Synopsis on…

Ration Shop Distribution system

Problem Statement:-

The present system uses manual methods of distribution of ration commodities like sugar, rice, wheat, etc. It is developed to dispense the correct quantity of ration to the card holders depending on type of card and the number of members in the family, and also maintain the details of transactions in database.

Objectives:-

- To reduce the paper work
- To automate the maintenance of the ration shop
- To maintain transparency of the traditional Ration system which could be manipulated by shop owner.

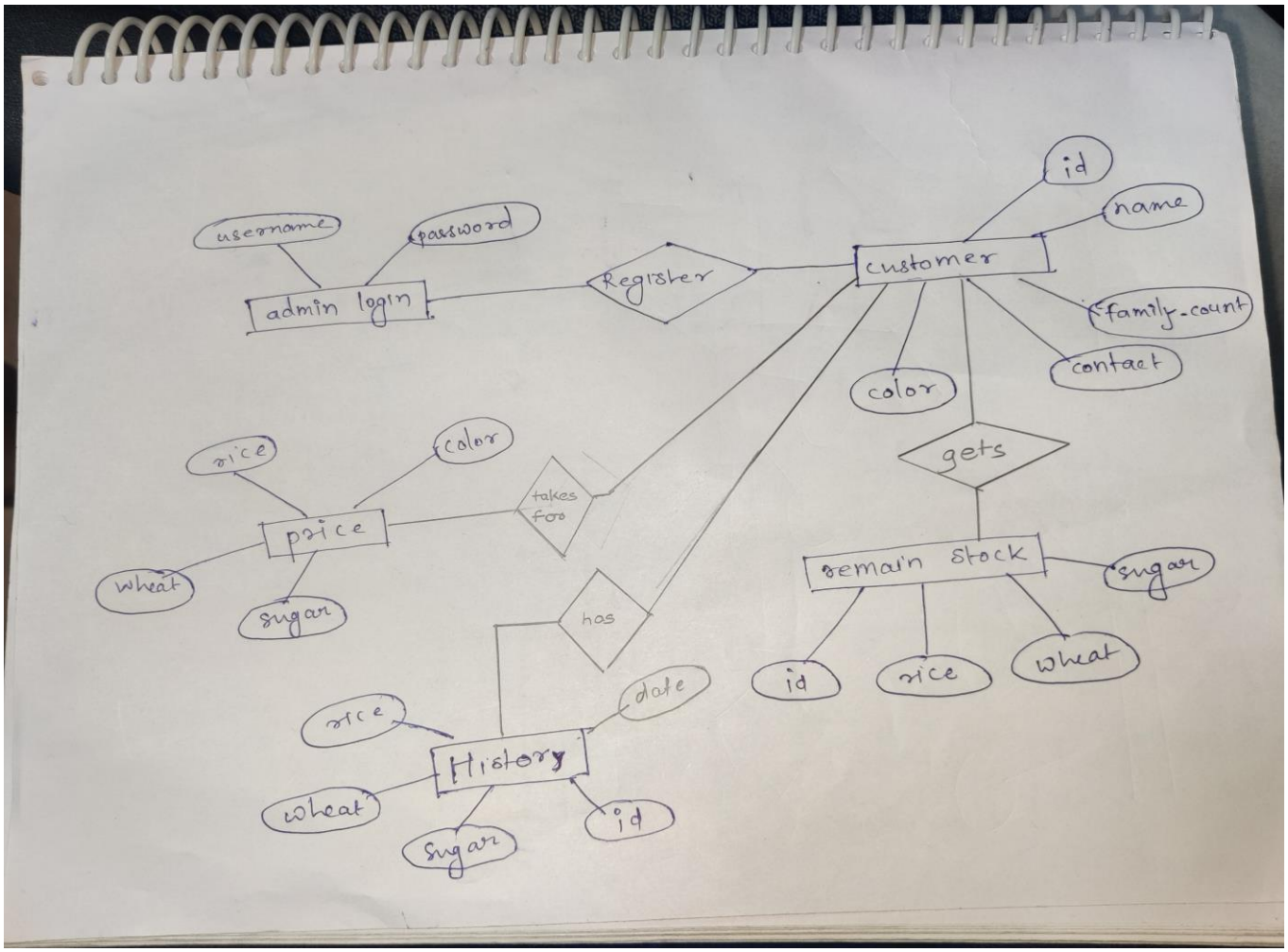
Functional Requirements:-

- Able to register the new customer using his information
- Able to search the customer using id which is unique.
- Enables admin to sell grains to new customer through automated website
- Can see the remaining stock of the Ration shop.

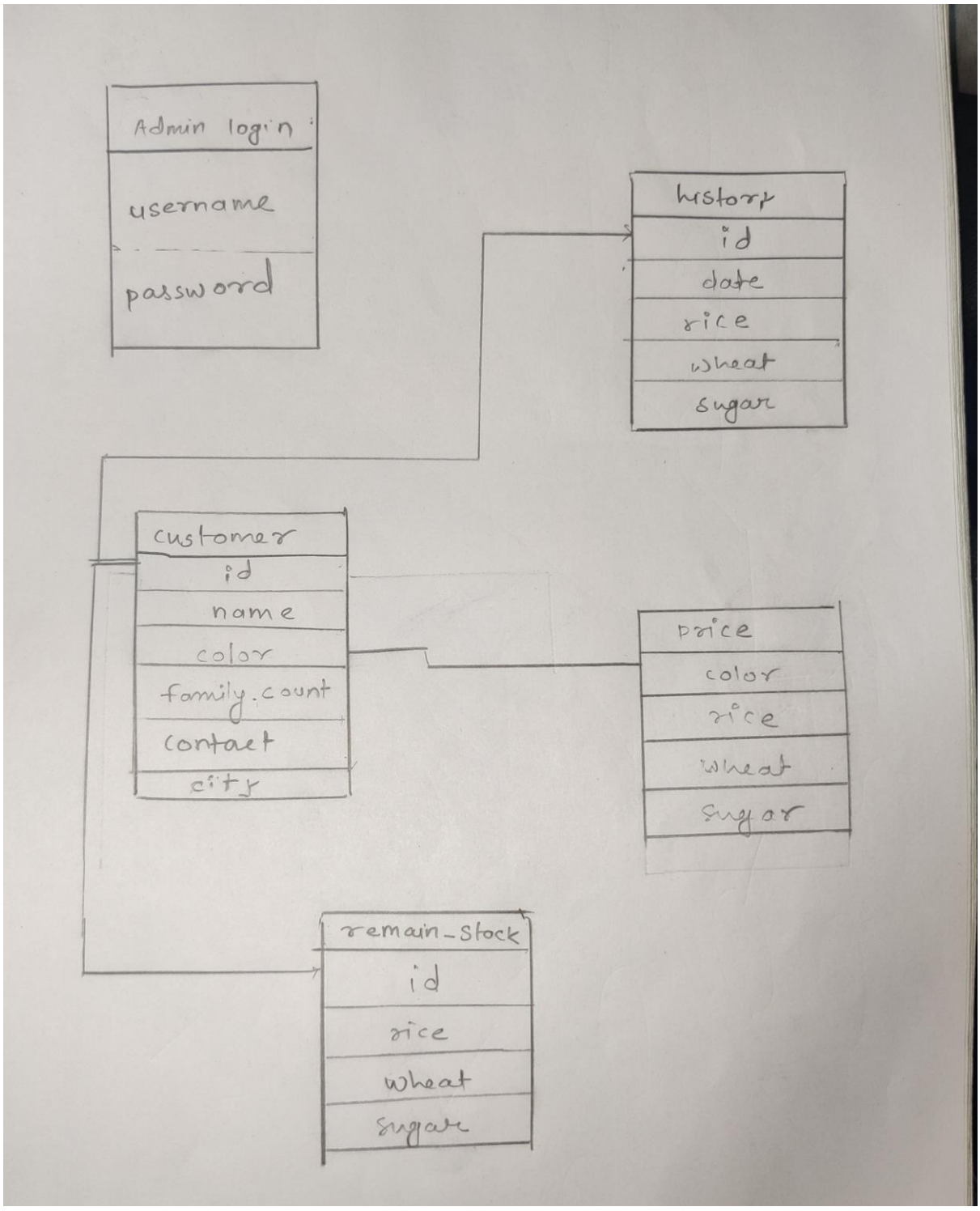
Technology used:-

- MySQL
- PHP
- HTML
- CSS

ER diagram:-


username
password
admin login
Register
customer
id
name
family_count
contact
color
gets
remain stock
sugar
id
rice
wheat
takes
foo
price
rice
color
wheat
sugar
has
History
rice
date
wheat
sugar
id

Schema:-


Admin login
username
password
history
id
date
rice
wheat
sugar
customer
id
name
color
family_count
contact
city
price
color
rice
wheat
sugar
remain_Stock
id
rice
wheat
sugar

Functional Dependency:-

- **Price**

Color->wheat

Color->rice

Color->sugar

- **History**

Id->sugar

Id->wheat

Id->sugar

Id->date

- **Remain Stock**

Id->rice

Id->wheat

Id->sugar

- **Customer**

Id->name

Id->contact

Id->family_count

Id->color

Relational Schema

Admin_login(username(primary key),

password)

Customer(id(primary key),

name,

family_count,

contact,

color(foreign key))

Remain_stock(id(foreignkey),

rice,

wheat,

sugar)

History(id(foreign key),

date,

rice,

wheat,

sugar)

Price(color(primary key),

rice,

wheat,

sugar)

Normalized table:-

- **Admin login**

1NF-no multivalued
2NF-no partial dependency
3NF –no transitive dependency

Given in functional dependency

- **Customer info**

1NF-no multivalued
2NF-no partial dependency
3NF –no transitive dependency

Given in functional dependency

- **History**

1NF-no multivalued
2NF-no partial dependency
3NF –no transitive dependency

Given in functional dependency

- **Price**

1NF-no multivalued
2NF-no partial dependency
3NF –no transitive dependency

Given in functional dependency

- **Remain stock**

1NF-no multivalued
2NF-no partial dependency
3NF –no transitive dependency

Given in functional dependency