Bangladesh University of Professionals (BUP)

Faculty of Science & Technology

Dept. of Information & Communication Technology

Lab Project On:

Bakery Management System

Submitted To

Name: Md. Jaber Al Nahian
Designation: Lecturer, ICT, BUP
Course: Database Management System Lab.
Course code : ICE-2206

Name: Md. Hasan Al Banna
Designation: Lecturer, ICT, BUP
Course: Database Management System Lab.
Course code : ICE-2206

Submitted By

Team Name: Catbus Passengers

Members:

Fariha Rahman Toha (2054901081)

Fariha Zaman Nishat (2054901008)

Khadiza Akter (2054901067)

Fabiha Kabir Aishwarya (2054901071)

Tasfia Zaima (2054901093)

Semester: 4th
Batch: BICE-2020

Submitted on: 7 September 2021

Abstract:

The main objective of the Online Bakery Management System to manage information about Item Category, Shopping Cart, Customer Information, Orders. This system allows both customer and owner to manage their daily activities of buying and selling baked treats via taking orders online and delivered safely. This project is built totally on administrator's end thus, only the administrator is guaranteed the access. The purpose of this project is to build an interface which includes all the required information and reduces the manual work. The database containing detailed information focuses on organized information regarding the customers and their orders. Whereas, the available baked treats and newly added ones are also added in the category through this system. This system also tracks Sale reports, Orders, Shopping cart.

TABLE OF CONTENTS

INTRODUCTION:

Bay-king is a bakery which was formed with an aim to deliver sweet treats to make any celebration at the top of clouds and to fulfill sudden cravings. This is a cooperative website which aims to provide more interactive menu so the ordering and exploring part can be carried out easily. In this digital era, Bay-king aims to provide online bakery services so that no occasion is hindered from the sweetness of baked treats. Other than delicious treats, this bakery also aims for ease of customers in ordering their desired treats. As a result, it also overcomes the traditional queuing system. Here, customers don't have to physically go to shop to buy their desired delicacies also administrators don't need to keep records manually. With a menu section, customers can easily track their orders. The user interface of this system is set focusing at ease in ordering items from customer's end. Since, most ordered items are placed at the topmost of 'Explore', it becomes easy for the customers to order their desired treats. Shopping cart section is used in this system so that customers can store their desired treats and buy or remove from the cart later if they want. The payment option over here is totally online based. For secure ordering and ease of performing query from administrative end individual ID is given to each customers. Also, separate IDs are made for the baked items in the menu. It also updates the sold items and total revenue in a systematic way which can only be accessed by the administrative side. This system also counts collected revenue after every transaction and available treats. It also tries to manage good performance and better service to clients. In all, this system aims for a hassle free environment to solely order baked treats.

FUNCTIONS OF THE SYSTEM:

The homepage contains all the required section which can be accessed by both customer and administrative side. As illustrated in the **FIGURE 1** most of the sections and sub-sections can be accessed from homepage.

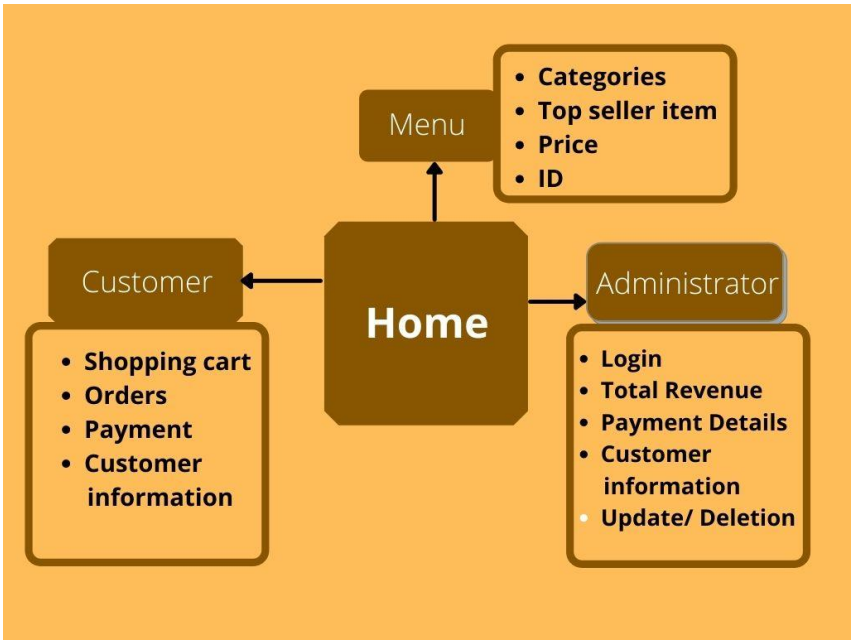


FIGURE 1: WORKFLOW FROM HOMEPAGE TO OTHER SECTIONS FROM DIFFERENT ENDS.

Below are the basic functions included in this project.

CUSTOMER'S END:

1. **Home:** The focus in this page is the items for sale. This is mainly responsible for displaying all or portion of data to the users. This section contains information regarding this service in its footer section. This section also contains the login sub-section which redirects to database on administrator's end.

2. **Contact Information:** This section contains the contact information of this Bakery. Customers may have various queries which can be solved through mutual communication between customer and service provider. Also, due to unavoidable circumstances, orders get delayed or cannot be delivered on that particular day. This section helps to get the answer to customer's query. Also, in today's age, social networking sites are used for business as well as promotions. This section contains link to the account on those sites which help people to track this bakery as well as helps in promotion of this bakery as well.

3. **About us:** This section gives a summary about this system. This section also contains the aim of forming this bakery and the story behind this opening this bakery. It also gives a brief description about the quality of this bakery's treats and it's achievements as well.

4. **Search:** Search bar is mainly used to find desired treats. Sometimes, customers forget their favorite delicacy's name. This section helps customer to find their desired treats as relevant treats are sorted if any particular alphabet is written on the search bar. This also helps customers to know easily whether their desired treat is available or not.

5. **Menu:** This section contains all the items available in the inventory in real-time. Also different sub-sections are added to the menu as preferences vary from person to person. There will be a category for variety in baked treats. Items such as- donuts, croissants, pastries, baked buns, cupcake and many other items are displayed in this section. Some prefer croissants, while some prefer pastries. Also, menus are updated based on various occasions.

6. **Explore:** This section is for exploring different items of the menu i.e. top seller items, treats for festivals etc. This can also be called as the only place to explore an assorted variety of baked delicacies. Customers are able to browse through the menu section. Each item will have a brief description about the ingredients used and price of the item (excluding VAT).

7. **My Cart:** This section helps customers to store their products after choosing them. Sometimes, people don't have enough budget available to pay for their desired treats. In that case, products can be simply put on the cart and it can be bought if it's still available on the menu.

8. **Order:** There will also be a section containing all the orders taken in a day. If any customer buys under the same contact multiple times a day, that is also stored in this section. However, it is only visible to the owner of the handles of this website.

9. **More Information:** More information regarding this sytem can be found in the footer section as well as this separate sun-section as well. More information regarding this bakery e.g. how to contact them, customized orders and others can also be found in this system.

ADMINISTRATION END:

1. **Admin:** Through this section, administrators can login to the database and monitor changes and see sale reports throughout the certain period as well.

2. **Sale Report:** Once an order is confirmed and payment is cleared, the information acquired is stored in database and the delivery process starts. The database containing required information about the customers can only be accessed by the admin section. Total revenue, items sold under individual customer's name, whether the item is delivered or not all the information are under this section. This section contains total sales of the day. It also contains details about which category is ordered the most throughout the day and total revenue collected

3. **Deletion/Update:** Sometimes, some of the item remains unavailable and some are not produced anymore due to lack of demand. Those items are deleted immediately from the menu. Also, new items are updated in the menu as a new item is created.

4. **Delivery:** Since delivery process is the most complex out of all, it is handled with much caution so that the orders are not switched with any other customers.

Entity-Relationship Diagram of Bakery Management System:

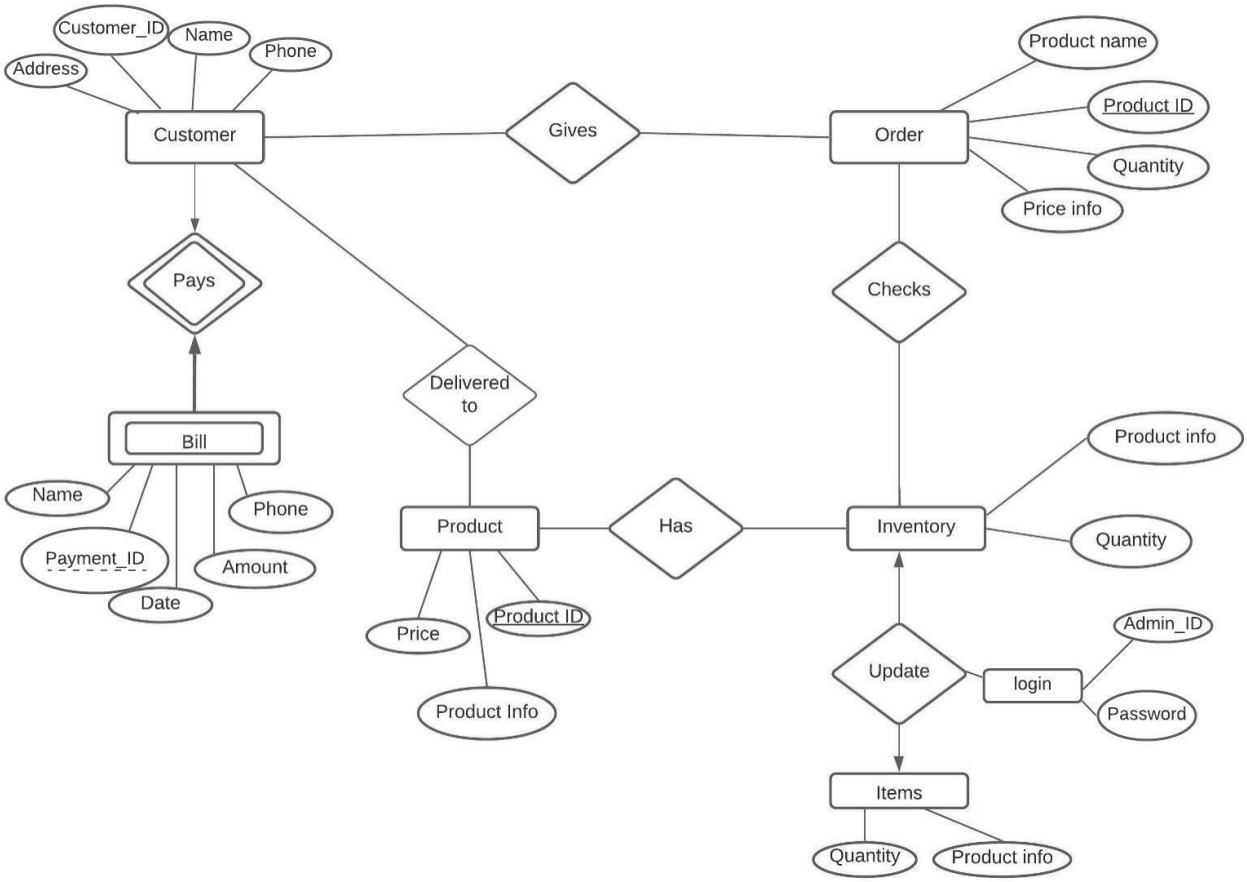


FIGURE 2: ENTITY RELATIONSHIP DIAGRAM OF BAKERY MANAGEMENT SYSTEM

DESCRIPTION:

The Entity Relationship diagram gives a graphical representation of the database. ER model shows relationship among entity sets. An entity set is a group of similar entities and these entities may also have attributes. ER diagram shows complete logical structure of the database by showing relationship among table of attributes.

In the ER diagram of Bakery Management System, 6 entity sets and 1 weak entity set is present. To place an order the customers need to provide their name, phone number and address. An ID is generated which acts as the key attribute. Also product related information e.g. product name, quantity, price information and etc are needed to place an order. In this case, Product_ID acts as the key attribute. To buy the mentioned delicacy, it has to be present in the inventory thus inventory includes product information and quantity as attribute. If the treat is present then a decrement in quantity happens in the inventory. Also updating the items in the inventory can be performed from the administrative end which requires admin_id and password to login. The ordered treat is delivered to the customer as the customer confirms their treats through product_id, price and description. Since bill is a weak entity, it can only be identified by owner entity's primary key. Here, payment_ID is the unique key.

The ER diagram portrayed above in **Figure-2** shows relationship among different tables of attributes of Bakery Management System.

PROGRESS REPORT:

Summary:

Functionalities of this Bakery Management System were finalized. A demo of workflow was made. ER models were built based on this project. A discussion on different attributes was made. Analyses on cardinalities were also made while making the ER model. Taking the ease of end users in consideration and smooth workflow from administrative side as well, this ER model was made. Exploring web answered various query related to this project also made us aware of the unknown sectors. Other than sitting group-wise, individual effort was also made by all the team members in finding necessary information regarding this project. Exchange of idea through discussions was also made. Also a huge amount of time went in browsing and brainstorming about the functionalities of this system.

Discussions were made regarding the upcoming stages. Also, views were exchanged regarding ways of implementation and moving forward.

Different goals were set for each member to accomplish within the reported time. Most of them are focused on studying different functionalities and noting down what's within our limits and what are our limitations.

GROUP PROGRESS REPORT:

Team Name: **Catbus Passengers**		**Reporting Period: 07/09/2021**
Team Member	**Total Hours (Period)** **Time spent on Discussions and Meetings**	**Remarks** (Signature)
Fariha Zaman Nishat (2054901008)	**8 hours**	Nishat
Khadiza Akter (2054901067)		Khadiza
Fabiha Kabir Aishwarya (2054901071)		Fabiha
Fariha Rahman Toha (2054901081)		Toha
Tasfia Zaima (2054901093)		Tasfia Zaima
Group Totals: 5		

INDIVIDUAL REPORT:

Team Name: Catbus Passengers

MEMBER NAME: FARIHA ZAMAN NISHAT
ROLL NO: 2054901008
Reporting Period: 07/09/2021

Date	Hours	Activities
25/08/2021	2	Watched HTML and PHP tutorials on YouTube. Attended group discussion regarding next steps to be taken.
28/08/2021	2	Learned more about making ER diagrams. Practiced drawing some diagrams. Explored online tools for ER model as well.
29/08/2021	1.5	Gone through online PDF and lecture slides for ER diagram of the project. Attended the group meeting as well.
30/08/2021	2.5	Attended group meeting and visualized the user interface.
31/08/2021	1	Visualized workflow of this system
03/09/2021	0.5	Watched similar project ideas on YouTube to get better idea on the project I'm working on.
05/09/2021	0.5	Watched videos on ER diagrams on YouTube
07/09/2021	2.5	Wrote individual assessment. Attended the group meeting for final revision.
Total:	12.5	

Team Name: Catbus Passengers

MEMBER NAME: KHADIZA AKTER
ROLL NO: 2054901067
Reporting Period: 07/09/2021

Date	Hours	Activities
24/08/2021	1	Installed Visual studio code and XAMPP for project purpose and fixed settings, added Extensions to VSC. Then I watched YouTube videos on how to works.
25/08/2021	0.5	Attended group discussion regarding next steps to be taken
26/08/2021	2	Went through lecture materials to be sure of some things, watched YouTube videos on Project making, videos on how bakery management works.
27/08/2021	2	Took certain notes on what points can be recommended adding to the project, watched videos on ER Model components and learned thoroughly about the cardinalities.
28/08/2021	1	Started working on my individual perspective on how the ER diagram should look like for the project and finished Working on ER diagram
29/08/2021	2	Sent my ER diagram to team leader and teammates for further reviewing. Watched videos on CSS and html
30/08/2021	1.5	Attended group meeting
31/08/2021	2	Resumed watching videos on CSS and html and how it all combines into a project and works perfectly
02/09/2021	1	Watched videos on CSS and html and did some notes that will come handy during the project making

04/09/2021	1	Started watching videos on PHP as well.But mostly studied CSS and html
07/09/2021	1.67	Worked on my individual report. Attended the group meeting for final revision.
Total:	15.67	

Team Name: Catbus Passengers

MEMBER NAME: FABIHA KABIR AISHWARYA
ROLL NO: 2054901071
Reporting Period: 07/09/2021

Date	Hours	Activities
25/08/2021	2	Learned details on ER model through lecture notes and lecture slides. Attended group discussion. Cleared some basic confusion by watching YouTube videos. Prepared a draft workflow and designed a rough ER model
28/08/2021	2	Searched for systematic ways to write project proposal. Watched some HTML and CSS tutorials on YouTube.
30/08/2021	1.5	Attended group meeting
01/09/2021	2	Watched related clips to the project on YouTube to develop this project further. Browsed on some topics I found confusing in the videos to learn in details.
04/09/2021	0.5	Collected information regarding particular sections of the project assigned to me after team meeting.
06/09/2021	1	Watched some tutorials on relevant topics
07/09/2021	2	Worked on my individual report. Attended the group meeting for final revision.
Total:	11	

Team Name: Catbus Passengers

MEMBER NAME: FARIHA RAHMAN TOHA
ROLL NO: 2054901081
Reporting Period: 07/09/2021

Date	Hours	Activities
25/08/2021	2.5	Attended the group discussion. Revised lecture materials to comprehend the concept of ER model. Tried to draw a few flowcharts on 'Lucidchart'.
26/08/2021	1.5	Researched for choosing this project and went through a few online abstracts regarding this project
28/08/2021	2	Tried HTML responsive web design and revised some tag descriptions
30/08/2021	2	Attended the group meeting and discussed on the chosen ER model. A brief discussion was made regarding how the individual reports and preset goals can be achieved. Took notes on necessary servers and compilers
31/08/2021	1.5	Resumed working on the demo website design. Also had a brainstorming session to write the abstract and functionalities of this project.
04/09/2021	1	Resumed working on the demo website design again
05/09/2021	1	Started finalizing the writings on the functionalities of the system
07/09/2021	2.5	Worked on individual report. Joined the meeting and made amendments based on team member's views. Re-organized the whole report.
Total:	14	

Team Name: Catbus Passengers

MEMBER NAME: TASFIA ZAIMA
ROLL NO: 2054901093
Reporting Period: 07/09/2021

Date	Hours	Activities
25/08/2021	0.5	Attended the group discussion
26/08/2021	3	Went through theory class notes about the ER model, its attributes, the relationship between entities and essential constraints or cardinalities associated to it as well as watched some more videos about it to extend own understanding
27/08/2021	2	Prepared a rough ER model for the project and looked up examples of how to write project proposals on the internet.
30/08/2021	1.5	Attended the group discussion
01/09/2021	2	Watched some tutorials and installed the essential software to do HTML and CSS work.
03/09/2021	2.25	Looked into different programming languages, such as java and PHP, to see what we could employ in the project to get the results we wanted.
05/09/2021	2.15	Looked at a few bakery websites to get a better idea of what people expect from a website in terms of information regarding bakery products.
07/09/2021	1.2	Completed the individual report by analyzing what had been learned over the previous weeks.
Total:	14.6	

PHASE-2

UPDATED ENTITY-RELATIONSHIP DIAGRAM OF BAKERY MANAGEMENT SYSTEM:

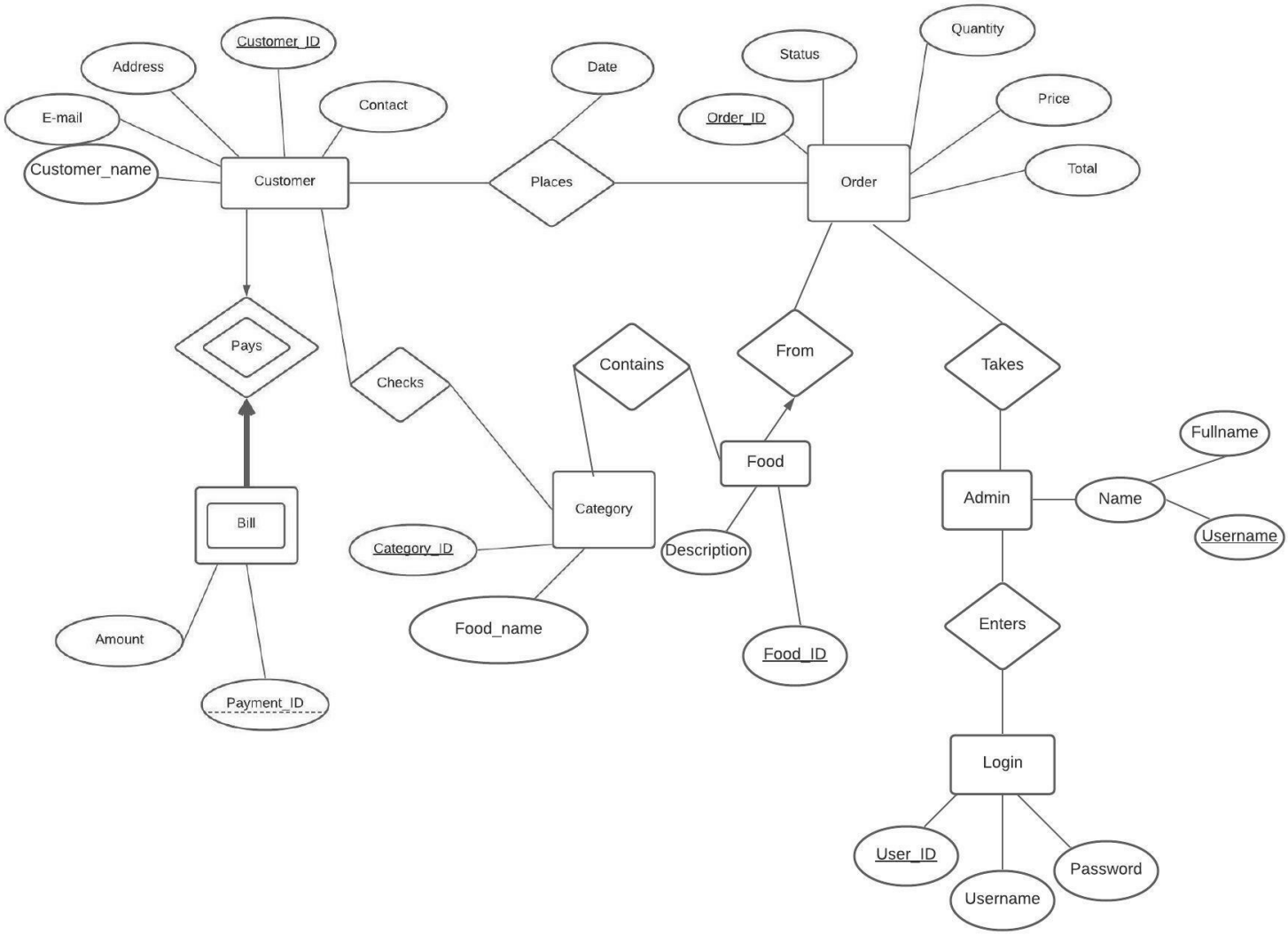


FIGURE 3: UPDATED ENTITY RELATIONSHIP DIAGRAM OF BAKERY MANAGEMENT SYSTEM

DESCRIPTION:

The Entity Relationship diagram gives a graphical representation of the database. ER model shows relationship among entity sets. An entity set is a group of similar entities and these entities may also have attributes. ER diagram shows complete logical structure of the database by showing relationship among table of attributes.

In the ER diagram of Bakery Management System, 6 entity sets and 1 weak entity set is present. To place an order the customers need to provide their name, phone number, E-mail and address. An ID is generated which acts as the primary key. Also order related information e.g. order name, quantity, price, status, total and etc are needed to place an order. In this case, Order_ID acts as the key attribute. Non-primary key status shows whether the ordered treat is available or not. To buy the mentioned delicacy, it has to be present in the food storage thus food includes food_ID and description as attribute. If the treat is present then it is shown in the category. Also updating the items in the inventory can be performed from the administrative end which requires user_id, username and password to login. The admin attribute contains a multivalued attribute, name. The ordered treat is delivered to the customer as the customer confirms their treats through payment_id, price and amount. Since bill is a weak entity, it can only be identified by owner entity's primary key. Here, payment_ID is the unique key.

The ER diagram portrayed above in **Figure-3** shows relationship among different tables of attributes of Bakery Management System.

RELATIONAL SCHEMA:

Schema before Normalization:

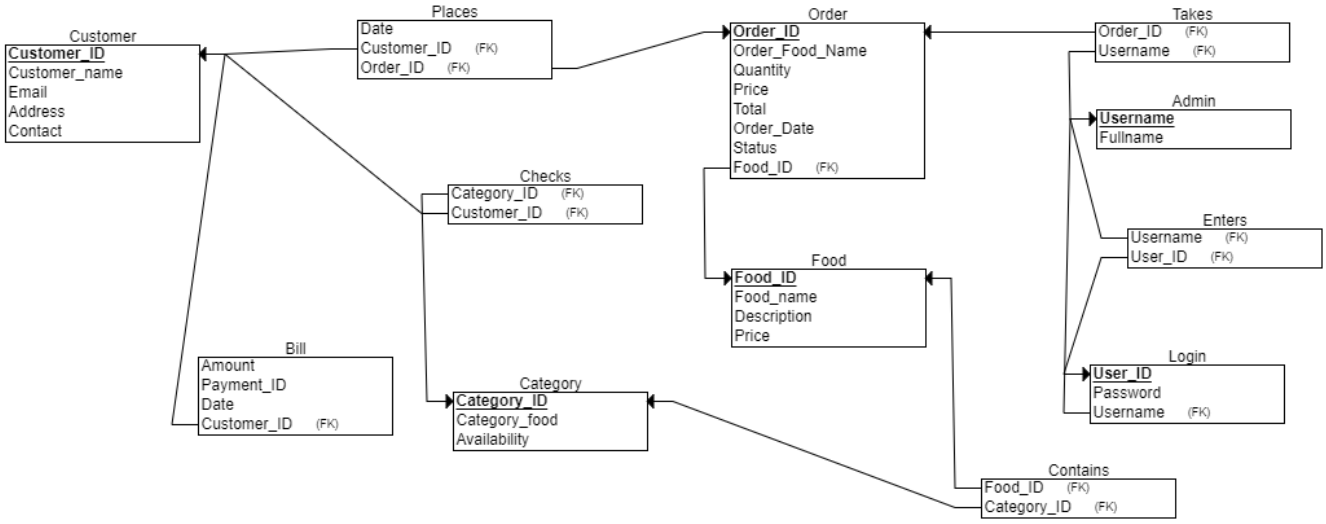


Figure 4: Relational Schema of Bakery Management System

Schema after Normalization:

:

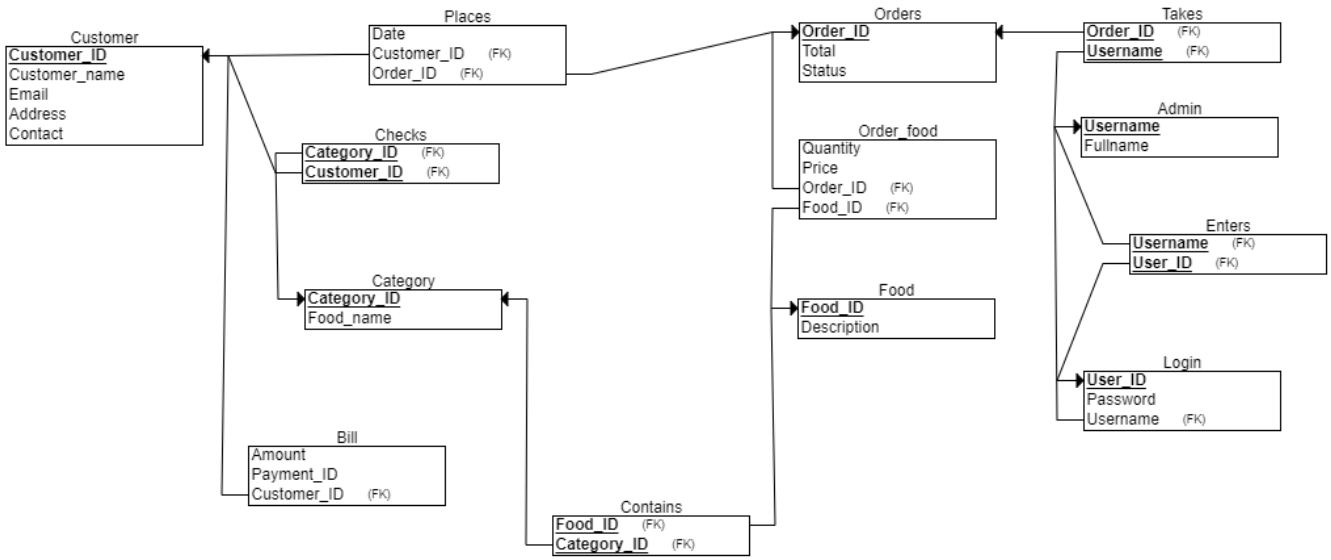


Figure 5: Refined Relational Schema of Bakery Management System

FUNCTIONAL DEPENDENCY:

Customer (Customer_ID, Customer_name, Email, Address, Contact):

Customer_ID (PK) ⟶ Customer_name

Customer_ID (PK) ⟶ Email

Customer_ID (PK) ⟶ Address

Customer_ID (PK) ⟶ Contact

Places (Customer_ID, Order_ID, Date):

Customer_ID(PK,FK) ⟶ Date,Order_ID (FK)

Order_ID(FK) ⟶ Date,Customer_ID (PK,FK)

Checks (Customer_ID, Category_ID):

Customer_ID(PK,FK) ⟶ Category_ID (FK)

Category (Category_ID, Food_name):

Category_ID (PK) ⟶ Food_name

Bill (Amount, Payment_ID, Customer_ID):

Customer_ID (PK,FK) ⟶ Amount, Payment_ID

Contains (Food_ID, Category_ID):

Category_ID(PK,FK) ⟶ Food_ID (PK,FK)

Food_ID(PK,FK) ⟶ Category_ID (PK,FK)

Orders (Order_ID, Total, Status):

Order_ID (PK) ⟶ Total

Order_ID (PK) ⟶ Status

Order_food (Order_ID, Food_ID, Price, Quantity):

Order_ID(PK,FK) ⟶ Food_ID (PK,FK), Price, Quantity

Food_ID(PK,FK) ⟶ Order_ID (PK,FK), Price, Quantity

Food (Food_ID, Description):

Food_ID (PK) ⟶ Description

Takes (Order_ID, Username):

Order_ID(FK) ⟶ Username (FK)

Admin (<u>Username</u>, Fullname):

Username(PK) ⟶ Fullname

Enters (<u>User_ID</u>, <u>Username</u>):

User_ID(PK,FK) ⟶ Username(PK,FK)

Login (<u>User_ID</u>, Username, Password):

User_ID(PK) ⟶ Username(FK)

User_ID(PK) ⟶ Password

NORMALIZATION:

Table-1: Customer

1NF: The table contains attributes with atomic and domain-specific values. Every cell in this table is single-valued and has a different column name. As a result, it fulfills the 1NF form.

2NF: Besides, there is no partial dependency in this table. All the non-prime attributes like Customer_name, Email, Address and Contact are fully dependent on primary key Customer_ID. Thus, it satisfies 2NF condition too.

3NF: There are no non-primary-key attributes that are transitively dependent on the primary key in this case.

As a result, we can conclude that the table is in 3NF format.

Customer

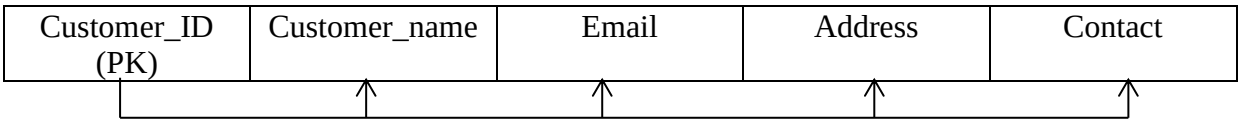


Table-2: Places

The table fulfills the 1NF 2NF and 3NF.

1NF: The attributes Date, Customer_ID and Order_ID all have atomic, single-valued and domain specific values with unique column names which satisfies the 1NF.

2NF: There is no partial dependency present here. Two foreign keys Customer_ID and Order_ID are present in the table along with data. So we can say it satisfies 2NF too.

3NF: Also there is no dependence between non-prime attributes.

Thus we can say it is already in 3NF form.

Places

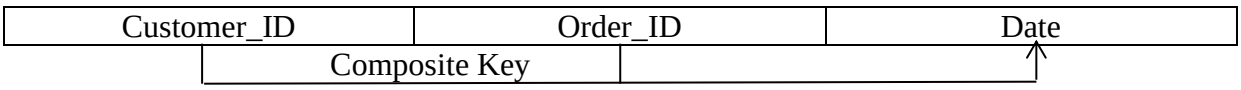


Table-3: Checks

1NF: This table, which is in 3NF format, contains two foreign keys: Category_ID and Customer_ID. These attributes have atomic single values, each with a unique column name and no duplication. 1NF is thus met.

2NF: Because there is no partial it is in 2NF

3NF: There is no transitive dependency. So, it is in 3NF form.

Checks

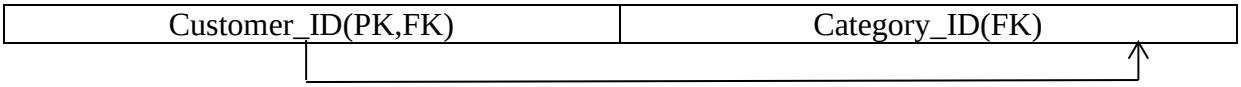


Table-4: Category

1NF: The attributes Category_ID and Food_name have atomic and domain-specific values in the table. Every cell in this table is single-valued and has a unique column name, with no overlap. As a result, it conforms to the 1NF form.

2NF: The Food name is solely dependent on the primary key Category ID, removing any possibility of partial dependency.

3NF: There are also no transitive functional dependencies that meet the 3NF form conditions.

Category

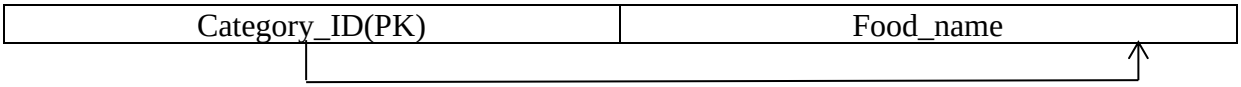


Table-5: Bill

1NF: This table contains attributes with single and atomic values, unique column names, and the same domain but no ordering. So it is in 1NF.

2NF: Here Payment_ID is a unique key of weak entity as bill is only paid when a customer places an order. A Customer_ID is used to distinguish the payment_ID of customers. There are also no partial dependencies, which fulfill 2NF conditions.

3NF: There is also no transitive dependency. So it is in 3NF.

Bill

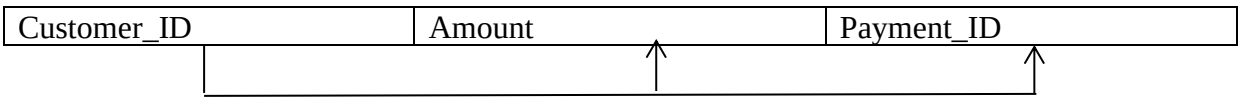


Table-6: Contains

1NF: This 3NF table has two foreign keys: Category_ID and Food_ID. These attributes have atomic single values, each with a unique column name and no overlap. As a result, 1NF is fulfilled.

2NF: It is defined in 2NF form because there is no partial dependency.

3NF: If there is no transitive dependency for non-prime attributes, the relation must be of the third normal form.

Contains

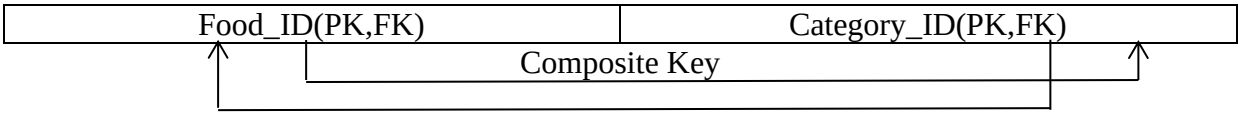


Table-7: Orders

1NF: The primary key of this table is Order_ID and other attributes are Total and Status.The attributes in the table have atomic and domain-specific values.This table's cells are all single-valued and have different column names. As a result, it conforms to the 1NF form.

2NF: Furthermore, there is no partial dependency here. The attributes are only determined by the primary key Order_ID. As a result, it is in 2NF.

3NF: The other keys total and status are not dependent on each other, but rather on the Order_ID is in. As a result, it is in 3NF form.

Orders

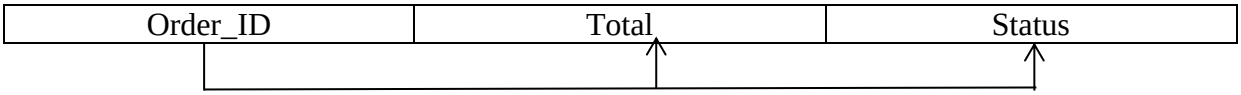


Table-8: Order_Food

1NF: The columns present in this table are quantity, price, Order_ID, and Food_ID. Here, Order_ID and Food_ID serving as foreign keys. There are single (atomic) valued attributes/columns present here, and the values stored in the columns are of the same domain with unique column names. As a result, it is in 1NF.

2NF: Aside from that, no attributes are dependent on each other, removing the risk of partial dependency and fulfilling the 2NF.

3NF: There is also no transitive dependency in this case as non-prime attributes such as Quantity and Price doesn't depend on each other. As a result, it is in 3NF.

Order_Food

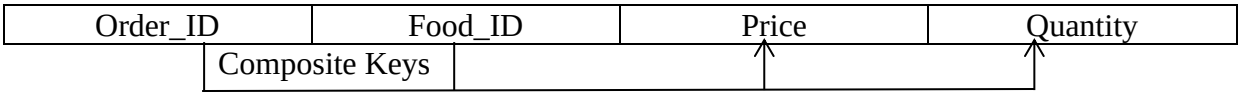


Table-9: Food

1NF: The table is already in 3NF form, which meets the requirements for 1NF, 2NF and 3NF. Food_ID and Description have atomic valued attributes in the same domain's columns, completing 1NF.

2NF: Furthermore, there is no partial dependency here because the Description is entirely dependent on the primary key Food_ID. As a result, 2NF is satisfied.

3NF: Also, there is no path for transitive dependency as non-prime attribute is dependent on prime attribute.

As a result, the table is in 3NF format.

Food

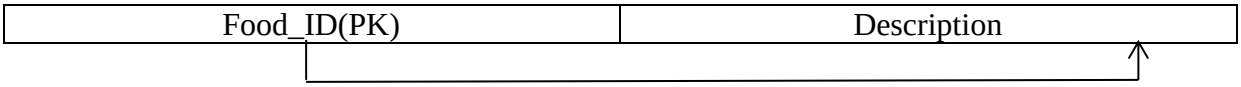


Table-10: Takes

1NF: There are two foreign keys in this table: Order_ID and Username.These attributes have atomic single values with distinct column names and no overlap. As a result, 1NF is satisfied.
2NF: It has no partial dependencies, so the table confirms to 2NF

3NF: It also does not have transitive dependency as both are foreign keys. So it confirms to 3NF.

Takes

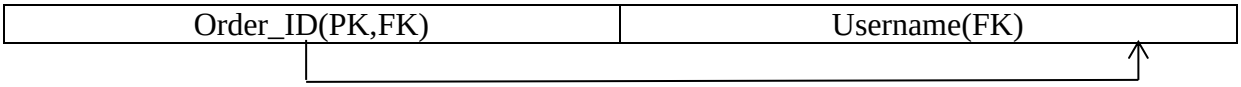


Table-11: Admin

1NF: The table is written in 1NF. The attributes Username and Fullname all have atomic, single-valued, domain-specific values with unique column names, satisfying the 1NF.

2NF: There is no partial dependency present there because Fullname is completely dependent on the primary key Username.

3NF: Furthermore, there are not enough attributes to support transitive dependency. As a result, we can say it is in 3NF.

Admin

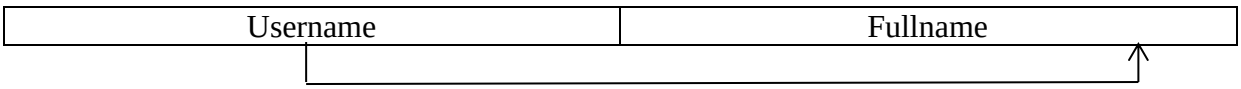


Table-12: Enters

1NF: There are two foreign keys in this table, User_ID and Username. These attributes have atomic single values with discrete columns and no intertwine. So it's a 1NF.

2NF: It has no partial dependencies, so the table matches up to 2NF.

3NF: Furthermore, no attributes are dependent on non-primary attributes. As a matter of fact, it is in 3NF.

Enters

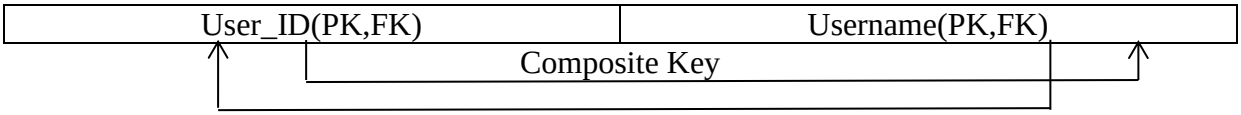


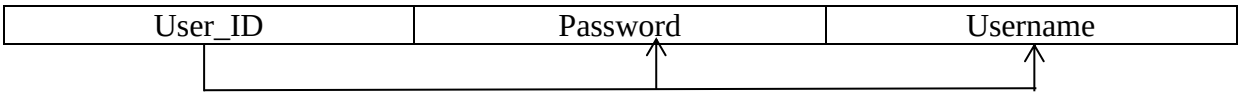
Table-13: Login

1NF: This table contains the attributes User ID Password and Username amongst which Username is a primary key and Username a foreign key. The attributes in the table have atomic and domain-specific values. This table's cells are all single-valued and have different column names. As a result, it conforms to the 1NF.

2NF: The non-prime attributes Password and foreign key Username cannot be determined without primary key User Id thus confirming 2NF.

3NF: Also Password and Username cannot determine each other. So, there is no transitive dependency and the table is in 3NF.

Login



SQL STATEMENTS:

Customer:

Table Creation:

```
Create table Customer(
        Customer_ID int NOT NULL AUTO_INCREMENT,
        Customer_name varchar(255),
        Email varchar(255),
        Address varchar(255),
        Contact varchar(255),
        PRIMARY KEY(Customer_ID)
);
```

Data Insertion:

```
INSERT INTO Customer(Customer_ID, Customer_name, Email, Address, Contact)
 Values(1001,'Riana Karim','riana1@gmail.com','442/2-C,Tejgaon','01876822222');


INSERT INTO Customer(Customer_name,Email,Address,Contact)
   Values('Omi San','0mi@yahoo.com','223,Block-B,Uttara','01923456781'),
   ('Rahima Afroze','r0se@yahoo.com','123,road-6,Banani','01934656781'),
   ('Miyoko Rahman','m1yok0@gmail.com','263,Block-A,Uttara','01928046781'),
   ('Kazi Karim','kar1m@yahoo.com','223,Block-C,Shantinagar','01923494821'),
   ('Omi Chan','0m1_c@yahoo.com','223,road no-4,Kazipara','01868456781'),
    ('Razia Banu','raz1a@gmail.com','22/3-A,Rupnagar','01829405881'),
    ('Karim Hossain','kar1m@yahoo.com','881/3-B,Gulistan','01761236781'),
    ('Namira Rahman','namU@gmail.com','43/A,Gulshan','01766667781'),
    ('Narumi Momose','m0m0@gmail.com','444/3-C,Nam Garden,Kafrul','01888875781'),
    ('Lee Chan','c_h_a_n@gmail.com','23/A,Uttara','01923456211'),
    ('Samira Hoque','5am@gmail.com','19/D,Link road','01983222281'),
    ('Kazi Nupur','f1shy@gmail.com','54/5,Block-B,Uttara','01875556781'),
    ('Eren Mia','3ren@yahoo.com','777/6-A,Gulshan-1','01833333381'),
   ('Faizan Hossain','fiza@yahoo.com','3/C,Dhanmondi','01844444981');
```

Places:

Table Creation:

```
create table Places(
Customer_ID int NOT NULL AUTO_INCREMENT,
Date varchar(255),
Order_ID int NOT NULL,
PRIMARY KEY(CUSTOMER_ID),
FOREIGN KEY(Customer_ID) REFERENCES customer(Customer_ID),
FOREIGN KEY(Order_ID) REFERENCES orders(Order_ID)
 );
```

Data Entry:

```
  Insert into places(Customer_ID, Date, Order_ID)
  Values(1001,'30/08/2021',2002);


   Insert into places(Date, Order_ID)
   Values('30/08/2021',2002),
   ('01/09/2021',2003),
   ('13/09/2021',2004),
   ('20/09/2021',2005),
   ('23/09/2021',2006),
   ('10/10/2021',2007),
   ('11/10/2021',2008),
   ('11/10/2021',2009),
   ('12/10/2021',2010),
   ('12/10/2021',2011),
   ('12/10/2021',2012),
   ('13/10/2021',2013),
   ('13/10/2021',2014),
   ('14/10/2021',2015);
```

Checks:

Table Creation:

```
Create table Checks(
Customer_ID int NOT NULL AUTO_INCREMENT,
Category_ID int NOT NULL,
PRIMARY KEY(Customer_ID),
Foreign key(Customer_ID) references customer(Customer_ID),
Foreign key(Category_ID) references Category(Category_ID)
);
```

Data Entry:

```
Insert into checks(Customer_ID,Category_ID)
Values(1001,101);

Insert into checks(Category_ID)
Values(102),
(103),
(104),
(105),
(106),
(107),
(108),
(109),
(110),
(111),
(112),
(113),
(114),
(115);
```

Category:

Table Creation:

```
Create table Category(
Category_ID int NOT NULL AUTO_INCREMENT,
Food_name varchar(255),
PRIMARY KEY(Category_ID)
);
```

Data Entry:

```
INSERT INTO Category(Category_ID,Food_name)
Values('101','Blueberry Bottom Cake');

Insert into Category(Food_name)
Values('Lemon Pie'),
('Cherry Cream Cake'),
('Cheese Dump Cake'),
('Peanut Butter Cookies'),
('Creamy Rice pudding'),
('Peaches Cream Pie'),
('Orange Fluff Cake'),
('Cinnamon Pudding Cake'),
('Creamy Coconut Cake'),
('Pineapple Cupcake'),
('Strawberry Cheesecake'),
('Gluten-free bread'),
('Combo Cookies'),
('Cherry Pie');
```

Bill:

Table Creation:

```
create table Bill(
Customer_ID int NOT NULL AUTO_INCREMENT,
Payment_ID int NOT NULL,
Amount int,
Primary Key(Customer_ID),
Foreign Key(Customer_ID) references Customer(Customer_ID)
);
```

Data Entry:

```
Insert into Bill(Customer_ID, Payment_ID, Amount)
Values(1001,88001,510);

Insert into Bill(Customer_ID,Payment_ID, Amount)
Values(1002,88002,650),
(1003,88003,1020),
(1004,88004,1520),
(1005,88005,300),
(1006,88006,800),
(1007,88007,1500),
(1008,88008,2500),
(1009,88009,580),
(1010,88010,670),
(1011,88011,590),
(1012,88011,380),
(1013,88012,400),
(1014,88013,700),
(1015,88014,900);
```

Contains:

Table Creation:

```
create table contains(
Food_ID int NOT NULL AUTO_INCREMENT,
Category_ID int NOT NULL,
PRIMARY KEY(Food_ID,Category_ID),
FOREIGN KEY(Food_ID) REFERENCES food(Food_ID),
FOREIGN KEY(Category_ID) REFERENCES category(Category_ID)
);
```

Data Entry:

```
INSERT INTO Contains(Food_ID,Category_ID)
Values(201,101);

Insert into Contains(Category_ID)
Values(102),
(103),
(104),
(105),
(106),
(107),
(108),
(109),
(110),
(111),
(112),
(113),
(114),
(115);
```

Orders:

Table Creation:

```
create table Orders(
Order_ID int NOT NULL AUTO_INCREMENT,
Total int,
Status varchar(255),
PRIMARY KEY(Order_ID)
);
```

Data Entry:

```
INSERT INTO Orders(Order_ID,Total,Status)
Values('2001','510','Available');

INSERT INTO Orders(Total,Status)
Values('650','Available'),
('1020','Available'),
('1520','Available'),
('300','Available'),
('800','Available'),
('1500','Available'),
('2500','Available'),
('580','Available'),
('670','Available'),
('590','Available'),
('340','Available'),
('350','Available'),
('400','Available'),
('50','Available');
```

Order_Food:

Table Creation:

```
create table Order_Food(
Order_ID int NOT NULL AUTO_INCREMENT,
Food_ID int NOT NULL,
Quantity int NOT NULL,
Price int NOT NULL,
PRIMARY KEY(Order_ID,Food_ID),
Foreign Key(Order_ID) references Orders(Order_ID),
Foreign Key(Food_ID) references Food(Food_ID)
);
```

Data Entry:

```
Insert into Order_Food(Order_ID, Food_ID, Quantity, Price)
Values(2001,201,1,450);

Insert into Order_Food(Food_ID, Quantity, Price)
Values(202,1,590),
(203,2,220),
(204,4,380),
(205,1,300),
(206,2,400),
(207,5,300),
(208,5,500),
(209,1,520),
(210,2,235),
(211,1,530),
(212,1,340),
(213,1,300),
(214,1,340),
(215,1,50);
```

Food:

Table Creation:

```
create table Food(
Food_ID int NOT NULL AUTO_INCREMENT,
Description varchar(255),
PRIMARY KEY(Food_ID)
);
```

Data Entry:

```
Insert into Food(Food_ID,Description)
Values(201,'Made with in season blueberries');
Insert into Food(Description)
Values('Made with Cream, egg whites, butter, lemon juice, egg yolks'),
('Made with zhite cake mix, cream cheese, cherry pie filling, cherry gelatin'),
('Made with Vanilla cake mix, whipped cream cheese filling'),
('Made with Creamy peanut butter, brown sugar, egg, baking soda'),
('Made with Grain white rice, vanilla bean paste, heavy whipping cream'),
('Made with Fresh peaches, heavy whipping cream, single crust pie, eggs'),
('Made with Fresh orange juice, cream, cake flour, butter, orange zest'),
('Made with Brown sugar, apple, butter, cinnamon, baking powder'),
('Made with Condensed coconut milk, coconut cream, self raising flour'),
('Made with Sour cream, pineapple juice, crushed pineapple'),
('Made with Cream cheese, sour cream, graham cracker crumbs'),
('Made with Almond flour, brown sugar'),
('Made with Egg, chocolate chips, sugar, raisin'),
('Made with Cherry, lemon juice, cake flour, sugar');
```

Takes:

Table Creation:

```
create table Takes(
Order_ID int NOT NULL AUTO_INCREMENT,
Username varchar(255),
Primary key(Order_ID),
Foreign key(Order_ID) references Orders(Order_ID),
Foreign key(Username) references Admin(Username)
);
```

Data Entry:

```
insert into  takes(Order_ID,Username)
values(2001,'Nishat');


insert into takes(Username)
values('Khadiza'),
('Fabiha'),
('Toha'),
('Tasfia'),
('Toha'),
('Khadiza'),
('Nishat'),
('Nishat'),
('Nishat'),
('Fabiha'),
('Toha'),
('Toha'),
('Tasfia'),
('Khadiza');
```

Admin:

Table Creation:

```
Create table Admin(
username varchar(255),
fullname varchar(255),
PRIMARY KEY(username)
);
```

Data Entry:

```
Insert into Admin(username,fullname)
Values('Nishat','Fariha Zaman Nishat'),
('Fabiha','Fabiha Kabir'),
('Khadiza','Khadiza Akter'),
('Tasfia','Tasfia Zaima'),
('Toha','Fariha Rahman Toha');
```

Enters:

Table Creation:

```
create table Enters(
User_ID INT NOT NULL,
Username varchar(255),
Primary key(User_ID,Username),
Foreign key(User_ID) references Login(User_ID),
Foreign key(Username)references Admin(Username)
 );
```

Data Entry:

```
 Insert into Enters(User_ID,Username)
 Values(2054901008,'Nishat'),
 (2054901067,'Khadiza'),
 (2054901071,'Fabiha'),
 (2054901081,'Toha'),
 (2054901093,'Tasfia');
```

Login:

Table Creation:

```
create table Login(
User_ID int NOT NULL,
Password varchar(255),
Username varchar(255),
Primary key(User_ID),
Foreign key(Username) references Admin(Username)
);
```

Data Entry:

```
Insert into Login(User_ID, Password, Username)
Values(2054901008,'CATBUS','Nishat'),
(2054901067,'CATBUS','Khadiza'),
(2054901071,'CATBUS','Fabiha'),
(2054901081,'CATBUS','Toha'),
(2054901093,'CATBUS','Tasfia');
```

DATA POPULATION:

The query statements of data insertion are in the **SQL STATEMENT** section. Output tables are as follows.

Customer:

```
MariaDB [(none)]> use dbms_project;
Database changed
MariaDB [dbms_project]> select * from customer;
+-------------+---------------+-------------------+--------------------------+-------------+
| Customer_ID | Customer_name | Email             | Address                  | Contact     |
+-------------+---------------+-------------------+--------------------------+-------------+
|        1001 | Riana Karim   | riana1@gmail.com  | 442/2-C,Tejgaon          | 01876822222 |
|        1002 | Omi San       | 0mi@yahoo.com     | 223,Block-B,Uttara       | 01923456781 |
|        1003 | Rahima Afroze | r0se@yahoo.com    | 123,road-6,Banani        | 01934656781 |
|        1004 | Miyoko Rahman | m1yok0@gmail.com  | 263,Block-A,Uttara       | 01928046781 |
|        1005 | Kazi Karim    | kar1m@yahoo.com   | 223,Block-C,Shantinagar  | 01923494821 |
|        1006 | Omi Chan      | 0m1_c@yahoo.com   | 223,road no-4,Kazipara   | 01868456781 |
|        1007 | Razia Banu    | raz1a@gmail.com   | 22/3-A,Rupnagar          | 01829405881 |
|        1008 | Karim Hossain | kar1m@yahoo.com   | 881/3-B,Gulistan         | 01761236781 |
|        1009 | Namira Rahman | namU@gmail.com    | 43/A,Gulshan             | 01766667781 |
|        1010 | Narumi Momose | m0m0@gmail.com    | 444/3-C,Nam Garden,Kafrul | 01888875781 |
|        1011 | Lee Chan      | c_h_a_n@gmail.com | 23/A,Uttara              | 01923456211 |
|        1012 | Samira Hoque  | 5am@gmail.com     | 19/D,Link road           | 01983222281 |
|        1013 | Kazi Nupur    | f1shy@gmail.com   | 54/5,Block-B,Uttara      | 01875556781 |
|        1014 | Eren Mia      | 3ren@yahoo.com    | 777/6-A,Gulshan-1        | 01833333381 |
|        1015 | Faizan Hossain | fiza@yahoo.com   | 3/C,Dhanmondi            | 01844444981 |
+-------------+---------------+-------------------+--------------------------+-------------+
15 rows in set (0.016 sec)
```

Places:

```
ariaDB [(none)]> use dbms_project;
atabase changed
ariaDB [dbms_project]> select * from places;
+-------------+------------+----------+
| Customer_ID | Date       | Order_ID |
+-------------+------------+----------+
|        1001 | 25/08/2021 |     2001 |
|        1002 | 30/08/2021 |     2002 |
|        1003 | 01/09/2021 |     2003 |
|        1004 | 13/09/2021 |     2004 |
|        1005 | 20/09/2021 |     2005 |
|        1006 | 23/09/2021 |     2006 |
|        1007 | 10/10/2021 |     2007 |
|        1008 | 11/10/2021 |     2008 |
|        1009 | 11/10/2021 |     2009 |
|        1010 | 12/10/2021 |     2010 |
|        1011 | 12/10/2021 |     2011 |
|        1012 | 12/10/2021 |     2012 |
|        1013 | 13/10/2021 |     2013 |
|        1014 | 13/10/2021 |     2014 |
|        1015 | 14/10/2021 |     2015 |
+-------------+------------+----------+
5 rows in set (0.031 sec)
```

Checks:

```
MariaDB [dbms_project]> select * from checks;
+-------------+-------------+
| Customer_ID | Category_ID |
+-------------+-------------+
|        1001 |         101 |
|        1002 |         102 |
|        1003 |         103 |
|        1004 |         104 |
|        1005 |         105 |
|        1006 |         106 |
|        1007 |         107 |
|        1008 |         108 |
|        1009 |         109 |
|        1010 |         110 |
|        1011 |         111 |
|        1012 |         112 |
|        1013 |         113 |
|        1014 |         114 |
|        1015 |         115 |
+-------------+-------------+
15 rows in set (0.009 sec)
```

Category:

```
MariaDB [dbms_project]> select * from category;
+-------------+----------------------+
| Category_ID | Food_name            |
+-------------+----------------------+
|         101 | Blueberry Bottom Cake |
|         102 | Lemon Pie            |
|         103 | Cherry Cream Cake    |
|         104 | Cheese Dump Cake     |
|         105 | Peanut Butter Cookies |
|         106 | Creamy Rice pudding  |
|         107 | Peaches Cream Pie    |
|         108 | Orange Fluff Cake    |
|         109 | Cinnamon Pudding Cake |
|         110 | Creamy Coconut Cake  |
|         111 | Pineapple Cupcake    |
|         112 | Strawberry Cheesecake |
|         113 | Gluten-free bread    |
|         114 | Combo Cookies        |
|         115 | Cherry Pie           |
+-------------+----------------------+
15 rows in set (0.014 sec)
```

Bill:

```
MariaDB [dbms_project]> select * from bill;
+-------------+------------+--------+
| Customer_ID | Payment_ID | Amount |
+-------------+------------+--------+
|        1001 |      88001 |    510 |
|        1002 |      88002 |    650 |
|        1003 |      88003 |   1020 |
|        1004 |      88004 |   1520 |
|        1005 |      88005 |    300 |
|        1006 |      88006 |    800 |
|        1007 |      88007 |   1500 |
|        1008 |      88008 |   2500 |
|        1009 |      88009 |    580 |
|        1010 |      88010 |    670 |
|        1011 |      88011 |    590 |
|        1012 |      88011 |    380 |
|        1013 |      88012 |    400 |
|        1014 |      88013 |    700 |
|        1015 |      88014 |    900 |
+-------------+------------+--------+
15 rows in set (0.009 sec)
```

Contains:

```
MariaDB [dbms_project]> select * from contains;
+---------+-------------+
| Food_ID | Category_ID |
+---------+-------------+
|     201 |         101 |
|     202 |         102 |
|     203 |         103 |
|     204 |         104 |
|     205 |         105 |
|     206 |         106 |
|     207 |         107 |
|     208 |         108 |
|     209 |         109 |
|     210 |         110 |
|     211 |         111 |
|     212 |         112 |
|     213 |         113 |
|     214 |         114 |
|     215 |         115 |
+---------+-------------+
15 rows in set (0.005 sec)
```

Orders:

```
MariaDB [dbms_project]> select * from orders;
+----------+-------+-----------+
| Order_ID | Total | Status    |
+----------+-------+-----------+
|     2001 |   510 | Available |
|     2002 |   650 | Available |
|     2003 |  1020 | Available |
|     2004 |  1520 | Available |
|     2005 |   300 | Available |
|     2006 |   800 | Available |
|     2007 |  1500 | Available |
|     2008 |  2500 | Available |
|     2009 |   580 | Available |
|     2010 |   670 | Available |
|     2011 |   590 | Available |
|     2012 |   340 | Available |
|     2013 |   350 | Available |
|     2014 |   400 | Available |
|     2015 |    50 | Available |
+----------+-------+-----------+
15 rows in set (0.018 sec)
```

Order_food:

```
ariaDB [dbms_project]> select * from order_food;
----------+---------+----------+-------+
 Order_ID | Food_ID | Quantity | Price |
----------+---------+----------+-------+
     2001 |     201 |        1 |   450 |
     2002 |     202 |        1 |   590 |
     2003 |     203 |        2 |   220 |
     2004 |     204 |        4 |   380 |
     2005 |     205 |        1 |   300 |
     2006 |     206 |        2 |   400 |
     2007 |     207 |        5 |   300 |
     2008 |     208 |        5 |   500 |
     2009 |     209 |        1 |   520 |
     2010 |     210 |        2 |   235 |
     2011 |     211 |        1 |   530 |
     2012 |     212 |        1 |   340 |
     2013 |     213 |        1 |   300 |
     2014 |     214 |        1 |   340 |
     2015 |     215 |        1 |    50 |
----------+---------+----------+-------+
5 rows in set (0.006 sec)
```

Food:

```
MariaDB [dbms_project]> select * from food;
+---------+-------------------------------------------------------------------------------+
| Food_ID | Description                                                                   |
+---------+-------------------------------------------------------------------------------+
|     201 | Made with in-season blueberries                                               |
|     202 | Made with Cream, egg whites, butter, lemon juice, egg yolks                   |
|     203 | Made with White cake mix, cream cheese, cherry pie filling, cherry gelatin    |
|     204 | Made with Vanilla cake mix, whipped cream cheese filling                      |
|     205 | Made with Creamy peanut butter, brown sugar, egg, baking soda                 |
|     206 | Made with Grain white rice, vanilla bean paste, heavy whipping cream          |
|     207 | Made with Fresh peaches, heavy whipping cream, single crust pie, eggs         |
|     208 | Made with Fresh orange juice, cream, cake flour, butter, orange zest          |
|     209 | Made with Brown sugar, apple, butter, cinnamon, baking powder                 |
|     210 | Made with Condensed coconut milk, coconut cream, self raising flour           |
|     211 | Made with Sour cream, pineapple juice, crushed pineapple                      |
|     212 | Made with Cream cheese, sour cream, graham cracker crumbs                     |
|     213 | Made with Almond flour, brown sugar                                           |
|     214 | Made with Egg, chocolate chips, sugar, raisin                                 |
|     215 | Made with Cherry, lemon juice, cake flour, sugar                              |
+---------+-------------------------------------------------------------------------------+
15 rows in set (0.009 sec)
```

Takes:

```
MariaDB [dbms_project]> select * from takes;
+----------+----------+
| Order_ID | Username |
+----------+----------+
|     2003 | Fabiha   |
|     2011 | Fabiha   |
|     2002 | Khadiza  |
|     2007 | Khadiza  |
|     2015 | Khadiza  |
|     2001 | Nishat   |
|     2008 | Nishat   |
|     2009 | Nishat   |
|     2010 | Nishat   |
|     2005 | Tasfia   |
|     2014 | Tasfia   |
|     2004 | Toha     |
|     2006 | Toha     |
|     2012 | Toha     |
|     2013 | Toha     |
+----------+----------+
15 rows in set (0.005 sec)
```

Admin:

```
MariaDB [dbms_project]> select * from admin;
+----------+---------------------+
| username | fullname            |
+----------+---------------------+
| Fabiha   | Fabiha Kabir        |
| Khadiza  | Khadiza Akter       |
| Nishat   | Fariha Zaman Nishat |
| Tasfia   | Tasfia Zaima        |
| Toha     | Fariha Rahman Toha  |
+----------+---------------------+
5 rows in set (0.018 sec)
```

Enters:

```
MariaDB [dbms_project]> select * from enters;
+------------+----------+
| User_ID    | Username |
+------------+----------+
| 2054901008 | Nishat   |
| 2054901067 | Khadiza  |
| 2054901071 | Fabiha   |
| 2054901081 | Toha     |
| 2054901093 | Tasfia   |
+------------+----------+
5 rows in set (0.004 sec)
```

Login:

```
MariaDB [dbms_project]> select * from login;
+------------+----------+----------+
| User_ID    | Password | Username |
+------------+----------+----------+
| 2054901008 | CATBUS   | Nishat   |
| 2054901067 | CATBUS   | Khadiza  |
| 2054901071 | CATBUS   | Fabiha   |
| 2054901081 | CATBUS   | Toha     |
| 2054901093 | CATBUS   | Tasfia   |
+------------+----------+----------+
5 rows in set (0.005 sec)
```

PROGRESS REPORT:

Summary:

ER model and database of this Bakery Management System were finalized. A demo of workflow was made. Some changes were made in the latest ER model. Focus of all the group members was directed to backend section. A refined schema was made and it was normalized to third normal form. After schema design was finalized, query statements for different tables were divided among the members and each did their part. Exchange of idea through discussions was also made. Also a huge amount of time went in browsing and brainstorming about the flexibility of newly formed database.

Discussions were made regarding the upcoming stages. Also, views were exchanged regarding ways of implementation and moving forward.

Different goals were set for each member to accomplish within the reported time. Most of them are focused on studying different functionalities and noting down what's within our limits and what are our limitations.

GROUP PROGRESS REPORT:

Team Name: **Catbus Passengers**		**Reporting Period: 18/10/2021**
Team Member	**Total Hours (Period)** **Time spent on Discussions and Meetings**	**Remarks** (Signature)
Fariha Zaman Nishat (2054901008)	**10 hours**	Nishat
Khadiza Akter (2054901067)		Khadiza
Fabiha Kabir Aishwarya (2054901071)		Fabiha
Fariha Rahman Toha (2054901081)		Toha
Tasfia Zaima (2054901093)		Tasfia Zaima
Group Totals: 5		

INDIVIDUAL REPORT:

Team Name: Catbus Passengers

MEMBER NAME: FARIHA ZAMAN NISHAT
ROLL NO: 2054901008
Reporting Period: 18/10/2021

Date	Hours	Activity
01/10/2021	2	Attended group meeting
02/10/2021	3	Installed Xampp in my laptop. Unfortunately, faced problems starting mysql in it. There seems to be a problem with port every time so Mysql keeps shutting down unexpectedly. Tried almost every video in YouTube, still didn't work. So, I left it at that.
03/10/2021	1	Watched project related videos on YouTube.
05/10/2021	2	Tried fixing xampp again. Took help from friends. Still didn't work. Attended group meeting.
07/10/2021	2	Attended the discussion on relational schema design.
12/10/2021	1	Made rough schema for group.
13/10/2021	2	Attended group meeting where we chose the new ER diagram and schema. Talked about normalizing the schema.
14/10/2021	4	Watched videos on normalizing tables in database and checked the normal form of our project tables. Sent the explanation to our group leader.
15/02/2021	2	Attended group meeting. We divided the tables among us to write query for data entry.

16/10/2021	2	Wrote the query of the tables in my part. Sent it to our group leader.
18/10/2021	1	Wrote individual assessment for the lap report of phase-2.
Total	22	

Team Name: Catbus Passengers

MEMBER NAME: KHADIZA AKTER
ROLL NO: 2054901067
Reporting Period: 18/10/2021

Date	Hours	Activity
01/10/2021	2	Attended group meeting
05/10/2021	1	Attended group meeting
07/10/2021	2	Attended group discussion on schema design.
10/10/2021	0.5	Watched videos on Database Schema
11/10/2021	1	Encountered some port problems in My sql, Xampp, php admin server. Tried to solve but failed
13/10/2021	4	Solved the mentioned problem above. Attended group meeting
14/10/2021	1	Again watched useful contents for DBMS Schema and made a draft Schema from the selected ER model.
15/10/2021	2	Completed my Schema and Tried to figure out in which NF it belongs to
16/10/2021	2	Made some tables & queries in database according to the schema.
18/10/2021	1.5	Attended the meeting and did individual meeting
Total	17	

Team Name: Catbus Passengers

MEMBER NAME: FABIHA KABIR AISHWARYA
ROLL NO: 2054901071
Reporting Period: 18/10/2021

Date	Hours	Activity
11/09/2021	1.5	Watched some related tutorials on YouTube to decide the preferred language
17/09/2021	3	Learnt to prepare a schema from ER model by watching lecture materials and tutorials. Prepared an individual draft schema for the project.
30/09/2021	1	Tried to install xampp. faced a problem at first so uninstalled mysql then reinstalled xampp after which it was fine. Practiced to use xampp.
01/10/2021	2	Attended the group meeting
05/10/2021	2	Attended the group meeting
06/10/2021	3	Studied normalization by revising class lectures and consulting some reference books.
08/10/2021	1	Revised mysql query statement. Contributed in preparing database tables for phase 2 schemas.
13/10/2021	2	Attended the group meeting
14/10/2021	2	Worked on particular sections of the project which was assigned to me
15/10/2021	2	Attended the group meeting.
18/10/2021	1	Accumulated my records of time spent and effort given to this project as an individual report which were noted down time to time. Attended the meeting.
Total	20.5	

Team Name: Catbus Passengers

MEMBER NAME: FARIHA RAHMAN TOHA
ROLL NO: 2054901081
Reporting Period: 18/10/2021

Date	Hours	Activity
26/09/2021	1	Did a crash course on PHP and PDO
27/09/2021	2	Continued completing the crash course
28/09/2021	1	Did a course on MySQL
01/10/2021	2	Attended the group meeting and discussed necessary steps to be taken for upcoming phase.
03/10/2021	2	Finished the crash course on PHP and PDO for beginners
05/10/2021	2	Attended the group meeting
06/10/2021	1	Made a few amendments in ER diagram and tried to make it more flexible
07/10/2021	1	Made a demo for relational schema
08/10/2021	1	Tried to look for ways to normalize a relational schema up to desired normal form
10/10/2021	2	Finalized the schema without normalization.
13/10/2021	2	Attended the meeting and normalized the schema along with fellow teammates
14/10/2021	1	Checked the functional dependencies of the normalized schema along with teammates.
15/10/2021	2	Attended the group meeting and tasks were assigned to all the members in writing SQL statements
16/10/2021	3	Wrote SQL statements of all the tables and entered the data given by teammates
18/10/2021	1	Attended the meeting and finished writing the report.
Total	23	

Team Name: Catbus Passengers

MEMBER NAME: TASFIA ZAIMA
ROLL NO: 2054901093
Reporting Period: 18/10/2021

Date	Hours	Activity
01/10/2021	2	Attended the group meeting
02/10/2021	2.5	I studied how to create a schema from an E R diagram and attempted to create one from our project's E R diagram.
05/10/2021	2	Attended the group meeting
08/10/2021	3	I studied normalization 1NF, 2NF, 3NF, and BNCF by watching YouTube videos and going over what we learned in class. Then, tried normalizing the prepared schema and make it achieve the 3NF normalized form
12/10/2021	2	I learned a little about PHP and HTML from watching videos.
13/10/2021	2	Attended the meeting
14/10/2021	1.5	Did some SQL syntax study. Practiced creating database tables and adding, updating, and deleting values in those tables using mysql query statements to gain a better understanding of the prior lab work.
15/10/2021	4.15	Used MySQLl query statements to prepare several database tables for phase 2 schemas. Attended the group meeting
16/10/2021	2	For project proposal phase-2, I researched how to locate functional dependencies and attempted to identify functional dependencies in the schema prepared for our proposal.
18/10/2021	1	Attended the meeting and prepared individual report
Total	22.15	