

Design Studio 2 - part 2

Discussion 9am Team #5

EXECUTIVE SUMMARY

Although we live in a predominantly affluent nation, the United States of America, there are many who have fallen on hard times. In fact, any of us, at any time, may find ourselves in need of food. Our team seeks to help those in need of food supplies in a specific way. We are proposing the design of a software application that will connect food-dispensing outlets such as grocery stores, restaurants, farmers' markets, and other food outlets, to those in need.

The over abundance of food that will soon be expiring can be transferred to food-transporters, and eventually make it to those in need of sustenance. Food outlets, such as the ones listed above, often have food items that are required by policy or state-law to be removed from shelves due to upcoming expiration dates. These food items are still edible and safe to consume if they make it into the hands of those in need on time. This application was chosen amongst other proposed solutions because of its likelihood of success. Our team concluded that there is a market for this application on the grounds that all users of this software are incentivized to participate. The food-dispensing outlets will benefit through the advertisement of their goodwill, tax write-offs, and not having to dispose of edible food products. Because of this benefit to the food outlets, they would be willing to pay a relatively low price to incentivize drivers to deliver food products to those deemed in need via a short consumer screening. So in our design, we envision three groups participating in the application: Food outlets/donors, Drivers, and the Needy.

01 - AUDIENCE AND OTHER STAKEHOLDERS

Below is the description of corporations and individuals who are involved in the development of NoLeftOver and interested in joining our food reuse program. These actors will interact with NoLeftOver on different roles and responsibilities as well as their influences in the operation of the project will also be listed.

Concerning our potential users, we envision them as:

➤ Grocery stores/outlets and food suppliers

- Due to their role in providing excess and edible food from overproduction, close to expired date, or food that does not meet packaging specifications for those who are in need. Instead of throwing away these excess food, grocery stores and food outlets can donate them to NoLeftOver and we will be in charge of distributing and delivering them to hungry people or food banks and food secure organizations. Donating food will be beneficial for the brand's reputation as people will know them through their goodwill endeavors.

➤ Restaurants/hotels/farmer's/schools/cafeterias

- Due to their need in contributing excess and unused food coming from different sources including overbuying, overproduction, uncertain market demand, inaccurate serving size and food portion control, etc. to people with lack of food.

➤ Food insecurity

- Due to their need in receiving food from donors and food providers. People with difficulty in retrieving food and experiencing hunger will use the app to access and consume those excess food provided/donated from individual donors and food organizations.

➤ Drivers

- Due to their role in being the means of transporting food from donors to recipients or food banks/organizations.

➤ Volunteers

- Due to their need to join our food rescue program to save wasted food from being thrown away to landfills that would contribute to the contamination of the environment and climate change. Volunteers will be in charge of holding food rescue programs on the app to propagate our message of waste reduction and raise people's awareness of the impacts of food waste.

➤ Food donors

- Due to their need in donating and distributing excess food that are still edible and unexpired food to the hands of those who need them the most. Food donors can

either be individuals or any organizations who wish to make their contribution to the development of our program.

And concerning our potential stakeholders, we have ascertained that they will be comprised of the following:

➤ **Government**

- Due to their need in aiding hunger and vulnerable families and achieving food security goals. Government agencies are one of the major forces in combating hunger in the country with a variety of feeding programs, fundings and nutrition assistance. Food secure program from NoLeftOver will provide a great contribution to end food insecurity with the support and assistance of the government.

➤ **Local community**

- NoLeftOver can play a crucial role in the well being of the local communities. With goals of distributing otherwise unwanted food items to people in need can benefit communities overall in decreasing food insecurity. As less and less people are experiencing food insecurity, the quality of life and poverty rates will become better as well. Furthermore, the entire community can join in on this upward initiative for bettering lives. Together people can make their communities a better place.

02 - GOALS, CONSTRAINTS, ASSUMPTIONS FOR THE OVERALL DESIGN SOLUTION

Our primary goal is to find a solution to the essence of the problem at hand. Every year, roughly one third of the food produced for human consumption goes to waste. In this case we are aiming to connect packaged foods near expiration or excess produce to serve those who are food insecure by hiring drivers and volunteers with incentives. Of course there will be limitations to what resources we can make use of as a team, and there will be assumptions made along the way that we need to be made conscious of.

➤ **Goals**

- **Prevent food waste and address the global food issue:** Growing food requires water, seeds, labor, machinery, energy, and fertilizer. Letting food go to waste is a frivolous use of natural resources that drives up costs, inflates food prices, and weakens the food supply chain. In addition to taking up space, decomposing food releases methane, a powerful greenhouse gas. Our goal is to ensure excess wholesome food that was otherwise being tossed away, helps feed people, not landfills. Making an effort towards fighting the global food crisis and climate change.
- **Fight food insecurity and improve nutrition in food-insecure communities:** Our goal is to address barriers to accessing healthy food for food-insecure communities. People who are food insecure may be at an increased risk for a variety of negative health outcomes and health disparities. By providing assistance for people in

poverty to access healthy food, we aim to improve nutrition in food-insecure communities.

- **Embrace “imperfect” produce:** A significant portion of produce that goes to waste are ugly fruits and vegetables. Sometimes the dents and scars or weird shapes and sizes have given them cosmetic challenges which kept them out of retail stores. Sometimes, there might be packing mistakes that keep them off of the store shelves. But they're perfectly edible, delicious and just as nutritious. Our goal is to provide an outlet which in a way rescues these ugly produce and gives them a better purpose.
- **Inspire thoughtful consumption of food:** NoLeftOver will also drive the initiative of thoughtful consumption of food. With educational information available in our application, we aim to transform mindless consumption into conscious decisions. Our goal is to encourage consumers and manufacturers to make sustainable consuming choices.
- **Offer people a chance to contribute to a good cause:** NoLeftOver will facilitate surplus food donations and give back to the community. Besides large food outlets, we encourage everyone who wants to make a difference to donate groceries to people in need. There's also other ways to put in effort like becoming a volunteer to make a change.

➤ Constraints

- **Users must provide a login to access NoLeftOver:** NoLeftOver is an account-based application. Hence, users will need to create an account and sign in before accessing the app. As users might need to provide their personal information while using the app, a login is required for private data security and authentication purposes.
- **NoLeftOver must be accessible via any public network connection:** in order to provide users with convenient access and accommodation while using the app to donate/receive food or join volunteering activities, NoLeftOver must be accessible through any public network connection or private cellular network.
- **NoLeftOver must be operated on different web browsers and mobile apps:** The app should be easy to navigate through the Internet and accessible from a variety of popular web browsers and platforms. It should be built to use on different types of technological devices with varied operating systems to provide portability and convenience for users.
- **NoLeftOver must have a built-in interactive map:** An interactive map is required for navigation purposes. Drivers will pick up excess food from donors and grocery

stores/outlets and transport them to households and individuals who are in need. Hence, an interactive map must be developed to represent users' locations and delivery progress.

- **To access NoLeftOver, users must have Internet access or WiFi connection:** Users must connect to the Internet or WiFi to interact with active features of NoLeftOver as well as getting constant updates of our programs, the availability of food and information about the delivery/pickup progress.
- **NoLeftOver shall enable Live Chat feature:** in order to assist users with fast response and solution as they might face issues during donation, delivery or receivment process. Live Chat helps our support team monitor situations that users are facing and provide them with constant assistance and appropriate approaches/options.
- **The UI of NoLeftOver should be user-friendly:** The user interface of the app should be designed simple and easy to use. Users should be able to use the app to achieve their purposes with sufficient information and instructions from simple features and functions.
- **Different UI versions of the application will be provided to different groups of users:** to serve different purposes of different groups of users, NoLeftOver may provide distinguishing features and operations from the app's user interface. When a user logs into NoLeftOver, depending on which role (food donors, food receivers or delivery men) that user is, the UI of the app will show information and instructions about the related activities with appropriate interface.

➤ **Assumptions**

- **Food donors and receivers are comfortable with sharing their locations:** Our ultimate goal is to prevent food waste while feeding those who are in need. Hence, we shall ask users to share their locations in order to get food from food donors and deliver them to recipients.
- **Donated foods are edible, unexpired and qualified by the food safety laws and regulations:** Before making any donation, food donors shall be provided with a food guideline to ensure that the donated foods are safe and well-preserved. The donated food should meet the safety requirements as stated in the guideline since we are opposed to any harmful acts to others either intentionally or unintentionally.
- **Food recipients are aware of the inconsistency and unavailability of food resources:** Since we're operating a non-profit and charitable program that helps prevent food waste and saves people who are in need, there might be an

inconsistency in the food amounts and resources supplied to us. Due to the desirability and variation of recipients in the needs of food and ingredient items, the availability of food might be affected.

- **All users provide a stable access to the app to get notified about the donation and delivery progress:** It's essential that all users have at least one stable access to technology since the operation of NoLeftOver requires public network connection. The availability of food resources and information about the donate, receive or pick-up processes will constantly be updated on the app. Hence, a constant and stable access to the app will prevent users from missing our notifications.
- **When users face any issues, they would contact support service through provided contact information:** issues and concerns might arise during donation, delivery or food receipt process. Thus, we enable a support service for users to provide them with the best solutions and approaches to walk them through their difficulties. We also expect users to contact our support team to provide them detailed information about the situation they're dealing with. That would help us address the issues and provide constant assistance to users.

03 - MAIN DESIGN

➤ Application Design:

- **Create User accounts:** Allow users who identify themselves as food insecure and can provide supporting documents to sign up for NoLeftOver's receiving list. Individuals who would like to volunteer or donate food items can sign up as a donor. Drivers who would want to earn incentives by distributing food items to food recipients can sign up to become Drivers.
- **Alert system:** Arranges efficient communication between food donors and charities and fast delivery of excess food. Delivery drivers, caterers and anyone working with large volumes of edible but rejected food create alerts in the app. Food pantries, processors and composters immediately receive these alerts and contact the source for delivery arrangements.
- **Communication system:** To ensure smooth and efficient transactions. Drivers, consumers, and food providers will be able to keep in touch with one another. If any issues arise users will be able to directly contact the responsible person and solve any issues as soon as possible to prevent delays or mistakes.
- **View/select available food items (consumers):** Food consumers will be given a list of available food items provided by food outlets/donors and their associated information. The consumers will be able to select the items they would want to

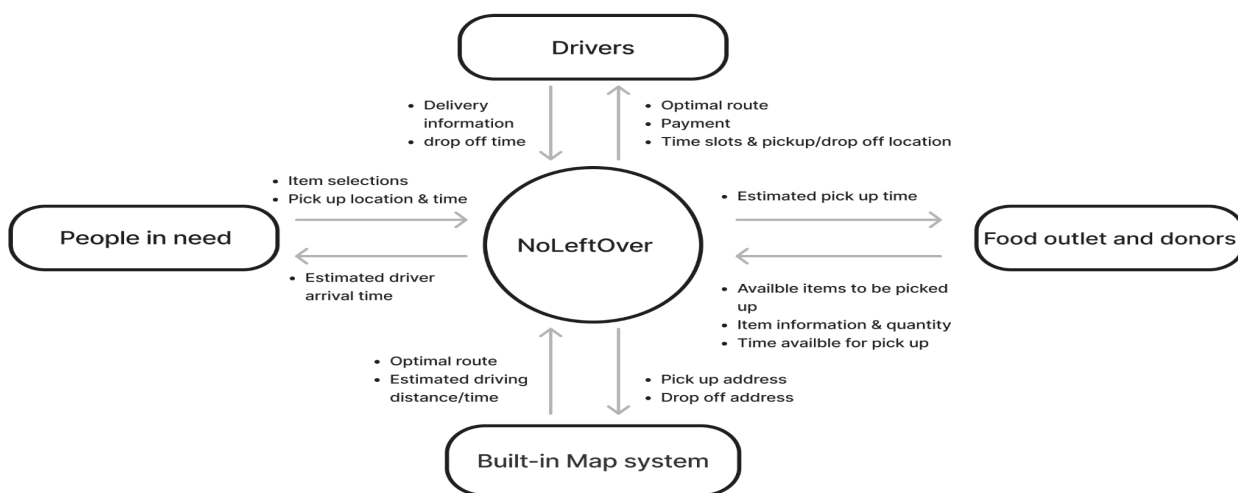
receive. The quantity of food items they are allowed to choose will be calculated considering the family size and item value. For example, each food item will be labeled with its appropriate market price, and consumers will be able to select items until the total price meets their predetermined limit.

- **Select from the list of available delivery time slots (consumers):** To ensure the freshness of products and better suit consumer's needs, a feature provided by NoLeftOver is allowing consumers to pre-select a delivery time-slot from the list of available times. Therefore, consumers can pick the times that they will be available at home and the information will be passed on to potential drivers to make the delivery.
- **List food items/quantity (Food outlets/donors):** Food outlets, retailers and donors can input the items ready to be distributed. Alongside the number of each item, the form will require the item's expiration dates, nutritional information and market form. There will also be an available comment section where item providers can explain the reason the item is for distribution. For example: excess inventory, discoloration/scarring, packaging error, near expiration date, etc. They give item consumers a better understanding as they make decisions.
- **Provide pick-up times and information (Food outlets/donors):** To better suit food outlets', retailer's and donor's availability, the system will ask and provide the time slot where drivers will come for the pickups. Food outlets, retailers and donors will also have the ability to set up time periods where they are available to accept pick-ups. This way, they won't be bothered or interrupted unexpectedly outside of the time slots they chose.
- **Assign driver pick-up/drop-off tasks:** The system would offer close-by drivers pick-up/drop-off tasks near the driver's area. To best save fuel and resources, the system will assign pick-up for multiple orders at once and the driver can go around and drop each off at the drop off locations.
- **Provide the optimal route (Drivers):** To best save time and resources, the system would organize pick -up/drop-off locations and provide the best route for drivers to take and will use the route to provide an estimated time for other users. By reducing delivery time, this can benefit all users in terms of saving time and getting item's faster and fresher.
- **Give payment estimate of each delivery:** To attract more drivers, each order will come with incentives calculated based on the total time it takes to deliver. To ensure fairness, the calculation will not be based on travel time alone. Time for picking-up and dropping-off will also be considered and included in the payment. Furthermore,

each completed order will be seen as a bonus, thus making drivers want to pick-up more orders at once and deliver them one by one.

- **Inform the time frame (Drivers):** To avoid interrupting food providers unexpectedly or making deliveries while people are not at home, drivers will need to be well informed about the times when the pick-up/drop-off will be expected. If it's near the closing time frame, an order may need to be canceled or further communications will need to be made.
- **Order cancellation and rescheduling:** Due to unforeseen circumstances, an order placed and be canceled or rescheduled. For example, if the driver was delayed and could not pick up or drop off an order in time, after communication both parties can request cancellation with valid reasonings.

Application Design Model



➤ Interaction Design

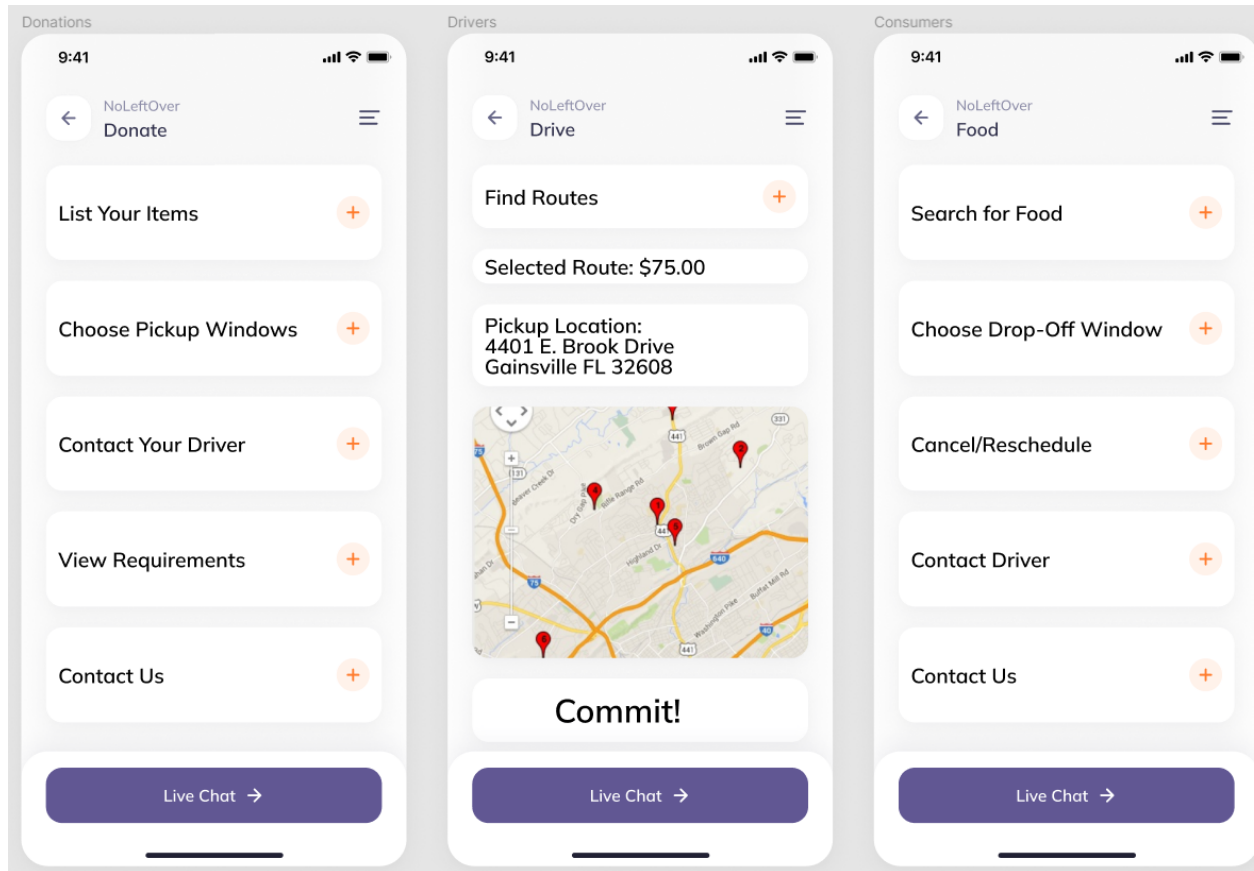
Three user interfaces will be produced, each providing a user experience that will serve the needs of the three major user-groups. These are the food-outlets/donors, drivers, and the recipients of food (consumers). Logins will correspond to the appropriate interface for each. The UI will comprise various screen layouts that can be adjusted to accommodate different devices and/or varying scenarios. Screen layout options will vary according to the type of device used. Touch screen enabled devices will allow for well-established gestures to navigate the GUI. The intention here is to provide familiar ways of interfacing with a GUI in order that the learning curve will be diminished. The application for all three

user-group interfaces will provide easy access to policy information as well as channels for help such as chat or phone calls.

- An initial login/sign-up dashboard will be displayed whereby users will choose at registration which group they belong to if not currently registered. After selection, the dialogue will guide users to the necessary information required of them to participate. Each user interface will be custom tailored to the needs of each of the three user-groups. After initial registration, subsequent login credentials will immediately bring up the appropriate user-interface.
- Food outlets/donors UI will:
 - allow for the listing of available items along with available quantities.
 - allow for the listing of food availability days & times.
 - provide a place to enter expiration dates.
 - be able to contact a driver en route.
 - be able to review policy requirements.
 - be provided with a link to phone and chat access.
 - will have access to food policy requirements.
 - be provided with a searchable history of their transactions.
 - have the opportunity to advertise through the application.
- Drivers UI will:
 - be provided with means to add direct-deposit or PayPal funding.
 - allow for drivers to search for immediately and future available routes.
 - contain a plot of locations on a route.
 - display the payment amount for a particular route.
 - inform the driver of the time frame in which the route must be completed.
 - allow drivers to commit to a route through the application.
 - be provided a final 'Commit' dialogue.
 - be provided with a link to phone and chat access.
 - will have access to food delivery requirements.
 - have access to completed route history and amounts paid.
 - have access to the amount currently owed them.
 - be able to take photos of their deliveries through the application.
- Consumers UI will:
 - see available food items in their area.
 - be able to select their items of interest.
 - be provided with a window of times when the items can be delivered.
 - have the ability to select a window of time for delivery
 - have the ability to cancel an order or reschedule a delivery time.
 - be able to contact a driver en route.

- be provided with a link to phone and chat access.
- be able to see the expected delivery time.
- have the ability to see their order history.

Interaction Design Interfaces

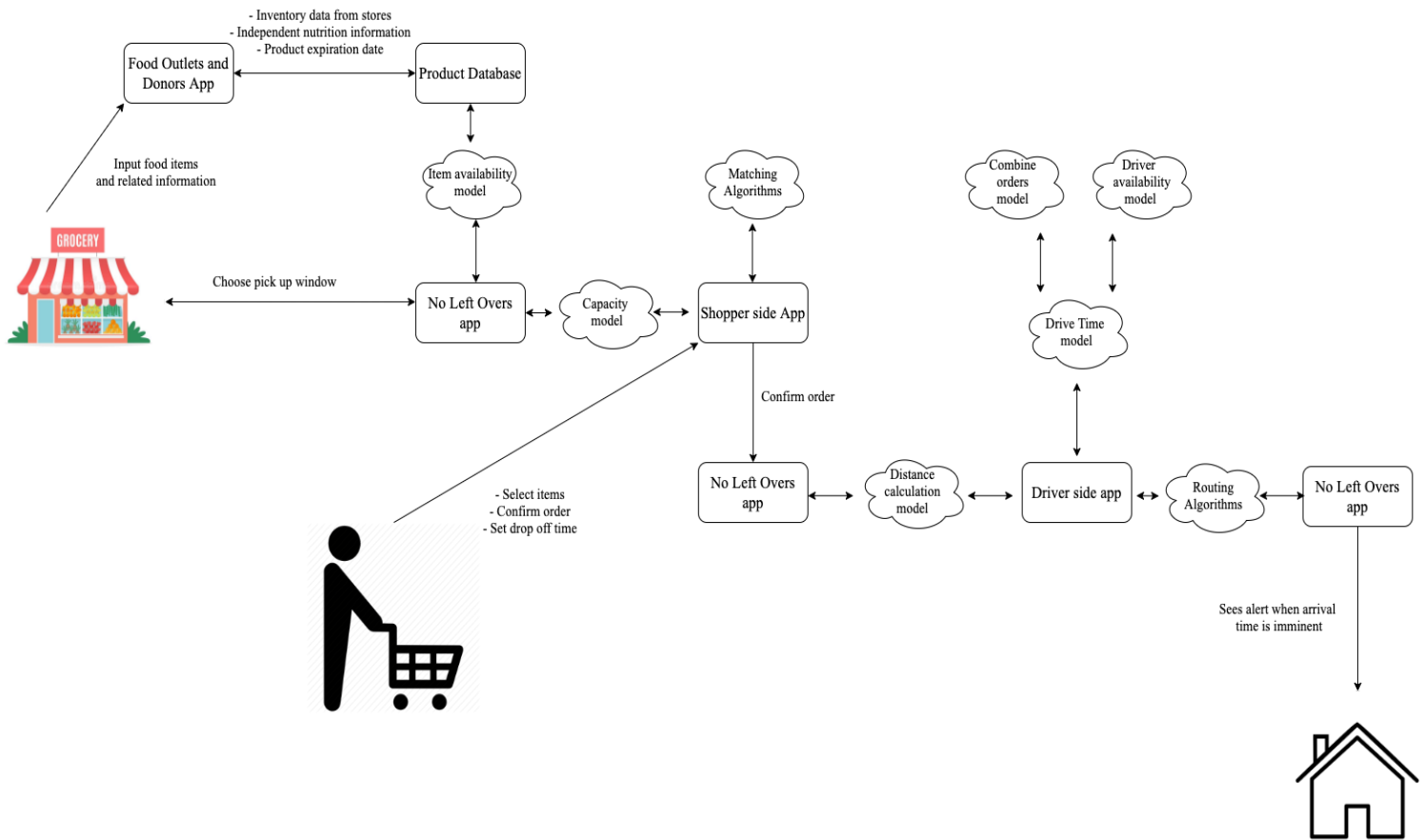


➤ Architecture Design

The architecture design serves as the basic backbone of a system. Maintaining the system's durability, consistency, and maintainability is critical in order to provide the greatest experience to the users. As a result, understanding the system needs, restrictions, and bottlenecks should be comprehensive to form the direction among developers and stakeholders.

Our decisions: To effectively communicate the architecture of No Left Overs, we have decided to incorporate a Data Flow diagram to better understand the flow of information, a context diagram to define and clarify the boundaries of our software system, an entity relationships diagram to convey major entities interacting within the system and their relationships with each other. Furthermore, we're using a 3-tier model to organize our application into three logical and physical computing tiers: the presentation tier, or user interface; the application tier, where data is processed; and the data tier, where the data associated with the application is stored and managed.

High Level Data Flow diagram No Left Overs



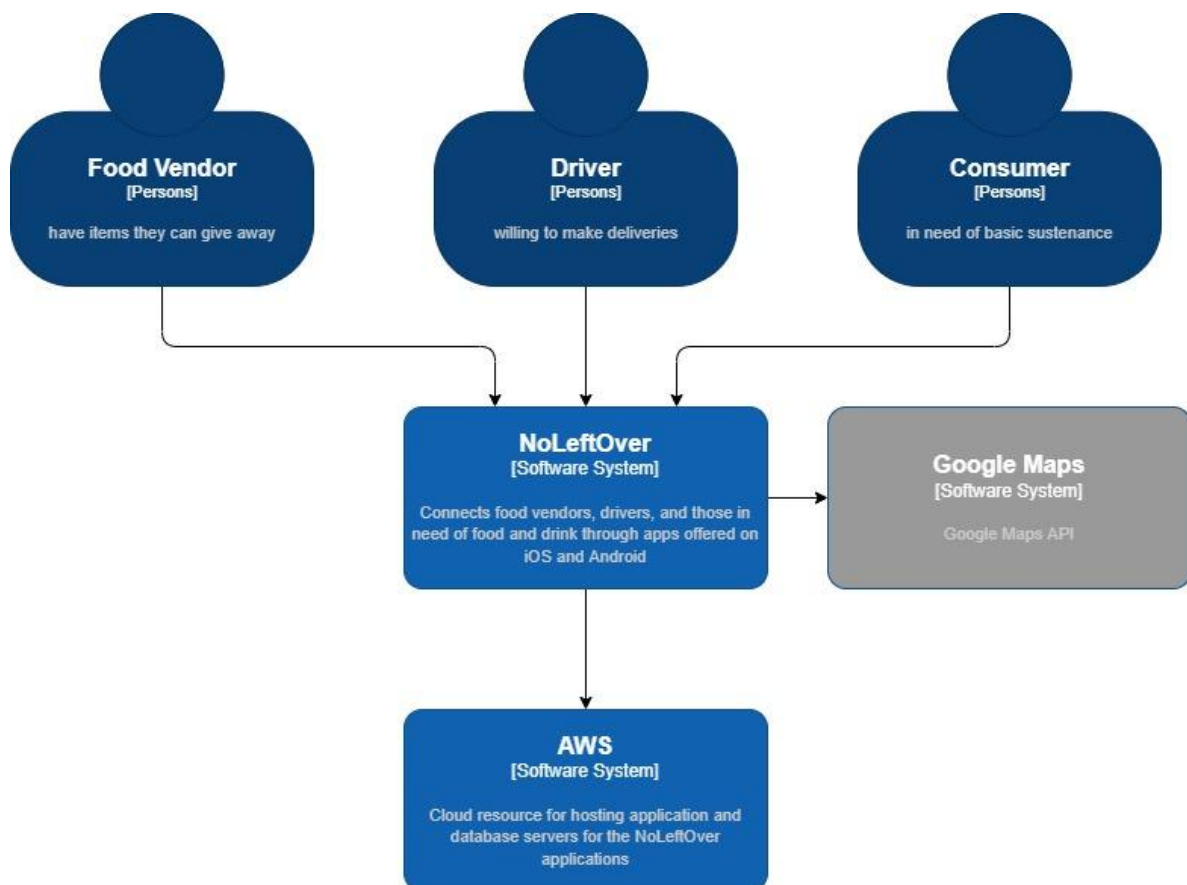
The above data flow diagram maps out the flow of information for NoLeftOvers's system. It uses defined symbols like rectangles, clouds and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination of our application. This diagram models our system's high-level detail by showing how input data is transformed to output results through a sequence of functional transformations.

Data flow steps explained:

1. Grocery outlets and food item donors use the company's app to list donation groceries and input related information. Meanwhile, they will choose their available time periods for pick up which will be handled by NoLeftOvers application.
2. The information about inventory and pricing data from the donors, item expiration and nutrition information, and other product details will be stored in our product database.
3. The information in the product database will enter through our item availability model and be made available on NoLeftOver's app.

4. Behind the scenes, databases, machine learning models, and a variety of algorithms interact with the shopper's side app and help them fine-tune their order making sure they're needs are satisfied.
5. Once consumers have selected their items, confirmed their order, and selected drop-off time, NoLeftOver app will organize the provided request and use distance availability model to calculate the optimal driving distance.
6. Each order's request will be grouped and through driver side application, where a drive time model will help group orders near the same neighborhood and find and inform nearby drivers. Meanwhile a reward estimation will be calculated based on distance and time for drivers to view.
7. After orders have been handed to drivers, drivers will be guided with the optimal pick-up and drop off time and be given an expected route with the help of driver availability model, combined orders model and algorithms.
8. Finally, the system will project an estimated arrival time, alert the donors and consumers that the driver's arrival is imminent, and prompt necessary communication if needed throughout the delivery process.

NoLeftOver System Context Diagram



➤ Users

○ Food Vendors

- are outlets that dispense and/or sell food and beverage items to the public, usually serving a local community.
- are willing to participate in NoLeftOver because they see the benefit of the tax write-offs they can enjoy, as well as the fulfillment of their desire to help the local community.
- have the means to pay drivers to deliver their food items to people in need locally.
- can benefit from advertising on the NoLeftOver platform.
- can benefit socially from their charitable contributions.
- will interact with the NoLeftOver either through iOS or Android mobile applications.
- will interact with Google Maps indirectly through the NoLeftOver application.

○ Drivers

- can be volunteers or paid drivers that desire to serve the local community by making food/beverage deliveries to the area.
- will interact with the NoLeftOver either through iOS or Android mobile applications.
- will interact with Google Maps indirectly through the NoLeftOver application.

○ Consumers

- are those in such financial need that they could benefit from charitable donations from local food vendors.
- will interact with the NoLeftOver either through iOS or Android mobile applications.
- will interact with Google Maps indirectly through the NoLeftOver application.

➤ Software Systems

○ NoLeftOver

- bridges the gap between food vendors, drivers, and food consumers for the purpose of serving the needy.

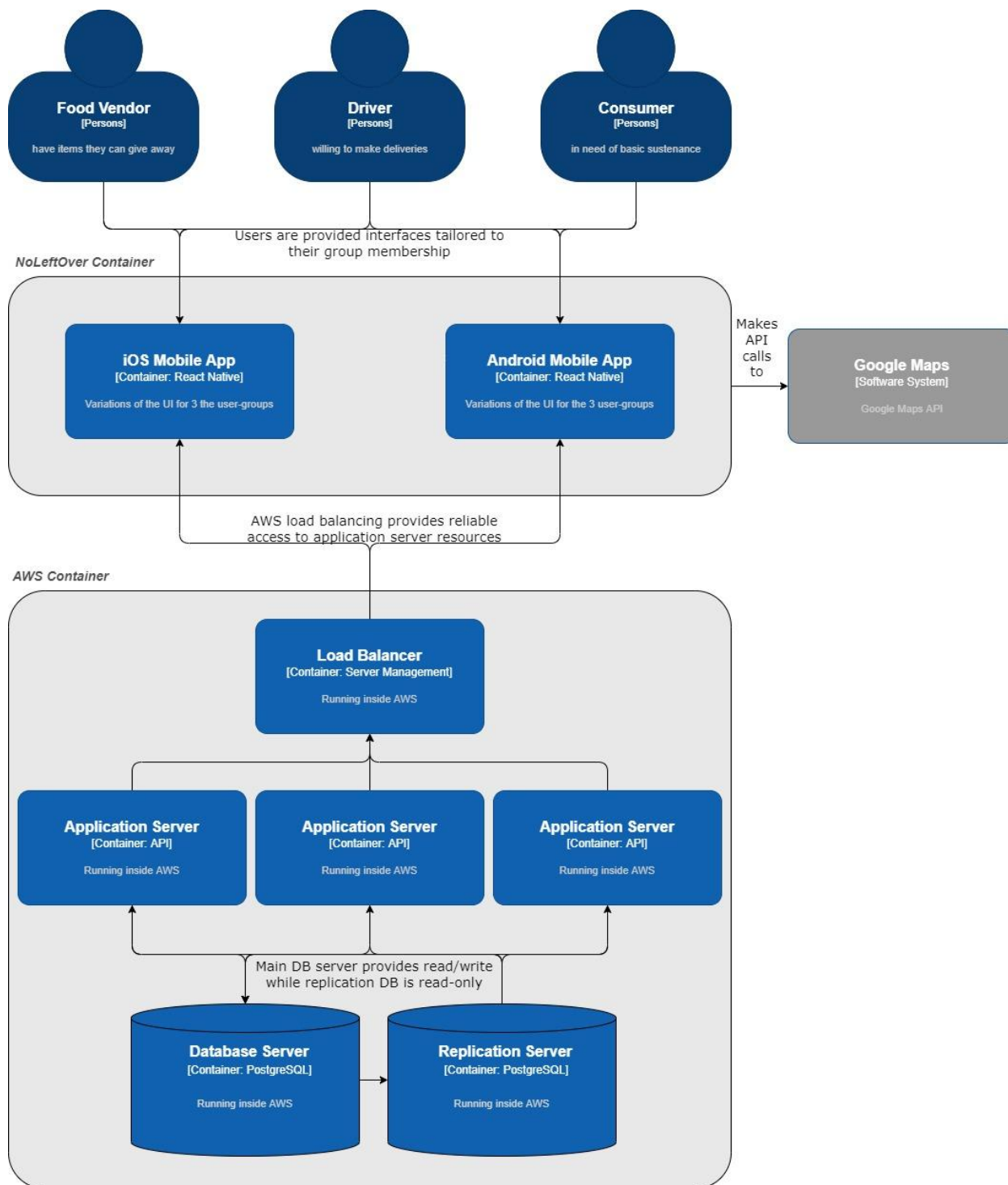
○ AWS

- is being utilized in order to facilitate the cloud-server overhead needed to support the NoLeftOver application.

○ Google Maps

- has an API that will be made use of for interaction of the three user-groups directly from the iOS and Android mobile applications.

NoLeftOver System Container Diagram



➤ Containers

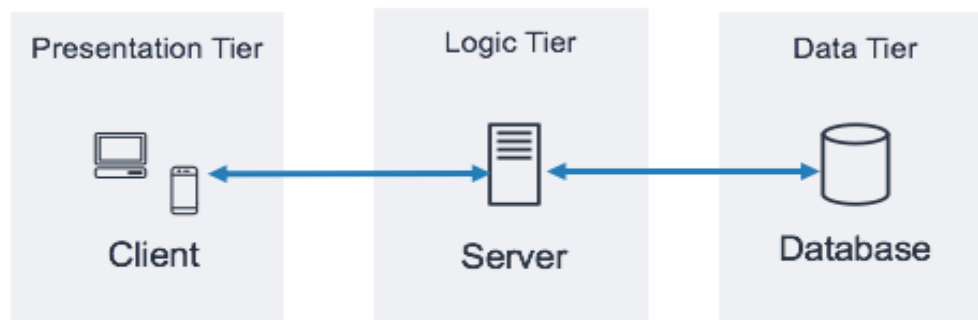
○ NoLeftOver

■ iOS Mobile Application

- will be developed with React Native.
- will be backward compatible for at least 2 generations.

- Android Mobile Application
 - will be developed with React Native.
 - will be backward compatible for at least 2 generations.
- Google Maps API will be accessed for mobile applications.
- AWS
 - Database Server
 - PostgreSQL's main database will provide read and write access as it serves as the main data store.
 - Replication Database Server
 - PostgreSQL's relational database will provide read-only access in order to equalize query demands (i.e. resources) between it and the main database server from the application servers.
 - Application Servers
 - Ubuntu Server 22.04 LTS
 - 3 servers will be utilized at first and others will be added as needed.
 - Load Balancer
 - will make use of the AWS Network Load Balancer.

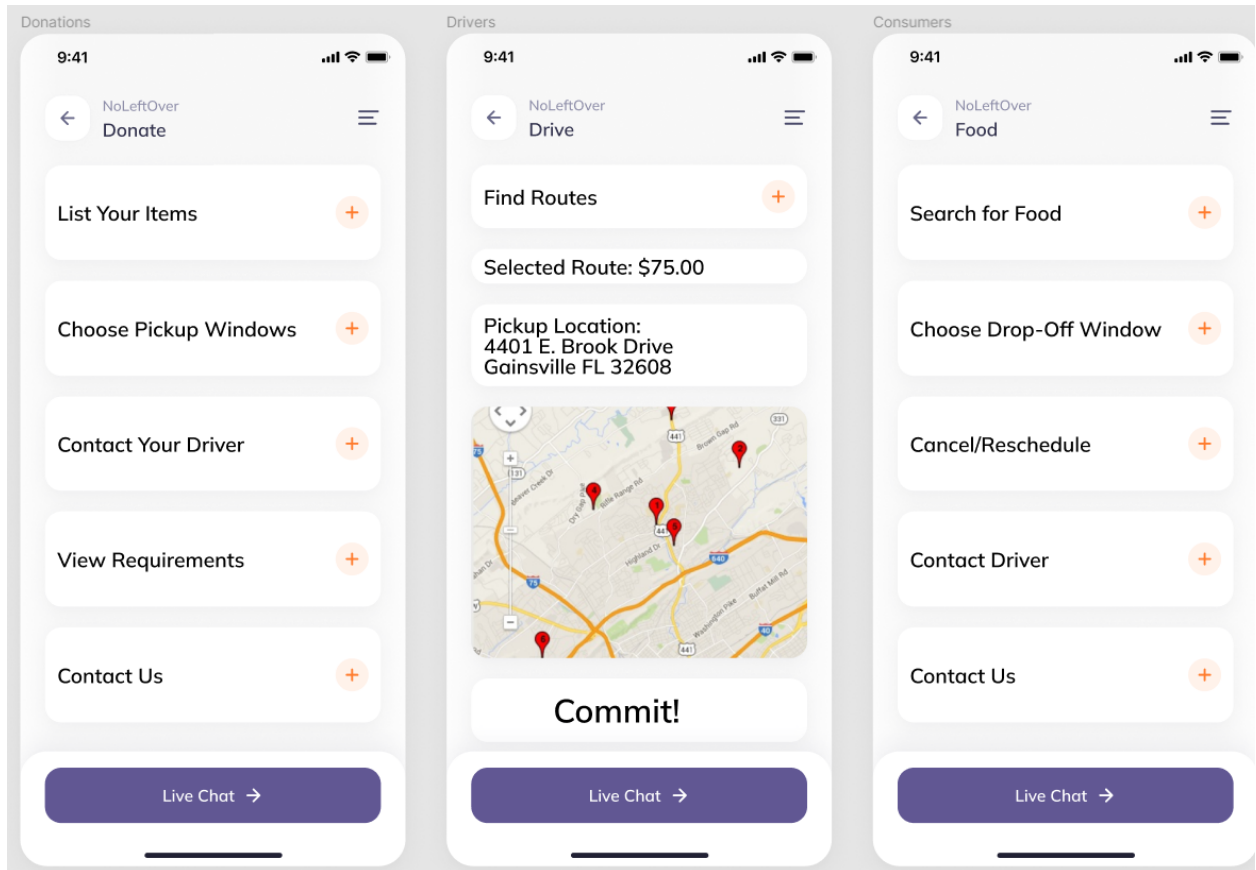
Our 3-Tier Model



We have decided for our program to be built in three layers: the presentation tier, the logic tier, and the data tier. Each layer executes a set of well-defined activities in order to achieve the software's application design goals. The user interface, which might be a web application or a mobile app, is the presentation tier. When a request is submitted from the client, all data is processed and returned to the client, where it is displayed; this process is known as the logic tier. Three layers can connect with one another by sending requests from the client to the server and getting information from the data tier and returning them to the client via API calls.

PRESENTATION TIER

The software application will have three different user interfaces designed specifically for each user group: Donations, Drivers, and Consumers. All three interfaces will have the feature of Contact Us and Live Chat.



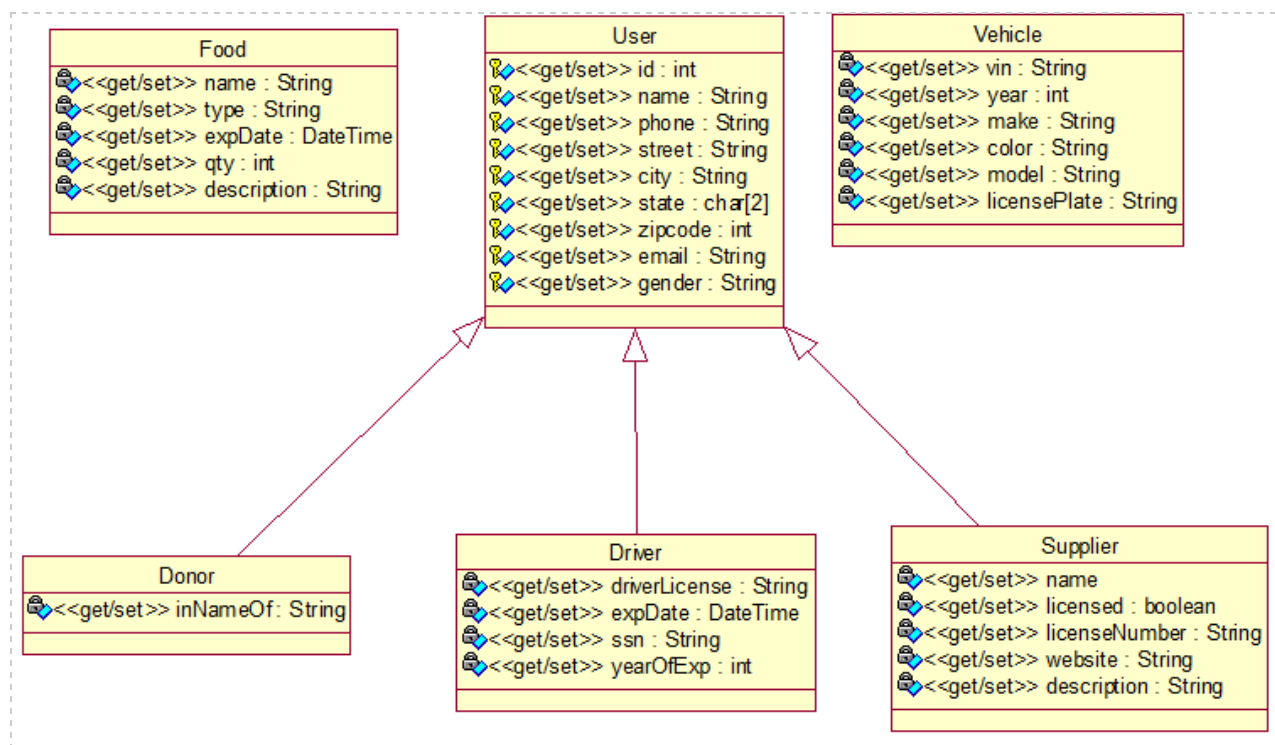
- **NoLeftOver - Donate:** This interface is designed for food donors, and will display upon logging into a donor's account. The unique features that food donors have are: list items, choose to pick up windows, contact drivers, and view requirements. These features are designed specifically for food donors to help them list and donate their food items, have their donations picked up by a set time window by delivery drivers, and finally understand the requirements that are needed in order for a food donation to meet the minimum listing requirements on the NoLeftOver app.
 - **List Your Items:** This feature allows food donors to list their food items on the NoLeftOver app. They must fill out the required information for their food items to be successfully listed on the app. Pictures and detailed descriptions are also required for listing food items.
 - **Choose Pickup Windows:** Food donors have the ability to set a pickup window for their food donations. A driver will get automatically assigned for the pick-up based on the driver's location and compensation expectations.

- **Contact Your Driver:** The food donors also have the ability to contact the delivery driver for any issues with their pick up or any problems with the order itself: missing labels, missing dates on food items, misplaced boxes, etc.
- **View Requirements:** The donors can view the requirements for eligible food listings, so that food donations can be safely consumed by the customers.
- **NoLeftOver - Drive:** This interface is designed for delivery drivers, and will display upon logging into a driver's account. The unique features that delivery drivers have are: find routes, commit to orders, see a selected route and the compensation for that route, and see the address and location of the order on an interactive mini-map.
 - **Find Routes:** The delivery drivers will have the option to manually input a route address, or they can select Commit! for automatically routing to the location for pick-ups. Manually inputting the address into find routes will auto-generate the address upon typing the first few keywords in the search.
 - **Commit!:** This feature allows drivers to commit to an order for pickup from the donors and delivery to the customers. The address will auto-populate and route to the locations of the donors and customers accordingly.
 - **Selected Route (non-interactive):** The selected route will calculate the compensation for the delivery route if it was committed. The selected route dollar amount will change according to the driver selecting different pinpoint locations on the minimap.
 - **Pick-up location (non-interactive):** The pickup location is designated for the pinpoint location on the minimap, and will change once the driver moves the pinpoint location.
- **NoLeftOver - Food:** This interface is designed for customers of the app, and will display upon logging into a customer's account. The unique features that customers have are: search for food items, choose a drop-off window for their deliveries, cancel or reschedule the drop-off appointment(s), and contact the driver for other purposes.
 - **Search for Food:** The customers can search for the type of food, name of the food, and/or the category of the food using this feature on the app. The most relevant query will generate and be on the food items list.
 - **Choose Drop-Off Window:** The customers will have the ability to set a drop-off window for their food order from the app. A delivery driver will get assigned to deliver the food items within the drop-off window.
 - **Cancel/Reschedule:** The customers will manage their food orders through this feature on the app. They're able to cancel or reschedule deliveries by choosing a different date and time for the delivery or cancel the delivery completely.

- **Contact Driver:** The customers will be able to contact the driver for emergencies that prevent them from receiving their food orders.
- **Standard Features:** Available across all interfaces.
 - **Contact Us:** The ability to contact the admin of the NoLeftOver app. For resolving issues or any concerns the users may have and cannot resolve themselves. Standard across all interfaces. Options can be: a phone call to the number listed on the app, a Q/A forum page, or a direct message to the support team through the live chat feature on the app.
 - **Live Chat:** Chat feature to seek customer support from the NoLeftOver support team.

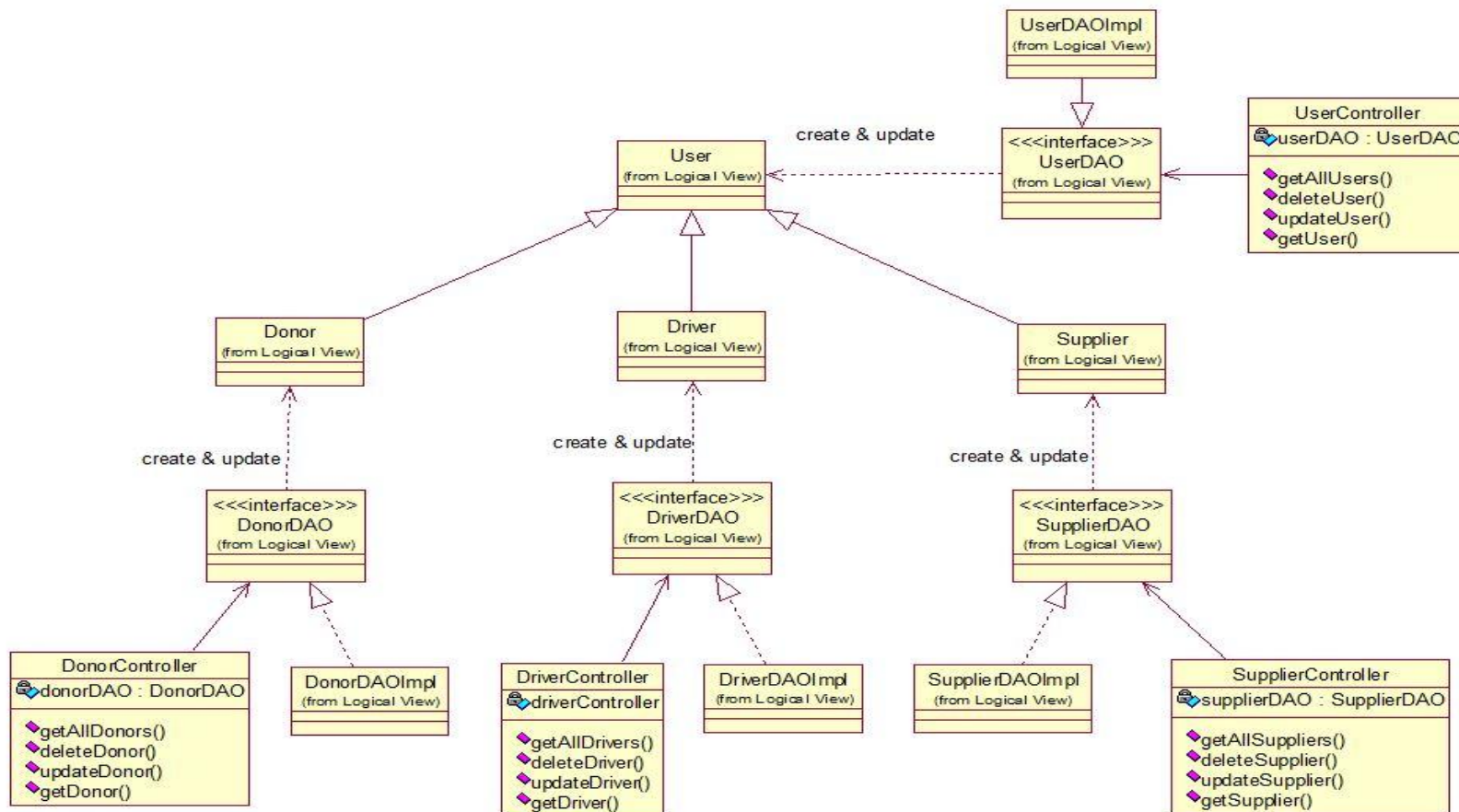
APPLICATION/LOGIC TIER

The logic layer is the application's heart. Using business logic, this layer processes information obtained in the presentation tier, frequently in conjunction with additional information obtained in the data tier. The application tier can also add, delete, or modify the data tier.



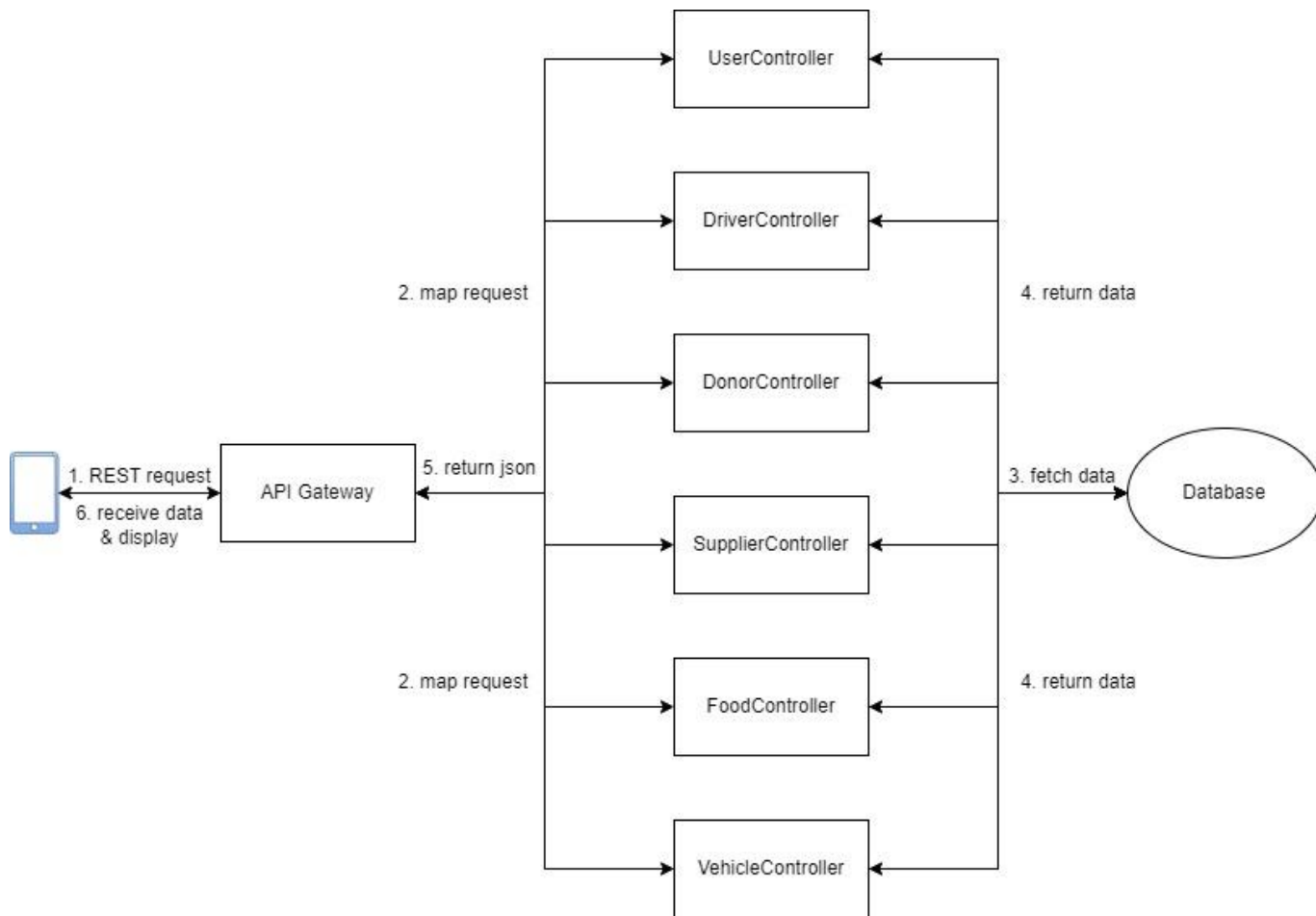
Spring MVC with Boot is our first choice for developing the NoLeftOver project and hosting it on our AWS server since it is a lightweight framework that enables auto-configuration and supports the JDBC framework, which improves productivity, reduces mistakes, and eliminates SQL injections. Following the MVC pattern also helps us to make complicated application development much more manageable by dividing it into three components:

- Models: the code often mirrors real-world objects, and data models, and is used to keep raw data from the database before returning it to the client. For the NoLeftOver project, we'll have six models in total:
 - User: holds all of the user's fundamental information, such as name, phone, street, city, and so on.... This class also serves as a container or parent class for the Driver, Donor, and Supplier classes, as these classes require all of the User class's data as well as some extra unique attributes to indicate their roles.
 - Food: this class contains information regarding food, such as name, category, expiration date, quantity, etc...
 - Vehicles: This class consists of the vehicles' information used to transport food from donors/suppliers to recipients.
- View: The user interface is used to display the current state of models and re-render when the model state changes.
- Controllers: The controller class is responsible for handling RESTFUL API requests, preparing obtained data from the database, and providing the result back to the client as JSON. In order to hide the complexity of the CRUD operations in the underlying storage mechanism, we will use the DAO pattern, the Data Access Object pattern, which allows us to isolate the application layer from the persistence data tier and decouple data models from the database.

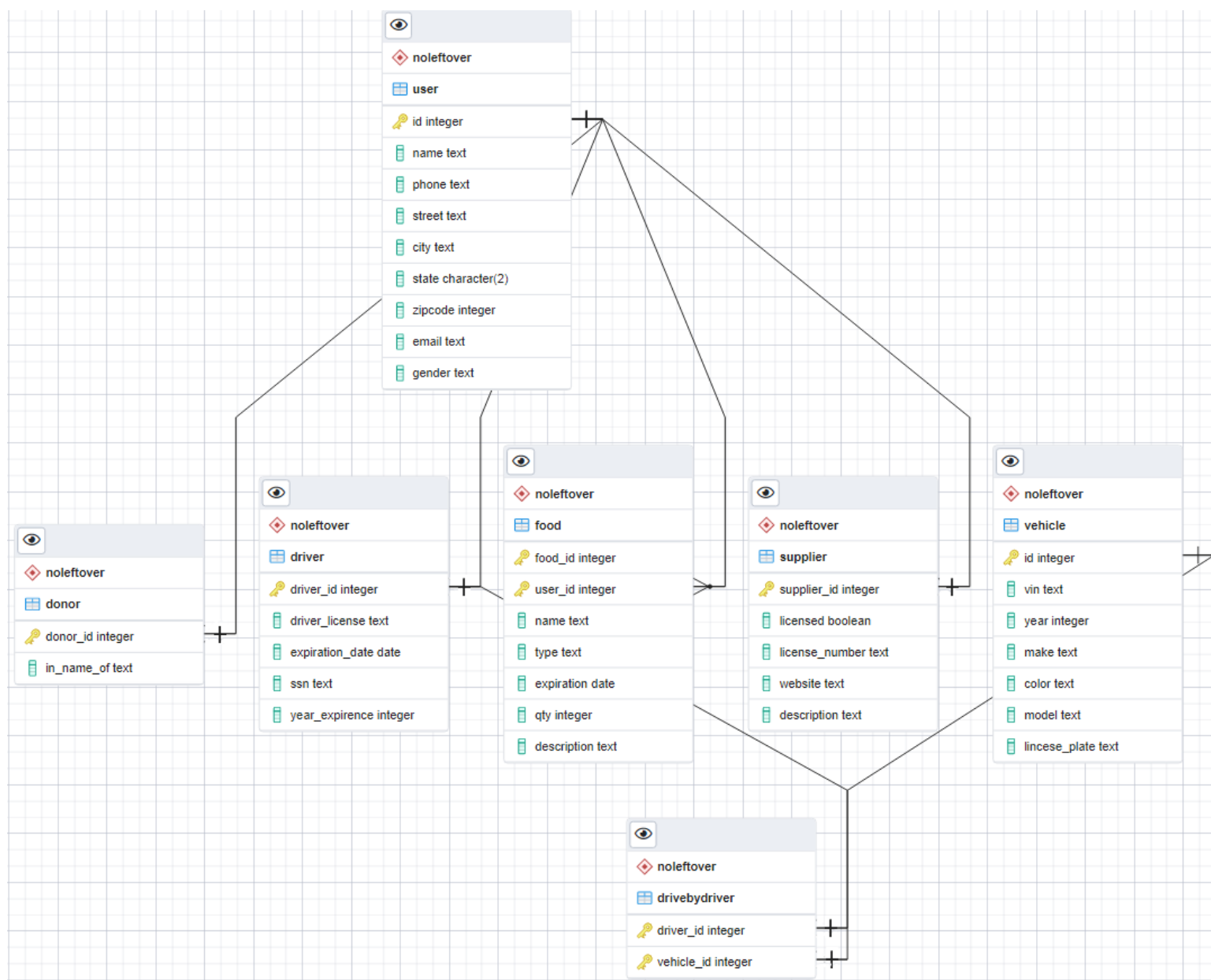


- For each model in the MVC pattern, we will construct an interface that has method signatures and all queries needed to access data, as well as an implementation class that does the logic to extract/update data from the database.
- In the controller classes, we will inject the appropriate DAO that we've created in the last step and implement methods performing all of the access to the database.

To summarize how the application tier works under the hood, the flow of the REST request is depicted in the diagram below. First, the client will submit a REST request to the server — the application tier in order to retrieve data and display it on their end. The request is delivered to the API gateway, which matches the request to the proper controllers or services for call/request processing. The controller will then retrieve information from the database and return it to the client through the API gateway.



DATA TIER



Architecture Design Diagram (Entity-Relationship)

[Explanation of the ER diagram]

The ER diagram demonstrates the major entities interacting within the system and their relationships with each other. There are 7 relational tables within this database design of NoLeftOver and each is associated with one or more entities. The components and features of the ER diagram are briefly explained below:

- **User:** User is the food recipient/food organization/donor who registers with an account on NoLeftOver. The user relation contains the user's ID as the primary key and other attributes

including: user name, phone number, street, city, state, zip code, email address, and gender. Each user would be provided with a unique ID as they register for an account on NoLeftOver. The IDs will be represented by a series of numbers (which is an integer on the database system). Each user, which is considered as a row in the table, will have data in the name, phone number, address, email, and gender attributes. The gender attribute could have null values but other attributes require real data.

- **Donor:** Donors could be an individual or charitable organization that donates food to NoLeftOver. The donor table consists of two attributes: donor's ID and name. Donor's ID is a foreign key from the user table and also the primary key of the donor table. Each donor's ID is unique and represented by an integer. Each donor will be registered with their ID and name on NoLeftOver's account.
- **Driver:** Drivers are in charge of picking up and delivering food from the address of donors to recipients' places. The donor relation contains driver ID, which is a primary key, and other attributes including driver license number, expiration date of the license, social security number, and year of driving experience. Each driver account will be differentiated by their ID and will contain information about the driver's license.
- **Food:** The food entity will contain food ID, user's ID, name and type of food, expiration date, quantity, and description. Each food item (which represents a row in the food table) will be differentiated by an ID and user's ID as primary keys. User's ID is a foreign key from the user table. This relation would need constant updates and modifications due to the addition and deletion of food items.
- **Supplier:** supplier table represents the suppliers (food organizations) who are one of the major forces in donating food to recipient users. The supplier table includes supplier ID as primary key (is also a foreign key from user table), license number, url to website (represented in text) and description. There is an attribute named licensed with the data type boolean. Licensed will represent if a supplier has a license or not based on its data value for this attribute.
- **Vehicle:** a list of vehicles registered for delivering will be contained in the vehicle table of the database system. Each has a unique ID and other non-key attributes including vin number, vehicle's model year, make (car's brand), color of car, car's model and license plate.
- **DriveByDriver:** this relation contains two primary keys which are driver's ID and vehicle's ID. Driver's ID is a foreign key from driver table and vehicle's ID is a foreign key from vehicle table

The relationship between entities in the ER diagram is described below:

- Recipients belong to user table where they need to input personal information about themselves and address to the database system as each food receiver must register for an account on NoLeftOver
- One user in the user table is associated with one donor, driver or supplier in the corresponding tables. The primary keys in donor, driver, and supplier tables are the foreign key from the user ID. Hence, donors, drivers and suppliers are also considered as users of NoLeftOver since they represent as entries in the user relation.

- One user can be associated with one or more food items. The relationship between user and food entities indicates that one recipient can receive many food items, a donor/supplier can make donations of many types of food, and delivery of food items can be handled by one driver.
- Each driver is associated with one vehicle through the DriveByDriver relation. In the driverbydriver table, driver ID and vehicle ID are foreign keys from the IDs in driver and vehicle relations respectively. They are super keys in those relations. We cannot put data about driver and vehicle in one relation as it will generate anomalies which is not a good database design.

Below is the description of some additional features and non-functional requirements that help our system achieve its goals and software qualities including usability, stability, scalability, security, and reliability.

- The server-side infrastructure:
 - All data will be stored and accessed using **PostgreSQL** relational database management. It aids in the preservation of data integrity, data correctness reduces data redundancy to a minimum or zero, data scalability, data flexibility, and simplifies the use of security mechanisms. The entity-relationship diagram can also help us gain from utilizing a relational database. The diagram above, for example, provides a fast overview of the relationships between models in the project NoLeftOver. As a result, developers may have a thorough understanding of the structure and create software by following the information on the diagram.
 - **Load-balancing** is a critical necessity for NoLeftOver since, with the increase of online traffic, a single server setup may bring the entire system down. A load balancer can assist in distributing incoming traffic evenly across servers designated in a load-balanced set. The private IPs are also hidden in this configuration, and users cannot directly reach the servers, which contributes to improved security. If one of the servers goes down or takes too long to respond, a load balancer may divert the incoming requests to other servers, avoiding bottlenecks and enhancing service availability.
 - **Database replication** can improve reliability and availability with a master-slave model. In most cases, a master database is only available for write operations, while a slave database receives copies of the master database's data and is available for read operations. For all operations, such as insert, remove, and update, the master must receive all data modified to replicate continuously and send it to the slaves. Supporting database replication can improve performance significantly by allowing multiple queries to be executed concurrently. Replicating data, furthermore, can enhance availability and reliability because when one of the database servers crashes or is destroyed by natural disasters, all data will be preserved and accessible from everywhere.

- The client-side application:
 - This software will be available across platforms, increasing the likelihood that every person can receive assistance from their communities.
 - This software will be connected with Google Maps to allow drivers, donors, and users to easily and effectively access the services.
 - The application also allows users to be notified when food supplies are available in their areas to pick up.
 - The application will allow users to use biometric authentication for quick login.

➤ **Architecture Review:**

The team's original design decisions have remained. We made use of placeholders in our original proposal and have now filled them in with specific development choices. Our initial design consisted of making use of AWS services to support our backend development which included the choice of multiple application servers, PostgreSQL as our database, and using React Native to develop our iOS and Android mobile applications. We have kept to our original design plan.

We have filled in the placeholders for our original architectural design. We decided that our application servers would run on Ubuntu Server 20.04 LTS, and that we would make use of Spring MVC with Boot as our choice for developing the core of the NoLeftOver project for the server-side of our applications.

04 - ALTERNATIVES CONSIDERED

➤ **Alternative #1: Allowing regular people from nearby neighborhoods and homes to donate food from their homes and let consumers to self-pickup.**

- Description: The *NoLeftOver* application will have an added feature that permits regular consumers to donate their food, of any amount, to other consumers for pick up only. This added feature will allow regular customer accounts to list their food donation items the same way as donor accounts; however, the donation listings created by consumers will not have the delivery option because of the small amount of food donations. This feature will allow neighbors to donate their food to one another, which enhances the interconnection amongst the neighborhoods and eliminates food wastes disposed of by families and homes.
- Comparison of Approaches: The original version of the *NoLeftOver* app does not have this feature implemented; And, adding this feature would cause further complications with the safety and security of the customers since everyone would be able to sign-up for a customer account and this allows room for illegal activities. Furthermore, the pick-up only option for customer donations will mislead food recipients and discourage them from using the app. Major donors like restaurants,

grocery stores, supermarkets, etc. may mistakenly use customer accounts for listing. As a result, we have decided to only allow major donor groups to donate food items.

➤ **Alternative #2: Designing separate applications for each target user group instead of different UIs on a single *NoLeftOver* application.**

- Description: The single *NoLeftOver* app will be replaced with three different functional apps designed specifically for each group of audience: *NoLeftOver* app for consumers, *NoLeftOverDonation* for donors, and *NoLeftOverDelivery* for delivery drivers. The users' inputs will be interconnected and updated constantly and consistently across the three apps. This would allow the three apps to have high performances and light storage memory due to the loads and memory being distributed into three different applications instead of having it on one single system.
- Comparison of Approaches: Although this method of dividing the *NoLeftOver* app into three separate applications would increase performances and reduce the memory loads, it would significantly extend the development timeline which is cost-inefficient and time-consuming as we have to come up with multiple architecture designs and interaction designs for the three systems. Another downside of having multiple systems is that there will be higher maintenance costs associated with the overall systems. Therefore, we decided to omit this multi-system approach to designing our *NoLeftOver* system.

➤ **Alternative #3: Adding a feature for donors to choose to give their food explicitly for a certain recipient group.**

- Description: By adding this feature, during the food donation listing process, donors will have an optional field to choose the recipient group(s) for their food donations (homeless shelters, orphanages, low-income households, etc.) or they could choose for everyone. This feature will allow a more targeted approach to focus on the recipient group(s) that need more assistance or a certain type of food.
- Comparison of Approaches: This feature will be beneficial to those most-in-need recipient groups, however, it will unfairly distribute donations for everyone else. Donors may have personal references toward a certain group(s) and this will cause unequal distribution of food donations and conflict-of-interest for donors using donation as a tax deduction method. In conclusion, it is better to have food donations be equally distributed among every recipient group(s) and automatically decided by the app, and not donor parties.

➤ **Alternative #4: Implementing a constant-update-of-the-delivery-process feature similar to that found in the UberEats app (picked up, current location/route, arrived).**

- Description: The *NoLeftOver* UI for consumers/recipients will have an interactive map integration onto the app, and it will update the exact location of the driver and where the driver is on the map. There will be a status information on each food donation order; The status information will be of four variations: placed, picked up, on the way, and delivered. Recipients will be able to track the status of their food packages/orders either on the interactive map or the status information bar.
- Comparison of Approaches: Although it can be useful information for recipients to get an idea of where their food packages/orders are at and be prepared to receive them promptly, it may cause privacy and security issues for the delivery drivers. In addition, this feature is not practical for the *NoLeftOver* app as it is not a for-profit food delivery service like UberEats or Grubhub where delivery drivers get the incentive to provide fast delivery services. The *NoLeftOver* system will prioritize convenience and efficiency over fast services; the main purpose of this system is to get as much food to as many people as possible while limiting the costs associated. Therefore, delivery drivers will probably gather as many food donations for donors as possible and deliver them in mass quantities to as many recipients on the route as possible.

05 - ETHICAL ANALYSIS

Technology interacts with humans daily and it is vital to ensure that it conforms to the ethical needs of the society and the standards that have been put in place concerning human decency and acceptability. The lines of code that we write as information technology personnel have a moral implication for the users and the developers. As developers put down the code they are writing it is important to ensure that it is safe for the users, it is important to consider the human factor in computing to ensure that there are no hazards. For the application that my team wishes to develop, the ethics we considered include:

- Security aspect: the application should be able to ensure the anonymity of the users in the application. Data that is obtained through the users accessing the system could be confidential and considered private by the users, it is, therefore, important to provide them with assurance and confidence while using the application.
- The food made available to the recipients should also be healthy and not hazardous to their health. The food security assessment should be involved in the inspection of the products to verify if they are consumer-friendly.
- The application should also provide a sense of belonging to the users, it should avail products that belong to all races and not only focus on one. Having such differentiation makes the given race a target for attack from dangerous people who wish to cause them harm.

- Tradition and conformity ethics; under this ethical aspect our application should be able to respect the traditional aspect of availing fresh food only and the application should not deliver expired food. The application should also stick to providing free products and not proceed to sell the products.
- The software we are developing could create a barrier to use by those who do not have smartphones. The application also requires the internet and it could disadvantage those who have issues with internet connection. This could be mitigated through the local governments and other sponsors contributing to the installation of the internet and having offers made on the sale of smartphones.
- Our application would involve the use of data centers and servers which would mean that the electricity consumed is in large amounts and causes an increase in the carbon footprint of the society. This can be mitigated through the use of data centers that are located in strategic places that ensure the emissions are cut down. The application could also be disadvantageous to the retail companies because customers could fail to shop and wait for the near expiration dates to gain products. This can be mitigated through collaboration with the local governments to ensure those in need of the food are the ones being supplied with the food. People's living situations are contained in the software we are building, and this is private data that may be weaponized. End-to-end encryption and other security measures should be considered and implemented. The data should be safeguarded so that unauthorized people cannot access it.