University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: __2_____

Date: ___October 15, 2024_____

Group Number: ____99_____

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Alex Kwok	70386099		a.kwok0191@gmail.com
Byeori Kim	74612821		bk.byeori.kim@gmail.com
Jun Lee	21913603		Junemessi040714@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

2. Summary of Project

This project models the operation of food charities to enable food donation and redistribution management. The application operates within the intersection of food logistics, food donor management, food recipient tracking, and food inventory level optimization. By connecting these four pillars to our database, we enable an informative information system to track the operation success of food charities.

3. Schema

With respect to the following schemas, attributes which are underlined denote primary key attributes, and attributes which are bolded denote foreign key attributes.

- Charities(CharityID: int, Address: varchar(255), Province: varchar(50), Name: varchar(100))
- DonationEvent(EventID: int, **CharityID**: int, Address: varchar(255), EventDate: date, EventName: varchar(100))
    - CharityID is a foreign key referring to Charities
- Recipients(SinNum: char(9), **EventID**: int, Age: int, Address: varchar(255), ContactNum: varchar(15), Gender: varchar(10))
    - EventID is a foreign key referring to DonationEvent
- Food(FoodID: int, FoodName: varchar(100), FoodDescription: varchar(255), FoodType: varchar(50))
- FoodItem(ReceiptNum: int, **FoodID**: int, **InventoryID**: int, **CharityID**: int NOT NULL, ExpirationDate: date, Quantity: int, Name: varchar(100))
    - InventoryID is a foreign key referring to Inventory
    - CharityID is a foreign key referring to Charities
    - FoodID is a foreign key referring to Food
- FoodReceived(FoodID: int, **SinNum**: char(9), Quantity: int, **InventoryQuantity**: int)
    - FoodID is a foreign key referring to Food
    - SinNum is a foreign key referring to Recipients
- Inventory(InventoryID: int, **CharityID**: int NOT NULL, Quantity: int, Status: varchar(50), Location: varchar(255))

- CharityID is a foreign key referring to Charities

- DonationReceipt(ReceiptNum: int, Date: date, **DonorID**: int NOT NULL)
  - DonorID is a foreign key referring to FoodDonors
- FoodDonors(DonorID: int, Name: varchar(100), Type: varchar(50), **ReceiptNum**: int)
- CharityWorker(PhoneNumber: varchar(15), Status: varchar(50), Name: varchar(100), **CharityID**: int, StartDate**:** date)
  - CharityID is a foreign key referring to Charities
- Staff(**PhoneNumber:** varchar(15), AdminCode: varchar(10), Salary: decimal(10, 2))
- Volunteers(**PhoneNumber**: varchar(15), VolunteerHours: int)
- Donate(CharityID: int, DonorID: int)

4. ER-Diagram

Please refer to final page of document.

5. Functional Dependencies

Charities
CharityID → Address, Province, Name
We can assume that Address → Province (since an address determines a province, i.e., no address can be in two different provinces)

DonationEvent
EventID → CharityID, Address, EventDate, EventName
We might add a non-PK FD: EventName → EventDate (assuming each event name can only occur once on a particular date)

Recipients
SinNum → EventID, Age, Address, ContactNum, Gender
We can add a reasonable non-PK FD: ContactNum → Address (assuming that for recipients, one phone number can be associated with one address)

Food
FoodID → FoodName, FoodDescription, FoodType, FoodItemID

Non-PK FD: FoodName → FoodType (for instance, specific food names imply the type of food, such as "Apple" implying it's a "Fruit")

FoodItem
(ReceiptNum, FoodID) → InventoryID, CharityID, ExpirationDate, Quantity, Name

FoodReceived
(FoodID, SinNum) → Quantity, InventoryQuantity

Inventory
InventoryID → CharityID, Quantity, Status, Location
Non-PK FD: Status → Quantity (perhaps inventory items with certain statuses tend to have a certain range of quantities)

DonationReceipt
ReceiptNum → Date, DonorID
Non-PK FD: DonorID → Date (assuming a donor can only donate once per day)

FoodDonors
DonorID → Name, Type, ReceiptNum

CharityWorker
(PhoneNumber, CharityID) → Status, Name, StartDate
Non-PK FD: PhoneNumber → Name (a person's phone number determines their name)

Staff
PhoneNumber → AdminCode, Salary
Non-PK FD: AdminCode → Salary (certain admin codes can determine a salary range)

Volunteers
PhoneNumber → VolunteerHours

Donate
(CharityID, DonorID) → (No other attributes)

Below are context-based functional dependencies:
Address → Province (in Charities)

EventName → EventDate (in DonationEvent)

ContactNum → Address (in Recipients)

FoodName → FoodType (in Food)

Name → ExpirationDate (in FoodItem)

Status → Quantity (in Inventory)

DonorID → Date (in DonationReceipt)

PhoneNumber → Name (in CharityWorker)

AdminCode → Salary (in Staff)

6. Normalization

1. Charities

- Normalization Steps:
  - Since Address → Province, we should decompose into two tables: one for Charities and another for AddressProvince.
- Final Tables:
  - Charities(CharityID: int, Address: varchar(255), Name: varchar(100), InventoryID: int)
    - PK: CharityID
  - AddressProvince(Address: varchar(255), Province: varchar(50))
    - PK: Address

2. DonationEvent

- Normalization Steps:
  - Decompose to remove the partial dependency EventName → EventDate.
- Final Tables:
  - DonationEvent(EventID: int, CharityID: int, Address: varchar(255), EventName: varchar(100))
    - PK: EventID
    - FK: CharityID references Charities(CharityID)
  - EventNameDate(EventName: varchar(100), EventDate: date)
    - PK: EventName

3. Recipients

- Normalization Steps:
  - Decompose to remove the partial dependency ContactNum → Address.
- Final Tables:
  - Recipients(SinNum: char(9), EventID: int, Age: int, ContactNum: varchar(15), Gender: varchar(10))
    - PK: SinNum
    - FK: EventID references DonationEvent(EventID)
  - ContactAddress(ContactNum: varchar(15), Address: varchar(255))
    - PK: ContactNum

4. Food

- Normalization Steps:
  - Decompose to remove the partial dependency FoodName → FoodType.
- Final Tables:
  - Food(FoodID: int, FoodName: varchar(100), FoodDescription: varchar(255))
    - PK: FoodID
  - FoodNameType(FoodName: varchar(100), FoodType: varchar(50))
    - PK: FoodName

5. FoodItem

- Normalization: Already in 3NF (composite key and no partial dependencies).
- Final Table: No change
  - FoodItem(ReceiptNum: int, FoodID: int, InventoryID: int, CharityID: int, ExpirationDate: date, Quantity: int, Name: varchar(100))
    - PK: (ReceiptNum, FoodID)
    - FK: InventoryID references Inventory(InventoryID)
    - FK: CharityID references Charities(CharityID)

6. FoodReceived

- Normalization: Already in 3NF.
- Final Table: No change
  - FoodReceived(FoodID: int, SinNum: char(9), Quantity: int, InventoryQuantity: int)
    - PK: (FoodID, SinNum)
    - FK: SinNum references Recipients(SinNum)
    - FK: FoodID references Food(FoodID)

7. Inventory

- Normalization Steps:
  - Decompose to remove the partial dependency Status → Quantity.
- Final Tables:
  - Inventory(InventoryID: int, CharityID: int, Quantity: int, Location: varchar(255))
    - PK: InventoryID
    - FK: CharityID references Charities(CharityID)
  - StatusQuantity(Status: varchar(50), Quantity: int)
    - PK: Status

8. DonationReceipt

- Normalization Steps:
  - Decompose to remove the partial dependency DonorID → Date.

- Final Tables:
  - DonationReceipt(ReceiptNum: int, Date: date, DonorID: int)
    - PK: ReceiptNum
    - FK: DonorID references FoodDonors(DonorID)
  - DonorDate(DonorID: int, Date: date)
    - PK: DonorID

9. FoodDonors

- Normalization: Already in 3NF
- Final Table: No change
  - FoodDonors(DonorID: int, Name: varchar(100), Type: varchar(50), ReceiptNum: int)
    - PK: DonorID
    - FK: ReceiptNum references DonationReceipt(ReceiptNum)

10. CharityWorker

- Normalization Steps:
  - Decompose to remove the partial dependency PhoneNumber → Name.
- Final Tables:
  - CharityWorker(PhoneNumber: varchar(15), CharityID: int, Status: varchar(50), StartDate: date)
    - PK: (PhoneNumber, CharityID)
    - FK: Char

7. SQL DDL Statements

```
CREATE TABLE Charities (
   CharityID INT PRIMARY KEY,
   Address VARCHAR(255) NOT NULL,
   Name VARCHAR(100) NOT NULL,
   UNIQUE (Address),
);

CREATE TABLE AddressProvince (
   Address VARCHAR(255) PRIMARY KEY,
   Province VARCHAR(50) NOT NULL
);

CREATE TABLE DonationEvent (
   EventID INT PRIMARY KEY,
   CharityID INT NOT NULL,
   Address VARCHAR(255) NOT NULL,
   EventName VARCHAR(100) NOT NULL,
   FOREIGN KEY (CharityID) REFERENCES Charities(CharityID),
```

```
    FOREIGN KEY (Address) REFERENCES AddressProvince(Address)
);

CREATE TABLE EventNameDate (
    EventName VARCHAR(100) PRIMARY KEY,
    EventDate DATE NOT NULL
);

CREATE TABLE Recipients (
    SinNum CHAR(9) PRIMARY KEY,
    EventID INT NOT NULL,
    Age INT NOT NULL,
    ContactNum VARCHAR(15) NOT NULL,
    Gender VARCHAR(10),
    FOREIGN KEY (EventID) REFERENCES DonationEvent(EventID)
);

CREATE TABLE ContactAddress (
    ContactNum VARCHAR(15) PRIMARY KEY,
    Address VARCHAR(255) NOT NULL,
    FOREIGN KEY (Address) REFERENCES AddressProvince(Address)
);
CREATE TABLE Food (
    FoodID INT PRIMARY KEY,
    FoodName VARCHAR(100) NOT NULL,
    FoodDescription VARCHAR(255),
    UNIQUE (FoodName)
);

CREATE TABLE FoodNameType (
    FoodName VARCHAR(100) PRIMARY KEY,
    FoodType VARCHAR(50) NOT NULL
);

CREATE TABLE FoodItem (
    ReceiptNum INT NOT NULL,
    FoodID INT NOT NULL,
    InventoryID INT NOT NULL,
    CharityID INT NOT NULL,
    ExpirationDate DATE NOT NULL,
    Quantity INT NOT NULL,
    Name VARCHAR(100) NOT NULL,
    PRIMARY KEY (ReceiptNum, FoodID),
    FOREIGN KEY (InventoryID) REFERENCES Inventory(InventoryID),
    FOREIGN KEY (CharityID) REFERENCES Charities(CharityID),
    FOREIGN KEY (FoodID) REFERENCES Food(FoodID)
```

```
);

CREATE TABLE FoodReceived (
    FoodID INT NOT NULL,
    SinNum CHAR(9) NOT NULL,
    Quantity INT NOT NULL,
    InventoryQuantity INT NOT NULL,
    PRIMARY KEY (FoodID, SinNum),
    FOREIGN KEY (FoodID) REFERENCES Food(FoodID),
    FOREIGN KEY (SinNum) REFERENCES Recipients(SinNum)
);

CREATE TABLE Inventory (
    InventoryID INT PRIMARY KEY,
    CharityID INT NOT NULL,
    Quantity INT NOT NULL,
    Location VARCHAR(255) NOT NULL,
    FOREIGN KEY (CharityID) REFERENCES Charities(CharityID)
);

CREATE TABLE StatusQuantity (
    Status VARCHAR(50) PRIMARY KEY,
    Quantity INT NOT NULL
);

CREATE TABLE DonationReceipt (
    ReceiptNum INT PRIMARY KEY,
    Date DATE NOT NULL,
    DonorID INT NOT NULL,
    FOREIGN KEY (DonorID) REFERENCES FoodDonors(DonorID)
);

CREATE TABLE DonorDate (
    DonorID INT PRIMARY KEY,
    Date DATE NOT NULL
);

CREATE TABLE FoodDonors (
    DonorID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Type VARCHAR(50),
    ReceiptNum INT NOT NULL,
    FOREIGN KEY (ReceiptNum) REFERENCES DonationReceipt(ReceiptNum)
);
```

```
CREATE TABLE CharityWorker (
    PhoneNumber VARCHAR(15) NOT NULL,
    CharityID INT NOT NULL,
    Status VARCHAR(50),
    StartDate DATE NOT NULL,
    PRIMARY KEY (PhoneNumber, CharityID),
    FOREIGN KEY (CharityID) REFERENCES Charities(CharityID)
);

-- Staff table for workers who are staff
CREATE TABLE Staff (
    PhoneNumber VARCHAR(15) PRIMARY KEY,
    AdminCode VARCHAR(10),
    Salary DECIMAL(10, 2),
    FOREIGN KEY (PhoneNumber) REFERENCES CharityWorker(PhoneNumber),
    CHECK (PhoneNumber NOT IN (SELECT PhoneNumber FROM Volunteers))
);


CREATE TABLE Volunteers (
    PhoneNumber VARCHAR(15) PRIMARY KEY,
    VolunteerHours INT NOT NULL,
    FOREIGN KEY (PhoneNumber) REFERENCES CharityWorker(PhoneNumber),
    CHECK (PhoneNumber NOT IN (SELECT PhoneNumber FROM Staff))
);


CREATE TABLE Donate (
    CharityID INT NOT NULL,
    DonorID INT NOT NULL,
    PRIMARY KEY (CharityID, DonorID),
    FOREIGN KEY (CharityID) REFERENCES Charities(CharityID),
    FOREIGN KEY (DonorID) REFERENCES FoodDonors(DonorID)
);
```

8. Insert Statements

```
INSERT INTO Charities (CharityID, Address, Name) VALUES
(1, '123 Charity St', 'Helping Hands'),
(2, '456 Kindness Ave', 'Food for All'),
(3, '789 Giving Rd', 'Care and Share'),
(4, '101 Hope Blvd', 'Charity for Change'),
(5, '202 Compassion Ln', 'Goodwill');

INSERT INTO AddressProvince (Address, Province) VALUES
```

```
('123 Charity St', 'Ontario'),
('456 Kindness Ave', 'British Columbia'),
('789 Giving Rd', 'Alberta'),
('101 Hope Blvd', 'Quebec'),
('202 Compassion Ln', 'Manitoba');

INSERT INTO DonationEvent (EventID, CharityID, Address, EventName) VALUES
(1, 1, '123 Charity St', 'Holiday Food Drive'),
(2, 2, '456 Kindness Ave', 'Summer BBQ'),
(3, 3, '789 Giving Rd', 'Thanksgiving Feast'),
(4, 4, '101 Hope Blvd', 'Winter Clothes Distribution'),
(5, 5, '202 Compassion Ln', 'Community Gathering');

INSERT INTO EventNameDate (EventName, EventDate) VALUES
('Holiday Food Drive', '2024-12-01'),
('Summer BBQ', '2024-06-15'),
('Thanksgiving Feast', '2024-10-10'),
('Winter Clothes Distribution', '2024-12-15'),
('Community Gathering', '2024-05-01');


INSERT INTO Recipients (SinNum, EventID, Age, ContactNum, Gender) VALUES
('123456789', 1, 34, '555-1234', 'Male'),
('987654321', 2, 45, '555-9876', 'Female'),
('112233445', 3, 29, '555-1122', 'Male'),
('223344556', 4, 60, '555-2233', 'Female'),
('334455667', 5, 18, '555-3344', 'Female');

INSERT INTO ContactAddress (ContactNum, Address) VALUES
('555-1234', '123 Charity St'),
('555-9876', '456 Kindness Ave'),
('555-1122', '789 Giving Rd'),
('555-2233', '101 Hope Blvd'),
('555-3344', '202 Compassion Ln');

INSERT INTO Food (FoodID, FoodName, FoodDescription) VALUES
(1, 'Apple', 'A fresh red apple'),
(2, 'Canned Beans', 'Canned black beans'),
(3, 'Bread', 'Whole wheat bread'),
(4, 'Rice', 'Long grain white rice'),
(5, 'Milk', '1% milk');

INSERT INTO FoodNameType (FoodName, FoodType) VALUES
('Apple', 'Fruit'),
('Canned Beans', 'Legume'),
('Bread', 'Grain'),
```

```
('Rice', 'Grain'),
('Milk', 'Dairy');

INSERT INTO FoodItem (ReceiptNum, FoodID, InventoryID, CharityID, ExpirationDate,
Quantity, Name) VALUES
(1, 1, 1, 1, '2025-01-01', 100, 'Fresh Apples'),
(2, 2, 2, 2, '2026-01-01', 50, 'Canned Beans'),
(3, 3, 3, 3, '2024-11-01', 200, 'Wheat Bread'),
(4, 4, 4, 4, '2025-07-01', 500, 'Rice Bags'),
(5, 5, 5, 5, '2024-12-01', 300, 'Cartons of Milk');

INSERT INTO FoodReceived (FoodID, SinNum, Quantity, InventoryQuantity) VALUES
(1, '123456789', 10, 90),
(2, '987654321', 5, 45),
(3, '112233445', 20, 180),
(4, '223344556', 30, 470),
(5, '334455667', 15, 285);


INSERT INTO Inventory (InventoryID, CharityID, Quantity, Location) VALUES
(1, 1, 1000, 'Warehouse 1'),
(2, 2, 500, 'Storage Room A'),
(3, 3, 200, 'Pantry B'),
(4, 4, 300, 'Fridge 1'),
(5, 5, 400, 'Storage Room C');

INSERT INTO StatusQuantity (Status, Quantity) VALUES
('Available', 100),
('Low Stock', 50),
('Out of Stock', 0),
('Surplus', 500),
('Reserved', 200);

INSERT INTO DonationReceipt (ReceiptNum, Date, DonorID) VALUES
(1, '2024-10-01', 1),
(2, '2024-11-01', 2),
(3, '2024-12-01', 3),
(4, '2024-12-15', 4),
(5, '2025-01-01', 5);

INSERT INTO DonorDate (DonorID, Date) VALUES
(1, '2024-10-01'),
(2, '2024-11-01'),
(3, '2024-12-01'),
(4, '2024-12-15'),
(5, '2025-01-01');
```

```
INSERT INTO FoodDonors (DonorID, Name, Type, ReceiptNum) VALUES
(1, 'John Doe', 'Individual', 1),
(2, 'ACME Corp', 'Corporation', 2),
(3, 'Jane Smith', 'Individual', 3),
(4, 'Charity Group A', 'Organization', 4),
(5, 'Local Market', 'Business', 5);

INSERT INTO CharityWorker (PhoneNumber, CharityID, Status, StartDate) VALUES
('555-0001', 1, 'Full-time', '2023-01-01'),
('555-0002', 2, 'Part-time', '2023-05-01'),
('555-0003', 3, 'Volunteer', '2024-06-15'),
('555-0004', 4, 'Full-time', '2022-09-01'),
('555-0005', 5, 'Volunteer', '2024-07-01'),
('555-0006', 1, 'Part-time', '2023-03-01'), -- Added to CharityWorker
('555-0008', 2, 'Full-time', '2024-01-01'), -- Added to CharityWorker
('555-0010', 3, 'Volunteer', '2024-02-01'), -- Added to CharityWorker
('555-0011', 4, 'Volunteer', '2023-08-15'), -- Added to CharityWorker
('555-0012', 5, 'Volunteer', '2023-10-01'); -- Added to CharityWorker

INSERT INTO Staff (PhoneNumber, AdminCode, Salary) VALUES
('555-0001', 'AD001', 50000.00),
('555-0002', 'AD002', 30000.00),
('555-0004', 'AD004', 60000.00),
('555-0006', 'AD005', 45000.00),
('555-0008', 'AD006', 35000.00);

INSERT INTO Volunteers (PhoneNumber, VolunteerHours) VALUES
('555-0003', 100),
('555-0005', 50),
('555-0010', 200),
('555-0011', 75),
('555-0012', 120);


INSERT INTO Donate (CharityID, DonorID) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5);
```

Food Charities Operation - ER Diagram