**ParcelWay Project Report**

Parcel delivery database management system.

horizontal line

# 

*Figure 1. “Parcel delivery” image, taken from* [*Vector Stock*](https://www.vectorstock.com/royalty-free-vector/courier-parcel-background-delivery-service-van-vector-34496494)*.*

4th-year student in Software Design with Artificial Intelligence in Cloud

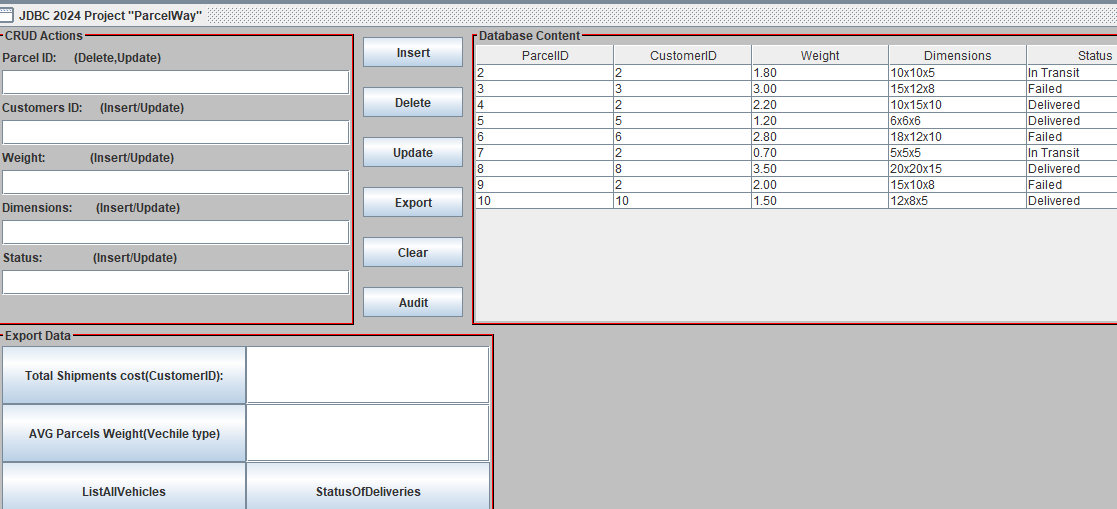
**Ivan Lapickij**

# 

# 

# Introduction

A database management system is essential for every business. To demonstrate my knowledge and skills in database management, I developed **ParcelWay**, a database management system designed for a parcel delivery company that is ease to use. In the following sections, I will introduce the workflow and implemented functionalities.



*Figure 2. “ParcelWay DMS GUI” image.*

# About

ParcelWay frontend - UI was provided & created with the JAVA language and Eclipse framework. Then, MySql was used within the MySql workbench to create backend database queries—functionality.

The database contains 4 tables:

* **Customers** - CustomerID, FullName, Address, Email, Phone, ShippingCostPP
* **Employees** - EmployeeID, FullName, Position, Phone, Email
* **Parcels**  - ParcelID, CustomerID, Weight, Dimensions, Status
* **Deliveries**  - DeliveryID, ParcelID, EmployeeID, Vehicle, RouteDetails, DeliveryTime

Each table contains at least 10 rows.

## Workflow

I started this project by implementing basic CRUD queries - Create, Read, Update, and Delete data for table Parcels, which we can see from the content displayed in “Figure 2”.

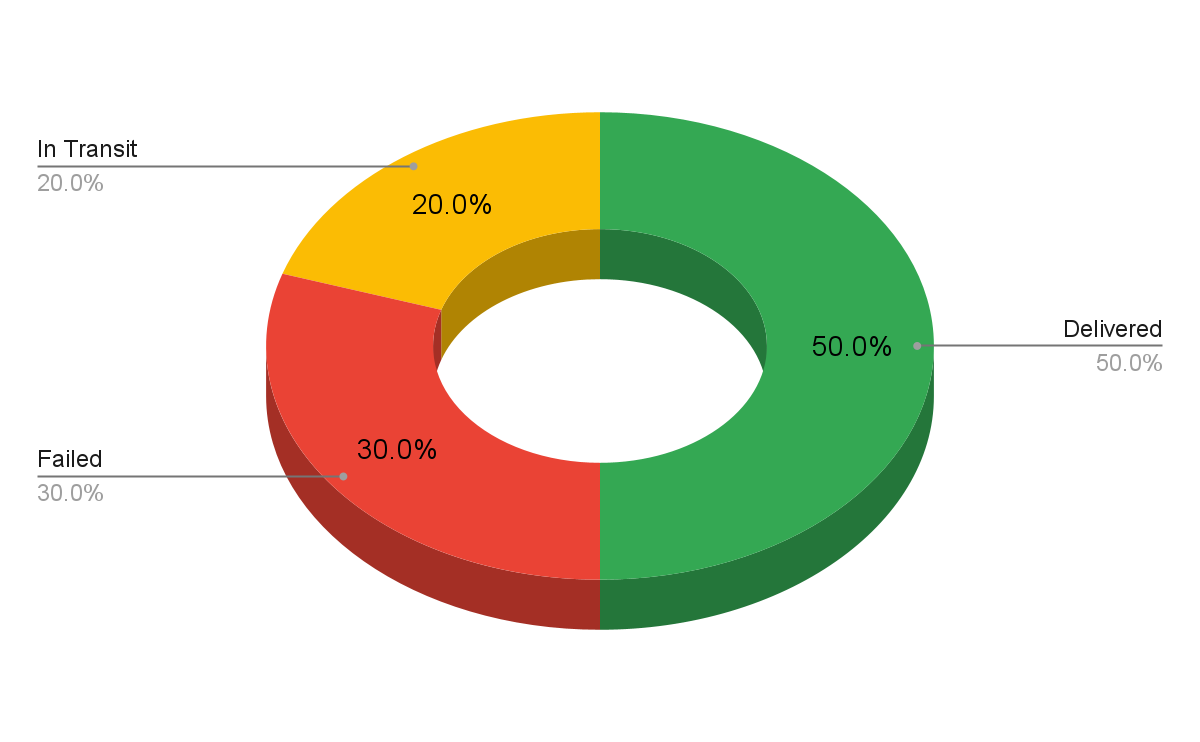
On the left side there is a textfield for user input insert based on desirable functionality etc. - Delete would require only ParcelID.

There are more buttons like:

* **Export** - that exports the current table of Parcels displayed on the right side of DMS.
* **Clear** - Clears user input in the CRUD Actions display on the top left side.
* **Audit** - records all deleted parcels to audit table “Parcels\_audit”, then recorded to CSV(Comma separated values) file MyOutput.csv

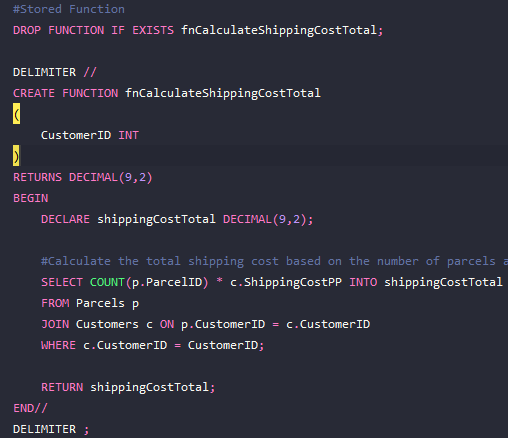
Next, at the bottom left we can see more complex queries and functions:

* **Status of deliveries** - records the status of total parcels.



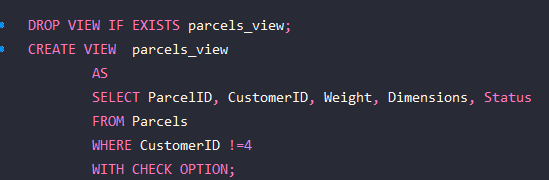
*Figure 3. Status of deliveries diagram.*

* **Total shipments cost** - based on user-inserted CustomerID on-click counts how many shipments the customer made and records in CSV file.
* **AVG parcels weight** - based on user-inserted vehicle type on-click counts average parcels weight carried on the vehicle.
* **List all vehicles** - on-click records to CSV file distinct types of vehicles the company uses.
* **fnCalculateShippingCostTotal** - stored function that calculates the total Shipping cost for a particular customer based on user input.



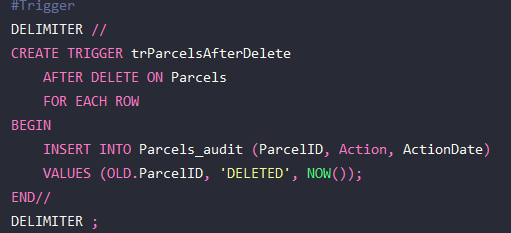
*Figure 4. A piece of code illustrates the creation of the stored function “*fnCalculateShippingCostTotal *”.*

Behind the scenes:

* **parcels\_view** - simulating real industry requirements I created a view that allows secure custom Customer data from being manipulated, etc. Updated or Deleted. In this case, a customer that is outside of the view has CustomerID 4.

*Figure 5. A piece of code illustrates the creation of View “parcels View”.*

* **trParcelsAfterDelete**- the trigger that activates every time a parcel is being deleted and records ParcelID, Action, ActionDate. Where Action stands for action taken (deleted) and Action date records the time of deletion.



*Figure 6. A piece of code illustrates the creation of View “parcels View”.*

## 

## 

*Figure 7.Image taken from*  [*EssayMaster*](https://www.essaymaster.com/how-to-write-essay-college-application/personal-statement-conclusion)

*.*

## Conclusion

* Implementation of a trigger to log deletion actions ensures that deleted records are tracked effectively. This enhances process traceability, which improves the quality of business services and facilitates better maintenance.
* Created View - “parcels\_view” secures custom data by preventing users from viewing, updating, or deleting.
* Exporting data in CSV allows data visualization and tracking.
* Basic CRUD allows users to interact and maintain the database as they please.
* Stored functions offer reusable, easy maintenance code.
* A relational database ensures lower error occurrence.
* Adding more functionality to the application increases its value and business efficiency.

## References

[ParcelWay on GIT](https://github.com/IvanLapickij/ParcelWay)

[ParcelWay Demo Video](https://go.screenpal.com/watch/cZXb2AnVFDD)

[MyOutput](https://docs.google.com/spreadsheets/d/1s3Fs_6YbNQZpGt6r8NVaRYnQQDL9pjp0Tnc82WwCcn0/edit?gid=0#gid=0)