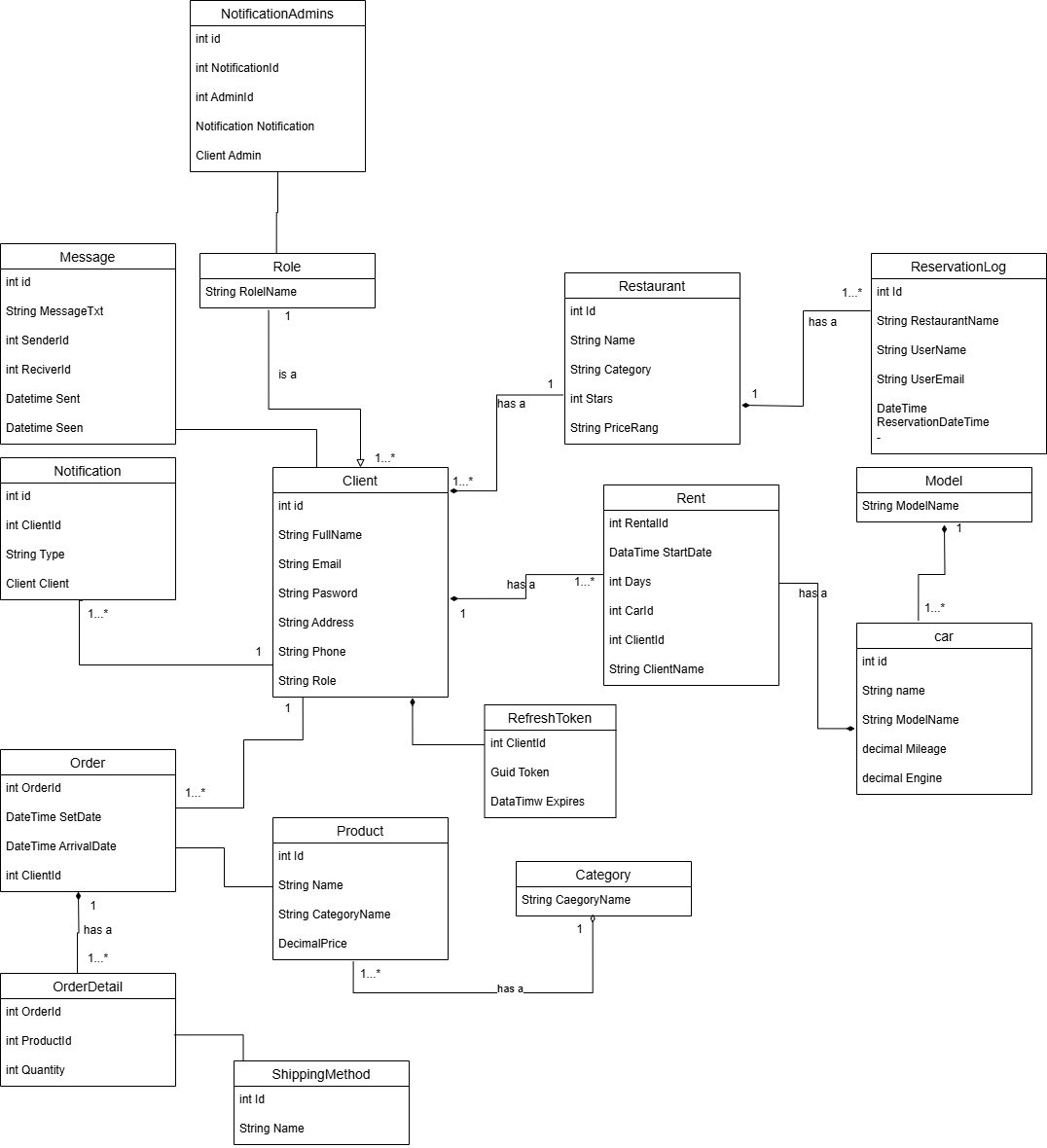
1. Dokumenti i Dizajnit të Detajuar (DDD)

Ky dokument përshkruan në detaje dizajnin për aplikacionin web City Services, duke përfshirë modelet e dizajnit të nivelit të mesëm dhe të ulët dhe lidhjen midis tyre. Objektivi kryesor është të sigurohet që sistemi të jetë i fortë, i shkallëzueshëm dhe i mirëmbajtshëm, duke respektuar parimet e programimit të orientuar drejt objekteve.

2. Mid-Level Design Models

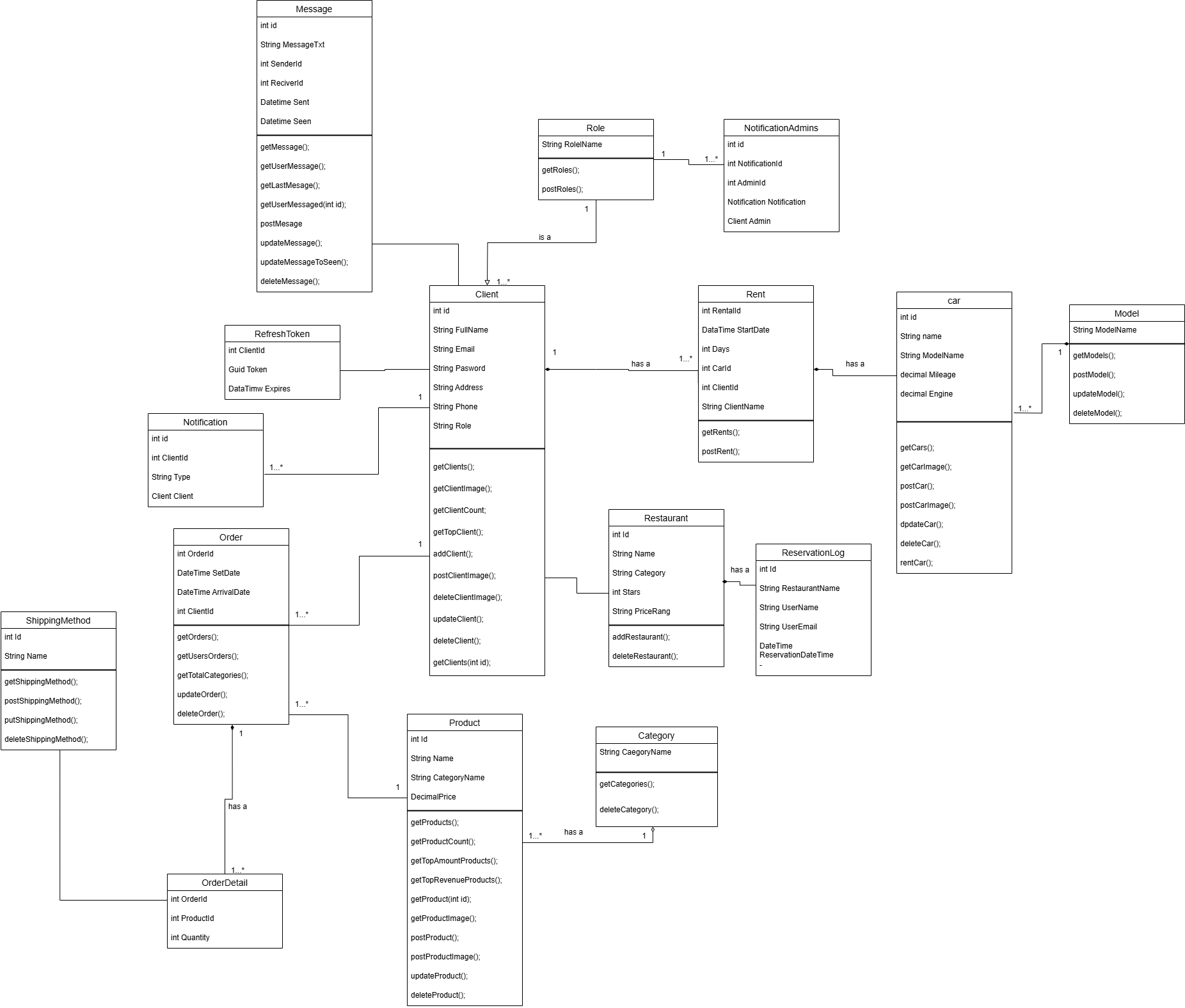
Modelet e dizajnit të nivelit të mesëm paraqesin strukturën dhe organizimin funksional të aplikacionit në një nivel modular. Këto diagrame paraqesin përmbajtjen dhe ndërveprimin e moduleve kryesore të sistemit.



* **Përshkrimi**: Ky diagram ilustron komponentët kryesorë si Menaxhimi i Përdoruesve, Menaxhimi i Porosive, Rent a Car, dhe Menaxhimi i Restoranteve. Secili modul përfaqëson një pjesë të dallueshme të funksionalitetit të sistemit, duke siguruar modularitet dhe ndarje të qartë të përgjegjësive.

3. Low-Level Design Models

Modelet e dizajnit të nivelit të ulët eksplorojnë më tej detajet e implementimit, duke përfshirë klasat individuale, atributet dhe operacionet e tyre, si dhe marrëdhëniet midis tyre. Këto diagrame ndihmojnë në kuptimin e mënyrës se si funksionojnë module të veçanta brenda sistemit.



* **Përshkrimi**: Ky diagram përqendrohet te klasat individuale dhe lidhjet e tyre, duke përfshirë klasa si User, Product, Order, Car, dhe Reservation. Atributet dhe operacionet e këtyre klasave janë modeluar për të përmbushur kërkesat funksionale të sistemit.

4. Mapping Between Models

Lidhja midis modeleve të nivelit të mesëm dhe të ulët siguron përputhshmëri midis strukturës modulare dhe implementimit të detajuar. Tabela më poshtë përfaqëson këtë lidhje:

|  |  |  |
| --- | --- | --- |
| Komponenti i Nivelit të Mesëm | Klasa/Paketa e Nivelit të Ulët | Përshkrimi |
| Menaxhimi i Përdoruesve | User, Role, Authentication | Menaxhon regjistrimin e përdoruesve, rolet dhe funksionalitetin e autentikimit. |
| Menaxhimi i Porosive | Order, OrderDetail, Product | Menaxhon krijimin e porosive, detajet e tyre dhe inventarin e produkteve. |
| Rent a Car | Car, Rent, ReservationLog | Menaxhon disponueshmërinë e makinave, procesin e marrjes me qira dhe rezervimet. |
| Menaxhimi i Restoranteve | Restaurant, Reservation, Review | Lehtëson listimin e restoranteve, rezervimet dhe komentet nga klientët. |

5. Detailed Design Rationale

Parimet e Aplikuara

1. **Modulariteti**: Sistemi është ndarë në module të veçanta si "Menaxhimi i Produkteve", "Rent a Car", dhe "Rezervimi i Restoranteve", për të siguruar mirëmbajtje të lehtë dhe të organizuar.
2. **Shkallëzueshmëria**:Ne projektin tone arkitektura mbeshtet shtimin e moduleve të reja pa ndikuar në funksionalitetet ekzistuese.
3. **Përdorshmëria**: Ka ndërfaqe te thjeshte për përdoruesit dhe adminët, e ndërtuar me qëllim të lehtësisë së përdorimit.

Arkitektura dhe Patterns e Përdorura

1. **MVC (Model-View-Controller)**: Ben ndarjen e logjikës së biznesit nga ndërfaqja e përdoruesit për të siguruar modularitet dhe mirëmbajtje të thjeshtë.
2. **Observer Pattern**: Përdoret për funksionalitete të notifikimeve, si njoftimi i përdoruesve për statusin e rezervimeve ose porosive.
3. **Factory Pattern**: Eshte implementuar për krijimin e objekteve si Car, Product, dhe User, duke siguruar një mënyrë të standardizuar dhe fleksibile krijimin e tyre.
4. **Singleton Pattern**: Aplikohet për komponentët e përbashkët si DataContext, që menaxhon qasjen në bazën e të dhënave në mënyrë të sigurt dhe të unifikuar.

Lidhjet midis Klasave

* **Agregimi**: Përfaqëson marrëdhënie pjesë-e-tërësisë, si Order që përmban disa OrderDetails.
* **Trashëgimia**: Përdorur për marrëdhënie tipologjike, si Admin dhe Customer që trashëgojnë nga klasa User.