COCOMO MODEL

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e **number of Lines of Code**. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. It was proposed by Barry Boehm in 1970 and is based on the study of 63 projects, which make it one of the best-documented models.

Boehm's definition of organic, semidetached, and embedded systems:

1. **Organic –** A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.
2. **Semi-detached –** A software project is said to be a Semi-detached type if the vital characteristics such as team-size, experience, knowledge of the various programming environment lie in between that of organic and Embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. Eg: Compilers or different Embedded Systems can be considered of Semi-Detached type.
3. **Embedded –** A software project with requiring the highest level of complexity, creativity, and experience requirement fall under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

All the above system types utilize different values of the constants used in Effort Calculations.

We Use Semi-Detached for our DE project

KLOC = 8

Effort = $3(KLOC)^{1.12}$ Person Month

$= 3(8)^{1.12}$

Effort = 30.802 Person Month

Time of Development = $2.5\ (Effort)^{0.35}$ Month

$= 2.5\ (30.802)^{0.35}$

TOD = 7.486 Month

Number of Person = Effort / TOD

= 30.802 / 7.486

Number of Person = 4.11 Person

Function Point

Function Point (FP) is an element of software development which helps to approximate the cost of development early in the process. It may measures functionality from user's point of view

```
0 - No Influence
1 - Incidental
2 - Moderate
3 - Average
4 - Significant
5 - Essential
```

F = 14 * Scale

= 14 * 3 (Scale = 3 because of Average)

F = 42

CAF(Complexity Adjustment Factor) = 0.65 + (0.01 + F)

= 0.65 + (0.01 + 42)

CAF = 1.07

User Input :- 25

First name, last name, email, mobile number, pincode, state, city, selection, address1, address2, password, con. Password, select product, product name, product price, product description, product photo, product id, add card, select payment method, place order, add to cart, select quantity, add review etc…

User Output:- 12

Display product, Display product info, sorting, filtering, product delivery, bill generate, Show Best Deal, View cart, Payment output, Tracking Id, Error message, See reviews etc…

User Inquiries:- 6

Contact Us, product inquiries, business inquiries, Social media, promotion inquiries, customer care etc…

External File:- 8

FUNCTION UNITS	LOW	AVG	HIGH
EI	3	4	6
EO	4	5	7
EQ	3	4	6
ILF	7	10	15
EIF	5	7	10

UFP(Unadjusted Function Point) = (UI * 4) + (UO * 5) + (EQ * 4) + (ILF * 10) + (EIF * 7)

= (25 * 4) + (12 * 5) + (6 * 4) + (8 * 10) + (8 * 7)

= 100 + 60 + 24 + 80 + 56

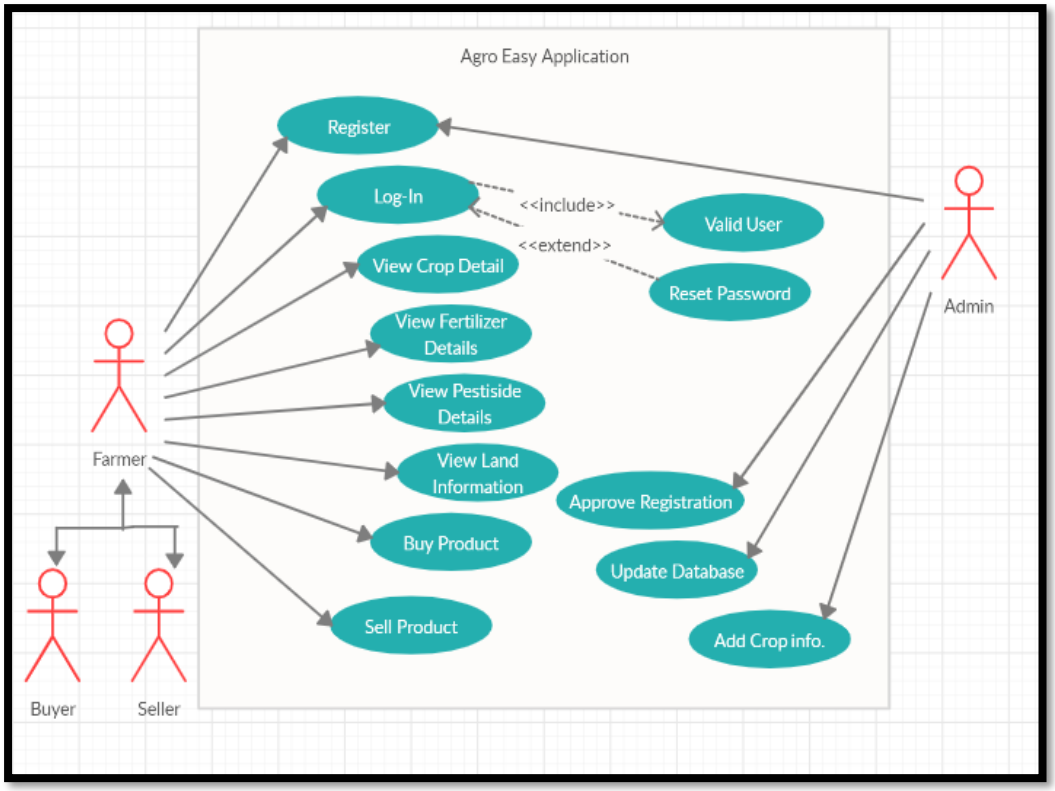
UFP = 320

FP = UFP * CAF

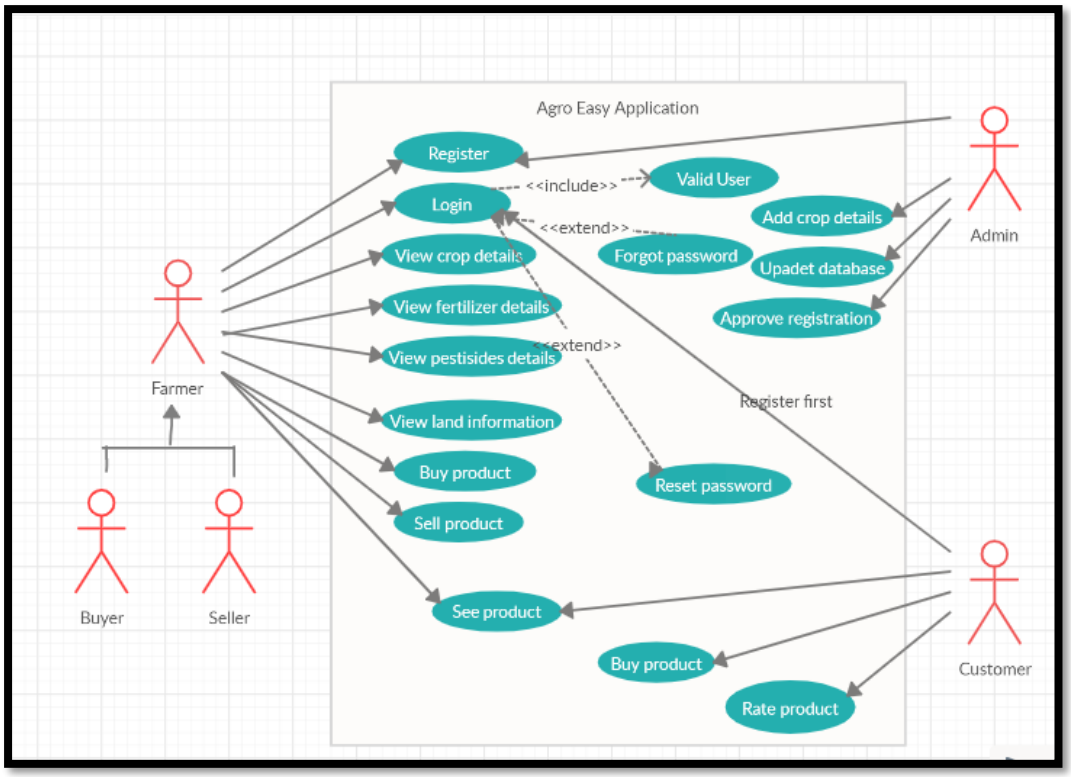
= 320 * 1.07

FP = 342.4

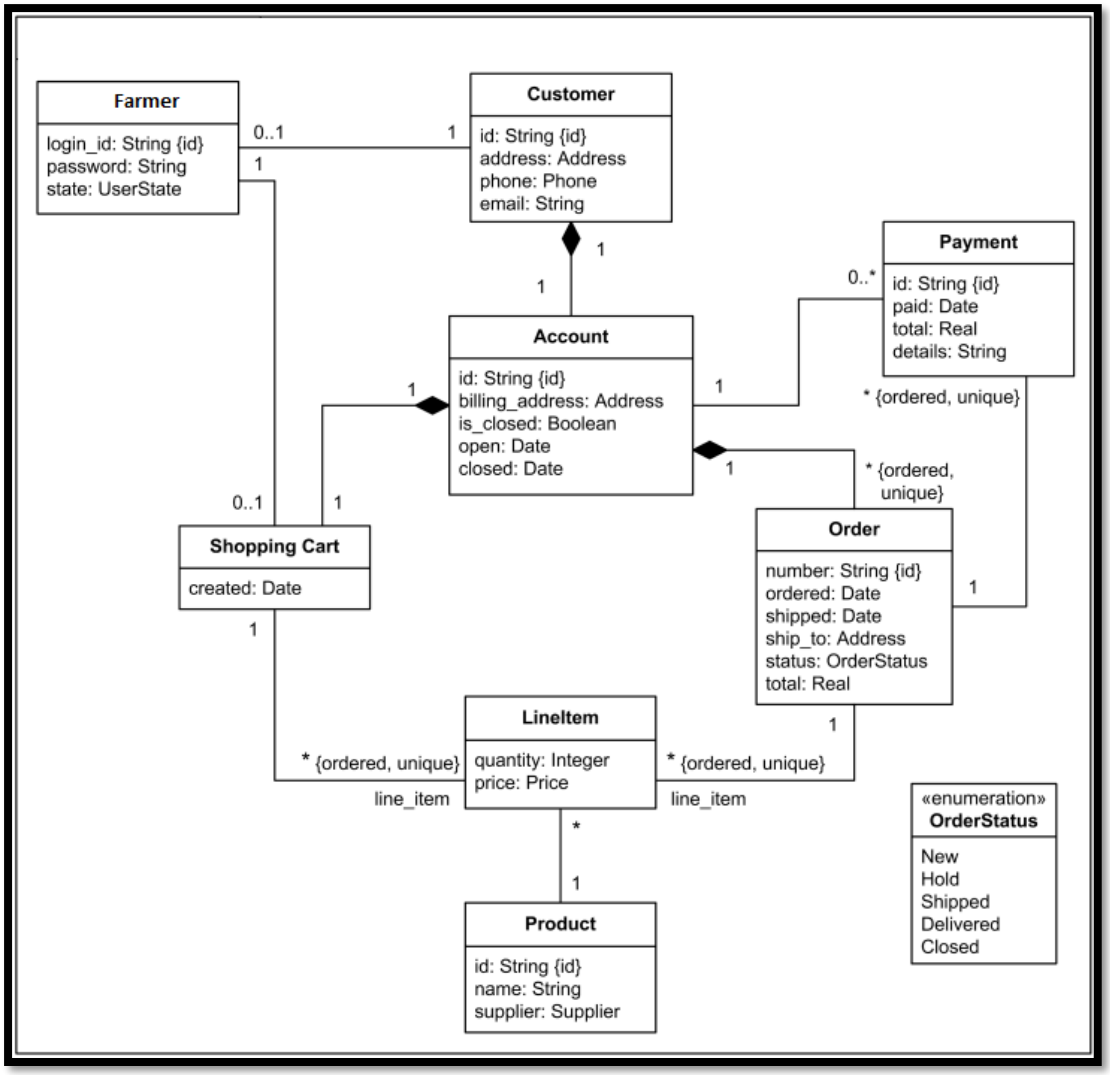
Use Case Diagram Without Iteration



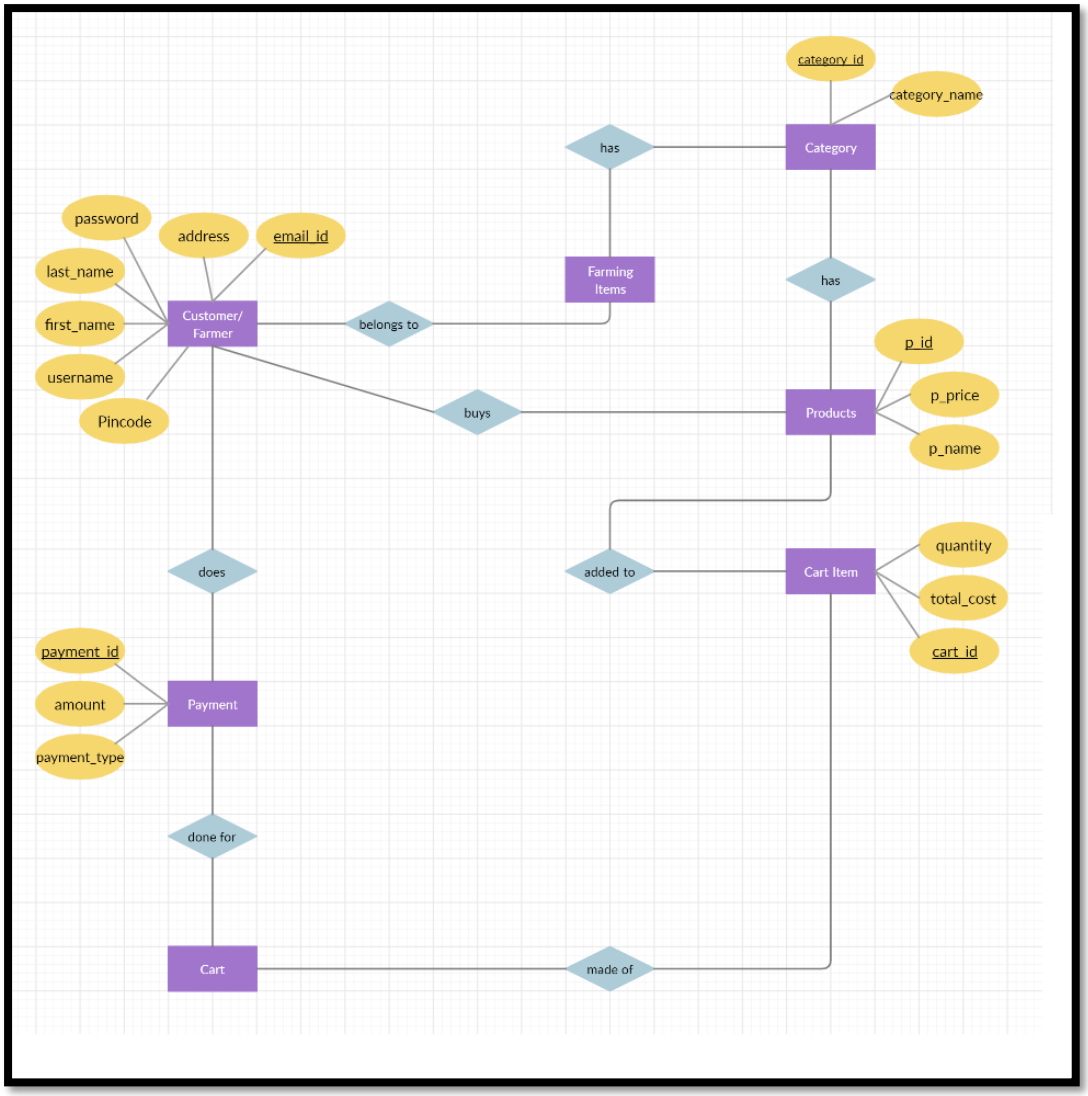
Use Case Diagram With Iteration



Class Diagram



E-R Diagram


category_id
category_name
has
Category
password
address
email_id
last_name
Farming Items
has
first_name
Customer/ Farmer
belongs to
p_id
username
p_price
Pincode
buys
Products
p_name
quantity
does
added to
Cart Item
total_cost
payment_id
cart_id
amount
Payment
payment_type
done for
Cart
made of