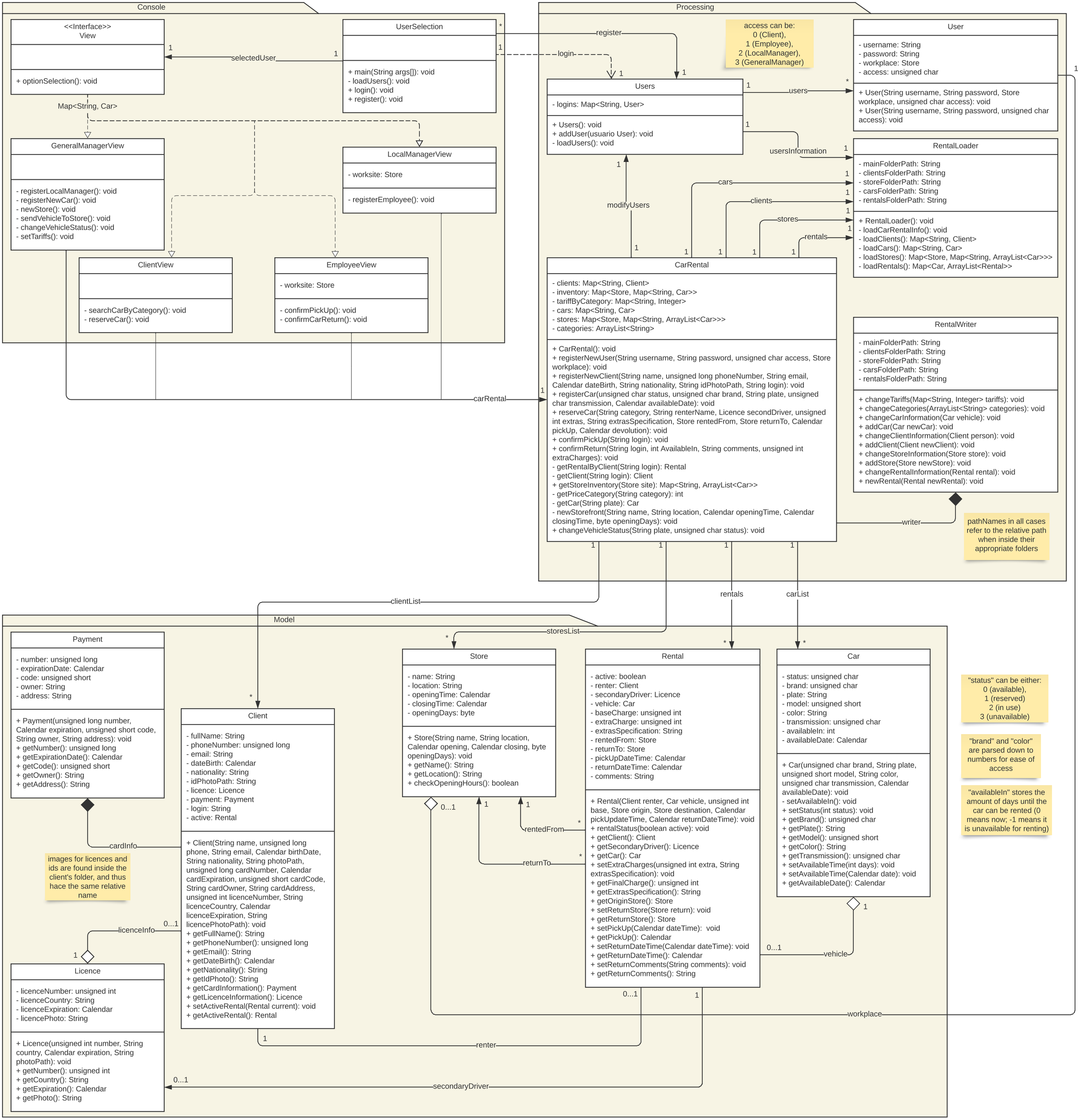
Console
<<Interface>>
View
+ optionSelection(): void
UserSelection
+ main(String args[]): void
- loadUsers(): void
+ login(): void
+ register(): void
selectedUser
Map<String, Car>
GeneralManagerView
- registerLocalManager(): void
- registerNewCar(): void
- newStore(): void
- sendVehicleToStore(): void
- changeVehicleStatus(): void
- setTariffs(): void
LocalManagerView
- worksite: Store
- registerEmployee(): void
ClientView
- searchCarByCategory(): void
- reserveCar(): void
EmployeeView
- worksite: Store
- confirmPickUp(): void
- confirmCarReturn(): void
Processing
register
login
access can be:
0 (Client),
1 (Employee),
2 (LocalManager),
3 (GeneralManager)
Users
- logins: Map<String, User>
+ Users(): void
+ addUser(usuario User): void
- loadUsers(): void
users
usersInformation
User
- username: String
- password: String
- workplace: Store
- access: unsigned char
+ User(String username, String password, Store workplace, unsigned char access): void
+ User(String username, String password, unsigned char access): void
RentalLoader
- mainFolderPath: String
- clientsFolderPath: String
- storeFolderPath: String
- carsFolderPath: String
- rentalsFolderPath: String
+ RentalLoader(): void
- loadCarRentalInfo(): void
- loadClients(): Map<String, Client>
- loadCars(): Map<String, Car>
- loadStores(): Map<Store, Map<String, ArrayList<Car>>>
- loadRentals(): Map<Car, ArrayList<Rental>>
modifyUsers
cars
clients
stores
rentals
CarRental
- clients: Map<String, Client>
- inventory: Map<Store, Map<String, Car>>
- tariffByCategory: Map<String, Integer>
- cars: Map<String, Car>
- stores: Map<Store, Map<String, ArrayList<Car>>>
- categories: ArrayList<String>
+ CarRental(): void
+ registerNewUser(String username, String password, unsigned char access, Store workplace): void
+ registerNewClient(String name, unsigned long phoneNumber, String email, Calendar dateBirth, String nationality, String idPhotoPath, String login): void
+ registerCar(unsigned char status, unsigned char brand, String plate, unsigned char transmission, Calendar availableDate): void
+ reserveCar(String category, String renterName, Licence secondDriver, unsigned int extras, String extrasSpecification, Store rentedFrom, Store returnTo, Calendar pickUp, Calendar devolution): void
+ confirmPickUp(String login): void
+ confirmReturn(String login, int AvailableIn, String comments, unsigned int extraCharges): void
- getRentalByClient(String login): Rental
- getClient(String login): Client
+ getStoreInventory(Store site): Map<String, ArrayList<Car>>
- getPriceCategory(String category): int
- getCar(String plate): Car
- newStorefront(String name, String location, Calendar openingTime, Calendar closingTime, byte openingDays): void
+ changeVehicleStatus(String plate, unsigned char status): void
RentalWriter
- mainFolderPath: String
- clientsFolderPath: String
- storeFolderPath: String
- carsFolderPath: String
- rentalsFolderPath: String
+ changeTariffs(Map<String, Integer> tariffs): void
+ changeCategories(ArrayList<String> categories): void
+ changeCarInformation(Car vehicle): void
+ addCar(Car newCar): void
+ changeClientInformation(Client person): void
+ addClient(Client newClient): void
+ changeStoreInformation(Store store): void
+ addStore(Store newStore): void
+ changeRentalInformation(Rental rental): void
+ newRental(Rental newRental): void
writer
pathNames in all cases refer to the relative path when inside their appropriate folders
carRental
clientList
rentals
carList
Model
storesList
Payment
- number: unsigned long
- expirationDate: Calendar
- code: unsigned short
- owner: String
- address: String
+ Payment(unsigned long number, Calendar expiration, unsigned short code, String owner, String address): void
+ getNumber(): unsigned long
+ getExpirationDate(): Calendar
+ getCode(): unsigned short
+ getOwner(): String
+ getAddress(): String
cardInfo
images for licences and ids are found inside the client's folder, and thus hace the same relative name
Client
- fullName: String
- phoneNumber: unsigned long
- email: String
- dateBirth: Calendar
- nationality: String
- idPhotoPath: String
- licence: Licence
- payment: Payment
- login: String
- active: Rental
+ Client(String name, unsigned long phone, String email, Calendar birthDate, String nationality, String photoPath, unsigned long cardNumber, Calendar cardExpiration, unsigned short cardCode, String cardOwner, String cardAddress, unsigned int licenceNumber, String licenceCountry, Calendar licenceExpiration, String licencePhotoPath): void
+ getFullName(): String
+ getPhoneNumber(): unsigned long
+ getEmail(): String
+ getDateBirth(): Calendar
+ getNationality(): String
+ getIdPhoto(): String
+ getCardInformation(): Payment
+ getLicenceInformation(): Licence
+ setActiveRental(Rental current): void
+ getActiveRental(): Rental
licenceInfo
Licence
- licenceNumber: unsigned int
- licenceCountry: String
- licenceExpiration: Calendar
- licencePhoto: String
+ Licence(unsigned int number, String country, Calendar expiration, String photoPath): void
+ getNumber(): unsigned int
+ getCountry(): String
+ getExpiration(): Calendar
+ getPhoto(): String
Store
- name: String
- location: String
- openingTime: Calendar
- closingTime: Calendar
- openingDays: byte
+ Store(String name, String location, Calendar opening, Calendar closing, byte openingDays): void
+ getName(): String
+ getLocation(): String
+ checkOpeningHours(): boolean
rentedFrom
returnTo
Rental
- active: boolean
- renter: Client
- secondaryDriver: Licence
- vehicle: Car
- baseCharge: unsigned int
- extraCharge: unsigned int
- extrasSpecification: String
- rentedFrom: Store
- returnTo: Store
- pickUpDateTime: Calendar
- returnDateTime: Calendar
- comments: String
+ Rental(Client renter, Car vehicle, unsigned int base, Store origin, Store destination, Calendar pickUpdateTime, Calendar returnDateTime): void
+ rentalStatus(boolean active): void
+ getClient(): Client
+ getSecondaryDriver(): Licence
+ getCar(): Car
+ setExtraCharges(unsigned int extra, String extrasSpecification): void
+ getFinalCharge(): unsigned int
+ getExtrasSpecification(): String
+ getOriginStore(): Store
+ setReturnStore(Store return): void
+ getReturnStore(): Store
+ setPickUp(Calendar dateTime): void
+ getPickUp(): Calendar
+ setReturnDateTime(Calendar dateTime): void
+ getReturnDateTime(): Calendar
+ setReturnComments(String comments): void
+ getReturnComments(): String
Car
- status: unsigned char
- brand: unsigned char
- plate: String
- model: unsigned short
- color: String
- transmission: unsigned char
- availableIn: int
- availableDate: Calendar
+ Car(unsigned char brand, String plate, unsigned short model, String color, unsigned char transmission, Calendar availableDate): void
- setAvailableIn(): void
+ setStatus(int status): void
+ getBrand(): unsigned char
+ getPlate(): String
+ getModel(): unsigned short
+ getColor(): String
+ getTransmission(): unsigned char
+ setAvailableTime(int days): void
+ setAvailableTime(Calendar date): void
+ getAvailableDate(): Calendar
"status" can be either:
0 (available),
1 (reserved)
2 (in use)
3 (unavailable)
"brand" and "color" are parsed down to numbers for ease of access
"availableIn" stores the amount of days until the car can be rented (0 means now; -1 means it is unavailable for renting)
vehicle
workplace
renter
secondaryDriver