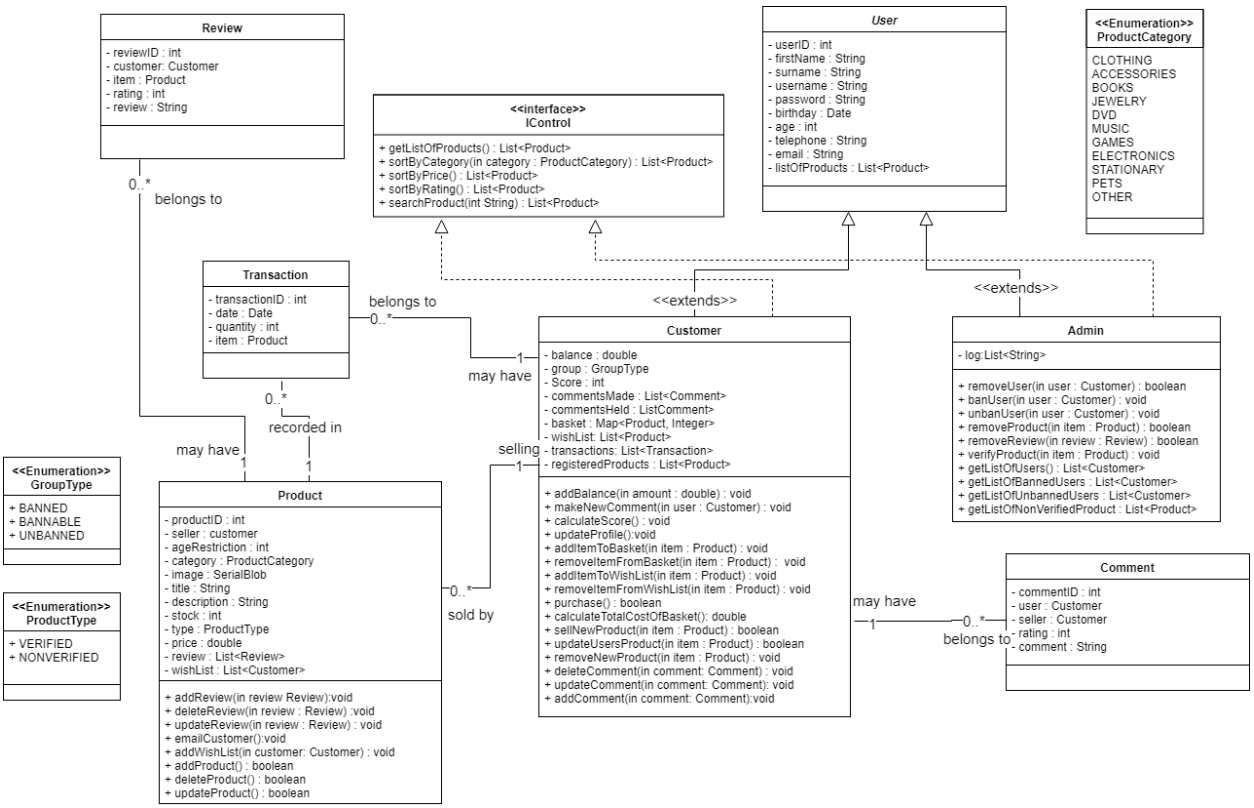
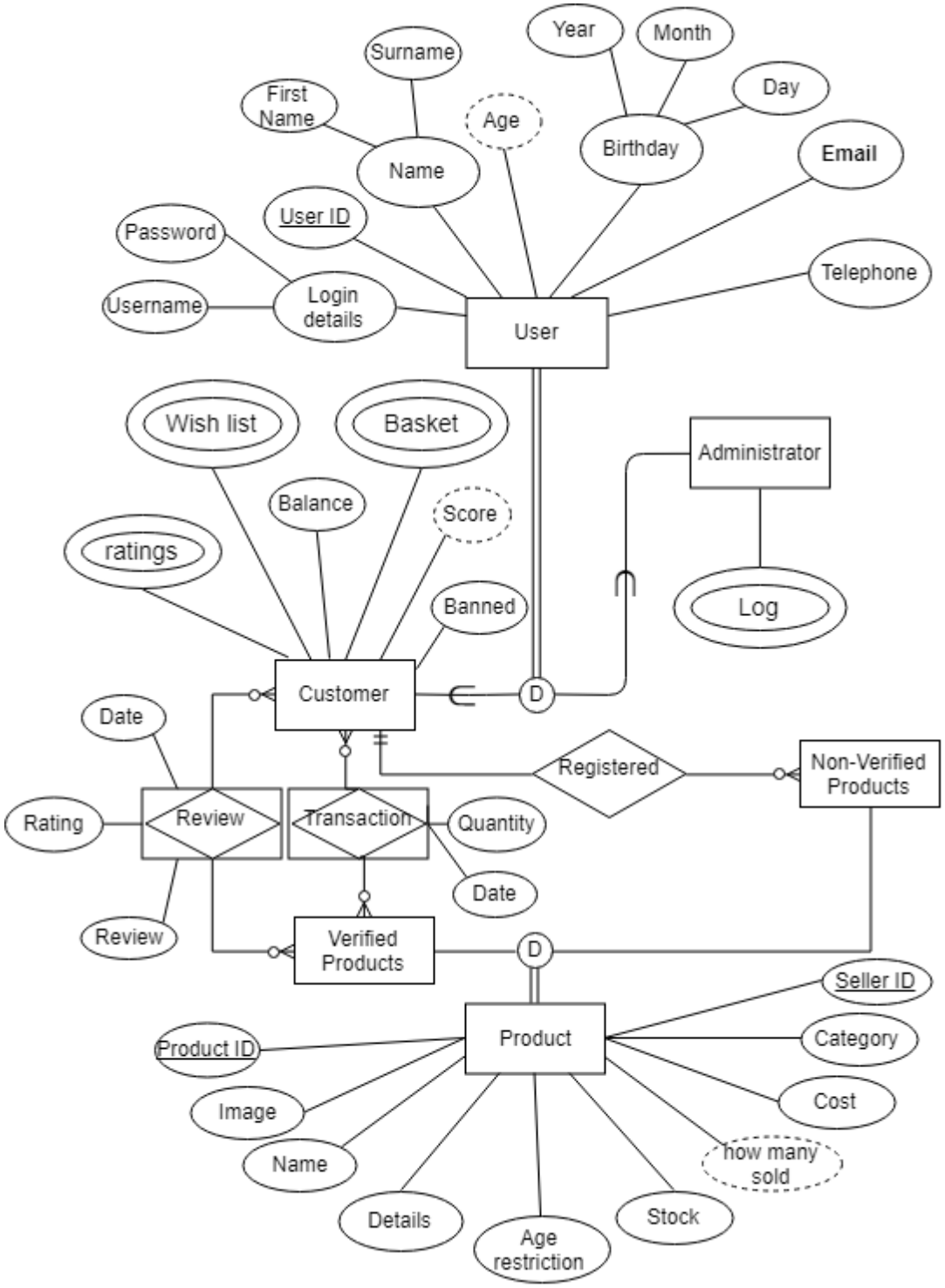

Review
- reviewID : int
- customer: Customer
- item : Product
- rating : int
- review : String
0..*
belongs to
<<interface>>
IControl
+ getListOfProducts() : List<Product>
+ sortByCategory(in category : ProductCategory) : List<Product>
+ sortByPrice() : List<Product>
+ sortByRating() : List<Product>
+ searchProduct(int String) : List<Product>
User
- userID : int
- firstName : String
- surname : String
- username : String
- password : String
- birthday : Date
- age : int
- telephone : String
- email : String
- listOfProducts : List<Product>
<<Enumeration>>
ProductCategory
CLOTHING
ACCESSORIES
BOOKS
JEWELRY
DVD
MUSIC
GAMES
ELECTRONICS
STATIONARY
PETS
OTHER
Transaction
- transactionID : int
- date : Date
- quantity : int
- item : Product
belongs to
0..*
1
may have
<<extends>>
<<extends>>
Customer
- balance : double
- group : GroupType
- Score : int
- commentsMade : List<Comment>
- commentsHeld : ListComment>
- basket : Map<Product, Integer>
- wishList: List<Product>
- transactions: List<Transaction>
- registeredProducts : List<Product>
+ addBalance(in amount : double) : void
+ makeNewComment(in user : Customer) : void
+ calculateScore() : void
+ updateProfile():void
+ addItemToBasket(in item : Product) : void
+ removeItemFromBasket(in item : Product) : void
+ addItemToWishList(in item : Product) : void
+ removeItemFromWishList(in item : Product) : void
+ purchase() : boolean
+ calculateTotalCostOfBasket(): double
+ sellNewProduct(in item : Product) : boolean
+ updateUsersProduct(in item : Product) : boolean
+ removeNewProduct(in item : Product) : void
+ deleteComment(in comment: Comment) : void
+ updateComment(in comment: Comment): void
+ addComment(in comment: Comment):void
Admin
- log:List<String>
+ removeUser(in user : Customer) : boolean
+ banUser(in user : Customer) : void
+ unbanUser(in user : Customer) : void
+ removeProduct(in item : Product) : boolean
+ removeReview(in review : Review) : boolean
+ verifyProduct(in item : Product) : void
+ getListOfUsers() : List<Customer>
+ getListOfBannedUsers : List<Customer>
+ getListOfUnbannedUsers : List<Customer>
+ getListOfNonVerifiedProduct : List<Product>
0..*
recorded in
may have
1
1
selling
1
<<Enumeration>>
GroupType
+ BANNED
+ BANNABLE
+ UNBANNED
<<Enumeration>>
ProductType
+ VERIFIED
+ NONVERIFIED
Product
- productID : int
- seller : customer
- ageRestriction : int
- category : ProductCategory
- image : SerialBlob
- title : String
- description : String
- stock : int
- type : ProductType
- price : double
- review : List<Review>
- wishList : List<Customer>
+ addReview(in review Review):void
+ deleteReview(in review : Review) :void
+ updateReview(in review : Review) : void
+ emailCustomer():void
+ addWishList(in customer: Customer) : void
+ addProduct() : boolean
+ deleteProduct() : boolean
+ updateProduct() : boolean
0..*
sold by
may have
1
0..*
belongs to
Comment
- commentID : int
- user : Customer
- seller : Customer
- rating : int
- comment : String


Year
Month
Surname
Day
First Name
Age
Birthday
Email
Name
User ID
Password
Telephone
Username
Login details
User
Wish list
Basket
Administrator
Balance
Score
ratings
Log
Banned
Customer
D
Date
Registered
Non-Verified Products
Rating
Review
Transaction
Quantity
Date
Review
Verified Products
D
Seller ID
Product
Product ID
Category
Cost
Image
how many sold
Name
Details
Stock
Age restriction

User(User ID:int, First_name:String, Surname:string, Username:String, Birthday:Date, Telephone:String, Password:String, Email:String)

Customer(User_ID:int, Balance:double, Banned:boolean)

Customer_Comments(User_ID:int, Seller_ID:int, Rating:int, Comment:String)

Customer_Wishlist(User_ID:int, Product_ID:int)

Customer_Basket(User_ID:int, Product_ID:int, quantity:int)

Customer_Registered_Product(User_ID:int, Product_ID:int)

Administrator(User_ID:int)

Log(User_ID:int, Action:String)

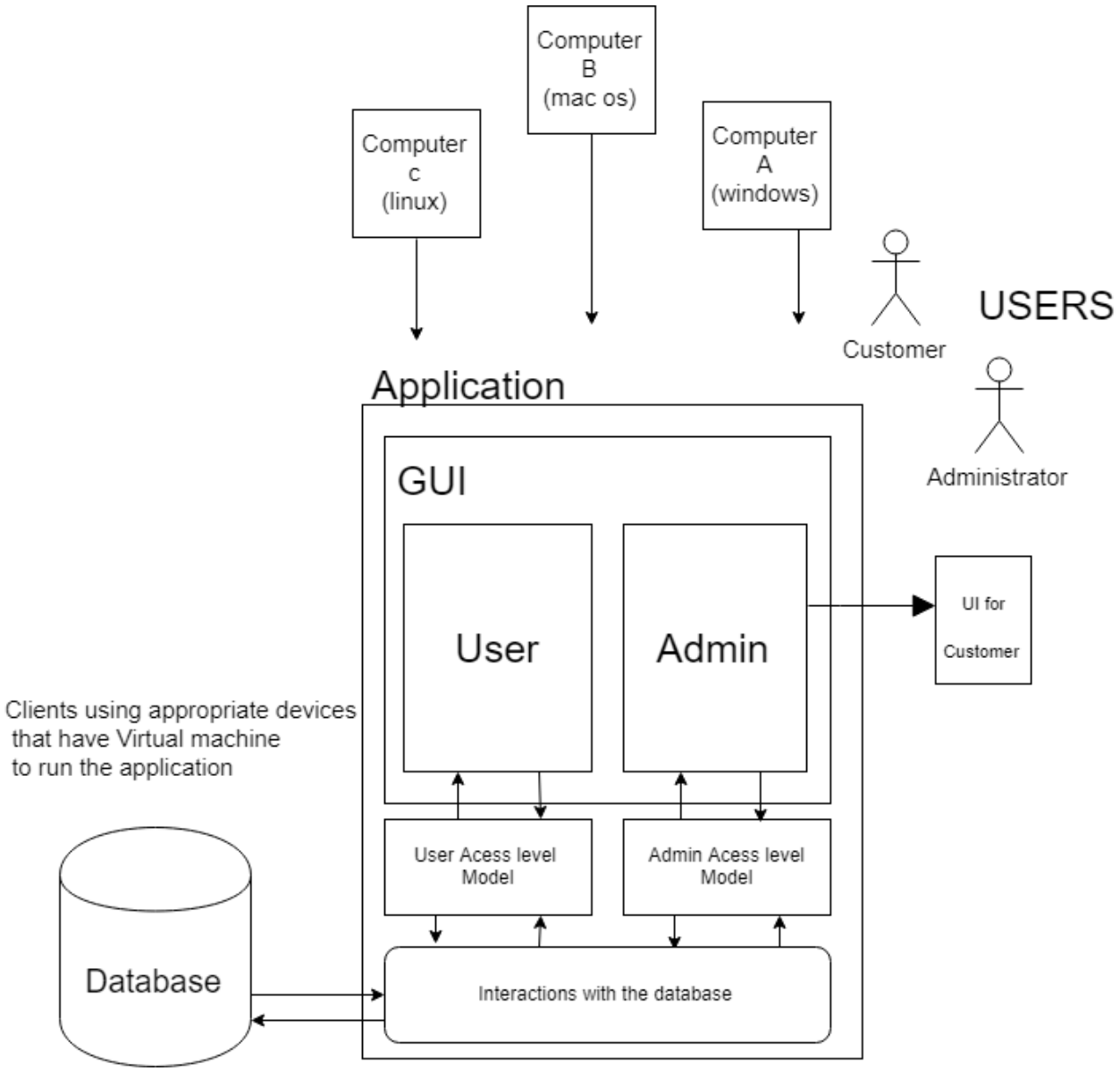
Customer_Transactions(Customer_ID:int User_ID:int, Product_ID:int, Date:date, Quantity:int)

Customer_Product_Reviews(Review_ID:int ,User_ID:int, Product_ID:int, Rating:int, Review:String)

Verified_Product(Product_ID:int)

Non-Verified_Prodruct(Product_ID:int)

Product(Product_ID:int, Seller_ID:int, image: varbinary(max), Name:String, Details:String, Initial_Stock:int, Current_Stock:int, number_sold:int, cost:double, AgeRestriction:int)


Computer
B
(mac os)
Computer
c
(linux)
Computer
A
(windows)
Customer
USERS
Application
Administrator
GUI
User
Admin
UI for
Customer
Clients using appropriate devices
that have Virtual machine
to run the application
User Acess level
Model
Admin Acess level
Model
Database
Interactions with the database