open util/boolean

sig string {}

sig Name, Surname{}

sig Email, Password{}

abstract sig User {

name: one Name,

surname: one Surname,

email: one Email,

password: one Password,

accessLevel: one Bool,

minedInfo: some MinedInfo,

}

sig EndUser extends User{

userLocation: one Location,

}{

accessLevel = False

}

sig Authority extends User{

tickets: Ticket,

}{

#tickets >= 1

accessLevel = True

}

sig Location {

latitude: one Int ,

longitude:one Int

}

sig Photo {}

sig Violation {

location: one Location,

addr: some ReverseGioCoding,

reporter: one EndUser,

//type: one string,

photo: one Photo,

licensePlate: one ALPR,

date: one Date

}

sig ReverseGioCoding {

loc: some Location,

//addr: one string

}{#ReverseGioCoding = 1}

sig ALPR { //remember to add somethin to tell it's only one

picture: some Photo ,

// licenseP: one string

}{ #ALPR = 1}

sig Date {}

abstract sig MinedInfo {

violations: some Violation,

}

sig MinedStreet extends MinedInfo{

/\*

// name: some string,

frequency: some Int,

location : one Location,\*/

}

sig MinedOffender extends MinedInfo{

/\*

n\_Violations: one Int,

licensePlate: one Int,

uuid: one Int\*/

}

sig Ticket {

violations: one Violation,

}

fact NoSameGPSForDifferentUsers {

no disjoint u1, u2 : EndUser |

u1.userLocation = u2.userLocation

}

fact NoSameGPSForDifferentReverseGio {

no disjoint revGio1, revGio2: ReverseGioCoding |

revGio1.loc = revGio2.loc

}

fact NoSamePhotoForDifferentViolation {

no disjoint v1,v2 : Violation |

v1.photo = v2.photo

}

fact NoSameViolationForDiffReporter{

no disjoint v1, v2 : Violation |

v1.reporter = v2.reporter

}

fact EachTicketOneAuthority {

all t: Ticket | one au: Authority |

au.tickets = t

}

//Two diﬀerent users can’t have the same email

fact NoSameEmailForDiﬀerentUsers {

no disjoint u1, u2 : User |

u1.password = u2.password

}

fact NoSamePasswordForDiﬀerentUsers {

no disjoint u1, u2 : User |

u1.email = u2.email

}

fact NoSameMinedInfoForDiﬀerentUsers {

all disjoint u1,u2: User |

u1.minedInfo != u2.minedInfo

}

fact EqualUserAndLocation{

#EndUser = #Location

}

//fact {all u: User | some n: Name | u.name = n}

//fact {all u: User | all n: Surname | u.surname = n}

fact EndUserRelateOnlyMinedStreet {

all user: EndUser |

user.minedInfo = MinedStreet

}

//fact { all mined: MinedInfo | one us: User | us.minedInfo = mined}

//fact { all us: User | some mined: MinedInfo | us.minedInfo = mined}

fact EachPhotoBelongsAlpr {

all ph: Photo|

one alpr: ALPR |

alpr.picture = ph

}

////

fact EqualLocationForEndUserAndGio {

one revGio: ReverseGioCoding|

one u: EndUser |

revGio.loc = u.userLocation

}

fact EqualLocationForViolationAndEndUser {

all viol: Violation |

some u:EndUser|

viol.location = u.userLocation

}

fact {all d: Date, viol: Violation | viol.date = d}

fact EachViolatioContainsOneTicket {

one t: Ticket , v:Violation |

t.violations = v

}

pred show {

}