CHAPTER-1

INTRODUCTION

1. INTRODUCTION

1.1. PROJECT OVERVIEW

The project named as “Hire and build” deals with the perfect utilization of equipment. This will help the customers (who would like to involve construction activities) to search the resources easily and effectively and facility to rent equipment. registered suppliers can participate in the renting process. Handling resources manually is impractical so we have to develop a system that will easily schedule, track and maintain the equipment. There will be an administrator login that is accessed by the admin who has approved the suppler into the website. Admin can put equipment list for suppliers and the supplier can register the equipment on in his hands and participate in the bidding process. And the customers can book these types of equipment. The booking process is fully automated. The functions are performed by customers and contractors. They can register to this website and join the part of the hiring process only after the approval of an admin.

Handing resources manually is impractical so we have to develop a system that will easily to get the types of equipment with lots of choices and minimum rent money. The developer of this website think like that ‘if a customer uses this website when he gets any benefit compared to the normal renting system ‘ then only he will use. So the developer applies the method linear programming problem in operation research. The concept of LPP is to maximize the profit with minimum input .so implement this concept here also by converting the mathematical format into the programming code. And also using some searching and sorting mechanisms. Anyway, the customer has lots of choices here and also he gets the types of equipment with the minimum cost at minimum distance. In the normal process, customers have only two or three chances finally he select the equipment as which one is nearest at a high cost. So in this case, if a customer using this system he can see lots of choices and select the appropriate supplier. In the case of a supplier, he has also benefited from using this website. Suppose if he has a lot of types of equipment are in his hand but he doesn't know who needs this equipment in this case if he uses this system he can present equipment to the society the customers will identify the suppliers so it will be more useful for the supplier. Here the admin also gets a certain benefit for using this system. I.e. if there any transactions are made with supplier and customer a particular amount of commission will be credited to the supplier account. So we can say that the system is a business-oriented.

THIS PROJECT INCLUDES THREES MODULES:

- Admin Module.
- Supplier Module.
- Customer Module.

CHAPTER-2

SYSTEM ANALYSIS

2.0 SYSTEM ANALYSIS

System analysis is the process of collecting and interpreting facts, understanding problems and using the information to suggest improvements on the system. This will help to understand the existing system and determine how computers make their operation more effective. This analysis aims to collect detailed information on the system and the feasibility study of the proposed system. This analysis focuses on the flow of the system module by module and the efficiency of each. To design the proposed system we need the exact processing logic as well as the extended features of the existing system such as reliability, consistency, storage capacity, etc. This report will discuss the advantages and drawbacks/disadvantages of the existing system and the modifications and enhancements can be done. This analysis will concentrate on the information gathering for the efficient, user-friendly and reliable system, which will carry forward the features of the existing system.

2.1. REQUIREMENT ANALYSIS

Requirements analysis results in the specification of software's operational characteristics indicate software's interface with other system elements and establish constraints that software must meet. Requirements analysis allows you to elaborate on basic requirements established during the inception, elicitation, and negotiation tasks that are part of Requirements engineering.

REQUIREMENT GATHERING

The requirement gathering can be done in the following ways.

- Interview.
- Website visit.

For this project is used Website visit and Interview method.

For this project, I visited the following resources.

- I visited a few Construction Websites. These sites are given below:
    - ✓ http://www.Equipment cafe.com
    - ✓ http://www.PC earthmovers.com
    - ✓ http://www.Equipment share.com

These websites help to get information about the construction company, what are the main functionalities of the construction company and how to develop a website very usefully.

- The interview method helped to collect more information from the supplier and customers.
- I interviewed four peoples to gather information
- I used the interview technique for gathering information and I asked the following questions.

  To the contractor of Jeena group Mr. Manoj

1. How are you storing the information about the customer?

   Data are stored in files and records.

2. What are the difficulties faced by the site engineer?

   When the data is stored in the files, it is difficult to identify the old user for any other reference.

3. Is there any equipment renting system?

   No. There is no booking system.

4. What are the legal formalities for handling human resources?

Inform police station with their id cards and keep one of the copies in hand. Medical checkup proof. Personal data etc

- Next, I interviewed a JCB driver Mr. Nameesh for gathering information and I considered he is a supplier and I asked the following questions.

1. Is there any available booking system available now?

   A: No. there is not any JCB booking system is available now.

2. How to calculate JCB rent normally?

   A: Normally calculate the rent per hour and also consider the risk of the work.

3. All times do you have work?

   A: Not at all rainy season work is very less.

4. How the customer identify you .how they book your work?

   A: Normally they contact me by phone calls .i have a visiting card they got my advertisement. Or

   The customer directly contacts me.

5. Work is a centralized nearest place or far place?

   A: Both nearest and far places

● I interviewed another small scale contractor Mr. Sanith for gathering information and I considered he is a user and I asked following questions
.

1. How to collect equipment for construction?

A: Normally all the equipment is in my hands if more equipment needed at the time of construction then only hire the equipment from others.

2. So you take the equipment from others. How many suppliers did you meet?

A: One or two suppliers only I meet they are nearest to me.

3. Suppose the equipment cost is very high from the current suppliers what you do?

A: We don't have more option so we will take the equipment from that supplier.

4. If you make any damages on the equipment then do you repair or make the corresponding payments?

A: It depends on the situation sometimes maybe make the payments sometimes maybe repair.

● Next, I interviewed Mr. Sajeevan for gathering information and I considered he is a supplier and I asked the following questions.

1. How the customer identifies the equipment are in your hand?

A: The customers are the nearest place so everyone knows to me.

2. So all the customers are from the nearest. There is any simplest way to identifies customer?

A: Not any proper system is available.

3. How to measure the rent for each item?

A: Normally consider per day a particular amount.

2.2. EXISTING SYSTEM

The existing system was handled manually. The Existing system is not an integrated system and it is not fully automated. The information's which consists of human resources and rental of equipment, auctions, etc are scattered in the entire system. All documentations are kept as MS-Word documents or excel sheets. In the current system not easy to find the resources for the customer. This will make the current system more time-consuming. It can't provide proper interaction between the users and the customer. The customer didn't know who the best supplier is where he can get equipment with minimum cost and minimum distance. No mathematical approach for finding appropriate equipment from the supplier. These problems will lead to certain loses for the customers. And also the supplier didn't know who needs equipment. He can only get publicity through the advertisement. But it has limitations. In a normal system, there is a chance to cheat by the customer i.e. after using the equipment he does not make the payment as well as there is a high chance of negotiation between supplier and customer in the case of the rent amount. But here in this system consist first to make the payment then only he can get the equipment. If there any damages are created by the customer in a normal system supplier can’t make the compensation from the corresponding customer.

From the evaluation of the existing system, we can find out the following drawbacks:

- No proper handling of resources.
- There are no provisions for the search using the equipment names and other search terms.
- The existing system is too slow for getting resources.
- All the records were done manually more human effort is needed.
- Since it requires high manual effort, files and books and there are chances of misplacing important details and files containing crucial data. This will lead to loss of data.
- Time-consuming.
- Payments are manual. It leads to high negotiation between the supplier and customer

- There are limited chances only for finding suppliers.

2.3. PROPOSED SYSTEM

The disadvantages of the Existing System have been solved by automating the **'HIRE AND BUILD'** which helps to current and accurate details. The proposed system is a step ahead in handling day to day activities in construction parties with the help of the computerized system. Since a change from a manual system to a computer system saves a lot of time here provides more selection opportunities thus saving an enormous amount of money. The system identifies which is the best option for the customers here applied the simplex method we reduce the cost by using this method and get the optimum solution. The system was designed in such a way that it is useful to any construction parties or single men.

Details of the customer information, amount paid, acknowledgment details, other information can be provided to the supplier on the click of the button. It is designed keeping in mind all the drawbacks of the present system to provide a permanent solution to the existing system. The main aim of the software is to the customer can get the equipment on a minimum cost with minimum distance and provide more details of suppliers so the customer gets a lot of resources.

The proposed system is also expected to reduce the amount of paperwork normally this kind of data is stored in the document. The hard copies of only the necessary document needed to be taken and the rest can be avoided.

Merits of the proposed system

- Customer can collect the equipment's minimum cost with minimum distance
- The minimum time is required.
- Lots of suppliers can be available to the customer.
- Increased speed and accuracy.
- Provide long term storage and fast retrieval of data.
- Eliminate all of the paper works.
- User-friendly and interactive

2.4. FEASIBILITY STUDY

A feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that is spent on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study is a test of the system proposed regarding its workability, impact on the organization, the ability to meet the needs and effective use of resources. Thus when a new project is proposed, it normally goes through a feasibility study before it's approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as technical, legal, economic and behavioral feasibilities.

The feasibility of a project can be ascertained in terms of technical, economic, behavioral and legal factors. A feasibility study is documented with a report showing all the ramifications of the project.
The system proposed is tested whether it is feasible by conducting the following

- Technical feasibility
- Economic feasibility
- Behavioral feasibility
- Legal feasibility

2.4.1 TECHNICAL FEASIBILITY

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs, procedures and staff. Having identified an outline system, the investigation must go on suggest the type of equipment, required method developing the system, of running the system once it has been designed. The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology.

Through the technology may become obsolete after some period of time, due to the fact that newer version of some software supports older versions, the system may still be used. So there are only minimal constraints involved with this project. The system has been developed using JAVA, the project is technically feasible for developed.

2.4.2. ECONOMIC FEASIBILITY

Economic feasibility is the most important and frequently used method for evaluating the effectiveness of the proposed system. It is very essential with increased efficiency of the cost versus the benefit and savings are expected from the proposed system. The proposed method is cost effective and efficient, since it has minimum requirements. The proposed system is economically feasible because it does not use any external hardware. The only requirement is an android platform based smartphone or internet connected PC for verification purpose. As we know that the system development costs are usually onetime costs that will not occur after the project has been completed.

2.4.3. BEHAVIORAL FEASIBILITY

People are resistant to change and computers have been known to facilitate change. The ‘Hire and Build’ is designed in a user-friendly manner and no need to provide training for the persons using this software. The operating system used is Windows 8, which is also user-friendly. It does not have any operational barriers. So we need to provide any special training for using this application software and hence it is behaviorally feasible.

2.4.4. LEGAL FEASIBILITY

The use of the Resource scheduler System will not violate rules and regulations of law. The management is also supportive. It will not make any violation of the norms and rules of the government. So the system is legally feasible.

2.5 SYSTEM REQUIREMENT SPECIFICATION

System requirements are expressed in a software requirement document. The Software requirement specification (SRS) is the official statement of what is required of the system developers. This requirement document includes the requirements definition and requirement specification. The software requirement document is not a design document. It should set out what the system should do without specifying how it should be done. The requirement set out in this document is complete and consistent.

The software specification document satisfies the following: -

- It specifies the external system behavior.
- It specifies constraints on the implementation.
- It is easy to change.
- It serves as reference tool for system maintainers.
- It records forethought about the life cycle of the system.
- It characterizes acceptable response to undesired events.

2.5.1. ACTOR IDENTIFICATION

An actor is someone or something that interacts with the system. An actor is he /she what uses the system. An actor exchanges information with the system. Asking certain questions as detailed below can identify the actors of the system.

1.	Who will use the main functionality of the system?	Administrator, customer
2.	Who will lead support from the system and do their daily tasks?	Administrator, supplier, customer
3.	Who will maintain and administrate the system?	Administrator
4.	With which other systems, does this system need to interact?	Database.
6.	Who was interested in the result produced by the system?	Administrator, supplier, customer

As per the above answers, we can conclude the actors. They are:

- Administrator.
- Supplier
- customer

2.5.2 USE CASE IDENTIFICATION

A use case represents the functionality of an actor. It is defined as a set of actions performed by a system, which yields an observable result. An ellipse containing its name inside the ellipse or below it represents it. It is placed inside the system boundary and connected to an actor with an association. This shows how the use cases and the actor interact. To find out the use cases, ask the following questions to each of the actors.

- Which functions does the actor require from the system? What does the actor need to do?
- Does the actor need to read, create, destroy, modify or store some kind of information in the system?

- Could the actor's daily work be simplified or made more efficient by adding new functions to the system?

2.5.2.1. USE CASES

A use case for the actor Administrator

1	Which functions does the Administrator require from the system? What does the Admin need to do?	The administrator requires the following functionalities from the system such as Add equipment, view booking, approved Booking, view feedback, view equipment, check the commission. Remove the suppliers, communication with supplier through message from the website
2	Does the Administrator need to read, create, destroy, modify or store some kind of information in the system?	Yes. An administrator needs to create and view the data if required.
3	Could the Administrator work be simplified by adding new functions to The system?	Yes, the system can reduce his/her work.

The above questions give the following use cases for the actor Administrator.

- Login
- Add equipment
- approve supplier registration
- view place
- view duration

- View equipment
- Communication with supplier
- Logout

A use case for the actor supplier

1	Which functions does the Supplier Require from the system? What does the Supplier need to do?	The supplier requires the following functionalities from the system such as add equipment his hand. and view Add queries, View equipment, view orders like pending approved returned. if there any problems in returned equipment he can ask the compensation from customer otherwise black list the customer .
2	Does the Supplier need to read, create, destroy, modify or store some kind of information in the system?	Yes. The supplier needs to create and view the data if required.
3	Could the Supplier daily updated be his stock in hand to the system?	Yes, he needed to check the stock is updated or not.

The above questions give the following use cases for the actor Site supplier.

- register
- Login
- Add equipment
- View equipment
- View his customer
- Current availability of resource
- View pending, approved, returned orders
- Check bank details

- Blacklisting customers
- Communication with admin and customer
- Add map for source to destination
- Logout

A use case for the actor Customer

1	Which functions does the customer require from the system? What does the customer need to do?	A customer requires the following functionalities from the system such as Add requirements, view requirements, View suppliers, equipment booking, book rent, and communication with supplier, payment option. view orders, view map
2	Does the customer need to read, create, destroy, modify or store some kind of information in the system?	Yes. The customer needs to create and view the data if required.
3	Could the customer’s simplified by searching and finding equipments easily through the system?	Yes, the system helps to searching and finding equipments easily.

The above questions give the following use cases for the actor Customer.

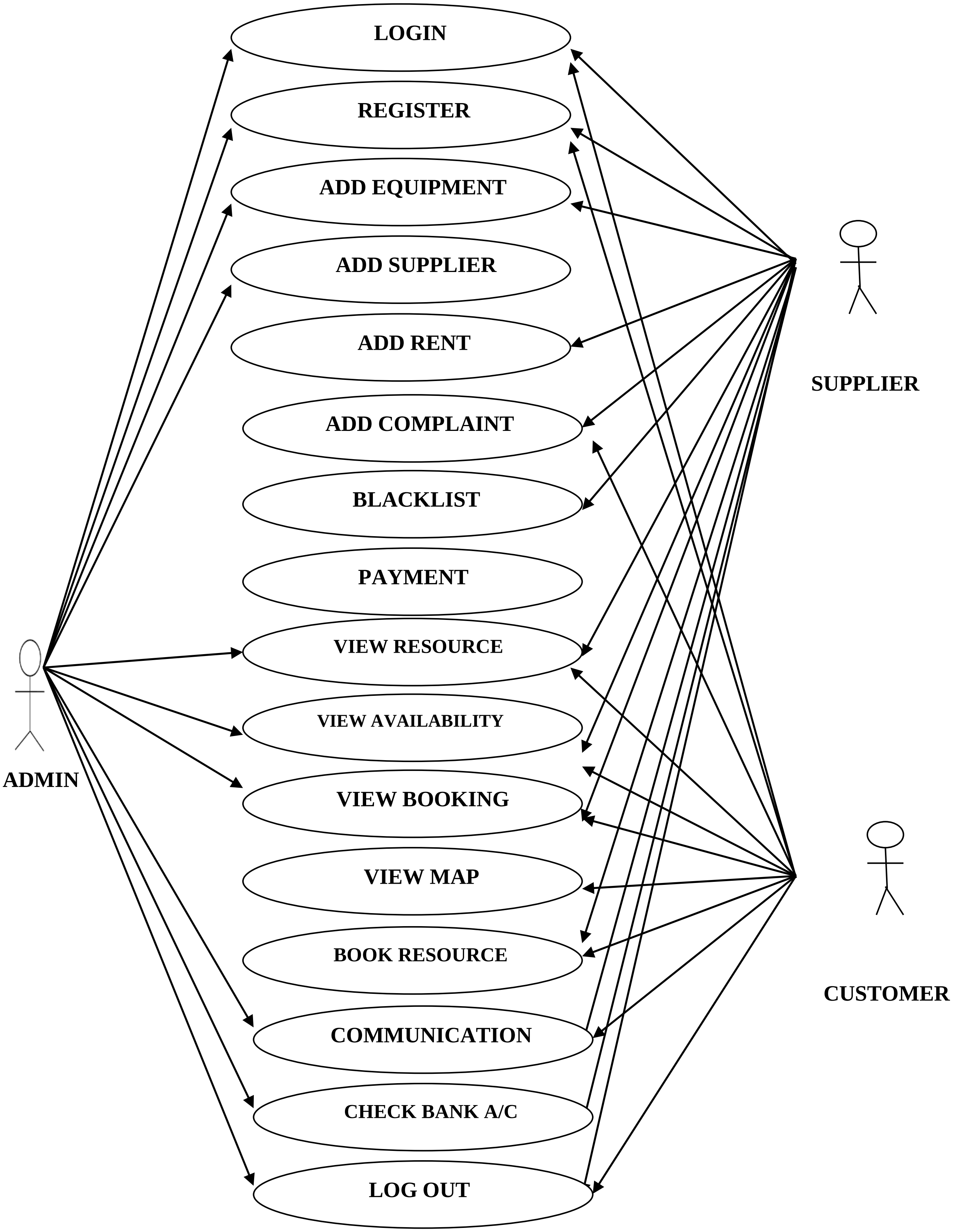
- Register
- Login
- View details of suppliers
- Map for source to destination
- Add requirement
- view rent
- View equipment

- equipment booking
- book rent
- rent payment
- paid the compensation
- communication with supplier
- Logout

2.5.2.2 USE CASE DIAGRAM

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system. An ellipse containing its name inside the ellipse or below it represents it. The use case diagram of this project has shown below. The actors of this project are Admin, Supplier, and Customer.

USE CASE DIAGRAM



2.5.3. ACTIVITY DIAGRAM

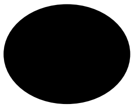
The activity diagram supplements the use case by providing a graphical representation of the flow of interaction within a specific scenario. It uses rounded rectangles to imply a specific system function, arrows to represent flow through the system, decision diamonds to depict a branching decision, and solid horizontal lines to indicate that parallel activities are occurring.

The basic purposes of activity diagrams are similar to other diagrams. It captures the dynamic behavior of the system. Other diagrams are used to show the message flow from one object to another but an activity diagram is used to show message flow from one activity to another.

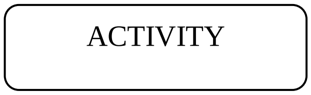
So the purposes can be described as:

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

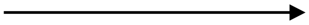
Activity diagram symbols



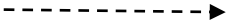
Start/ Initial Node



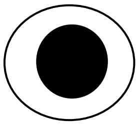
Activity / Action State



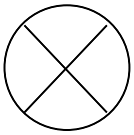
Control Flow / Edge



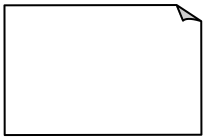
Object Flow / Control Edge



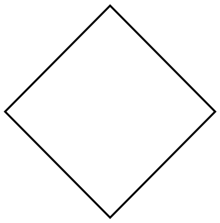
Activity Final Node



Flow Final Node



Note/ Comment



Decision Node

Diagram for login all users

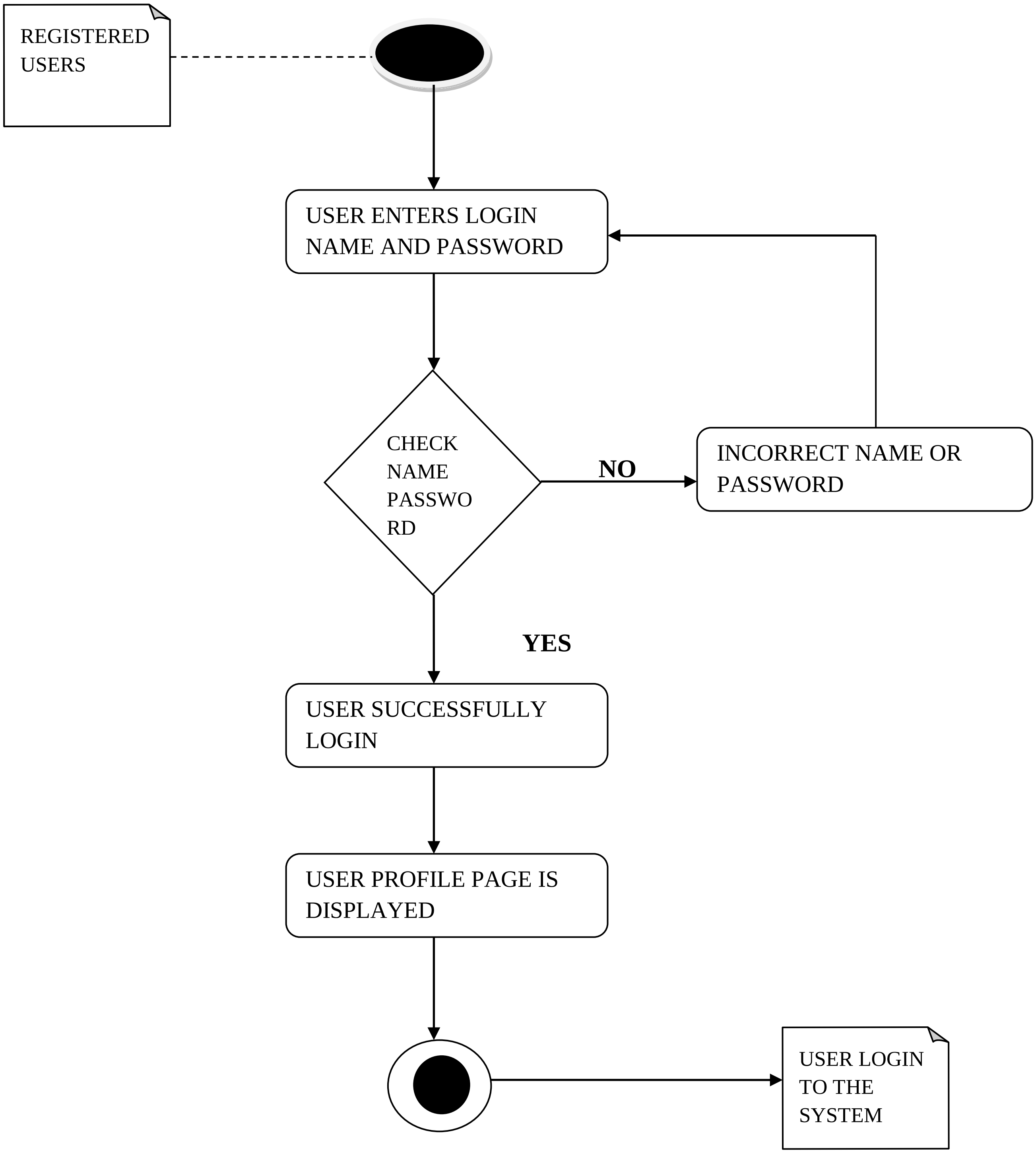

REGISTERED USERS
USER ENTERS LOGIN NAME AND PASSWORD
CHECK NAME PASSWO RD
NO
INCORRECT NAME OR PASSWORD
YES
USER SUCCESSFULLY LOGIN
USER PROFILE PAGE IS DISPLAYED
USER LOGIN TO THE SYSTEM

Diagram for suppler approval

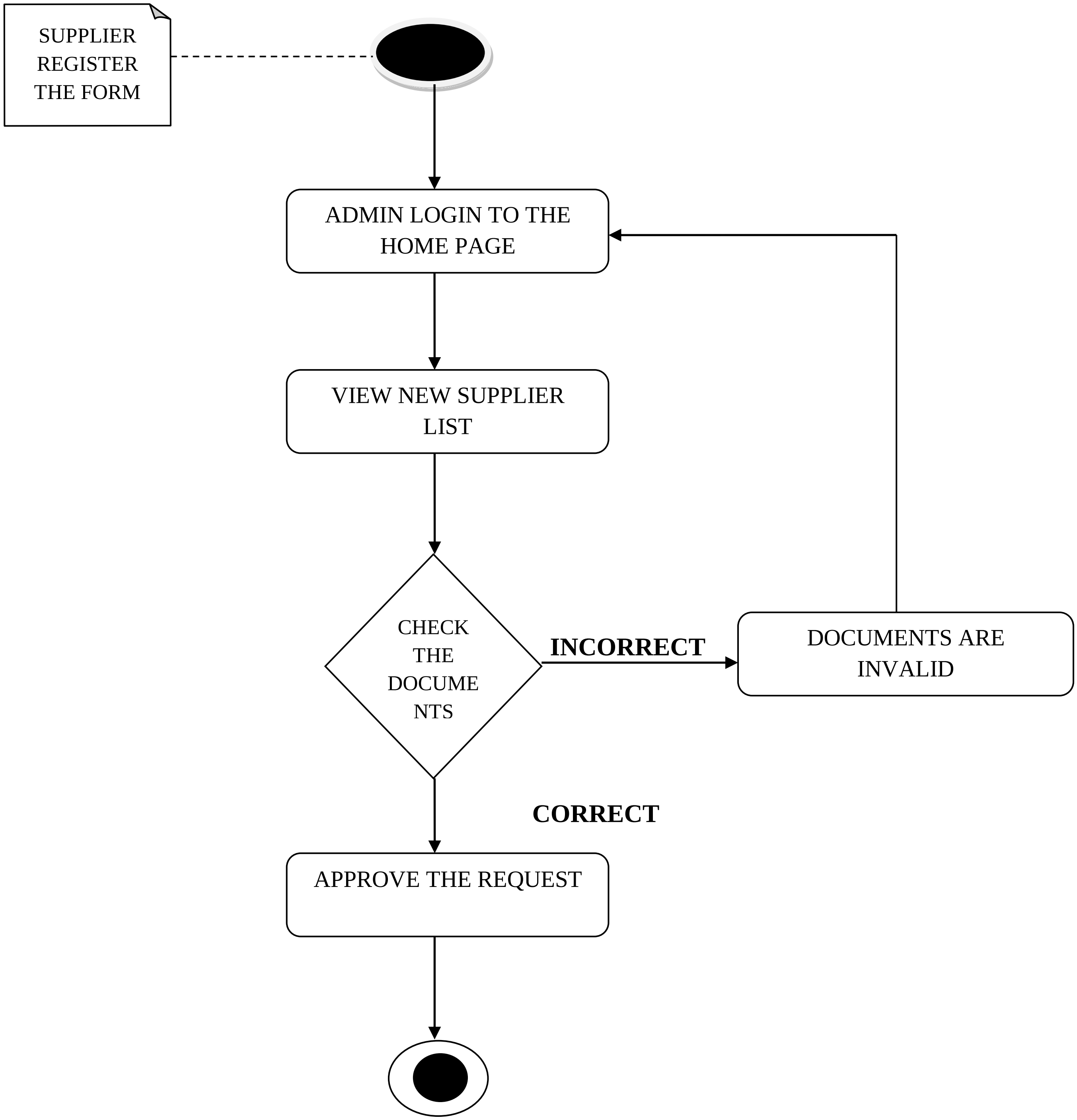


Diagram for customer booking equipment and payment

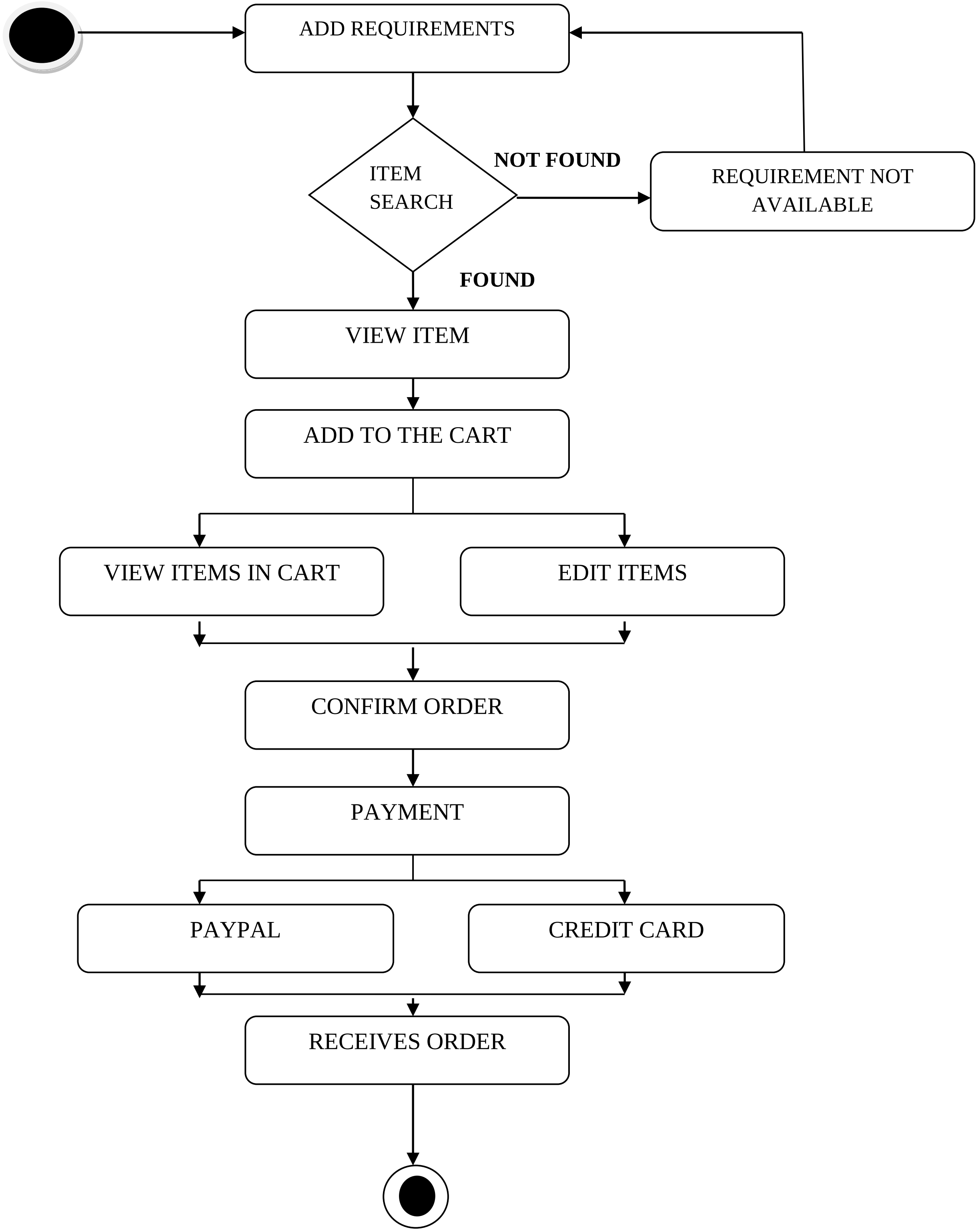
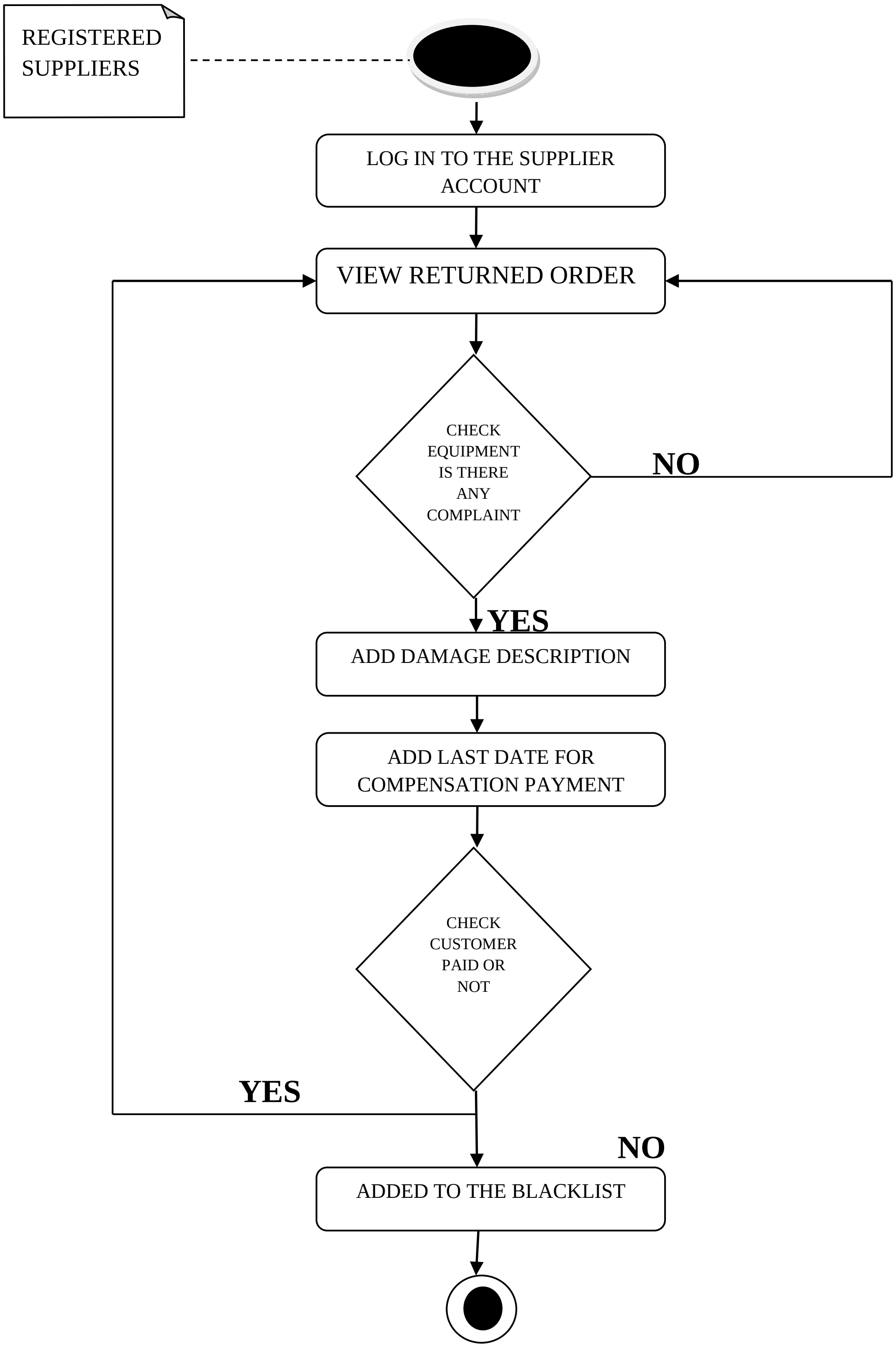


Diagram for blacklisting customer


REGISTERED SUPPLIERS
LOG IN TO THE SUPPLIER ACCOUNT
VIEW RETURNED ORDER
CHECK EQUIPMENT IS THERE ANY COMPLAINT
NO
YES
ADD DAMAGE DESCRIPTION
ADD LAST DATE FOR COMPENSATION PAYMENT
CHECK CUSTOMER PAID OR NOT
YES
NO
ADDED TO THE BLACKLIST

2.5.4. SEQUENCE DIAGRAM

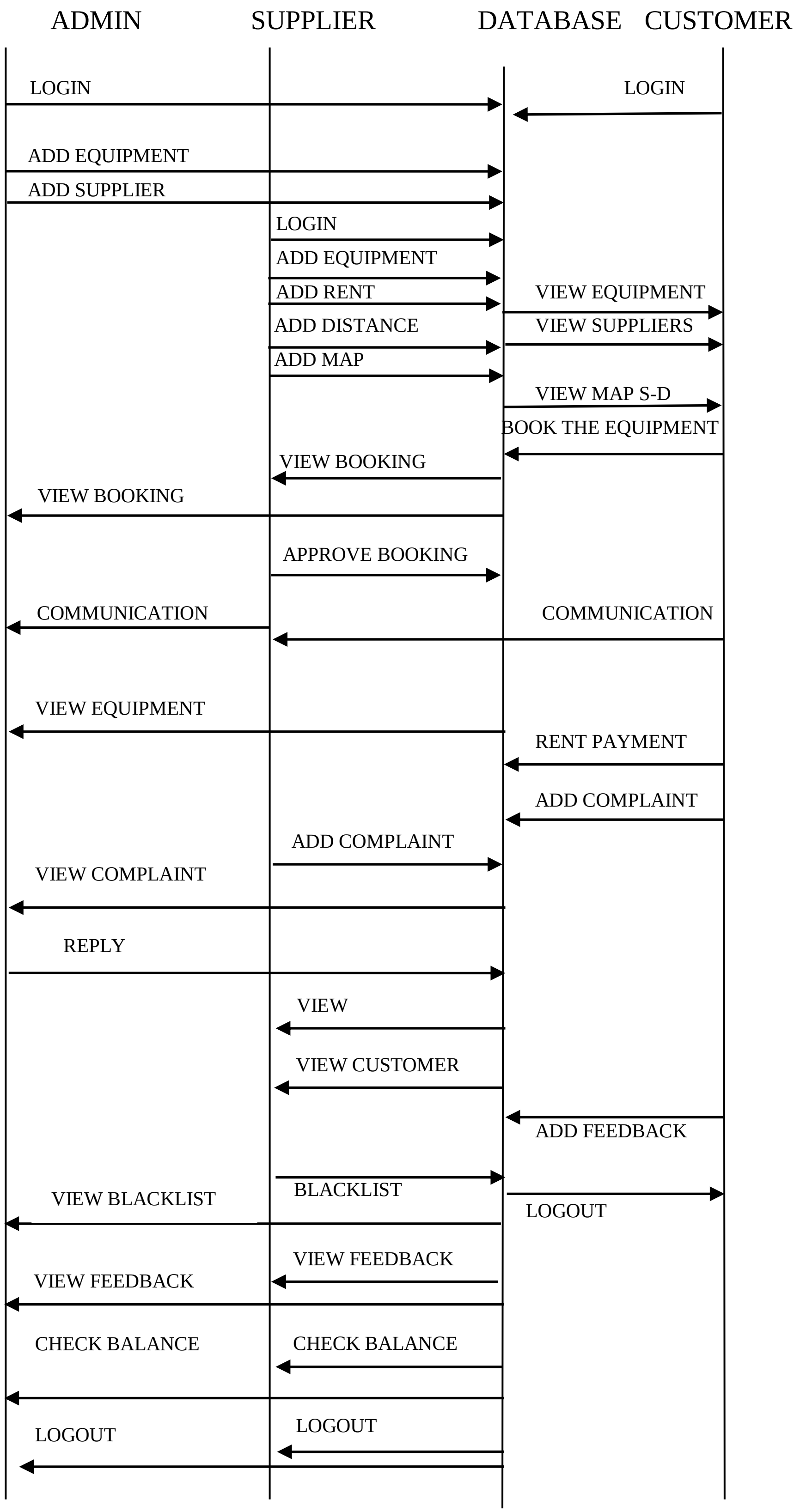
Sequence diagrams are an easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment. A sequence diagram shows an interaction arranged in a time sequence. It shows the objects participating in the interaction by their lifelines and the messages they exchange, arranged in a time sequence.

A sequence diagram has two dimensions: a vertical dimension represents time, the horizontal dimension represents different objects. The vertical line is called the object's lifeline. The lifeline represents the object's existence during the interaction. This form was first popularized by Jacobson. An object is shown as a box at top of a dashed vertical line. A role is a slot for an object within a collaboration that describes the type of object that may play the role and its relationships to other roles. However, a sequence diagram does not show the relationships among the roles of the association among the objects. An object role is shown as a vertical dashed line, the lifeline.

Each message is represented by an arrow between the lifelines of two objects. The order in which these messages occur shown top to bottom on the page. Each message is labeled with the message name. The label also can include the argument and some control information and show self-delegation, a message that an object sends to itself, by sending the message arrow back to the same lifeline. The horizontal ordering of the lifelines is arbitrary. Often, all arrows are arranged to proceed in one direction across the page, but this is not always possible and the order conveys no information.

The sequence diagram is very simple and has immediate visual appeal- this is its greatest strength. A sequence diagram is an alternative way to understand the overall flow of the control of a program. Instead of looking at the code and trying to find out the overall sequence of behavior, we can use the sequence diagram to quickly understand that sequence.

SEQUENCE DIAGRAM



2.6. SYSTEM REQUIREMENTS

2.6.1. HARDWARE REQUIREMENTS

Hardware and software requirements for the installation and smooth functioning of this product could be configured based on the requirements needed by the component of the operating environment that works as front-end system here we suggest minimum configuration for the both hardware and software components. Working off with this application is requirements concrete on system environments. It includes two phases

The hardware required for the application depends on the following:

- Determining size and capacity requirements
- Computer evaluation and measurements
- Financial factors
- Maintenance and support.

Processor: Core I3 higher version (32bit/64bit) Processor

RAM: 2GB and Above

Hard Disk: 50GB

Keyboard: Standard Keyboard with 104 keys

Mouse: Standard Mouse with 3 buttons

Monitor: Display Panel (1366*768)

2.6.2. SOFTWARE REQUIREMENTS

The software required for the application depends on the following factors:

- The flexibility of the software
- Software contracts
- Limitation of the software

Programming language	- JAVA, JAVASCRIPT
DBMS	- SQL Server 2008
Development tool	- Netbeans IDE
Development platform	- Windows 10

CHAPTER-3

SYSTEM DESIGN

3. SYSTEM DESIGN

Design is a meaningful engineering representation of something that is to be built. It is an iterative process through which requirements are translated into a blueprint for constructing the software. The goal of the design phase is to plan a solution to the problem specified by the requirements document.

Major activities during the design phase are:

- Database Design
- Architectural Design
- Interface Design
- Modular Design

3.1. DATABASE DESIGN

A database is collections of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. In database design data independence, accuracy, privacy, and security are given higher priority. Database design is an integrated approach to file design. This activity deals with the design of the physical database. All entries and attributes have been identified while creating the database. The database design deals with the grouping of data into the number of tables to reduce the duplication of data, minimize storage space, and retrieve the data efficiently.

Guidelines for designing a database:

- ➢ Design a relational schema so that it is easy to explain its meaning. Do not combine attributed from multiple entities and relationship types into a single relation.
- ➢ Design the database schema so that no insertion, deletion or modification anomalies are present in the relation.
- ➢ As far as possible, avoid placing attributes in a base relation whose values may frequently be null.

- Design relation schemas so that they can be joined with equality conditions on attributes that are either primary keys or foreign keys in a way that no spurious tuples are generated.

Advantage

- Ease of use
- Data independence
- Accuracy and integrity
- Avoiding inordinate delays
- Recovery from failure
- Privacy and security.

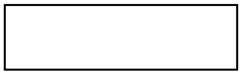
3.1.1. E-R DIAGRAM

An entity-relationship diagram is a data modeling technique that creates a graphical representation of the entities, and the relationship between entities, within an information system.

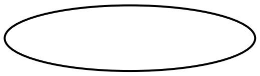
There are three basic elements in ER models:

- **Entities** are the “things” about which we seek information
- **Attributes** are the data we collect about entities.
- **Relationships** provided the structure needed to draw information from multiple entities.

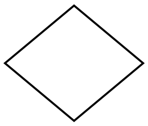
E-R Diagram Symbols:



Entity

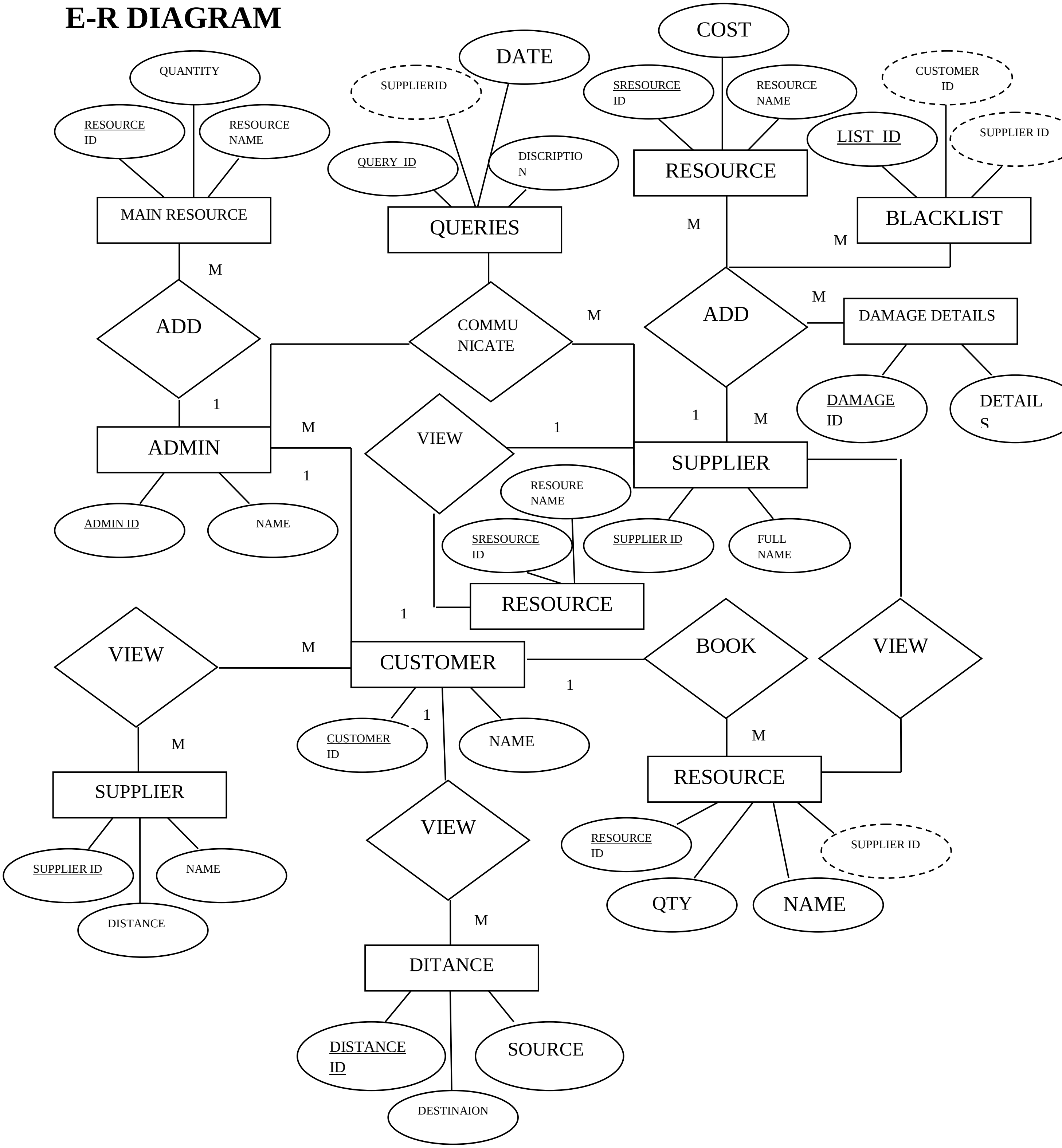


Attributes



Relation

E-R DIAGRAM


COST
DATE
QUANTITY
SUPPLIERID
SRESOURCE ID
RESOURCE NAME
CUSTOMER ID
RESOURCE ID
RESOURCE NAME
LIST ID
SUPPLIER ID
QUERY ID
DISCRIPTIO N
RESOURCE
BLACKLIST
MAIN RESOURCE
QUERIES
M
M
M
ADD
COMMU NICATE
M
ADD
M
DAMAGE DETAILS
1
1
M
DAMAGE ID
DETAIL S
M
VIEW
1
ADMIN
SUPPLIER
1
RESOURE NAME
ADMIN ID
NAME
SRESOURCE ID
SUPPLIER ID
FULL NAME
RESOURCE
1
VIEW
M
CUSTOMER
BOOK
VIEW
1
1
CUSTOMER ID
NAME
M
M
SUPPLIER
RESOURCE
VIEW
RESOURCE ID
SUPPLIER ID
SUPPLIER ID
NAME
DISTANCE
QTY
NAME
M
DITANCE
DISTANCE ID
SOURCE
DESTINAION

3.1.2 TABLE DESIGN

In the database, all the information is stored in the form of tables. A table is simply a way of storing data in rows and columns. In the system, data is stored in many tables.

The table structures are shown below with sample data.

Table 1: admin_tb

Field Name	**Data Type**	**Constraints**	**Description**
Id	Bigint	Primary key	Uniquely identify the user
Username	Varchar(10)	Not null	Username
Password	Varchar(10)	Not null	Password

Table2:Customer_tb

Field Name	**Data Type**	**Constraints**	**Description**
Id	Bigint	primary key	Uniquely identify the user
First name	Varchar(15)	Not null	To store the first name
Middle name	Varchar(10)	Not null	To store middle name
Last name	Varchar(10)	Not null	To store the last name
Gender	Varchar(10)	Not null	To store gender
DOB	Date	Not null	To store DOB
Place	Varchar(15)	Not null	To store place

Post	Varchar(15)	Not null	To store post
Pin code	Varchar(15)	Not null	To store pin code
State	Varchar(15)	Not null	To store state
Phone no	Varchar(15)	Not null	To store phone no
Username	Varchar(10)	Not null	Login
Password	Varchar(8)	Not null	Password

Table 3: addresourcemain_tb

Field Name	**Data Type**	**Constraints**	**Description**
Resourceid	Bigint	Primary Key	uniquely identify the resource
Resourcename	Bigint	Not null	Resource name
Description	Int	Not null	To store the description
Uploads	Varchar(10)	Not null	To store the photos related files

Table 4: add_supplier_resource_tb

Field Name	**Data Type**	**Constraints**	**Description**
Sresourceid	Bigint	Primary Key	uniquely identify the equipment
Quantity	Bigint	Foreign key	Quantity of resource
Amount	Int	Not null	To store the amount
Travel_cost	Int	Not null	store travel cost per km
Resource_id	Int	Not null	Forign key
Supplier_id	Int	Not null	Forign key

Table 5:distance_tb

Field Name	**Data Type**	**Constraints**	**Description**
Distance_id	Bigint	Primarykey	Uniquely identify
Source	Varchar(10)	Not null	Resource availabile place
destination	Varchar(15)	Not null	Destination
distance	Int	Not null	To calculate the distance

Table 6: place_tb

Field Name	**Data Type**	**Constraints**	**Description**
Placeid	Bigint	Primary key	Uniquely identify the site
placename	Varchar(20)	Notnull	To store the place name

Table 7: queries_to_admin_tb

Field Name	**Data Type**	**Constraints**	**Description**
Query_id	Bigint	Primary key	Uniquely identify
Supplier_id	Bigint	Foreign key	Uniquely identify
About	Varchar(50)	Not null	To store description
Details	Varchar(100)	Not null	description
Status	Varchar(10)	Not null	Viwed or not

Table 8: Addfeedback

Field Name	**Data Type**	**Constraints**	**Description**
Feedback_Id	Bigint	primary key	Uniquely identify
Costumerid	Bigint	Foreign key	Uniquely identify
Description	Varchar(50)	Not null	To store description
Date	Date	Not null	Date of complaint

Table 9: queries_reply_tb

Field Name	**Data Type**	**Constraints**	**Description**
reply_id	Bigint	Primary key	Uniquely identify
Query_id	Varchar(50)	Forignkey	Uniquely identify
Reply	Varchar(50)	Not null	Response
Date	Date	Not null	Date

Table 10: resourceorder_tb

Field Name	**Data Type**	**Constraints**	**Description**
Order_id	Bigint	Primarykey	Uniquely identify
Resource_id	Bigint	Foreign key	Uniquely identify
Supplier_id	Varchar(10)	Foreign key	Uniquely identify
Days	Int	Not null	Days for resource
Required_qunatity	Varchar(10)	Not null	Quantity
Totalcost	Varchar(10)	Not null	Cost

Table 11: supplier_tb

Field Name	**Data Type**	**Constraints**	**Description**
Supplier_id	Bigint	primary key	Uniquely identify the user
First name	Varchar(15)	Not null	To store first name
Middle name	Varchar(10)	Not null	To store middle name
Last name	Varchar(10)	Not null	To store last name
Gender	Varchar(10)	Not null	To store gender
DOB	Date	Not null	To store DOB
Place	Varchar(15)	Not null	To store place
Post	Varchar(15)	Not null	To store post
Pin code	Varchar(15)	Not null	To store pin code
State	Varchar(15)	Not null	To store state
Phone no	Varchar(15)	Not null	To store phone no
Username	Varchar(10)	Not null	Login
Password	Varchar(8)	Not null	Password
Type	Varchar(8)	Not null	Usertype
Proof	Varchar(8)	Not null	Identity

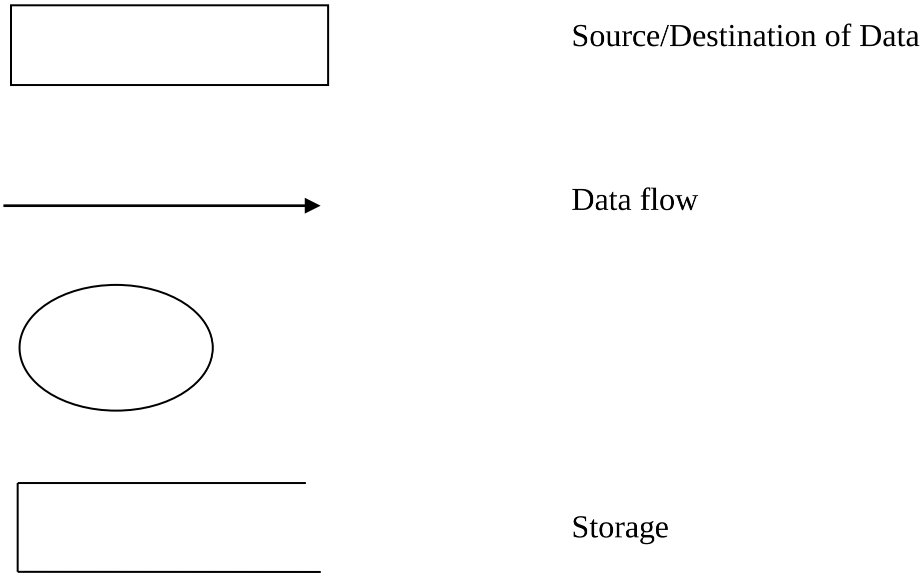
3.2 ARCHITECTURAL DESIGN

The architectural design develops a modular program structure and represents the control relationships between modules. It also defines interfaces that enable data to flow throughout the program.

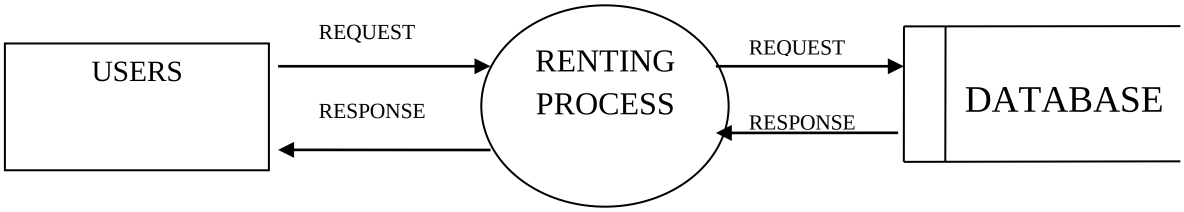
3.2.1. DATA FLOW DIAGRAM

A data flow diagram is a graphical technique that depicts data flow and transforms that are applied as data move from input to output. The DFD is used to represent increasing information flow and functional details. A Level 0 DFD also called a fundamental system model or context model represents the entire software elements as a single bubble with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in the next level i.e., Level 1 DFD. Each of the processes represented at level 1 is sub-functions of the overall system depicted in the context model.

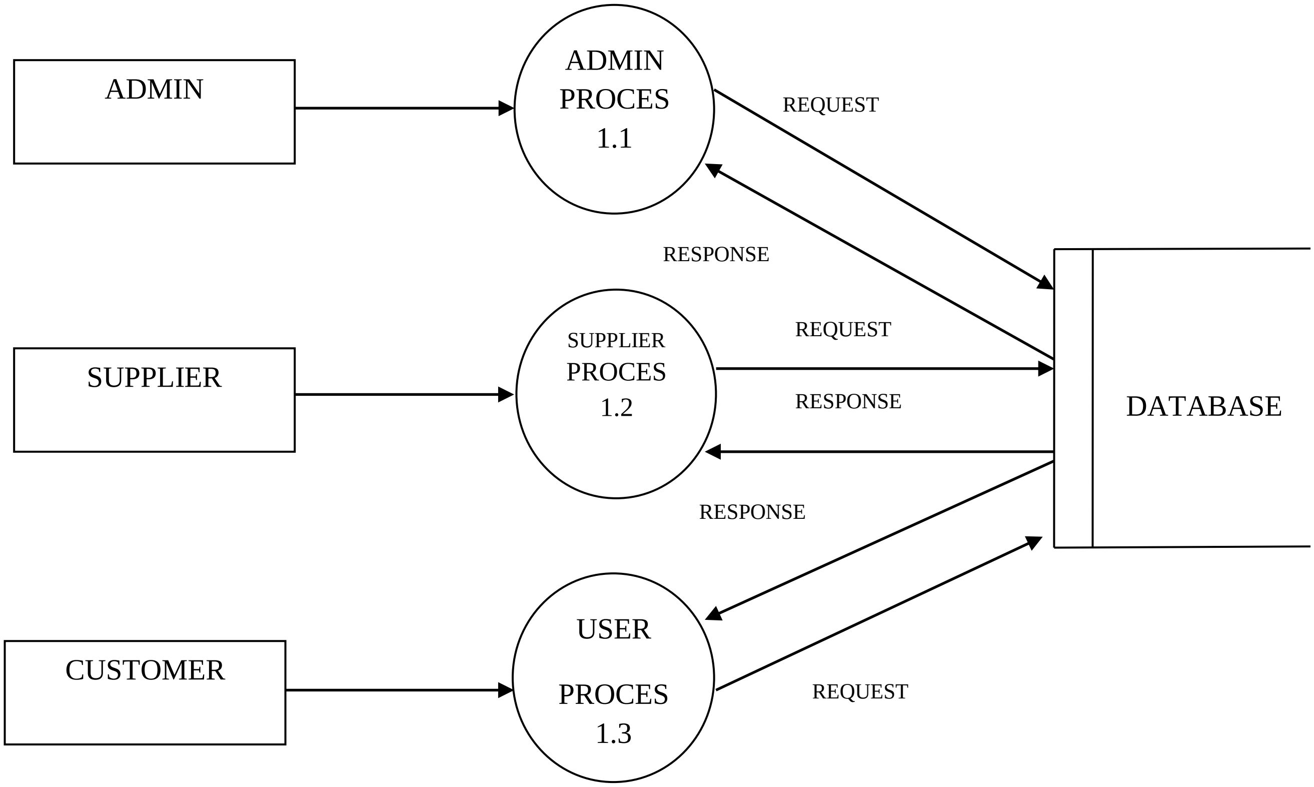
Data flow diagram symbol:



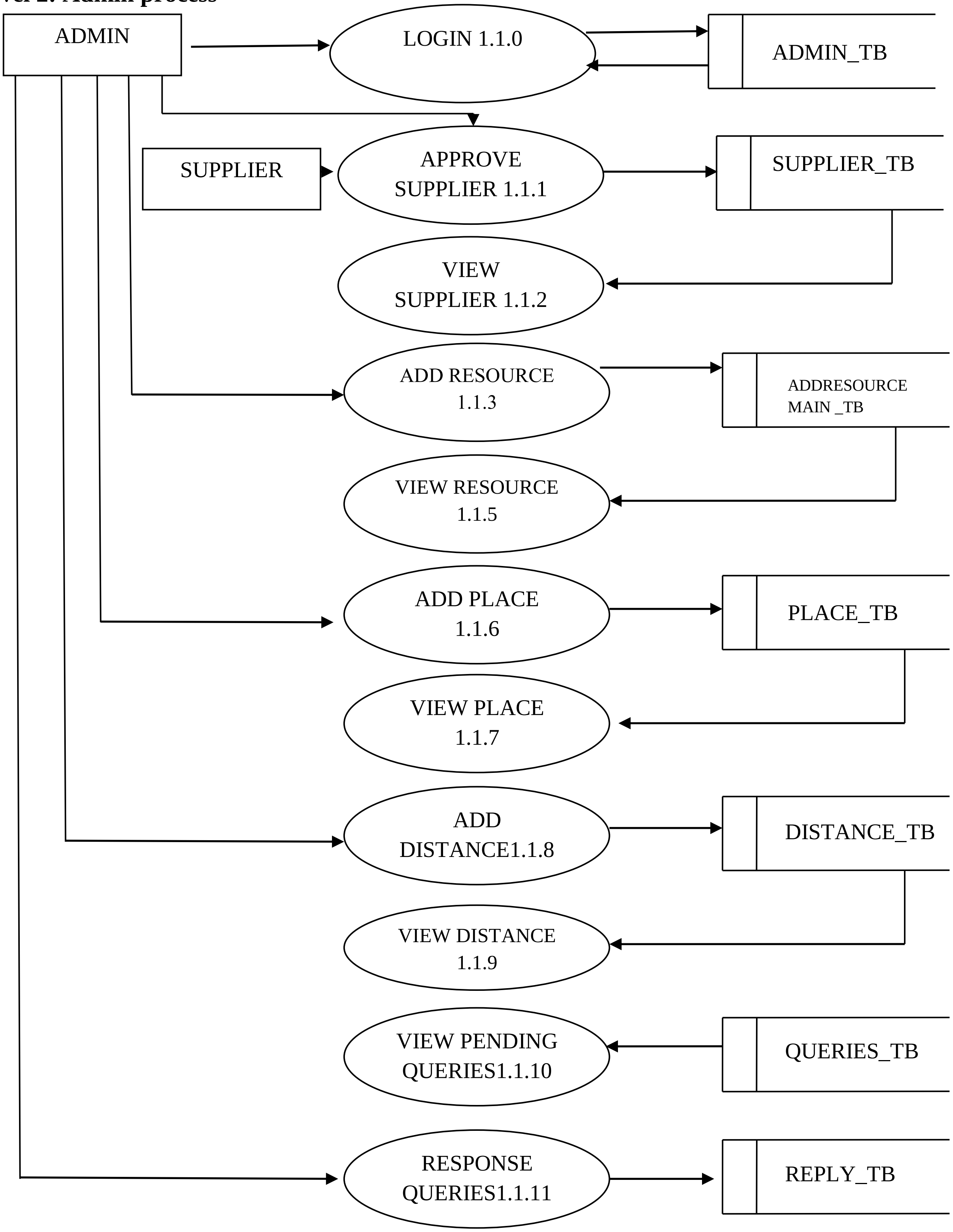
Level 0: Context Level



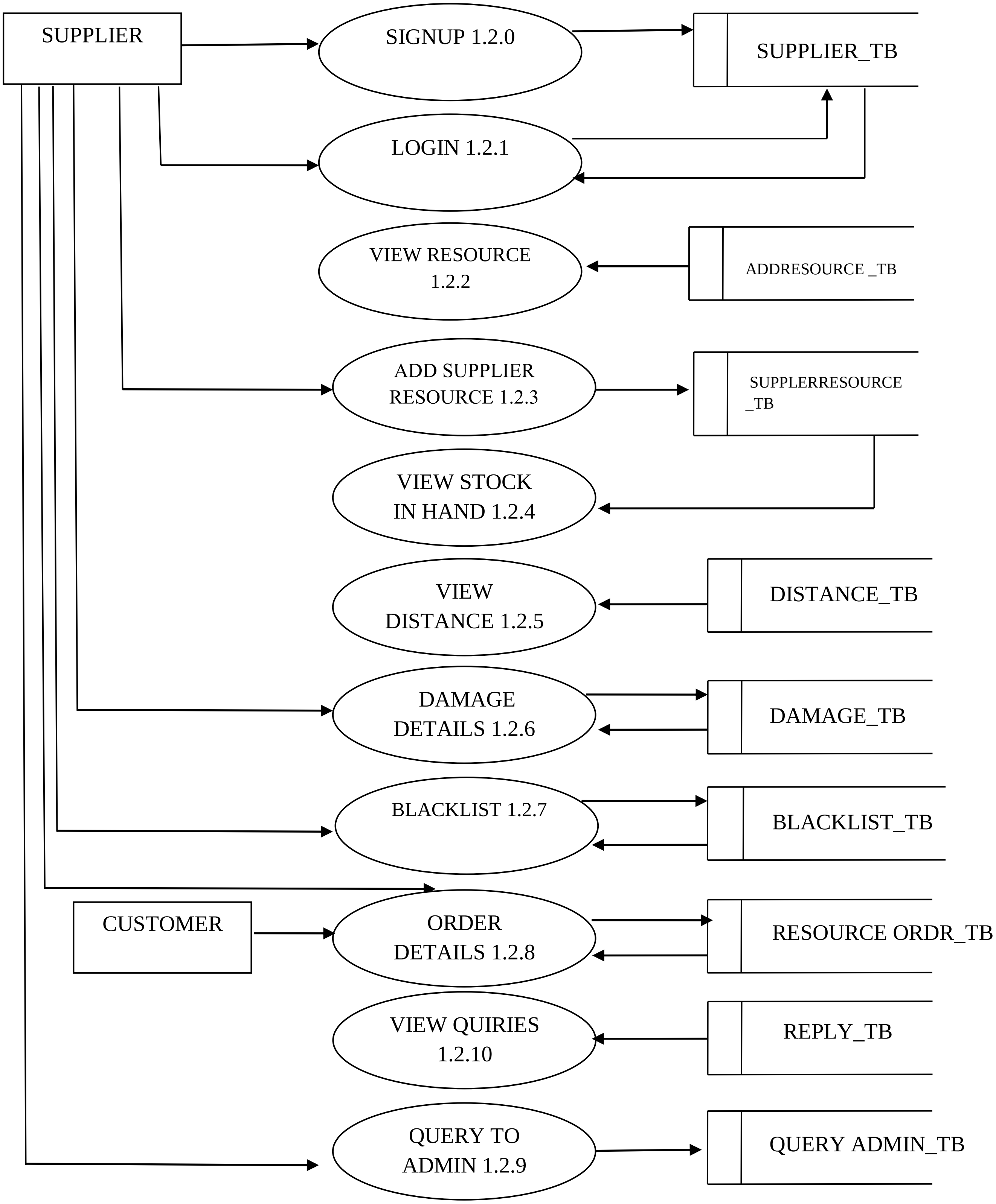
Level 1Hiring process



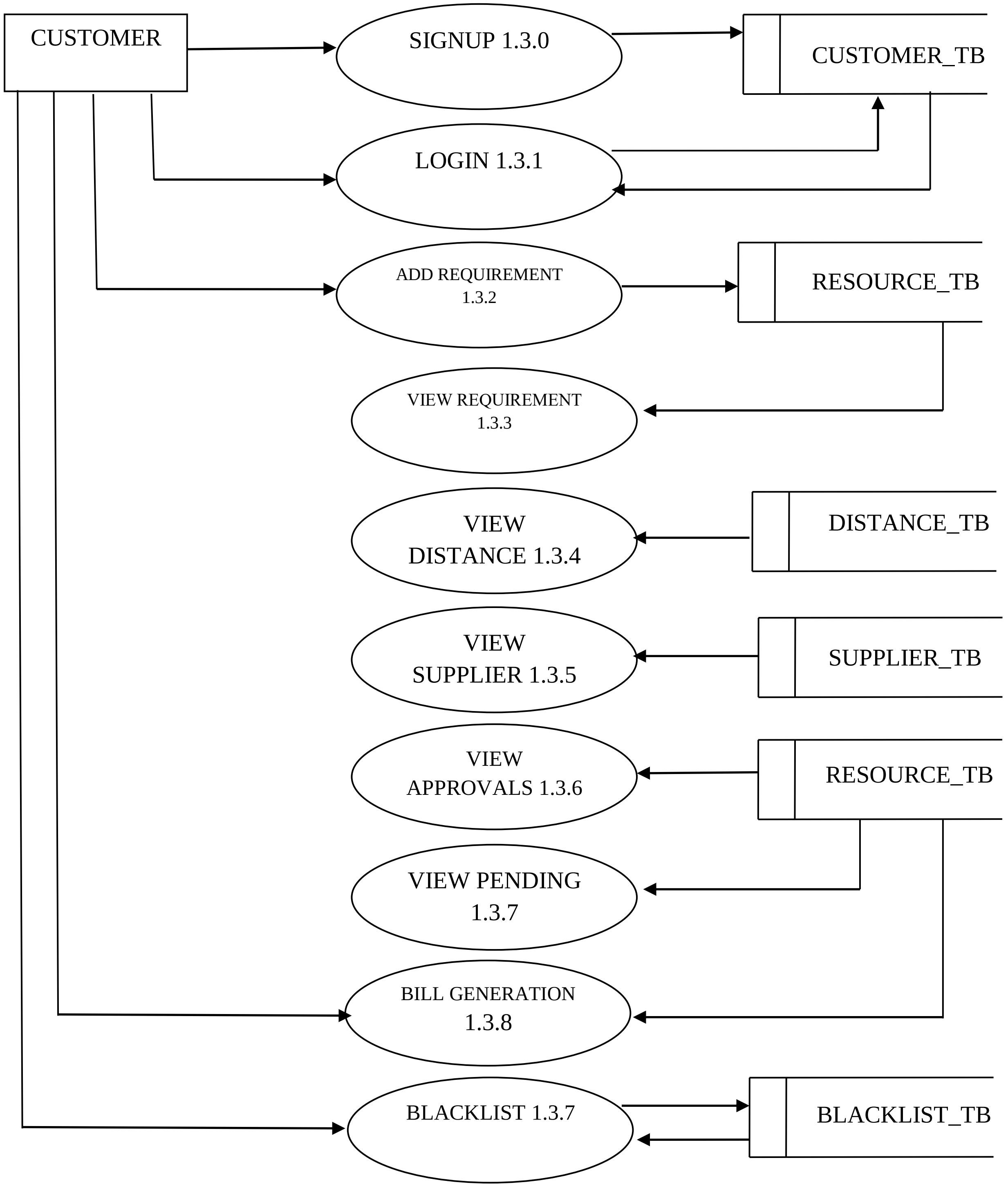
Level 2: Admin process



Leval 2: Supplier process

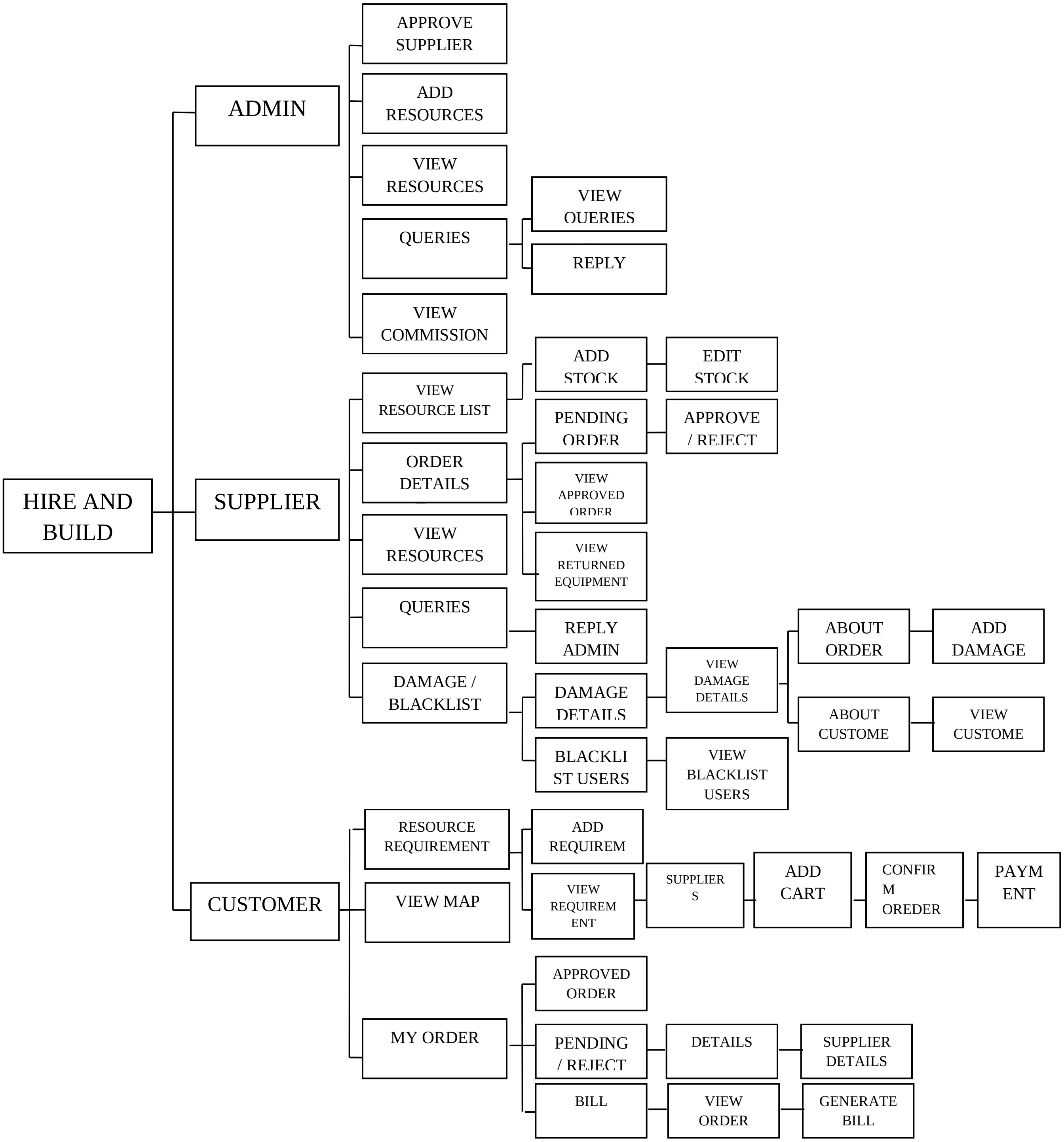


Leval 2: Customer process



3.2.2. HIERARCHICAL DIAGRAM

The hierarchical diagram is a technique for representing the modules of a system as a hierarchy and for documenting each module. It was used to develop requirements, construct the design, and support the implementation of an expert system to demonstrate and verify the system. Structure charts can be used to display several types of information.



3.3. INTERFACE DESIGN

An interface design elements for the software tell how information flows into and out of the system and how it is communicated among the components as part of the architecture.

3.3.1 INPUT DESIGN

Input design is the link between the information system and users and those steps that are necessary to put transaction data into a usable form for processing data entry. Instructing the computer to read data from a written printed document can active the activity of putting data into the computer for processing or it can occur by keying data directly into the system. The design of input focusing on controlling the errors, avoid delay and keeping the process simple. The system analyst decides the following input design details.

- What data to input?
- What medium to use?
- How the data is arranged and coded?

In my project named ‘HIRE AND BUILD’, I tried to include the following design constraints provided in software engineering.

1: Avoid scattering of fields in the forms

In all forms of the software, the textboxes (which provided to input some data), label (which label the text boxes), combo box (list a set of values), etc all are arranged in a neat and well format. It provides a simple look to the pages. The buttons are placed at the bottom of the page and easily accessible to the user. The menus are arranged below the heading and at a minimum level of menus are arranged with pages. The menu provides continuity to the pages.

2: User only needs to enter a minimum amount of data

All forms contain a minimum amount of data, but most essentials. No page provides or wanted bulky of data. It provides more easiness to the user. It creates more software to the end-user. Also, the operation continues by a single click.

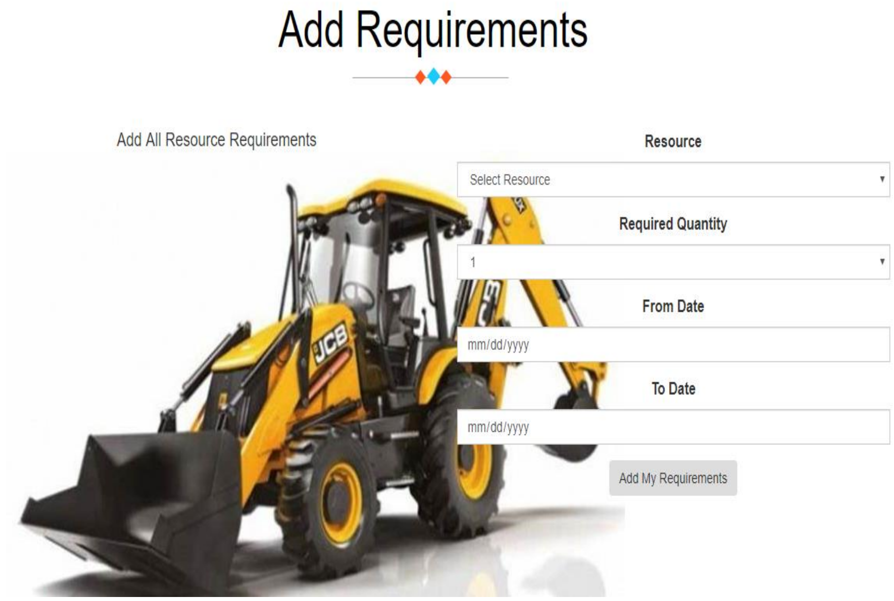
3: Avoid confusion in the forms

All forms have a well-defined menu and each menu name indicates its purpose. So the user can easily access various forms without confusion. Each form and its subforms are well labelled. So the user can easily identify the forms and work on that.

The following are the input forms present in this project:

- Supplier registration form
- Customer registration form
- User login forms
- Supplier registration form
- Add equipment form (Admin)
- Add equipment form (Supplier)
- Add requirement form
- Add booking form
- Approved Booking form
- Add distance form
- Add place form
- Add queries form
- Add map form
- Add damage form
- Add blacklist form
- Add equipment form for supplier
- Profile updating forms

E.g. Input design to add the requirements:



Add Requirements

Add All Resource Requirements

Resource

Select Resource ▾

Required Quantity

1 ▾

From Date

mm/dd/yyyy

To Date

mm/dd/yyyy

Add My Requirements

3.3.2 OUTPUT DESIGN

Designing a computer should proceed in well thought out manner. The term output means any information produced by the information system whether printed or displayed. Output design is a process that involves designing necessary outputs that have to be used by various users according to requirements. The efficient intelligent output design should remove the system relationship with the users and help in decision making.

When designing the output, system analyst must accomplish the following:

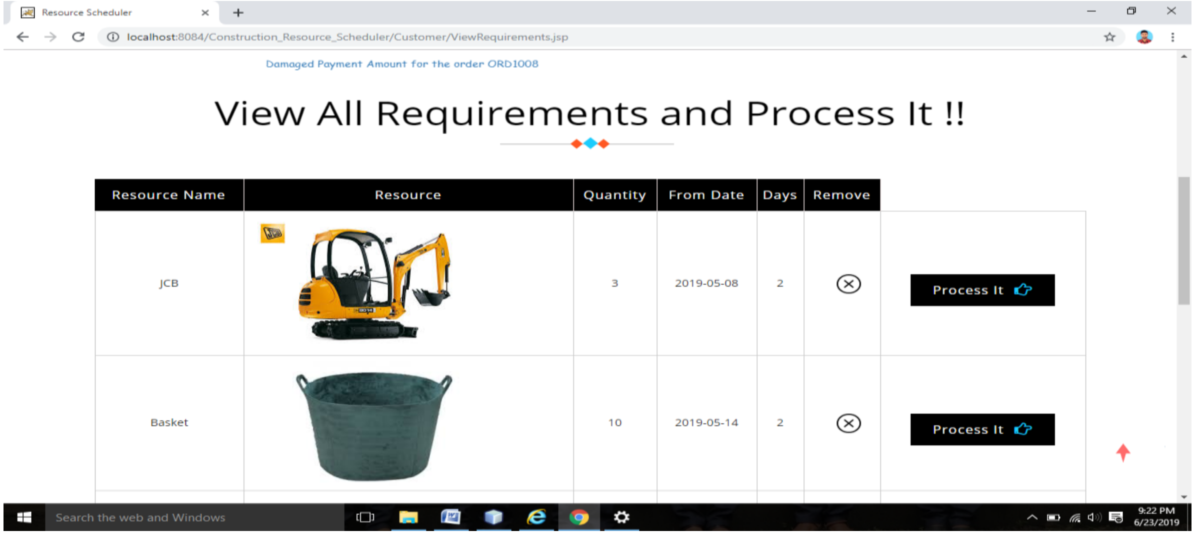
- Determine the information present
- Decide whether to print, display the information and select output medium
- Arrange information in an acceptable format.

In my project, the outputs are in the form of reports. They are well format and it provides the output in a correct and neat format.

The following are the output forms present in this project:

- Form for view equipment details
- Form for view booking details
- Form for view order details
- Form for view pending details
- Form for view returned equipment details
- Form for view map
- Form for view supplier details
- Form for view my order details
- Form for view place details
- Form for view distance details
- Form for view blacklisting customer details
- Form for view customer details
- Form for view payment details
- Form for view commission details
- Form for view premium for customer details
- Form for view bill creation details

E.g. Output design for viewing added requirements for the user details


Resource Scheduler
localhost:8084/Construction_Resource_Scheduler/Customer/ViewRequirements.jsp
Search the web and Windows
9:22 PM
6/23/2019

Damaged Payment Amount for the order ORD1008

View All Requirements and Process It !!

Resource Name	Resource	Quantity	From Date	Days	Remove	
JCB		3	2019-05-08	2	⊗	Process It
Basket		10	2019-05-14	2	⊗	Process It

3.4 PROCEDURAL DESIGN

The procedural design determines the modules included in the whole project which help us to identify the major functions.

MODULE SPECIFICATIONS

The following are the modules in this application.

- Admin
- Supplier
- Customer

1. Admin Module

Admin is one who manages the suppler. He gets the certain amount of money after hiring process.

Following are the main functions that can be performed by Center Admin:

- Supplier approval: Admin can view the pending requests of supplier he can check the profile and add or reject the supplier into the website.
- Add resource: Admin can add the equipment list.
- View equipment: Admin can view the registered equipments by the supplier.
- Add the place: Admin can add the place.
- View place: Admin can view added place.
- Add the distance: Admin can add the distance.
- View distance: Admin can view distance.
- Add the queries: Admin can add the queries to supplier.
- View queries: Admin can view queries added by the supplier.
- View commission: Admin can view commission for every hiring process .

2. Supplier Module: -

Following are the main functions that can be performed by supplier:

- View equipment list: Supplier can view equipment list added by the admin.
- Add equipment: Supplier can add the equipment in his hands.
- View stock: Supplier can view balanced stock.
- View distance: He can view the distance.
- View damage list: He can view the damage list of equipment.
- Add black list: He can add the blacklist.
- View pending order: He can view the pending orders.
- View approved order: He can view the approved orders.
- View returned orders: He can view the returned orders.
- Add the queries: Admin can add the queries to supplier.
- View queries: Admin can view queries added by the supplier.

3. Customer Module

Following are the main functions that can be performed by customer:

- Add requirement: He can add the requirement.
- View requirement: View his requirement.
- Book the equipment: customer can book the equipment.
- Make the payment: customer can make the payment.
- View distance: He can view the distance.
- View map: customer can view map.
- View approved orders: customer can view approved orders.
- View pending or rejected orders: customer can view pending or rejected orders.
- Bill print: customer can print bill.

CHAPTER-4

CODING

4. CODING

4.1. ABOUT THE SOFTWARE TOOLS USED

4.1.1 JAVA

Java is a general-purpose programming language that is class-based, object-oriented (although not a pure OO language, as it contains primitive types), and designed to have as few implementation dependencies as possible. It is intended to let application developers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them. As of 2018, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers.

Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GNU General Public License. Meanwhile, others have developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

The latest versions are Java 12, released in March 2019, and Java 11, a currently supported long-term support (LTS) version, released on September 25, 2018; Oracle released for the legacy Java 8 LTS the last free public update in January 2019 for commercial use, while it will otherwise still support Java 8 with public updates for personal use up to at least December 2020. Oracle (and others) highly recommend that you uninstall older versions of Java, because of serious risks due to unresolved security issues. Since Java 9 (and 10) is no longer supported, Oracle advises its users to immediately transition to Java 11 (Java 12 is also a known-LTS option)

Features of JAVA

- **Object Oriented**

In Java, everything is an Object. Java can be easily extended since it is based on the Object model.

- **Platform Independent**

Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform-independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

- **Simple**

Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.

- **Secure**

With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

- **Architecture-neutral**

Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

- **Portable**

Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. The compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

- **Robust**

Java makes an effort to eliminate error-prone situations by emphasizing mainly on compile time error checking and runtime checking.

- **Multithreaded**

With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.

- **Interpreted**

Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.

- **High Performance**

With the use of Just-In-Time compilers, Java enables high performance.

- **Distributed**

Java is designed for the distributed environment of the internet.

- **Dynamic**

Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects at run-time.

4.1.2 . SPRING FRAMEWORK

The Spring Framework is probably the most well-known Java framework out there, with a huge ecosystem and an active community around it. It allows you to build enterprise-level Java applications, web services, and microservices.

The Spring Framework started as a dependency injection tool, but over the years it has developed into a full-scale application framework. It provides you with an all-inclusive programming and configuration model that comes with support for generic tasks such as establishing a database connection or handling exceptions. Besides Java, you can also use the framework together with Kotlin and Groovy, both of which run on the Java Virtual Machine.

4.1.3 JQUERY

jQuery is a concise and fast JavaScript library that can be used to simplify event handling, HTML document traversing, Ajax interactions and animation for speedy website development. jQuery simplifies the HTML's client-side scripting, thus simplifying Web 2.0 application development.

jQuery is a free, open-source and dual-licensed library under the GNU General Public License. It is considered one of the favorite JavaScript (JS) libraries available today. As of 2012, it is used by more than half of the Web's top sites.

jQuery was introduced in January 2006 by John Resig at BarCamp NYC.jQuery also offers functionality that allows developers to build plug-ins, in addition to the JavaScript library. This allows for the development of abstractions for animation and low-level interaction, sophisticated effects and themeable, high-level widgets. The modular mechanism of the jQuery library facilitates the development of highly effective, potent Web applications and Web pages.

The jQuery library provides several user-friendly strategies and functions for rich application development. Because the functions of jQuery are simple, it is very popular among developers. jQuery may be used in all Web-based applications, in spite of the technology. It may be used with ASP, PHP, JSP, CGI, Servlets and most Web programming languages.

4.1.4 . JAVASCRIPT

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **LiveScript,** but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

The ECMA-262 Specification defined a standard version of the core JavaScript language.

4.1.5 . Microsoft SQL Server 2008

Microsoft SQL Server 2008 provides the Microsoft Windows Server System integrated server software with a database platform for the next generation of connected, scalable, and reliable enterprise applications. The breadth and depth of innovation in this version is in response to the needs of customers. This white paper is targeted to database administrators, to give you an understanding of the new features in and capabilities of SQL Server 2005. From many enhancement of existing features to an entirely new security model, database administrative is now more productive and in tune with the needs of the administrator.

At the core of SQLServer 2008 are new infrastructure application capabilities. SQL Service Broker is a distributed application framework that provides a new form of scalability and reliability for asynchronous message delivery. Though not new, Microsoft SQL Server Notification Services, Reporting Services, and SQL Server Mobile Edition (formerly called SQL Server CE) are all greatly enhanced in SQL Server 2005.

SQL (Structured Query Language) is a database computer language designed for the retrieval and management of data in relational database scheme creation and modification, and database object access control management.

4.2. CODING PRINCIPLE

The input to the coding phase is the design document. During the coding phase, modules identified in the design document are coded according to the module specification. Objectives of the coding phase are, to transform a design into code and unit test the code.

4.2.1 Coding Guidelines

- The code should be easy to understand.
- Don’t take pride in cryptic code.
- The code should be well documented.
- Comments should be present.
- Functions should be small.
- Do not use the Go-to statement.
- Do not use the same variable for multiple purposes.

4.3 SAMPLE CODE

Code for validation admin adding resource page using javascript

```
<script>
        addEventListener("load", function () {
              setTimeout(hideURLbar, 0);
        }, false);
        function hideURLbar() {
              window.scrollTo(0, 1);
        }
</script>
    <script type="text/javascript">
     function SuppliersValidation()
      {
         var numeric=/^[0-9]+$/;
        var alpha=/^[a-z A-Z]+$/;
        var beta=/^[a-z A-Z 0-9\.]+$/;
        var email=/^[\w\-\.\+]+\@[a-z A-Z 0-9\.\-]+\.[a-z A-Z 0-9]{2,4}$/;
        var name=  document.getElementById("txtfullname").value;
        if(name.length==0)
        {
              alert("First Name field is required");
              return false;
         }
        else if(!(name.match(alpha)))
         {
                     alert("First Name is invalid");
              return false;
         }
        var address=  document.getElementById("txtcadddr").value;
        if(address.length==0)
        {
              alert("address is required");
              return false;
        }
        var fcontact=  document.getElementById("txtcontact1").value;
        if(fcontact.length==0)
        {
              alert("Contact  is required");
              return false;
        }
        else if(!(fcontact.match(numeric)))
        {
                     alert("Contact is invalid");
                     return false;
        }
         var scontact=  document.getElementById("txtcontact2").value;
         if(scontact.length==0)
        {
              alert("Contact  is required");
              return false;
```

```
                }
                else if(!(scontact.match(numeric)))
               {
                            alert("Contact is invalid");
                           return false;
                 }
}
</script>
<script>
                window.onload = function () {
                document.getElementById("password1").onchange = validatePassword;
                document.getElementById("password2").onchange = validatePassword;
                }
                function validatePassword() {
                           var pass2 = document.getElementById("password2").value;
                           var pass1 = document.getElementById("password1").value;
                           if (pass1 != pass2)

     document.getElementById("password2").setCustomValidity("Passwords Don't Match");
                           else
                                      document.getElementById("password2").setCustomValidity('');
                }
</script>
```

Including jquery code for scroll animation

```
<script>
                jQuery(document).ready(function ($) {
                           $(".scroll").click(function (event) {
                                      event.preventDefault();

                                      $('html,body').animate({
                                                 scrollTop: $(this.hash).offset().top
                                      }, 1000);
                           });
});
```

```
        </script>
```

Adding requirement by admin JSP code

```
<form action="../addResourceMainAction" method="post" class="f-color"
enctype="multipart/form-data">
    <div class="form-group">
            <label>Resource</label>
            <input type="text" class="contact-formw3ls form-control" name="txtres"
     id="txtres" required>
    </div>
    <div class="form-group">
            <label>Description</label>
            <textarea class="contact-formw3ls form-control" rows="5" id="txtdesc"
            name="txtdesc" required></textarea>
    </div>
    <div class="form-group">
            <label>Upload</label>
            <input type="file" class="contact-formw3ls form-control" name="file" id="file"
     required />
    </div>
    <button type="submit" class="btn submit contact-submit">Submit</button>
</form>
```

Java servlet code for adding resource

```
out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet addResourceMainAction</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet addResourceMainAction at " + request.getContextPath() +
"</h1>");
        boolean isMultipart = ServletFileUpload.isMultipartContent(request);
   if (!isMultipart)
```

```
{
}
else
{
  FileItemFactory factory = new DiskFileItemFactory();
  ServletFileUpload upload = new ServletFileUpload(factory);
  List items = null;
  try
  {
          items = upload.parseRequest(request);
  }
  catch (FileUploadException e)
  {
    out.print("FileUploadException");
          e.printStackTrace();
  }
  Iterator itr = items.iterator();
String filename="",resource_name="",desc="";
  while (itr.hasNext())
  {
      FileItem item = (FileItem) itr.next();
      if (item.isFormField())
       {
                 String name = item.getFieldName();
                 String value = item.getString();
                  if(name.equals("txtres"))
                  {
                          resource_name=value;
                  }
                else if(name.equals("txtdesc"))
                   {
                             desc=value;
                   }
    }
```

```
            else
             {
              String
path="F:\\IMSAY\\PROJECTS\\nProjectz_Copy\\No\\ReadyMade_JSP\\Construction_Resourc
e_Scheduler\\web\\admin\\ResouceUploads\\";
                   String itemName = item.getName();
                    //filename=request.getRealPath("") + "/ResouceUploads/";
                   filename=path+itemName;
                    out.println("fname"+filename);
                   File savedFile=new File(filename);
                    try {
                        item.write(savedFile);
                    } catch (Exception ex) {

Logger.getLogger(addResourceMainAction.class.getName()).log(Level.SEVERE, null, ex);
                    }
                    out.println("successfull");
                String s="select * from addresoucemain_tb where
ResourceName='"+request.getParameter("txtres")+"' ";
            ResultSet rs=dm.ForSelect(s);
            try {
                try {
                    if(rs.next())
                    {
                            out.print("<script type='text/javascript'>alert('Resource already exists...');
      window.location='index.jsp'; </script>");
                   }
                    else
                    {
                String resource_mainid=dm.GenID("addresoucemain_tb","Resource_id","REM");
                     String sql="insert into addresoucemain_tb
values('"+resource_mainid+"','"+resource_name+"','"+desc+"','"+itemName+"') ";
            int r=dm.ForInsert(sql);
              if(r>0)
            {
              out.print("<script type='text/javascript'>alert('Resource has been added');
    window.location='admin/addResourcemain.jsp'; </script>");
    }
              }
            } catch (SQLException ex) {
              Logger.getLogger(addResourceMainAction.class.getName()).log(Level.SEVERE,
    null, ex);
            }
          out.println("</body>");
          out.println("</html>");
        } finally {
          out.close();
        }
      }
    }
  }
}
```

CHAPTER-5

TESTING

5. SYSTEM TESTING

For software that is newly developed, primary importance is given to testing the system. It is the last opportunity for the developer to detect the possible errors in the software before handing over it to the customer. Testing is the process by which the developer will generate a set of data, which gives the maximum probability of finding all types of errors that can occur in the software.

The various steps in testing the system can be listed as below:

1. Running the program to identify any errors that might have occurred while feeding the program into the system.

2. Applying the screen formats to regulate users to extend, so that the screens are comprehensible to the user.

3. Presenting the formats to the administration to obtain approval and check if any modification has to be done. Obtaining feedback from users and analyzing the scope for improvement.

4. Checking the data accessibility from the data server and whether any improvement is needed or not.

The following are the types of Testing:

- Unit Testing
- Integration Testing
- System Testing
- Validation Testing

5.1. UNIT TESTING

Unit testing is carried out to screen wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to locate in coding and logic.

In unit testing,

- The module interface is tested to ensure that information properly flows into and out of the program under test.
- Local data structures are examined to ensure that data stored temporarily maintains its integrity during all steps in algorithm execution.
- The boundary condition is tested to ensure that the module operates properly at boundaries established to limit or restrict processing.
- All independent paths through the control structures are executed to ensure that all statements in the module have been executed at least once.
- Error handling paths are also tested.

5.2. TEST CASES

Login Form

No:	**Test Scenario**	**Expected Result**	**Observed Result**	**Result**
1.	Enter the wrong user name and password.	Display login form again with a warning message.	Message displayed.	Pass
2.	Enter the correct user name and wrong password.	Display login form again with a warning message.	Message displayed.	Pass
3.	Enter the correct user name and password.	Administrator or other users can log in into the system.	The appropriate home page is displayed.	Pass
4.	Press the login button without filling the user name and password.	Display a warning message to fill the fields.	A warning message is displayed.	Pass

Supplier registration form

No:	Test Scenario	Expected Result	Observed Result	Result
1.	Form displayed.	Display the registration form.	Form loaded	Pass
2.	Enter the name in integers.	Display an invalid message.	An invalid message displayed.	Pass
3.	Enter the mobile number in characters.	Display an invalid message.	An invalid message displayed.	Pass
4.	Enter the mobile number more than and less than 10 integers.	Display an invalid message.	An invalid message displayed.	Pass
5.	Click the save button without filling the details.	Display a warning message to fill the details.	Warning message displayed.	Pass
6.	Click on the save button with filled fields.	Accept the details.	Registration is successfully done.	Pass
7.	Click cancel button	Clear all fields to blank.	All fields cleared.	Pass

View customer Details

No.	Test scenario	Expected result	Observed result	Result
1.	Form displayed.	Display form with all controls.	The form loaded with all controls.	Pass
2.	Click the add button without data.	Display a warning message to fill the details.	Warning message displayed.	Pass
3.	Click the save button with data.	Inserted successfully.	Added successfully.	Pass

5.3. INTEGRATION TESTING

Integration testing is an asymmetric technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. Unit tested module was taken and a single program structure was built that has been dictated by and tested in small segments, where errors were easy to locate and rectify. Each database or table manipulation operation was written as a single program that was tested again with numerous test data to check for its functionality.

No.	Input/procedure	Expected Result	Actual Result	Pass/Fail
1.	Check the value pass between different forms are an appropriate format	Appropriate operations of different forms.	Same as expected.	Pass

5.3. SYSTEM TESTING

System testing is used to test the entire system (Integration of all modules). It also tests to find the discrepancies between the system and the original objective, current specification and system documentation. The entire system is checked to correct deviation to achieve correctness.

No.	**Input/procedure**	**Expected Result**	**Actual Result**	**Pass/Fail**
1.	Check whether the indented output is obtained.	All operations are carried out properly.	Same as expected.	Pass

5.4. VALIDATION TESTING

After integration testing, the software is completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software tests begin validation test has been conducted one of the two possible conditions exists. One is the function or performance characteristics conform to specification and are accepted and the other is a deviation from the specification is uncovered and a deficiency list is created.

CHAPTER-6

IMPLEMENTATION

6.1. IMPLEMENTATION

System implementation is the stage where the theoretical design is turned into a working system. The system can be implemented only after thorough testing is done and if it is found to work according to specifications. The following methods were undergone.

- Testing developed programs with updating.
- Correction of errors identified.
- Creating the tables of the system with actual data.
- Making necessary changes with actual data.
- Doing a parallel run of the system to find out any errors identified and to correct them.
- Training of user personnel's.

The implementation method used to implement Smart Scheduling is Parallel Run. That is, the new system will work parallel to the existing system. The new system will replace the existing system completely.

Smart Scheduling is developed as a web application, as usual, some web development technologies are used in the implementation of the project. The language selected to program this software JAVA with JSP, javascript, jquery. The reason for selecting java is a simple and powerful language that especially developed to create a web application. And also java is an evergreen programming language it has a robust set of features, especially security. In this project, JSP pages are used to designing and java servlet pages are act as an action page.

Technologies used in the development of the software are:

Programming language	- JAVA
Database	- MySQLQuery Browser 1.2.17
Development tool	- NetBeans IDE 7.3
Development platform	- Windows 10

CHAPTER-7

CONCLUSION

7. CONCLUSION

To conclude the description of the project developed using JAVA with JSP, jQuery JavaScript and MySQLQuery Browser 2008 is based on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement. This will help the customers (Who are willing to involve any construction activities) to search the resources easily and effectively and book the items for the facility to rent equipment. Registered suppliers can participate in the auction process. Handing resources manually is impractical so we have to develop a system that will easily to get the equipment with lots of choices and minimum rent money. The developer of this website think like that ‘if a customer uses this website when he gets any benefit compared to the normal renting system ‘ then only he will use. So the developer applies the method linear programming problem in operation research. The concept of LPP is to maximize the profit with minimum input .so implement this concept here also by converting the mathematical format into the programming code. And also using some searching and sorting mechanisms. Anyway, the customer has lots of choices here and also he gets the equipment with the minimum cost at minimum distance. In the normal process, customers have only two or three chances finally he select the equipment as which one is nearest at a high cost. So in this case, if a customer using this system he can see lots of choices and select the appropriate supplier. In the case of a supplier, he has also benefited from using this website. Suppose if he has a lot of equipment is in his hand but he doesn't know who needs this equipment in this case if he uses this system he can present types of equipment to the society the customers will identify the suppliers so it will be more useful for the supplier. Here the admin also gets a certain benefit for using this system. I.e. if there any transactions are made with supplier and customer a particular amount of commission will be credited to the supplier account. So we can say that the system is a business-oriented.

CHAPTER-8

REFERENCES

8.0. REFERENCES

8.1. WEBSITE

- ✓ http://www.Equipment cafe.com
- ✓ http://www.PC earthmovers.com
- ✓ http://www.Equipment share.com

8.2. REFERENCE BOOKS

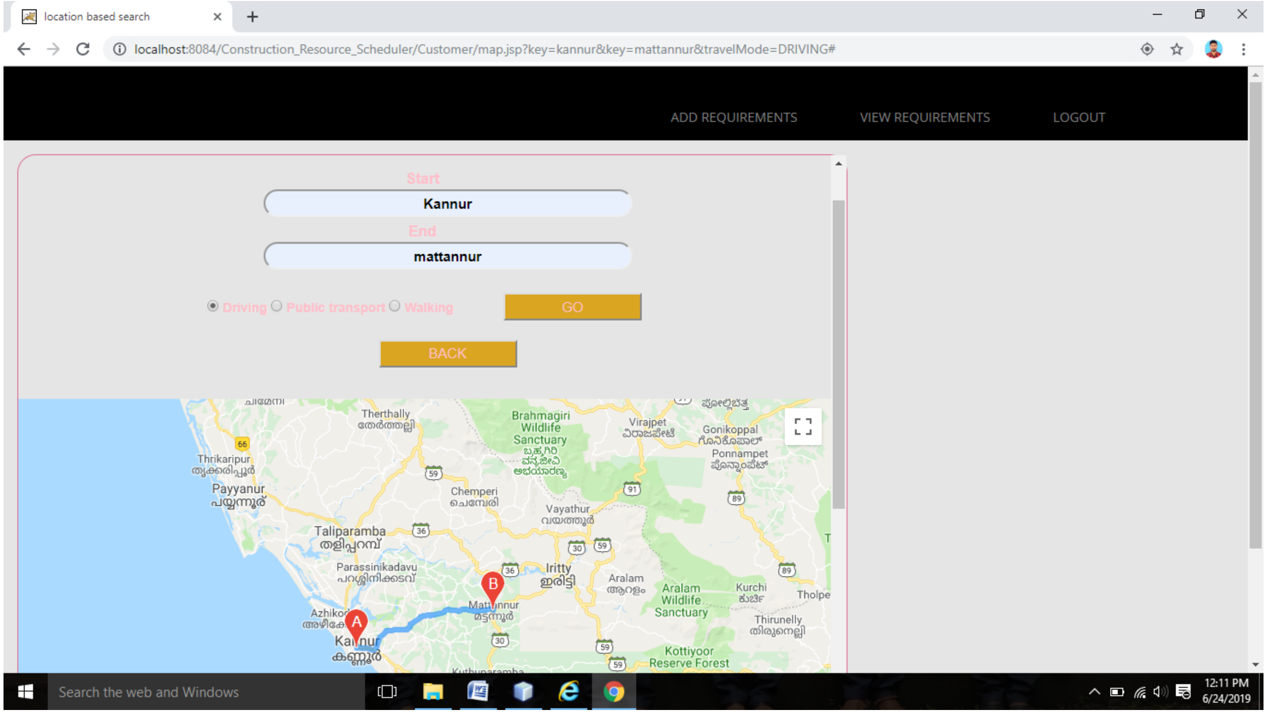
- ➢ Roger. S. Pressman, ‘Software Engineering’, Tata McGraw Hill, Fifth Edition, 2004.
- ➢ L.R Potty, ‘Operation Research'.
- ➢ Kanthi Swarup, ‘Operation Research'.

CHAPTER-9

APPENDIX

9.1. SAMPLE SCREEN WITH VALID DATA

1. Form for view the distance between source and destination.

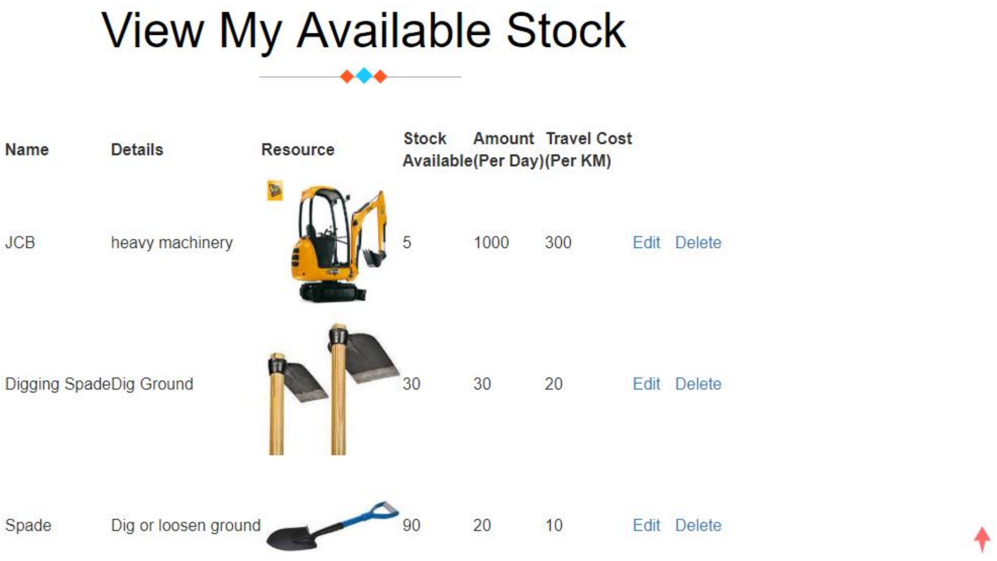


Start: Kannur

End: mattannur

● Driving ○ Public transport ○ Walking

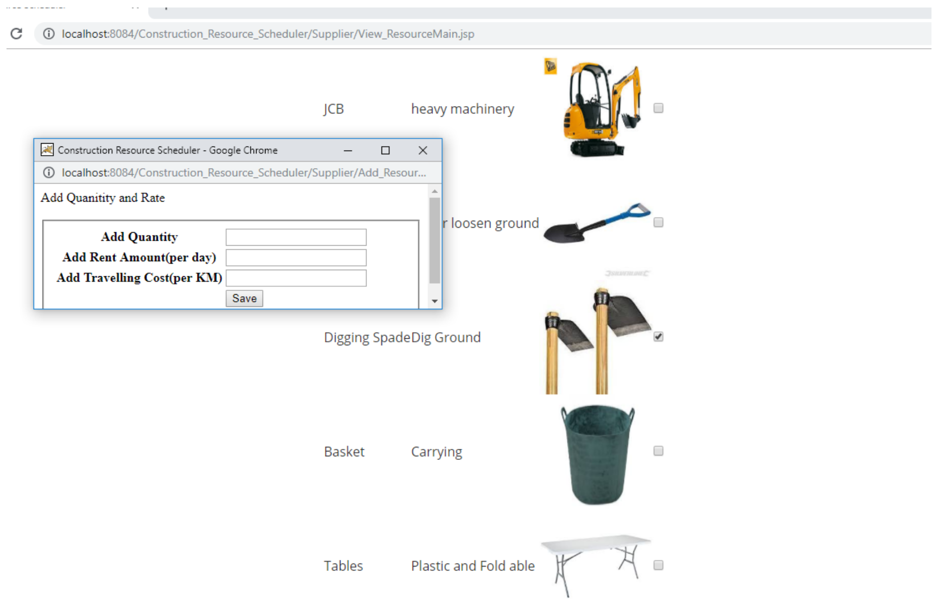
2. Supplier view stock in hand



View My Available Stock

Name	Details	Resource	Stock Available	Amount (Per Day)	Travel Cost (Per KM)		
JCB	heavy machinery		5	1000	300	Edit	Delete
Digging Spade	Dig Ground		30	30	20	Edit	Delete
Spade	Dig or loosen ground		90	20	10	Edit	Delete

3. Form for supplier adds resources.



localhost:8084/Construction_Resource_Scheduler/Supplier/View_ResourceMain.jsp

JCB	heavy machinery		☐
	r loosen ground		☐
Digging Spade	Dig Ground		☑
Basket	Carrying		☐
Tables	Plastic and Fold able		☐

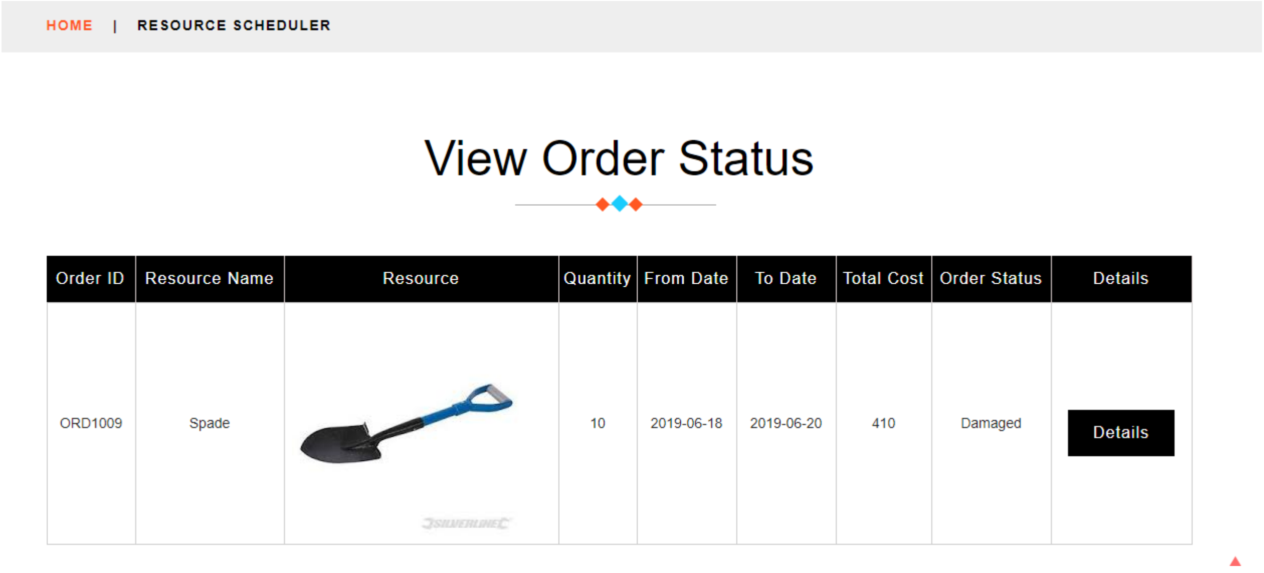
Construction Resource Scheduler - Google Chrome

localhost:8084/Construction_Resource_Scheduler/Supplier/Add_Resour...

Add Quanitity and Rate

Add Quantity	
Add Rent Amount(per day)	
Add Travelling Cost(per KM)	

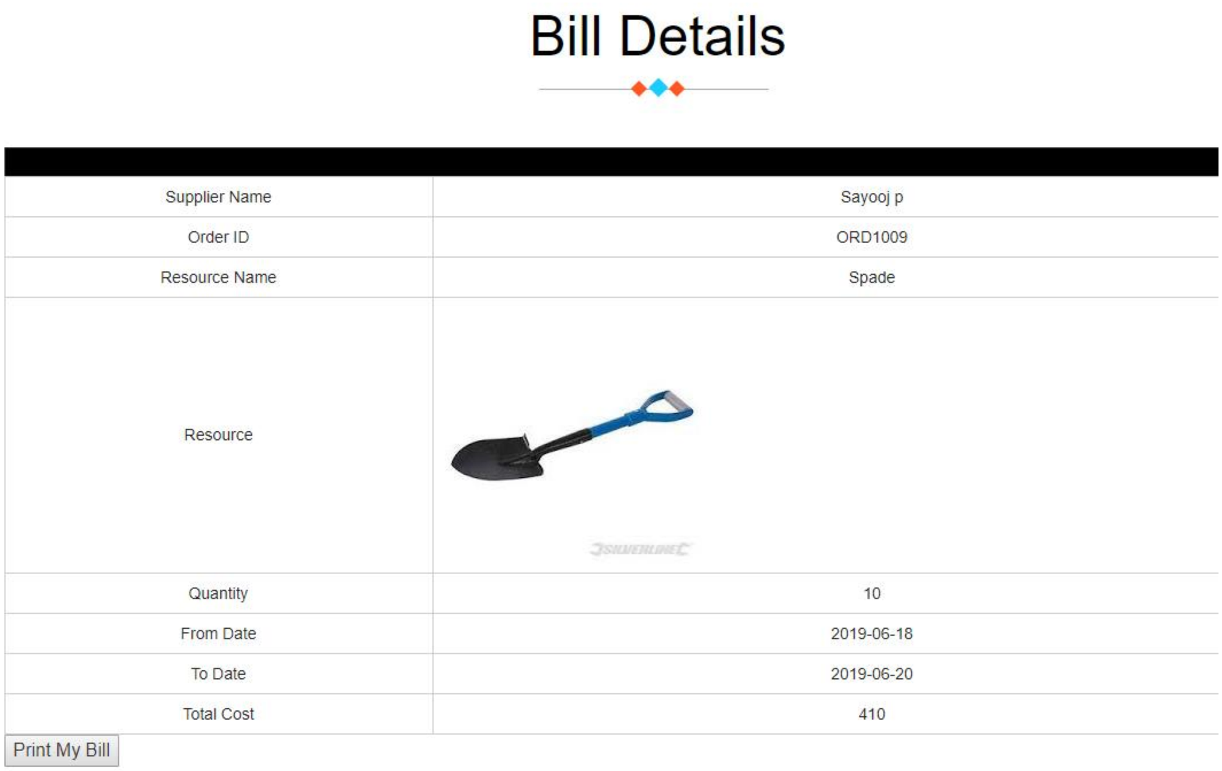
4. form for customer view details of orders



View Order Status

Order ID	Resource Name	Resource	Quantity	From Date	To Date	Total Cost	Order Status	Details
ORD1009	Spade		10	2019-06-18	2019-06-20	410	Damaged	Details

5. form for customer bill generation view details.



Bill Details

Supplier Name	Sayooj p
Order ID	ORD1009
Resource Name	Spade
Resource	
Quantity	10
From Date	2019-06-18
To Date	2019-06-20
Total Cost	410