This is a UML class diagram.

controller

C StrategyController

- △ RentalProperty rentalProperty
- △ Tenant tenant
- △ Lease lease

- ● String createRentalPropertyController(int,String,String,String,String,double,String)
- ● String addTenantController(String,String,List<String[]>)
- ● String rentAUnitController(String,String,int,String,String,String,String,String,String)
- ● List getAllPropertiesController()
- ● List getAllTenantsController()
- ● List getAllRentedUnitsController()
- ● List getAllVacantUnitsController()
- ● List getAllLeasesController()
- ● List getCustomerInterestPropertiesVacancyController()
- ● String updatePaymentStatusController(String,String,String,int,String,String,String,String)
- ● List getNonPaidRentalController()

(top-right class)

- ◇ int userChoice
- ◇ String input
- ◇ String address
- ◇ String city
- ◇ String state
- ◇ String postalCode
- ◇ double rent
- ◇ String civicAddress
- ◇ String apartmentNumber
- ◇ int numberOfApartments
- ◇ int numberOfBedrooms
- ◇ int numberOfBathrooms
- ◇ int squareFootage
- ◇ String streetNumber
- ◇ String unitNumber
- ◇ int propertyType
- ◇ String name
- ◇ String email
- ◇ String startDate
- ◇ String endDate
- ◇ String paymentStatus
- △ String[][] apartmentList
- △ ArrayList<String[]> interestedProperties
- △ StrategyController controller
- △ String serverResponse
- △ List list

- ● void main(String[])
- ● void printMenu()
- ● void setStrategy()
- ● String getInput()
- ● void executeStrategy()
- ● void getPropertyCommonDetails()
- ● void getPropertySpecificDetails()
- ● void getTenantDetails()
- ● void getPropertyIdentificationDetails()
- ● void getLeaseDetails()
- ● void displayAllLeases()
- ● void displayAllProperties()
- ● void displayAllTenants()
- ● void displayAllRentedUnits()
- ● void displayAllVacantUnits()
- ● void getInterestedPropertyList()
- ● void notifyVacantProperties()
- ● void getpaymentStatus()
- ● void displayNonPaidRentals()

JUnitPackage

C TestTenant
- ● void addTenant()
- ● void testAddTenant()
- ● void testDisplayTenant()

C TestLease
- △ Lease lease
- ● void addLease()
- ● void TestGetAllLeases()

java.util

C Observable

C Observer

database

C Database
- ○ List<Object> mockObject

C TestRentalProperty
- ● void addProperty()
- ● void rentAUnit()
- ● void addTenant()
- ● void testCreateRentalProperty()
- ● void testRentAUnit()
- ● void testDisplayProperties()
- ● void testDisplayRentedUnits()
- ● void testDisplayVacantUnits()
- ● void testGetCustomerInterestPropertiesVacancy()
- ● void testUpdatePaymentStatus()
- ● void getNonPaidRental()

rentals

C RentalProperty
- ○ String PROPERTY_RENTED
- ○ String PROPERTY_VACANT
- ○ String PROPERTY_ALL
- ○ String PAYMENT_PAID
- ○ String PAYMENT_UNPAID
- ○ String address
- ○ String city
- ○ String state
- ○ String postalCode
- ○ double rent
- ○ Tenant tenant
- ○ Lease lease
- ○ LeaseBuilder leaseBuilder
- ○ String rentalStatus
- ○ String paymentStatus
- △ Database database
- △ String response
- △ String filter

- ● String getAddress()
- ● String getCity()
- ● String getState()
- ● String getPostalCode()
- ○ double getRent()
- ● void displayDetails()
- ● String createRentalProperty(int,String,String,String,String,double,String)
- ● String rentAUnit(String,String,int,String,String,String,String,String)
- ● List getAllProperties()
- ● List getAllRentedUnits()
- ● List getAllVacantUnits()
- ● boolean isPropertyPresent(int,String,String,String,String)
- ● RentalProperty getPropertyDetails(int,String,String,String,String)
- ● List getCustomerInterestPropertiesVacancy()
- ● String updatePaymentStatus(String,String,String,int,String,String,String,String)
- ● List getNonPaidRental()

C Lease
- □ Tenant tenant
- □ RentalProperty rentalProperty
- □ Date startDate
- □ Date endDate
- □ double rentAmount
- □ boolean isPaid
- □ String leaseId
- □ String leaseType

- ● Tenant getTenant()
- ● Date getStartDate()
- ● Date getEndDate()
- ● double getRentAmount()
- ● boolean isPaid()
- ● String getLeaseId()
- ● String getLeaseType()
- ● void setPaid(boolean)
- ● boolean isExpired()
- ● List getAllLeases()
- ● void updateLeaseList(Lease)

C Tenant
- ◇ String name
- ◇ String email
- ◇ List<RentalProperty> interests
- ◇ List<Lease> leases

- ● String getName()
- ● String getEmail()
- ● void addInterest(RentalProperty)
- ● void removeInterest(RentalProperty)
- ● void addLease(Lease)
- ● void removeLease(Lease)
- ● void displayLeases()
- ● void displayInterests()
- ● void update(Observable,Object)
- ● String addTenant(String,String,List<String[]>)
- ● boolean isTenantRegistered(String,String)
- ● void updateTenantRentalList(Lease)
- ● List getAllTenants()

beans

C Apartment
- ○ String apartmentNumber
- ○ int numberOfBedrooms
- ○ int numberOfBathrooms
- ○ int squarefootage

C Condo
- ○ int streetNumber
- ○ int unitNumber
- ● void displayDetails()

C House
- ○ int streetNumber
- ● void displayDetails()

C LeaseBuilder
- ○ Tenant tenant
- ○ RentalProperty rentalProperty
- ○ Date startDate
- ○ Date endDate
- ○ double rentAmount
- ○ String leaseType