

# Automated Delivery System

---

*Final report for Database Management System*  
*University of Houston – Clear Lake*

*By,*

Vaibhav R. Modi (2006925)

Juhi J. Bahrani (1996407)

## **Table of Contents**

1. Description
2. The Working of the system
3. Objective of the project
4. System Design
5. System's data requirement
6. Entity Set
7. Relationship Set
8. Work flow Diagram
9. Entity-Relation Diagram
10. Reducing the ER-Diagram to Empty Tables
11. Relational Schema
12. The Schema Diagram
13. Comparison and Conclusion
14. Contributions

## **1. Description**

This project aims at developing an automated delivery system that can be used to send and receive objects or goods from one place to another. It will have two objectives – Anyone can send any courier to any other people those are within a city and uses this application, and one can also buy things online from local stores. It will be beneficial to the local shops and their warehouses to deliver things with ease to the right customers and also for the people who would want to send important documents with urgency right from their doorsteps to any destination.

## **2. The working of the system**

There will be multiple real-life entities involved in this project to make it a successful one. Typically, the work of automated delivery should be done by robots, but in this project, we will make a prototype considering delivery person that can travel within the city and this application will involve a database to work with multiple different entities and sharing the data, and MySQL data manipulation language to retrieve and store, and organize the database. The entities will be shops, warehouses, customers, delivery person (team of delivery people). The customer can send any documents or couriers to any other customer using this application. The delivery people will be assigned to deliver things from one place to another, depending on their availability and their location – people closest to the source of pick up will be prioritized. The local shop keepers can deliver things from their stores or their warehouses to anyone who uses this system. The customers will have an option to buy things online from any store nearer to them. The system will be able to handle multiple deliveries from one person or multiple deliveries from multiple people to multiple people – users won't have to worry about it, the delivery people will have an organized and optimized information which will help them handle complex delivery service. Users will be able to select recipients by looking up their name and profile instead of searching for the recipient's locations. There will also be a feature of tracking the order. The location and details of the courier in transit will be uploaded to the database and will be accessible by the end users – both the sender and the recipient. It will also have a simple user interface, so customers don't have to worry about selecting the shops near-by, it will be user friendly. Looking at the security aspect of the application, a barcode or a unique number generation will be used that will be shared to the sender and recipients to secure the

delivery. The delivery acknowledgement will be sent as a push notification to confirm the delivery and the delivery history will be available to each user.

### 3. Objective of the project

Here we are going to talk about why we chose this project and how it will benefit people. We chose to work on this project to address the issue of inefficient and lesser availability of applications in the market that is mainly focused on local people and shops. There is a need to improvise the system of delivering information and things. Due to the busy lives of people and given this pandemic, most employees and businesses have been forced to work/run from home and there is a need of an application which can provide immediate delivery of documents and things from one place to another remote place. As it can be inconvenient for employee in a company to deliver things frequently to their colleagues or friends – they would have to drive themselves to deliver it (things which cannot be sent in an email). This is where our application comes in use – people can deliver things easily and quickly right from their doorsteps. It can also help local shops and businesses which are not affiliated to bigger companies like Amazon, Flipkart and have lesser customers visiting their shops due to the pandemic or might just want to expand their businesses. They can deliver goods to any destination in the city and the customers can buy things from any shops without stepping out of their home. This project addresses two major problems – 1. Difficulties in quickly mailing documents to someone in the same city (going to the postal office might not be the solution here) 2. Customers and shopkeepers having difficulties to buy products and deliver products respectively. As this is a prototype project which will tackle real world conditions, it will have a consideration of implementing an actual robot in the future.

### 4. System Design

The Automated Delivery System will be developed using the frontend technologies - **JavaScript, HTML, and CSS**, for the backend – **Flask microframework – Python**, for the database – **MySQL** will be used for its fast and robust built, providing a leading open source RDBMS. It will follow **three-tier architecture** where client will communicate with server side application which will then communicate with the database. This will increase the security and

integrity of the data stored and will provide an organized way of accessing and manipulating them.

It will contain an admin side which will be a super user that can moderate most of the entity such as Customer, Shops, deliverybot, deliverieschedules, etc. The admin has an important role in the management of the whole system.

The functionality of user or the customer will include secure login feature, followed by a whole variety of product browser and shops filtering. They can order items online from any shops listed in the system. The order items list will have delivery ID and will be picked up by the automation bot from the shop and will deliver the product to the customer in timely manner, plus it will include the tracking functionality which will track the location of the bot. If there are multiple deliveries scheduled, it will update the tracking and delivery ID, which can be accessed by the customers as well as the shopkeepers or the suppliers. Along with the items bought online, there is an additional functionality offered to the users where they can send couriers and packages to the other users. It uses the same delivery system with the delivery bot and the tracking system to pick-up packages from one user's doorstep and deliver to another user.

The shopkeepers can keep track of their products and their quantities that is up for selling. They can also update the prices and the details of the products. When the customers orders a list of products, they get notified, and also the information on the bot for the pickup of products is scheduled. They have the tracking features available too, just like the customers, and in addition to that, they also get the acknowledgement of the delivery of their products. There will be domains constraints too to check for the availability of the quantity of the product. If the quantity reaches to one, customers wont be able to buy it and the shops will be notified about it.

The system will check for the availability of the delivery bots and when they are available, the new deliveries will be scheduled whenever an order is placed and the bots will have the information of the source and the destination of the pick up and deliveries. After picking up the order, It will update the tracking system and the corresponding related entities will be updated with it. After the deliveries, it will send an acknowledgement to the customers as well as the shops. After a successful delivery, it will change its status of availability from 0 to 1 or no to yes. Then there will be options for the customers to provide reviews and feedbacks of the delivery. The feedback will have its own primary key and will be associated with customers and the delivery ID which will enable to get feedback on the shops, products or the deliveries.

The payment feature will be used by the customers to make payments for their orders and will also have their payment details stored in the history to reuse them for future payments or to keep a track of all the payments. The history of deliveries and orders will also be saved in the respective entities which can be accessed by respective entities such as customers and shops, along with the bot storing all the history of the deliveries in the database.

## 5. System Requirements

**Software:** Modern Web Browser – Chrome, Firefox, Edge

**OS:** Windows/Linux/Mac OS

**Frontend:** HTML, CSS, JavaScript

**Backend:** Flask - Python

**Database:** MySQL

### Hardware Requirements:

**Processor:** Intel dual core, i5–i7

**RAM:** 4 GB

**Hard Disk:** 20 GB

## 6. Entity Set Designation

### Strong Entities:

- Shops(ShopId, name, street, city, zip)
- Products(ProductId, Pname, quantity, price)
- Delivery(DeliveryId, time, DeliveryType, source, destination)
- Customer(CustomerId, name, street, city, zip)
- Tracking(TrackingId, status, location)

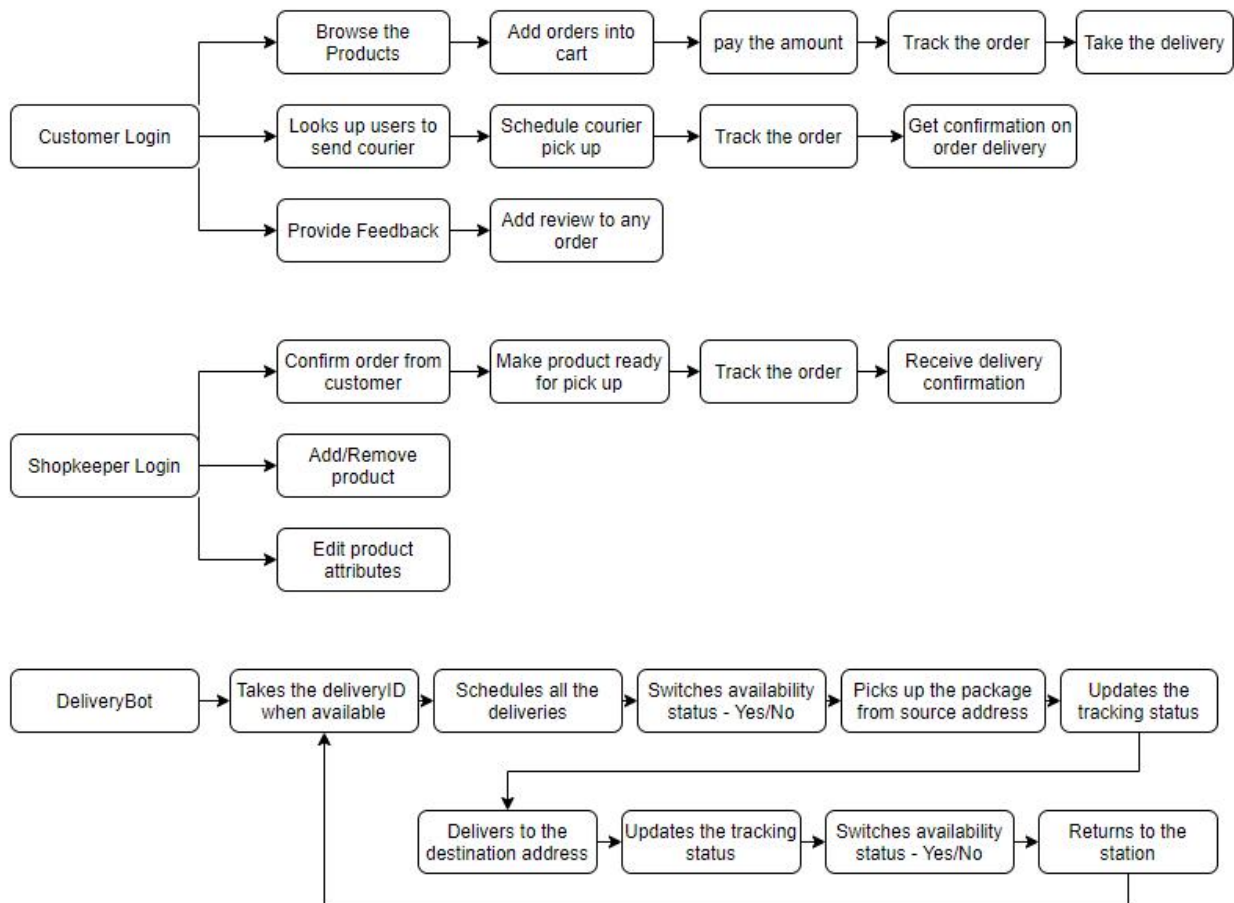
### Weak Entities:

- Payment(PaymentId, CustomerId, account)
- Feedback(DeliveryId, FID, comment)
- CustLogin(CustomerId, Username, Password)
- ShopsLogin(Username, Password, ShopId)

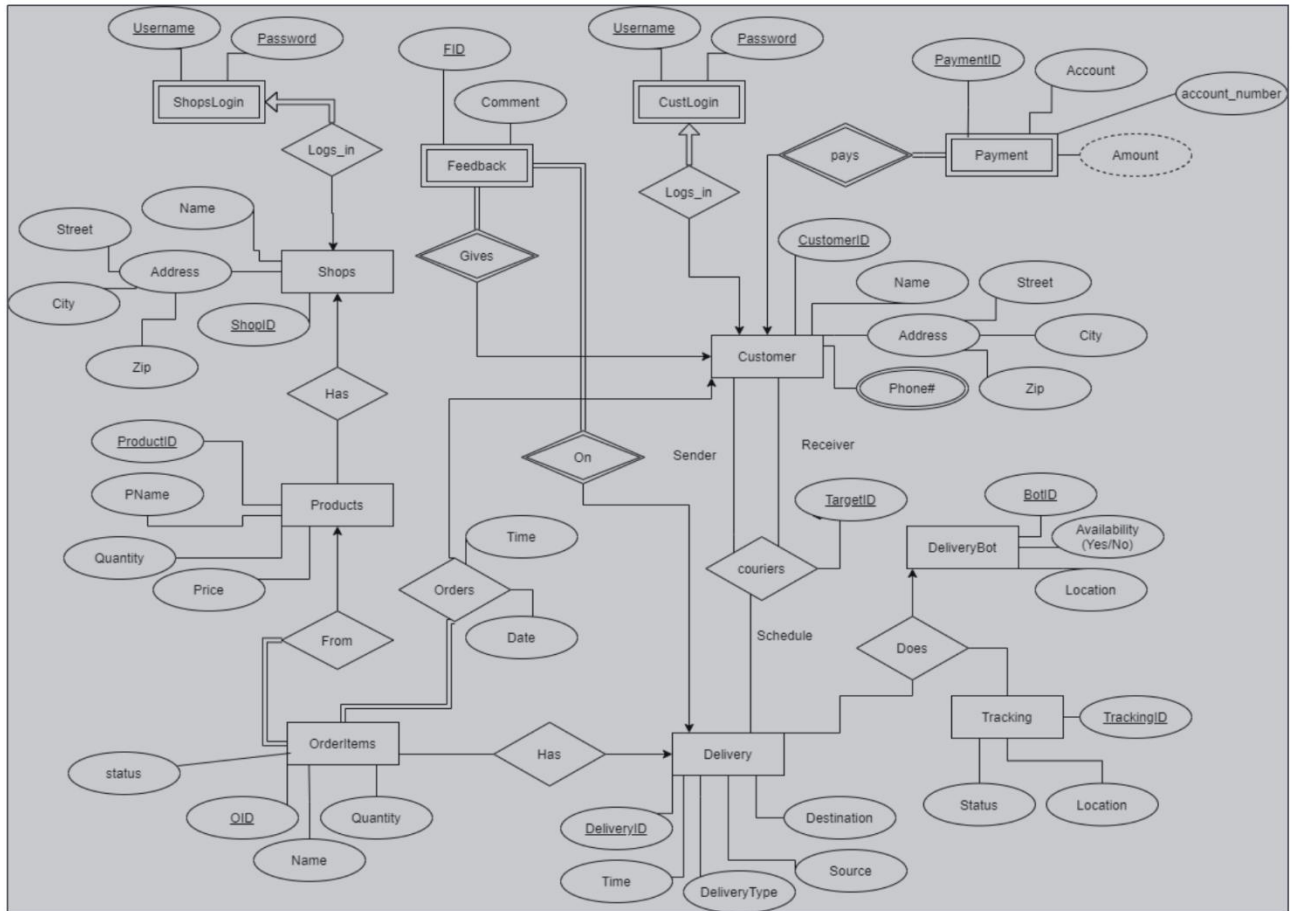
## 7. Relationship Set Designation

- Products-shops(ShopId, ProductId)
- OrderItems-delivery(OID, DeliveryId)
- DeliverySchedule(RID, DeliveryId, TrackingId)
- Couriers(CustomerId, DeliveryId)
- DeliveryBot(BotId availability, location)

## 8. Work flow diagram



## 9. Entity - Relation Diagram



Assumptions:

- (1) A customer can give multiple feedbacks but a feedback will be associated with only one customer.
- (2) A customer can order multiple items from products.
- (3) Customer can schedule a package delivery only to another customer who has an user account in the System.
- (4) An order can be associated with only one customer.
- (5) There can be multiple tracking ID for the same DeliveryID.
- (6) Shops will have distinct products in the products entity but can have multiple items in quantity.



## 10. Reducing ER Diagram to Empty Tables

1) Shops:

ShopId	name	street	city	zip

2) Products:

ProductId	Pname	quantity	price

3) OrderItems:

OID	customerId	productId	name	quantity	time	date	status

4) Delivery:

DeliveryId	time	DeliveryType	source	destination

5) Customer:

CustomerId	name	street	city	zip

6) Phone-customer:

Phone	CustomerId

7) Tracking:

TrackingId	status	location

8) Payment:

PaymentId	CustomerId	account	account_number

9) Feedback:

DeliveryId	FID	comment

10) CustLogin:

CustomerId	Username	Password

11) ShopsLogin:

ShopId	Username	Password

12) Product-Shops:

ShopId	ProductId

13) OrderItems-delivery:

OID	DeliveryId

14) DeliverySchedule:

RID	DeliveryId	TrackingId

15) Couriers:

CustomerId	DeliveryId	TargetID

16) DeliveryBot:

BotId	availability	location

17) Botlogin:

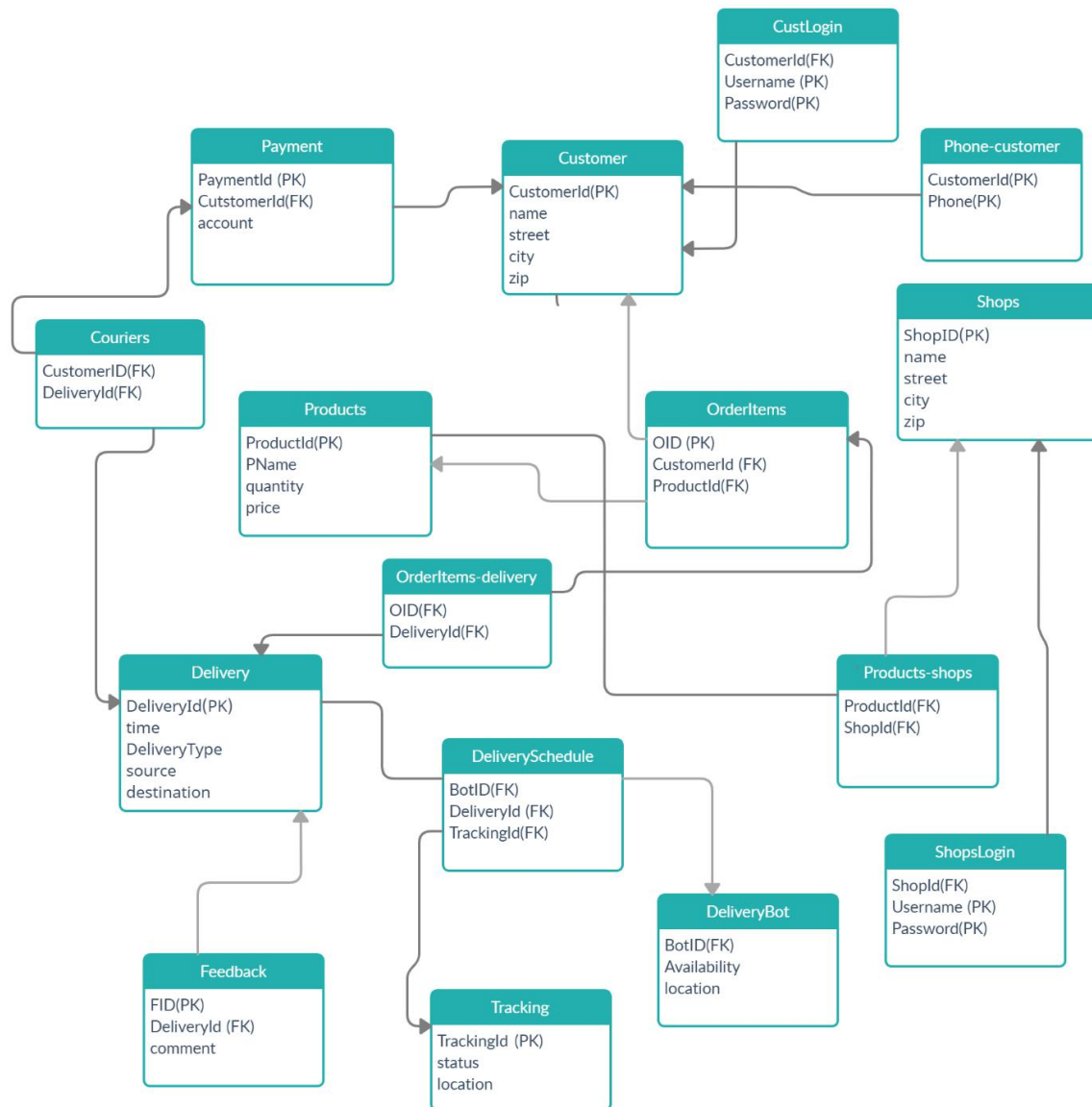
BotId	username	password

## 11. Relation Schema Diagram:

- a. Shops (ShopId, name, street, city, zip)
- b. Products (ProductId, Pname, quantity, price)
- c. OrderItems (OID, customerId, ProductId, name, quantity, time, date, status)
- d. Delivery (DeliveryId, time, DeliveryType, source, destination)
- e. Customer (CustomerId, name, street, city, zip)
- f. Phone-customer (Phone, CustomerId)
- g. Tracking (TrackingId, status, location)
- h. Payment (PaymentId, CustomerId, amount, account\_number)
- i. Feedback (DeliveryId, FID, comment)
- j. CustLogin (CustomerId, Username, Password)
- k. ShopsLogin (Username, Password, ShopId)

- l. Products-shops (ShopId, ProductId)
- m. OrderItems-delivery (OID, DeliveryId, status)
- n. DeliverySchedule (RID, DeliveryId, TrackingId)
- o. Couriers (CustomerId, DeliveryId, TargetID)
- p. DeliveryBot (BotId, availability, location)
- q. BotLogin(BotId, username, password)

## 12. The Schema Diagram



### **13. Comparison and Conclusion**

The automated system focuses on reducing the human labor and improving the efficiency of delivering the important things with lower costs. When we compare it with a regular delivery system, it has many benefits such as lower shipping and labor costs – Humans can be more expensive and are prone to errors, thus implementing a robust system that can do all the delivery optimization and include automation, it will provide better results. There are delivering applications such as Amazon, which enforces heavy labor on their employees which will be reduced by automating the delivery process. And an addition of the feature of pick-up and delivering documents within a city, with delivery not taking more than 2-3 hours (duration may vary on external factors like road block and traffic), it will be beneficial to many people who work from home and needs instant delivery. We hope to address as many real-world problems as we can with this project. As this is a prototype project which will tackle real world conditions, it will have a consideration of implementing an actual robot in the future.

This automated delivery system solves problem of large human forces that are required for deliveries who are prone to human error and are less efficient due to human limitations. This is where this system comes to rescue, it replaces the humans with the automated bots that can do the transportation and delivery of things with ease, quickly and with less errors.

One another place where such a system can be used is a company to deliver things frequently to their colleagues or friends – they would have to drive themselves to deliver it (things which cannot be sent in an email) as due to this time of pandemic, people are going totally “work-from-home”. This is where this proposed system comes in use – people can deliver things easily and quickly right from their doorsteps. It can also help local retailers to get their products out quickly to people to boost their sales and save their financials that is spent on people for delivering goods.

As this is a prototype project which will tackle real world conditions, we don't have actual working bots but we have simulated the activities and data that a bot would require to do and have. This system aims at upgrading the delivery system to go fully automated in the near future where it will still have time to grow and adapt by the society.

## **14. Contributions**

Vaibhav Modi: The front end part that includes developing pages for customers include shopping lists, checkouts, payment, delivery scheduling, tracking, and page view of the bot (updating delivery status, switching availability). The back-end logic for the above-mentioned pages is developed alongside by me.

Juhi Bahrani: The front end part which includes developing pages like the home page for shops, history of the products sold page, the updating/deleting/adding of products list was done by me. Also, its back-end logic was added to communicate with the database.

Contribution in the system design, the project ideas, writing SQL queries, preparing the ER diagram and also the Schema diagram was discussed and done by both of us.