Chapter III

**Analysis and Design**

The Transit Reservation System is implemented using the python programming language. The django web framework has been used for the purpose of easier organization of files that consitute the system.

The waterfall method was used to develop the system. This followed the following steps:

Planning, Analysis, Design, Implementation, Maintenance

**Analysis**

In the analysis of the transit reservation system, the following were the requirements gathered.

* The system should enable a user to book a seat for a trip
* If all the seats in a certain trip are booked then the trip should be unavailable
* Each trip should have fourteen seats.
* The system should validate payments before a user can book a seat
* The user should be able to view a ticket after he or she has booked a seat
* The user should be able to download his/her ticket.

Methods used to gather requirements

*Discovery Prototyping*

This is whereby prototypes of the booking system were made for the purposes of testing user friendliness. The system was eventually developed from multiple prototypes.

The aim of the prototypes was:

* To find out if the system is user friendly
* To find out if the system can be used concurrently by multiple users
* To find out if all the systems components are working as desired

Observation

This is whereby long distance travel agencies were observed while in operation. This was done to determine the following:

* How do customers book trips in a non computerized booking system
* How are trips scheduled
* Would a computerized system serve more customers per unit time as compared to the manual system.
* What details are required when booking a trip

**Design**

***Entities***

Customer

This is the client who uses the system to book a specific trip. The customer entity is not included in the database since tickets are what is used to keep records of customers

TravelTime

This entity denotes the various times when a user can travel. If a trip is scheduled to start at 11:am then that is the travel time.

The attributes are:

* Time

Place

This denotes the various geographical locations that the travel company serves. For example, if a trip is starting form Machakos and the destination is Nairobi, then Machakos and Nairobi are objects in the Place entity

The attributes are:

* name

Route

This entity is made up of two places, that is, a route from place A to place B, i.e route from Nairobi to Machakos

The attributes are:

* name – This is the name of the route e.g route 105
* dst1 – This is the first desitnation
* dst2 – This is the second destination

For one trip, dst1 can be the start of the trip and dst2 the destination whereas for another trip, dst1 can be the destination while dst2 the start of the trip

Vehicle

This denotes the vehicle that is used for travel.

The attributes are:

* plate – This is the number plate of the vehicle e.g KQT 136S
* route – This is a foreign key field to the route entity

Trip

A trip is an instance of travelling.

The attributes are:

* date – This is the date of the trip
* departure – This is the time set for the trip to start
* start – This is a foreign key field to the Place entity. It denotes the start location of the trip
* end – This is a foreign key field to the Place entity. It denotes the destination
* vehicle – This is the vehicle that is to be used for a trip. It is a foreign key field to the Vehicle entity.
* Available – This is a boolean field that is true if the trip is available for booking and false if the trip is unavailable. A trip is unavailable if all the seats are booked.

Seat

This is a representation of an actual seat that is to be reserved for a specific client

The attributes are:

* trip – This is the trip to which the seat belongs. It is a foreign key field to the Trip entity..
* number – This is the seat number
* occupied – This is a boolean field that is true if the seat is already booked and false if the seat has not yet been booked.

Ticket

The document that shows that a user has actually booked a seat

The attributes are:

* client\_name – This is the name of the customer
* telephone – This is the telephone number of the customer
* seat – This is the seat that the customer has booked
* trip – This is the trip that the customer is travelling.

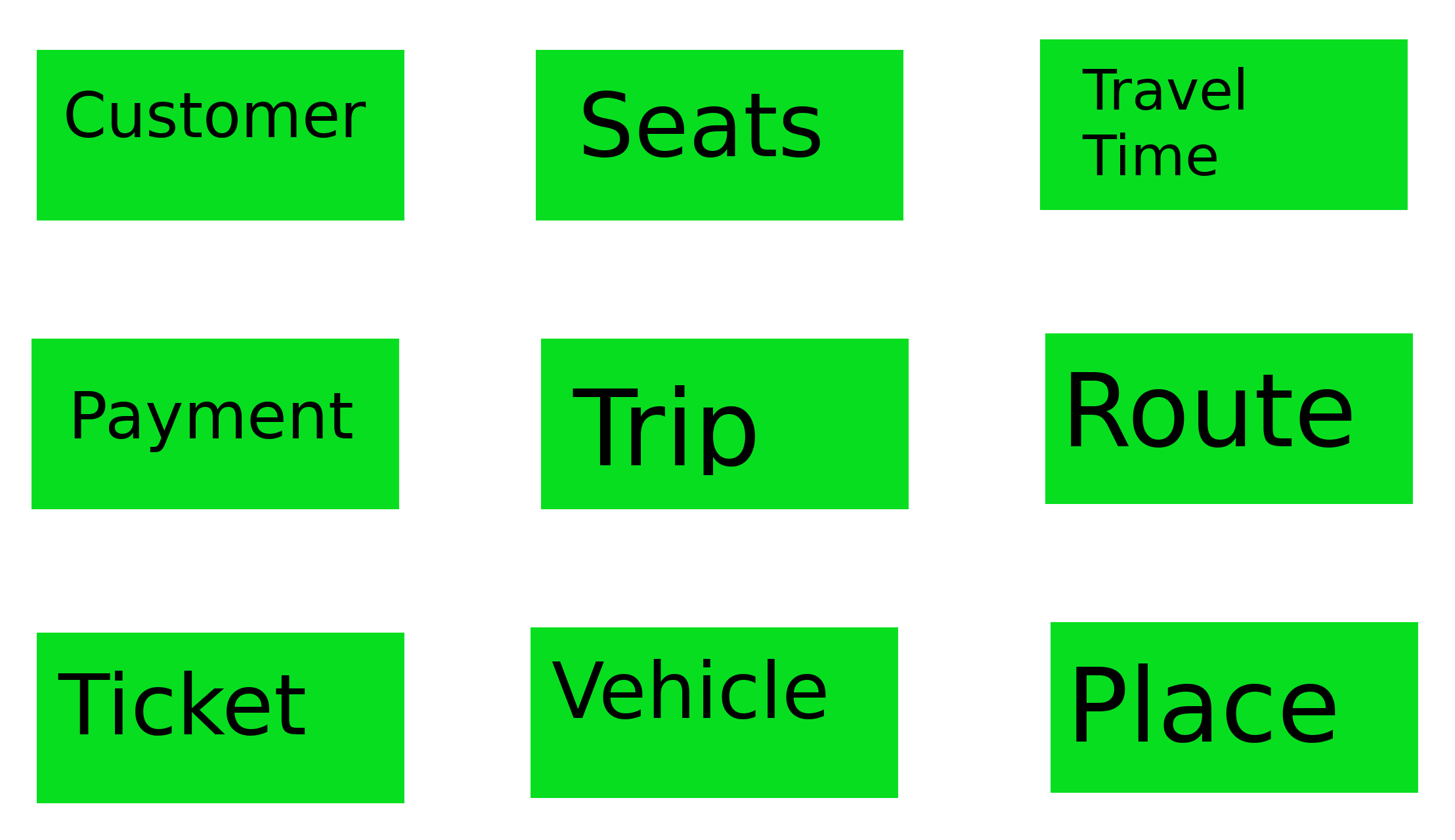
Payment

The payment made by user when he or she makes a reservation

The attributes are:

* telephone – This is the phone number of the customer that has made the payment
* code – This is the payment code generated after payment
* amount – This is the money value of the payment
* used – This is a boolean field that is true if the payment has already been used and false if the payment has yet to be used.

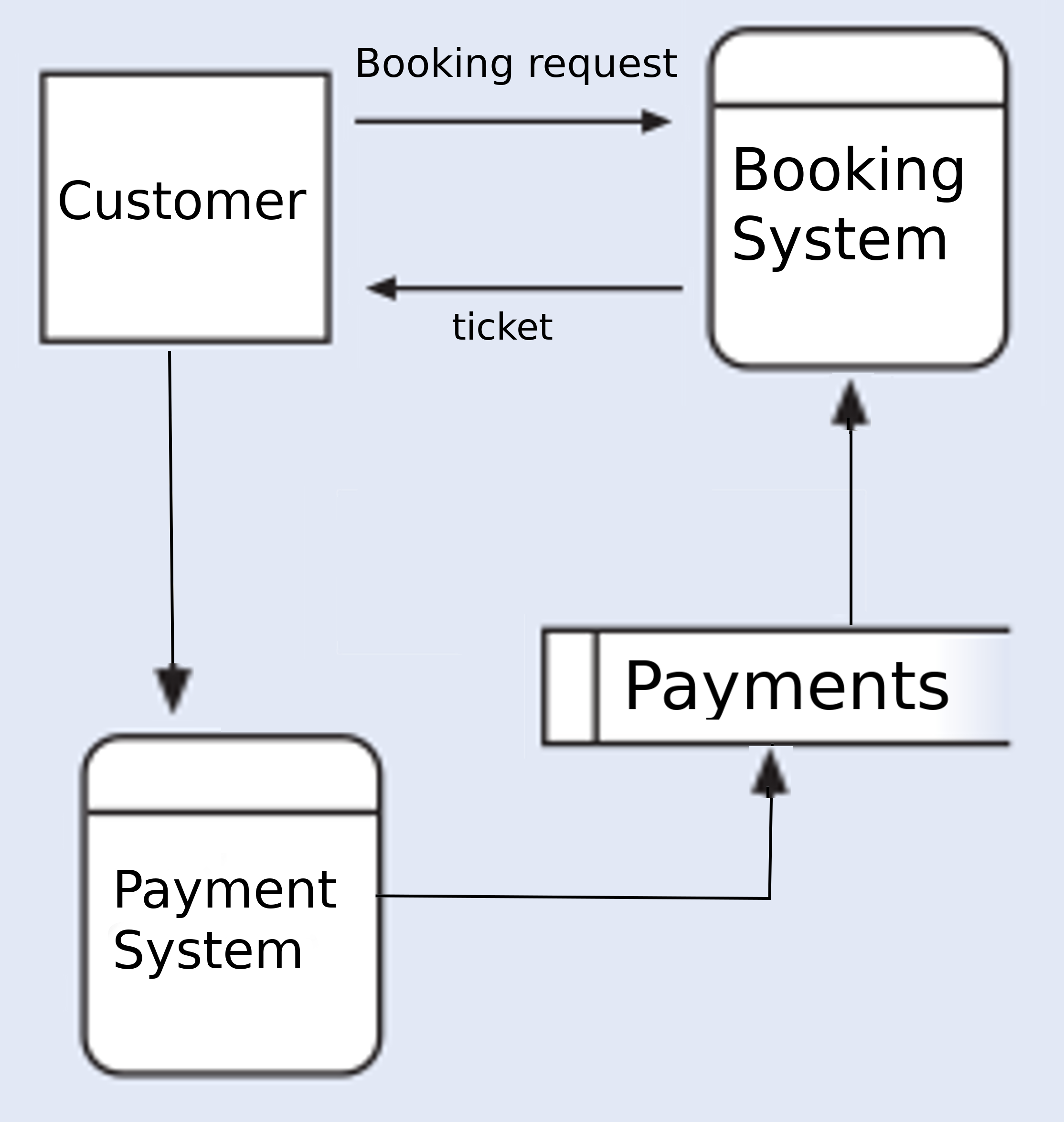
**Entities**

****

*Data Flow Diagrams*

***Context Diagram***

This is the most abstract view of the transit Reservation System.

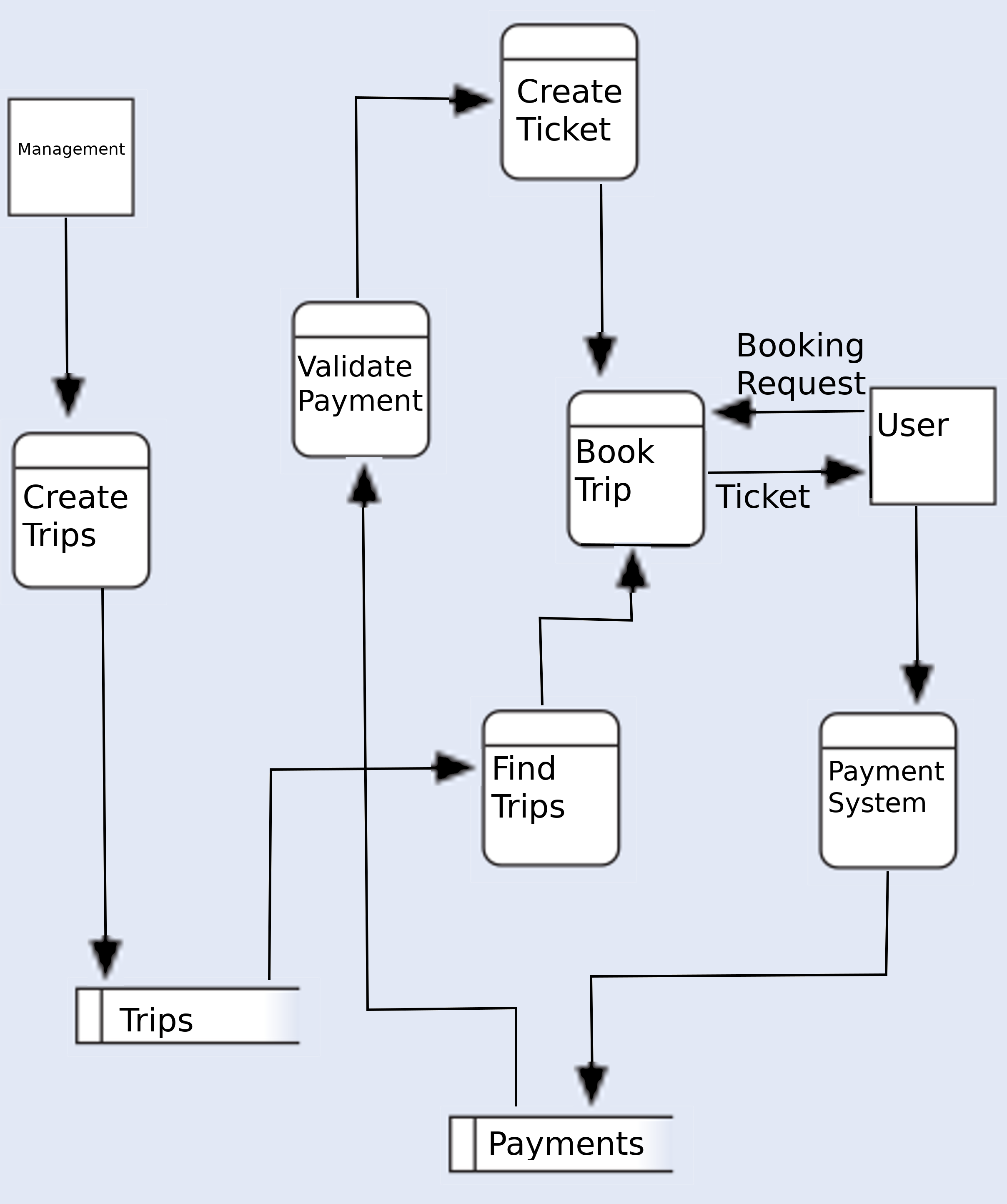


The Transit Reservation System has two basic subsystems, the booking system and the payment system.

Payments made by customers are handled by the Payment System while the booking system depends on the Payment system for payment validation.

The trigger for the booking system is the user making a booking request. Upon successful booking of a trip, the booking systems returns to the user a ticket as output.

***Data Flow Diagram of the Booking System***



**Explanation**

The management in the Transit Company should beforehand populate the database and create trips.

The trips are in the data store ‘Trips’

The customer should also beforehand make ticket payments using the payment system since the payment code given by the payment system is what is used to validate the payment on the booking system.

Once a customer triggers a booking event, the system, through user input, finds available trips stored in the ‘trips’ data store. Once a trip is found and the user triggers the booking event, the users payment code is first validated against the payment code provided by the payment system.

After payment validation, the system generates a ticket for the user for which the payment has been validated. After this, the user is presented with his/her ticket.

**Entity Relationship Diagrams**

Customer and Trip



A user can book zero or more trips

A trip can be booked by zero or more users

Trip and Seats



A trip can have zero or more seats

A seat belongs to one and only one trip

Trip and Travel Time



A trip has one and only one travel time i,e time scheduled to start trip.

A travel time can be for zero or more trips for example, two trips can be scheduled to start at the same travel time.

Trip and Place



A trip object has a destination and a starting point.

A trip can have one and only one place as a starting point.

A trip can have one and only one place as a destination.

A place can be a starting point for multiple trips

A place can be a destination for multiple trips

Trip and Vehicle



A trip has one and only one vehicle.

A vehicle can belong to zero or more trips provided the trips are not scheduled to start at the same time

Vehicle and Route



A vehicle has one and only one route for which it travels

A route can be assigned to zero or more vehicles

Customer and Payment



A customer can make zero or more payments.

A payment must be made by one and only one customer

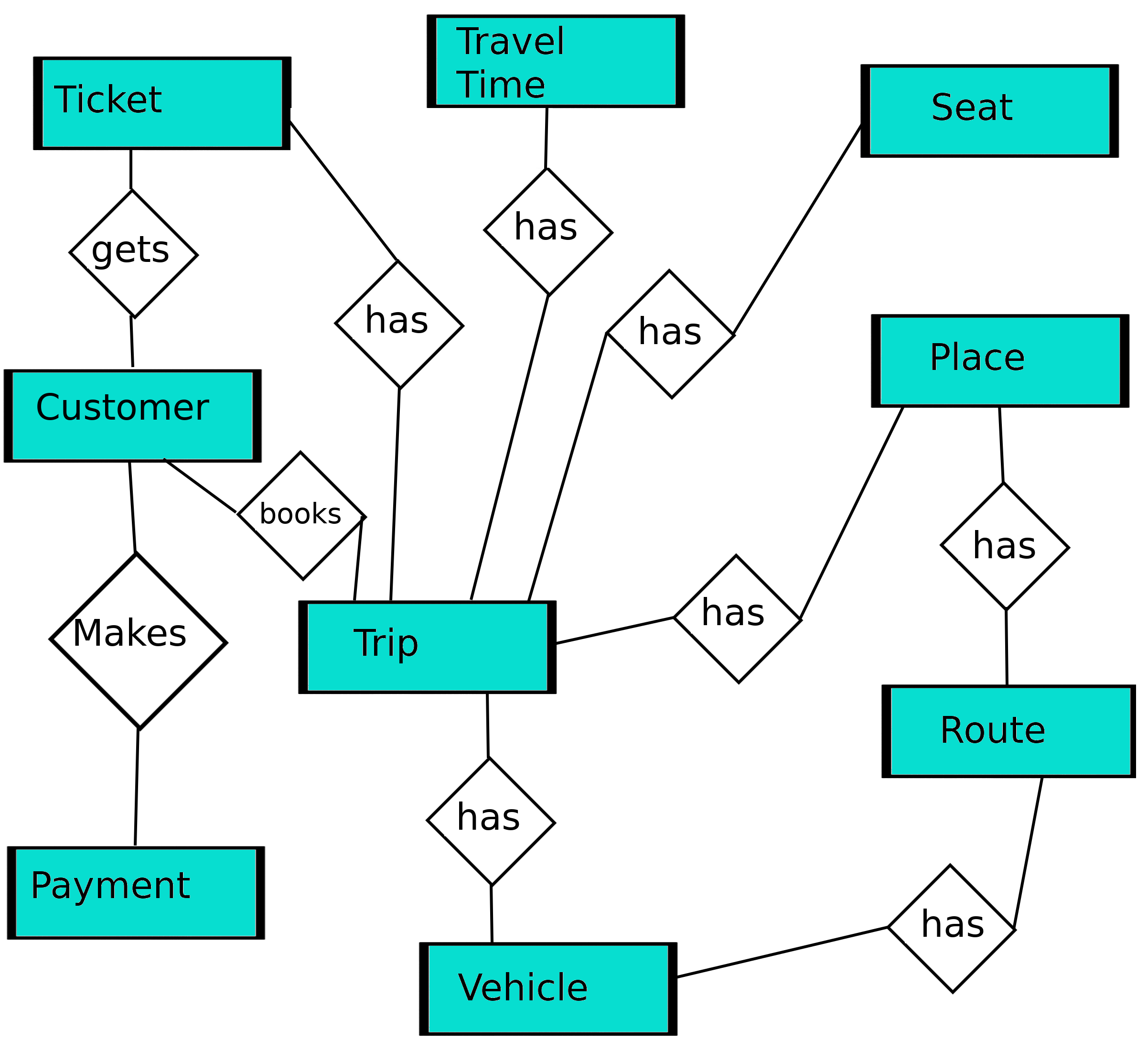
Ticket and Trip



A ticket belongs to one and only one trip

A trip can have zero or more t

Entity Relationship Diagram



Key

Customer makes payment

Customer gets ticket

Ticket has Trip

User books Trip

Trip has Travel Time

Trip has Place

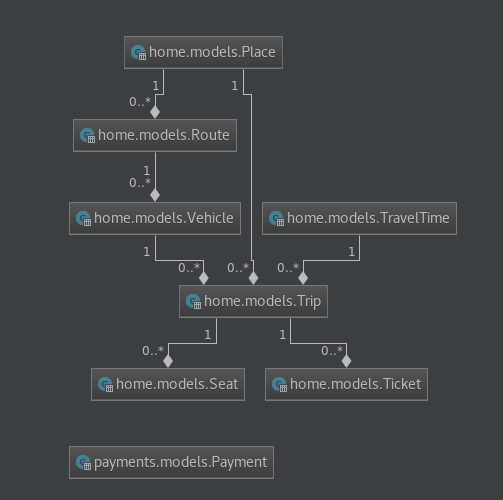
Trip has Seat

Trip has vehicle

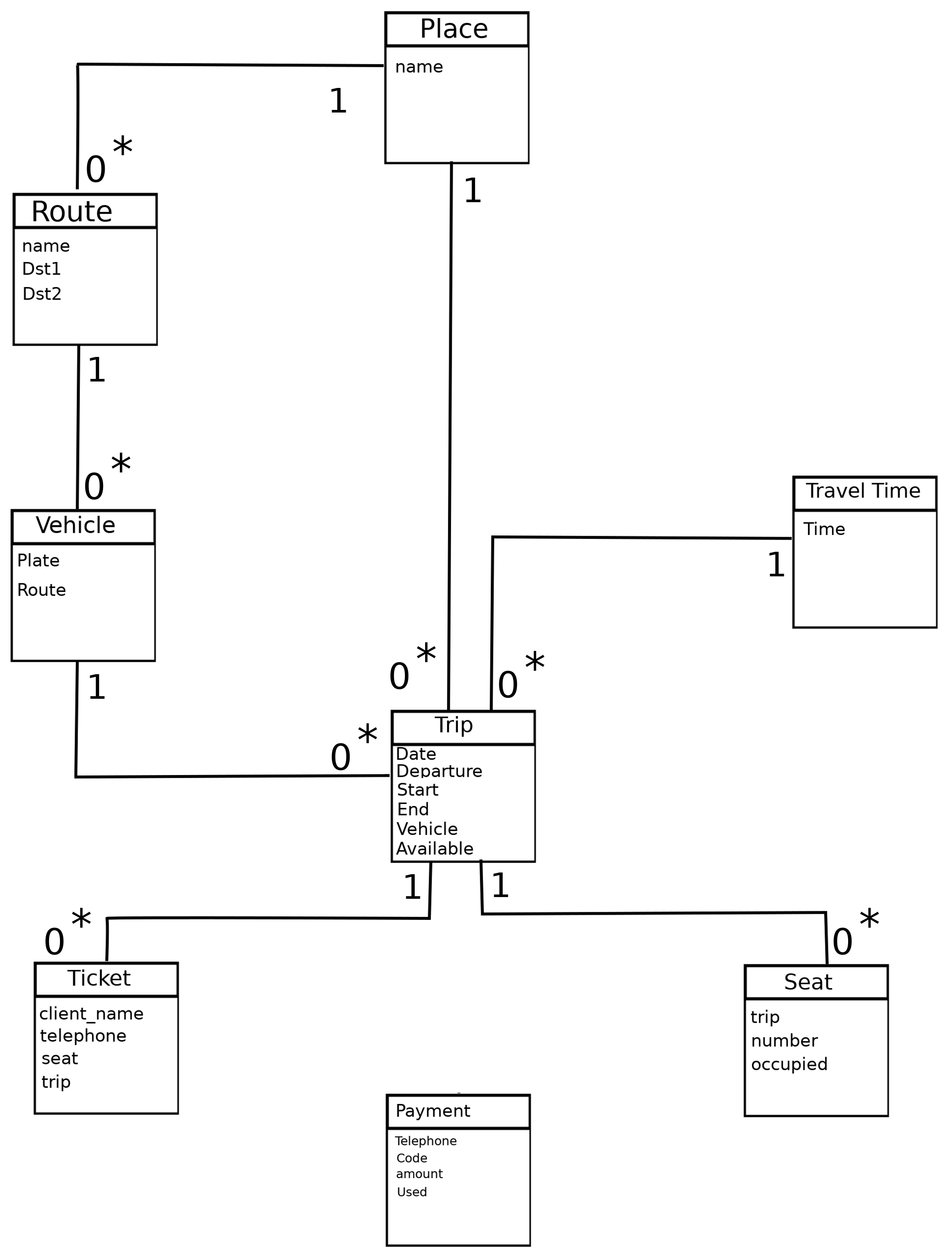
Vehicle has Route

Route has Place

**Entity Relationship Diagram 2**

****

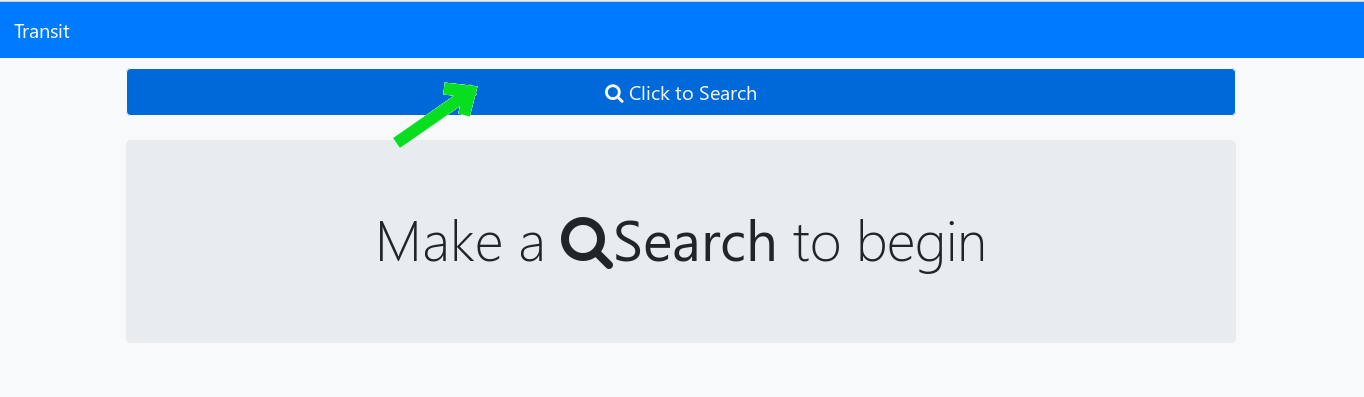
**Database Schema**

****

**Chapter IV**

**Inputs and Outputs**

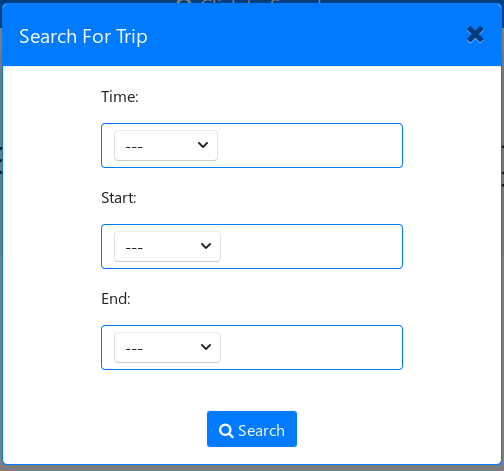
Once on the home page, the user is prompted to click the search button in order to display a modal of the ‘find trip’ form.



Find Trip

This input takes in the time, start location and end destination.

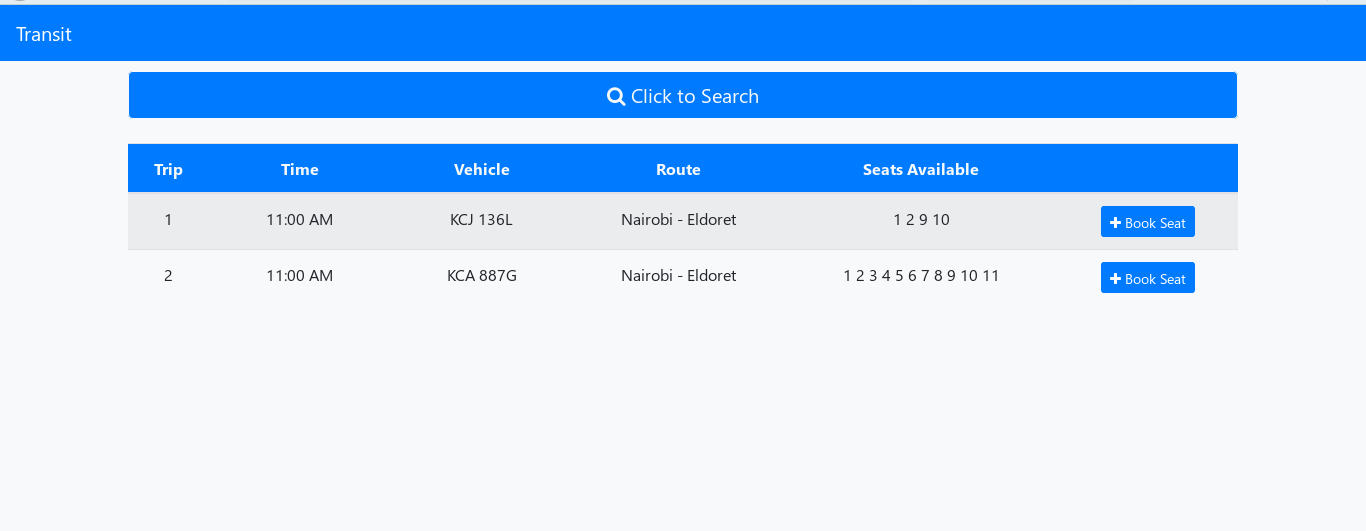
These variables are used to query the database to find the trips



After the database queries are made, the results are displayed to the user as below:

The output for trips shows the time the trip is scheduled to start, the vehicle that is for that trip, the route that the trip follows and finally the seats that are vacant for the trip.

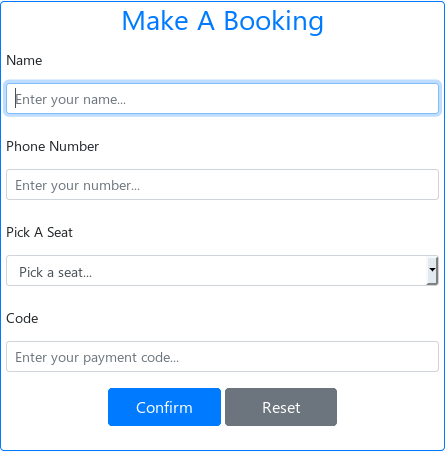
Once a seat is booked by a user, the seat is no longer shown in the ‘seats available’ section of the output below.



Make A Booking

This input takes in details of the user and the desired seat. It also takes in the payment code to be used for booking of the trip.

The payment is handled by a different system whereby when a user makes a payment, he or she is left with a code and the transit company is also left with the same code. The code input below is used to verify against the code that the company has. Once the payment is verified, it is marked as used and thus cannot be reused.

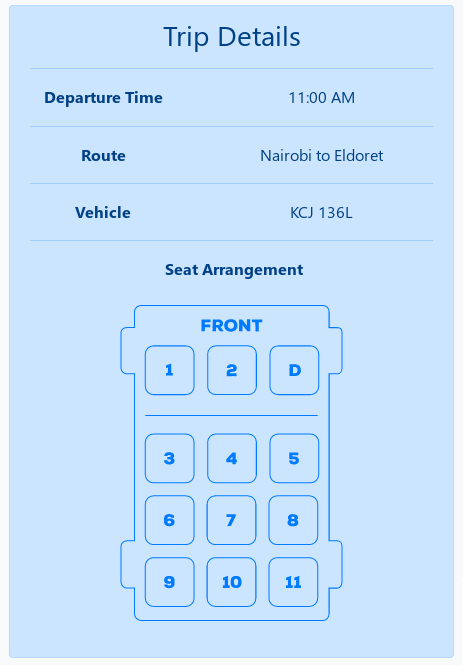


The ‘make a booking’ input form is presented to the user after he or she clicked the ‘book seat’ button in the trip results page.

The pick a seat section is a select field which only shows the seats that are vacant. Once a seat’s status is ‘occupied’ the seat is no longer displayed as an option.

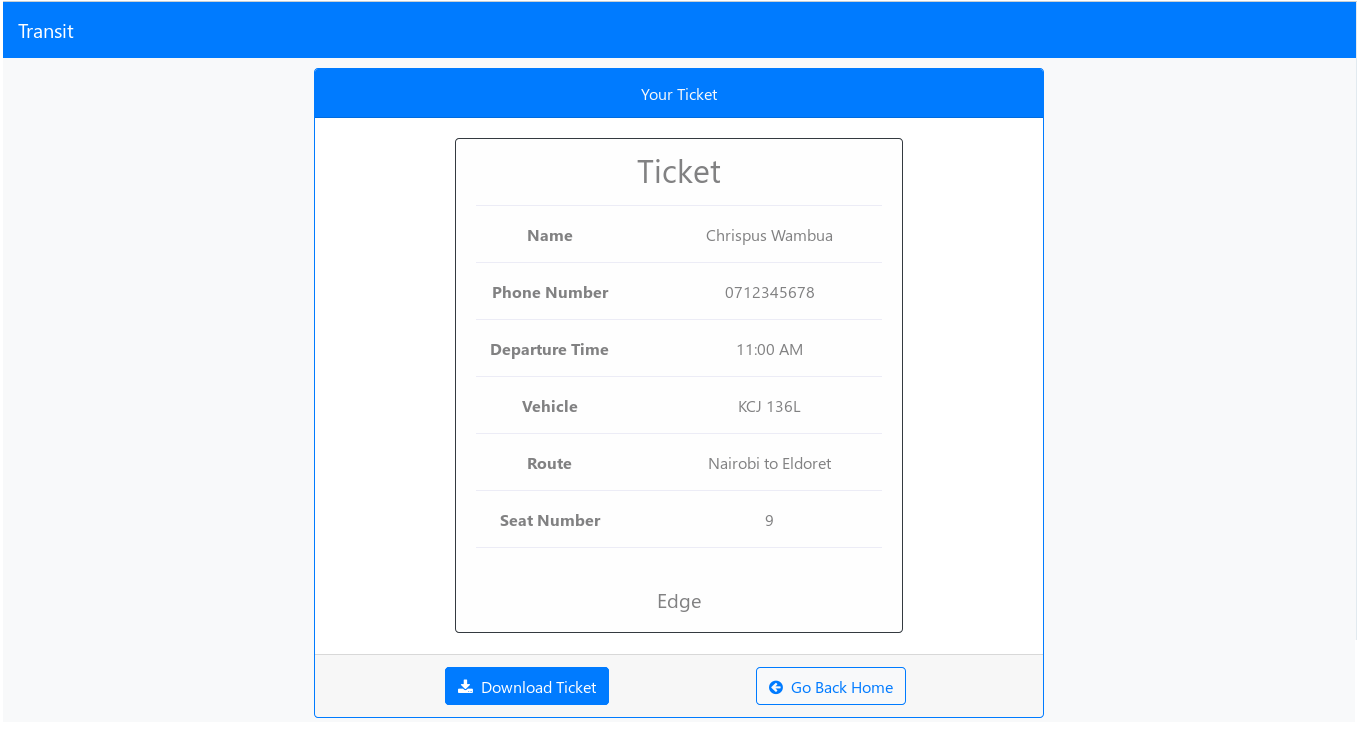
Trip Details

The above input form is accompanied by an output showing details of the selected trip. The ‘Trip Details’ output has a graphical drawing of a vehicle blueprint to help guide the user while selecting a seat.

****

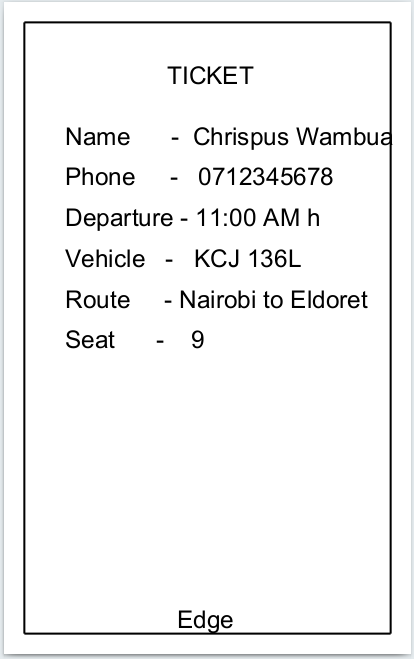
Ticket

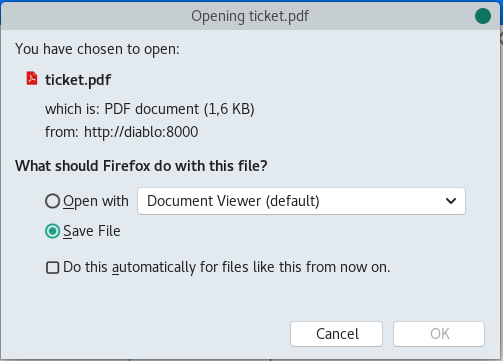
The ticket output is brought about after the user clicks the confirm button on the ‘make a booking’ input and everything is in order including the payment.



The ticket contains the details of the user, the departure time of the trip, the vehicle he/she is to travel in, the route of the trip and finally the seat that the user has booked. This ticket is what will be used to allow the user to board the vehicle.

The ticket output page also has a ‘Download Ticket’ button that allows the user to download the ticket. Once clicked, the save option is presented to the user.



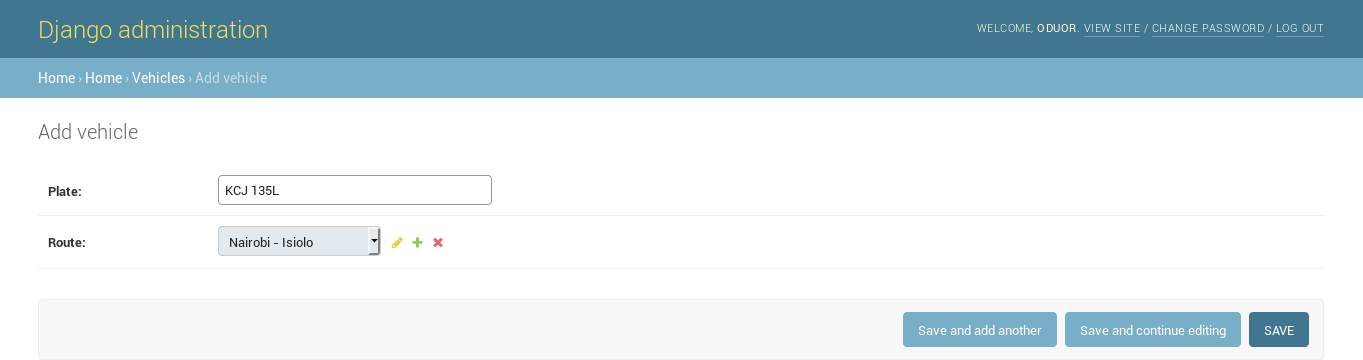


Once saved, the user gets the ticket in pdf form and he or she can print it.

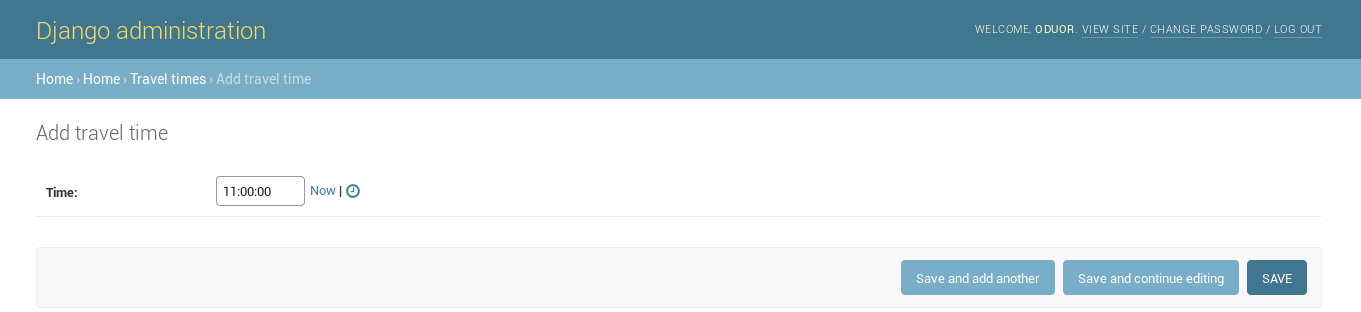
**Administration Inputs**

The django administration page provides a simple way to populate the database. The management will use the administration page to manage the website.

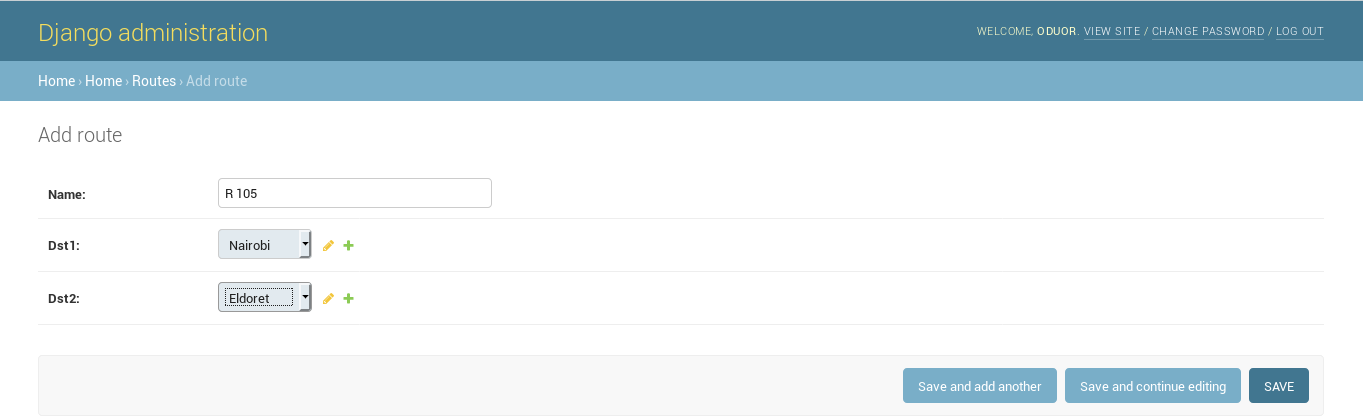
Add Vehicle



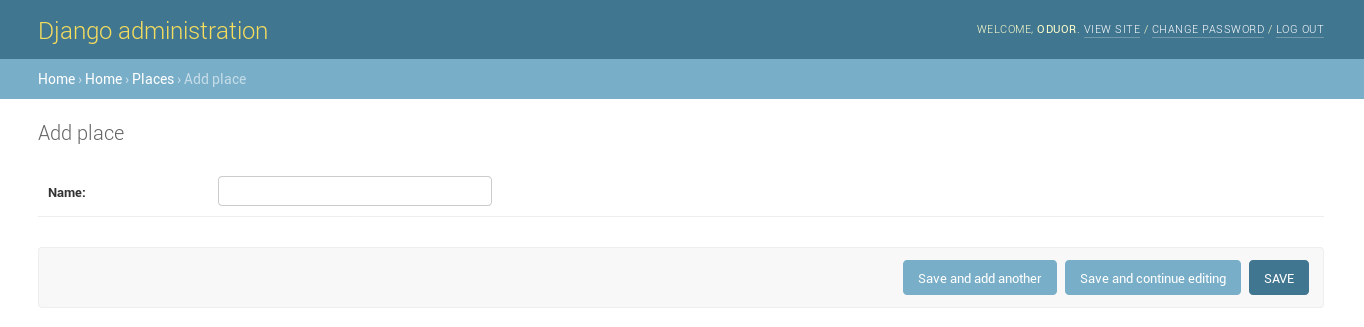
Add TravelTime



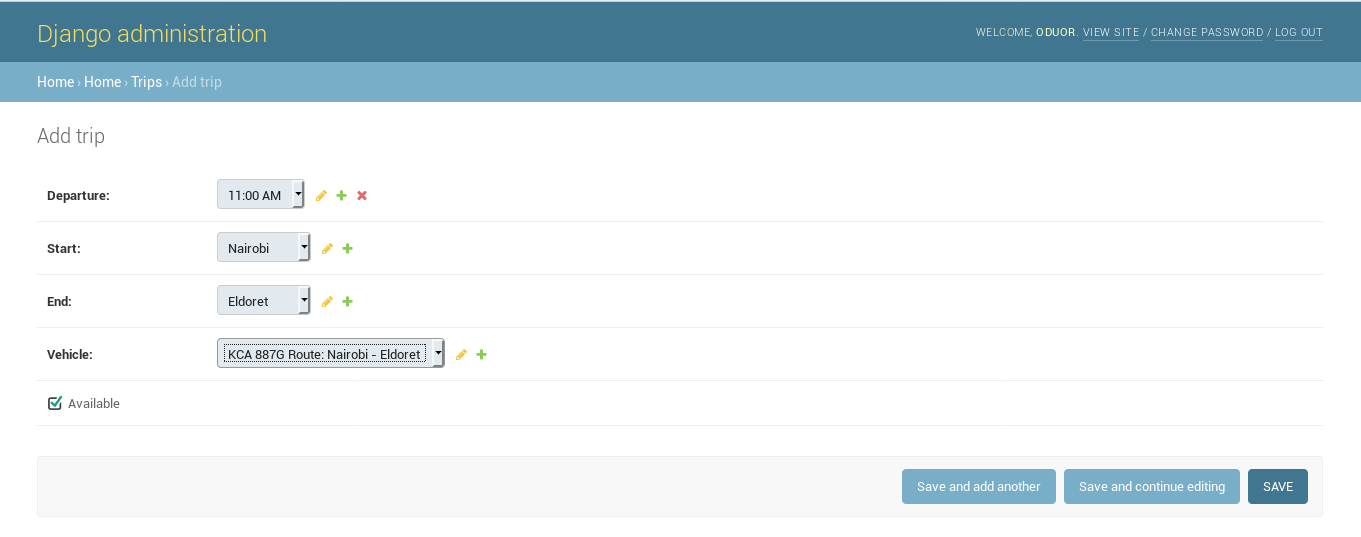
Add Route



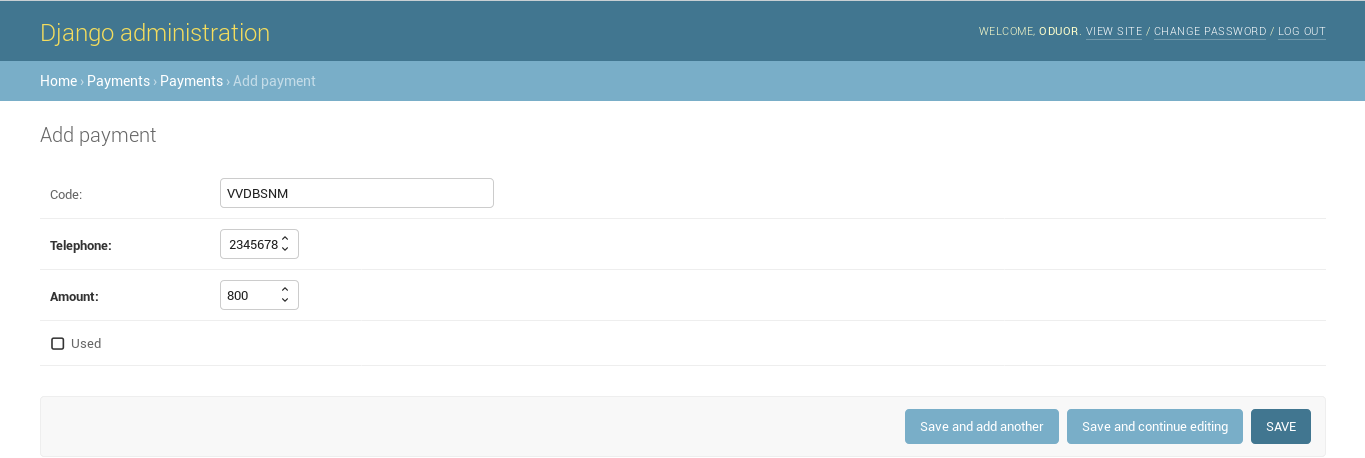
Add Place



Add Trip



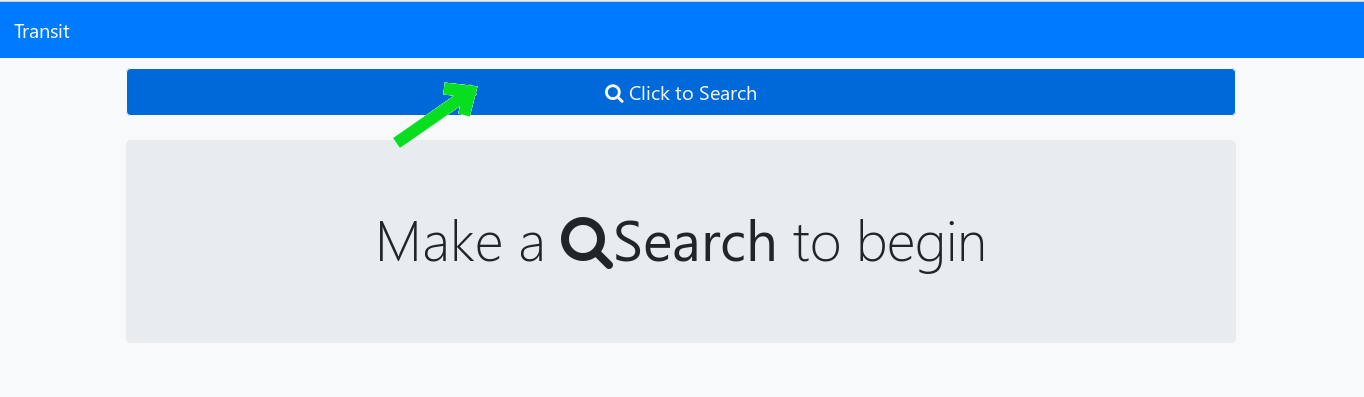
Add Payment



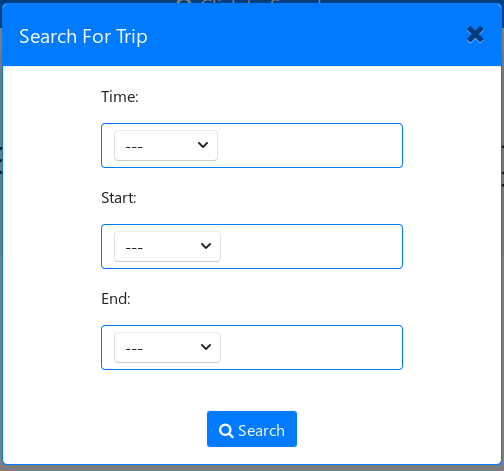
**Chapter V**

**User Manual**

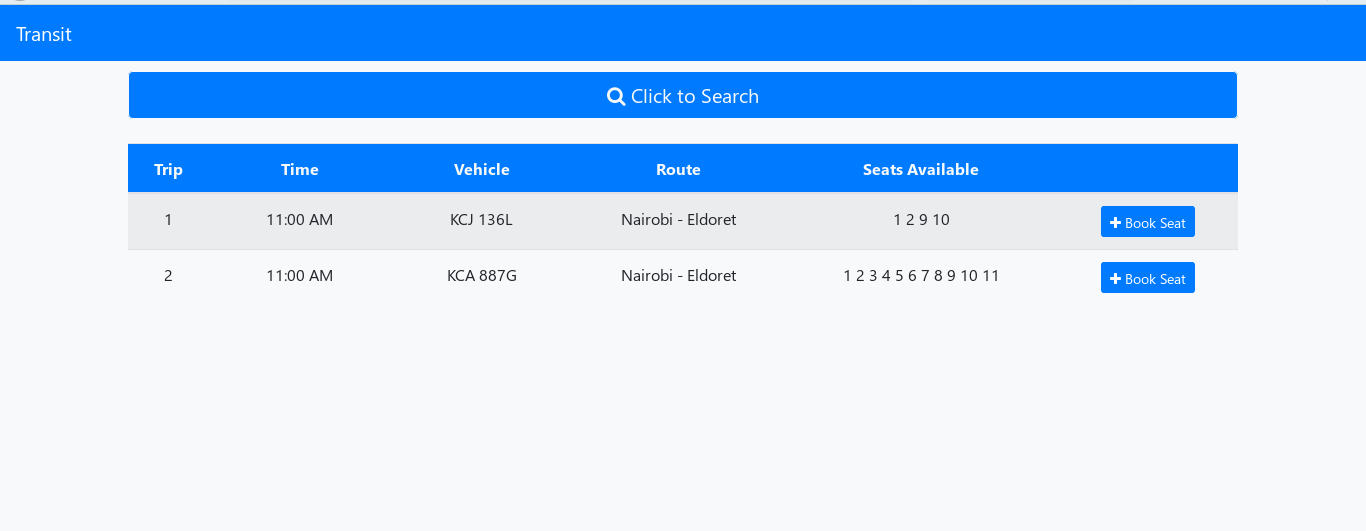
* Access transit website.
* Click to search



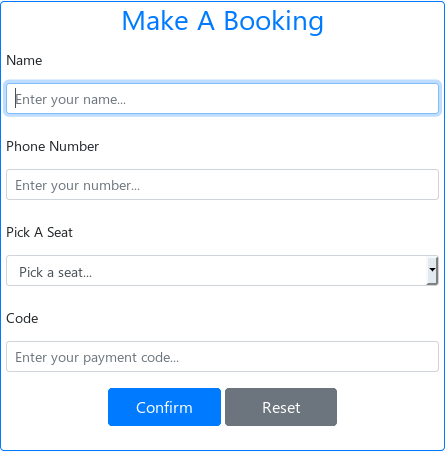
* Fill trip details



* Click Search button to find trips
* Click ‘Book Seat’ button for desired trip to book trip



* Fill personal details and payment code



* Click on confirm to book trip
* Download ticket

