**Constraints for task:** “Add or remove a destination(private or public) from an existing private destination list. For every added destination specify stay dates and a rating from 1 to 5” are:

1. Any list should have less than one item and no more than 100

context PrivateDestinationList:

inv: self.destinations -> size() > 0 and

self.destinations -> size() < 100

1. A private destination can be added in just one private destination list (for the same user)

context User:

inv: self.role=’user’ implies

self.role\_destinations -> forAll(d:Destination |

d.isKindOf(‘Private’) implies

d.pdl -> size() = 1

)

1. Any destination can be added only one time in the same list

context PrivateDestinationList::addDestination(destId:String):

pre: self.destinations -> forAll(d:Destination |

self.destinations -> count(d) = 1

)

1. The current number of destinations must be exactly one more than the number of destinations before the invocation of addDestination().

context PrivateDestinationList::addDestination(destId:String):

def number:

let number:Integer = self.destinations -> size()

post: number = number@pre + 1

1. The current number of destinations must be exactly one less than the number of destinations before the invocation of removeDestination().

context PrivateDestinationList::removeDestination(destId:String):

def number:

let number:Integer = self.destinations -> size()

post: number = number@pre - 1

1. The difference between stay dates for a destination must be less 1 day and no more than 1 month

context PrivateDestinationList:

inv: self.stayDates -> forAll(s: Tuple{startDate: Date, endDate: Date} |

s.endDate - s.startDate >= 1 and

s.endDate - s.startDate <= 30

)

1. In the same private list the duration of stays can not overlap

context PrivateDestinationList:

inv: self.destinations -> forAll(d1,d2:Destination |

d1 <> d2 implies

d1.stayDates -> forAll(s1: Tuple{startDate: Date, endDate: Date} |

d2.stayDates -> forAll(s2: Tuple{startDate: Date, endDate: Date} |

not (s1.startDate <= s2.endDate and s1.endData >= s2.startDate)

))

)