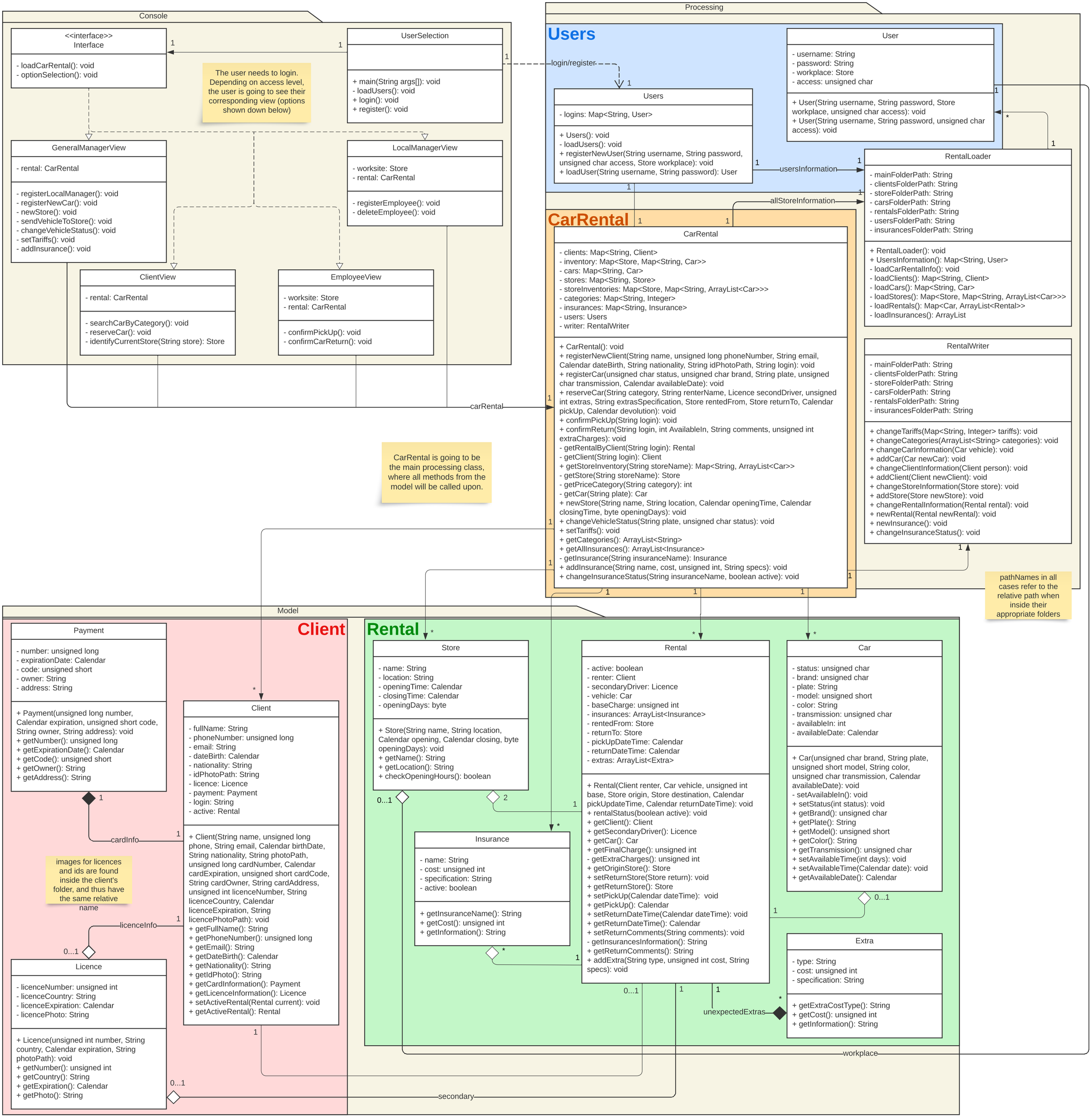
Console
<<interface>>
Interface
- loadCarRental(): void
- optionSelection(): void
UserSelection
+ main(String args[]): void
- loadUsers(): void
+ login(): void
+ register(): void
The user needs to login. Depending on access level, the user is going to see their corresponding view (options shown down below)
GeneralManagerView
- rental: CarRental
- registerLocalManager(): void
- registerNewCar(): void
- newStore(): void
- sendVehicleToStore(): void
- changeVehicleStatus(): void
- setTariffs(): void
- addInsurance(): void
LocalManagerView
- worksite: Store
- rental: CarRental
- registerEmployee(): void
- deleteEmployee(): void
ClientView
- rental: CarRental
- searchCarByCategory(): void
- reserveCar(): void
- identifyCurrentStore(String store): Store
EmployeeView
- worksite: Store
- rental: CarRental
- confirmPickUp(): void
- confirmCarReturn(): void
carRental
Processing
Users
login/register
Users
- logins: Map<String, User>
+ Users(): void
- loadUsers(): void
+ registerNewUser(String username, String password, unsigned char access, Store workplace): void
+ loadUser(String username, String password): User
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
usersInformation
allStoreInformation
RentalLoader
- mainFolderPath: String
- clientsFolderPath: String
- storeFolderPath: String
- carsFolderPath: String
- rentalsFolderPath: String
- usersFolderPath: String
- insurancesFolderPath: String
+ RentalLoader(): void
+ UsersInformation(): Map<String, User>
- loadCarRentalInfo(): void
- loadClients(): Map<String, Client>
- loadCars(): Map<String, Car>
- loadStores(): Map<Store, Map<String, ArrayList<Car>>>
- loadRentals(): Map<Car, ArrayList<Rental>>
- loadInsurances(): ArrayList
CarRental
CarRental
- clients: Map<String, Client>
- inventory: Map<Store, Map<String, Car>>
- cars: Map<String, Car>
- stores: Map<String, Store>
- storeInventories: Map<Store, Map<String, ArrayList<Car>>>
- categories: Map<String, Integer>
- insurances: Map<String, Insurance>
- users: Users
- writer: RentalWriter
+ CarRental(): 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(String storeName): Map<String, ArrayList<Car>>
- getStore(String storeName): Store
- getPriceCategory(String category): int
- getCar(String plate): Car
+ newStore(String name, String location, Calendar openingTime, Calendar closingTime, byte openingDays): void
+ changeVehicleStatus(String plate, unsigned char status): void
+ setTariffs(): void
+ getCategories(): ArrayList<String>
+ getAllInsurances(): ArrayList<Insurance>
- getInsurance(String insuranceName): Insurance
+ addInsurance(String name, cost, unsigned int, String specs): void
+ changeInsuranceStatus(String insuranceName, boolean active): void
RentalWriter
- mainFolderPath: String
- clientsFolderPath: String
- storeFolderPath: String
- carsFolderPath: String
- rentalsFolderPath: String
- insurancesFolderPath: 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
+ newInsurance(): void
+ changeInsuranceStatus(): void
CarRental is going to be the main processing class, where all methods from the model will be called upon.
pathNames in all cases refer to the relative path when inside their appropriate folders
Model
Client
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
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
cardInfo
licenceInfo
images for licences and ids are found inside the client's folder, and thus have the same relative name
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
secondary
Rental
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
Insurance
- name: String
- cost: unsigned int
- specification: String
- active: boolean
+ getInsuranceName(): String
+ getCost(): unsigned int
+ getInformation(): String
Rental
- active: boolean
- renter: Client
- secondaryDriver: Licence
- vehicle: Car
- baseCharge: unsigned int
- insurances: ArrayList<Insurance>
- rentedFrom: Store
- returnTo: Store
- pickUpDateTime: Calendar
- returnDateTime: Calendar
- extras: ArrayList<Extra>
+ 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
+ getFinalCharge(): unsigned int
- getExtraCharges(): unsigned int
+ getOriginStore(): Store
+ setReturnStore(Store return): void
+ getReturnStore(): Store
+ setPickUp(Calendar dateTime): void
+ getPickUp(): Calendar
+ setReturnDateTime(Calendar dateTime): void
+ getReturnDateTime(): Calendar
+ setReturnComments(String comments): void
- getInsurancesInformation(): String
+ getReturnComments(): String
+ addExtra(String type, unsigned int cost, String specs): void
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
Extra
- type: String
- cost: unsigned int
- specification: String
+ getExtraCostType(): String
+ getCost(): unsigned int
+ getInformation(): String
unexpectedExtras
workplace