Warehouse Management System

Database Management

CMPT 308N-113

Bezos' Indentured Servants

Marist College
School of Computer Science and Mathematics

Submitted To:
Dr. Reza Sadeghi

9/6/2023

Table of Contents

Table of Figures----------Page 2

Project Description----------Page 3

Project Objectives----------Page 5

Review the Related Work----------Page 7

Merits of Project----------Page 8

Entity Relationship Model (ER Model)----------Page 9

Enhanced Entity Relationship Model (EER Model)----------Page 14

References----------Page 17

Table of Figures

Figure 1 (ER Model)----------Page 9

Project Description of *Bezos' Indentured Servants*

Team Name

Name of the Team ----------*Bezos' Indentured Servants*

Github Link

https://github.com/gabrielle-knapp1/Project-1-Database-Man-

Team Members

1. Ethan Morton ---------- ethan.morton1@marist.edu (Team Head)

2. Gabrielle Knapp----------gabrielle.knapp1@marist.edu (Team Member)

Description of Team Members

1. ***Ethan Morton***

   I am a sophomore and I'm majoring in computer science with a concentration in software development. I am minoring in mathematics, information technology, information systems, and cybersecurity. I know C#, Java, Python, HTML, CSS, JavaScript, a little bit of SQL, XML, and Lua. I am from Granby, Connecticut and I enjoy playing tennis, hanging out with my friends, and playing games. Some clubs I'm involved in are campus ministry, computer society, games society, and club tennis. Together we chose me to be the Team Head because of my ability to manage communications.

2. ***Gabrielle Knapp***

   Gabrielle Knapp is a sophomore at Marist College majoring in Computer Science with minors in Spanish and Economics. She comes from Carlisle, Pennsylvania and has been enjoying her time adjusting to college life. Activities she currently is participating in include intramural badminton and the games society. In her free time, she loves reading,

going on long walks, and playing board games. She is excited to see how she learns and grows her programming skills throughout this class and her entire time at Marist. Gabrielle is excited to bring her hardworking, can-do attitude to her group in this project for her Database Management class. She chose to work with Ethan on this project because she has enjoyed working with him in other classes, including working together on their project in Dr. Sadeghi's Intro to Programming class. Together, as a pair, they chose Ethan to be the Team Head because of his excellent ability to manage communications.

Project Objectives

Summary

We have selected project sample 4, the warehouse management system. The warehouse management system (WMS) provides an organized way of storing different products and elements in a warehouse. You can consider a library as a warehouse, which maintains books' details and user libraries. A general WMS stores details of name and identification number of products, their store time, the required storage condition, price, weight, height, etc. Following this, this system allows guest users to search for different content and request to borrow/buy them. Our WMS will store the data of different user types in distinct SQL tables.

Modules

- Admin Roles
    - Admin has login (username/password) and ability to change that login
    - Admin can remove users from WMS
    - Admin has ability to add a guest user with a login (guest has limited abilities)
        - Guest user cannot define/remove other users
    - Admin can add, delete, and edit items to WMS with various details
    - Admin can view, accept, and reject the list of borrowing requests
- User Roles
    - Users can search through items in WMS depending on various item details
    - Users have ability to save favorite items
    - Users can request to borrow/buy specific items at specific times
    - Users have ability to view the history of borrowed/bought items
- WMS should be user-friendly software

    - Welcome page
    - Menu with all functions
    - All functions in a tabular format
    - Well-organized list of requested items
    - Exit function with friendly goodbye
    - Should show warnings IF:
        - The Admin user tries to add a new item to the library with an existing ID
        - If a guest user tries to borrow more than 3 items
        - A user search request returns null items
- WMS should protect User's information/data
    - WMS passwords & recorded info should be Ciphered using Caesar Cipher Method

Review the Related Work

1) BR Williams Warehouse Management System link

a) Positive Aspects

i) Inventory history and transaction logs.

b) Negative Aspects

i) Uses barcodes to scan which automatically update the database with the correct information.

2) Koha Management System link

a) Positive Aspects

i) Intuitive navigation for users.

b) Negative Aspects

i) Built for library management rather than warehouse management.

3) ShipHero Warehouse Management System link

a) Positive Aspects

i) Users can return items.

b) Negative Aspects

i) Multi-carrier shopping: searching the same items from different stores won't work in our project.

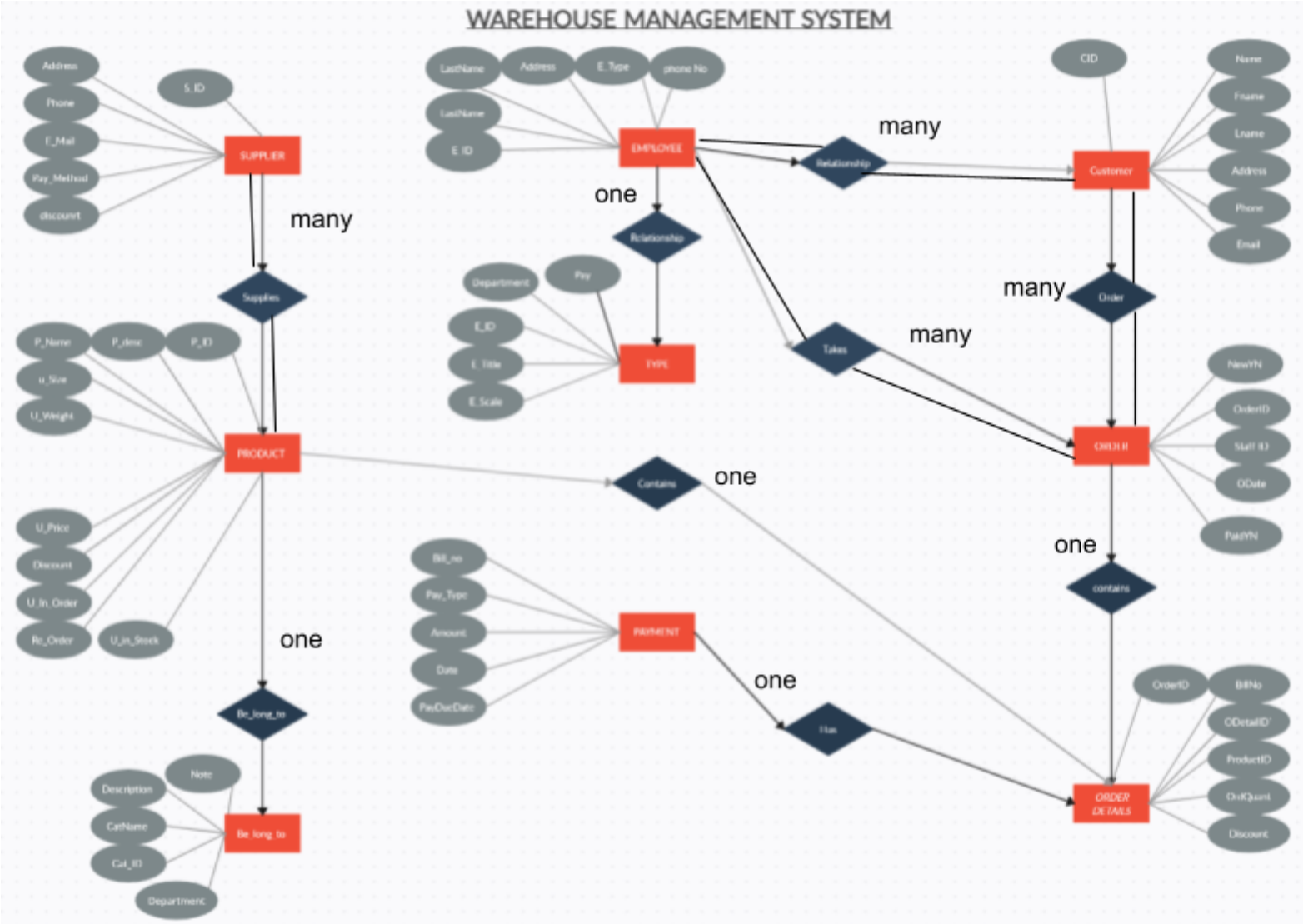
Merits of Project

Merits:

- The WMS will organize all of the items in the warehouse and will allow users to easily access a list of all of these items and their details
- We plan on implementing the positives of other Warehouse Management Systems, including the implementation of detailed inventory logs, allowing for clear details about returning items, and creating an intuitive navigation for users.
- The WMS will clearly differentiate the roles of User and Admin, and allow Admin complete control over the warehouse and the privileges of the users.

Entity Relationship Model (ER Model)

Figure 1 (ER Model):



Description:

- Here, we have a more detailed version of what we might need for the warehouse and what is described in the EER Diagram. We have the Supplier, who supplies the product, to the warehouse and the Be_long_to entity to track who owns what stock. The product is also contained in the

order details, which is has the payment of the customer who places the order, the order taken by the employee of a specific employee type. The overlap with the requirements stated in the EER diagram is that the Accounts tab holds the data for Suppliers, Employees, Employee Type, and Customers. The Transactions hold the Payment, Order information, Order Details, and the be_longs_to data. The Warehouse tab holds the Product information.

Description of each entity, attribute, relationship, participation, and cardinality:

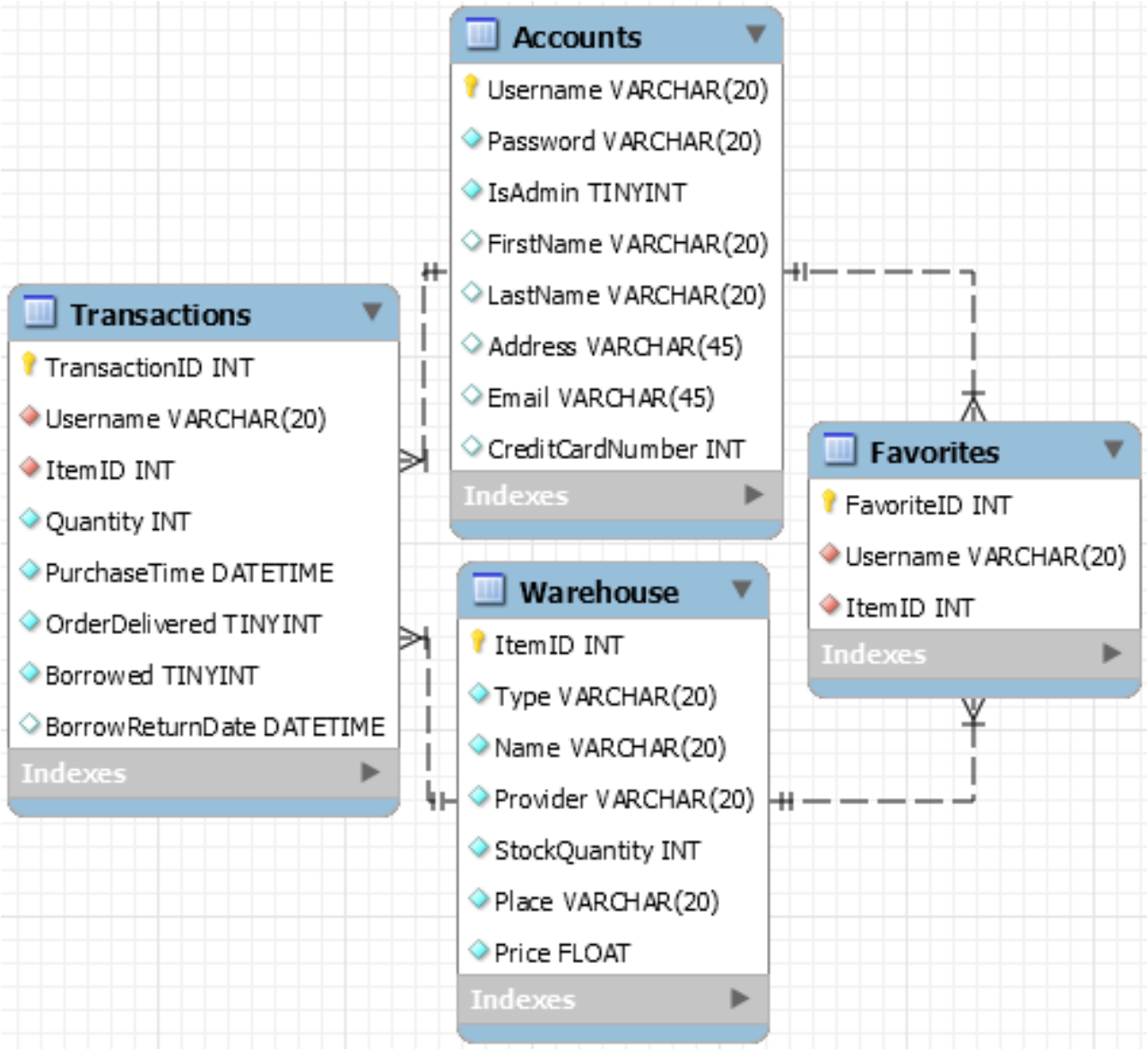
- Supplier
    - SID: ID of the supplier
    - Address, Phone, Email: contact
    - Pay Method: payment method of supplier
    - Discount: if there is a discount for this supplier
- Product
    - PID: ID of the product
    - P_desc: Describes the product
    - P_name: Name of product
    - U_ Size, U_Weight: dimensions of product
    - U_Price, U_Discount: Price and if there's a discount
    - U_inOrder, U_reOrder: booleans, if unit is ordered/ will be reordered
    - U_inStock: boolean, whether unit is in stock

- Be_Long_To: who owns the specific unit
    - Note, Description: Describes Owner
    - Cat_Name: Category of owner (regular, new)
    - Cat_ID: ID of owner
    - Department: Department of goods
- Employee
    - Address, PhoneNo: contact information
    - FirstName, LastName: name
    - EID: employee ID
    - E Type: full time, part time
- Type: Employee Type
    - Pay
    - Department
    - EID: EID classification of employee
    - E_Title: name of job
    - E_Scale: ranking on employee scale of pay
- Customer
    - Name, Fname, Lname: name
    - Address, Phone, Email: contact
    - CID: customer ID

- Order
    - NewYN: Boolean, Whether this is a first-time owner
    - OrderID: ID of order
    - StaffID: ID of employee who took the order
    - ODate: date of order
    - PaidYN: Boolean, whether order is paid
- Order Details
    - OrderID: ID of the order
    - BillNo: Number of Bill
    - Product ID: IDs of products ordered
    - OrdQuant: Quantity of what is ordered
    - Discount: If order has a discount, how much
- Payment
    - BillNo: Number of bill
    - PayType: How customer pays
    - Amount: How much customer owes
    - PayDueDate: When payment is Due
    - Date: When payment was sent
- Supplier → Suplies → Product

    - The supplier supplies the product, one supplier can supply many products
- Product → Belongs to → Be_long_to
    - The product belongs to an owner, and a product can only belong to one owner but one owner can own many different products
- Employee → Has a → Type
    - Every employee has only one type
- Employee → Has a relationship with a → Customer
- Customer → Orders an → Order
    - A customer can put in many orders, but each order can only have one customer
- Employee → Takes an → Order
    - One employee is responsible for each order, and each employee can take multiple orders but each order has only one employee on file
- Order → Contains → Order Details
- Order Details → Has → Payment
    - Each Order has one payment on file, one payment can be on file for multiple orders

Enhanced Entity Relationship Model (EER Model)



Description about keys & relationships:

- Each Account:
  - is identified by the username
  - must have a password
  - is either an admin or guest
  - has a first and last name
  - has an address to ship the items to
  - has an email to contact the user

    - has a credit card number to purchase items
- Each Warehouse Item:
    - has a unique item ID
    - has a type
    - has a name
    - has a provider
    - has a quantity of that item in the warehouse to prevent redundancy
    - has a place it's stored in the warehouse
    - has a price per unit
- Each Transaction:
    - has a unique transaction ID
    - has the username of the account doing the transaction
    - has the ID of the item purchased
    - has the quantity of that item being purchased
    - has the date and time of the transaction
    - has either been delivered or not
    - has either been borrowed temporarily or purchased
    - has a borrow return date if and only if the item is being borrowed
- Each Favorite:
    - has a unique favorite ID
    - holds the username of the account that has favorited the item
    - has the item being favorited

Description of implementation of these features:

- ***Account***
    - ***Logging in will give you access to the website and only show your personal data.***
    - ***Admins may remove specific accounts.***
- ***Warehouse***
    - ***All items in the warehouse will be displayed.***
    - ***Users can search for items and sort the list.***
    - ***Admins can edit the list of items.***
- ***Transactions***
    - ***Users will be able to view their past transactions.***
    - ***Users will be able to make transactions.***
    - ***Admins can view all transactions.***

- ***Favorites***
    - ***Each user can view the items they favorited to access them quickly.***

References

- --
- --