FEUP – LEIC
2021/2022

TRAVEL AGENCY SERVICE

DATABASE PROJECT

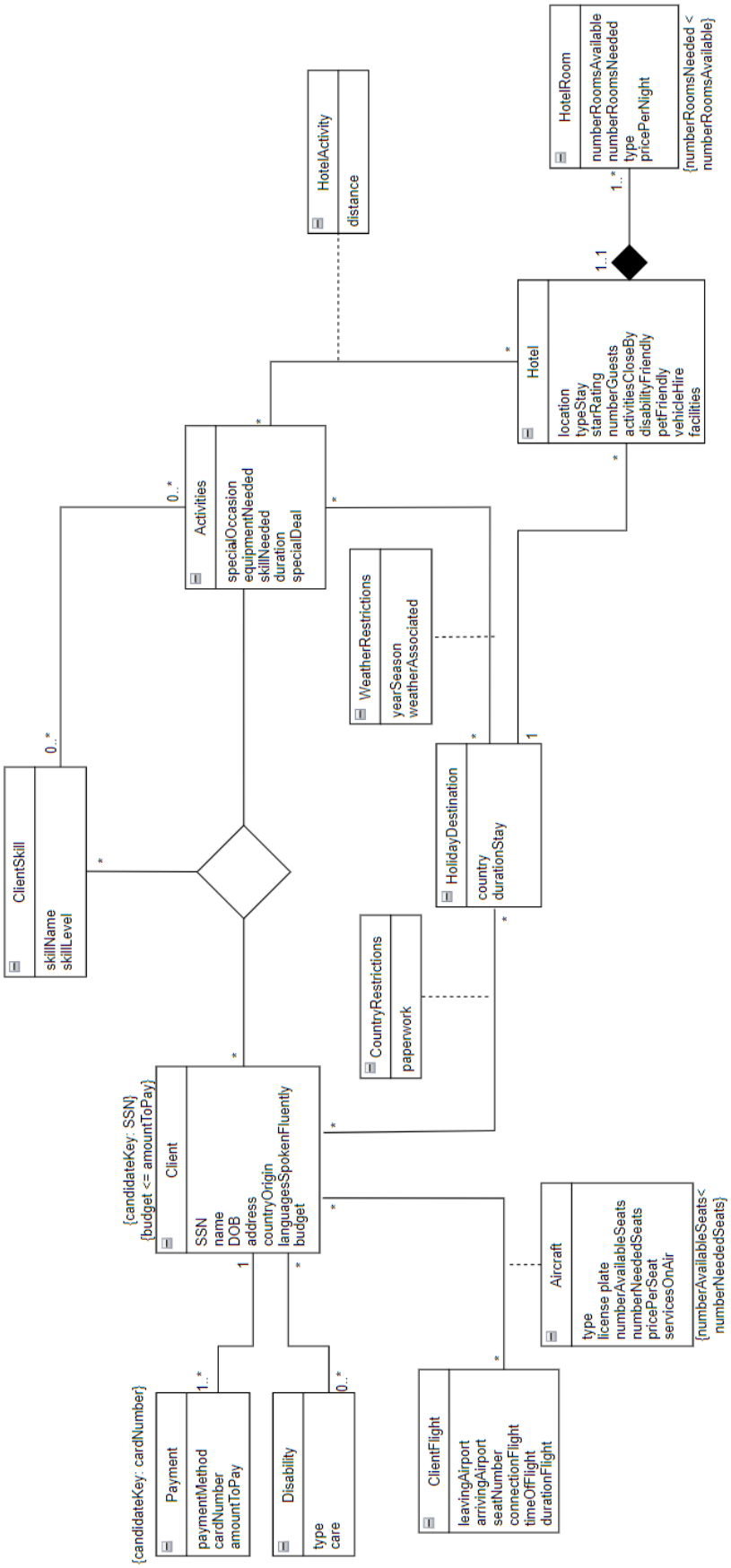
CREATED BY RAFAELA DUARTE (202006757), JOAO LOPES (202007797)
AND SIMAO RODRIGUES (202005700)

CONTENTS

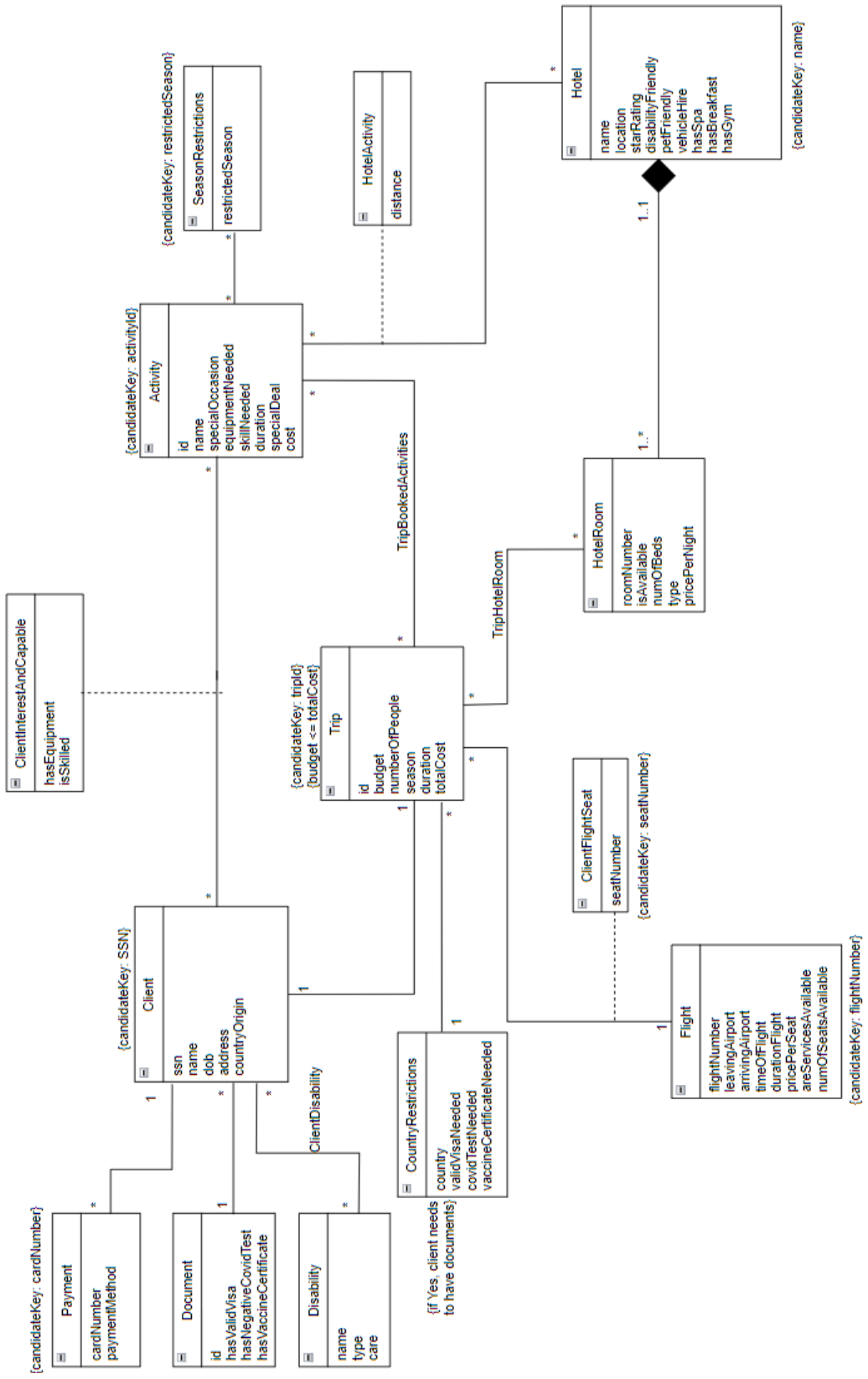
PROJECT CONTEXT:

As a new travel agency that is ready to launch soon, its job is to obtain and store their clients' details (like SSN, name, address, budget for the trip; along with all the necessary payment details –card number, etc.), and then successfully plan their dream holiday.
To start with, the client will be asked where they want to travel to, and how long they wish to stay. Although the destination is chosen by the client, there may be restrictions that the client needs to comply with – these can be related to COVID-19. Travelers may need to show a vaccine certificate or a negative result test to be allowed to enter the country. Countries like the UK have also banned certain countries from flying over, and other passengers may need to quarantine upon arrival. All paperwork's the clients' responsibility, and the agency cannot be at fault if the holiday cannot be enjoyed due to this.
Next, the agency needs to know what activities clients enjoy doing while on holiday, along with skills they have. If a client doesn't have a specific skill needed for an activity, help may be offered by the activity's staff. This makes sure everyone will enjoy the activities without risk. The same principle applies to equipment – even if provided by the company, it could be dangerous to use without skill.
Activities can also be filtered in terms of what kind of holiday it is: a special occasion (wedding, stag/hen party, birthday party), a family event (involving children, older people, or pets), or some simple time away (work escape, unwinding). This can then lead to special deals like cheaper activities for kids, extra time for half price, or even complementary drinks at arrival. In some cases, some activities may not be available due to the season of the year and the weather associated. If a family is looking for a holiday at the beach, then travelling to a country during the winter can be a bad option. The client has to either compromise (e.g. Choose a different location) or travel at a different time.
Lastly, the agency requires records of any disabilities the client has. These records will help deciding on a hotel and the best suitable activities for those people, so they can enjoy their holiday to the fullest.
The client will then have a choice about where to stay. The hotel is determined by the distance between the main activities selected and where the place is located. The client can then choose from a selection of places within that area. The search can be filtered down by selecting certain things – e.g. A place that is pet friendly; has wheelchair access; if it includes facilities like spa and gym. When the perfect place is found, room(s) is/are picked out. If there are not enough rooms left, or the filtered search came up with no results, then the client needs to rethink their choices.
The last thing to consider are flights. Which flight(s) is/are booked will depend on location, ie. closest airport to the hotel. For each flight, the client will know which plane to catch and where, where and when it'll land, and the seat number. If the flight's time is changed or the flight's cancelled altogether, the agency will contact the client and come up with a solution.
The trip can only be booked if the client has enough to pay for it – i.e.. the price of the whole trip is within their budget. The whole trip will be paid through the travel agency, and the client will enjoy their holiday as a packet – flight/hotel/activities included. Additional expenses throughout their adventure will be paid separately at the clients' responsibility.

UML DIAGRAM (1st SUBMISSION):



UML DIAGRAM (REVISED DIAGRAM):



RELATIONAL MODEL:

Client(ssn, name, dob, address, countryOrigin, id -> Document)
Payment(cardNumber, paymentMethod, ssn -> Client)
Document(id, hasValidVisa, hasNegativeCovidTest, hasVaccineCertificate)
Disability(name, type, care)
Trip(id, country -> CountryRestrictions, budget, numOfPeople, season, duration, totalCost, ssn -> Client)
Flight(flightNumber, leavingAirport, arrivingAirport, timeOfFlight, durationFlight, pricePerSeat, areServicesAvailable, numOfSeatsAvailable)
Hotel(name, location, starRating, disabilityFriendly, petFriendly, vehicleHire, hasSpa, hasBreakfast, hasGym)
HotelRoom(roomNumber, isAvailable, id -> Trip, numOfBeds, type, pricePerNight, name -> Hotel)
Activity(id, name, specialOccasion, equipmentNeeded, skillNeeded, duration, specialDeal, cost)
CountryRestrictions(country, validVisaNeeed, covidTestNeeded, vaccineCertificateNeeded)
ClientFlightSeat(id -> Trip, flightNumber -> Flight, seatNumber)
SeasonRestriction(id -> Activity, restrictedSeason)
ClientDisability(ssn -> Client, name -> Disability)
TripBookedActivities(id -> Activity, id -> Trip)
HotelActivity(id -> Activity, name -> Hotel, distance)
ClientInterestedAndCapable(ssn -> Client, id -> Activity, hasEquipment, isSkilled)
TripHotelRoom(roomNumber -> HotelRoom, id -> Trip)

FUNCTIONAL DEPENDENCIES & NORMAL FORMS:

Client(ssn, name, dob, address, countryOrigin, id -> Document)

{ssn -> name, dob, address, countryOrigin, id}

The relation is in BCNF and in 3NF because ssn is a key of Client.

There cannot be two clients with the same name or ssn. A client must have a name, ssn, and country.

Payment(cardNumber, paymentMethod, ssn -> Client)

{cardNumber -> paymentMethod, ssn}

The relation is in BCNF and in 3NF because cardNumber is a key of Payment.

There cannot be two payment information details with the same card number. A payment information must have a card number, a payment method and a ssn which belongs to an existing client.

Document(id, hasValidVisa, hasNegativeCovidTest, hasVaccineCertificate)

{id -> hasValidVisa, hasNegativeCovidTest, hasVaccineCertificate}

The relation is in BCNF and in 3NF because id is a key of Document.

There cannot be two different documents with the same id. A document must have an id and say if the client has a valid visa, a negative covid test and a vaccine certificate.

Disability(name, type, care)

{name -> type, care}

The relation is in BCNF and in 3NF because name is a key of Disability.

There cannot be two different disabilities with the same name. A disability must have a name, a type and a needed care.

Trip(id, country -> CountryRestrictions, budget, numOfPeople,, season, duration, totalCost, ssn -> Client)

{id -> country, budget, numOfPeople, season, duration}

The relation is in BCNF and in 3NF because id is a key of Trip.

There cannot be two trips with the same id. A trip must have an id, an ssn which belongs to an existing client, a budget which must be a positive number, a number of people equal to or greater than 1, a duration which must be equal to or greater than 1 hour and a total cost which must be a positive number.

Flight(flightNumber, leavingAirport, arrivingAirport, timeOfFlight, durationFlight, pricePerSeat, areServicesAvailable, numOfSeatsAvailable)

{flightNumber -> leavingAirport, arrivingAirport, timeOfFlight, durationFlight, pricePerSeat, areServicesAvailable, numOfSeatsAvailable}

The relation is in BCNF and in 3NF because flightNumber is a key of Flight.

There cannot be two flights with the same flight number. A flight must have a flight number, an airport which it leaves from and one where it arrives, the time the flight is at, the duration of the flight which must be greater than 1 minute, the price per seat which must be greater than 1 euro, information

on whether there are inflight services available, and a number of available seats equal to or greater than 0.

Hotel(name, location, starRating, disabilityFriendly, petFriendly, vehicleHire, hasSpa, hasBreakfast, hasGym)

{name -> location, starRating, disabilityFriendly, petFriendly, vehicleHire, hasSpa, hasBreakfast, hasGym}

The relation is in BCNF and in 3NF because name is a key of Hotel.

There cannot be two hotels with the same name. A hotel must have a name, a location, a star rating between 1 and 5 stars and information on whether it is disability/pet friendly, has vehicles for renting, has a spa/gym and offers breakfast.

HotelRoom(roomNumber, isAvailable, id -> Trip, numOfBeds, type, pricePerNight, name -> Hotel)

{roomNumber -> isAvailable, id, numOfBeds, type, pricePerNight, name}

The relation is in BCNF and in 3NF because roomNumber is a key of HotelRoom

There cannot be two hotel rooms in the same hotel with the same room number. A hotel room must have a room number, belong to a hotel, say whether it is available and have information on the number of beds, the type of the room and the price per night. If the room has been booked, it will also have the id of the trip it was booked from.

Activity(id, name, specialOccasion, equipmentNeeded, skillNeeded, duration, specialDeal, cost)

{id -> name, specialOccasion, equipmentNeeded, skillNeeded, duration, cost}

The relation is in BCNF and in 3NF because id is a key of Activity.

There cannot be two activities with the same id. An activity must have an id, a name, say whether equipment and skills are needed, say the duration and the cost.

CountryRestrictions(country, validVisaNeeded, covidTestNeeded, vaccineCertificateNeeded)

{country -> validVisaNeeded, covidTestNeeded, vaccineCertificateNeeded}

The relation is in BCNF and in 3NF because country is a key of CountryRestrictions.

There cannot be two country restrictions for the same country. A country restriction must have a country and say whether a valid visa, negative covid test, or vaccine certificate is needed.

ClientFlightSeat(id -> Trip, flightNumber -> Flight, seatNumber)

{seatNumber, flightNumber -> id}

The relation is in BCNF and in 3NF because {seatNumber, flightNumber} is a key of ClientFlightSeat.

There cannot be two client seat bookings with the same flight number and seat number. A client flight seat booking must have a seat number and an id to the trip it belongs to, plus it must have the number of the flight it's on.

SeasonRestriction(id -> Activity, restrictedSeason)

The relation has no FDs because all its attributes are keys. Therefore, it is in BCNF and in 3NF.

There cannot be two season restrictions for the same activity and the same season. A season restriction must say the id of the activity it's for and what season it can't be performed on.

ClientDisability(ssn -> Client, name -> Disability)

The relation has no FDs because all its attributes are keys. Therefore, it is in BCNF and in 3NF.

There cannot be two client disability entries for the same client and same disability. A client disability entry must have the ssn of the client and the name of his disability.

TripBookedActivities(id -> Activity, id -> Trip)

The relation has no FDs because all its attributes are keys. Therefore, it is in BCNF and in 3NF.

The same activity cannot be booked twice for the same trip. A booked activity must have the id of said activity and the id of the trip it was booked for.

HotelActivity(id -> Activity, name -> Hotel, distance)

{id, name -> distance}

The relation is in BCNF and in 3NF because {id, name} is a key of HotelActivity.

There cannot be two activities with the same id for a hotel. An activity for a hotel must have the distance from the hotel, the name of the hotel and the id of the activity.

ClientInterestedAndCapable(ssn -> Client, id -> Activity, hasEquipment, isSkilled)

{ssn, id -> hasEquipment, isSkilled}

The relation is in BCNF and in 3NF because {ssn, id} is a key of ClientInterestedAndCapable.

There cannot be two entries for the same activity and client. An entry must say what client and activity it relates to, and say whether or not the client has equipment and skills for that activity.

TripHotelRoom(roomNumber -> HotelRoom, id -> Trip)

The relation has no FDs because all its attributes are keys. Therefore, it is on BCNF and in 3NF.

The same hotel room cannot be booked twice for the same trip. A booked hotel room must have its unique room number and the id of the trip it was booked in.

QUERIES:

1- Find the cheapest room per hotel.
2- Find all hotel names.
3- Find the average duration of a flight.
4- Find the document status (valid visa, negative covid test and vaccine certificate) for each client.
5- Find the number of countries that require a valid visa, negative covid test or vaccine certificate per value.
6- Find the room occupation percentage per hotel.
7- Find the client/s with the highest budget.
8- Find the profit (sum of the cost per participant) for every activity.
9- Find the number of rooms per hotel.
10- Find the total cost per trip.

TRIGGERS:

1- When a client books a hotelRoom, add the cost of the hotel room to the client's trip's total cost.
2- When a country's restrictions are changed, delete the trips of the clients who don't have the new required documents.
3- When a client tries to book an activity, stop them if they activity can't be booked for the season their trip is booked for.

PARTICIPATION EVALUATION:

The project was completed by Rafaela Duarte (up202006757), Joao Lopes (up202007797) and Simao Rodrigues (up202005700).
The SQL programming part of the project was completed by Joao Lopes, and then reviewed by the group.
The UML diagram was drawn by Rafaela Duarte, and then also reviewed by the group.
The report was written and put together by Rafaela Duarte and Joao Lopes, and then reviewed multiple times by the group.
Overall, our teamwork allowed us to finish the project with some ease by giving each other meaningful feedback on our tasks.