TravelEase
Winter 2025

Jatin Aggarwal, 2023258
Nipun Aggarwal, 2023349
Pratyush Gangwar, 2023395
Ujjval Dargar, 2023564

February 16, 2025

Contents

Chapter 1

Deadline 1 - Project Overview

1.1 Objective

Our application is a travel management system that provides different travel packages, transportation options, and accommodation choices. Appropriately, different prices will be calculated for various combinations of these components. Our application will provide a smooth experience to travelers when booking packages for tourist locations. We will also provide interfaces to travel agencies and accommodation providers so that they can list their services efficiently.

1.2 Problem Statement

Travelers in today's hectic world face many challenges when planning their trip, which includes the time-consuming process of comparing prices and finding the best modes of transportation and accommodation through different providers.

With a wide range of transportation options alongside different accommodation choices, it's hard for users to find deals that match their preferences, especially their budgets.

Most existing travel platforms focus on a specific service, such as flight booking or hotel reservations. However, they fail to provide a one-stop solution that provides an efficient way to plan their journey. As a result, travelers are forced to navigate between multiple websites.

1.3 Stakeholders

The most important stakeholders for our application are:

1. **Customers:** Customers want a one-stop solution for their travel needs instead of visiting various sites. Our site will provide an extensive selection of travel and accommodation options.

2. **Tourism Agencies:** Tourism agencies will provide itineraries for tourism locations. These will be detailed day-wise plans for tourist locations and the best ways to explore them. However, they won't include travel or accommodation options.

3. **Transportation Providers:** Transportation providers like airlines and train agencies will list their travel options for different departure and destination locations and timings. Appropriately, the prices will vary. Users can select the ones that suit their preferences.

4. **Accommodation Providers:** Accommodation providers, such as hotels, resorts, etc., will provide travelers with places to stay. They will have different locations, prices, and distances to tourist locations. Users can select the ones that suit their preferences. They will have a separate interface on the site to list their services.

1.4 Functional Requirements

1. **User Management**: Users can sign up with details like name, contact number, email, and password. They can also update their details later as they feel fit.

2. **Packages**: The travel agencies will provide only an itinerary. This itinerary will contain details about the most popular tourist attractions at that location and the best way to visit them. The user cannot change these.

3. **Travel and Accommodation**: As mentioned above, tourism packages do not include travel or accommodations. So, users can choose their transport and accommodations according to their budget and preferences. For the same, we will provide an extensive database of flights, trains, and accommodation options.

4. **Bookings Management**: We will keep track of user bookings, including the package they choose, their travel choices, and their accommodation options.

5. **Payment Management**: The system will include a dummy payment feature allowing users to complete transactions online using various means of online payment, such as credit/debit cards.

6. **Tourism Agencies Management**: Tourism agencies can create travel itineraries for different destinations with different travel days and user preferences.

7. **Search**: Provide users with search functionality, enabling them to search different travel packages and itineraries according to their preferences for duration, expenses, travel locations, etc.

8. **Recommendation**: Our system would provide recommendations to users based on different metrics like most popular, most rated, most booked, etc.

9. **Administrator Management**: The administrator can handle user inquiries and remove packages and hotels based on user feedback and responses. Administrators can manage the website's data, cancellation and refund requests, and keep track of payments.

10. **Reviews / Feedback and Report**: Users can rate and review their travel experience, which will help other users book the best hotels/transport /packages. Users can also report issues like fraud or poor service, based on which administrators can take appropriate action.

11. **Package Comparisons**: Users can compare travel packages for a destination from different agencies based on their reviews, prices, and activities, which helps users book the best package.

12. **Filters**: Users can fine-tune their search results based on criteria such as location, price, travel dates and timing, transport type, accommodation type, ratings, and reviews.

13. **Dynamic Pricing**: The system adjusts booking prices based on demand and availability. For example, popular/high-demand tour package prices increase due to limited availability and season.

14. **Coupons**: Users can apply promotional codes to take advantage of offers and discounts when booking.

15. **Trip Cancellation and Refund Policies**: Users can cancel bookings based on specific guidelines. Refund eligibility depends on factors such as cancellation time and fare type (refundable/non-refundable) and might include cancellation fees.

1.5 Assumptions

1. The Customer will "Make" a Booking. This Booking has three components - accommodation (Hotel entity), transport (Train or Airplane route), and travel package (Itinerary). There are corresponding attributes - Room ID, Transport ID, and Itinerary ID. These can be NULL (with at least one being not NULL) if the user doesn't wish to use that particular service.

2. The Admin entity is not related to other entities. Instead, it is a management role that overlooks the maintenance and upkeep of the website. The Admin can insert/delete tuples from tables such as Coupons, Hotels, Flights, etc.

3. Booking is a weak entity because it only exists when a customer creates it. It is not a stand-alone entity. Similarly, Payment is a weak entity because it only exists when a booking is made.

4. Reviews will be associated with a Booking to display which customer made the review for which booking. Reviews will also contain an Item Type attribute to specify which of the three components (Accommodation, Transport, Itinerary) they're unsatisfied with.

5. The Description of the itinerary provided by the Tourism Agency will have a specific day-wise format. But this is not enforced by the database. It will be done in the front end during data entry.

Chapter 2

Deadline 1 - Entities and Relationships

2.1 Entities

1. Customer
   (a) Customer ID: Primary key
   (b) Name: Composite
      i. First Name: Must not be NULL.
      ii. Last Name: Can be NULL.
   (c) Contact Details: Composite. One of these must be not NULL. When the user signs up, they must provide at least one of these (or both). They may change these later.
      i. Email Address
      ii. Mobile Number
   (d) Password
2. Admin
   (a) Admin ID: Primary key
   (b) Name: Composite
      i. First Name: Must not be NULL.
      ii. Last Name: Can be NULL.

(c) Contact Details: Composite. One of these must be not NULL. The admin must provide at least one of these (or both). They may change these later.

i. Email Address

ii. Mobile Number

(d) Password

3. Tourism Agency

(a) Agency ID: Primary key

(b) Name: Corporate Name

(c) Location: Headquarters Address

(d) Contact Details: Composite. One of these must be not NULL. The Tourism Agency must provide at least one of these (or both) when registering. They may change these later.

i. Email Address

ii. Mobile Number

(e) Password

4. Itinerary. It just has the day-wise agenda and other static information. It does not represent a specific booking for a travel package, so it does not store information about when a particular travel package is scheduled. The scheduling information is in the I_Book_Includes Relationship.

(a) Itinerary ID: Primary key

(b) Description: Day-wise description in a specific format enforced by the application but not the database.

(c) Duration: Composite

i. Days

ii. Nights

(d) Price: In rupees

(e) Destination: Composite

i. City

    ii. State
    iii. Country

5. Transport Provider

    (a) Provider ID: Primary key
    (b) Name: Corporate Name
    (c) Contact Details: Composite. One of these must be not NULL. When registering, the Transport Provider must provide at least one of these (or both). They may change these later.
        i. Email Address
        ii. Mobile Number
    (d) Service Type: Train or Airplane
    (e) Password

6. Train

    (a) Train ID: Primary key
    (b) Name: For example, Shatabdi, Vande Bharat, etc.
    (c) Capacity

7. Train Route: This only represents a geographical route characterized by the arrival and destination stations. It does not include time or price data. These are present in the T_Route_Follows relationship.

    (a) Route ID: Primary Key
    (b) Arrival Location
    (c) Departure Location

8. Airplane

    (a) Airplane ID: Primary key
    (b) Name: For example, SpiceJet, Indigo, etc.
    (c) Capacity

9. Airplane Route: This only represents a geographical route characterized by the arrival and destination stations. It does not include time or price data. These are present in the A_Route_Follows relationship.
    (a) Route ID: Primary Key
    (b) Arrival Location
    (c) Departure Location
10. Accommodation Provider
    (a) Provider ID: Primary key
    (b) Name: Corporate Name
    (c) Contact Details: Composite. One of these must be not NULL. When the Accommodation Provider registers, they must provide at least one of these (or both). They may change these later.
        i. Email Address
        ii. Mobile Number
    (d) Password
11. Hotel. It just has the hotel description and other static information. It does not represent a specific hotel booking, and it does not store information about when a particular hotel stay is scheduled. The scheduling information is in the H_Book_Includes Relationship.
    (a) Hotel ID: Primary key
    (b) Name
    (c) Location
    (d) Price per Night
    (e) Contact Details: Composite. One of these must be not NULL. When the Hotel is added, they must provide at least one of these (or both). They may change these later.
        i. Email Address
        ii. Mobile Number
    (f) Total Rooms

(g) Available Rooms: Derived attribute. It will be calculated when the user makes a booking depending on the booking dates and reservations made by other users for those dates.

(h) Hotel Description

12. Review

    (a) Review ID: Primary key

    (b) Comment

    (c) Rating

    (d) Booking ID: Foreign Key (references Booking). Used to display information about the Customer and Booking.

    (e) Item Type: Hotel or Itinerary

13. Coupon

    (a) Code_Coupon: Primary key

    (b) Discount Percentage

    (c) Expiry Date

14. Booking (Weak entity)

    (a) Booking ID: Primary key

    (b) Transport Type: Airplane or Train (if any). It may be NULL if the user does not use our transport options.

    (c) Status: Confirmed, Pending or Canceled

    (d) Booking Date: Date when the Booking was Confirmed. Not the date when the travel begins.

15. Payment (Weak entity)

    (a) Payment ID: Primary key

    (b) Amount: In rupees

    (c) Payment Method: Net banking, UPI, Credit Card, Debit Card, etc.

    (d) Payment Status: Confirmed, Pending, or Canceled

2.2 Relationships

1. Makes
   (a) Customer makes Booking
2. Has
   (a) Booking has Payment
   (b) Itinerary has Review
   (c) Hotel has Review
3. Avails
   (a) Coupon is availed on Booking
4. Provides
   (a) Transport Provider provides Train
   (b) Transport Provider provides Airplane
   (c) Accommodation Provider provides Hotel
   (d) Tourism Agency provides Itinerary
5. T_Route_Follows: A relationship between a specific Train and a geographical Train Route. It includes price and time data. But it does not represent a train booking made by a Customer. That is handled by the T_Book_Includes relationship.
   (a) TRF_PKey: Primary Key. Although the primary key of a many-to-many relationship is the union of primary keys of participating entities (Train, Train Route) and attributes of the relationship (if necessary), we have introduced a surrogate key (TRF_PKey) so that the T_Route_Follows table can be easily referenced by other tables.
   (b) Price
   (c) Available Seats
   (d) Travel Time: Derived attribute
   (e) Arrival Details

i. Time
ii. Date

(f) Departure Details

i. Time
ii. Date

6. A_Route_Follows: A relationship between a specific Airplane and a geographical Airplane Route. It includes price and time data. But it does not represent an airplane booking made by a Customer. That is handled by the A_Book_Includes relationship.

(a) ARF_PKey: Primary Key. Although the primary key of a many-to-many relationship is the union of primary keys of participating entities (Airplane, Airplane Route) and attributes of the relationship (if necessary), we have introduced a surrogate key (ARF_PKey) so that the A_Route_Follows table can be easily referenced by other tables.

(b) Price

(c) Available Seats

(d) Travel Time: Derived attribute

(e) Arrival Details

i. Time
ii. Date

(f) Departure Details

i. Time
ii. Date

7. Includes

(a) I_Book_Includes: Booking includes Itinerary. It represents a specific booking for a travel package with a specific start and end time.

i. Itinerary Start Date
ii. Itinerary End Date: Derived attribute calculated from Itinerary Start Date and Duration in Days/Nights of Itinerary

(b) H_Book_Includes: Booking includes Hotel. It represents a specific booking for a hotel with a particular start and end time.

   i. Check-In Date
   ii. Check-Out Date
   iii. Rooms Booked

(c) T_Book_Includes: Booking includes T_Route_Follows (which represents a Train traveling on a specific Train Route along with price and date data). It represents a particular booking for a train and train route.

   i. Tickets Booked

(d) A_Book_Includes: Booking includes A_Route_Follows (which represents an Airplane traveling on a specific Airplane Route along with price and date data). It represents a particular booking for an airplane and airplane route.

   i. Tickets Booked

E-R Model

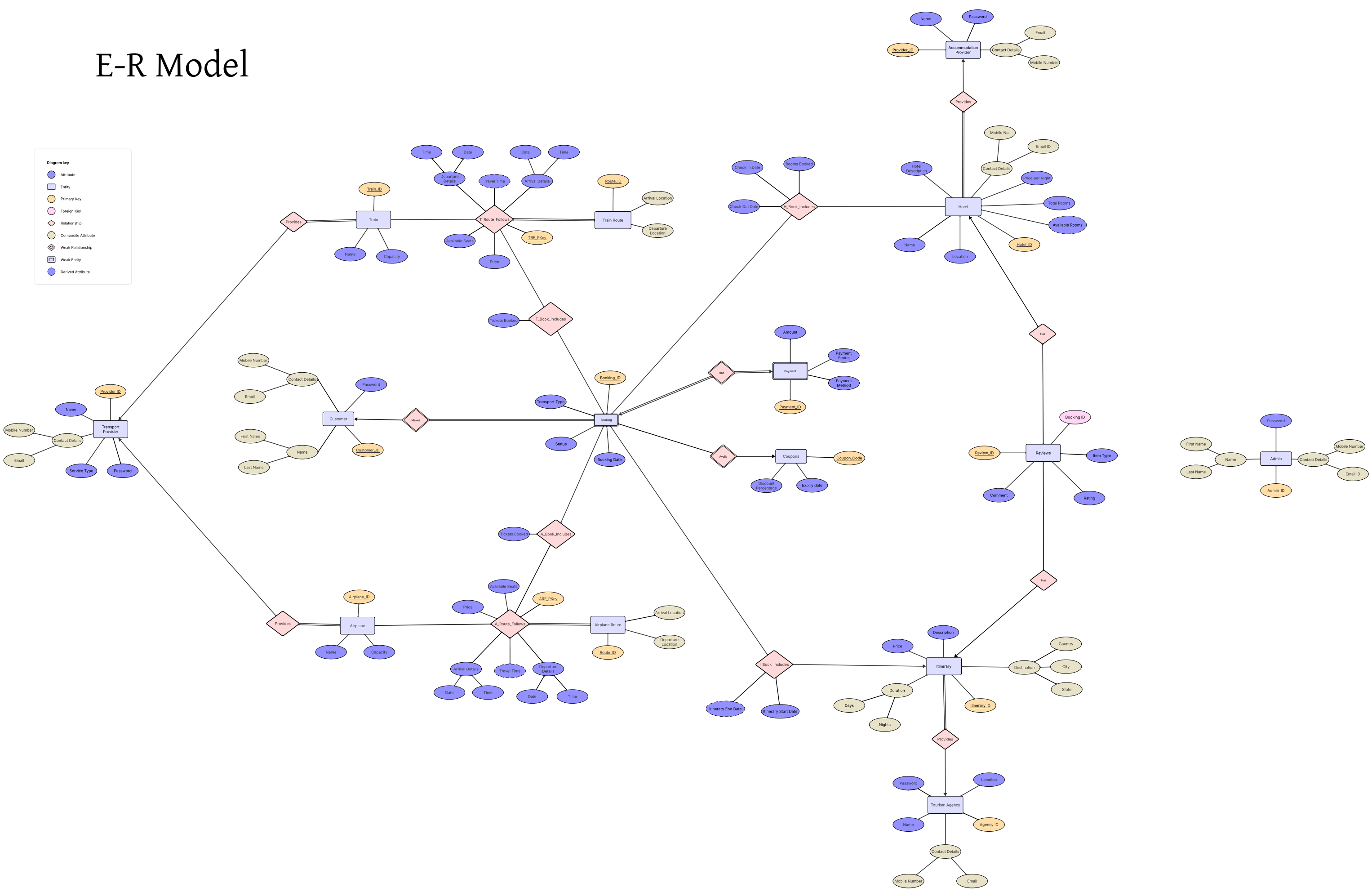


Diagram key
Attribute
Entity
Primary Key
Foreign Key
Relationship
Composite Attribute
Weak Relationship
Weak Entity
Derived Attribute
Transport Provider
Provider ID
Name
Mobile Number
Contact Details
Email
Service Type
Password
Provides
Train
Train_ID
Name
Capacity
T_Route_Follows
Time
Date
Departure Details
Travel Time
Date
Time
Arrival Details
Available Seats
Price
TRF_PKey
Train Route
Route_ID
Arrival Location
Departure Location
T_Book_Includes
Tickets Booked
Customer
Mobile Number
Contact Details
Email
Password
First Name
Name
Last Name
Customer_ID
Makes
Booking
Booking_ID
Transport Type
Status
Booking Date
Has
Payment
Amount
Payment Status
Payment Method
Payment_ID
Avails
Coupons
Coupon_Code
Discount Percentage
Expiry date
A_Book_Includes
Tickets Booked
Provides
Airplane
Airplane_ID
Name
Capacity
A_Route_Follows
Available Seats
Price
ARF_PKey
Arrival Details
Travel Time
Departure Details
Date
Time
Date
Time
Airplane Route
Arrival Location
Departure Location
Route_ID
Accommodation Provider
Name
Password
Provider_ID
Contact Details
Email
Mobile Number
Provides
Hotel
Mobile No.
Email ID
Contact Details
Hotel Description
Price per Night
Total Rooms
Available Rooms
Hotel_ID
Name
Location
H_Book_Includes
Check In Date
Rooms Booked
Check Out Date
Has
Reviews
Booking ID
Review_ID
Item Type
Comment
Rating
Has
Itinerary
Description
Price
Destination
Country
City
State
Duration
Days
Nights
Itinerary ID
I_Book_Includes
Itinerary End Date
Itinerary Start Date
Provides
Tourism Agency
Password
Location
Name
Agency ID
Contact Details
Mobile Number
Email
Admin
Password
First Name
Name
Last Name
Contact Details
Mobile Number
Email ID
Admin_ID

Chapter 3

Deadline 2 - Relational Model

1. Customer (<u>customer_id</u>, first_name, last_name, email, mobile_number, password)
   (a) Primary Key: <u>customer_id</u>
   (b) Constraints: email $\neq$ NULL OR mobile_number $\neq$ NULL
2. Admin (<u>admin_id</u>, first_name, last_name, email, mobile_number, password)
   (a) Primary Key: <u>admin_id</u>
   (b) Constraints: email $\neq$ NULL OR mobile_number $\neq$ NULL
3. TourismAgency (<u>agency_id</u>, name, location, email, mobile_number, password)
   (a) Primary Key: <u>agency_id</u>
   (b) Constraints: email $\neq$ NULL OR mobile_number $\neq$ NULL
4. Itinerary (<u>itinerary_id</u>, agency_id, description, duration_day, duration_night, price, destination_city, destination_state, destination_country)
   (a) Primary Key: <u>itinerary_id</u>
   (b) Foreign Key: agency_id $\rightarrow$ TourismAgency(agency_id)
5. TransportProvider (<u>provider_id</u>, name, email, mobile_number, service_type, password)

   (a) Primary Key: provider_id
   (b) Constraints: email $\neq$ NULL OR mobile_number $\neq$ NULL

6. Payment (payment_id, amount, payment_method, payment_status)
   (a) Primary Key: payment_id

7. Coupon (coupon_code, discount_percentage, expiry_date)
   (a) Primary Key: coupon_code

8. Booking (booking_id, customer_id, payment_id, coupon_code, transport_type, status, booking_date)
   (a) Primary Key: booking_id
   (b) Foreign Keys:
      i. customer_id $\rightarrow$ Customer(customer_id)
      ii. payment_id $\rightarrow$ Payment(payment_id)
      iii. coupon_code $\rightarrow$ Coupon(coupon_code)

9. Airplane (airplane_id, name, capacity, provider_id)
   (a) Primary Key: airplane_id
   (b) Foreign Key: provider_id $\rightarrow$ TransportProvider(provider_id)

10. AirplaneRoute (route_id, arrival_location, departure_location)
    (a) Primary Key: route_id

11. Train (train_id, name, capacity, provider_id)
    (a) Primary Key: train_id
    (b) Foreign Key: provider_id $\rightarrow$ TransportProvider(provider_id)

12. TrainRoute (route_id, arrival_location, departure_location)
    (a) Primary Key: route_id

13. AccommodationProvider (provider_id, name, email, mobile_number, password)

(a) Primary Key: provider_id

(b) Constraints: email $\neq$ NULL OR mobile_number $\neq$ NULL

14. Hotel (hotel_id, name, location, price_per_night, email, mobile_number, total_rooms, hotel_description, provider_id)

    (a) Primary Key: hotel_id

    (b) Foreign Key: provider_id $\rightarrow$ AccommodationProvider(provider_id)

    (c) Constraints: email $\neq$ NULL OR mobile_number $\neq$ NULL

15. Reviews (review_id, comment, rating, booking_id, item_type, item_id)

    (a) Primary Key: review_id

    (b) Foreign Key: booking_id $\rightarrow$ Booking(booking_id)

    (c) Constraints: rating $\in$ [1,5]

16. A_Route_Follows (arf_pkey, route_id, price, available_seats, arrival_time, arrival_date, departure_time, departure_date, airplane_id)

    (a) Primary Key: arf_pkey

    (b) Foreign Key: airplane_id $\rightarrow$ Airplane(airplane_id)

17. T_Route_Follows (trf_pkey, route_id, price, available_seats, arrival_time, arrival_date, departure_time, departure_date, train_id)

    (a) Primary Key: trf_pkey

    (b) Foreign Key: train_id $\rightarrow$ Train(train_id)

18. A_Book_Includes (booking_id, arf_pkey, tickets_booked)

    (a) Primary Keys: booking_id, arf_pkey

    (b) Foreign Keys:

        i. booking_id $\rightarrow$ Booking(booking_id)

        ii. arf_pkey $\rightarrow$ A_Route_Follows(arf_pkey)

19. T_Book_Includes (booking_id, trf_pkey, tickets_booked)

    (a) Primary Keys: booking_id, trf_pkey

(b) Foreign Keys:

i. booking_id → Booking(booking_id)

ii. trf_pkey → T_Route_Follows(trf_pkey)

20. H_Book_Includes (booking_id, hotel_id, check_in_date, check_out_date, room_booked)

(a) Primary Keys: booking_id, hotel_id

(b) Foreign Keys:

i. booking_id → Booking(booking_id)

ii. hotel_id → Hotel(hotel_id)

21. I_Book_Includes (booking_id, itinerary_id, itinerary_start_date)

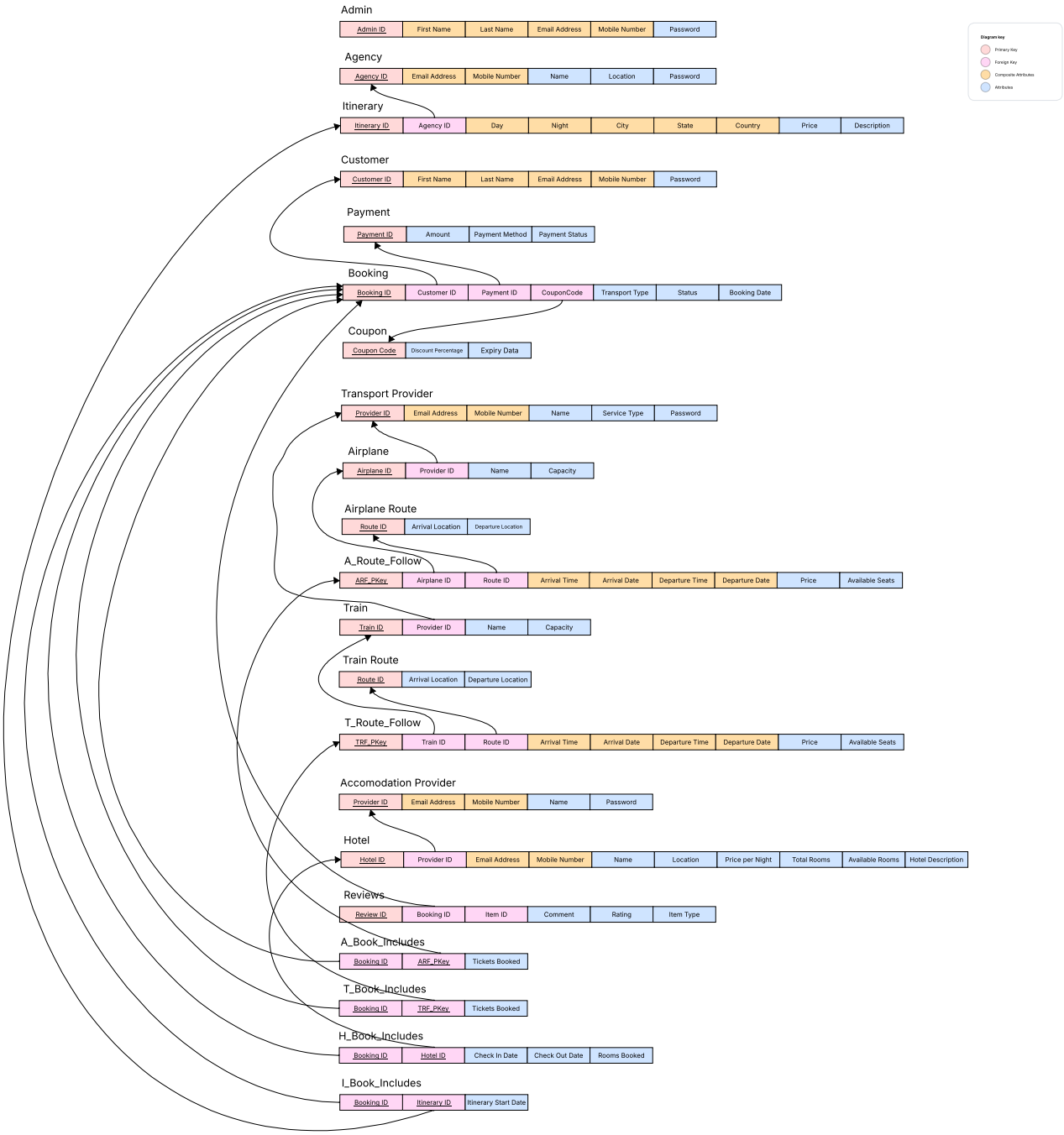
(a) Primary Keys: booking_id, itinerary_id

(b) Foreign Keys:

i. booking_id → Booking(booking_id)

ii. itinerary_id → Itinerary(itinerary_id)

Relational Schema



Chapter 4

Sources

1. ChatGPT - We took some help for LaTeX syntax. We also used it to clarify the relationships between some of our entities.
2. Publicly available past-year GitHub Projects