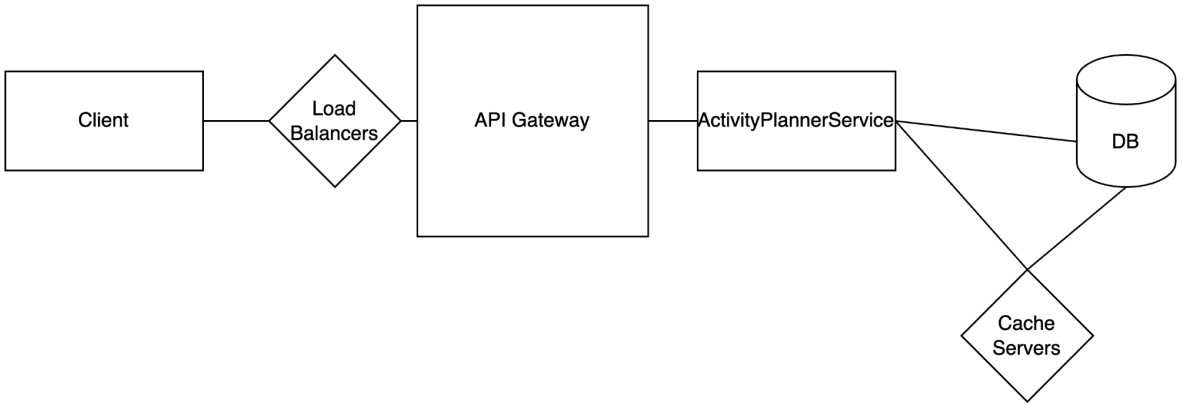
ACTIVITY PLANNER

HLD:



API Definitions:

API Endpoints	Method	Request Params	Request Body	Response
api/addPassenger	POST		{ "name":"P1", "passengerType": "STANDARD" }	
api/addTravelAgencyAdmin	POST		{ "name": "TA2" }	
api/addTravelDestination	POST		{ "destinationName" : "D1" }	
api/addTravelPackage	POST		{ "travelPackageName" : "T1", "itineraryName" : "I1",	

			"capacity" :10 }	

api/addItinerary	POST		{
			"itineraryName" :
			"I1",
			"destinationNames"
			: ["D1"]
			}	

api/getAllTravelPackage	GET			[
				{
				"id": 1,
				"travelPackageName": "T1",
				"itineraryName":
				"I1",
				"travelStartDate": {
				"year": 2023,
				"month":
				"AUGUST",
				"era": "CE",
				"dayOfYear": 240,
				"dayOfWeek":
				"MONDAY",
				"leapYear": **false**,
				"dayOfMonth": 28,
				"monthValue": 8,
				"chronology": {
				"id": "ISO",
				"calendarType":
				"iso8601"
				}
				},
				"travelEndDate": {
				"year": 2023,
				"month":
				"AUGUST",
				"era": "CE",
				"dayOfYear": 240,
				"dayOfWeek":
				"MONDAY",

				"leapYear": **false**,
				"dayOfMonth": 28,
				"monthValue": 8,
				"chronology": {
				"id": "ISO",
				"calendarType":
				"iso8601"
				}
				},
				"capacity": 10,
				"passengerListIds":
				null
				}
				]

api/getTravelPackage	GET	travelPackageId		{
				"id": 1,
				"travelPackageNam
				e": "T1",
				"itineraryName":
				"I1",
				"travelStartDate": {
				"year": 2023,
				"month":
				"AUGUST",
				"era": "CE",
				"dayOfYear": 240,
				"dayOfWeek":
				"MONDAY",
				"leapYear": **false**,
				"dayOfMonth": 28,
				"monthValue": 8,
				"chronology": {
				"id": "ISO",
				"calendarType":
				"iso8601"
				}
				},
				"travelEndDate": {
				"year": 2023,
				"month":
				"AUGUST",
				"era": "CE",

				"dayOfYear": 240,
				"dayOfWeek":
				"MONDAY",
				"leapYear": **false**,
				"dayOfMonth": 28,
				"monthValue": 8,
				"chronology": {
				"id": "ISO",
				"calendarType":
				"iso8601"
				}
				},
				"capacity": 10,
				"passengerListIds":
				[
				1
				]
				}

pi/addTravelPackageForPassenger	POST	userId travelPackageId		

api/addActivity	POST		{
			"destinationName" :
			"D1",
			"activityName":
			"A2"
			}	

api/getTravelPackageForPassenger	GET	userId
		travelPackageId
				{
				"id": 1,
				"travelPackageNam
				e": "T1",
				"itineraryName":
				"I1",
				"travelStartDate": {
				"year": 2023,
				"month":
				"AUGUST",
				"monthValue": 8,
				"chronology": {
				"calendarType":
				"iso8601",

				"id": "ISO"
				},
				"era": "CE",
				"dayOfYear": 240,
				"dayOfWeek":
				"MONDAY",
				"leapYear": **false**,
				"dayOfMonth": 28
				},
				"travelEndDate": {
				"year": 2023,
				"month":
				"AUGUST",
				"monthValue": 8,
				"chronology": {
				"calendarType":
				"iso8601",
				"id": "ISO"
				},
				"era": "CE",
				"dayOfYear": 240,
				"dayOfWeek":
				"MONDAY",
				"leapYear": **false**,
				"dayOfMonth": 28
				},
				"capacity": 10,
				"passengerListIds":
				[
				1
				]
				}

api/getAllTravelPackageForPassenger	GET	userId		[
				{
				"id": 1,
				"travelPackageName": "T1",
				"itineraryName":
				"I1",
				"travelStartDate": {
				"year": 2023,
				"month":

				"AUGUST",
				"era": "CE",
				"dayOfYear": 240,
				"dayOfWeek":
				"MONDAY",
				"leapYear": **false**,
				"dayOfMonth": 28,
				"monthValue": 8,
				"chronology": {
				"id": "ISO",
				"calendarType":
				"iso8601"
				}
				},
				"travelEndDate": {
				"year": 2023,
				"month":
				"AUGUST",
				"era": "CE",
				"dayOfYear": 240,
				"dayOfWeek":
				"MONDAY",
				"leapYear": **false**,
				"dayOfMonth": 28,
				"monthValue": 8,
				"chronology": {
				"id": "ISO",
				"calendarType":
				"iso8601"
				}
				},
				"capacity": 10,
				"passengerListIds":
				[
				1
				]
				}
				]

api/getAllActivities ForPassenger	GET	userId		[{ "id": 1, "destinationName":

				"D1", "activityName": "A2", "description": "Activity for destination D1", "capacity": 10, "cost": 100.0 }]
api/addActivityForPassenger	POST	userId activityName		
api/addBalanceForPassenger	POST	userId balance		

api/getPassengerDetails	GET	userId		{
				"name": "P1",
				"passengerNumber": 1,
				"balance": 170.0,
				"activityList": [
				{
				"id": 1,
				"destinationName":
				"D1",
				"activityName":
				"A2",
				"description":
				"Activity for
				destination D1",
				"capacity": 10,
				"cost": 100.0
				}
				]
				}

api/getAvailableActivities	GET	destinationName		{ "destinationName": "D1", "activitiesList": [{ "activityName":

				"A2", "count": 9 }] }

api/getTravelPackagePassengerDetails	GET	travelPackageId		{
				"travelPackageName": "T1",
				"capacity": 10,
				"numberOfPassengers": 1,
				"passengerList": [
				"P1"
				]
				}

api/getTravelPackageItineraryDetails	GET	travelPackageId		

DATABASE:

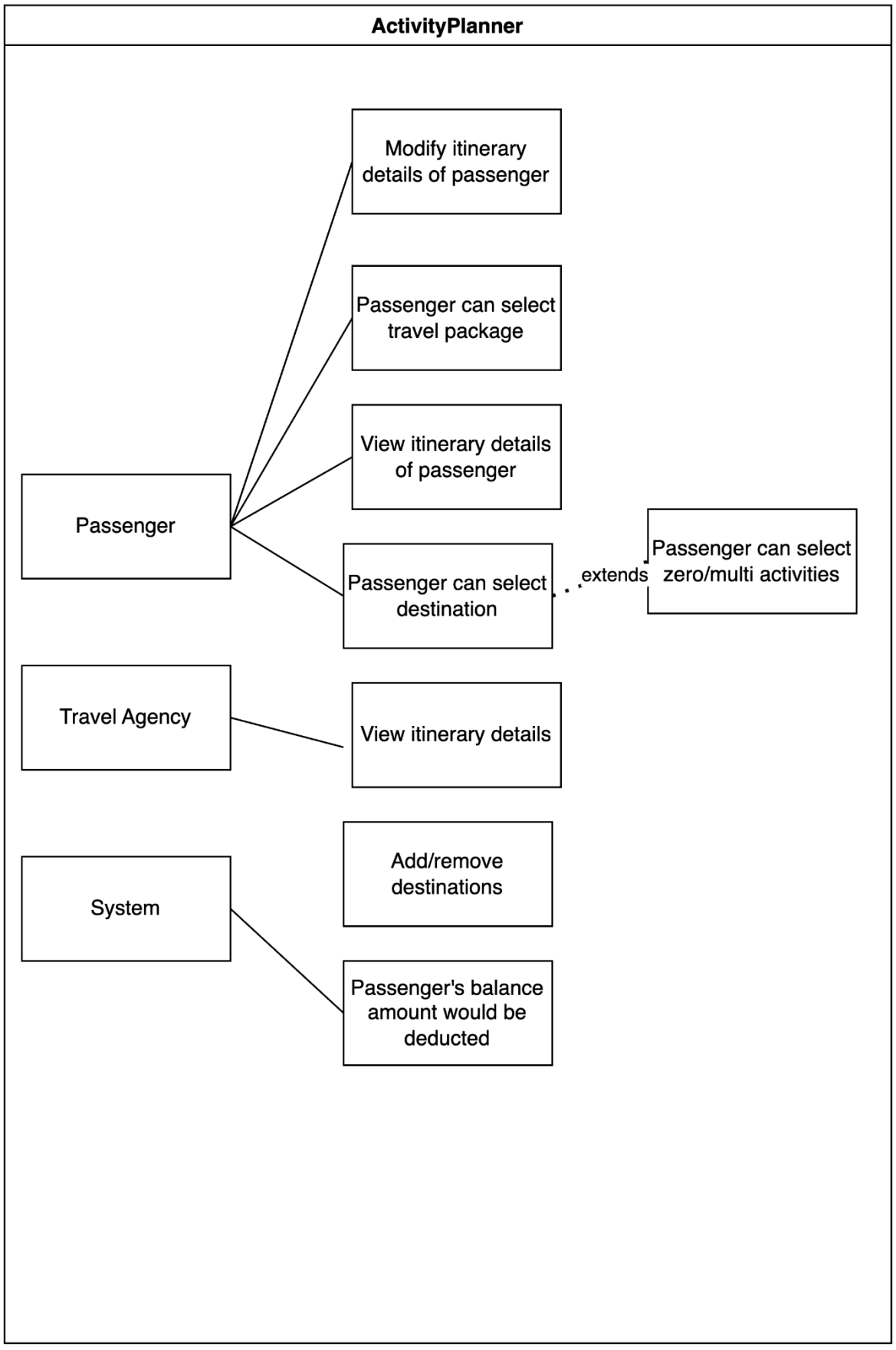
PostgresSQL

LLD:

Requirements:

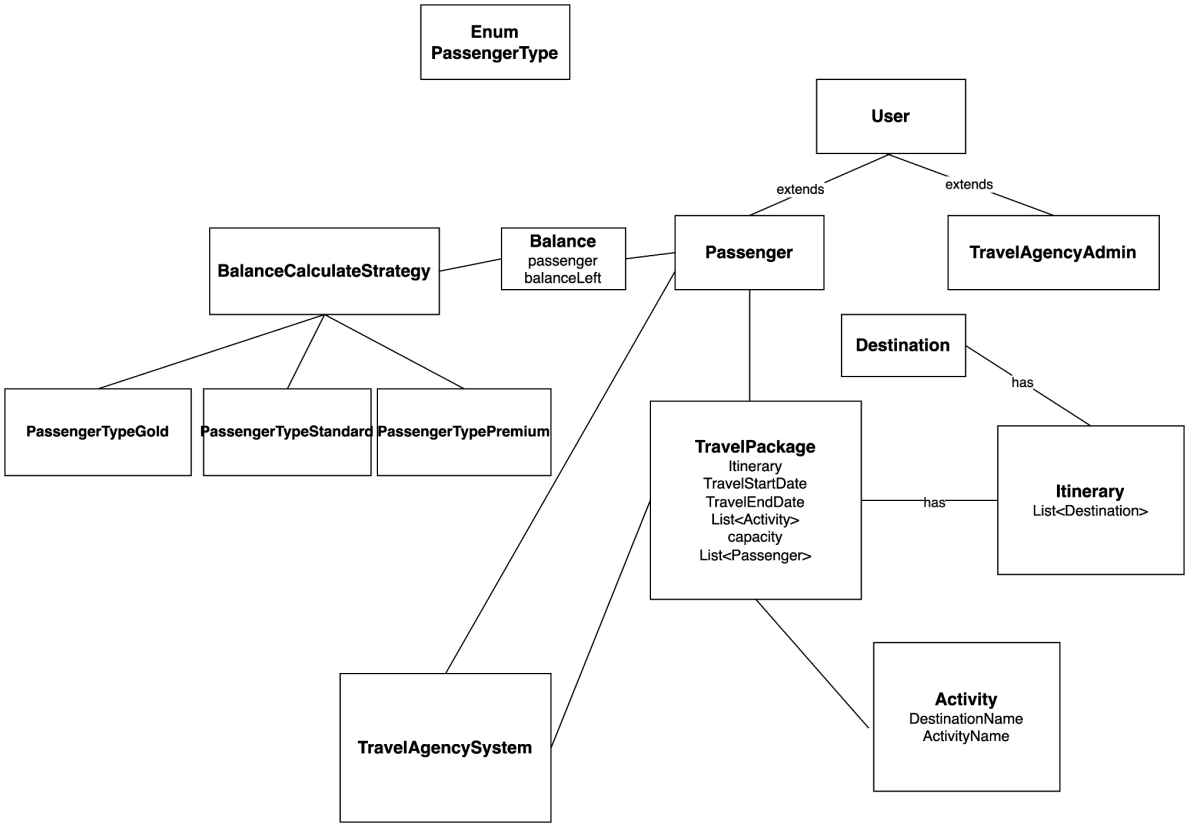
1. Passenger will be able to select a travel package
2. Travel package would have destination
3. Destination would have a list of activities
4. Passenger would be able to sign up for activity/list of activities
5. Passenger's balance would be deducted based on the passenger's type
6. Travel agency/passenger can view itinerary details
7. Passenger can modify itinerary details
8. Travel agency can add/remove destinations

Use-cases diagram:


ActivityPlanner
Modify itinerary details of passenger
Passenger can select travel package
View itinerary details of passenger
Passenger
Passenger can select destination
extends
Passenger can select zero/multi activities
Travel Agency
View itinerary details
System
Add/remove destinations
Passenger's balance amount would be deducted

Classes and Enums:

1. User
2. Passenger
3. Travel Agency admin
4. Itinerary
5. TravelPackage
6. Source
7. Destination
8. Activity
9. Balance
10. PassengerType (Enum)
11. ActivityPlannerSystem



Additional Requirements:

1. Print itinerary of the travel package including:
    1. travel package name,
    2. destinations and details of the activities available at each destination, like name, cost, capacity and description. -

       API to get this - **api/getTravelPackageItineraryDetails**

2. Print the passenger list of the travel package including:
    1. package name,
    2. passenger capacity,
    3. number of passengers currently enrolled and
    4. name and number of each passenger

        API to get this- **api/getTravelPackagePassengerDetails**

3. Print the details of an individual passenger including their
    1. name,
    2. passenger number,
    3. balance (if applicable),
    4. list of each activity they have signed up for, including the destination the at which the activity is taking place and the price the passenger paid for the activity.

        API to get this- **api/getPassengerDetails**

4. Print the details of all the activities that still have spaces available, including how many spaces are available.

        API to get this- **api/getAvailableActivities**