# USE CASE STUDY REPORT

## Title : Hands-Me-Down (A Recommerce platform)

**Group No.:** 13
**Student Names:** Fathima Salim and Pratiksha Pradhan

## Executive Summary:

Nowadays, buying used apparel is a common trend among young people since it minimizes the need for brand-new clothing, which in turn reduces the demand for resources and the harm it causes to the environment. The resale industry is far less damaging to the environment and one's pockets, than the new clothing industry.

During peak migration periods, when one might be in need of many necessary products, one might find them to be out of stock at the popular stores. In this case, one can turn to re-commerce platforms like Hand Me Downs. This platform can connect those moving into new cities and in need of products across various categories that can be reused, to those who might be leaving the same city and are looking to get rid of the same products. The buyers would get their needs met at cheaper rates while also ensuring good product quality, which is verified by the quality assurance team. Thus, the needs of both the parties are met.

The primary objective of this project is to design and implement a relational database that is industry-ready and covers the complete cycle of a re-commerce platform like Hand Me Downs- from putting up products for resale to the purchase of these products. Based on this idea of a re-commerce platform, the conceptual models (EER and UML) were created. Then these models were mapped to the logical model, which was normalized, and the primary and foreign keys were specified along with the null/null not allowed criteria. This relational model was then implemented in MySQL and data was queried, followed by an implementation in NoSQL using MongoDB Compass. Then the database was accessed using Python, through which the analytics capabilities are endless. A few analytics were conducted and visualizations created, which have been documented in this report.

## I. Introduction

One of the consumer behaviors that is rapidly expanding is resale, a topic of much discussion in the fashion business. Over 33 million shoppers in the US alone made their first-ever purchases of used clothing in 2020. In the following five years, the resale market is expected to quadruple and reach $77 billion. 2022 ought to be the year when everyone start thinking about resale as the technology required to enable it becomes more widely available. The digital secondhand market is positioned for significant development, despite the fact that resale is still primarily an offline activity. Not just smaller, simpler items like clothing and books are sold online. The way that house furnishings are sold for resale is also changing. While consumers will always be interested

in secondhand treasure hunts, evolving platforms and digital tools can assist would-be sellers in getting beyond the obstacles preventing them from taking part in the secondhand economy.

The adoption of a peer-to-peer or consignment model must be decided upon before a brand enters the resale market. In the peer-to-peer method, individual buyers and sellers get in touch with each other directly to discuss costs, seek and receive clarification on questions, and arrange shipping. Additionally, listings are the sole responsibility of the sellers. When using a consignment model, the brand or platform often handles every aspect of the transaction, and buyers only need to deal with one business to complete the transaction.

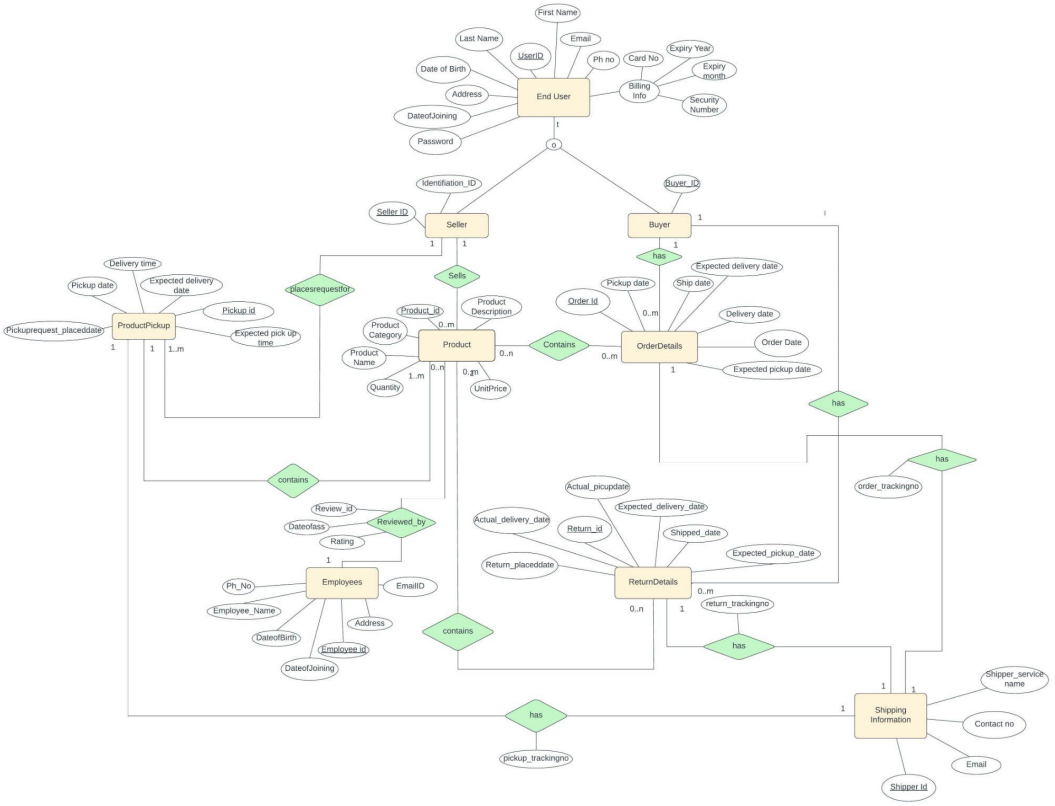
One of the largest resale site clothing, ThredUp, conducted a resale study in 2022 and found that secondhand clothing is becoming a global phenomenon, driven by North America. It is predicted that the U.S. market would more than double by 2026, reaching $82 billion.
Some notable findings from the report were:

- According to 41% of shoppers, buying used clothing is where they start their search for clothing.
- 62% of Gen Z and Millennial shoppers claimed to shop second hand before making a new purchase.
- Consumers in Generation Z and Millennials reported that 46% of them think about the resale value of clothing before making a purchase.
- 65% of people who made their first thrift store purchase a year ago said they wanted to stop purchasing quick fashion.

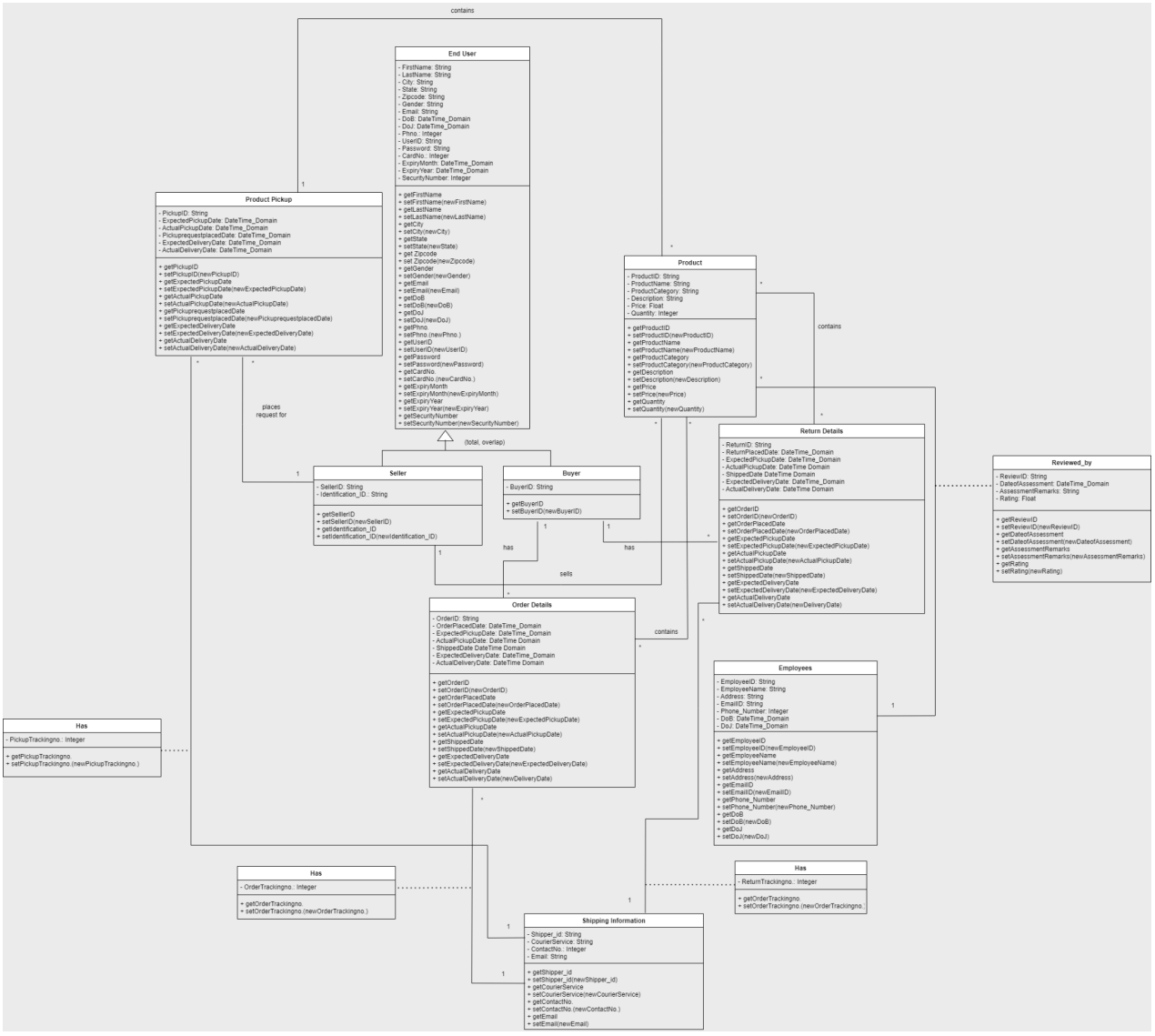
The analysis comes to the conclusion that resale is a more environmentally friendly solution to the wastefulness of the fashion sector. The report claims that in 2021, purchases of used clothing will outnumber new clothing purchases by about one billion. Additionally, almost two thirds of consumers think their personal consumption patterns have a big impact on the environment. Any company that disregards the need for the desire to become more sustainable does so at their own risk. The retail industry's green credentials could change as a result of this creative resale strategy.

## II. Conceptual Data Modeling

1. Enhanced Entity Relationship (EER) Model




2. Unified Modeling Language (UML) model



## III. Mapping the Conceptual Model to Relational Model

Primary Key**:** underlined
*Foreign Key*: italicized

**End-User**(UserID, FirstName, LastName, DateofBirth, Gender, City, State, Zipcode, Phno., DateofJoining, Email, Password, *CardNo.*)
**BillingInformation**(Card No., ExpiryMonth, ExpiryYear, SecurityNumber)
**Seller**(SellerID, Identification_ID, *UserID*)
**Buyer**(BuyerID, *UserID*)
**ProductPickup**(Pickup_id, Expected_pickupdate, Pickuprequest_placeddate, Actual_pickupdate, Expected_deliverydate, Actual_deliverydate, PickupTrackingno., *SellerID*, *Shipper_id*)
**Product**(Product_id, Product_Category, Product_Name, Product_Description, UnitPrice, Quantity, *SellerID, Pickup_id, ReviewID*)
**Reviews**(ReviewID, DateofAssessment, AssessmentRemarks, Rating, *EmployeeID*)
**Employees**(Employee_id, Employee Name, DateofBirth, DateofJoining, City, State, Zipcode, Phone_Number, Email)
**OrderDetails**(Orderid, Orderdate, Expected_pickup_date, Actual_pickup_date, Shipped_date, Expected_delivery_date, Actual_delivery_date, *BuyerID, Shipper_id,* OrderTrackingno.)
**Prod_ord**(*Orderid, Product_id*)
**ReturnDetails**(Returnrid, Returnplaceddate, Expected_pickup_date, Actual_pickup_date, Shipped_date, Expected_delivery_date, Actual_delivery_date, *BuyerID, Shipper_id*, ReturnTrackingno.)
**Prod_return**(*Returnid, ProductID*)
**ShippingInformation**(Shipper_id, Shipperservicename, ContactNo., Email)

## IV. Implementation of Relational Model via SQL (MySQL)

1. Retrieve the count of products that has been returned since 2021

```
select count(*) as Noofproducts_returned_since_2021
from returndetails
where act_delivdate>"2021-01-01"
```

| Noofproducts_returned_since_2021 |
|---|
| 37 |

2. Find the average ratings of products under each category

```
select p.product_category, avg(r.rating) as avg
from product p, reviews r
```

| product_category | avg |
|---|---|
| Electronics | 3.6939 |
| Books | 3.6757 |
| Accessories | 3.6515 |
| Organizers | 3.6491 |
| Bags | 3.6042 |
| Bedding | 3.5397 |
| Furniture | 3.4561 |
| Garments | 3.4407 |
| Sports Gear | 3.4390 |
| Winter Apparels | 3.4242 |
| Decor | 3.3617 |
| Utensils | 3.2642 |
| Footwear | 3.1754 |

```
where p.review_id=r.review_id
group by product_category
order by avg desc
```

3. Find the top 5 sellers with the most sold products

```
select pr.seller_id, count(o.product_id)
from product pr, ordered_products o
where pr.product_id=o.product_id and
pr.seller_id in
(select m1.seller_id from (select
p.seller_id, count(op.product_id) as num
from ordered_products op, product p
where op.product_id=p.product_id group by seller_id ) as m1
where 5 > (select count(o.product_id) from (select p1.seller_id, count(op1.product_id) as
num
from ordered_products op1, product p1
where op1.product_id=p1.product_id group by seller_id)  as m2
where m1.num < m2.num)) group by pr.seller_id;
```

| seller_id | count(o.product_id) |
|---|---|
| 1883473 | 5 |
| 5759046 | 5 |
| 1191511 | 5 |
| 4283889 | 5 |
| 4507677 | 5 |

4. Find the product categories in demand for males and females separately

```
select p.product_category, count(op.product_id) as
from ordered_products as op, product as p, end_user as
buyer as b, orderdetails as od
where op.product_id=p.product_id and
b.buyer_id=od.buyer_id and b.user_id=eu.user_id and
od.order_id=op.order_id and eu.user_id IN
(select user_id
from end_user
where gender='female')
group by p.product_category
order by count desc
```

count
eu,

| product_category | count |
|---|---|
| Winter Apparels | 24 |
| Utensils | 17 |
| Garments | 16 |
| Organizers | 16 |
| Accessories | 16 |
| Footwear | 14 |
| Furniture | 14 |
| Bedding | 13 |
| Sports Gear | 13 |
| Decor | 13 |
| Books | 11 |
| Bags | 11 |
| Electronics | 11 |

```
select p.product_category, count(op.product_id) as
from ordered_products as op, product as p,
end_user as eu, buyer as b, orderdetails as od
where op.product_id=p.product_id
and b.buyer_id=od.buyer_id
and b.user_id=eu.user_id
and od.order_id=op.order_id
and eu.user_id IN
```

count

| product_category | count |
|---|---|
| Winter Apparels | 20 |
| Bags | 17 |
| Electronics | 17 |
| Utensils | 17 |
| Bedding | 16 |
| Footwear | 15 |
| Organizers | 14 |
| Decor | 14 |
| Garments | 14 |
| Accessories | 13 |
| Books | 13 |
| Sports Gear | 12 |
| Furniture | 9 |

```
(select user_id
from end_user
where gender='male')
group by p.product_category
order by count desc
```

5. Retrieve the name of the supplier who charges the highest price for winter apparels

```
select eu.first_name, eu.last_name
from end_user eu, seller s
where s.user_id=eu.user_id
and s.seller_id in
(select p1.seller_id
from product p1
where p1.product_category='Winter Apparels'
and p1.unitprice>= ALL
(select p1.unitprice
from product p1
where p1.product_category='Winter Apparels'))
```

| first_name | last_name |
|---|---|
| Andre | Cummerata |

6. Retrieve the names of the sellers who do not charge the lowest price for product category = bags

```
select s.seller_id, e.first_name, e.last_name
from seller s, end_user e
where s.user_id=e.user_id and s.seller_id in
(select distinct seller_id
from product
where product_category=" Bags" and unitprice > ANY
(select unitprice from product
where product_category=" Bags"));
```

| seller_id | first_name | last_name |
|---|---|---|
| 8625224 | Eduardo | Waters |
| 8048643 | Sheila | Bashirian |
| 7481199 | Nasir | Jacobs |
| 1641084 | Marc | Romaguera |
| 9953068 | Ciara | Carroll |
| 1189128 | Thea | Sauer |
| 7865717 | Jessica | Stiedemann |
| 2587954 | Kariane | Gislason |
| 1158707 | Albert | Weimann |

Result 8

Output

Action Output

| # | Time | Action | Message |
|---|---|---|---|
| 9 | 00:20:52 | select s.seller_id, eu.first_name, eu.last_name, eu.city from seller s, end_user eu where s.use... | 5 row(s) returned |
| 10 | 00:22:39 | select s.seller_id, e.first_name, e.last_name from seller s, end_user e where s.user_id=e.user... | 37 row(s) returned |

7. Retrieve the seller ID, name and, city of the sellers whose products have not been sold yet

```
select s.seller_id, eu.first_name, eu.last_name, eu.city

from seller s, end_user eu
where s.user_id=eu.user_id
and not exists
(select s.seller_id
from ordered_products op, product p
where op.product_id=p.product_id
and s.seller_id=p.seller_id)
```

| seller_id | first_name | last_name | city |
|---|---|---|---|
| 2173135 | Orlando | O'Kon | Ravenbury |
| 7267819 | Annalise | Corwin | North Katrinaberg |
| 8367401 | Chaim | Labadie | North Hillary |
| 1921156 | Pierce | Lesch | Koelpinport |
| 0663887 | Tobin | Huels | Hudsonport |

8. Retrieve the id and name of buyers who are located is Massachussets or have bought bags

```
select b.buyer_id, e.first_name
from buyer b, end_user e
where e.user_id=b.user_id and
e.state="Massachusetts"
union
select b.buyer_id, e.first_name
from buyer b, end_user e, orderdetails od,
ordered_products op, product p
where od.buyer_id=b.buyer_id and
od.order_id=op.order_id and
op.product_id=p.product_id and
e.user_id=b.user_id and p.product_category=" Books"
```

| buyer_id | first_name |
|---|---|
| 40317kh2 | Emile |
| 67887vx3 | Emery |
| 71129xp6 | Theodora |
| 02853av8 | Antone |
| 22738xd5 | Kasey |
| 84677tt8 | Albertha |
| 49460nn3 | Jalyn |

## V. Implementation of Model via NoSQL (MongoDB)

1. Retrieve the distinct product categories

db.product.distinct('product_category')

```
> db.product.distinct('product_category')
< [
    ' Accessories',     ' Bags',
    ' Bedding',         ' Books',
    ' Decor',           ' Electronics',
    ' Footwear',        ' Furniture',
    ' Garments',        ' Organizers',
    ' Sports Gear',     ' Utensils',
    'Winter Apparels'
  ]
```

2. Get all the products with product category as furniture or bedding:
db.product.find({$or:[{product_category:" Furniture"}, {product_category:" Bedding'}]})

```
> db.product.find({$or:[{product_category:" Furniture"}, {product_category:" Bedding"}]})
< { _id: ObjectId("63926304d40f9a7c1c9f3332"),
    product_id: 'a83729',
    product_category: ' Bedding',
    product_name: 'molestias',
    product_description: 'Nulla accusantium nostrum molestiae ab. Conse',
    unitprice: 527,
    quantity: 1,
    seller_id: '4448556',
    pickup_id: '407xk537',
    review_id: '51718' }
  { _id: ObjectId("63926304d40f9a7c1c9f3333"),
    product_id: 'a85415',
    product_category: ' Bedding',
    product_name: 'cupiditate',
    product_description: 'Et esse ipsa nostrum cumque repellat. Et corr',
    unitprice: 391,
    quantity: 1,
    seller_id: '2715521',
    pickup_id: '203wt249',
    review_id: '06646' }
```

3. Calculate the average price of products within each category:
db.product.aggregate([{$group:{_id:"$product_category", avg:{$avg:"$unitprice"}}}])

```
> db.product.aggregate([{$group:{_id:"$product_category", avg:{$avg:"$unitprice"}}}])
< { _id: ' Bedding', avg: 276.55555555555554 }
  { _id: ' Utensils', avg: 292.0566037735849 }
  { _id: 'Winter Apparels', avg: 284.6212121212121 }
  { _id: ' Books', avg: 313.13513513513516 }
  { _id: ' Bags', avg: 270.6041666666667 }
  { _id: ' Decor', avg: 309.6595744680851 }
  { _id: ' Garments', avg: 309.6271186440678 }
  { _id: ' Furniture', avg: 283.87719298245617 }
  { _id: ' Accessories', avg: 312.77272727272725 }
  { _id: ' Footwear', avg: 252.50877192982455 }
  { _id: ' Sports Gear', avg: 299.3414634146341 }
  { _id: ' Electronics', avg: 301.59183673469386 }
  { _id: ' Organizers', avg: 321.6666666666667 }
```

## VI. Database Access via Python

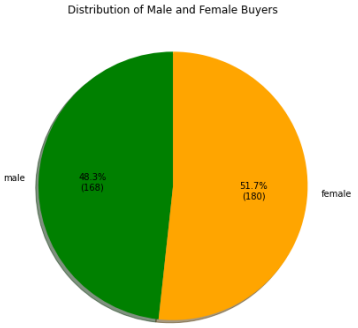
The database is accessed via Python code implemented in the Jupyter notebook. A few queries implemented in MySQL have been implemented in Jupyter, along with some additional queries to conduct analysis and visualization. The connection to MySQL is done using mysql.connector. The data is queried, transferred into a dataframe, and then analysis/visualization is performed.

1. Bar graph of distribution of buyers across different states

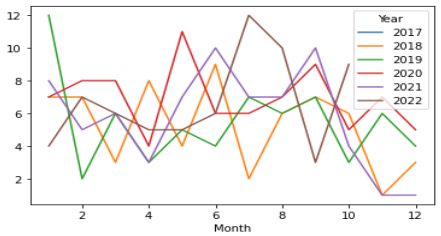

2. Input the state and product category to find the respective sellers

```
Enter the state to search for sellers!
Connecticut
Enter the product category to search for sellers! Winter Apparels, Footwear, Sports Gear, Organizers, Decor, Furniture, Books,
Garments, Bedding, Bags, Utensils, Electronics, Accessories
 Decor
    product_id product_category seller_id        state
90      a95976            Decor   1153415  Connecticut
92      i92676            Decor   1153415  Connecticut
134     z39254            Decor   1510548  Connecticut
170     n17836            Decor   2012099  Connecticut
```

3. Finding the distribution of male and female buyers



4. Retrieving the number of customers who joined over the years




## VI. Summary and Recommendations

The entire process of organizing and carrying out effective product shipment and storage from the point of origin to the site of consumption is included in our industry-ready consignment model for resale. This resale platform can expand greatly, especially when it comes to international students, with a well-built quality assurance team. Students who are moving abroad but have limited funds in need of the essentials like furniture, bedding, winter clothing for harsh weather, sports equipment, refurbished electronics, and even school supplies can totally rely on a platform like this to shop, and also sell goods when they are ready to move out.

**Drawbacks of the model:** To ensure no redundancy in the data, we normalized our model which caused our data to be very spread out. This led to a lot of joins to query relevant data, which is a tedious and computationally expensive process. Large number of tables made the model complex.

**Improvements:** An improvement on the database would be to implement data governance measures.