## **Introduction**

The Panucci’s Pizza Order and Delivery system is a program that will suit the customers’ needs when using it to order. The program will have a menu, that will make it easy when placing an order. Upon checking out, you will be displayed a checkout screen, where you will be required to enter your billing details. If you are wanting delivery, you will be required to enter your delivery details (House, Apartment, Office, Zip Code, City, etc.). If you are wanting pickup, you will enter your Zip Code and City. After checking out, depending on if you chose pickup or delivery. You will be displayed a GPS or the nearest intersection and address.

## **High Level Description**

The software will communicate with the customer record and customer database using HTTPS and authorization. If a customer sends an order, the request will be sent through the database and alert the employee. The customer’s data and transaction history will be permanently stored in the customer database. An active order will be stored in temporary storage until payment has been complete. Then it will be moved into transaction history.

* **Detailed descriptions**
  + **Details of each functionality (functional requirement)**
    - Log In – Employees or customers will be prompted to log in. Upon input from user, the input will be sent to the authentication software to authenticate it is the user. Once confirmed, the user will be successfully logged in.
    - Check Schedule/Request Off – If an employee is logged in, pull up an interface exclusive to employees. This interface includes the ability to check the employee’s schedule and request time off from work. Customers cannot view this.
    - View Options/Menu – Regardless of login status, users may view the variety of options and the menu for Panucci’s Pizza Order and Delivery through this interface. This includes favorites, toppings, sizes, crust, etc...
    - Make Order/Purchase – Regardless of login status, users may order from Panucci’s Pizza Order and Delivery. This allows customers to customize their order to their liking. If logged in, users may use their previously stored information to place the order. Once an order has been placed, the software will communicate with the customer record and customer database using HTTPS and authorization. The request is sent through the database and alerts the employee after being authorized. The data will be permanently stored in the customer database. The active order will be stored temporarily until the payment has been successfully submitted where it will then be moved into transaction history.
  + **Details of nonfunctional requirements**
    - Performance requirements (response time, throughput, etc.)
      * Response time between authentication software and Panucci’s Pizza Order and Delivery system – The peak response time should be no greater than 5000ms. This will allow for users to flow through the process steadily and without interruptions. It is preferable to keep the average response time below 4000ms and the error rate low.
      * Response time between payment completion and Panucci’s Pizza Order and Delivery system – While the payment completion will depend on the card issuer or bank, the completion of the transaction should be fairly low. The peak response time should be no greater than 6000ms with the average response time preferably below 4000ms. The error rate should be exceptionally low to reduce any chances of overcharges, double payments, etc...
      * Throughput from customers to program- The ability to keep the throughput high without causing major delays is a must. During peak hours, if the performance drops, it lessens the customer content with the program and website, resulting in loss of these customers.
    - Additional attributes such as security, reliability, etc.
      * Security and reliability of authentication software – After verifying the user’s credentials, the software should allow the user to continue without error. The error rate should remain low and the security needs to be exceptionally high to prevent the leakage of any user’s personal data.
      * Security and reliability of database – The flow between the data base and the program should be fluent and quick. The error rate should also be low and the security should be exceptionally high to prevent a user’s personal data from being leaked.
      * Security and reliability of payment system – Once an order has been made, it is crucial for the payment system to be reliable and with miniscule amounts of errors if any at all. The error rate needs to be exceptionally low while the security remains extremely high to prevent a user’s payment data from being leaked.
  + **Interfaces, including user interfaces and system interfaces**
    - User interface
      * User interface will be simple to navigate with clearly labeled buttons and entry requirements
      * Objects within the user interface will be centered and properly aligned
      * The color scheme of the user interface will be as pleasant as possible
      * The appearance will be kept professional without any out of place images or fonts
    - System interface
      * Should determine whether a user is a customer or employee after receiving initial login information
      * Will send different information to user interface depending on whether the user is a customer or employee
      * Set up so as to always send the correct information to the user interface depending on the button pressed by the user
  + **Design constraints, standards, etc.**
    - Will be written using Java
    - User must have adequate hardware to run; if able to run Windows 10 or macOS Big Sur, will be able to run application
    - Windows 10 application will follow the [Windows design standards](https://docs.microsoft.com/en-us/windows/apps/design/basics/)
    - MacOS Big Sur application will follow the [Mac design standards](https://developer.apple.com/design/human-interface-guidelines/macos/overview/themes/)

## ​**​​​​​​Use Case Diagram**Diagram Description automatically generated

## ​​​​​​​

## **Data Flow** **Diagram**

Diagram

Description automatically generated