Software Design Specification: Leftover Food Application, “QuedFood”

Chris Crisostomo, Andrew Benancourt, Kevin Ruvalcaba, Juan Cota

Computer Science 250: Intro to Software Systems

Professor Basil Sajid Shaikh

October 4, 2023

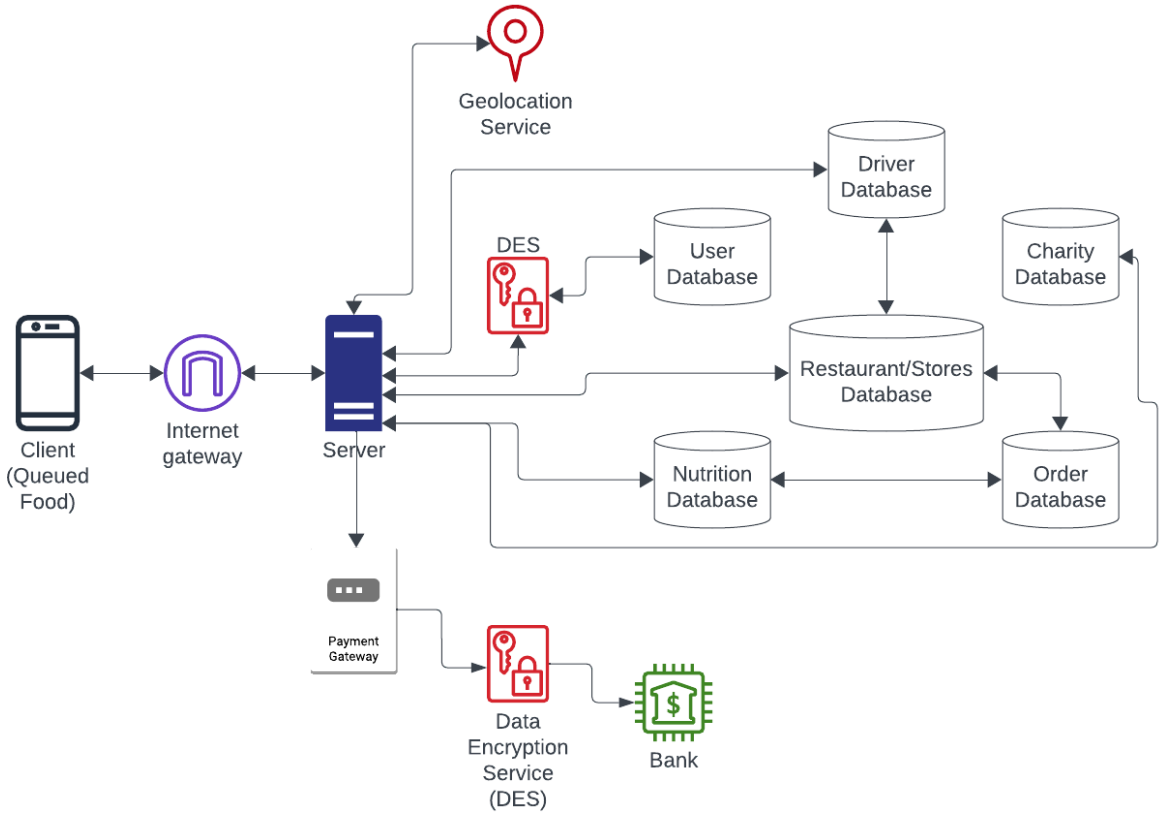
Overview

In today's economy, the average consumer and business seems to be struggling. The supply chains in which businesses buy their products have increased in price exponentially leading many to hike up their prices. With so many price hikes, consumers are finding it difficult to be constantly spending. This can be especially prevalent in the restaurant industry. Restaurants are being forced to up their prices which many customers simply cannot afford to pay for anymore. What ends up happening is that at the end of the day, there is a lot of food left over that is thrown away meaning lost profit. In order to combat this, we are designing an app called, QuedFoods that will help struggling restaurant owners and customers who are looking for cheaper food options.

QuedFoods will be a way for customers to search for local restaurants/stores in their area that join our app and list any leftover food that they have at a discounted price (businesses will be encouraged to do so as they will be able to make some of their money back from left over food and build customer loyalty). From there, customers will select their restaurant and collect their food. Businesses can set the pick up time in order to not disrupt natural flow of customers. Not only will customers be able to go pick up their food but we could also partner up or make our own delivery service. To keep up with our sustainability, we'll make it so that delivery cars have to be electrical or through e-bikes.

In the long term, we want to be able to bring our brand of sustainability and eco-friendliness by teaching our users on different ways to help the environment aside from buying leftover food such as ways to grow their own food or learning how to recycle. We also plan on further developing our own line of merchandise and in order to spread awareness to our app, we could donate portions of merch profit to participating charities.

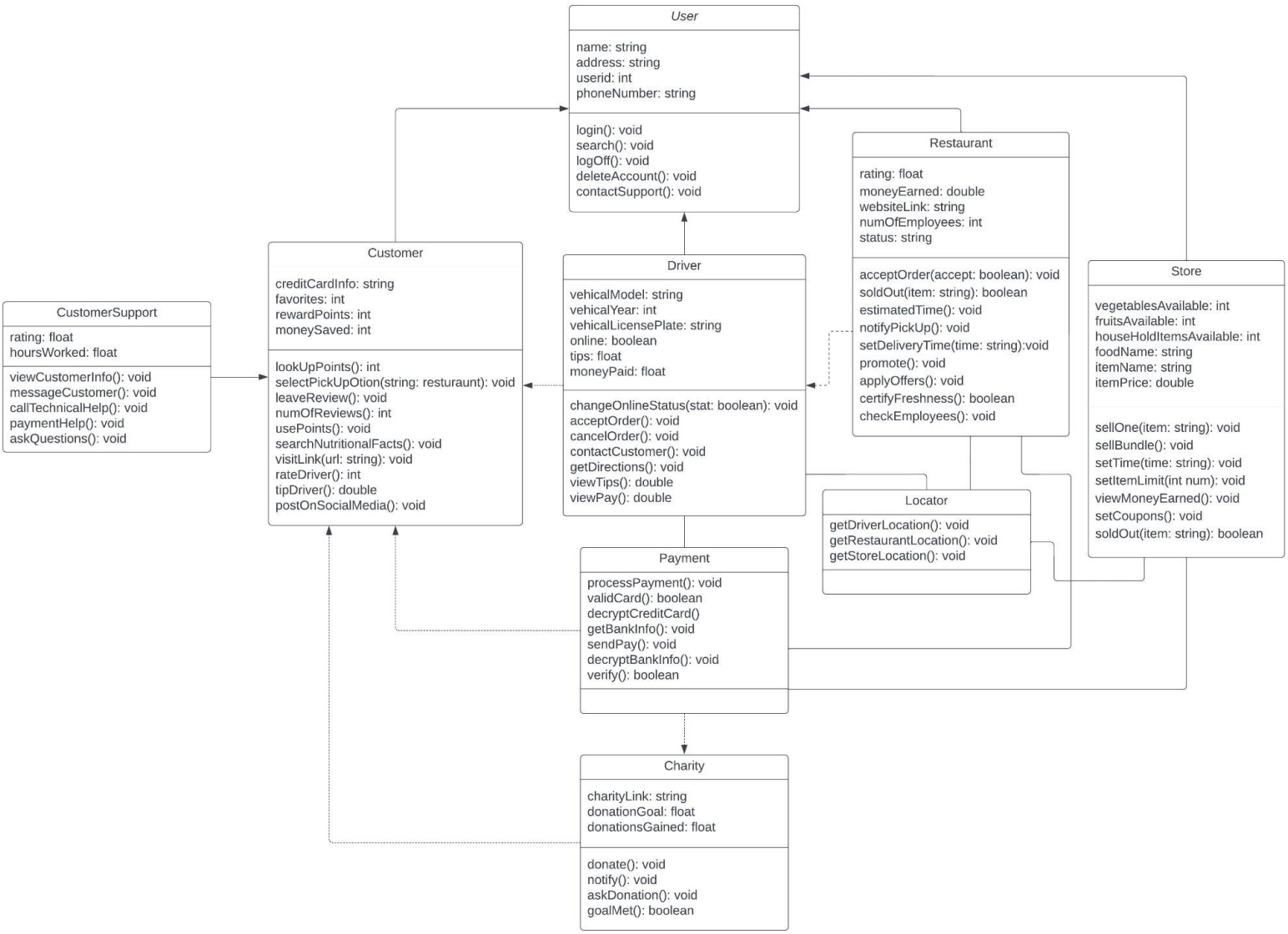
Architecture Diagram



Description:

- Queued Food goes through the internet gateway, connecting to the server
- Through the server, all the databases required for the application are able to be accessed
- Certain Databases are connected to one another to access certain information from each other
- Database of drivers picking up food connects with the restaurants and stores to work together and get food delivered
- User database is isolated by itself and encrypted so that it is not easily accessible

Class Diagram



Classes

User

Abstract class that will contain basic information and abilities that every user will be able to have.

Attributes:

- name (name of the user)
- password (unique password used to login)
- Address (user's address)
- userId (each user will have a unique ID to internally differentiate between users and will be kept in our user database)
- phoneNumber (each user will provide their phone number for communication purposes)

Functions:

- login(userId, password) {verify user's information and signs them into the app}
- search() {allows user to search for restaurants or stores around them}
- logOff() {log a user off the server}
- deleteAccount(userId, password) {delete a user's account and removes them from user database}
- contactSupport() {in case of trouble, this functions connects a user to customer support}

Customer - extends User - ~uses Locator, Payment~

A customer will be the everyday people who will use our program to look for their discounted food or home supplies

Attributes:

- creditCardInfo (the payment information that the customer puts in to pay for their items)
- rewardPoints (the points that the customer accumulates the more they use the program)
- moneySaved (shows the user the amount of money they have saved by using the program)
- favorites (the number of locations that a customer has saved for future purchases)
- numberOfReviews (number of reviews that a customer has left on different locations)

Functions:

- lookUpPoints() {retrieves the number of points that a customer has}
- selectPickUpOption() {when ordering, this function will pop up to make the customer choose if they want to pick up their order or have it delivered}
- cancelOrder() {communicates with the order database and restaurant database to remove the order and refund user's money}
- Pay() {when order is placed, this will use the customer's payment information to confirm their order and place it into the order database and restaurant database, will also call the decryption service for security}
- leaveReview {when a customer's order is complete, they will have the option to leave a review on a business}
- usePoints() {customer can use their accumulate points for discounts on their order or on merchandise store}
- searchNutritionalFacts() {uses the nutrition database to look up the nutritional information for their order}
- visitRestaurantLink() {will direct the customer of the program and into a business's own website, will verify the link is safe}

Restaurant - extends User - ~Uses Payment, Locator~

A user that is dedicated to being a restaurant, will sell only food products.

Attributes:

- rating (rating given by customer's)
- websiteLink (the link used to direct to a restaurant's official website)
- moneyEarned (the money a business has earned through the program)
- numberOfEmployees (amount of workers in a restaurant)
- status (let's customer know if the restaurant is family owned, small restaurant, chain restaurant, or higher end restaurant)

Functions:

- acceptOrder() {receives a customer's order and adds it to their order list if they are able to fulfill it}
- soldOut() {if the item that the restaurant offers is sold out, this will make it so users can not order that item}
- estimatedTime() {calculates how long a customer will have to wait for their order to be ready}
- notifyPickUp() {sends a customer a message that their order is ready}

- setDeliveryTime() {in order to accommodate restaurant's with high traffic, this function let's restaurants set what time will customers be able to pick up their order}
- applyOffers() {a restaurant can set offers on their available items with this function}
- certifyFreshness() {this will be a set of questions that a restaurant has to answer and pass to be able to sell their food, will let them inside the restaurant database}
- checkEmployees() {depending on the number of employees, the status of the restaurant will be set in order to give customers different options}

Driver - extends User - ~uses Locator, Payment~

A driver is a special user that works for our company as a certified delivery driver for the program.

Attributes:

- carModel (the model that the driver will be using and saved into our driver database)
- bikeModel (the type of bike a driver is using if they decided to use a bike, saved into drive database)
- LicensePlate (will be used to keep track of driver's identities and will be placed in driver database)
- moneyPaid (amount of money a driver has been paid out by us)
- tips (the amount of money made in tips by driver)
- rating (rating given to a driver)

Functions:

- acceptOrder() {when a order pops up, the driver can select it and will be able to pick up and deliver an order)}
- notifyCustomer() {will be used to communicate with a customer the status of their order}
- registerCar() {registers a driver's car to be placed into the database}
- registerBike() {registers a driver's bike into the database}
- viewTips() {access the amount of tip money earned}
- viewPay {access the amount of money paid out by program}

Store - extends User - ~uses Locator~

A store is a business that sells more than just food items, can sell food, household items, or other approved items.

Attributes:

- vegetablesAvailable (number of vegetables the store has to offer)
- fruitAvailable (number of fruits available)
- householdItemsAvailable (household items available)
- otherItemsAvailable (other type of items that are sold)
- ItemName (name of the item being sold)

Functions:

- sellOne() {sell one of an item, could be any}
- sellBundle() {stores, can hold bundles of different items to be bought by consumers}
- setTime() {can set a time in which a customer can pick up their items}
- setItemLimit {can set a threshold on the number of items a customer can buy}
- viewMoneyEarned {check the amount of money made through app}
- setCoupons {can apply coupons to their items}
- soldOut {lists an item as sold out and will not be able to be bought}

Payment

This class will store the payment information of users and store it safely, will also disburse payments to drivers and businesses. This class has no attributes of its own but rather is used to store and secure different payment information.

Functions:

- getCreditCardInfo() {saves a user's payment information}
- decryptCreditCard() {decrypts user's payment information}
- getBankInfor() {used to save all employees' and businesses' banking information}
- SendPay() {sends out payment money}
- decryptBankInfo {sends over the bank information to decryption and security service}
- verify() {verifies payment information}

Charity - extends User - ~uses Payment~

Our app will work with charities to help boost our reputation of sustainability and helping others. Charities or local food banks we work with will involve those that involve helping the climate or feeding the needy.

Attributes:

- CharityLink (the official link to the charity's website)
- donationGoal (goal needed to be met)
- donationsGained (money earned by customer's donations)

Functions:

- notifyUsers() {notify users if they would like to make a donation}
- goalMet {will send out a notification if the donation goal has been met}
- displayDonationGoal {displays the donation goal}
-

Locator

This class uses the geolocation services to allow users to search for different restaurants and stores. Also allows the tracking of drivers to see where they are at.

Functions:

- getDriverLocation() {shows the current location of a driver}
- getRestaurantLocation() {shows the location of a restaurant}
- getStoreLocation() {shows the location of a store}

CustomerSupport - extends User -

Our customer support service that helps out any user with technical or payment issues.

Attributes:

- Rating (rating given to the specific customer support employee)
- usersHelped (number of users helped out)
- hoursWorked (number of hours worked per week)

Functions:

- callTechnicalHelp (will be able to provide information to provide help in technical problems)
- paymentHelp (provide help for any payment information)
- askQuestions (ask questions to the user to see what type of help they need)
- viewCustomerInfo (will have access to the information of the customer)
- messageCustomer (sends out messages to customer)

Gantt Chart

ID	Name	Start Date	End Date	Duration	Progress %	Resources
1	Title Page	Sep 20, 2023	Sep 21, 2023	2 days	100	Chris Crisostomo
2	Software Title	Sep 20, 2023	Sep 21, 2023	2 days	100	Chris Crisostomo
3	Team members	Sep 20, 2023	Sep 20, 2023	1 day	100	Chris Crisostomo
4	System Description	Sep 22, 2023	Sep 28, 2023	5 days	100	Kevin Ruvalcaba