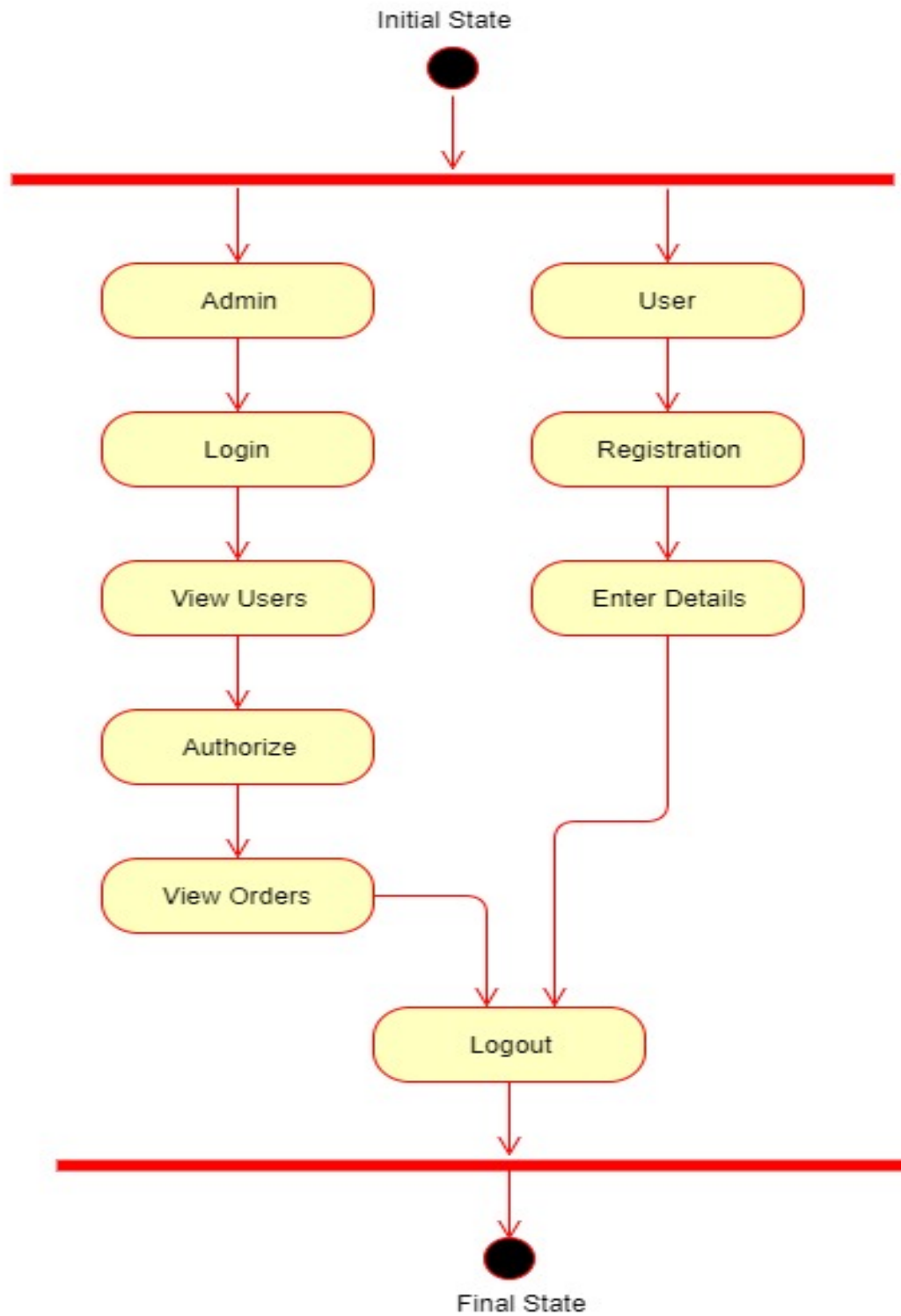
Admin	
username : varchar	
password : varchar	
+ login :	( )
+ view users :	( )
+ authorize :	( )
+ view orders:	( )
+ logout :	( )

User	
username : varchar	
password : varchar	
+ registration :	( )
+ login :	( )
+ enter Details :	( )
+ logout :	( )

Use Case Diagram

Admin

- Login
- View Users
- Authorize
- View Orders
- Logout

User

- Register
- Login
- Enter Details
- View Orders

Sequence Diagram

Admin	Login	View Users	Authorize	View Orders	Logout

Login

View Users

Authorize

View Orders

Logout

User	Registration	Login	Enter Details	Logout

Registration

Login

Enter Details

View Orders

Activity Diagram

Initial State

Admin

User

Login

Registration

View Users

Enter Details

Authorize

View Orders

Logout

Final State

Chapter 4. Requirements

System Requirements

SOFTWARE REQUIREMENTS
• Operating System : Windows.
• Server: Tomcat Apache.
• Platform: J2EE(servlet/jsp).
• Programming Language: java
• Database: Oracle DB Server/MySQL.
• Web Technologies: HTML/CSS/JS/BOOTSTRAP

Hardware Requirements
• Minimum 500GB Hard Disk
• Processor: Pentium series and above
• RAM: 512MB

Softwares Required

Netbeans

NetBeans is an integrated development environment (IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called *modules*. NetBeans runs on Windows, macOS, Linux and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5, and Javascript. Applications based on NetBeans, including the NetBeans IDE, can be extended by third party developers.

Platform

The NetBeans Platform is a framework for simplifying the development of Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications; no additional SDK is required.

Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

•User interface management (e.g. menus and toolbars)

•User settings management

•Storage management (carries out efficient storage)

•Window management

•Wizard framework (supports step-by-step dialogs)

•NetBeans Visual Library

•Integrated development tools

NetBeans IDE is an open-source integrated development environment. NetBeans IDE supports development of all Java application types (Java SE(including JavaFX), Java ME, web, EJB and mobile applications) out of the box. Among other features are an ANT-based project system, MAVEN support, refractorings, version control (supporting CVS, Subversion, Git, Mercurial and Clearcase).

Modularity: All the functions of the IDE are provided by modules. Each module provides a well-defined function, such as support for the Java language, editing, or support for the CVS versioning system, and SVN. NetBeans contains all the modules needed for Java development in a single download, allowing the user to start working immediately. Modules also allow NetBeans to be extended. New features, such as support for other programming languages, can be added by installing additional modules. For instance, Sun Studio, Sun Java Studio Enterprise, and Sun Java Studio Creator from Sun Micro-systems are all based on the NetBeans IDE.

The NetBeans Profiler is a tool for the monitoring of Java applications: It helps developers find memory leaks and optimize speed. Formerly downloaded separately, it is integrated into the core IDE since version 6.0. The Profiler is based on a Sun Laboratories research project that was named JFluid. That research uncovered specific techniques that can be used to lower the overhead of profiling a Java application. One of those techniques is dynamic bytecode instrumentation, which is particularly useful for profiling large Java applications. Using dynamic bytecode instrumentation and additional algorithms, the NetBeans Profiler is able to obtain runtime information on applications that are too large or complex for other profilers. NetBeans also support

Profiling Points that let you profile precise points of execution and measure execution time.

The **NetBeans IDE Bundle for Web & Java EE** provides complete tools for all the latest Java EE 6 standards, including the new Java EE 6 Web Profile, Enterprise Java Beans (EJBs), servlets, Java Persistence API, web services, and annotations. NetBeans also supports the JSF 2.0 (Facelets), JavaServer Pages (JSP), Hibernate, Spring, and Struts frameworks, and the Java EE 5 and J2EE 1.4 platforms. It includes Glassfish and Apache Tomcat.

Some of its features with Java EE include:

•Improved support for CDI, REST services and Java Persistence

•New support for Bean Validation

•Support for JSF component libraries, including bundled PrimeFaces library

•Improved editing for Expression Language in JSF, including code completion, refactoring and hints

NetBeans IDE offers first-class tools for Java web, enterprise, desktop, and mobile application development. It is consistently the first IDE to support the latest versions of the JDK, Java EE, and JavaFX. It provides smart overviews to help you understand and manage your applications, including ouf-of-the-box support for popular technologies such as Maven.

With its end-to-end application development features, constantly improving Java Editor, and continual speed and performance enhancements, NetBeans IDE sets the standard for application development with cutting edge technologies out of the box.
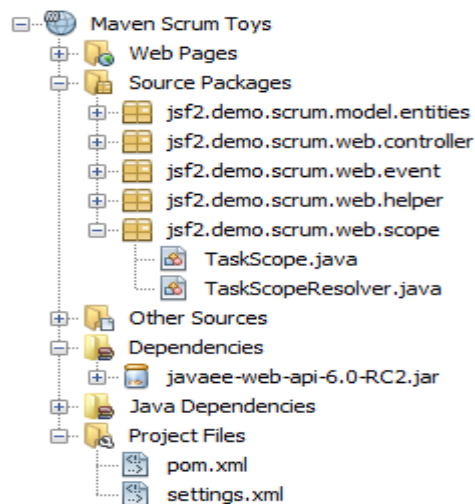
Apache Tomcat

Apache Tomcat (sometimes simply "Tomcat") is an open-source implementation of the Java Servlet, Java Server Pages, Java Expression Language and WebSocket technologies.Tomcat provides a "pure Java" HTTP Web Server environment in which Java code can run.

Components

Tomcat 4.x was released with Catalina (a servlet container), Coyote (an HTTP connector) and Jasper (a JSP Engine).

Catalina

Catalina is Tomcat's servlet Container. Catalina implements Sun Microsystems specifications for servlet and JavaServer Pages (JSP). In Tomcat, a Realm element represents a "database" of usernames, passwords, and roles (similar to Unix groups)

assigned to those users. Different implementations of Realm allow Catalina to be integrated into environments where such authentication information is already being created and maintained, and then use that information to implement Container Managed Security as described in the Servlet Specification.

Coyote

Coyote is a Connector component for Tomcat that supports the HTTP 1.1 protocol as a web server. This allows Catalina, nominally a Java Servlet or JSP container, to also act as a plain web server that serves local files as HTTP documents.Coyote listens for incoming connections to the server on a specific TCP port and forwards the request to the Tomcat Engine to process the request and send back a response to the requesting client. Another Coyote Connector, Coyote JK, listens similarly but instead forwards its requests to another web server, such as Apache, using the JK Protocol. This usually offers better performance.

Jasper

Jasper is Tomcat's JSP Engine. Jasper parses JSP files to compile them into Java code as servlets (that can be handled by Catalina). At runtime, Jasper detects changes to JSP files and recompiles them.

As of version 5, Tomcat uses Jasper 2, which is an implementation of the Sun Microsystems' JSP 2.0 specification. From Jasper to Jasper 2, important features were added:

•JSP Tag library pooling – Each tag markup in JSP file is handled by a tag handler class. Tag handler class objects can be pooled and reused in the whole JSP servlet.

•Background JSP compilation – While recompiling modified JSP Java code, the older version is still available for server requests. The older JSP servlet is deleted once the new JSP servlet has finished being recompiled.

•Recompile JSP when included page changes – pages can be inserted and included into a JSP at runtime. The JSP will not only be recompiled with JSP file changes but also with included page changes.

•JDT Java compiler – Jasper 2 can use the Eclipse JDT (Java Development Tools) Java compiler instead of Ant and javac

Three new components were added with the release of Tomcat 7:

Cluster

This component has been added to manage large applications. It is used for load balancing that can be achieved through many techniques. Clustering support currently requires the JDK version 1.5 or higher.

High availability

A high-availability feature has been added to facilitate the scheduling of system upgrades (e.g. new releases, change requests) without affecting the live environment. This is done by dispatching live traffic requests to a temporary server on a different port while the main server is upgraded on the main port. It is very useful in handling user requests on high-traffic web applications.

Web application

It has also added user— as well as system-based web applications enhancement to add support for deployment across the variety of environments. It also tries to manage sessions as well as applications across the network.

Tomcat is building additional components. A number of additional components may be used with Apache Tomcat. These components may be built by users should they need them or they can be downloaded from one of the mirrors.

Tomcat 7.x implements the Servlet 3.0 and JSP 2.2 specifications. It requires Java version 1.6, although previous versions have run on Java 1.1 through 1.5. Versions 5 through 6 saw improvements in garbage collection, JSP parsing, performance and scalability. Native wrappers, known as "Tomcat Native", are available for Microsoft windows and Unix for platform integration.

Tomcat 8.x implements the Servlet 3.1 and JSP 2.3 Specifications. Apache Tomcat 8.5.x is intended to replace 8.0.x and includes new features pulled forward from Tomcat 9.0.x. The minimum Java version and implemented specification versions remain unchanged.

The Java ecosystem supports several kinds of application server, so let's disambiguate them and see where Tomcat fits in:

•A **servlet container** is an implementation of the Java Servlet specification, used primarily for hosting Java servlets.

•A **web server** is a server designed to serve files from the local system, like Apache.

•A **Java enterprise application server** is a full-blown implementation of the Java EE (now Jakarta Enterprise) specification.

At heart, Tomcat is a servlet and JSP container. A java Servlet encapsulates code and business logic and defines how requests and responses should be handled in a Java server. JSP is a server-side view rendering technology. As the developer, you write the servlet or JSP page, then let Tomcat handle the routing.

Tomcat also contains the Coyote engine, which is a web server. Thanks to Coyote, it's possible to extend Tomcat to include a variety of Java enterprise specs and capabilities, including the Java Persistence Application. Tomcat also has an extended version, called Tom EE, that includes more enterprise features. I'll briefly introduce TomEE later in this article.

Chapter 5. Implementation

Frontend

HTML

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

HTML is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning HTML:

- **Create Web site** - You can create a website or customize an existing web template if you know HTML well.

- **Become a web designer** - If you want to start a carrer as a professional web designer, HTML and CSS designing is a must skill.

- **Understand web** - If you want to optimize your website, to boost its speed and performance, it is good to know HTML to yield best results.

- **Learn other languages** - Once you understands the basic of HTML then other related technologies like javascript, php, or angular are become easier to understand.

```
<!DOCTYPE html>
<html>
   <head>
      <title>This is document title</title>
   </head>
   <body>
      <h1>This is a heading</h1>
      <p>Hello World!</p>
   </body>
</html>
```

CSS

CSS is used to control the style of a web document in a simple and easy way.CSS is the acronym for **"Cascading Style Sheet"**. This tutorial covers both the versions CSS1,CSS2 and CSS3, and gives a complete understanding of CSS, starting from its basics to advanced concepts.

CSS is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning CSS:

- **Create Stunning Web site** - CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.

- **Become a web designer** - If you want to start a carrer as a professional web

designer, HTML and CSS designing is a must skill.

- **Control web** - CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

- **Learn other languages** - Once you understands the basic of HTML and CSS then other related technologies like javascript, php, or angular are become easier to understand.

```html
<!DOCTYPE html>
<html>
   <head>
      <title>This is document title</title>
      <style>
      h1 {
         color: #36CFFF;
      }
      </style>
   </head>
   <body>
      <h1>Hello World!</h1>
   </body>
</html>
```

JavaScript

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

Javascript is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning Javascript:

- Javascript is everywhere, it comes installed on every modern web browser and so to learn Javascript you really do not need any special environment setup. For example Chrome, Mozilla Firefox , Safari and every browser you know as of today, supports Javascript.

- Javascript helps you create really beautiful and crazy fast websites. You can develop your website with a console like look and feel and give your users the best Graphical User Experience.

- JavaScript usage has now extended to mobile app development, desktop app development, and game development. This opens many opportunities for you as Javascript Programmer.

- Due to high demand, there is tons of job growth and high pay for those who know JavaScript. You can navigate over to different job sites to see what having JavaScript skills looks like in the job market.

- Great thing about Javascript is that you will find tons of frameworks and Libraries already developed which can be used directly in your software development to reduce your time to market.

\<html\>

```
   <body>
      <script language = "javascript" type = "text/javascript">
         <!--
            document.write("Hello World!")
         //-->
      </script>
   </body>
</html>
```

Bootstrap

Bootstrap is the most popular front end framework in the recent time. It is sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development. It uses HTML, CSS and Javascript.

Here are some listed applications of bootstrap

- Scaffolding − Bootstrap provides a basic structure with Grid System, link styles, and background. This is is covered in detail in the section Bootstrap Basic Structure

- CSS − Bootstrap comes with the feature of global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system. This is covered in detail in the section Bootstrap with CSS.

- Components − Bootstrap contains over a dozen reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more. This is covered in detail in the section Layout Components.

- JavaScript Plugins − Bootstrap contains over a dozen custom jQuery plugins. You can easily include them all, or one by one. This is covered in details in the section Bootstrap Plugins.

- Customize − You can customize Bootstrap's components, LESS variables, and jQuery plugins to get your very own version

```
<!DOCTYPE html>
<html>
```

```html
  <head>
    <title>Bootstrap 101 Template</title>
    <meta name = "viewport" content = "width = device-width, initial-scale =
1.0">

    <!-- Bootstrap -->
    <link href = "css/bootstrap.min.css" rel = "stylesheet">

    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media
queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

    <!--[if lt IE 9]>
    <script src =
"https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
    <script src =
"https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>
    <![endif]-->

  </head>

  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src = "https://code.jquery.com/jquery.js"></script>

    <!-- Include all compiled plugins (below), or include individual files as
needed -->
    <script src = "js/bootstrap.min.js"></script>

  </body>
</html>
```

Jquery

jQuery is a fast and concise JavaScript library created by John Resig in 2006. jQuery

simplifies HTML document traversing, event handling, animating, and Ajax interactions

for Rapid Web Development.

- jQuery is a small and lightweight JavaScript library.
- jQuery is cross-platform.
- jQuery means "write less do more".
- jQuery simplifies AJAX call and DOM manipulation.

Backend

JSP

Adding dynamic content via expressions

As we saw in the previous section, any HTML file can be turned into a JSP file by changing its extension to .jsp. Of course, what makes JSP useful is the ability to embed Java. Put the following text in a file with .jsp extension (let us call it **hello.jsp**), place it in your JSP directory, and view it in a browser.

```
<HTML>
<BODY>
Hello!  The time is now <%= new java.util.Date() %>
</BODY>
</HTML>
```
Notice that each time you reload the page in the browser, it comes up with the current time.

The character sequences <%= and %> enclose Java expressions, which are evaluated at run time.

This is what makes it possible to use JSP to generate dyamic HTML pages that change in response to user actions or vary from user to user.

Scriptlets

We have already seen how to embed Java expressions in JSP pages by putting them between the **<%=** and **%>** character sequences.

But it is difficult to do much programming just by putting Java expressions inside HTML.

JSP also allows you to write blocks of Java code inside the JSP. You do this by placing your Java code between **<%** and **%>** characters (just like expressions, but without the = sign at the start of the sequence.)

This block of code is known as a "scriptlet". By itself, a scriptlet doesn't contribute any HTML (though it can, as we will see down below.) A scriptlet contains Java code that is executed every time the JSP is invoked.

Here is a modified version of our JSP from previous section, adding in a scriptlet.

```
<HTML>
<BODY>
<%
    // This is a scriptlet.  Notice that the "date"
    // variable we declare here is available in the
    // embedded expression later on.
    System.out.println( "Evaluating date now" );
    java.util.Date date = new java.util.Date();
%>
Hello! The time is now <%= date %>
</BODY>
</HTML>
```

If you run the above example, you will notice the output from the "System.out.println" on the server log. This is a convenient way to do simple debugging (some servers also have techniques of debugging the JSP in the IDE. See your server's documentation to see if it offers such a technique.)

By itself a scriptlet does not generate HTML. If a scriptlet wants to generate HTML, it can use a variable called "out". This variable does not need to be declared. It is already predefined for scriptlets, along with some other variables. The following example shows how the scriptlet can generate HTML output.

```
<HTML>
<BODY>
<%
   // This scriptlet declares and initializes "date"
   System.out.println( "Evaluating date now" );
   java.util.Date date = new java.util.Date();
%>
Hello!  The time is now
<%
   // This scriptlet generates HTML output
   out.println( String.valueOf( date ));
%>
</BODY>
</HTML>
```

Here, instead of using an expression, we are generating the HTML directly by printing to the "out" variable. The "out" variable is of type javax.servlet.jsp.JspWriter.

Another very useful pre-defined variable is "request". It is of type javax.servlet.http.HttpServletRequest

A "request" in server-side processing refers to the transaction between a browser and the server. When someone clicks or enters a URL, the browser sends a "request" to the server for that URL, and shows the data returned. As a part of this "request", various data is available, including the file the browser wants from the server, and if the request is coming from pressing a SUBMIT button, the information the user has entered in the form fields.

The JSP "request" variable is used to obtain information from the request as sent by the browser. For instance, you can find out the name of the client's host (if available, otherwise the IP address will be returned.) Let us modify the code as shown:

```
<HTML>
<BODY>
<%
```

```
   // This scriptlet declares and initializes "date"
   System.out.println( "Evaluating date now" );
   java.util.Date date = new java.util.Date();
%>
Hello!  The time is now
<%
   out.println( date );
   out.println( "<BR>Your machine's address is " );
   out.println( request.getRemoteHost());
%>
</BODY>
</HTML>
```

A similar variable is "response". This can be used to affect the response being sent to the browser. For instance, you can call response.sendRedirect(anotherUrl); to send a response to the browser that it should load a different URL. This response will actualy go all the way to the browser. The browser will then send a different request, to "anotherUrl". This is a little different from some other JSP mechanisms we will come across, for including another page or forwarding the browser to another page.

Servlet

Servlet technology is used to create a web application (resides at server side and generates a dynamic web page).

Servlet technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. However, there were many disadvantages to this technology. We have discussed these disadvantages below.

There are many interfaces and classes in the Servlet API such as Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, etc.

Servlet can be described in many ways, depending on the context.

- Servlet is a technology which is used to create a web application.

- Servlet is an API that provides many interfaces and classes including documentation.

- Servlet is an interface that must be implemented for creating any Servlet.

- Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.

- Servlet is a web component that is deployed on the server to create a dynamic web page.

Servlet Interface
Servlet interface provides common behavior to all the servlets.Servlet interface defines methods that all servlets must implement.

Servlet interface needs to be implemented for creating any servlet (either directly or indirectly). It provides 3 life cycle methods that are used to initialize the servlet, to service the requests, and to destroy the servlet and 2 non-life cycle methods.

```
import java.io.*;
import javax.servlet.*;

public class First implements Servlet{
```

```
ServletConfig config=null;

public void init(ServletConfig config){
this.config=config;
System.out.println("servlet is initialized");
}

public void service(ServletRequest req,ServletResponse res)
throws IOException,ServletException{

res.setContentType("text/html");

PrintWriter out=res.getWriter();
out.print("<html><body>");
out.print("<b>hello simple servlet</b>");
out.print("</body></html>");

}
public void destroy(){System.out.println("servlet is destroyed");}
public ServletConfig getServletConfig(){return config;}
public String getServletInfo(){return "copyright 2007-1010";}

}
```

Generic Servlet

GenericServlet class implements Servlet, ServletConfig and Serializable interfaces. It
provides the implementation of all the methods of these interfaces except the service
method.

GenericServlet class can handle any type of request so it is protocol-independent.

```
import java.io.*;
import javax.servlet.*;

public class First extends GenericServlet{
public void service(ServletRequest req,ServletResponse res)
throws IOException,ServletException{

res.setContentType("text/html");

PrintWriter out=res.getWriter();
out.print("<html><body>");
out.print("<b>hello generic servlet</b>");
out.print("</body></html>");
```

```
}
}
```

HttpServlet

The HttpServlet class extends the GenericServlet class and implements Serializable interface. It provides http specific methods such as doGet, doPost, doHead, doTrace etc.

```
import javax.servlet.*;
import java.io.*;
public class DemoServ extends HttpServlet{
public void doGet(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException
{
res.setContentType("text/html");
PrintWriter pw=res.getWriter();
String name=req.getParameter("name");
pw.println("Welcome "+name);
}}
```

MySql

My SQL is an application used to create computer databases for the Microsoft Windows family of server operating systems. It provides an environment used to generate databases that can be accessed from workstations, the web, or other media such as a personal digital assistant (PDA). MY SQL is probably the most accessible and the most documented enterprise database environment right now. This also means that you can learn it a little quicker than most other database environments on the market

What is SQL Used for:

Using SQL one can create and maintain data manipulation objects such as table, views, sequence etc. These data manipulation objects will be created and stored on the server's hard disk drive, in a tablespace, to which the user has been assigned.

Once these data manipulation objects are created, they are used extensively in commercial applications.

DML, DCL, DDL:

In addition to the creation of data manipulation objects, the actual manipulation of data within these objects is done using SQL.

The SQL sentences that are used to create these objects are called DDL's or Data Definition Language. The SQL sentences used to manipulate data within these objects are called DML's or Data Manipulation Language. The SQL sentences, which are used to control the behavior of these objects, are called DCL's or Data Control Language.

Data Type

Built-in data types

In My Sql, each object (such as column, variable, or parameter) has a related data type, which is an attribute that specifies the type of data that the object can hold.

User-defined data types

My SQL supports user-defined data types too. User-defined data types provide a mechanism for applying a name to a data type that is more descriptive of the types of values to be held in the object. Using user-defined data type can make it easier for a programmer or database administrator to understand the intended use of any object defined with the data type. The user-defined data types are based on the system data types and can be used to predefine several attributes of a column, such as its data type, length, and whether it supports NULL values. To create a user-defined data type, you

can use the sp_addtype system stored procedure or you could add one using the Enterprise Manager. When you create a user-defined data type, you should specify the following three properties:

- Data type's name.
- Built-in data type upon which the new data type is based.
- Whether it can contain NULL values.

Hibernate

Hibernate is a Java framework that simplifies the development of Java application to interact with the database. It is an open source, lightweight, ORM (Object Relational Mapping) tool. Hibernate implements the specifications of JPA (Java Persistence API) for data persistence.

An ORM tool simplifies the data creation, data manipulation and data access. It is a programming technique that maps the object to the data stored in the database.

Java Persistence API (JPA) is a Java specification that provides certain functionality and standard to ORM tools. The javax.persistence package contains the JPA classes and interfaces.

Some advantages of framework are listed below:

- Hibernate framework is open source under the LGPL license and lightweight.

- The performance of hibernate framework is fast because cache is internally used in hibernate framework. There are two types of cache in hibernate framework first

level cache and second level cache. First level cache is enabled by default.

- HQL (Hibernate Query Language) is the object-oriented version of SQL. It generates the database independent queries. So you don't need to write database specific queries. Before Hibernate, if database is changed for the project, we need to change the SQL query as well that leads to the maintenance problem.

- Hibernate framework provides the facility to create the tables of the database automatically. So there is no need to create tables in the database manually.

- Fetching data from multiple tables is easy in hibernate framework.

- Hibernate supports Query cache and provide statistics about query and database status.

HQL
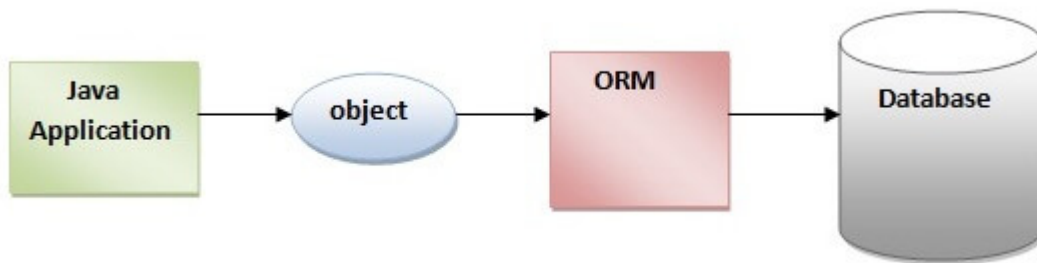
Hibernate Query Language (HQL) is same as SQL (Structured Query Language) but it doesn't depends on the table of the database. Instead of table name, we use class name in HQL. So it is database independent query language.

There are many advantages of HQL. They are as follows:

- database independent
- supports polymorphic queries
- easy to learn for Java Programmer

Architecture
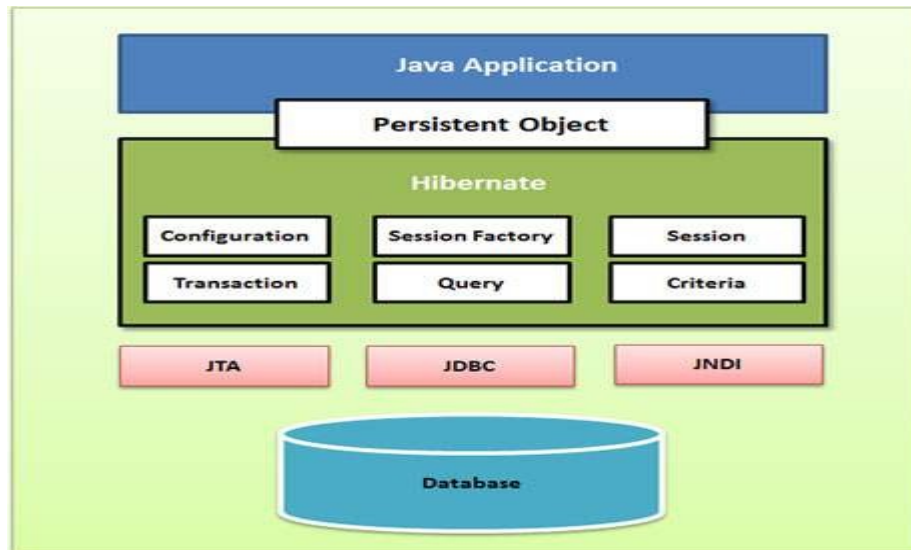
Configuration Object

The Configuration object is the first Hibernate object you create in any Hibernate application. It is usually created only once during application initialization. It represents a configuration or properties file required by the Hibernate.

The Configuration object provides two keys components −

- Database Connection − This is handled through one or more configuration files supported by Hibernate. These files are hibernate.properties and hibernate.cfg.xml

- Class Mapping Setup − This component creates the connection between the Java classes and database tables.

SessionFactory Object

Configuration object is used to create a SessionFactory object which in turn configures Hibernate for the application using the supplied configuration file and allows for a Session object to be instantiated. The SessionFactory is a thread safe object and used by all the threads of an application.

The SessionFactory is a heavyweight object; it is usually created during application start up and kept for later use. You would need one SessionFactory object per database using a separate configuration file. So, if you are using multiple databases, then you would have to create multiple SessionFactory objects.

Session Object

A Session is used to get a physical connection with a database. The Session object is lightweight and designed to be instantiated each time an interaction is needed with the database. Persistent objects are saved and retrieved through a Session object.

The session objects should not be kept open for a long time because they are not usually thread safe and they should be created and destroyed them as needed.

Transaction Object

A Transaction represents a unit of work with the database and most of the RDBMS supports transaction functionality. Transactions in Hibernate are handled by an underlying transaction manager and transaction (from JDBC or JTA).

This is an optional object and Hibernate applications may choose not to use this interface, instead managing transactions in their own application code.

Query Object

Query objects use SQL or Hibernate Query Language (HQL) string to retrieve data from the database and create objects. A Query instance is used to bind query parameters, limit the number of results returned by the query, and finally to execute the query.

Criteria Object

Criteria objects are used to create and execute object oriented criteria queries to retrieve objects.

Chapter 6. Software Quality Attributes

RELIABILITY

Reliability of a system is a measure of the ability of a system to keep operating overtime.

It is typically measured as its mean time between failures (MTBF), expected type of system. Delivery of data to intended recipient (notification to sender by email or message).

Reliability service – notify user – if delivery fail.

Together TCP & IP provide reliable service.

Vaccinate has the ability to behave consistently in a user acceptable manner when operating within the environment for which the system was intended.

AVAILABILITY

It is apt that resources that should be available to authorized user actually are available (Safety & dependability).

A fault associate with availability is denial of service attack. Degree to which system is in specified operable and committable state at the start. It will operate satisfactorily at given pt.
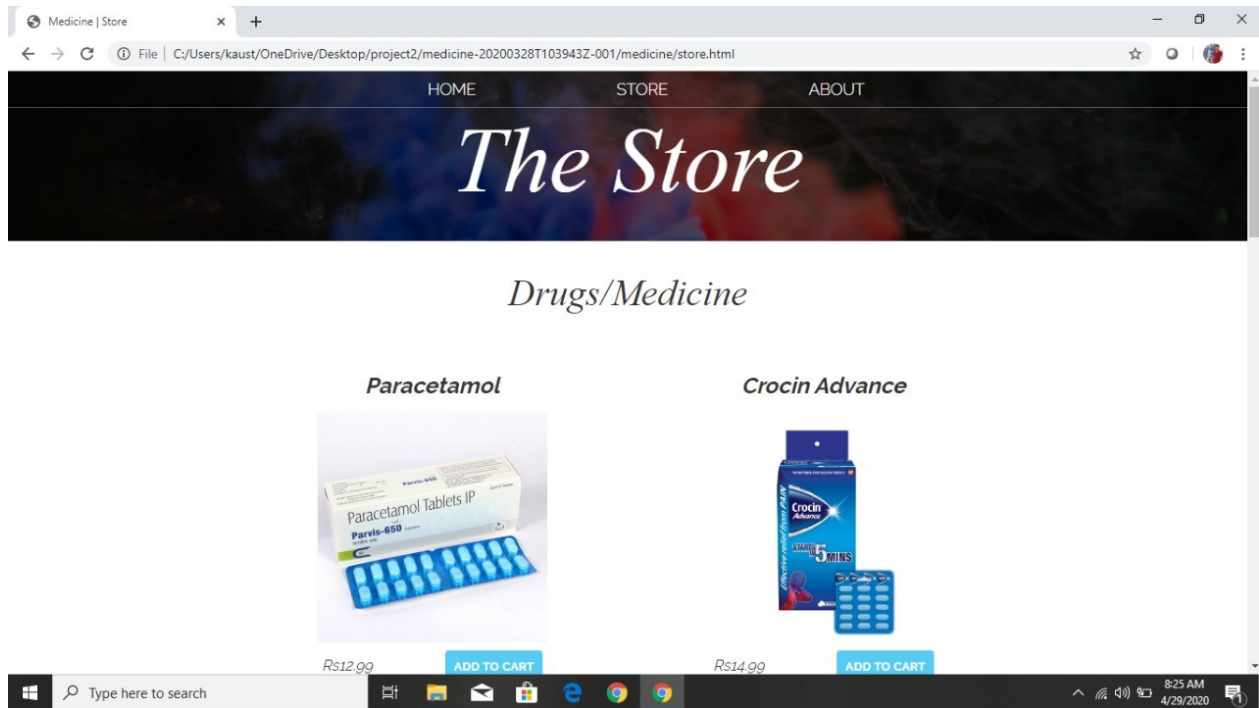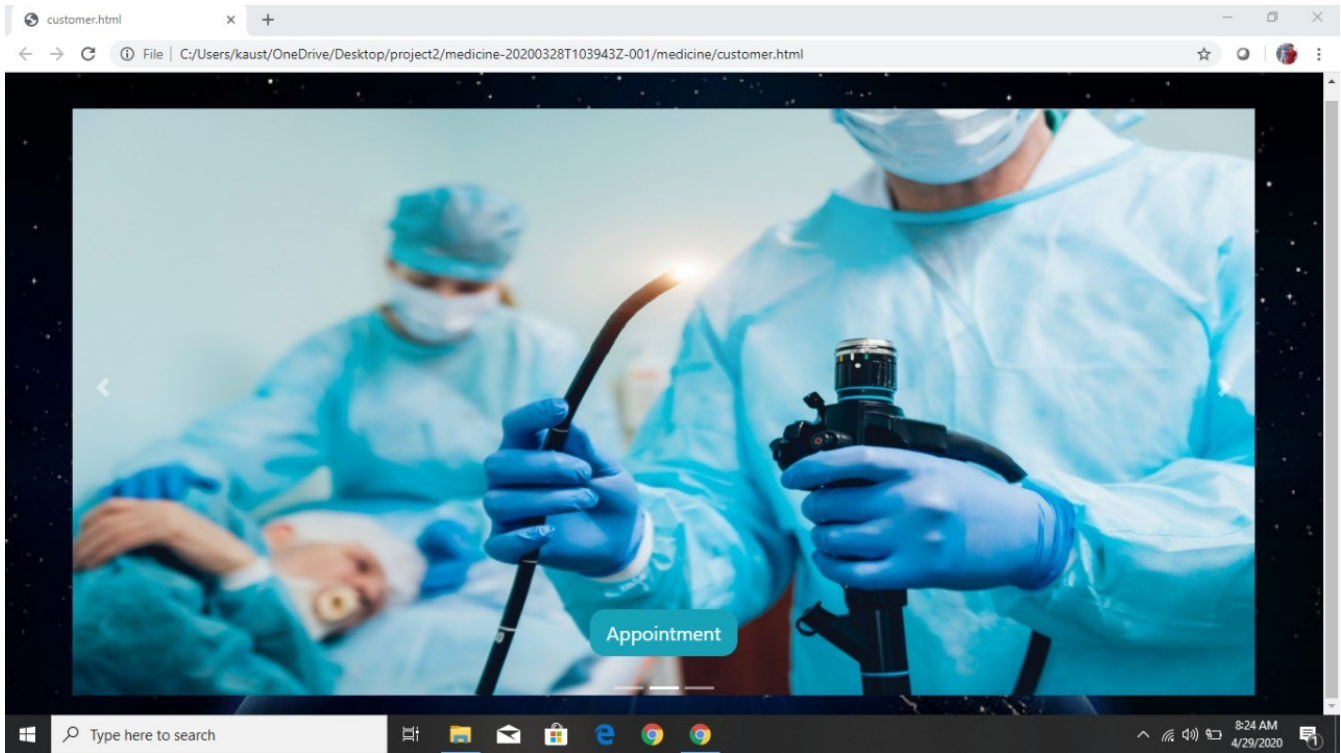
Chapter 7 . Conclusion

As we are well aware about the existing way of purchasing medicines, So to overcome these difficulties we are here with the solution of ordering medicines online. The main aim of developing this application is to supply the medicines all over the country by just a single click and to reduce the time consumption.It is always easier to receive medicine at your doorstep than to wait in line for some hours and waste your precious time in for doing some work that could be done just within matter of minutes.

Chapter 8. Output
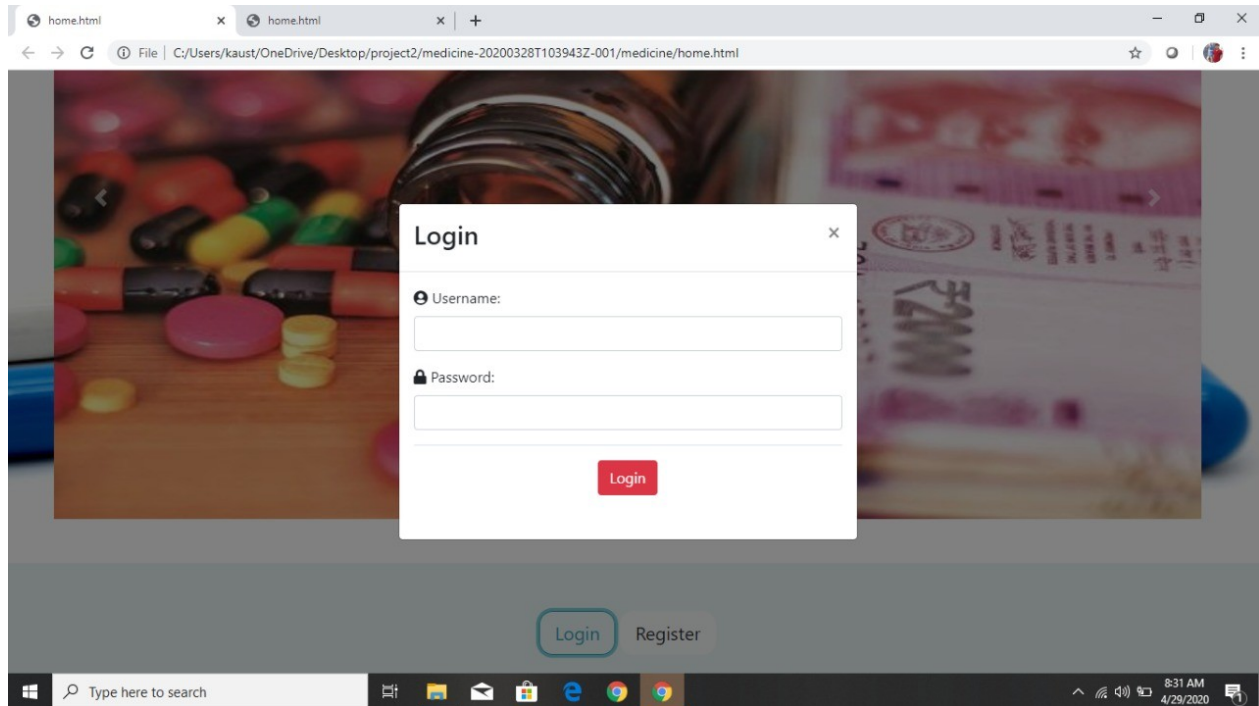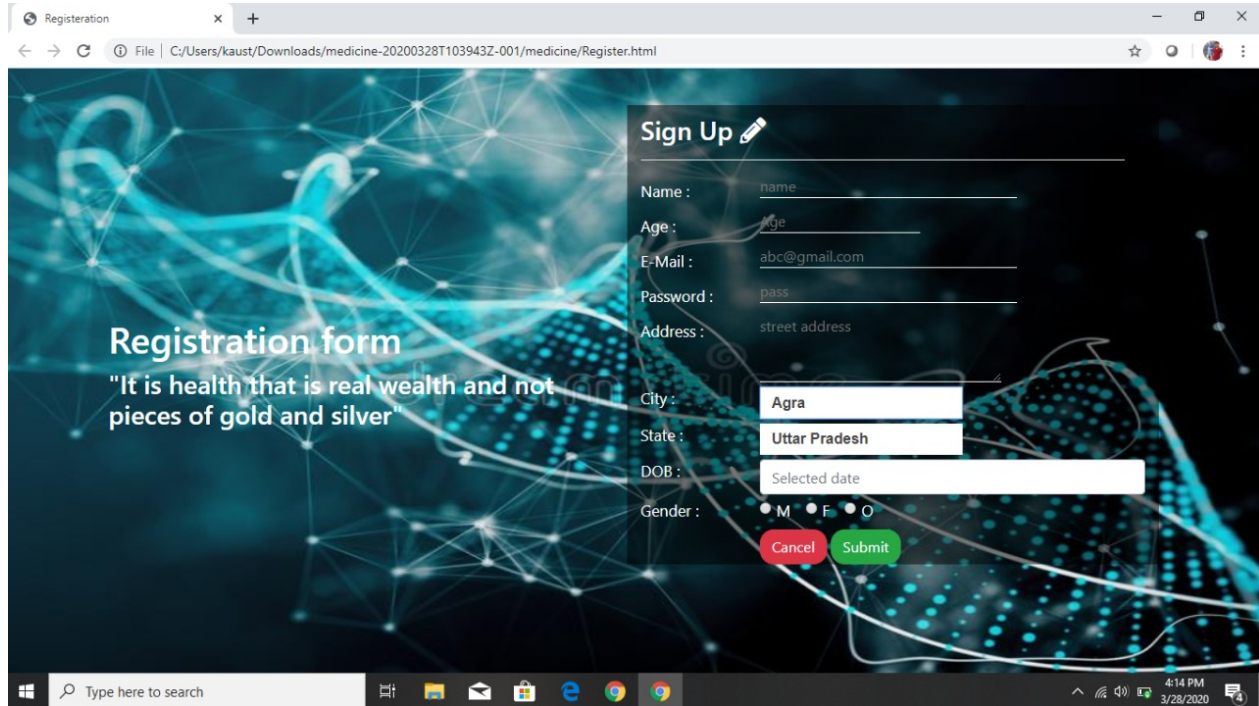
Chapter 9. References

1. Google
2. Head First Java
3. Web developer udemy course
4. https://www.tutorialspoint.com/servlets/index.html
5. https://www.tutorials.com/jsp/index.html
6. https://www.javatpoint.com/hibernate-tutorial