User Needs

User Needs

As a	I want to	So that
business owner	get all the tours list	I can know how many tours I have
business owner	add one tour	I can read and update it later
business owner	delete one tour	I can remove past tours or unnecessary tours
business owner	update one specific tour	I can change the error
business owner	keep the information of my customers(tourists)	I can manage my customer, keep in touch and maybe send the promote sales to them
business owner	know how many tourists join a specific tour	I can arrange a suitable bus to carry all the tourists
business owner	know how many tours does a specific tourist joined	I can caculate the right fee
business owner	know for a specific tour, how many seats are still available	I can do more promoting if the tour didn't sell very well
business owner	know how much sold	I can let the other shareholders know

Software Requirement

Tour Agency

Nan is the boss of a tour agency in Edinburgh. He built his business 6 years before. As the business grows, he is struggling with managing his business using a pen and notebook. He heard that you are an expert of software and wondered if you could build a piece of software to help him manage his business bookings and logic.

MVP(Basic requirements)

1. The app should allow CRUD tours. Tours may have the properties like name, start date, how long(1 day, 2 days or 3 days or even more), price, (option img url).
2. The app should allow CRUD tourists. Tourists may have the properties like name, phone number, email address, home address. (in case the agency may need to send a bill to a tourist!)
3. The app should allow tourists to book their tours. One tourist can book many tours. And one tour can have many tourists.
4. The app should allow to show all tourists that are booked in for a particular tour.
5. The app should allow to show all tours that a particular tourist booked.

Extensions(further requirements)

1. Booking a tour should decrease the funds of the tourist by the price. (add the property of funds in Class Tourist)
2. Check how much a tourist needs to pay.
3. Check how many tourists are going to join a particular tour.

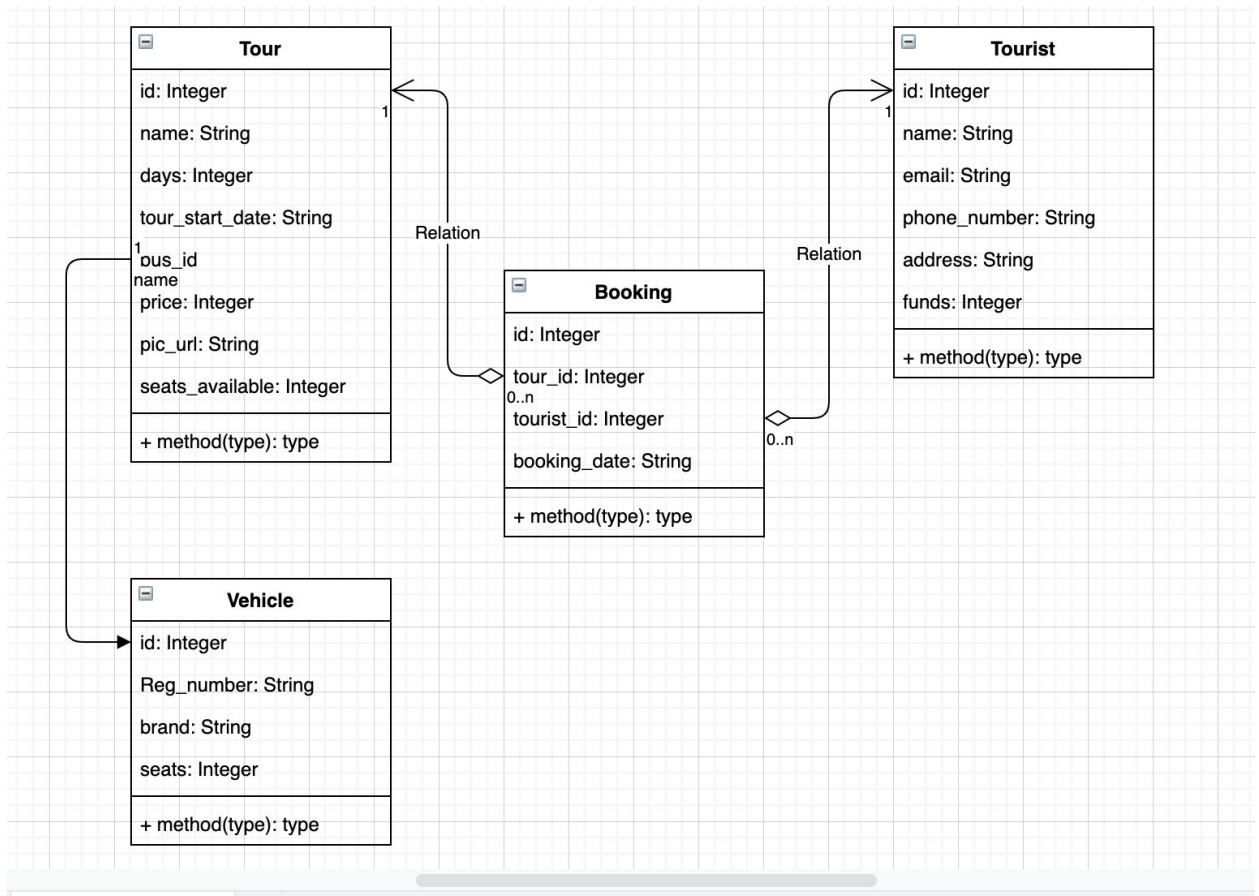**Advanced Extensions (advanced requirements)**

1. Most of the tours may need a suitable vehicle to carry all the tourists. Vehicles may have limited seats. Create a class vehicle holding the properties like registration number, number of seats.
2. Nan wishes to check how many seats are left for a particular tour, so he may do some promoting like give tourists 10% discounts if the available seats below 10% of all the seats.

More than Advanced Extensions (greedy requirements)

1. Nan wishes to know the total money of all the bookings!

2. If you wish to update more than one table in one action, how to avoid some tables updated but the others didn't?
For example, when tourists book a tour. Their funds should be reduced. Booking should be generated. And the available seats for a particular tour should decrease. But what happens if the funds of tourists are reduced, but no booking? You will receive complaints from the tourists!

User Journey

User action	User action	User action	User action
type the home page of the website	when click the tour link	click the add new link	click the delete button
1 Start	**2**	**3**	**4**
System response	System response	System response	System response
show the welcome page, there should have a nav bar	display all the tours information	jump to a new page , there should have a form allow user to input the tour information	remove that tour and redirect to the tours page
User action	User action	User action	User action
click the update link	click the check tourists button	in tourists page, click a specifical tourist, check tours button	
5	**6**	**7**	**8 End**
System response	System response	System response	System response
jump to a new page, display the detail of the tour and can allow user to update	jump to a new page and display all the tourist who joined in this tour	jump to a new page and display all the tours which booked by this tourists	

Class Diagram

Tour
- id: Integer
- name: String
- days: Integer
- tour_start_date: String
- bus_id
- name
- price: Integer
- pic_url: String
- seats_available: Integer
- + method(type): type

Tourist
- id: Integer
- name: String
- email: String
- phone_number: String
- address: String
- funds: Integer
- + method(type): type

Booking
- id: Integer
- tour_id: Integer
- tourist_id: Integer
- booking_date: String
- + method(type): type

Vehicle
- id: Integer
- Reg_number: String
- brand: String
- seats: Integer
- + method(type): type

Demo time

Code proud of

Complex SQL functions suit this case and how to use(call) in ruby.

Learning points

Fast learning, self-learning, learning from others, when and how to ask who.