Requirements Documents
BrightLink Systems
SWE3312

Team Members:
Yeonnkuk Woo
Keilah Bonaby
Kevin Nolan

Table of Contents

TABLE OF CONTENTS

COVER PAGE **1**

TABLE OF CONTENTS **2**

REQUIREMNTS DEFINITION **3**

SECTION 2.1 ..... **4**

USE CASE DIAGRAM **5**

USE CASE FLOW OF EVENTS **6**

SECTION 3.2 ..... **7**

SECTION 3.3 ..... **8**

USE CASE DIAGRAM **9**

ER diagram ***10***

Class documentation ***11***

Subsection 3.3.c ***12***

Subsection 3.3.c ***13***

Subsection 3.3.c ***14***

Class diagram ***15***

State transition diagram ***16***

Requirements Definition

The purpose of this Requirements Document is to outline the data requirements and specifications for our client's pizza ordering & delivery system. In this section we will outline the nonnegotiable application requirements. There is a total of nine. 1). A Record for each new customer. 2). The option to proceed as a new or existing customer. 3). Account lookup via phone number. 4). GUI for existing customers record account. 5) Record of payment type. 6) Physical receipt to sign for pickup orders. 7) Electronic receipt for online orders. 8) Complete Restaurant Menu. 9) Employee sign-in for placing orders (GUI). The list below outlines additional details that will be implemented into each requirement to ensure consistency and completeness of the systems design.

1) A Record for each new customer.

- Interface reads input from the user and writes it to the data management system (DMS) to retrieve record accounts for returning customers.

2) The option to proceed as a new or existing customer.

- System link to database for stored account records

3) Account lookup via phone number.

- Efficient database query for stored accounts and employees

4) GUI for existing customers record account.

- Interface should display stored:

  First Name, Last Name, Phone, Address, Type of Charge Account, Delivery Notes.
- Order History (Receipt Preview)

5) Record of payment type.

- checks, cash or credit cards. (Visa/Mastercard).

6) Physical receipt to sign for pickup orders.

- Employee interface needs to have am Action listener for printing a receipt in reference to takeout orders. Receipts must contain:
  A signature line, payment type used, customer information, a list of items ordered, order type (Delivery or Pickup) and Amount due.

7) Electronic receipt for online orders.

- A consent prompt for account charge, payment type used, customer information, a list of items ordered, order type (Delivery or Pickup) and Amount due.
- Receipt Preview

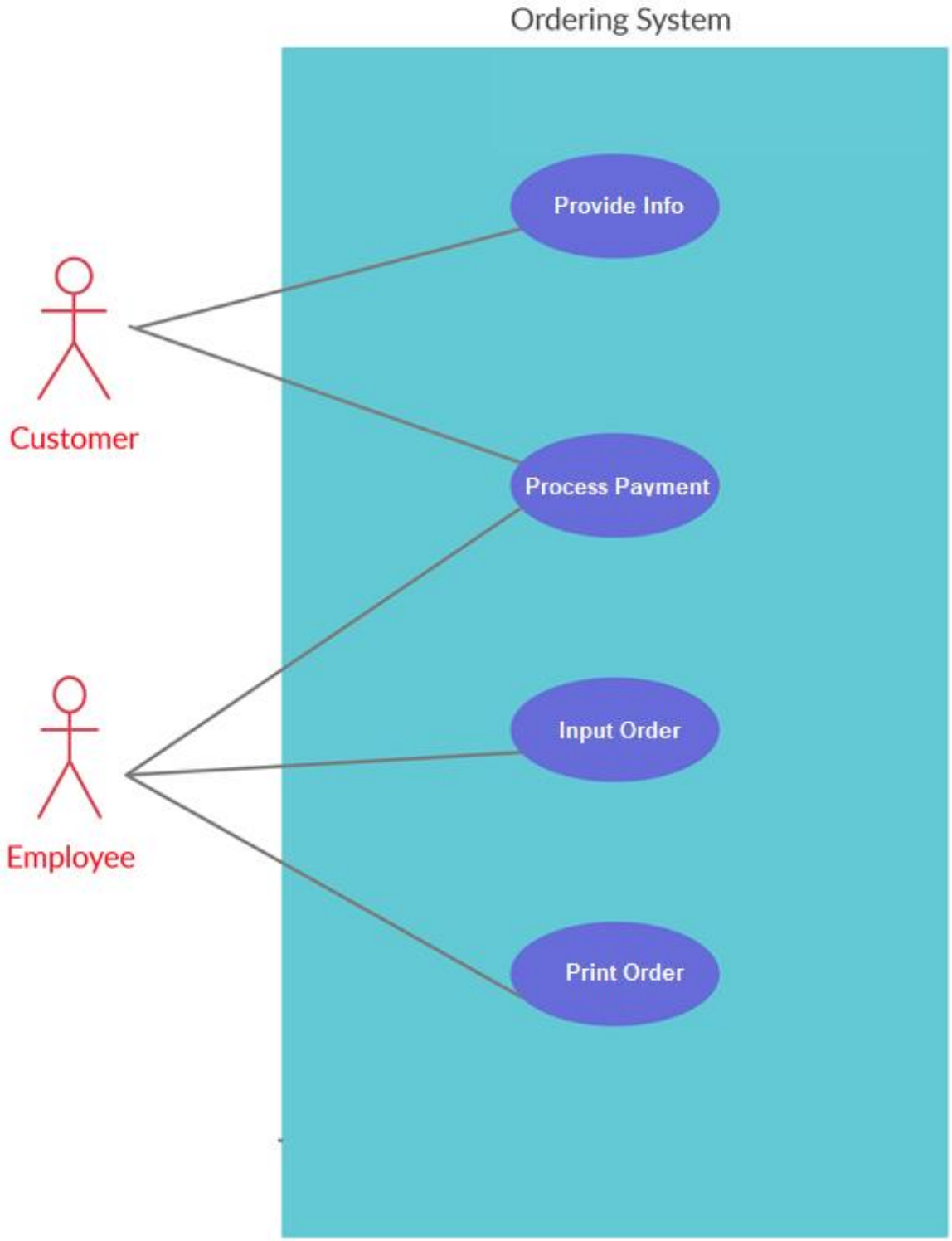
8) Complete Restaurant Menu.

- Full menu, efficient order form for fast checkout. Must have various sizes of pizzas, the typical types of toppings, various crust options deserts and beverages.

9) Employee sign-in for placing orders (GUI)

- System link to database to query for employee user accounts.
- An interface mirroring the customer online account retrieval and creation of order.
- POS (Point of Sale) ActionListener and Receipt Preview.

Use-Case Diagram



Use-Case Flow of Events

1.0 Flow of Events for the Provide Info use case

1.1 Preconditions

The employee is logged in to the system and on the main display screen. The customer has called in or walked up to the employee and is ready to order their food.

1.2 Normal Flow

The use case begins when the customer contacts employee to place their order. The customer provides their contact info to the employee who then accesses the customer database through the search customer options and then chooses delivery or pickup option.

- If customer is found in database, then S-1: Verify Customer subflow is activated.
- If customer is not found in database, then S-2: Add Customer subflow is activated.
- If customer is found and verified, then S-3: Start Order subflow is activated and use case ends.

1.3 Sub flows

S-1: Verify Customer

The employee searches the customer database and verifies the customer is in the database(E-1) and that their contact info is correct (E-2). The use case continues.

S-2: Add Customer

The employee accesses the customer database and inputs the customer contact info. The use case continues.

S-3: Start Order

After completing customer selection, the employee selects "Delivery" or "Carryout" option(E-3) on screen and clicks "Start Order" button and use case ends.

1.4 Exceptional Flows

E-1: The user tries to find a customer in the database that does not exist. The user can prompt customer for their info and add customer to database or terminate use case.
E-2: The user receives input from the customer that their info is incorrect in the record. The user can modify contents or terminate use case.
E-3: The user tries to choose an order option (Carryout or delivery) that is currently unavailable. The user can change to another option or terminate use case.

2.0 Flow of Events for the Input Order use case

2.1 Preconditions

Use Case 1.0 must be initiated with the customer before this use case can commence.

2.2 Normal Flow

The use case begins after the customer has provided their contact info in use case 1.0 to the employee and chosen Delivery or Pickup. The employee then accesses the menu screen.

- Customer adds pizza to order, then S-1: Add Pizza subflow is activated.
- Customer adds appetizer to order, then S-2: Add Appetizer subflow is activated.
- Customer adds drink to order, then S-3: Add Drink subflow is activated.
- Customer adds side to order, then S-4: Add Side subflow is activated.
- Customer completes giving order to employee. Use case terminates.

2.3 Sub flows

S-1: Add Pizza

The employee chooses the Pizza menu option and asks the customer for the size(E-1), toppings(E-2) and crust type (E-3). The use case continues.

S-2: Add Appetizer

The employee chooses the Appetizer menu option and asks the customer for the appetizer type(E-4), and any specific notes. The use case continues.

S-3: Add Drink

The employee chooses the Drink menu option and asks the customer for the drink type(E-5). The use case continues.

S-4: Add Side Item

The employee prompts the customer for any side items they may want (E-6). The use case continues.

2.4 Exceptional Flows

E-1: The user tries to add a pizza size that is not available. The user can choose a different size or terminate use case.

E-2: The user tries to add a topping that is not available. The user can modify toppings or terminate use case.

E-3: The user tries to choose a crust type that is currently unavailable. The user can change to another option or terminate use case.

E-4: The user tries to choose an appetizer option that is currently unavailable. The user can change to another menu option or terminate use case.

E-5: The user tries to choose a drink option that is currently unavailable. The user can change to another option or terminate use case.

E-6: The user tries to choose a side item that is currently unavailable. The user can change to another option or terminate use case.

3.0 Flow of Events for the Process Payment use case

3.1 Preconditions

The "Input Order" use case must execute by the Employee actor first.

3.2 Normal Flow

The use case begins after the employee has input the customer's order into the system in use case 2.0. The customer then provides the employee with their form of payment.

- If customer pays via Credit Card, then S-1: Process Credit payment is activated.
- If customer pays via Check, then S-2: Process Check payment is activated.
- If customer pays via Cash, then S-3: Process Cash payment is activated.

3.3 Sub flows

S-1: Process Credit Payment
The system displays a form with boxes for credit card #, expiration and cvv code for input. If an input is invalid, then E-1 will activate and if the payment is declined then E-2 is activated. The use case terminates.

S-2: Process Check Payment
The system displays a form with boxes for check #, bank name and account number. If an input is invalid then E-1 will activate. The use case terminates.

S-3: Process Cash Payment
The system displays a form with a box asking for total cash given. It will then display the change to be given to the customer. If the amount is incorrect then E-3 will activate. The use case terminates.

3.4 Exceptional Flows

E-1: The user tries to input a number that is not of valid length or formatted incorrectly. The user can correct the number or terminate the use case.
E-2: The user tries to run a credit card payment that is declined by the payment processing system. The user can input a different card number to process or terminate the use case.
E-3: The user tries to input a value in the cash box that is lower than the total of the order. The user can enter a different amount or terminate the use case.

4.0 Flow of Events for the Print Order use case

4.1 Preconditions

The "Process Payment" use case must execute by the Employee actor first.

4.2 Normal Flow

The use case begins after the employee has processed the payment for the customer's order into the system in use case 3.0. The employee then prompts the system to print a receipt.

- If customer pays via Credit Card, then S-1: Print Credit receipt is activated.
- If customer pays via Check, then S-2: Print Check receipt is activated.
- If customer pays via Cash, then S-3: Print Cash receipt is activated.

4.3 Sub flows

S-1: Print Credit Receipt

The system automatically prints the receipt with Tip and Signature lines for customer to sign. The employee inputs the tip amount into system (if any) (E-1) and gives customer a copy. A separate receipt can be printed with listing of customer order items if requested by customer. E-2 and E-3 can activate on this use case as well. The use case terminates.

S-2: Print Credit Receipt

The system prints out a listing of the order items and on the bottom prints the Check number and time processed with amount. E-2 and E-3 can activate as well. The use case terminates.

S-3: Print Cash Receipt

The system prints out a listing of the order items and on the bottom prints out the amount of cash given, and the change given to customer. E-2 and E-3 can activate as well. The use case terminates.

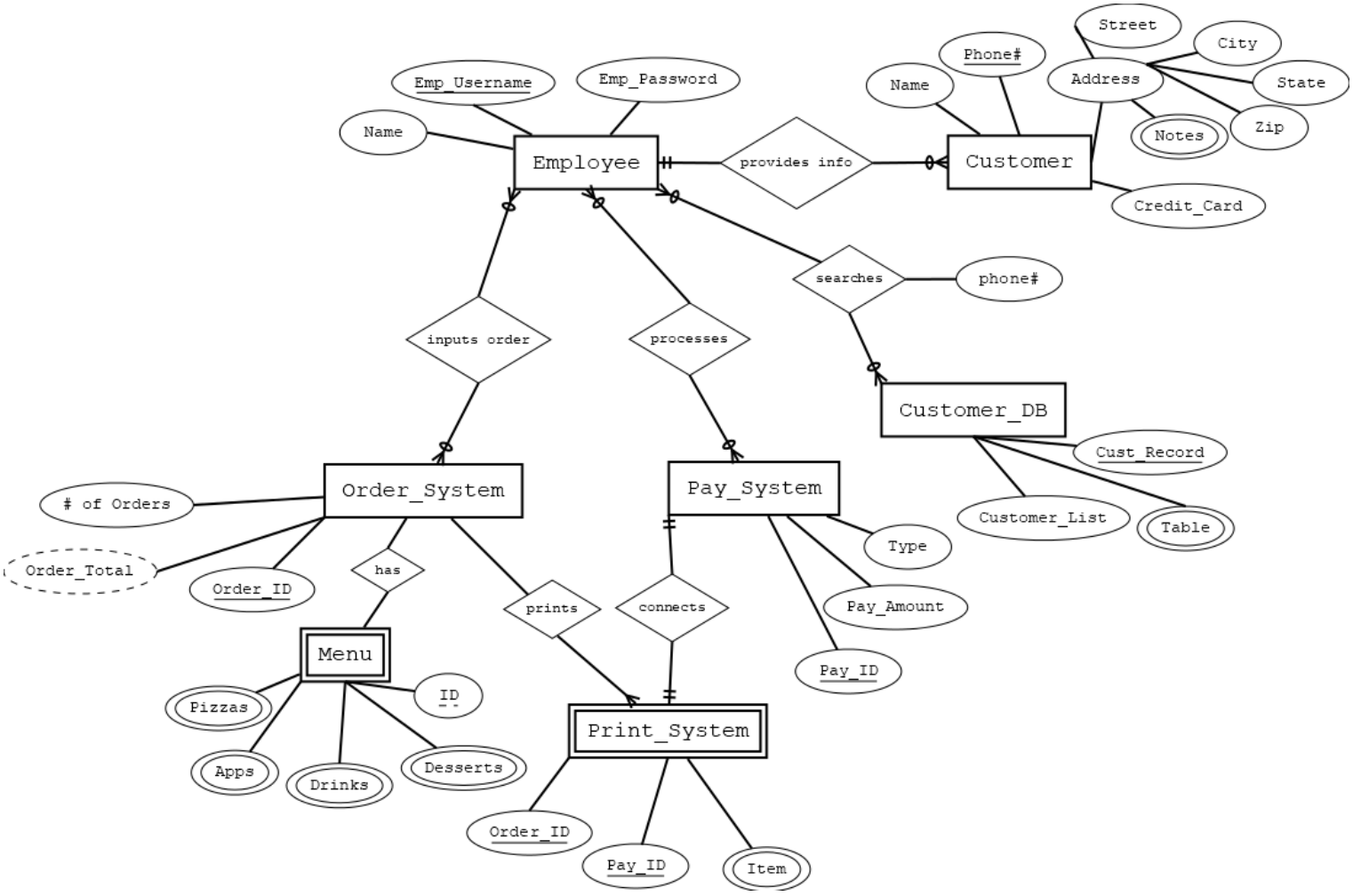
4.4 Exceptional Flows

E-1: The user tries to input a number that is not of valid length or formatted incorrectly. The user can correct the number or terminate the use case.

E-2: The receipt printer output device is unable to print out a receipt due to lack of paper. The employee can add more paper or terminate use case.

E-3: The receipt printer output device is unable to print due to low ink. The user may add more ink to the printer or terminate use case.

E-R Diagram


Street
City
Phone#
Address
State
Name
Zip
Notes
Emp_Username
Emp_Password
Name
Employee
provides info
Customer
Credit_Card
searches
phone#
inputs order
processes
Customer_DB
Cust_Record
Order_System
Pay_System
# of Orders
Customer_List
Table
Type
Order_Total
Order_ID
has
prints
connects
Pay_Amount
Pay_ID
Menu
ID
Pizzas
Print_System
Apps
Drinks
Desserts
Order_ID
Pay_ID
Item

Class documentation

Class name: Employee

Category: data

Operation Name: n/a

Public member of: Employee

// stores Employee data

Name is not NULL;

Employee_Username is not NULL;

Employee_Passward is not NULL;

Private Attributes: none

Public Attributes: Name; Employee_Username; Employee_Passward

Class name : Customer

Category : data

Operation Name: n/a

Public member of: Customer

Documentation:

// stores customer data

Name is not NULL;

Phone# is not NULL;

Address is not NULL;

Street is not NULL;

City is not NULL;

Zip is not NULL;

Private Attributes: none

Public Attributes: Credit_Card; Name; Phone#; Address; Street; City; Zip;

State; Note;

Class name : Customer_DB

Category : data

Public member of: Customer

// stores customer data from previous orders

Customer_List is not NULL;

Cust_Record is not NULL;

Table is not NULL;

Private Attributes: none

Public Attributes: Customer_List; Cust_Record; Table;

Class name : Pay_System

Category : data

Public member of: Pay_System

Documentation:

//stores billing information

Pay_ID is not NULL;

Pay_Amount is not NULL;

Private Attributes: none

Public Attributes: Type; Pay_ID; Pay_Amount;

Class name : Order_System

Category : data

Public member of: Order_System

Documentation:

// stores order information

# of Orders is not NULL;

Order_Total is not NULL;

Order_ID is not NULL;

Private Attributes: none

Public Attributes: # of Orders; Order_Total; Order_ID;

Class name : Print_System

Category : data

Public member of: Order_System, Pay_System Documentation:

// Prints order/billing information

Pay_ID is not NULL;

Order_ID is not NULL;

Item is not NULL;

Private Attributes: none

Public Attributes: Pay_ID; Order_ID; Item;

Class name: Menu

Category : data

Public member of: Menu

Documentation:

// stores/displays menu info

Pizzas is not NULL;

Apps is not NULL;

Drinks is not NULL;

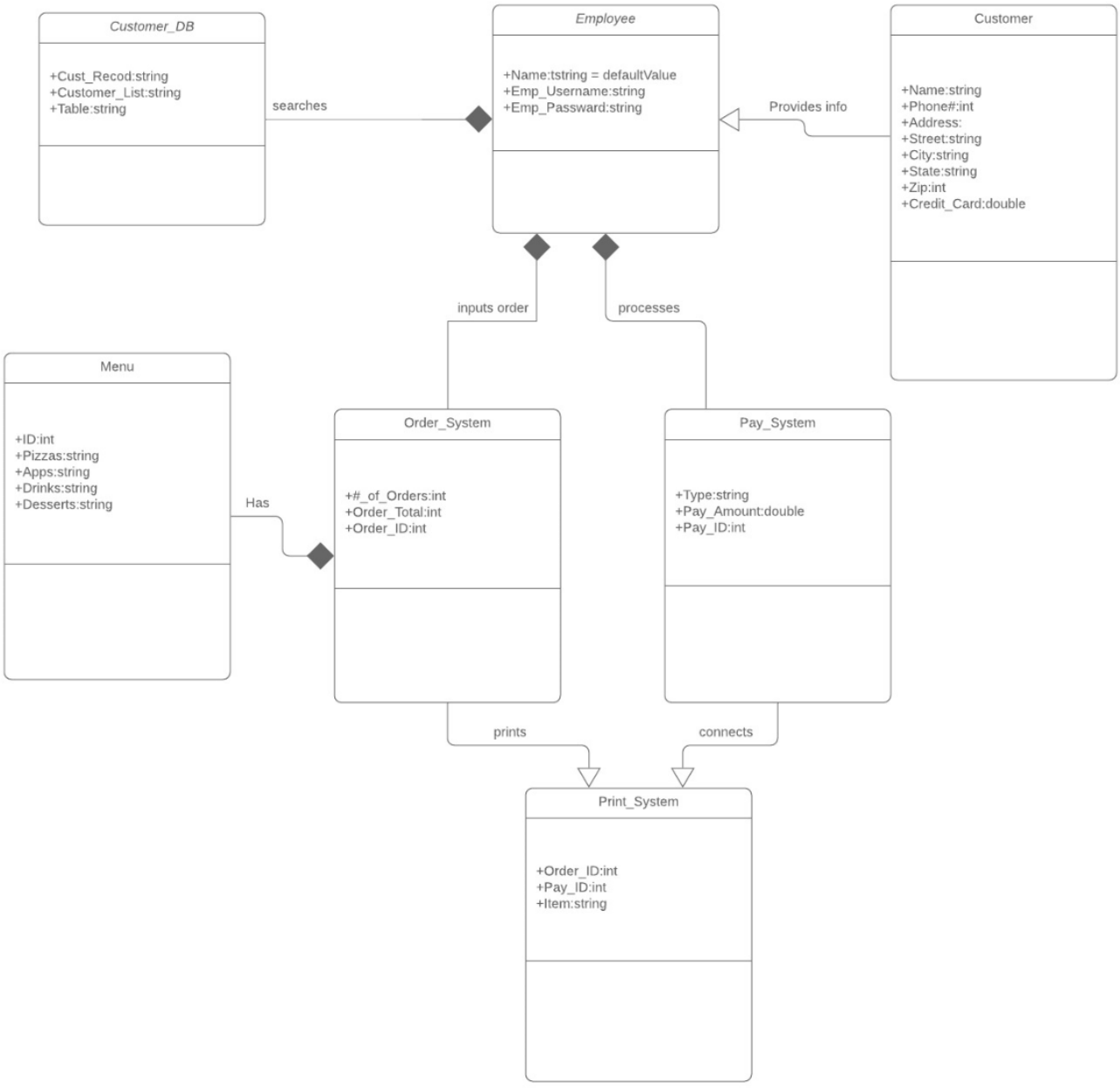
Desserts is not NULL;

ID is not NULL; // can be calculated by previous information

Private Attributes: none

Public Attributes: Pizzas; Apps; Drinks; Desserts; ID;

Class diagram



State Transition Diagram