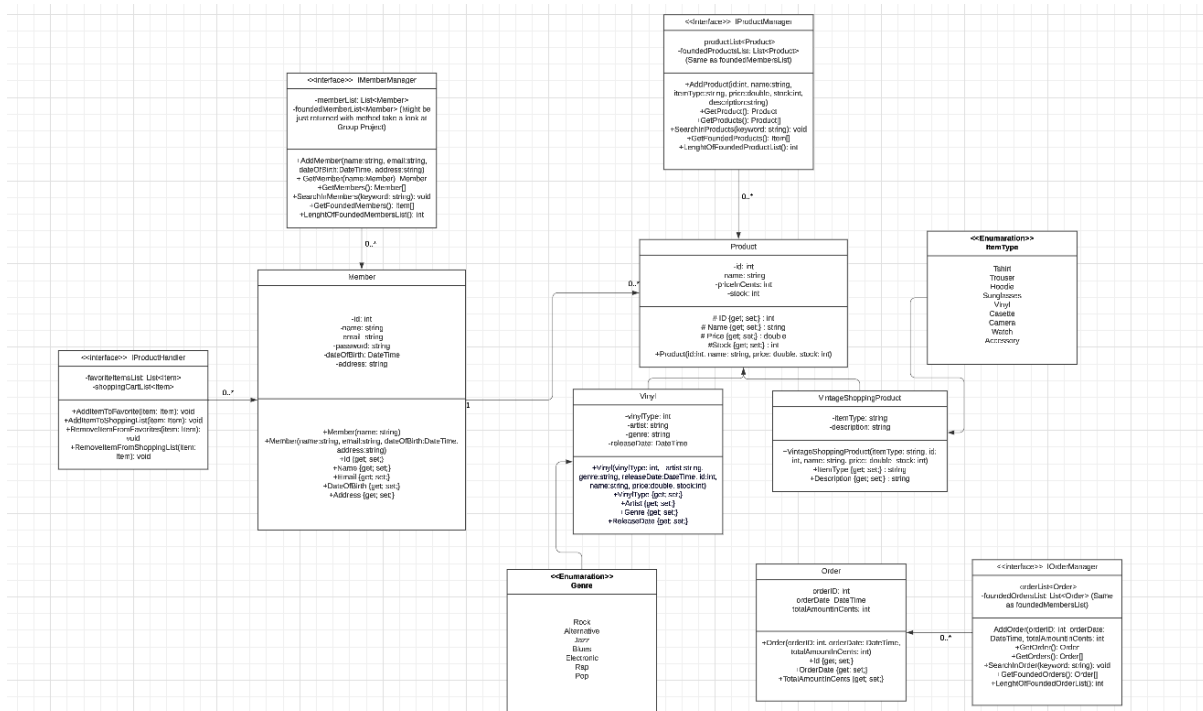This is the whole UML Diagram. In case of it is too blurry I also putted smaller(cutted in half) version below.

- I also added a Order Table but I couldn't find how should I connect that. So, for now I have left it like that.

<<interface>> IProductManager

- -productList<Product>
- -foundedProductsList: List<Product>
- (Same as foundedMembersList)

- +AddProduct(id:int, name:string, itemType:string, price:double, stock:int, description:string)
- +GetProduct(): Product
- +GetProducts(): Product[]
- +SearchInProducts(keyword: string): void
- +GetFoundedProducts(): Item[]
- +LenghtOfFoundedProductList(): int

<<interface>> IMemberManager

- -memberList: List<Member>
- -foundedMemberList<Member> (Might be just returned with method take a look at Group Project)

- +AddMember(name:string, email:string, dateOfBirth:DateTime, address:string)
- + GetMember(name:Member): Member
- +GetMembers(): Member[]
- +SearchInMembers(keyword: string): void
- +GetFoundedMembers(): Item[]
- +LenghtOfFoundedMembersList(): int

<<interface>> IProductHandler

- -favoriteItemsList: List<Item>
- -shoppingCartList<Item>

- +AddItemToFavorite(item: Item): void
- +AddItemToShoppingList(item: Item): void
- +RemoveItemFromFavorites(item: Item): void
- +RemoveItemFromShoppingList(item: Item): void

Member

- -Id: int
- -name: string
- -email: string
- -password: string
- -dateOfBirth: DateTime
- -address: string

- +Member(name; string)
- +Member(name:string, email string, dateOfBirth:DateTime, address:string)
- +Id {get; set;}
- +Name {get; set;}
- +Email {get; set;}
- +DateOfBirth {get; set;}
- +Address {get; set;}

Product

- -id: int
- -name: string
- -priceInCents: int
- -stock: int

- # ID {get; set;} : int
- # Name {get; set;} : string
- # Price {get; set;} : double
- #Stock {get; set;} : int
- +Product(id:int, name: string, price: double, stock: int)

<<Enumaration>> ItemType

- Tshirt
- Trouser
- Hoodie
- Sunglasses
- Vinyl
- Casette
- Camera
- Watch
- Accessory

Vinyl

- -vinylType: int
- -artist: string
- -genre: string
- -releaseDate: DateTime

- +Vinyl(vinylType: int, , artist:string, genre:string, releaseDate:DateTime, id:int, name:string, price:double, stock:int)
- +VinylType {get; set;}
- +Artist {get; set;}
- +Genre {get; set;}
- +ReleaseDate {get; set;}

VintageShoppingProduct

- -itemType: string
- -description: string

- +VintageShoppingProduct(itemType: string, id: int, name: string, price: double, stock: int)
- +ItemType {get; set;} : string
- +Description {get; set;} : string

Member

- -Id: int
- -name: string
- -email: string
- -password: string
- -dateOfBirth: DateTime
- -address: string

- +Member(name; string)
- +Member(name:string, email:string, dateOfBirth:DateTime, address:string)
- +Id {get; set;}
- +Name {get; set;}
- +Email {get; set;}
- +DateOfBirth {get; set;}
- +Address {get; set;}

Product

- -id: int
- -name: string
- -priceInCents: int
- -stock: int

- # ID {get; set;} : int
- # Name {get; set;} : string
- # Price {get; set;} : double
- #Stock {get; set;} : int
- +Product(id:int, name: string, price: double, stock: int)

<<Enumaration>> ItemType

- Tshirt
- Trouser
- Hoodie
- Sunglasses
- Vinyl
- Casette
- Camera
- Watch
- Accessory

Vinyl

- -vinylType: int
- -artist: string
- -genre: string
- -releaseDate: DateTime

- +Vinyl(vinylType: int, , artist:string, genre:string, releaseDate:DateTime, id:int, name:string, price:double, stock:int)
- +VinylType {get; set;}
- +Artist {get; set;}
- +Genre {get; set;}
- +ReleaseDate {get; set;}

VintageShoppingProduct

- -itemType: string
- -description: string

- +VintageShoppingProduct(itemType: string, id: int, name: string, price: double, stock: int)
- +ItemType {get; set;} : string
- +Description {get; set;} : string

<<Enumaration>> Genre

- Rock
- Alternative
- Jazz
- Blues
- Electronic
- Rap
- Pop

Order

- orderID: int
- orderDate: DateTime
- totalAmountInCents: int

- +Order(orderID: int, orderDate: DateTime, totalAmountInCents: int)
- +Id {get; set;}
- +OrderDate {get; set;}
- +TotalAmountInCents {get; set;}

<<interface>> IOrderManager

- -orderList<Order>
- -foundedOrdersList: List<Order> (Same as foundedMembersList)

- +AddOrder(orderID: int, orderDate: DateTime, totalAmountInCents: int
- +GetOrder(): Order
- +GetOrders(): Order[]
- +SearchInOrder(keyword: string): void
- +GetFoundedOrders(): Order[]
- +LenghtOfFoundedOrderList(): int

Relationship multiplicities shown: 0..*, 1