CONTAINER LEASE MANAGEMENT

Submitted in partial fulfillment of the requirements for the award of Bachelor of Computer Science Degree

By

MEENAKSHI.S (Reg. No. 39290066)

VISHNU PRIYA. K (Reg. No. 39290127)



DEPARTMENT OF COMPUTER SCIENCE SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119, TAMILNADU

APRIL 2022





SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY) Accredited with Grade "A" by NAAC (Established under Section 3 of UGC Act, 1956) JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119 www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **MEENAKSHI.S** (Reg. No. 39290066) and VISHNU PRIYA.K (Reg. No. 39290127) who carried out the project entitled "CONTAINER LEASE MANAGEMENT" under my supervision from December 2021 to April 2022

> Internal Guide Dr.Asha

Head of the Department

Dr. L.Lakshmanan, M.E., Ph.D.,

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

We, MEENAKSHI.S (Reg. No. 39290066), VISHNU PRIYA. K (Reg. No. 39290127) hereby declare that the Project Report entitled " CONTAINER LEASE MANAGEMENT" done by me under the guidance of Dr. Asha is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D**, **Dean**, School of Computing **Dr. L. Lakshmanan M.E., Ph.D.,** and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr. Asha for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department** of **Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Around the ports, there are usually plenty of container suppliers. So, if you want to buy a shipping container that's one place to look. You can also find containers through websites, who have regional networks to find containers you need. That way is just one way to trade containers. In this study, a dynamic programming method is brought forward to solve this problem based on the solution of a mix-integer linear programming sub-problem. This paper presents Design of Project Contract Management System Based on J2EE Architecture to solve the container management problems. The system design follows the principles of advanced, economical, reliable, high efficient, open, easy to operate and easy to maintenance. According to the characteristics of projects. On the basis of the feasibility study and requirement analysis, this paper discusses the system function design and the database design in detail. By the way, we carry out the system functions by using the J2EE integration architecture based on Web Work + SERVLET + JSP.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO.
	ABSTRACT	5
1	INTRODUCTION	10
	1.1 OVERVIEW OF PROJECT	10
2	LITERATURE SURVEY	11
	2.1 MODULUS	11
	2.2 MODULE DESCRIPTION	12
3	AIM AND SCOPE OF THE PRESENT INVESTIGATION	16
	3.1 AIM	16
	3.2 SCOPE	16
	3.3 SYSTEM REQUIREMENTS	16
	3.3.1 HARDWARE REQUIREMENTS	16
	3.3.2 SOFTWARE REQUIREMENTS	17
4	MATERIAL METHODS AND ALGORITHM USED	18

4.1 DESIGN ENGINEERING	19
4.2 STATE DIAGRAM	20
4.3 ACTIVITY DIAGRAM	21
4.4 CLASS DIAGRAM	22
4.5 SEQUENCE DIAGRAM	23
4.6 DATA – FLOW DIAGRAM	24
RESULT & DISCUSSION, PERFORMANCE ANALYSIS	26
5.1 SYSTEM ARCHITECTURE	26
5.2 IMPLEMETATION	27
5.2.1 INDEX jsp	29
5.2.2 OWNER LOGIN	35
5.2.3 OWNER REGISTER	41
5.2.4 OWNER MAIN PAGE	53
5.3 ADMIN ACTIVATE	54
5.4 ADMIN ACTIVATE	61
CONCLUSION	69

7 REFERENCE

CHAPTER 1

INTRODUCTION:

1.1 OVERVIEW OF PROJECT:

The container buying activities of the Project, although experimental in nature, can be considered by the admin as the initial step of the Container adding Program. The training and experience gained in the Project area will support progressive expansion of survey and registration activities throughout the containers. The container redistribution policy dimension of the Project is more limited, because of the technical complexity and political sensitivity of this question. The research to be undertaken will provide valuable information on the need for and mechanics of redistribution in the seller container. Implementation of a nationwide program will require serious politicaldecisions regarding the future structure of container.

CHAPTER 2

LITERATURE SURVEY

2.1 MODULES:

- ➢ Register
- Registration approve
- > Login
- > Add container
- Admin approve
- ➢ Container view
- User Request
- Owner Accept
- Sold detail

2.2 MOLDULE DESCRIPTION:

REGISTER:

In this module the new user enter web page the user need to update their details to

the page. The entered details directly stored in a database.

REGISTRATIONAPPROVE:

In this module in our project when the new user enter the web page. User register the web page. After registration the data forward to the admin. The admin review the user is valid user approve the user registration.

LOGIN:

This is module in our project, here symbolizes a unit of work performed within a database management system (or similar system) against a database, and treated in a coherent and reliable way independent of other transactions. A transaction generally represents any change in database user will transfer the amount to provider.

ADD CONTAINER:

In this module is used to help to the container owner add and upload the image ofselling container with the container longitude. It will be stored in database.

ADMIN APPROVE:

In this module the owner added the container to the data base. The container details show to the admin after admin accept the selling request then buyer will view the selling container to his pages.

VIEW CONTAINER:

In this module the owner and admin will view the seller added containerdetails. Admin will take the action to seller land.

USER REQUEST:

In this project what we are going to perform means, user view the file andthen add the user details then give the request to buying container.

OWNER ACCEPT:

In this module the user need to buy the container then give the request toowner. Then user check the details about owner and accept the user request.

SOLD DETAILS:

In this module the admin will view the all details about buyer and seller likecontainer related details.

2.1 MODULE DIAGRAM:

Register:



Admin Approve:



details

View container:



CHAPTER 3

AIM & SCOPE OF THE PRESENT INVESTIGATION

3.1 AIM

The aim of this project is to trade containers where a dynamic programming method is brought forward to solve this problem based on the solution of a mix integer linear programming sub problem. This project presents design of project contract management system based on J2EE architecture to solve the container management problems.

3.2 SCOPE

This project helps to propose a seamless, eay to use and hustle free platform which can be used for making the container selling easy. This platform will elimate the problems associated with container selling in India as in many parts of world

3.3 HARDWARE REQUIREMENTS

- The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.
- PROCESSOR: DUAL CORE 2 DUOS
- RAM : 2 GB DD RAM
- HARD DISK : 250gb

3.2 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- FRONT END : J2EE (JSP, SERVLET)
- BACK END : MY SQL5.5
- OPERATING SYSTEM : WINDOWS 7
- IDE: ECLIPSE

Chapter 4

MATERIAL METHODS AND ALGORITHM USED

4.1 DESIGN ENGINEERING

Design Engineering deals with the various UML [Unified Modeling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product

4.1 USE CASE DIAGRAM:



EXPLANATION:

The use case diagram is the main building block of <u>object oriented</u> modeling. It is used both for general <u>conceptual modeling</u> of the systematic of the application, and for detailed modeling translating the models into <u>programming code</u>. For this in our use case diagram first propose a data in this proposed method. In this diagram will present the user, admin and owner.

4.2 STATE DIAGRAM:



EXPLANATION:

St state diagram first propose. The flow of 3 modules in container management system.ate diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

4.3 ACTIVITY DIAGRAM:



EXPLANATION:

Activity diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while the user and owner communicate regards selling and buying process depends upon the admin database.

4.4 CLASS DIAGRAM:



EXPLANTION:

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. The classes in a class diagram represent both the main objects and or interactions in the application and the objects.

4.5 SEQUENCE DIAGRAM:



EXPLANATION:

In our sequence diagram specifying processes operate with one another and in order. In our sequence diagram first propose for this in our component diagram first propose a data in this proposed method.

4.6 DATAFLOW DIAGRAM:

LEVEL 1:



EXPLANATION:

Data flow diagrams are categorized as either logical or physical. A logical data flow diagram focuses on the business and how the business operates. It is not concerned with how the system will be constructed. We can ignore implementation specifics such as, computer configuration, data storage technology, communication or message passing methods by focusing on the functions performed by the system, such as, data collection, data to information transformation and information reporting.

4.7 E-R DIAGRAM:



EXPLANATION:

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure. For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data-flow diagram, which subdivides this process into sub-processes.

CHAPTER 5

RESULT&DISSCUSSION, PERFORMANCE ANALYSIS

5.1SYSTEM ARCHITECTURE:



EXPLANATION:

The systems architect establishes the basic structure of the system, we can put a small part of data in local machine and fog server in order to protect the privacy. Moreover, based on computational intelligence, this algorithm can compute the distribution proportion stored in cloud, fog, and local machine, respectively. Through the theoretical safety analysis and experimental evaluation, the feasibility of our scheme has been validated.

5.2 IMPLEMENTATION

GENERAL

In this we implement the coding part using eclipse. Below are the coding's that are used to apply for the various schemes available.

5.2.1Index.jsp:

```
<!DOCTYPE html>
```

<html lang="en">

<head>

```
<title>Bootstrap Example</title>
```

```
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
k href="css/bootstrap.min.css" rel="stylesheet">
```

<script

```
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.2/dist/js/bootstrap.bundle.min.js">
```

</script>

```
</head>
```

<style>

body{

```
background-image:url("image/r8.jpg");
```

```
background-size:cover;
```

}

```
</style>
```

<body>

<nav class="navbar navbar-expand-sm bg-dark navbar-dark">

```
<div class="container-fluid">
```

```
<image src="image/r9.png" style="width:100px;"></a>
```



```
<span class="navbar-toggler-icon"></span>
```

</button>

<div class="collapse navbar-collapse" id="collapsibleNavbar">

```
<l
```

```
<a class="nav-link" href="userlogin.jsp">USER</a>
```

```
<a class="nav-link" href="ownerlogin.jsp">OWNER</a>
```

```
<a class="nav-link" href="adminlogin.jsp">ADMIN</a>
```

</div>

</div>

</nav>

</body>

</html>

5.2.2 Ownerlogin:

<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>

left: -20px; <!DOCTYPE html Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"> <html> <head> <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Insert title here</title>

<style>

@import url(https://fonts.googleapis.com/css?family=Exo:100,200,400); @import url(https://fonts.googleapis.com/css?family=Source+Sans+Pro:700,400,300);

body{

margin: 0; padding: 0; background: #fff;

color: #fff; font-family: Arial; font-size: 12px;

}

.body{

position: absolute; top: -20px;

PUBLIC "-//W3C//DTD HTML 4.01

```
; width: auto; height: auto;
       r
                        background-image: skyline-wallpapers-008.jpg):
       i
       g
       hurl(http://ginva.com/wp-content/uploads/2012/07/city-
      t
       :
                 }
       _
       4
                 .grad{
       0
       Background-size: cover;
       Х
       -webkit-filter: blur(5px);z-
       index: 0;
       b
       0
       t
       t
       position: absolute;
       top: -20px; left:
      :20px; right: -
      40px; bottom: -
       40px;width:
      auto; height:
       Quto;
       Background:
                       -webkit-gradient(linear,
                                                   left
                                                          top,
                                                                 left
                                                                        bottom,
                                                                                   color-
       х
stop(0%,rgba(0,0,0,0)), color-stop(100%,rgba(0,0,0,0.65))); /*
       Chrome,Safari4+ */z-index: 1;
      opacity: 0.7;
```

}

.header{

```
position: absolute;
top: calc(50% -
35px);
left: calc(50% - 255px);
z-index: 2;
```

}

```
.header div{
```

```
float: left;
color: #fff;
font-family: 'Exo', sans-serif;
font-size: 35px;
font-weight:
200;
```

```
}
```

```
.header div span{
```

```
color: #5379fa !important;
```

```
}
```

```
.login{
```

```
position: absolute;
top: calc(50% -
```

75px);left: calc(50%

- 50px);

height:

150px; width:

350px

;padding:

10px;

```
z-index: 2;
```

}

.login

input[type=email]
{width: 250px;
height: 30px;
background: transparent;
border: 1px solid
rgba(255,255,255,0.6);borderradius: 2px;
color: #fff;
font-family: 'Exo', sansserif;font-size: 16px;

font-weight: 400;

padding: 4px;

}

.login

input[type=password]{width: 250px; height: 30px; background: transparent; border: 1px solid rgba(255,255,255,0.6);borderradius: 2px; color: #fff; font-family: 'Exo', sansserif;font-size: 16px; font-weight: 400; padding: 4px; margin-top: 10px;

}

.login input[type=submit]{ width: 260px; height: 35px; background: #fff; border: 1px solid #fff; cursor: pointer; border-radius: 2px; color: #a18d6c; font-family: 'Exo', sans-serif; font-size: 16px; font-weight: 400; padding: 6px; margin-top: 10px;

}

.login input[type=submit]:hover{ opacity: 0.8;

}

.login input[type=submit]:active{

```
opacity: 0.6;
```

}

```
.login
```

```
input[type=text]:focu
s{outline: none;
border: 1px solid rgba(255,255,255,0.9);
```

}

```
.login input[type=password]:focus{
```

outline: none;

```
border: 1px solid rgba(255,255,255,0.9);
```

```
}
```

```
.login
input[type=submit]:focu
s{outline: none;
}
```

```
::-webkit-input-placeholder{
color:
rgba(255,255,255,0.6);
```

}

```
::-moz-input-placeholder{
```

```
color:
rgba(255,255,255,0.6);
```

}

</style>

</head>

<body>

<center>

<form action="ownetloginservlet" method="post">

```
<div class="body"></div>
```

<div class="grad"></div>

<div class="header">

<div>EmployeeLogin</div>

</div>

<div class="login">

<input type="email" name="email" placeholder="Email ID" ">

<input type="password" name="password" placeholder="password" ">

<input type="submit" value="Submit"">

Newuser?<button>Signup</button>

</div>

</form>

</center>

</body>

</html>

5.2.3 Owner register:

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"pageEncoding="ISO-8859-1"%> <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Insert title here</title>

<style>

@import url(https://fonts.googleapis.com/css?family=Exo:100,200,400);@import url(https://fonts.googleapis.com/css?family=Source+Sans+Pro:700,400,300);

body{

margin: 0; padding: 0; background: #fff;

color: #fff;

font-family: Arial; font-size: 12px;

}

.body{

position: absolute; top: -20px; left: -20px;

```
background-image: skyline-wallpapers-
                right: -40px;
                                              008.jpg);
                bottom: -40px;
                width: auto;
                                     url(http://ginva.com/wp-content/uploads/2012/07/city-
                height: auto;
                background-size: cover;
                -webkit-filter:
                blur(5px);z-index: 0;
         }
                op: -20px; left:
         .grad{ -20px; right: -
                                                                        bottom,
                                                                                    color-
                40px; bottom: -
                40px;width:
position:
                auto; height:
absolute;
                auto;
                background:
                              -webkit-gradient(linear,
                                                        left
                                                                    left
                                                              top,
         stop(0%,rgb(121 162 253 / 0%)), color-stop(100%,rgb(19128 28))); /*
         Chrome,Safari4+ */
               z-index: 1;
               opacity: 0.7;
         }
```

t

.header{

```
position: absolute;
top:calc(50% -
35px);
left: calc(50% -
255px);z-index: 2;
```

```
}
```

```
.header div{
```

```
float: left;
color: #fff;
font-family: 'Exo', sans-
serif;font-size: 35px;
font-weight: 200;
```

}

```
.header div span{
```

```
color: #5379fa !important;
```

```
}
```

```
.login{
```

```
position: absolute;
top: calc(50% -
```

75px);left: calc(50%

- 50px);height:

150px;

width: 350px;

```
padding:
10px;z-index:
2;
```

}

.login

input[type=text] {width: 250px;

height: 30px;

background: transparent;

border: 1px solid

rgba(255,255,255,0.6); border-

radius: 2px;

color: #fff;

font-family: 'Exo', sans-

serif;font-size: 16px;

font-weight:

400;padding:

4рх;

}

.login

input[type=email] {width: 250px; height: 30px; background: transparent; border: 1px solid rgba(255,255,255,0.6);borderradius: 2px;

color: #fff;

font-family: 'Exo', sansserif;font-size: 16px; w

```
color: #a18d6c;
```

font-family: 'Exo', sans-serif; font-size: 16px; font-weight: 400; padding: 6px; margin-top: 10px;

}

```
.login input[type=submit]:hover{ opacity: 0.8;
```

}

```
.login input[type=submit]:active{ opacity: 0.6;
```

}

```
.login input[type=text]:focus{
```

outline: none;

border: 1px solid rgba(255,255,255,0.9);

}

```
.login input[type=password]:focus{outline:
none;
border: 1px solid rgba(255,255,255,0.9);
```

}

<div class="login">

<input type="text" name="username" placeholder="UserName ">

<input type="text" name="phoneno" placeholder="phoneno" >

<input type="email" name="email" placeholder="email" >


```
<input type="password" name="password1"
id="password1" ><br><br>
<inputtyp e="password" name="password2"
id="password2" ><br><br>
```

<input type="submit" value="Submit">

placeholder="Confirm password"

```
window.onload = function ()
                            </b document.getElementById("password1").onchange = validatePassword;
        </div>
                            ody document.getElementById("password2").onchange = validatePassword
         </form>
         </center>
                   <script>
function validatePassword()
var
document.getElementById("password2").value;
pass2
var
document.getElementById("password1").value;
if (pass1 != pass2)
               document.getElementById("password2").setCustomValidity("Password
         Doesn't Match");
            else
document.getElementById("password2").setCustomValidity(");
                                        //empty string means no validation error
```

5.2.40wner main page:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
k rel="stylesheet" href="css1/bootstrap.min.css">
```

</head>

<style>

body{

background-image:url("image/r10.jpg");

background-size:cover;}

</style>

<body>

```
<nav class="navbar navbar-expand-sm bg-light justify-content-center">
```

```
<l
```

```
<a class="nav-link" href="addcontainer.jsp">Add Container</a>
```

```
<a class="nav-link" href="ownerview.jsp">View</a>
```

```
<a class="nav-link" href="logout.jsp">Logout</a>
```

</nav>

</body></html>

5.3Admin activate:

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"pageEncoding="ISO-8859-1"%>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<%@page import=" dbcon.Dbcon"%>
```

```
<%@page import="java.sql.ResultSet"%>
```

```
<%@page import="java.sql.PreparedStatement" %>
```

```
<%@page import="java.sql.*" %>
```

```
<%@page import="java.util.*" %>
```

<html>

<head>

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
```

k href="css/style.css" rel="stylesheet" type="text/css" media="all" />

k href="css/bootstrap.min.css.map" rel="stylesheet" type="text/css"media="all" />

```
<script src="text/javascript" src='js/jquery-3.6.0.min.js'></script>
```

```
<script src="text/javascript" src='js/bootstrap.min.js'></script>
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
k rel="stylesheet" href="css/font-awesome.min1.css">
```

</head>

<style

>

body{

background-color:#9fa9bb;

}

table, td, th {

border: 1px solid black;

}

table {

border-collapse: collapse; width: 70%; margin-rigth: 70px;

}

td{

text-align: center; padding-top: 1.0em; padding-bottom: 1.0em; }

th{

border: 3px solid black;

}

</style>

<body>

<center>

<div class="container-fluid">

<thead>

```
OWNER NAME
```

OWNER EMAIL

CONTACT NUMBER

STATUS

ACTIVATE

</thead>

<%

Connection con;

con=Dbcon.create();

PreparedStatement

`container`.`ownerreg` ")

ps=con.prepareStatement("SELECT * FROM

ResultSet rs=ps.executeQuery();

while(rs.next())

{

String email=rs.getString(2); String cname= rs.getString(1);

```
%>
```

```
<%=rs.getString(2)%>
<%= rs.getString(3) %>

<%= rs.getString(3) %>

<%= rs.getString(4) %>

<%= rs.getString(7) %>

<<%= rs.getString(7) %>
```

</div>

</center>

</body>

</html>

5.4Activate:

<%@ page import="dbcon.Dbcon" %>

<%@page import="java.sql.PreparedStatement"%>

<%@page import="java.sql.*"%>

<%

```
String id=request.getParameter("id");
```

```
String status="Activate";
```

try{

Connection con=Dbcon.create();

Statement st=con.createStatement();

```
st.executeUpdate("UPDATE `container`.`ownerreg` set status='"+status+"'
where id='"+id+"' ");
```

response.sendRedirect("adminmainpage.jsp?valid");

}

```
catch(Exception e){
```

response.sendRedirect("error.jsp?inval id");

```
System.out.println(e);
```

}

%>

VARIOUS SNAPSHOTS

Home page:



Owner login:

S Insert title here x +	
← → C (D) localhost:8080/Container/ownerlogin.jsp	Q 🖻 🕁 😩 :
🛗 Apps 🕝 Gmail 😰 YouTube 💡 Maps 💶 ata k Kaggle: Your Home 🥠 Android Based Coll 🚹 Untitled Diagram	Reading list
<section-header><section-header><text><text><text><text></text></text></text></text></section-header></section-header>	
	- 🚯 🛍 🔥 4:15 PM

Owner register:

Insert title here x +		
← → C ① localhost:8080/Container/ownerreg.jsp		९ 🖻 ☆ 😩 :
👖 Apps 🕝 Gmail 💼 YouTube 💡 Maps 🧰 ata 🗼 Kaggle: Your Home 🀬 Android Based Coll	🚹 Untitled Diagram	II Reading list
Register		
	and the second	
	Submit	
		415.004
👌 🕝 📜 🔍 🦁 🜍 🌮 🔍 🖳 📳		- 😼 💷 🌜 4:15 PM 12/8/2021

Owner home:



Owner container add:

S Reach Me	× +			
\leftrightarrow \rightarrow C (1) localhost:80	80/Container/addcontainer.jsp		Q. (8 🛦 😩 :
III Apps G Gmail 💽 YouTube	e 🥥 Maps 👩 ata 🗼 Kaggle: Your H	ome Android Based Coll 🚹 Untitled Diagram		E Reading list
		Add Container Details		10 C
		alle	Concerning the second second	
		Owner Name:	and the second second second second	- and the second
		Enter Full Name	A DECEMBER OF THE OWNER	
		Email:	And the second second	10
		prajith@gmail.com	7	TATE
		Container ID:		
in the second		C18C393E31		
and the second s		Size of Container:		
		609.6 cm/20ft	~	
		Types OF Container:		
		DOUBLE DOOR CONTAINERS		
		₹Price Per Year		
	and an an an	Enter price for year		4111415
		Address of Owner		Sector Sector
1-1-1-1		Enter Address		
\varTheta 🙆 📜 🖸				4:17 PM 12/8/2021

User login page:



User register page:

O USER REGISTRATION × +	
← → C ① localhost:8080/Container/userreg.jsp	ର୍ଜ 🛧 😩 :
🛗 Apps 💪 Gmail 😰 YouTube 💡 Maps 🔯 ata k Kaggle: Your Home ೡ Android Based Coll 🚹 Untitled Diagram	E Reading list



🚱 🏉 🚆 💽 🌍 🌮 🔊 🕎 😰 🔯 🕥 - 🛚 🖬 💶 418	PM /2021
-----------------------------------	-------------

User home page:

Insert title here x +		
← → C ① localhost:8080/Container/usermain.jsp		•• Q 🖻 🕁 🛔 :
🔢 Apps 🌀 Gmail 😰 YouTube 💡 Maps 😰 ata k Kaggle:	Your Home 🏼 🎢 Android Based Coll 🔂 Untitled Diagram	E Reading list
Home View Containers Confirmation Logout		

▲ 138 PM ▲ 12/8/2021

Container request page:

Owner Name: abc Fmai: bb@gmail.com Container ID: 9EC823076D Container Size: 1219.2 cm/40ft Container Type: TANK CONTAINERS RPrice Per Year 100000 Address of Owner Tangar User name Enter user name							
abc Email: bb@gmail.com Container ID: 98C823076D Container Size: 1202 Container Type: TANK CONTAINERS EPrice Per Year 1000000 Address of Owner Tagar User name Enter user name Enter user name	MARCHARD AND AND AND AND AND AND AND AND AND AN	NIA COLOR	A STOCKS				
Fnail: bb@gmail.com Container ID: 98C8230760 21219.2 cm/40ft Container Type: TANK CONTAINERS TPrice Per Year 1000000 Address of Owner Tangar User name Interview	21	AND DESCRIPTION					
Container ID: 98C823075D Container Size: 1219.2 cm/40ft Container Type: TANK CONTAINERS RPrice Per Year 1000000 Address of Owner Tangar User name Enter user ame Enter user ame	E AN ARTIGATION	A MERICAN AND	om				
98C823076D Container Size: 1219.2 cm/40ft Container Type: TANK CONTAINERS Price Per Year 1000000 Address of Owner Tangar User name Enter user name User on mainter Size: Container Si	A a SIL XI						
Container Sze: 1219.2 cm/40ft Container Type: TANK CONTAINERS Price Per Year 1000000 Address of Owner Tnagar User name Enter user name Enter user name		A CALLER POR					
1219.2 cm/40ft Container Type: TANK CONTAINERS Price Per Year 1000000 Address of Owner Tnagar User name Enter user name Enter user name	A MARK	Calles States		c c c	-		
Container Type: TANK CONTAINERS Price Per Year 1000000 Address of Owner Tragar User name Enter user name Enter user name	2 AMALON	1 Charles and the second	Oft		1		
TANK CONTAINERS CPrice Per Year. 1000000 Address of Owner Tragar User name Enter user name	NIL AL	CAMPAN AND AND AND AND AND AND AND AND AND A	e	C			
RPrice Pei Year 1000000 Address of Owner Tragar User name Enter user name	SS STATISM		AINERS	-			
100000 Address of Owner Tnagar User name User name			r de la constante d	33990 र			
Address of Owner Tnagar User name Enter user name				and the second			
Tragar User name Enter user name	A BARAN II'Y		vner	A	all and the	Aw	
User name Enter user name							
Enter user name			and the second se	U Start			
			ame				
User main				the state of the s			
syed@gmail.com			l.com				_

Admin home page:

e) Insert	title here	>	< +													0	ð	<u> </u>
÷	\rightarrow	C O I	ocalhost:8080,	/Container/	adminmain	ipage.jsp								0	, Q	Ċ	☆		:
	Apps	G Gmail	YouTube	Maps	🖸 ata	Kaggle: Yo	ur Home	ndroid Ba	sed Coll	🚹 Untitled D	iagram						E F	Reading	list
	0wn	ier																	
	User	Activate																	
	l Vie																		
	i Vie	w Request																	
=	Con	firmation																	
	Log	out																	
(6				8									- R	i 🐑 ((4:41 PM	
	1			0											100	11 (1997) 11 (1997)	1	.2/8/202	L

CHAPTER 6

CONCLUSION:

In this paper the authors proposed a seamless, easy to use and hustle-free platform which can be used for making the container selling easy. There are many problems such as involvement of brokers or middleman, time delays, etc. This platform will eliminate the problems associated with containers selling in India as well as in many parts of the world. The steps involved in the process of containers are discussed in details in the paper. Making container selling paperless will not only make the process easier but also secure the papers of ownership of containers from various man-made and natural disasters. There are many additional features that can be added to the platform of containers registry. Nowadays, land is not a liquidated asset. By using the platform land assets can also be liquidated using the crypto currency, that maps with the land record created by a seller on the platform. Hence the scope is wide and there can be many use cases of the platform created.

CHAPTER 7

REFERENCE

1. virtualization.

http://www.kernelthread.com/publications/virtualization

2. docker. https://www.docker.com

3. uchiya, t., hara, h., sugawara, k., kinoshita, t.: repositorybased multiagent framework for developing agent systems. in: transdisciplinary advancements in cognitive mechanisms and human information processing, chap. 4, pp. 60–79. igi global(2011)google scholar