Software Requirement Specification(SRS)
For
Online Buy And Rent Shopping System(OBRSS)

Introduction

1.1 Purpose:

The Online Buy And Rent Shopping System for products shop web application is intended to provide complete solutions for vendors
as well as customers through a single get way using the internet. It will enable vendors to setup online shops , customer to
browse through the shop and purchase and rent them online without having to visiting the shop physically.

1.2 Scope

This system allows the customers to maintain their product for add ,remove and choose rent option for the product over the internet
The system allows customers to check day wise price of the particular product that they want for rent and accordingly pay for that product.
Shoppers will be able to manage their products stock

Definitions:

OSS- Online shopping System (for electronics item shop)

SRS- Software Requirement Specification GUI- Graphical User Interface

Stack holder- The person who will participate in system Ex. Customer, Administrator, Visitor etc.

1.5 Overview:

This system provides an easy solution for customers to buy and rent the product without going to the shop and also to shop owner to sale the product.

This proposed system can be used by any naïve users and it does not require any educational level, experience or technical expertise in computer field but it will be of good use if user has the good knowledge of how to operate a computer

1. Overall Description:

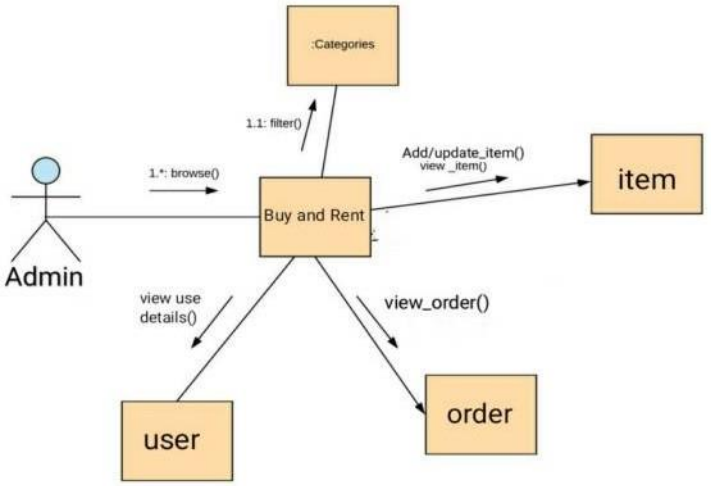
The Online buy and rent shopping system application enables vendor to set up online shops,customers to browse through the shops, and a system administrator to approve and reject requests for new user or existing user. Also the developer is designing an online shopping site to manage the items in the shop and also help customers to purchase them online with out visiting the shop physically.The online shopping system will use the internet as the sole method for selling goods to its consumers.

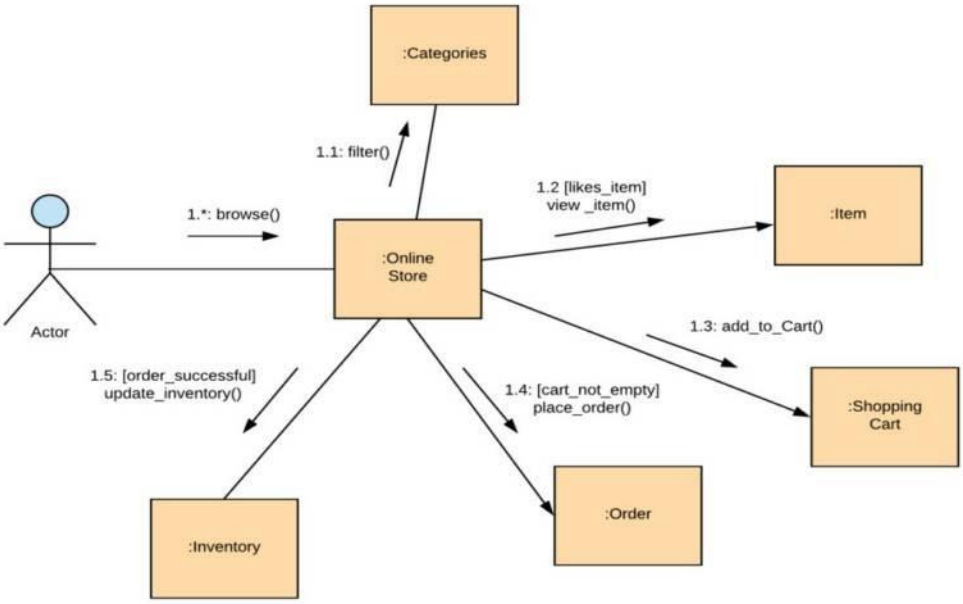
Product Perspective:

This product aimed toward a person who don't want to visit the shop as he might don't get time for that or might not interested in visiting there and dealing with lot of formalities.

Product Functions:

OBRSS should support this use case:





User characteristics:

User should be familiar with the terms like login , register , order system etc.

Principle Actors:

2 Principle Actors are Customer and Administrator.

General Constraints:

A full internet connection is required for OSS.

Assumptions and Dependencies: Working of OSS need Internet Connection.

2. **Specific Requirements**:

Functional Requirements:

This section provides requirement overview of the system.

Various functional modules that can be implemented by the system will be -

Description:

Registration:

If customer wants to buy the product then he/she must be registered, unregistered user can't go to the shopping cart.

Login:

Customer logins to the system by entering valid user id and password for the shopping.

Changes to Cart:

Changes to cart means the customer after login or registration can make order or cancel order of the product from the shopping cart.

Payment:

In this system we are dealing the mode of payment by Cash.We will extend this to credit card debit card etc in the future.

Logout:

After ordering or surfing for the product customer has to logout.

Non-Functional Requirements:

Following Non-Functional Requirements will be there in the insurance to the internet:

- Secure access to consumer's confidential data.
- 24X7 availability**.**
- Better component design to get better performance at peak time.
- Flexible service based architecture will be highly desirable for future extension Non-Functional Requirements define system properties and constraints

Various other Non-Functional Requirements are:

- Security
- Reliability
- Maintainability
- Portability
- Extensibility
- Reusability
- Compatibility
- Resource Utilization

Performance Requirements:

In order to maintain an acceptable speed at maximum number of uploads allowed from a particular customer as any number of users can access to the system at any time.
Also the connections to the servers will be based on the attributes of the user like his location and server will be working 24X7 times.

Technical Issues:

This system will work on client-server architecture. It will require an internet server and which will be able to run Angular or J2EE application. The system should support some commonly used browser such as IE, mozzila firefox,chrome etc.

Interface Requirement:

Various interfaces for the product could be-

- Login Page
- Registration Form
- There will be a screen displaying information about product that the shop having.
- If the customers select the buy button then another screen of shopping cart will be opened.

Software Interface:

- Operating System:Windows7 Ultimate which supports networking.
- JAVA development toolkit.

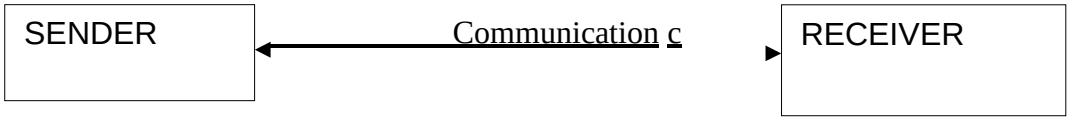
Hardware Interface:

Hardware requirements for insurance on internet will be same for both parties which are as follows:

Processor : Dual Core RAM:2 GB
Hard Disk : 320 GB
NIC:For each party

Communication Interfaces:

The two parties should be connected by LAN or WAN for the communication purpose.



3. System Design Specification:

Architecture Design:

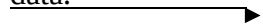
Data Flow Diagram(DFD):

It is a way of representing system requirements in graphical form this led to modular design A DFD describes a data flow(logical) rather than how they are processed .So they do not depend upon software , hardware , data structure or fileorganization It is also known as 'bubble sort'.

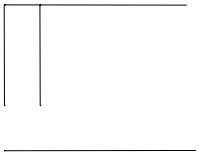
A DFD is a structured analysis and a design tool that can be used for flowcharting in place of ,or in association with ,information-oriented and process- oriented system flowcharts. A DFD is considered as an abstract of the logic of information-oriented or process-oriented system flowchart The four basic symbols used to construct data flow diagrams are-

A rectangle represents a data stination.

A directed line represents flow of data.



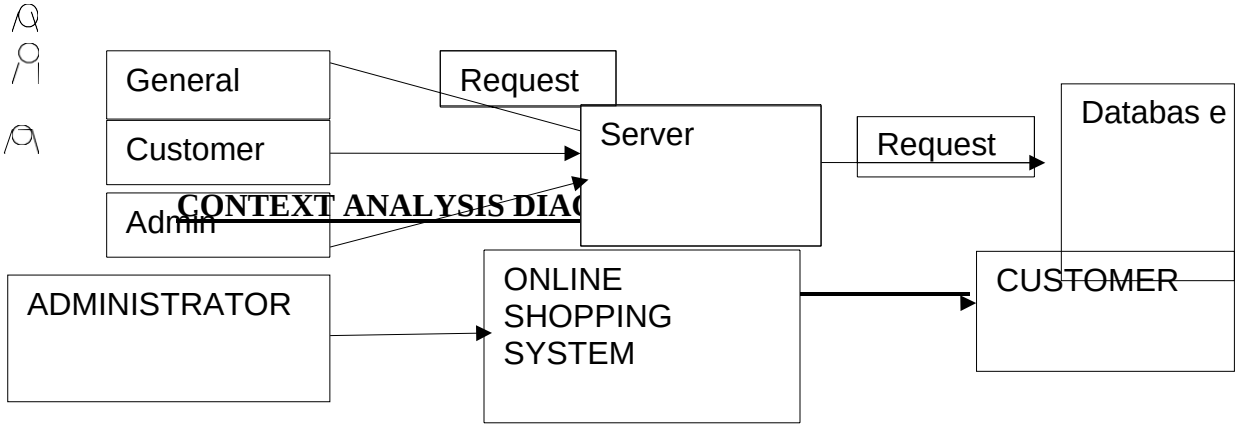
An Oval represents a process that streams.

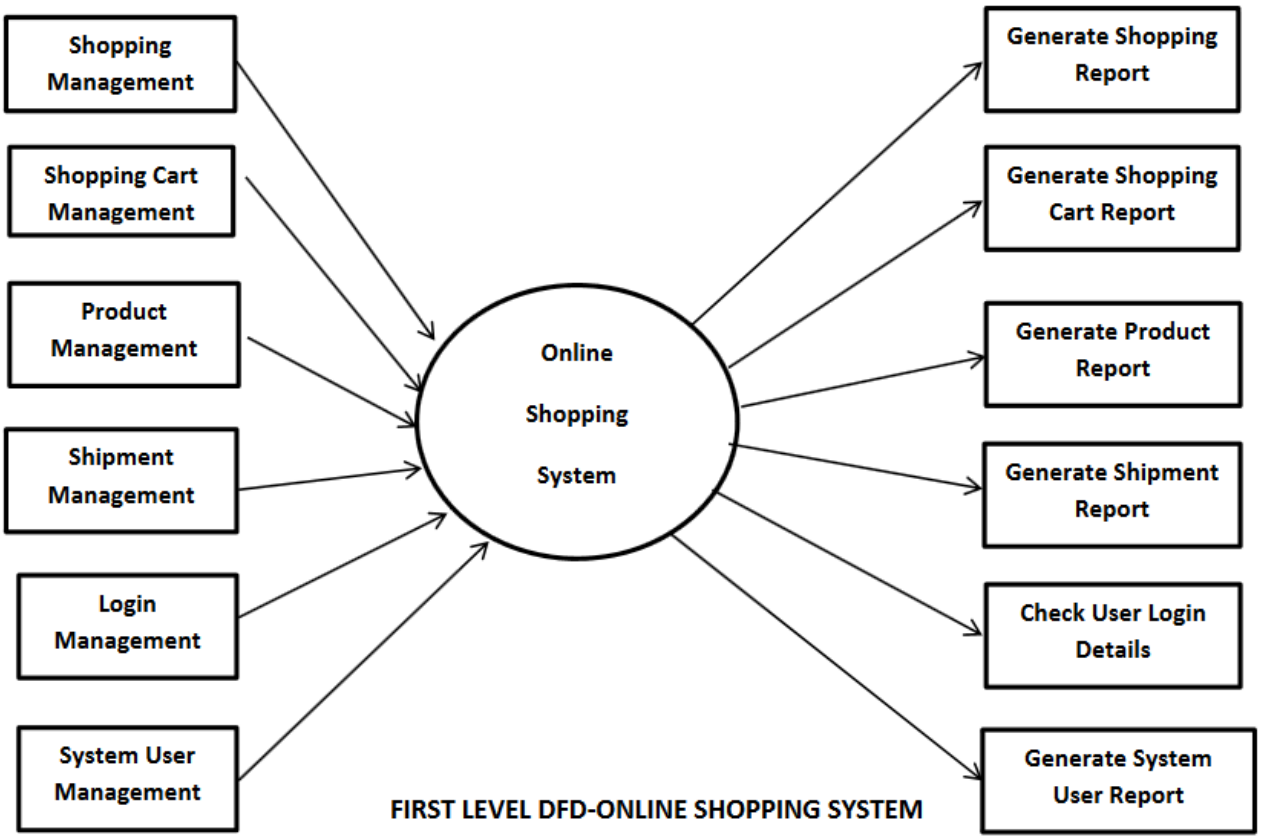


An Open ended rectangle represents storage.

The points at which data is transformed are called as nodes.The principle processes that take place at nodes are:

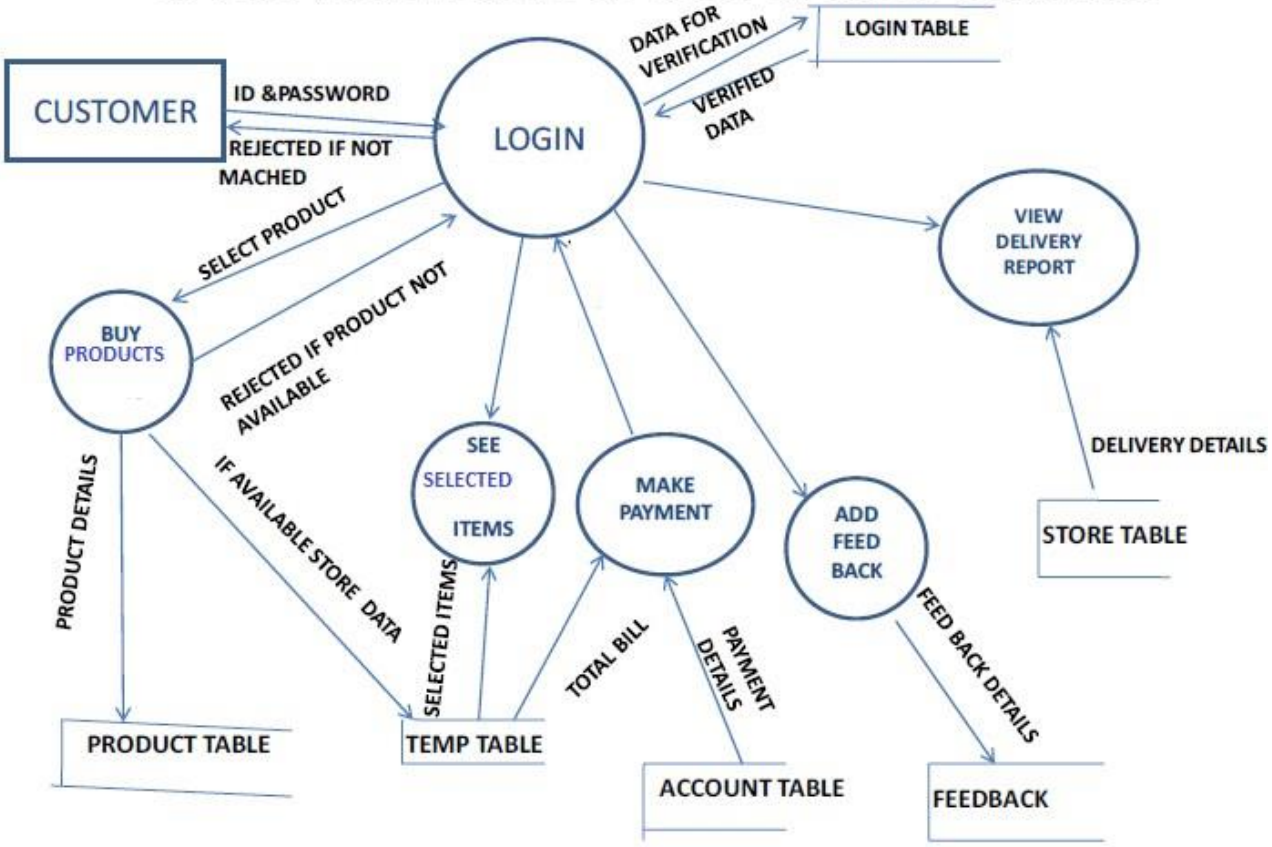
1.Combining data streams 2.Splitting data streams
3.Modifiying data streams



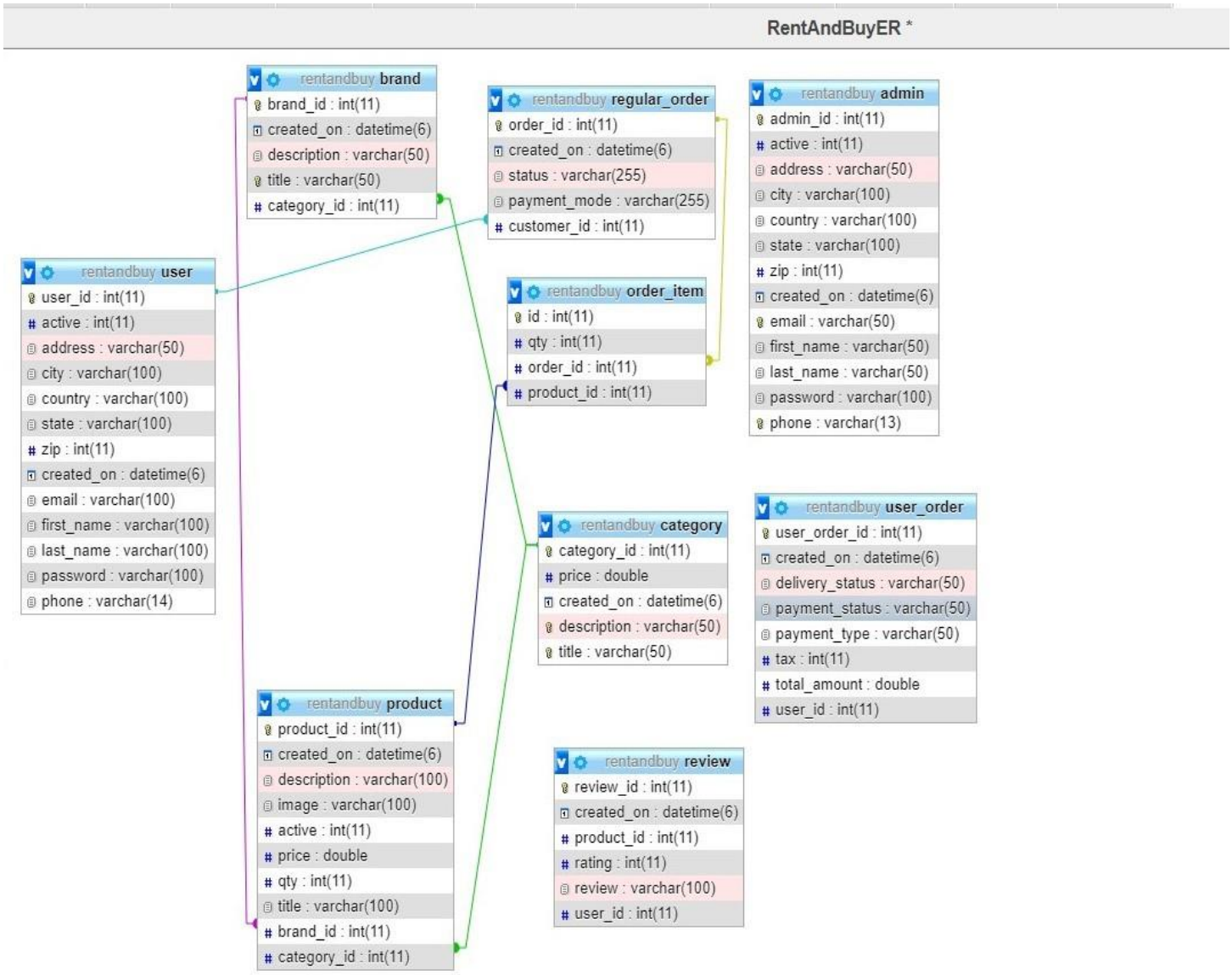


FIRST LEVEL DFD-ONLINE SHOPPING SYSTEM

1 LEVEL DFD FOR CUSTOMER



ER-DIAGRAM


RentAndBuyER *
rentandbuy brand
brand_id : int(11)
created_on : datetime(6)
description : varchar(50)
title : varchar(50)
category_id : int(11)
rentandbuy regular_order
order_id : int(11)
created_on : datetime(6)
status : varchar(255)
payment_mode : varchar(255)
customer_id : int(11)
rentandbuy admin
admin_id : int(11)
active : int(11)
address : varchar(50)
city : varchar(100)
country : varchar(100)
state : varchar(100)
zip : int(11)
created_on : datetime(6)
email : varchar(50)
first_name : varchar(50)
last_name : varchar(50)
password : varchar(100)
phone : varchar(13)
rentandbuy user
user_id : int(11)
active : int(11)
address : varchar(50)
city : varchar(100)
country : varchar(100)
state : varchar(100)
zip : int(11)
created_on : datetime(6)
email : varchar(100)
first_name : varchar(100)
last_name : varchar(100)
password : varchar(100)
phone : varchar(14)
rentandbuy order_item
id : int(11)
qty : int(11)
order_id : int(11)
product_id : int(11)
rentandbuy category
category_id : int(11)
price : double
created_on : datetime(6)
description : varchar(50)
title : varchar(50)
rentandbuy user_order
user_order_id : int(11)
created_on : datetime(6)
delivery_status : varchar(50)
payment_status : varchar(50)
payment_type : varchar(50)
tax : int(11)
total_amount : double
user_id : int(11)
rentandbuy product
product_id : int(11)
created_on : datetime(6)
description : varchar(100)
image : varchar(100)
active : int(11)
price : double
qty : int(11)
title : varchar(100)
brand_id : int(11)
category_id : int(11)
rentandbuy review
review_id : int(11)
created_on : datetime(6)
product_id : int(11)
rating : int(11)
review : varchar(100)
user_id : int(11)