REPUBLIQUE DU CAMEROUN
PAIX – TRAVAIL – PATRIE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

FACULTE D'INGENIERIE ET DE TECHNOLOGIE

REPUBLIC OF CAMEROON
PEACE – WORK – FATHERLAND

MINISTRY OF HIGHER EDUCATION

FACULTY OF ENGINEERING AND TECHNOLOGY

************************** **UNIVERSITY OF BUEA** *************************

DEPARTMENT OF COMPUTER ENGINEERING
COURSE TITLE: INTERNET PROGRAMMING AND MOBILE PROGRAMING
COURSE CODE: CEF440

TASK
PRACTICAL DESIGN OF A FOOD WASTE MANAGEMENT SYSTEM

COURSE INSTRUCTOR: Dr KAMENI VALERY

Group 12 Members:

S\N	NAMES	MATRICULE
I.	AYAMBA BLESS BISSONG	FE20A018
II	NGONG MARINUS NGONG	FE20A077
III	NDIBONG ENOW EBAI-TABOT	FE20A072
IV	MBI ENOW LEONARD APELGRYN	FE20A063
V	AMINJA SIMPLICE AJONG	FE20A009

Table of content

- Software and tools used for database creation

- SQL code for creating the tables

- Inserting sample data into the tables

- Testing and validation of the database

4. Usage scenarios

- Use cases for the database, including queries and reports

- Examples of how the database can be used to reduce food waste and promote sustainability

5. Conclusion

- Summary of the database design and implementation

- Discussion of limitations and future work

- Overall assessment of the potential impact of the database

6. Appendices

- ERD diagram

- SQL code for creating the tables

- Sample queries and reports

1. Introduction

Background and motivation for the database:

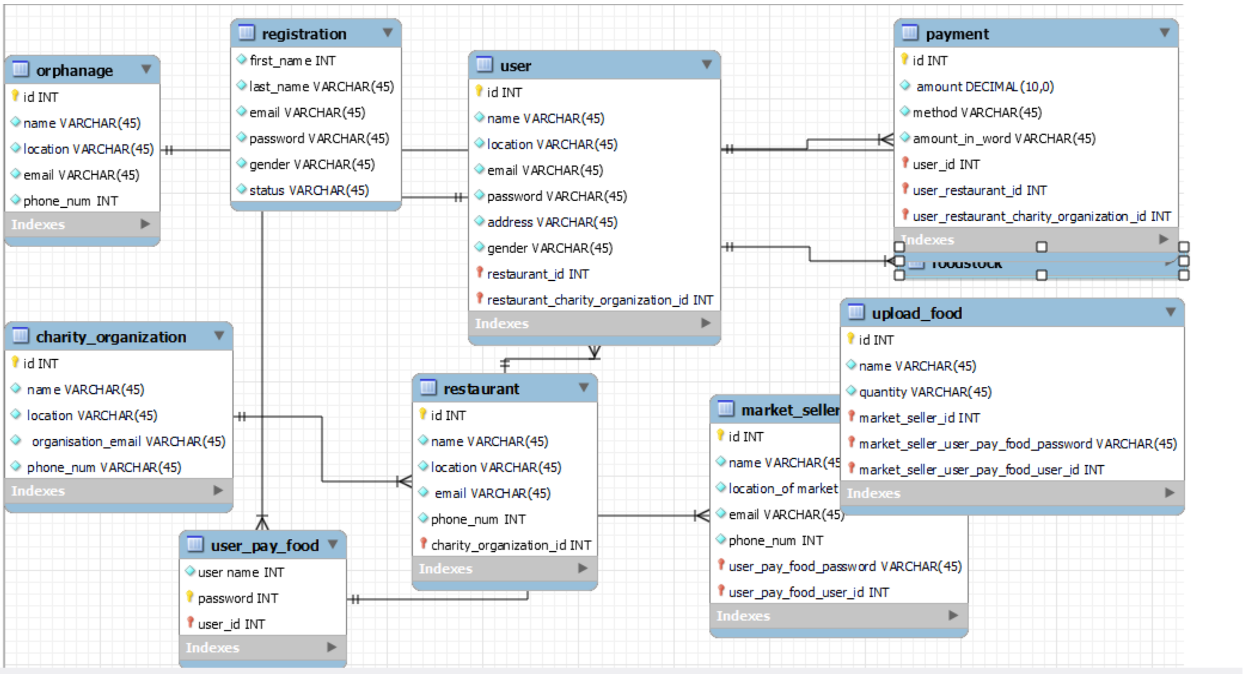
Food waste is a significant problem globally, with up to one-third of all food produced being wasted each year. This waste has a significant impact on the environment, contributing to greenhouse gas emissions and other environmental issues. To address this problem, a food waste management system can be developed to connect restaurants with charity organizations and other users who can make use of excess food.

Objectives and scope of the database:

The objective of the food waste management system is to reduce food waste by allowing restaurants to identify excess food and connect with charity organizations and other users who can make use of the food. The scope of the database includes tables for restaurants, charity organizations, users, food stock, and payments. The system will allow users to track food inventory and expiration dates, generate reports on food waste reduction and sustainability, and facilitate the transfer of excess food from restaurants to charity organizations and orphanage.

2. Database Design

Entity-relationship diagram (ERD):



The ERD for the food waste management system includes the following tables: charity_organization, restaurant, user, and foodstock. The relationships between the tables are as follows:

- A charity organization can have multiple restaurants that it works with.

- A restaurant can have multiple users and food stock items.

- A user can be associated with only one restaurant.

- A food stock item can only be associated with one restaurant.

Discussion of the tables and their attributes:

- The charity_organization table includes attributes for id, name,location, organisation_email, and phone_num.

- The restaurant table includes attributes for id, name, location, email, phone_num, and charity_organization_id.

- The user table includes attributes for id, name, location, email, password, address, and restaurant_id.

- The foodstock table includes attributes for id, name, quantity, expiry_date, and restaurant_id.

Data types and constraints used in the tables:

- The id attribute in each table is an integer with the NOT NULL constraint and UNIQUE index.

- The name, location, email, and password attributes are all strings with the NOT NULL constraint.

- The phone_num attribute is a string with the NOT NULL constraint and a maximum length of 45.

- The quantity attribute in the foodstock table is a decimal with the NOT NULL constraint.

- The expiry_date attribute in the foodstock table is a date with the NOT NULL constraint.

Relationships between the tables:

- The charity_organization table is related to the restaurant table through a one-to-many relationship, where a charity organization can work with multiple restaurants.

- The restaurant table is related to the user table through a one-to-many relationship, where a restaurant can have multiple users.

- The restaurant table is related to the foodstock table through a one-to-many relationship, where a restaurant can have multiple food stock items.

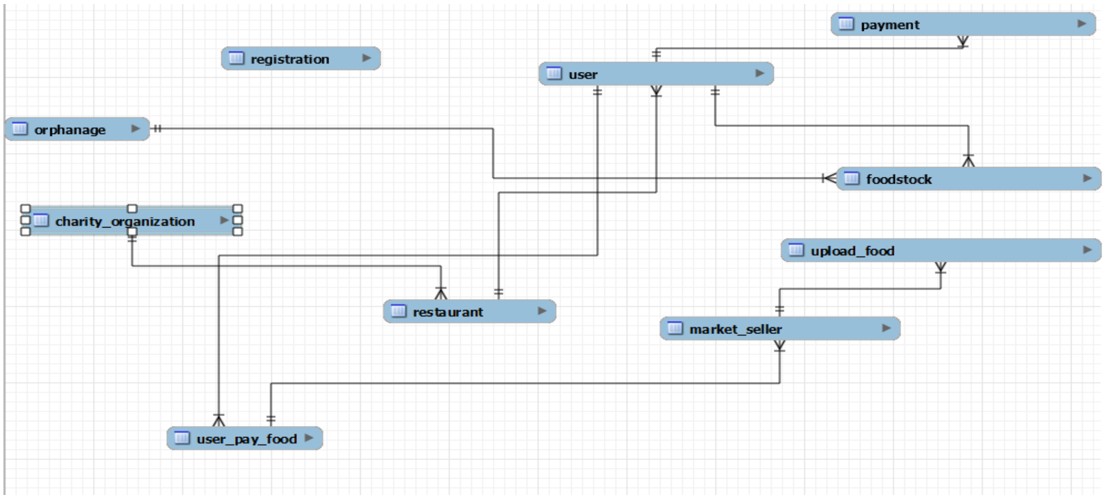
- The user table is related to the restaurant table through a one-to-one relationship, where a user can be associated with only one restaurant.

- The foodstock table is related to the restaurant table through a one-to-one relationship, where a food stock item can only be associated with one restaurant.

3. Implementation

Software and tools used for database creation:

The food waste management system can be developed using a variety of software and tools, including a database management system such as MySQL, and a programming language such as PHP or Javascript.



SQL code for creating the tables:

```
CREATE SCHEMA IF NOT EXISTS `food_waste_database` DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_0900_ai_ci;

USE `food_waste_database`;


CREATE TABLE IF NOT EXISTS `charity_organization` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `location` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `phone_num` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE
) ENGINE = InnoDB
  DEFAULT CHARACTER SET = utf8mb4
  COLLATE = utf8mb4_0900_ai_ci;


CREATE TABLE IF NOT EXISTS `restaurant` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `location` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `phone_num` VARCHAR(45) NOT NULL,
  `charity_organization_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE,
  INDEX `fk_restaurant_charity_organization_idx` (`charity_organization_id` ASC) VISIBLE,
  CONSTRAINT `fk_restaurant_charity_organization`
    FOREIGN KEY (`charity_organization_id`)
    REFERENCES `charity_organization` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
) ENGINE = InnoDB
  DEFAULT CHARACTER SET = utf8mb4
  COLLATE = utf8mb4_0900_ai_ci;

CREATE TABLE IF NOT EXISTS `user` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `location` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `password` VARCHAR(45) NOT NULL,
  `address` VARCHAR(45) NOT NULL,
  `restaurant_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE,
  INDEX `fk_user_restaurant_idx` (`restaurant_id` ASC) VISIBLE,
  CONSTRAINT `fk_user_restaurant`
    FOREIGN KEY(`restaurant_id`)
    REFERENCES `restaurant` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
) ENGINE = InnoDB
  DEFAULT CHARACTER SET = utf8mb4
  COLLATE = utf8mb4_0900_ai_ci;

CREATE TABLE IF NOT EXISTS `foodstock` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `quantity` DECIMAL(10,2) NOT NULL,
  `expiry_date` DATE NOT NULL,
  `restaurant_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE,
  INDEX `fk_foodstock_restaurant_idx` (`restaurant_id` ASC) VISIBLE,
  CONSTRAINT `fk_foodstock_restaurant`
    FOREIGN KEY (`restaurant_id`)
    REFERENCES `restaurant` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
) ENGINE = InnoDB
  DEFAULT CHARACTER SET = utf8mb4
  COLLATE = utf8mb4_0900_ai_ci;
```

Testing and validation of the database:

The database should be tested and validated to ensure thatit is functioning correctly and meets the requirements. This can include running queries to retrieve data and generate reports, testing the system's ability to track food inventory and expiration dates, and ensuring that the system can facilitate the transfer of excess food from restaurants to charity organizations.

4. Usage scenarios

Use cases for the database, including queries and reports:

The food waste management system can be used for a variety of purposes, including:

- Tracking food inventory and expiration dates

- Identifying excess food and connecting restaurants with charity organizations and other users who can make use of the food

- Generating reports on food waste reduction and sustainability

- Facilitating the transfer of excess food from restaurants to charity organizations

Examples of how the database can be used to reduce food waste and promote sustainability:

- A restaurant can use the system to track food inventory and expiration dates, reducing the likelihood of food waste due to spoilage.

- A charity organization can use the system to identify restaurants with excess food and connect with them to receive the food, reducing food waste and providing food to those in need.

- Users can use the system to generate reports on food waste reduction and sustainability, providing insight into how the system is making a positive impact on the environment.

5. Conclusion

Summary of the database design and implementation:

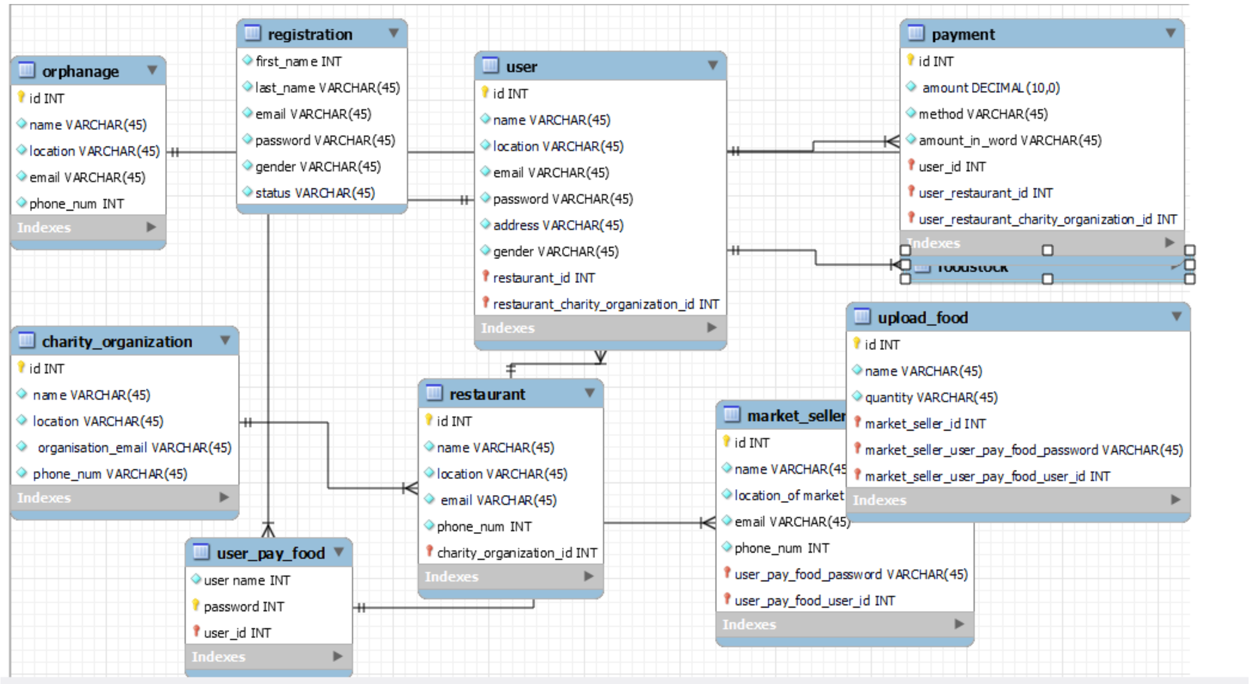
The food waste management system is designed to reduce food waste by allowing restaurants to identify excess food and connect with charity organizations and other users who can make use of the food. The system includes tables for restaurants, charity organizations, users, food stock, and payments, and is implemented using a MySQL database management system and a programming language such as PHP or Python. The tables are linked through relationships that allow for tracking of food inventory and expiration dates, generating reports on food waste reduction and sustainability, and facilitating the transfer of excess food from restaurants to charity organizations.

Overall assessment of the potential impact of the database:

The food waste management system has the potential to make a significant impact on reducing food waste and promoting sustainability by connecting restaurants marketsellers with charity organizations and other users who can make use of excess food. The system's ability to track food inventory and expiration dates, generate reports on food waste reduction and sustainability, and facilitate the transfer of excess food make it a valuable tool in promoting sustainable food practices.

6. Appendices

ERD diagram:



SQL code

```
 CREATE SCHEMA IF NOT EXISTS `food_waste_database` DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_0900_ai_ci;

USE `food_waste_database`;


CREATE TABLE IF NOT EXISTS `charity_organization` (

  `id` INT NOT NULL AUTO_INCREMENT,

  `name` VARCHAR(45) NOT NULL,

  `location` VARCHAR(45) NOT NULL,

  `email` VARCHAR(45) NOT NULL,

  `phone_num` VARCHAR(45) NOT NULL,

  PRIMARY KEY (`id`),

  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE

) ENGINE = InnoDB

  DEFAULT CHARACTER SET = utf8mb4

  COLLATE = utf8mb4_0900_ai_ci;
```

```
CREATE TABLE IF NOT EXISTS `restaurant` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `location` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `phone_num` VARCHAR(45) NOT NULL,
  `charity_organization_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE,
  INDEX `fk_restaurant_charity_organization_idx` (`charity_organization_id` ASC) VISIBLE,
  CONSTRAINT `fk_restaurant_charity_organization`
    FOREIGN KEY (`charity_organization_id`)
    REFERENCES `charity_organization` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
) ENGINE = InnoDB
  DEFAULT CHARACTER SET = utf8mb4
  COLLATE = utf8mb4_0900_ai_ci;

CREATE TABLE IF NOT EXISTS `user` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `location` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `password` VARCHAR(45) NOT NULL,
  `address` VARCHAR(45) NOT NULL,
  `restaurant_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE,
  INDEX `fk_user_restaurant_idx` (`restaurant_id` ASC) VISIBLE,
  CONSTRAINT `fk_user_restaurant`
    FOREIGN KEY(`restaurant_id`)
    REFERENCES `restaurant` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
) ENGINE = InnoDB
  DEFAULT CHARACTER SET = utf8mb4
  COLLATE = utf8mb4_0900_ai_ci;

CREATE TABLE IF NOT EXISTS `foodstock` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `quantity` DECIMAL(10,2) NOT NULL,
  `expiry_date` DATE NOT NULL,
  `restaurant_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE,
  INDEX `fk_foodstock_restaurant_idx` (`restaurant_id` ASC) VISIBLE,
  CONSTRAINT `fk_foodstock_restaurant`
    FOREIGN KEY (`restaurant_id`)
    REFERENCES `restaurant` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

```
) ENGINE = InnoDB
  DEFAULT CHARACTER SET = utf8mb4
  COLLATE = utf8mb4_0900_ai_ci;
```