**Store To Door**

SOFTWARE ENGINEERING PROJECT REPORT

[Submitted in partial fulfilment]

As a part of the curriculum of

**B.sc(H) COMPUTER SCIENCE**

From



Mata Sundri College ,New Delhi

University of Delhi

Submitted by:

Muskan Rajput(17044570017)

Vinay Goyal(17044570018)

Mahi Chaudhary (17044570029)

**B.sc (H) COMPUTER SCIENCE**

**IV SEMESTER**

MATA SUNDRI COLLEGE , ITO, NEW DELHI,110026

**ACKNOWLEDGEMENT**

We are grateful to our mentor Ms. Ashema Hasti, who gave us the opportunity to work on this project and also motivated us to gain knowledge throughout the course of this project.

And last but not the least, we are thankful to all our friends, batchmates, and other people who have directly or indirectly helped us in the preparation of this project.

Thank you.

Muskan Rajput (17044570017)

Vinay Goyal(17044570018)

Mahi Chaudhary(1704570029)

**CERTIFICATE**

This is to certify that the project titled “**Store 2 Door”** is the bona fide work carried out by

Muskan Rajput,

Vinay Goyal,

Mahi Chaudhary,

Students of B.Sc(H) Computer Science of Mata Sundri College For Women, Delhi affiliated to Delhi University under the supervision of Ms. Ashema Hasti.

**Ms. Ashema Hasti**

**(Mentor)**

**ABSTRACT**

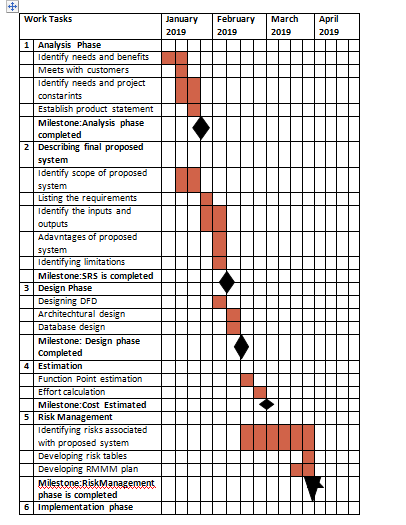
Internet has converted a world into a global village. With the popularization of internet, online shopping has become a new and unique trend. From clothing to electronics, all the things are available on internet.

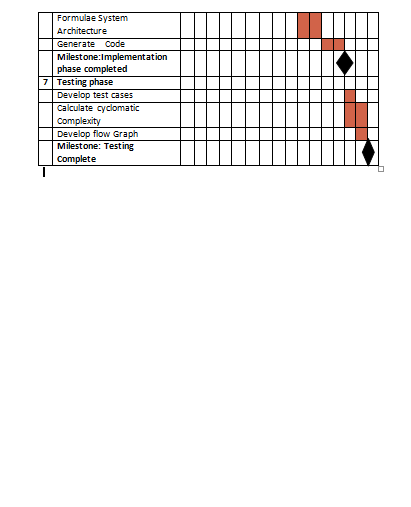
The Customer can search any grocery product according to his/her need and side by side customer can add to cart the product .In order to purchase the product customers will login or register herself/himself by filling personal details. He/she can pay by different methods like cash on delivery or by smartcards. The customer can also filter the price according to her/his budget and can select the size according to her need. The customer can cancel the order and the refund will directly reach to his/her account without any dispute. If the customer wants to know where his/her product reached he/she can track the order. After the delivery of the product a confirmation message is send to the customer.

According to season ”Store 2 Door” provides new sales and offers and new launches is updated time to time. “Store 2 Door ’’provide each brand type product .

This online grocery shop is secure and believable . It fulfils the customer needs and also assures the quality of the product.

**TIMELINE CHART**





**CONTENTS**

Problem Statement…………………………………………………………………

Process Model…………………………………………………………………………

1. Software Requirement Specification…………………………………….
   1. Introduction………………………………………………………………
      1. Purpose……………………………………………………………….
      2. Scope…………………………………………………………………….
      3. Definitions…………………………………………………………….
      4. Additional Information………………………………………….
   2. General Description……………………………………………………..
   3. Functional Requirement………………………………………………
      1. Description…………………………………………………………..
      2. Technical Issues………………………………………………………
   4. Non Functional Requirement……………………………………………
      1. Security………………………………………………………………..
      2. Reliability………………………………………………………………..
      3. Availability………………………………………………………………
      4. Maintainability………………………………………………………………
      5. Portability…………………………………………………………………..

1.5Interface Requirement…………………………………………………….

15.1Hardware Interface………………………………………………..

* + 1. Software Interface………………………………………………..
  1. Performance Requirement……………………………………………
  2. Data Flow Diagram…………………………………………………………..
  3. Data Dictionary…………………………………………………………….
  4. ER Diagram.........................................................

1. Estimations………………………………………………………………………..
   1. Function points……………………………………………………………….
   2. Efforts………………………………………………………………………………..
2. Scheduling……………………………………………………………………………..
3. Risk Management………………………………………………………………..
4. Design……………………………………………………………………………………

**Problem statement**

Online shopping is a form of electronic commerce which allows consumers to directly buy goods or services from a seller over the Internet using a particular website.

**Why Grocery is widely accepted Model?**

->It saves time.

->The very best grocery websites also have online chat facilities that allow us to interact with a real person.

->it is safe and secure.

->Efficient

Our Grocery System is a chain of many shops. It is not just another hyper market. It caters to every need of your family. This Grocery Shop scores over other stores because it provides value for money for Indian customers.

In this we plan to add much more to complete our customers shopping experience.

5 Reasons to Order Groceries online:

1) You can order anytime, 24\*7.

2) Delivery is convenient.

3) Stores send best quality & features to minimize complaints.

4) Avoid crowds and parking queues.

5) Get the best prices, browse offer.

**Main Objective**:

\*Promoting a service or product online.

\*Selling a service or product.

\*Providing product support or customer service.

\*Providing corporate information.

\*Establishing brand awareness and corporate identity.

**AIM**

The main of this project is to help the people finding the groceries they need without any trouble.

**SOFTWARE LIFECYCLE MODEL**

**PROTOTYPING MODEL**

 In this model, the basic idea is that rather than freezing the requirements before coding or designs we can proceed, using the throwaway prototype requirements of the development understands.

On the basis of requirements which are currently known. The prototype development is inside contains the coding, designing as well as testing. But each of these phases is not done very formally or thoroughly. Thus using this mechanism, end users can get the systems actual feel, only because of the thing that prototype interactions will allow clients to understand the requirements better way for the desired system.

**Software Requirement analysis**

**1. INTRODUCTION**

**1.1 PURPOSE**

This document is meant to delineate the features of Online Grocery System (OGS), so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other. The OGS for grocery shop web application is intended to provide complete solutions for vendors as well as customers through a single gateway using the internet. It will enable vendors to setup online shops, customer to browse through the shop and purchase them online without having to visit the shop physically. The administration module will enable a system administrator to approve and reject request for new shops and maintain various lists of shop category.

**1.2 SCOPE**

The Store2Door system allows the customer’s to maintain their cart for add or remove the

 Product over the internet.

**1.3 Definitions**

OGS- Online Grocery System (for grocery shop)

 SRS- Software Requirement Specification

 GUI- Graphical User Interface Stockholder- The person who will participate in system

Eg. Customer, Admin, Visitor etc.

**1.4 Overview**

This system provides an easy solution to customers to buy the product without going to the shop.

**1.5 Additional Information**

The system works on internet server, so it will operated by any end user for the buying purpose.

**2. General Description**

The Online Grocery System (OGS) application enables vendors to set up online shops, customers to browse through the shops, and a system administrator to approve and reject requests for new shops and maintain lists of shop categories. Also the developer is designing an online shopping site to manage the items in the shop and also help customers purchase them online without having to visit the shop physically. The online shopping system will use the internet as the sole method for selling goods to its consumers.

**3. Functional Requirement**

This section provides requirement overview of the system. Various functional modules that can be implemented by the system will be -

**3.1 Registration**

 If customer wants to buy the product then he/she must be registered, unregistered

user can’t go to the shopping cart.

**3.2 Login**

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

**3.3 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.

**3.4 Payment**

 For customer, there are many type of secure billing will be prepaid as debit or credit card, post paid as after shipping, check or bank draft. The security will provide by the third party like Pay-Pal etc.

**3.5 Logout**

After the payment or surf the product the customer will logged out.

**3.6 Report Generation**

After all transaction the system can generate the portable document file (.pdf) and

then sent one copy to the customer’s Email-address and another one for the system data base to calculate the monthly transaction .

**4. Interface Requirement**

Various interfaces for the product could be-

Login Page2. 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.

After all transaction the system makes the selling report as portable document file (.pdf)

and send to the customer’s Email address.

**4.1 Hardware Interface**

The System must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for e.g. Modem, WAN-

 LAN, Ethernet Cross-Cable.

**4.2 Software Interface**

The system is on server so it requires the any scripting language like PHP, VBScript etc. The system require Data Base also for the store the any transaction of the system like MYSQL etc. system also require DNS(domain name space) for the naming on the internet. At the last user need web browser for interact with the system.

**5. Performance Requirement**

There is no performance requirement in this system because the server request and response is depended on the end user internet connection.

**6. Design Constrain**

The system shall be built using a standard web page development tool that

Conforms to Microsoft’s GUI standards like HTML, XML etc.

**7. Non Functional requirements**

**7.1 Security**

 The system use SSL (secured socket layer) in all transactions that include any confidential customer information.

The system must automatically log out all customers after a period of inactivity.

 The system should not leave any cookies on the customer’s computer containing the user’s password.

 The system’s back -end servers shall only be accessible to authenticated administrators.

Sensitive data will be encrypted before being sent over insecure connections like the internet.

**7.2 Reliability**

-The system provides storage of all databases on redundant computers with automatic switchover.

 -The reliability of the overall program depends on the reliability of the separate components.

-The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Thus the overall stability of the system depends on the stability of container and its underlying operating system.

**7.3 Availability**

The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. In case of a of a hardware failure or database corruption, a replacement page will be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the administrator. Then the service will be restarted. It means 24 X 7 availability.

**7.4 Maintainability**

 A commercial database is used for maintaining the database and the application server takes care of the site. In case of a failure, a re-initialization of the program will be done. Also the software design is being done with modularity in mind so that maintainability can be done efficiently.

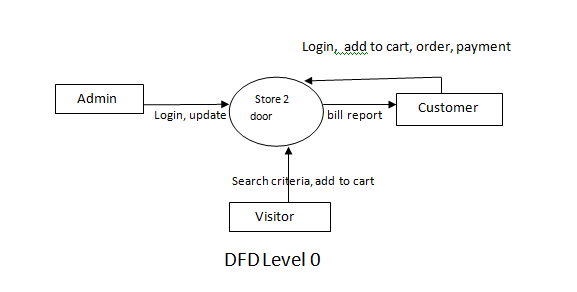
**7.5 Portability**

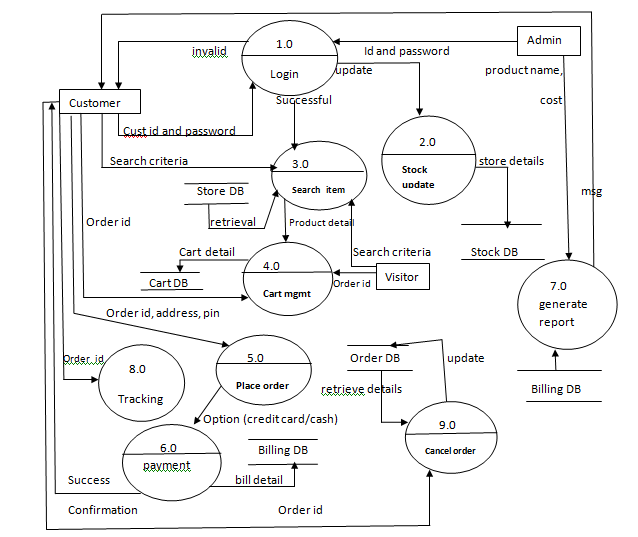
--The application is HTML and scripting language based. So The end-user part is fully portable and any system using any web browser should be able to use the features of the system, including any hardware platform that is available or will be available in the future.

--An end-user is use this system on any OS; either it is Windows or Linux.

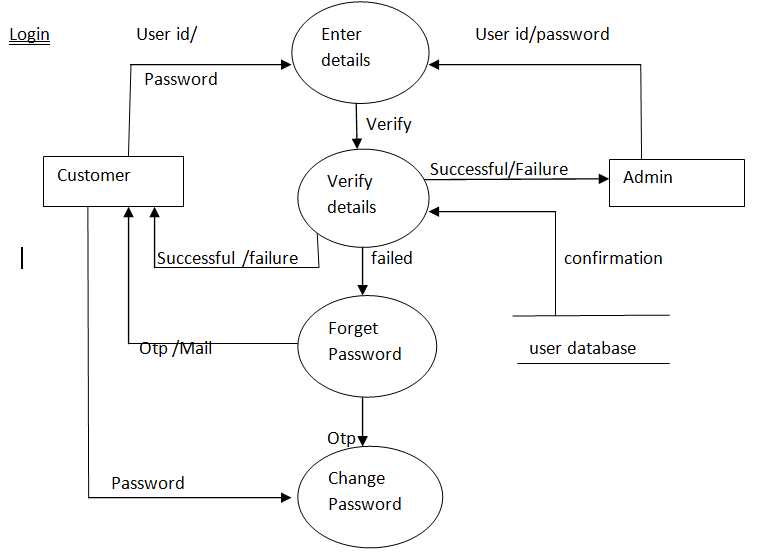
--The system shall run on PC, Laptops, and PDA etc.

**DATA FLOW DIAGRAMS**

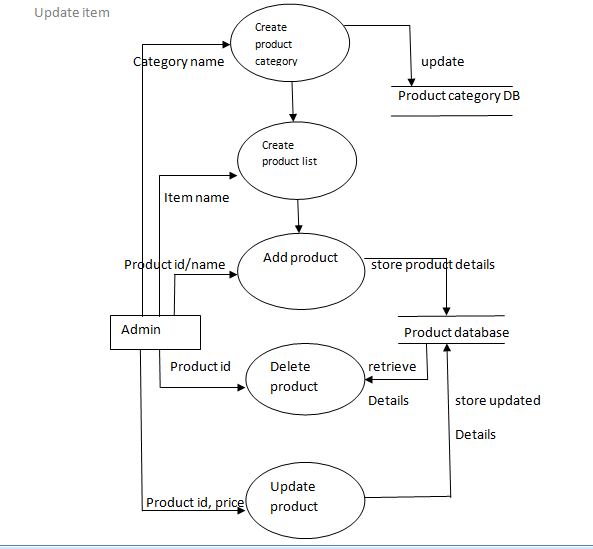




LEVEL 1

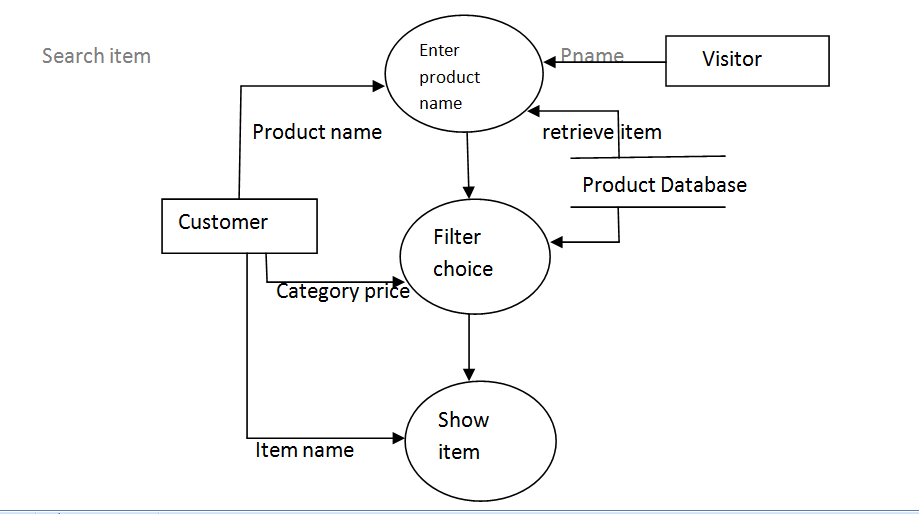


LEVEL 1.1

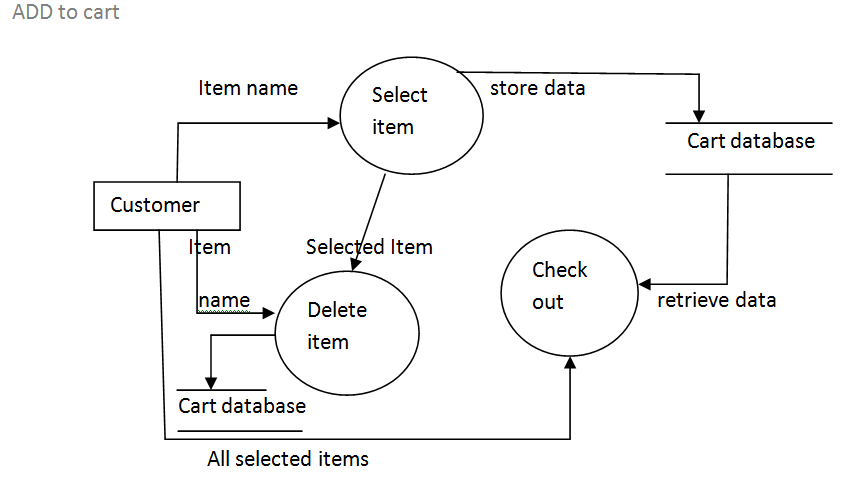


STOCK UPDATE

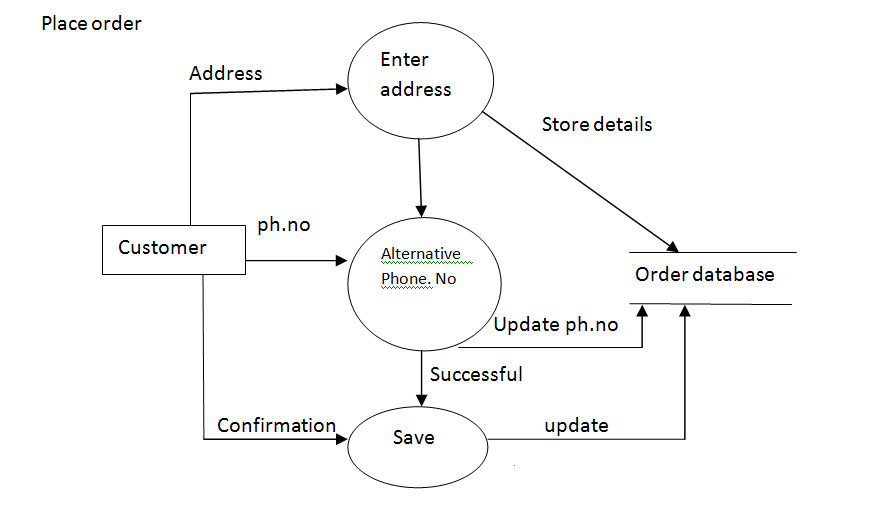
LEVEL 1.2



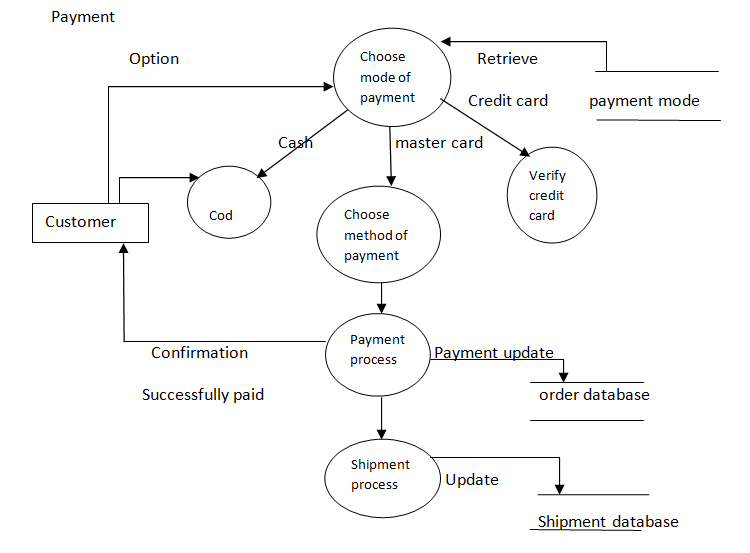
LEVEL 1.3



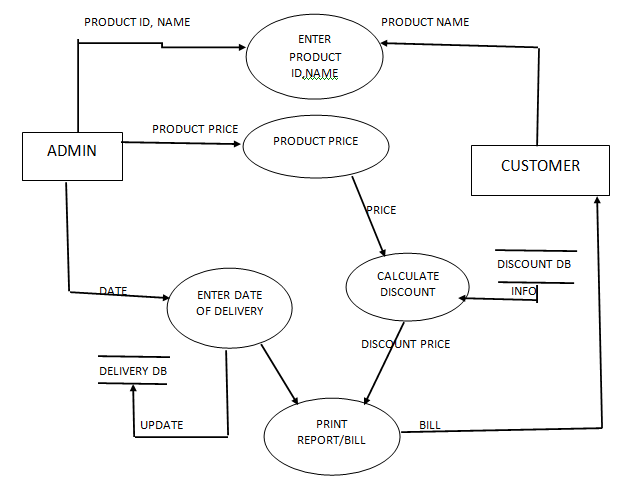
LEVEL 1.4



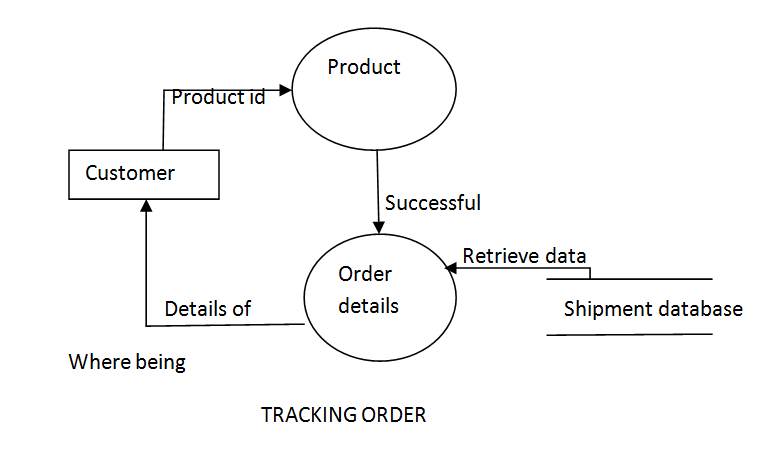
LEVEL 1.5



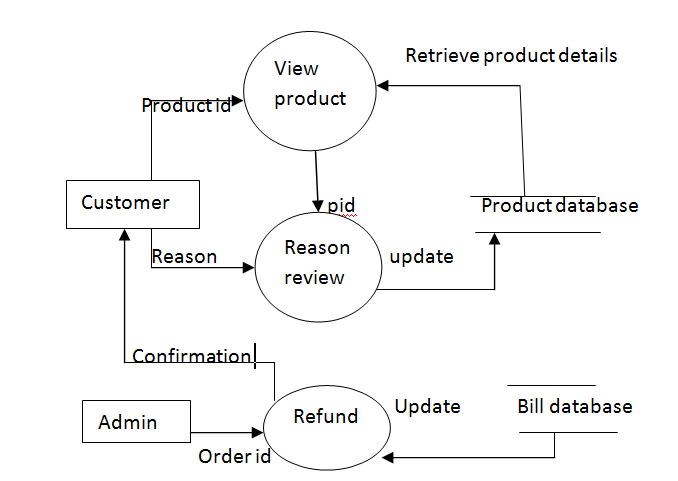
LEVEL 1.6



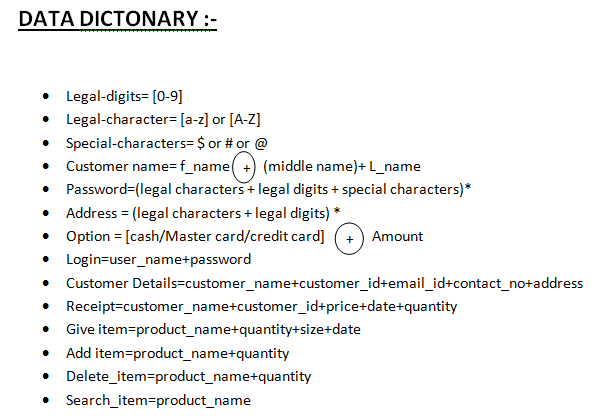
LEVEL 1.7 GENERATE REPORT



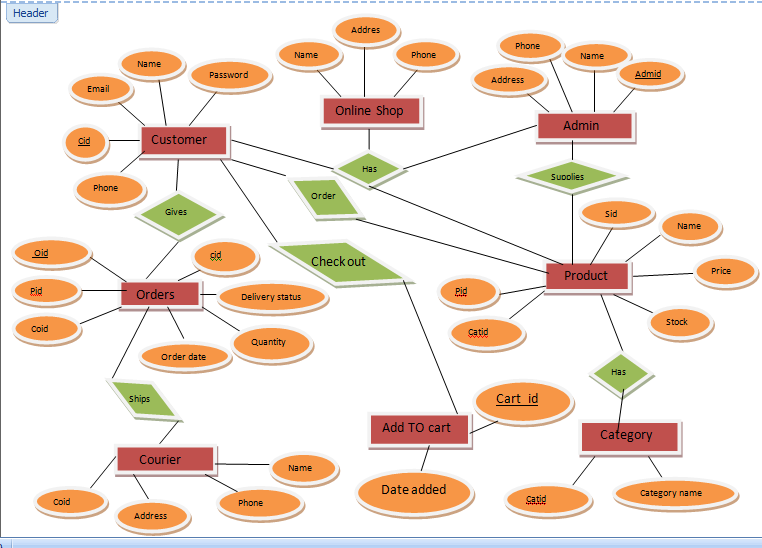
LEVEL 1.8

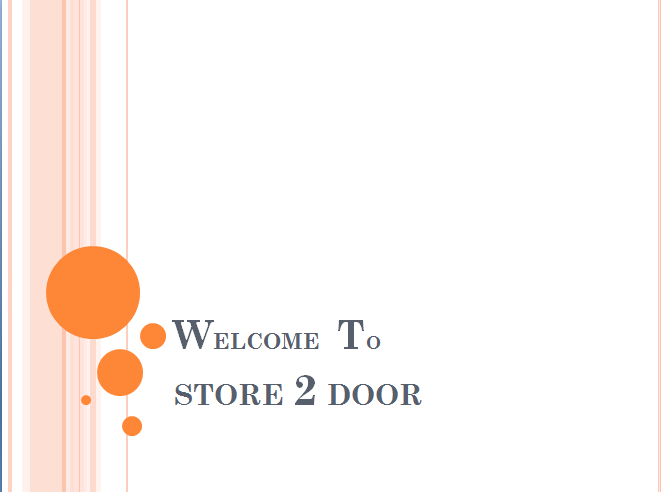


LEVEL 1.9 CANCEL ORDER

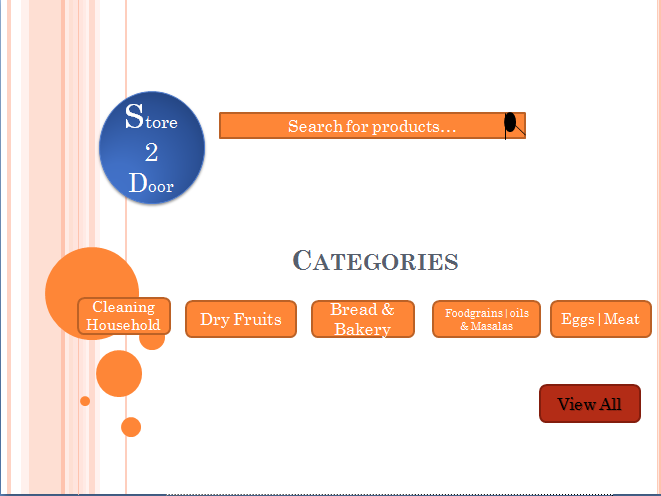


**ER Diagram**

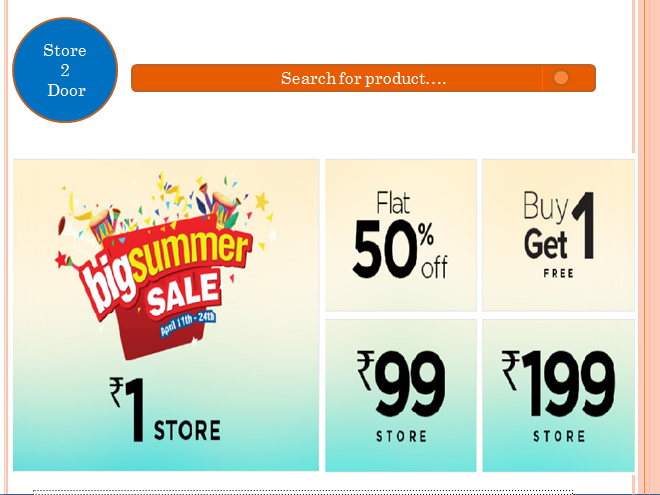


****

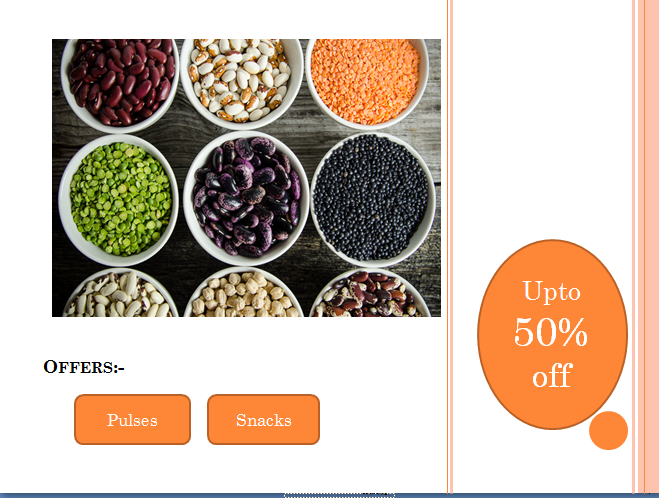
No. of External Inputs: 0 No. of External Outputs:1 No. of External Inquiries:0 No. of internal logical files:0 No. of external interface files:0



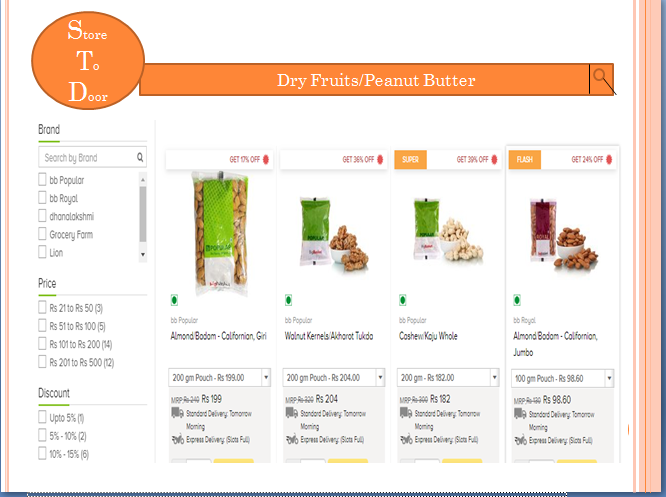
No. of External Inputs: 9 No. of External Outputs: 1 No. of External Inquiries:1 No. of internal logical files:1 No. of external interface files:0

****

No. of External Inputs: 0 No. of External Outputs:1 No. of External Inquiries:1 No. of internal logical files:1 No. of external interface files:0

****

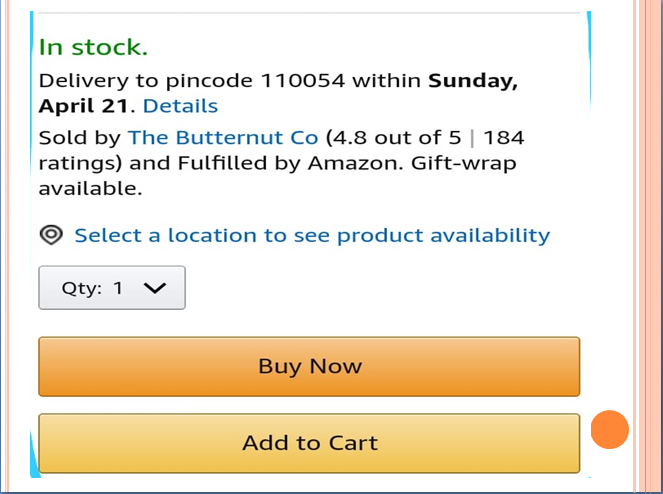
No. of External Inputs: 0 No. of External Outputs:1 No. of External Inquiries:0 No. of internal logical files:0 No. of external interface files:0

****

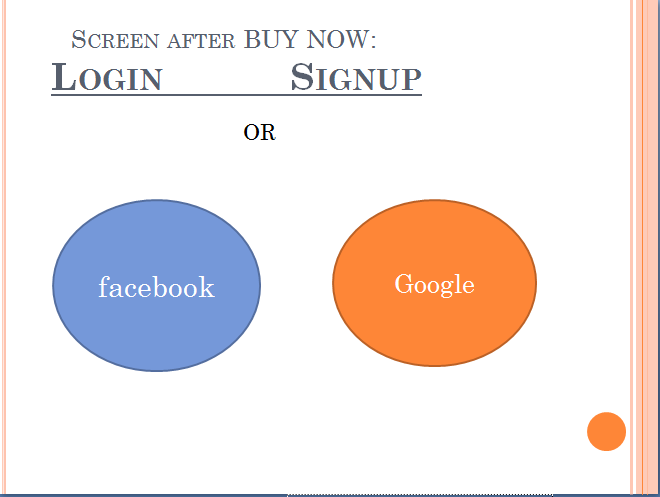
No. of External Inputs: 3 No. of External Outputs:1 No. of External Inquiries:1 No. of internal logical files:0 No. of external interface files:0

****

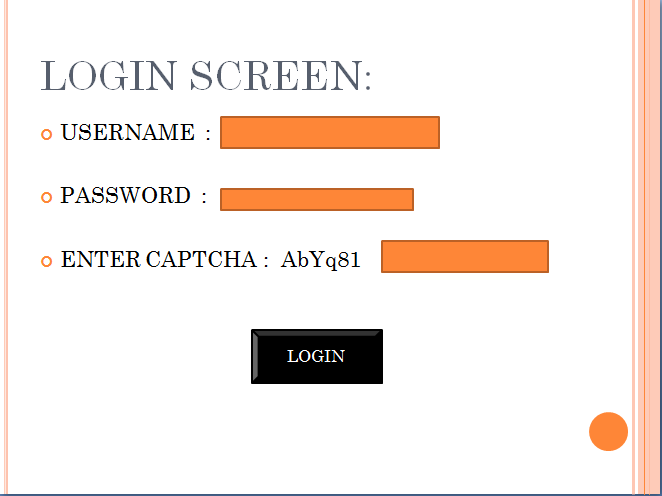
No. of External Inputs: 1 No. of External Outputs:1 No. of External Inquiries:0 No. of internal logical files:0 No. of external interface files:0

****

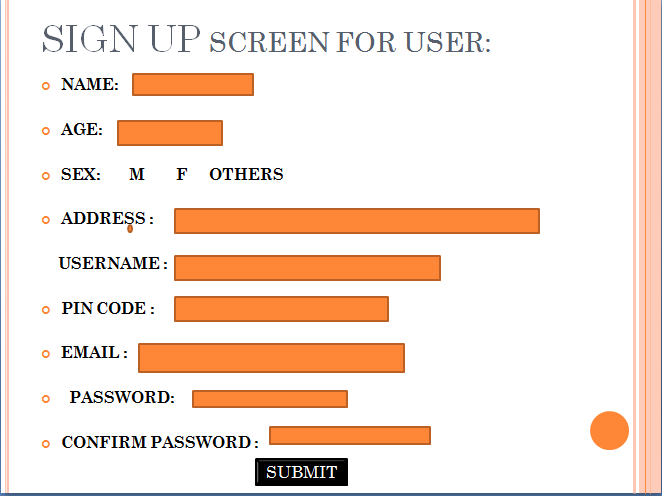
No. of External Inputs: 3 No. of External Outputs:1 No. of External Inquiries: 0 No. of internal logical files:3 No. of external interface files:0

****

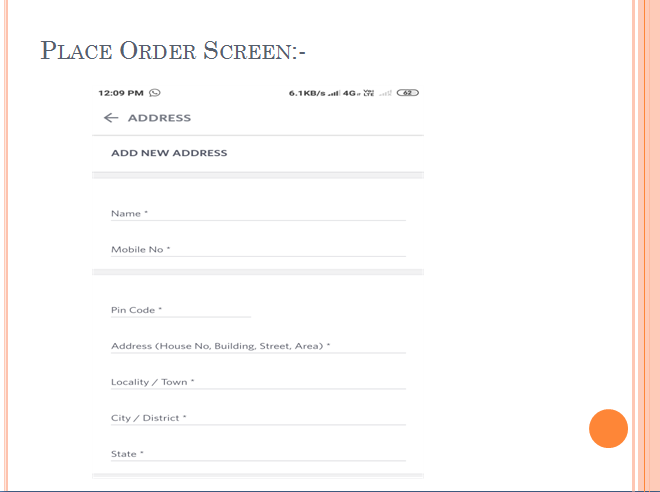
No. of External Inputs: 4 No. of External Outputs:1 No. of External Inquiries:0 No. of internal logical files:1 No. of external interface files:0

****

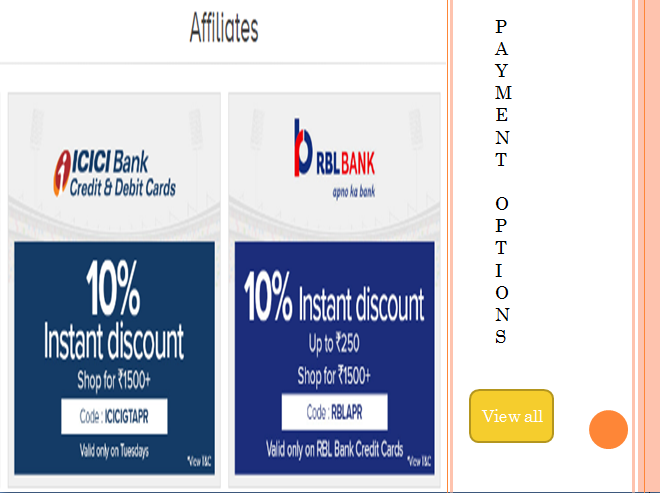
No. of External Inputs: 4 No. of External Outputs:1 No. of External Inquiries:0 No. of internal logical files:1 No. of external interface files:0

****

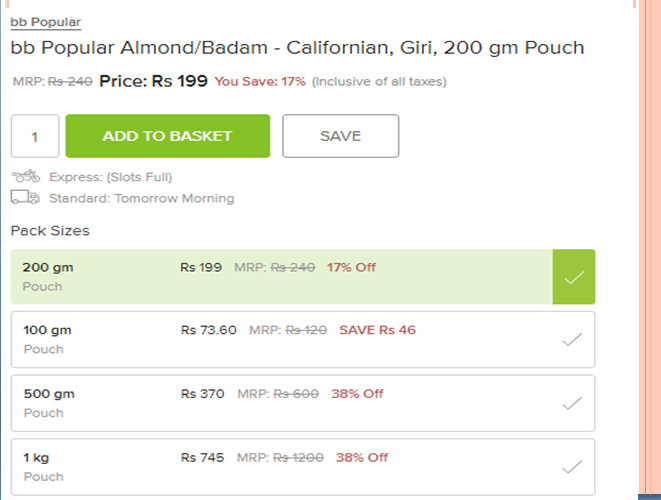
No. of External Inputs: 9 No. of External Outputs:1 No. of External Inquiries:0 No. of internal logical files:1 No. of external interface files:0

****

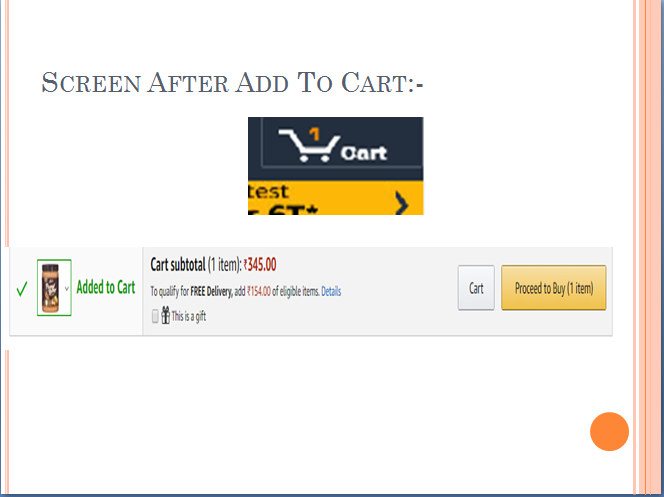
No. of External Inputs: 7 No. of External Outputs:1 No. of External Inquiries:0 No. of internal logical files:1 No. of external interface files:0

****

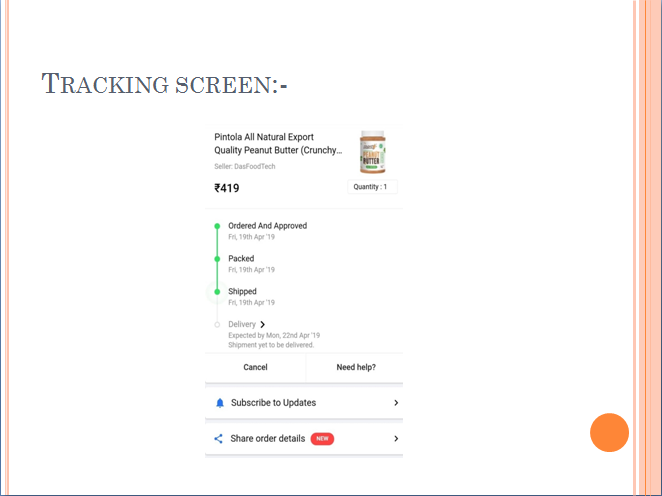
No. of External Inputs:2 No. of External Outputs:1 No. of External Inquiries:0 No. of internal logical files:1 No. of external interface files:0

****

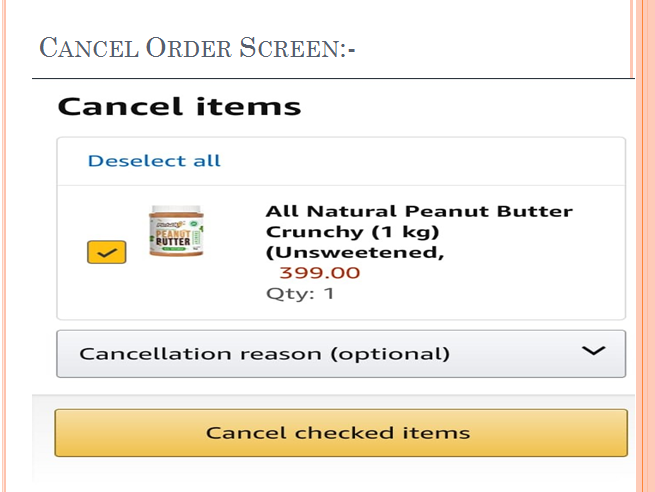
No. of External Inputs: 2 No. of External Outputs:1 No. of External Inquiries:0 No. of internal logical files:1 No. of external interface:0

****

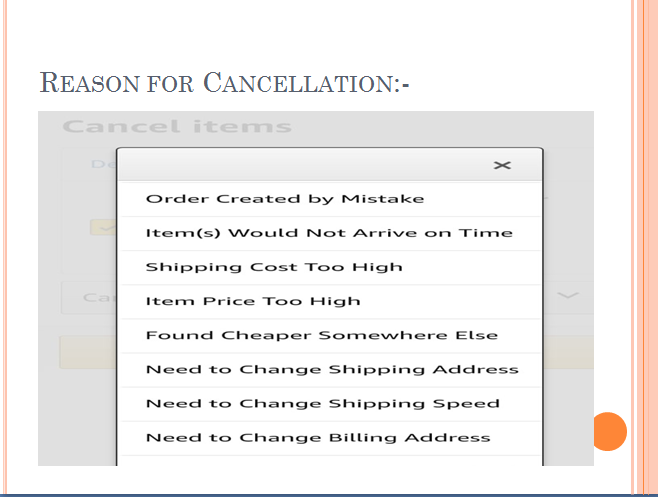
No. of External Inputs: 1 No. of External Outputs:1 No. of External Inquiries:0 No. of internal logical files:1 No. of external interface files:0

****

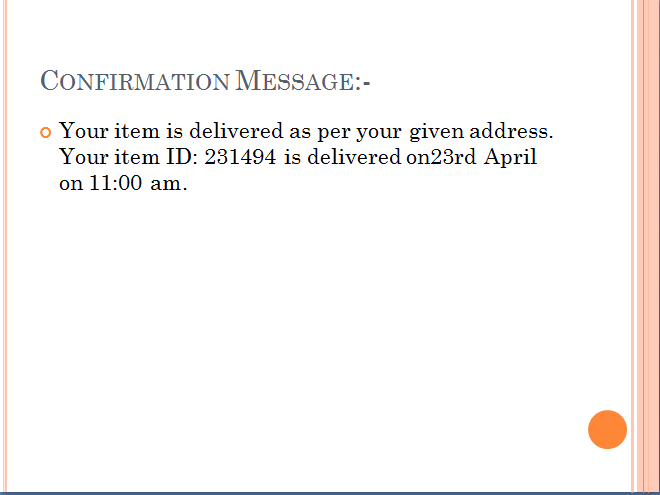
No. of External Inputs: 3 No. of External Outputs:2 No. of External Inquiries:0 No. of internal logical files:1 No. of external interface files:0

****

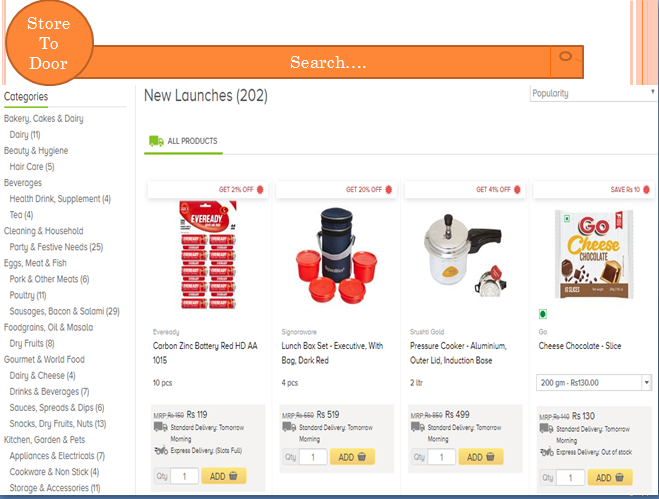
No. of External Inputs: 2 No. of External Outputs:2 No. of External Inquiries:0 No. of internal logical files:1 No. of external interface files:0

****

No. of External Inputs: 1 No. of External Outputs:1 No. of External Inquiries:0 No. of internal logical files:1 No. of external interface files:0

****

No. of External Inputs: 0 No. of External Outputs:1 No. of External Inquiries:0 No. of internal logical files:1 No. of external interface files:0

****

No. of External Inputs: 0 No. of External Outputs:1 No. of External Inquiries:1 No. of internal logical files:1 No. of external interface files:0

**PROJECT METRICS**

Project metrics are used to control and coordinate software engineering process and to improve quality of the software to be produced. Project specific metrics provide indication of productivity and insight into the technical activities.

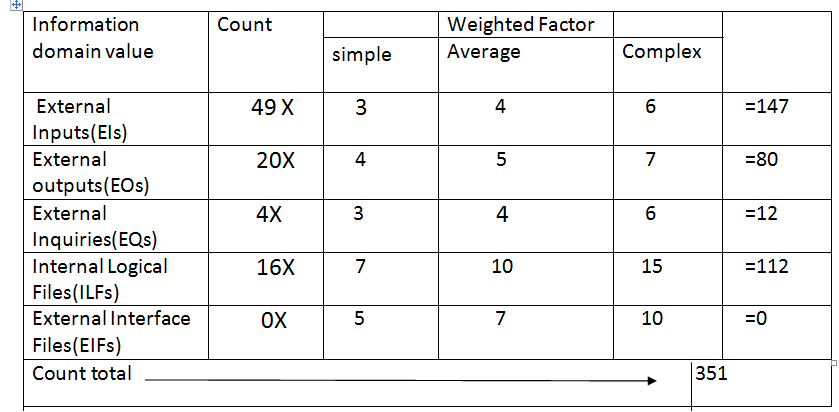
FUNCTION ORIENTED METRICS

Function oriented metrics use function point as normalization value. Function points are derived using empirical relationship based on countable measure of software’s information domain and qualitative assessments of software complexity.

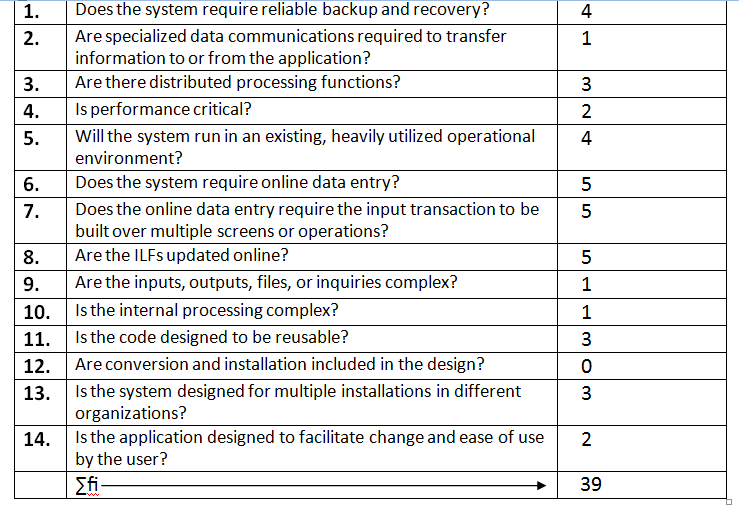
To compute function points (FP), the following relationship is used:

**FP= count total \* [0.65 + 0.01 \* ∑(fi)]**

table:



The fi (i= 1 to 14) are value adjustment factors (VAF) based on following responses:



Therefore,

FP= 351 \*[0.65 + (0.01\*39)] =360.04

**EFFORT ESTIMATION USING COCOMO MODEL**

Constructive Cost Model (COCOMO II) is a more comprehensive estimation model.

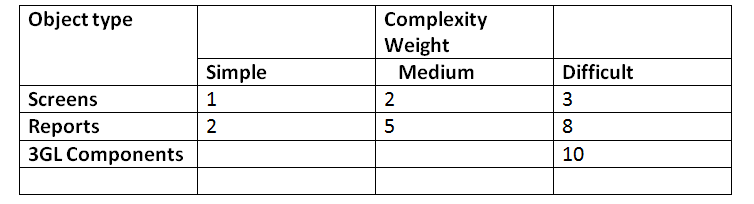
The COCOMO II model requires sizing information. Three different sizing options are available as part of the model hierarchy:

.object points

.function points

.lines of source code

The object point is an indirect software measure that is computed using counts of the number of (1) screens (at the user interface),(2) reports, and (3) components likely to be required to build the application. Each object instance is classified into one of three complexity levels based on the following table-



**Figure**-Complexity Weighting for Object Types

The object count is determined by multiplying the total number of objects instances by weighting factor. When component based development or general software re-used is to be applied, the percent of reuse is estimated and object count is adjusted:

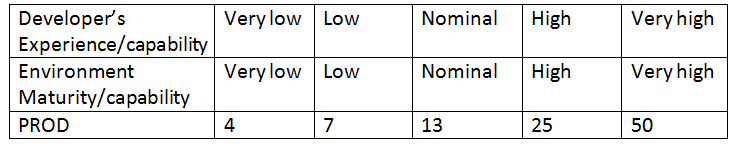
**NOP=(object points)X[(100-%re-use)/100)]**

Where NOP is defined as new object points

To derive an estimate of effort based on the computed NOP value, a “productivity rate” must be derived.

PROD=NOP/person-month

**Figure-**Productivity rate for object points



Once the productivity rate has been determined , an estimate of project effort is computed using-

**Estimated Effort=NOP/PROD**

COCOMO Estimation for our project-

Number of screens=19

Number of reports=1

Number of 3GL components used=0

Person-month= 3

In our project , there are simple screens and reports.

So,

**Object point=19\*1+1\*2=21**

Since, we are not re-using any of the components in our project , the % reuse is zero her.

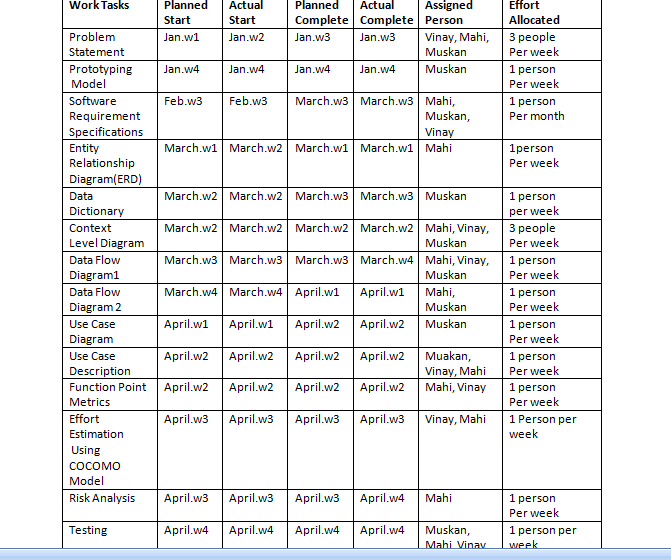
**NOP=23\*[(100-0)/100]=23**

**PROD=7**

**Now,**

**Estimated Effort=** 21/7=3

**PROJECT SCHEDULING**



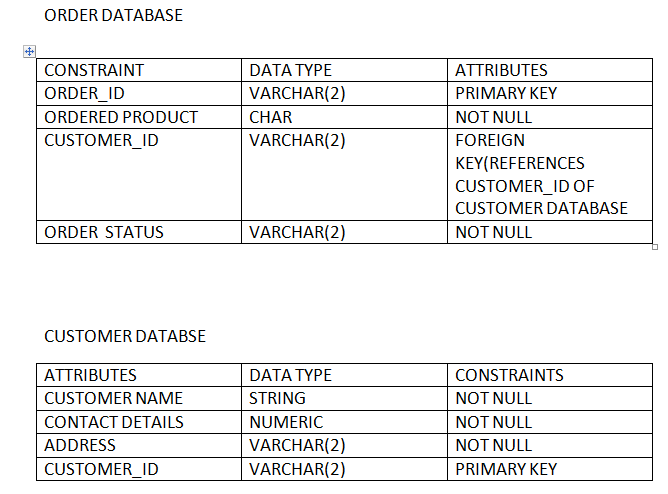
**DATA DESIGN**

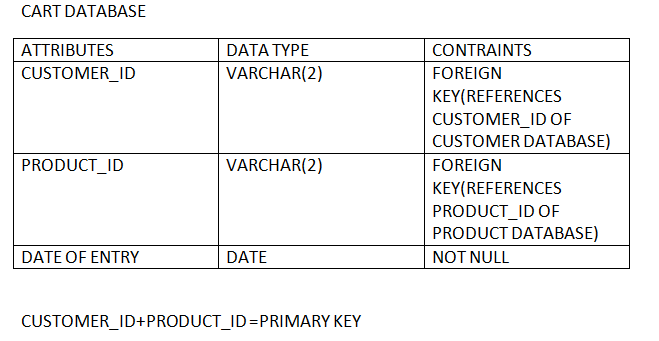
Name: ADMIN

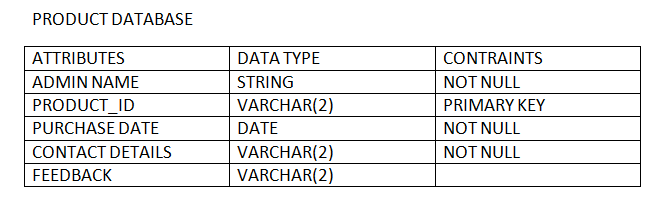
|  |  |  |  |
| --- | --- | --- | --- |
| S.no | ATTRIBUTE | DATA TYPE | CONSTRAINTS |
| 1 | Name | Varchar(50) | Gives name to the admin |
| 2 | Id | Varchar(15) | Primary key |
| 3 | Password | Varchar(15) | Used to login into the s/w |

Name: Login table

|  |  |  |  |
| --- | --- | --- | --- |
| S.no | ATTRIBUTE | DATA TYPE | CONSTRAINTS |
| 1 | User name | Varchar(15) | Not null |
| 2 | Password | Varchar(15) | Not null |





****

**RISK MANAGEMENT**

**ASSESSING OVERALL PROJECT RISKS**

**1. Have top software and customer managers formally committed to support the project?** YES

**2. Are end users enthusiastically committed to the project and the system product to be built?** YES

**3. Are requirements fully understood by the software engineering team and its customers?** YES

**4. Have customers been involved fully in the definition of requirements?** YES

**5. Do end users have realistic expectations?** YES

**6. Is the project scope stable?** YES

**7. Does the software engineering team have the right mix of skills?** YES

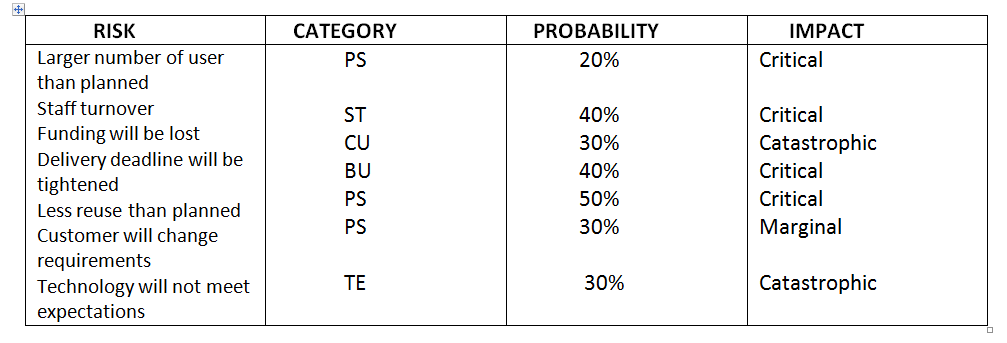
**8. Are project requirements stable?** YES

**9. Does the project team have experience with the technology to be implemented?** YES

**10. Is the number of people on the project team adequate to do the job?** YES

**11. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?** YES

**RISK TABEL**



**TESTING**

**VALIDATION CODE**

CHECKING PASSWORD VALIDATION

**1. Ifstream fin(“USERACC\_FILE”, ios::binary);**

**2. String id, password;**

**3. Flag=0; 1**

**4. gets (id);**

**5. gets (password);**

**6. While (fin. read ((char\*)&e, sizeof(e)) 2**

**7. {**

**8. If (id==e.id && password==e.password) 3,4**

**9. {**

**10. Flag=1;**

**11. Given access to system continue; 5**

**12. }**

**13. } 6**

**14. If(flag==0) 7**

**15. cout<<”access denied”; 8**

**16. fin.close ();**

**17. return; 9**

**FLOW GRAPH:-**

4

5

2

1

**CYCLOMATIC COMPLEXIT**

V(G) = E-V+2

= 12 -9 + 2

= 5

Number of region = number of predicate node +1

= 4+1 = 5

Number of independent paths = 5

**INDEPENDENT PATHS:-**

* **1-2-7-8-9**
* **1-2-7-9**
* **1-2-3-4-5-6-2-7-9**
* **1-2-3-6-2-7-9**
* **1-2-3-4-6-2-7-9**