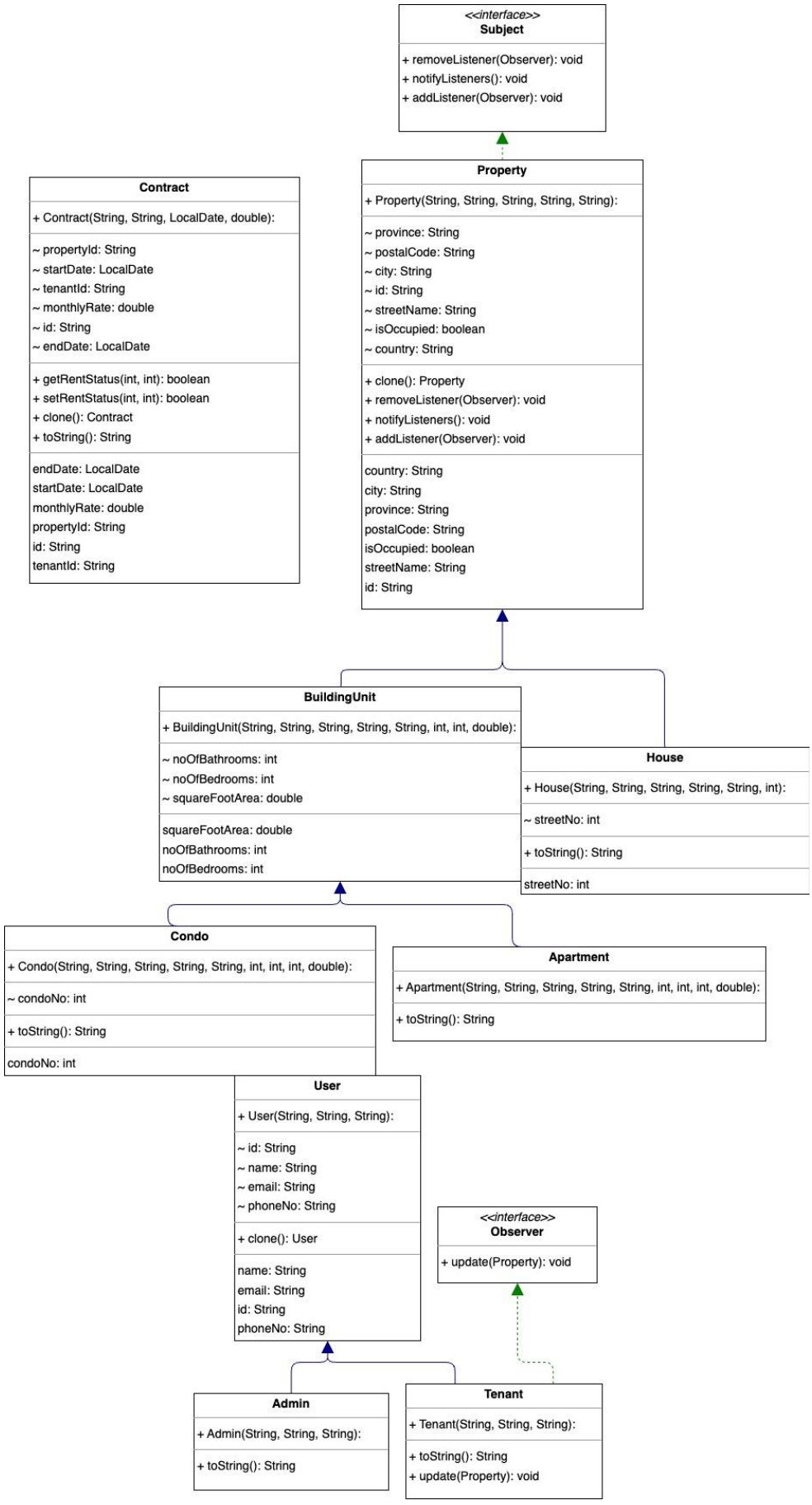
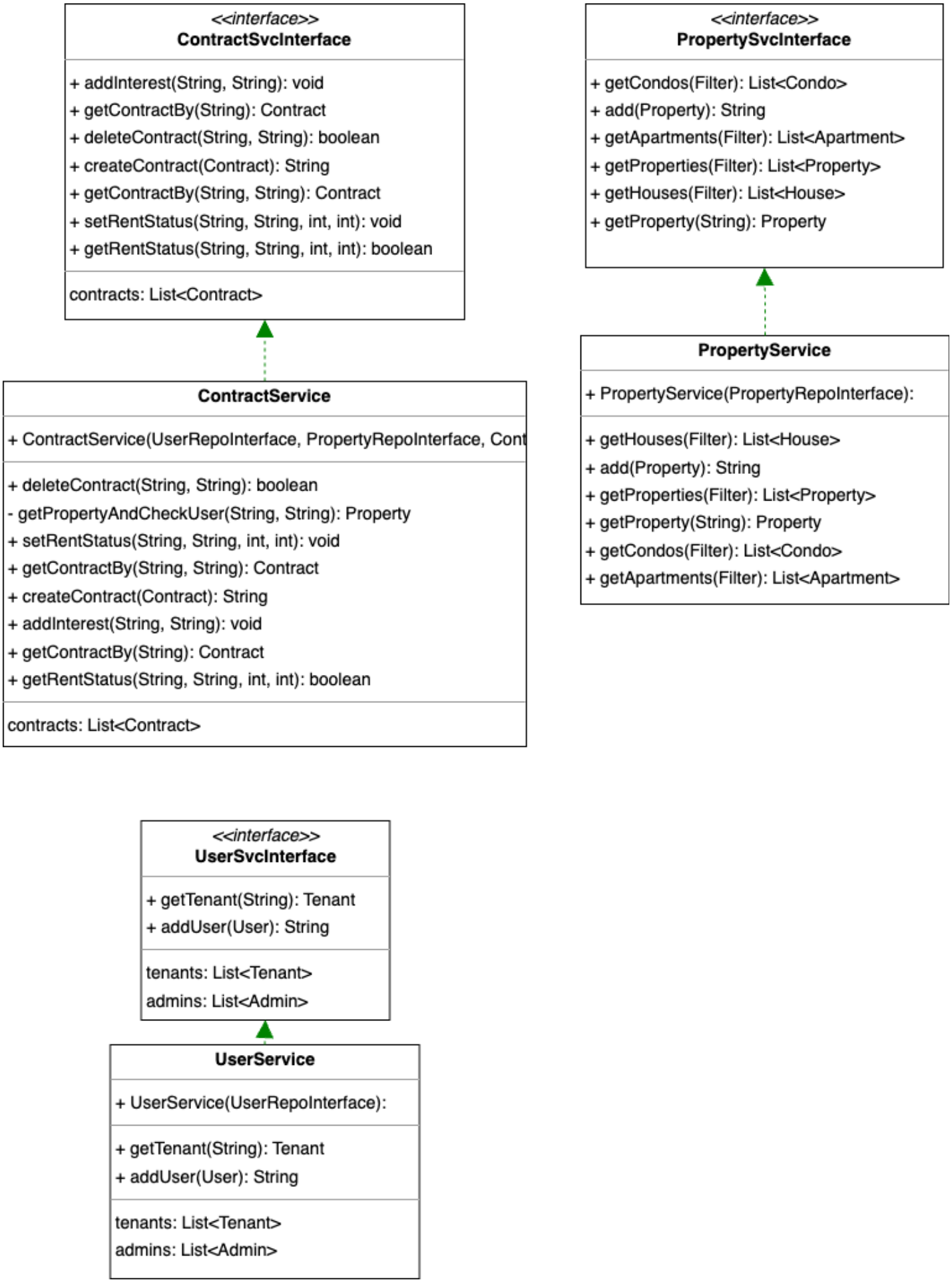
MODEL (With OBSERVER)



REPOSITORY

<<interface>>
ContractRepoInterface
+ getContract(String, String): Contract
+ isContractExists(String): boolean
+ getContract(String): Contract
+ addContract(Contract): String
+ deleteContract(String): boolean
+ updateContract(Contract): boolean
contracts: List<Contract>
<<enumeration>>
Filter
+ Filter():
+ values(): Filter[]
+ valueOf(String): Filter
ContractRepository
+ ContractRepository():
- contracts: Map<String, Contract>
+ getContract(String): Contract
+ deleteContract(String): boolean
+ isContractExists(String): boolean
+ addContract(Contract): String
+ getContract(String, String): Contract
+ updateContract(Contract): boolean
contracts: List<Contract>
<<interface>>
UserRepoInterface
+ isExists(String): boolean
+ delete(String): boolean
+ get(String): User
+ add(User): String
all: List<User>
<<interface>>
PropertyRepoInterface
+ updateProperty(String, Property): void
+ addProperty(Property): String
+ getProperty(String): Property
+ isPropertyExists(String): boolean
+ getAll(Filter): List<Property>
+ addInterest(String, Tenant): void
+ deleteProperty(String): boolean
+ notifyAll(String): void
UserRepository
+ UserRepository():
+ isExists(String): boolean
+ get(String): User
+ add(User): String
+ delete(String): boolean
all: List<User>
PropertyRepository
+ PropertyRepository():
+ deleteProperty(String): boolean
+ addProperty(Property): String
+ getProperty(String): Property
+ getAll(Filter): List<Property>
+ isPropertyExists(String): boolean
+ notifyAll(String): void
+ addInterest(String, Tenant): void
+ updateProperty(String, Property): void

SERVICE


<<interface>>
ContractSvcInterface
+ addInterest(String, String): void
+ getContractBy(String): Contract
+ deleteContract(String, String): boolean
+ createContract(Contract): String
+ getContractBy(String, String): Contract
+ setRentStatus(String, String, int, int): void
+ getRentStatus(String, String, int, int): boolean
contracts: List<Contract>
<<interface>>
PropertySvcInterface
+ getCondos(Filter): List<Condo>
+ add(Property): String
+ getApartments(Filter): List<Apartment>
+ getProperties(Filter): List<Property>
+ getHouses(Filter): List<House>
+ getProperty(String): Property
ContractService
+ ContractService(UserRepoInterface, PropertyRepoInterface, Cont
+ deleteContract(String, String): boolean
- getPropertyAndCheckUser(String, String): Property
+ setRentStatus(String, String, int, int): void
+ getContractBy(String, String): Contract
+ createContract(Contract): String
+ addInterest(String, String): void
+ getContractBy(String): Contract
+ getRentStatus(String, String, int, int): boolean
contracts: List<Contract>
PropertyService
+ PropertyService(PropertyRepoInterface):
+ getHouses(Filter): List<House>
+ add(Property): String
+ getProperties(Filter): List<Property>
+ getProperty(String): Property
+ getCondos(Filter): List<Condo>
+ getApartments(Filter): List<Apartment>
<<interface>>
UserSvcInterface
+ getTenant(String): Tenant
+ addUser(User): String
tenants: List<Tenant>
admins: List<Admin>
UserService
+ UserService(UserRepoInterface):
+ getTenant(String): Tenant
+ addUser(User): String
tenants: List<Tenant>
admins: List<Admin>

CONTROLLER

PropertyController
+ PropertyController(PropertySvcInterface):
+ addHouse(String, String, String, String, String, int): String + getAllCondos(Filter): List<Condo> + getAllApartments(Filter): List<Apartment> + addApartment(String, String, String, String, String, int, int, int, double): String + getAllHouses(Filter): List<House> + getAllProperties(Filter): List<Property> + addCondo(String, String, String, String, String, int, int, int, double): String

ContractController
+ ContractController(ContractSvcInterface):
+ getRentStatus(String, String, int, int): boolean + terminateContract(String, String): boolean + registerInterest(String, String): void + setRentStatus(String, String, int, int): void + rentUnit(String, String, LocalDate, double): String
leases: List<Contract>

UserController
+ UserController(UserSvcInterface):
+ addAdmin(String, String, String): String + addTenant(String, String, String): String
tenants: List<Tenant> admins: List<Admin>