**Full Stock: Restaurant Ordering and Automated Inventory Management System**

**Team 8**

Cameron Nguyen, Nathan Carney, Rachel Freedman, and Sam Song

[**Executive Summary**](#_ruf8qr4hv95o) **3**

[**Project Plan**](#_vr03363g0kgo) **4**

[Project Statement](#_38v5nvufq7y) 4

[Development Approach](#_am93y9wcej0) 5

[Figure 1.1 Parallel Development Methodology Diagram](#_oflpgz9nklwo) 6

[Project Work Plan](#_tbgawiu3eosc) 7

[Table 1.2 Work Plan Overview](#_c611q1rpehq7) 8

[Figure 1.3 Work Breakdown Structure](#_j9cczlehvclg) 9

[Staffing Plan](#_sso9kb320jy8) 10

[Figure 1.4 Staffing Plan](#_e87vrgey8eu9) 10

[Project Time Estimate](#_luhz32prf26j) 12

[Project Cost Estimate](#_s6xv9ilfrzok) 12

[**Requirements**](#_p1rr7w8k7xg1) **13**

[Table 3.1 Requirements Traceability Matrix](#_r522dvrkocn8) 16

[Methodology](#_elxnezskksof) 18

[**Business Requirement Use Case**](#_ry39xdwsqjq) **21**

[**V. User Requirement Use Case**](#_z9oz8p1pqisi) **24**

[**VI. Data Flow Diagrams**](#_7ppgtxdczkoi) **28**

[Context level Diagram](#_wgb6yawijw8r) 28

[Level 0 Diagram](#_khgar1ayqweu) 28

[Level 1 Diagram](#_fz80dfzcblaw) 29

[**VII. ERD Diagram**](#_z4htf1ufphyz) **30**

[VIII. CRUD](#_jth5k9okntnd) 31

# **Executive Summary**

Whether people need a quick bite to eat or need something different, restaurants can satisfy that craving. In the US, the restaurant industry's share of the money spent on food has risen from 25% in 1955 to 51% in 2019. This industry employed 15.3 million people in 2019, which is approximately 10% of the working population[[1]](#footnote-0). By the end of the year, the industry is projected to make $899 billion.

America’s largest fast-food franchises have access to the country’s brightest consulting firms and most advanced analytics tools. With limited access to capital, independently owned eateries are often left to sort out their own logistics nightmare. The onset of COVID-19 has only exacerbated this glaring weakness and forced 60%[[2]](#footnote-1) of the nation’s restaurants to close their doors. The pandemic has brought drastic negative economic repercussions for the restaurant industry. This has forced many of these establishments to reevaluate their inventory management and customer ordering processes. The inevitable reopening of America’s economy presents a unique opportunity for restaurants to implement information technology to bolster profit margins, assure safe, touchless customer experiences, and streamline the archaic ordering and inventory management process.

Our team has created Full Stock that will help restaurants save money and better target consumers. The application is composed of three parts: (1) People can use this technology to order from a restaurant. (2) The software will tell the restaurant the demographics of the user. (3) Every time an order is placed, our system will update the inventory system and show how much is in stock. Overall, our team wants to help restaurants and streamline operations.

# **Project Plan**

## Project Statement

The purpose of this project is to help streamline business operations for restaurants. We ultimately hope to develop a business system that allows customers to complete the entire restaurant patron journey from their mobile device. If a restaurant wishes to utilize our services, we will place an interactive version of their menu on our mobile application. Here, patrons can create a checkout cart of the items they would like to order as well as not any special requests or dietary restrictions. The benefit is twofold, as restaurants reserve the right to fluctuate prices on our menu in real-time to reflect demand data. Also, items that are no longer available due to inventory shortages will be removed from the menu. Our application will then send the customer orders and immediate payment to the restaurant which will begin order preparation. The application user will then be sent an estimated preparation timeline for their order. We hope to further integrate our system with the restaurant's existing inventory management processes. Each time a customer places an order there will be an estimated ingredient requirement. The required ingredients will then be subtracted from a database of stock and warn the restaurant when levels begin to run low. The goal of this feature is to ensure inventory is never out of stock. Lastly, our system will provide demographic and demand data to restaurants. This will encourage them to update their menu and run specials accordingly. This will also help companies better target customers. With COVID-19 and the detrimental impacts it has on the restaurant industry, we want to help companies save money wherever possible. Full Stock will bring in more customers, help keep track of inventory, and allow customers to get takeout more easily.

## Development Approach

Our team decided to use the parallel development methodology to execute the project. This methodology was chosen for its ability to have sub projects that can be worked on at the same time. The restaurant industry is always changing due to COVID-19. Our team wants to provide our clients with a speedy project turnaround time. Below is a general overview of the parallel methodology.

## 

### Figure 1.1 Parallel Development Methodology Diagram: Figure 1.1 shows the execution of our project using the parallel methodology. Below is a more in-depth description of our team’s tasks.

A potential project plan that follows parallel methodology is as follows:

Planning

* Looking at the restaurant’s existing structure and how our application can best help the company.

Analysis

* Deciding what the restaurant needs and the problems the organization faces.

Design

* Taking the needs of the restaurant and designing an application that caters to those needs.

Subproject 1

* Programmers code the portion of the application that allows people to order and the demographics of the customers.

Subproject 2

* Programmers code the portion of the application that tells the back of the house how much inventory is left.

Subproject 3

* Programmers design the user interface for the application.

Implementation

* The application is installed and staff at the restaurant is trained on how to use it.

System

* The application is put into practice.

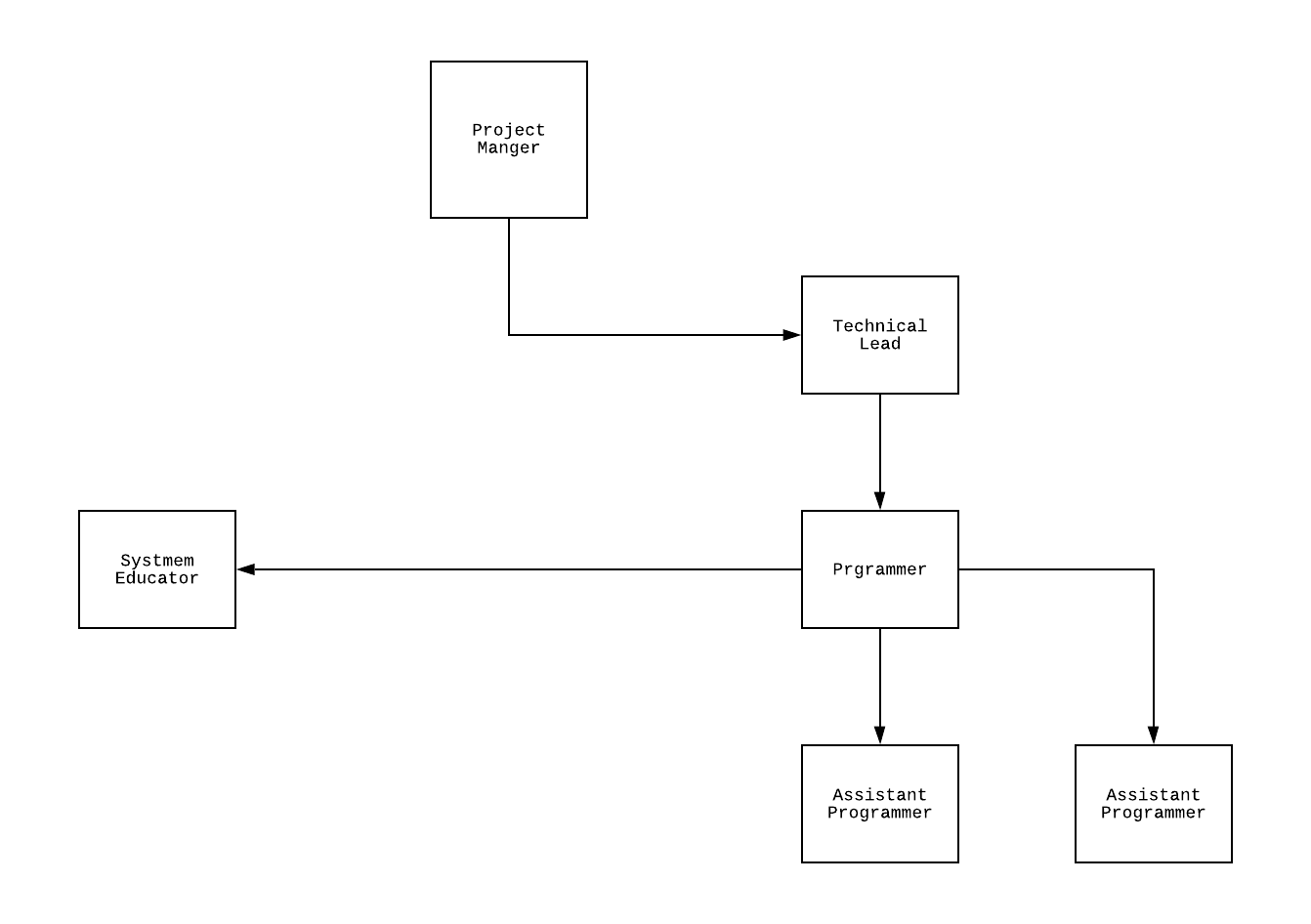
## **Project Work Plan**

| **Task ID** | **Task Name** | **Assigned To** | **Duration (Days) (Estimated)** | **Start Date (Estimated)** | **End Date (Estimated)** | **Dependency** | **Status** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Design Phase | Tech Lead + Programmer | 20 | 11/1/2020 | 11/21/2020 |  | Open |
| 1.1 | Create an Outline | Tech Lead + Programmer | 5 | 11/22/2020 | 11/27/2020 |  | Open |
| 1.1.2 | Get feedback from client | Project Manager | 5 | 11/28/2020 | 12/1/2020 |  | Open |
| 1.2 | Create a Project Timeline | Tech Lead + Programmer | 3 | 12/2/2020 | 12/4/2020 |  | Open |
| 1.2.1 | Assign Tasks | Tech Lead + Programmer | 2 | 12/5/2020 | 12/6/2020 |  | Open |
| 1.3 | Coding: Ordering Features + Demographics | Programmer | 10 | 12/7/2020 | 12/16/2020 | 1.3 +1.3.1 | Open |
| 1.3.1 | Coding: Inventory + Backend Connection | Programmer | 10 | 12/17/2020 | 12/26/2020 |  | Open |
| 1.3.2 | Design User Interface | Programmer | 10 | 12/27/2020 | 1/5/2021 |  | Open |
| 1.3.3 | Review Application | Tech Lead + Programmer | 14 | 1/6/2021 | 1/15/2021 |  | Open |
| 1.3.4 | Get feedback from client | Project Manager | 5 | 1/16/2021 | 1/20/2021 |  | Open |
| 1.4 | Implement project in Restaurant | Systems Educator | 14 | 1/21/2021 | 2/4/2021 |  | Open |

### Table 1.2 Work Plan Overview: The Project Plan above shows the plan for our deliverable to the restaurant. This is just a tentative plan and can always be revised as needed. The project plan does not plan for the preliminary speaking to clients. That task will be done before the design phase.

### **Figure 1.3 Work Breakdown Structure:** This Work Breakdown Structure (WBS) shows all tasks that need to be completed. The WBS like the project plan does not account for the initial client meeting. The WBS only shows the structure for the project itself

## Staffing Plan



### Figure 1.4 Staffing Plan: This staffing plan shows a potential layout for our team members. The execution and implementation of Full Stock has clear hiecharical structure. Please see below for a more detailed description of the plan.

To implement the project our team designed a staffing plan to maximize efficiency. The project manager is on top of the chain of command and oversees the entire operation. The project manager checks in with the technical lead to ensure that all deadlines are met. This person is also responsible for communication with this company and the client. The technical leader is responsible for overseeing the programming and coding portion of the project. The leader ensures that the deadlines are met and communicates with the project manager about the progress of the tasks. The programmer is in charge of the coding and user interface of the project. The programmer also communicates with the technical lead if any problems arise. The assistant programmers are responsible for helping the programmer with any technical aspects of the project. The assistant programmers are not always needed for all projects. The size and complexity of the project dictate whether or not the assistance is required. Lastly, the System Educator is responsible for teaching the client how to use the technology. This person is in contact with the programmer to ensure that they know how everything works. This project team is at most six people, which helps tasks get done. A hierarchy is created to make sure communication is efficient. Overall, this staffing plan is designed to promote efficiency and productivity.

## Project Time Estimate

Limitations with this estimate include no errors and no personal days are taken. If errors or personal days occur, then the time for project completion is extended by the number of days for which a team member is out of the office or the days needed to correct errors. Also, the 98 days estimate does not account for the initial meeting between the project manager and the client. We are anticipating 1 day for the initial client meeting. If the client requests additional time, this might impact the project delivery timeline. The timeline for the project has the ability to always change.

## Project Cost Estimate

The app industry in 2018 was worth over $60 billion, and developers made over $20 billion from app sales. As technology expands rapidly, the app industry brings in more money. According to the website Utility[[3]](#footnote-2), apps can range from 80,000-250,00 depending on the complexity and features of the application. The project cost is based on labor wages and the complexity of the app. For each client, the price will change. The only cost that will remain constant is the labor wages. All employees are salaried and not paid per hour. The complexity of the application will mostly rely on the user interface. For example, companies that want their company logo and a special color scheme will be charged more. The overall cost will be determined between the project manager and the client.

# Requirements

Business Needs Requirement: **Provide estimates for the quantity of ingredients used in each dish offered on the menu and assure this information is up to date with any portion changes or menu additions.**

Demand volatility has become a challenging reality for many restaurant owners during the Pandemic. Traditional ordering schedules and systems have become obsolete leaving many of our interviewees with empty fridges. Thus, it is crucial our system accurately estimates real-time inventory levels. In order for this to happen, it is essential that our business users provide accurate ingredient estimations per dish. When the restaurant confirms a diner’s order through the system, the stock count will automatically be reduced by the ingredient estimation required to make the ordered dish.

User Needs Requirement: **Confirm a customer’s menu item selection and approve the system’s estimated wait time.**

By its very nature our system is multi-faceted. We will be serving both primary users(the restaurant owner) and secondary users(the diner). Both have a unique set of user requirements but for the scope of this report we will just focus on the primary user’s requirement. When a diner places an order through our system, the restaurant will receive a notification asking if they’d like to confirm. It is essential that restaurant workers confirm the order and validate or adjust the estimated wait time that will be sent back to the user.

Process Oriented Functional Requirement: **The system must generate an out-of-stock notification when an ingredient inventory quantity reaches the item reorder point.**

The onset of the COVID-19 pandemic has introduced an unprecedented level of demand volatility. From our analysis we have found that a majority of dining establishments rely on traditional traffic flows for contacting ingredient suppliers. Most of our interviewees have set “Ordering Days” in which they manually take inventory counts and place orders, going largely off “gut feelings” and historical trends. We have found that many managers are ignorant to what is in stock until the set ordering days arrive, proving disastrous in times of uncertain demand such as now. The system remedies this by tracking order inflow and subtracting the estimated ingredient requirement for each new order from the inventory database. This allows for real time inventory levels as well as out-of-stock notifications.

Process Oriented Functional Requirement: **The system must track customer interactions with digital menus across Android and IOS devices and send completed orders to the restaurant.**

The global pandemic has forced businesses to shift to a completely contactless transaction model. Restaurants are no exception, and we believe that our system’s seamless ordering experience will smooth this transition. Secondary users will not only be able to view their favorite eateries menu from their mobile device, but they will also be able to place orders by interacting with the menu. This assures a safe and efficient ordering process.

Information Oriented Functional Requirements: **The system must include real-time menus and prices of all restaurant locations on the platform**

We pride ourselves on restaurant users’ ability to fluctuate prices and edit menu items with ease. Such functionality will allow for greater profit margins, reduction in orders of items that are out of stock, and the ability to run “specials”. It is thus pivotal that the system has the most up to date menu items and prices. This will assure that all items being selected from the menu are in stock and can be made.

Information Oriented Functional Requirements**: The system must retain individual customer order history for six months**

We hope to provide users with a “recently ordered from” section to provide easy access to favorites from our restaurant roster.

Political/Cultural non-Functional Requirement: **Users credit card information must be kept private to assure compliance with PCI DSS**

The Payment Card Industry Data Security Standard (PCIDSS) is an information security standard for organizations that handle credit card transactions. The major card brands mandate that all merchants abide by this standard when making sales to prevent card fraud. Our secondary users will be uploading their credit card information to our application so that they can pay restaurants wirelessly. In order to assure compliance our system must abide by the PCIDSS standards.

Performance Non-functional Requirement : **The system updates the restaurant’s order queue within one minute of the customer placing the order**

It is crucial diners get their orders prepared as quickly as possible and that restaurants have ample notice to prepare them. Our goal is to simulate the speed of ordering in person and as such we require our system to update the restaurant’s que within one minute of the user ordering.

**Requirements Matrix**

## **Table 3.1 Requirements Traceability Matrix:** The following table details the method for tracking requirements and their implementation

| **Full Stock** |  | | |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Requirement ID\*** | **Requirement Description\*** | **Requirement Priority\*** | **Requirement Owner\*** | **Requirement Level\*** | **Requirement Attribute\*** |
|  |  |  |  |  |  |
| PROJ User 10 | Log into system using email and password | M | Dev Test Team | Capability | F |
| PROJ User 20 | Restaurant Selection | M | Dev Test Team | Capability | F |
| PROJ User 30 | Menu Item Selection | M | Dev Test Team | Capability | F |
| PROJ User 40 | Input payment information and confirm order | M | Dev Test Team | Capability | F |
| PROJ User 50 | Update Menu information and prices at restaraunt's discretion | M | Dev Test Team | Business-Level | N |
| PROJ User2 10 | Confirm User 1's Order and system's recommended wait time | M | Dev Test Team | Capability | F |
| PROJ User2 20 | Confirm Order Completion and Pickup | M | Dev Test Team | Capability | F |
| PROJ Business 10 | Initialize system by uploading menu items and providing estimated ingredient use for each menu item | M | Dev Test Team | Business-Level | F |
| PROJ Functional 10 | System must generate an out of stock notification when ingredient inventory quantity reaches item reorder point | M | Dev Test Team | Solution | F |
| PROJ Functional 20 | The system must track customer interactions with digital menus across Android and IOS Devices | M | Dev Test Team | Capability | F |
| PROJ Functional 30 | System must send completed order notifications to the restaurant | M | Dev Test Team | Capability | F |
| PROJ Functional 40 | System must process user payment information and send it to the merchant | M | Dev Test Team | Capability | F |
| PROJ Functional 50 | System must provide an accurate wait time estimate pre-defined by restaurant | M | Dev Test Team | Capability | F |
| PROJ Functional 60 | System must deduct ingredient inventory for each menu item | M | Dev Test Team | Solution | F |
| PROJ Functional 60 | System must include real time menus and prices of all restaurant locations on platform | M | Dev Test Team | Solution | F |
| PROJ Non-Functional 10 | User credit card information must kept private | M | Finance Department | Business-Level | N |
| PROJ Non-functional 20 | System must update the restaurant's order queue within one minute of the customer placing the order | S | Dev Test Team | Capability | N |

## **Methodology: The most effective elicitation technique for this project will be interviews**

In selecting a requirements elicitation technique our goal was to gather data for a report that showcased the validity of our entrepreneurial endeavor. With this in mind it is clear that interviews were the most effective way of building rapport and interest with investors and potential users. Our interview schedule consisted of five locally owned “fast-casual/takeout” restaurants in the DMV Area. To assure diversity in perspectives we spoke with individuals in three different business levels at each location. This includes, a manager, a line cook, and the store owner.

Manager

Purpose of Interview: **Understand current inventory management process and current/past attempts to implement information systems**

Questions:

Close-Ended:

* Do you currently use an information technology system to expedite your inventory management? If not have you tried in the past?
* How has the way your customers place orders changed during COVID-19?
* What information is currently missing from your inventory reports?

Open-Ended:

* What do you think of the current inventory management process?
* What are some problems you face on a daily basis?
* What are some problems you see with the current customer ordering process?

Line Cook

Purpose of Interview: **Gain insight into the Customer Ordering Process and Food Preparation Process**

Questions:

Close-Ended:

* How many orders do you receive per day?
* Do you often find inventory stock outs a problem?
* What has changed about the ordering and food preparation process since the onset of COVID-19?

Open-Ended:

* Do you see any issues with the way orders are managed?
* What are some problems you face on a daily basis?
* What are some things you would change to make your job easier?

Restaurant Owner

Purpose of Interview: **Understand the Strategic Vision for the Restaurant**

Questions:

Close-Ended:

* Do you currently use an information technology system to expedite your inventory management? If not have you tried in the past?
* Have you recently unveiled any new strategic initiatives in light of COVID-19?
* What information is currently missing from your inventory reports and customer orders?

Open-Ended:

* What would you say has been the biggest threat to your strategic initiatives?
* What do you believe customers want out of the ordering process during COVID-19 ?
* Where do you see the industry headed in the next 10 years?

A complete transcript was recorded of the entire interview and compiled into key takeaways. This was then sent to the respective interviewees for their approval and securely placed in our records.

# Business Requirement Use Case

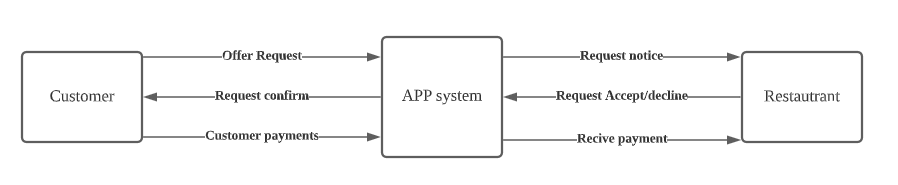
| **Use Case Name:** Receive an order | | | **ID: 2A** | | **Priority:** High | |
| --- | --- | --- | --- | --- | --- | --- |
| **Brief Description:** The restaurant receives the request, and decides to accept or decline the order. | | | | | | |
| **Actor:** The app user’s requested restaurant | | | | | | |
| **Trigger:**  **Type**  **External**  **Temporal** | | | | | | |
| **Preconditions:**   1. The restaurant is registered in the system database 2. The restaurant is selected by the app user, and is within their pickup range during operational hours 3. The app system is fully operational and online | | | | | | |
| **Normal Course**   1. Receive orders from the app user 2. The system will notify the restaurant of an order request from the user 3. The system will notify the restaurant of the order details 4. The restaurant will check the requirements to prepare the food 5. If the requirement is met, the restaurant will accept the request and the system will notify the user that the request has been accepted 6. If the requirement is not met, the restaurant will give a brief explanation (e.g. ingredient shortage shortage, equipment malfunction) 7. The system messages the user about the order status (accept/decline) and uploads data for the inventory database 8. The system receives the pickup information from the restaurant 9. The system will record and send the pickup information to the user 10. The restaurant completes the order, and the user picks it up. 11. Restaurants receive payment after the order is complete | | | | **Information for step**   1. Restaurant receives the request 2. Restaurant accepts or rejects the request 3. The restaurant confirms order and pickup 4. Restaurant finishes preparing the order 5. Pickup is complete, and payment is received | | |
| **Alternative Course(s):**   1. Receives an order cancellation 2. The system receives a cancellation request from the user if it is within the allowed timeframe 3. The system will notify the restaurant to cancel the order 4. The restaurant cancels the order | | | | | | |
| **Postconditions:**   1. The system stores the order request in the inventory system 2. The system updates new inventory data information to the inventory database 3. The restaurant receives a notification about a requested order 4. Order and pickup request is confirmed | | | | | | |
| **Exceptions:**  E1: Delivery is complete   1. The system displays the message “order has been received by the user; payment is complete” 2. The system updates new inventory data   E2: Order cancellation request   1. System displays the cancellation reason and updates status to complete | | | | | | |
| **Summary:**  **Inputs Source Outputs Source** | | | | | | |
| Inputs:   1. Order request notification 2. Restaurant information 3. Menu ID | Source:   1. System inventory database 2. Notification system 3. App | Outputs:   1. Order ID 2. Pick up information | | | | Source:  1.System inventory database  2. Notification system  3.Manager  4.Employee  5.Supplier |

# V. User Requirement Use Case

| **Use Case Name:** Request an order | | | **ID: 2B** | | **Priority:** High | |
| --- | --- | --- | --- | --- | --- | --- |
| **Brief Description:** The app user selects the restaurant and menu items they want, and sends an order request to the system. The system will transfer the request to the restaurant, and update inventory levels when the restaurant confirms. | | | | | | |
| **Actor:** An app user who wants to order from a restaurant | | | | | | |
| **Trigger:**  **Type**  **External**  **Temporal** | | | | | | |
| **Preconditions:**   1. The user registers from the app 2. The app user uploads a valid payment method and billing address 3. The app system is fully operational and active online | | | | | | |
| **Normal Course**   1. Requests order from a restaurant 2. The system will show the user the available restaurant based on the pickup range and operational hours 3. The system offers users a restaurant from the list 4. If there is no restaurant available due to the location or time, it will display a notification. 5. The system offers a fully descriptive menu from the restaurant to the user 6. If the quantity of an item in the menu is in shortage, it will display as “unavailable” from the menu list 7. If the quantity of the item is sufficient, the system will show the available quantity on the list 8. The system gives users different options for pickup including the time 9. The system shows confirmation from the user 10. The system notifies the restaurant with the order request information, and waits for confirmation from the restaurant. 11. If the restaurant confirms the request, the system will receive confirmation and waiting time which will transfer back to the user. 12. If the restaurant declines the request, the system will notify the user and offer an alternative option 13. The system arranges for pick up 14. The system updates inventory data for the restaurant and future user | | | | **Information for step**   1. User signs into the app 2. User selects the restaurant 3. User selects menu items and quantity 4. User selects the contactless pickup option and time 5. The user confirms selection, and sends an order request 6. Restaurant receives a notification, and approves or rejects the request 7. The user receives confirmation from the restaurant, and picks up the order at the specified pickup time 8. The user receives their order, and the system closes the order | | |
| **Alternative Course(s):**   1. Request error 2. User stops responding to the order request 3. system waiting for a response 4. system stops waiting for a response when the selected restaurant is out of operation, or is out of the user’s pickup range 5. The user receives a declined notification from their order 6. system displays the reason   a1. the system will show an explanation from the restaurant  a2. if the decline is due to the payment method, the system will suggest to users to input an alternative payment method until payment is successful  a3. if the decline is due to a system malfunction, system analysis will report a failure in system services, or internet connectivity issues, etc.   1. User cancels after the request has been sent 2. system notifies if cancellation is still available 3. restaurant receives cancellation notification and executes cancellation. | | | | | | |
| **Postconditions:**   1. The system stores the order request in the inventory and order system 2. The system updates new inventory data information to the inventory database 3. The restaurant receives a notification about an order request 4. User receives different messages based on the order information | | | | | | |
| **Exceptions:**  E1: Order is confirmed and waiting to finish   1. The system displays the message “ your order request was confirmed by the restaurant, pickup by a certain time.” 2. The system asks the user if they would like to place additional orders   2a. user requests new order  2b. user exits app while waiting for order to reach completion  E2: Order request is declined   1. system displays the decline reason (a decline from restaurant or payment decline, etc.) 2. The system offers different options based on the circumstances   E3: Order cancellation   1. The system sends a user cancellation notification to the restaurant 2. The notification is received and the order is cancelled completely | | | | | | |
| **Summary:**  **Inputs Source Outputs Source** | | | | | | |
| Inputs:   1. Restaurant ID 2. Menu ID 3. Payment information 4. Customer ID 5. Customer Information | Source:   1. System inventory database 2. Notification system | Outputs:   1. Order ID 2. Pickup Information 3. Order Request Notification | | | | Source:  1. Notification system |

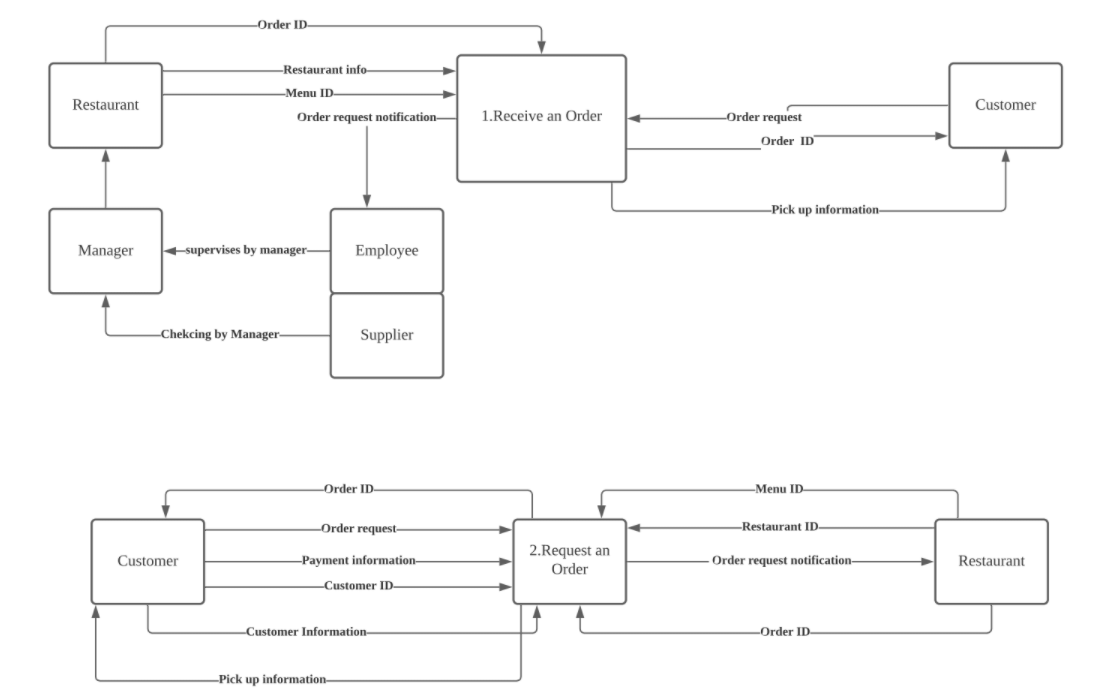
# VI. Data Flow Diagrams

## Context level Diagram



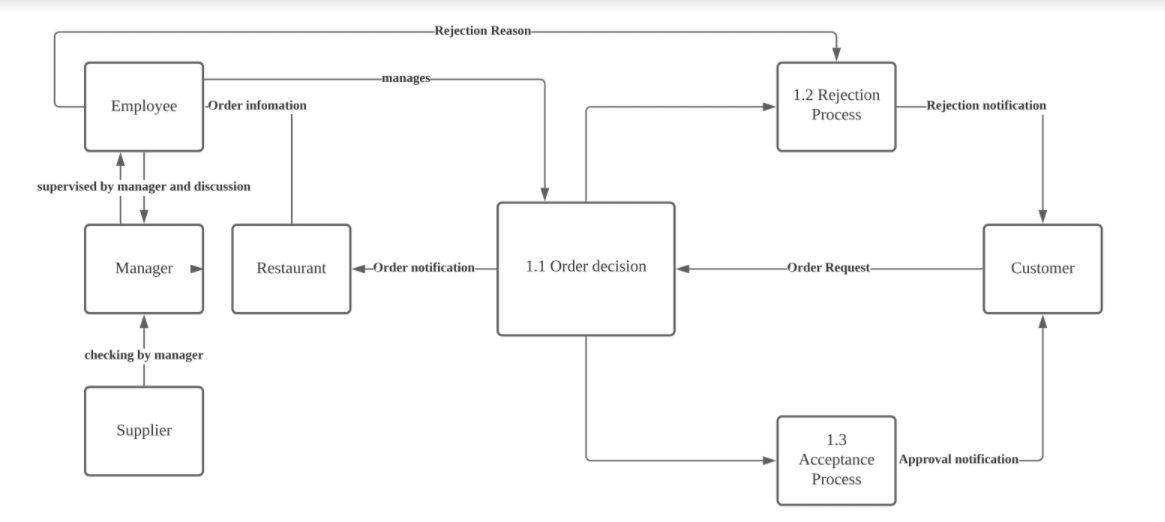
**Figure 6.1 Context Level Diagram:** The app user will operate the system by requesting an order, which leads to a message notification to the restaurant from the app notification system. Then the restaurant will accept or decline the order request, which notifies the user

## Level 0 Diagram



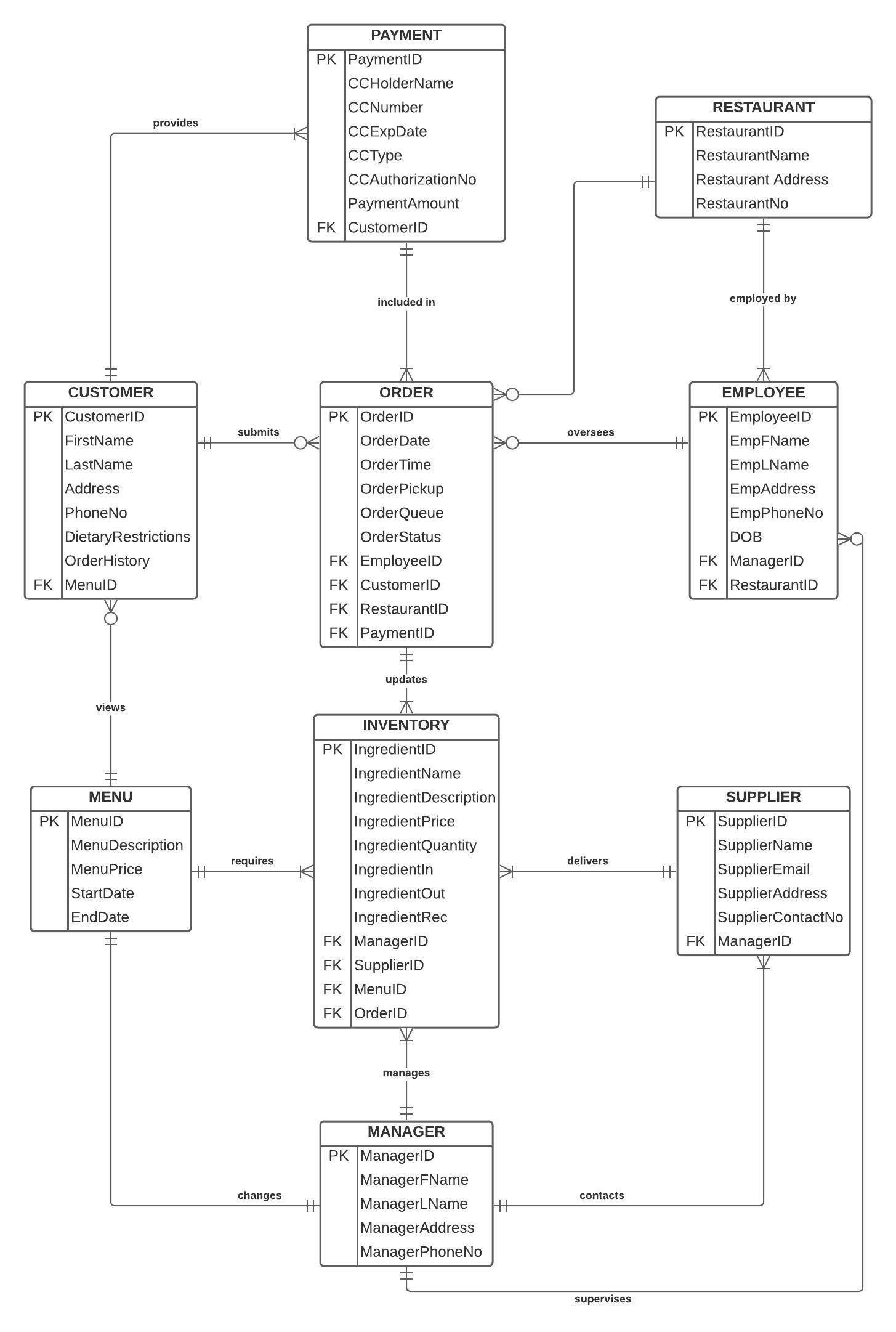
**Figure 6.2 Level 0 Diagram:** There are two basic structures from the user perspective and business perspective. Users will need all the basic information and a verified payment to make a successful request, and restaurants will need to go through different sources like employee, manager, supplier to meet the requirements for operation. After a decision is made from the restaurant, the notification system will send a message from the same information tunnel to the user.

## Level 1 Diagram



**Figure 6.3 Level 1 Diagram:** Level 1 Diagram is a deeper level breakdown about the decision making logical structure in the system. From the input of the user, the restaurant will make decisions from checking all the sources in the business system which helps them meet the requirements for the request. The decision will be either an acceptance or rejection. Either of the decisions will send a notification to the user for the next operation.

# VII. ERD Diagram

****

**Figure 7.1 Entity Relationship Diagram:** The ERD for Full Stock describes how its various entities relate to each other. At its core, the relationship between the ordering and automated inventory management system is a one-to-many relationship. Inventory levels are updated continuously as one or many orders are placed.

## VIII. CRUD

| **Entity Process** | **Customer** | **Employee** | **Manager** | **Supplier** | **Menu** | **Order** | **Payment** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Customer submits an order to the restaurant** | **CRUD** | **RUD** | **R** |  | **R** | **R** | **R** |
| **Customer provides payment information** | **CRUD** | **RU** | **RU** |  |  |  | **RUD** |
| **Employee oversees an order** | **R** | **RUD** | **RUD** |  |  | **R** | **R** |
| **Order updates inventory system** |  | **R** | **R** | **R** |  |  |  |
| **Manager manages inventory system** |  |  | **CRUD** | **RU** |  |  |  |
| **Supplier delivers and replenishes inventory system** |  |  | **RUD** | **RU** |  |  |  |
| **Manager changes menu** | **R** | **R** | **CRUD** |  | **RU** | **RU** |  |

**Figure 7.2 CRUD Diagram:** This diagram demonstrates the types of processing performed on the Full Stock data system for each function of the order and automated inventory process

1. https://marketrealist.com/2019/09/the-restaurant-industry-an-overview/#:~:text=A%20higher%20number%20of%20Americans,4%25%20of%20the%20US%20GDP. [↑](#footnote-ref-0)
2. <https://www.nrn.com/news/national-restaurant-association-report-pandemic-has-forced-100000-restaurant-closures-six> [↑](#footnote-ref-1)
3. <https://utilitynyc.com/blog/app-development-cost#:~:text=The%20average%20cost%20to%20make,feature%20apps%20cost%20%24150%2C000%20%E2%80%93%20%24250%2C000>. [↑](#footnote-ref-2)