**Entities:**

* User(String name, String username, Address address, String password):
* Long userId;
* String name, username, password
* Address address;
* InsurancePolicy(String type, String policyholderUsername, int numberofClaims, Car car, MyDate expirtyDate):
* Long id;
* String insuranceType, policyHolderUsername;
* Int numberofClaims;
* User user;
* Car car;
* MyDate expiryDate;
* ComprehensivePolicy **extends** InsurancePolicy (super + driverAge, level)
* ThirdPartyPolicy **extends** InsurancePolicy (super + comments)
* Address(String address, String suburb, String state, int postcode)
* Car(CarType type, String model, int manufacturingYear, double carPrice) :
* enum CarType {SUV, SED, LUX. HATCH, TRUCK}
* MyDate(int day, int month, int year)

**Services:**

* InsurancePolicyService.java
* List<InsurancePolicy> getAllInsurancePolicies()

Retrieve all policies from db.

* InsurancePolicy getInsurancePolicyById (Long id)

Retrieve policy by policy’s id

* List<InsurancePolicy> getInsurancePolicyByType(String insuranceType)

Retrieve all insurance from specified type

* InsurancePolicy saveInsurancePolicy(InsurancePolicy insurancePolicy)

Save policy into the db.

* Static double calcTotalPayment(List<InsurancePolicy> policies, int flatRate)

Calculate total payment from a list of policies along with provided flat rate, return in double

* Double calcSinglePayment(Long policyId, int flatRate)

Calculate cost for single policy

* List<InsurancePolicy> getPoliciesByUsername(String username)

Retrieve all by policies owned by specified username

* InsurancePolicy createInsurancePolicy (InsurancePolicyDTO insurancePolicyDTO) throws InvalidCarTypeException

Create new insurance policy either comprehensive or third party