CS5200 – Final Project Report

Bus Ticket Booking System

- By Shivani Sharma and Ananya Asthana

Table of Contents

Intro

We have created a system that facilitates the booking of bus tickets, including the management of the bus ticket provider as well as the customer portal, using an interactive CLI based UI that takes in user inputs and provides options to perform tasks based on their roles. Our system facilitates CRUD operations for adding/editing/updating and viewing data in the tables.
The 2 roles available are Admin and Customer.

- Admin – Admin can perform CRUD operations on cities, bus types, drivers and existing trip routes. They can manage trip logistics.
- Customer – Customer can perform CRUD operations related to booking their journey and making payments.

Tables

1. bus_journey_booking – Contains details of the seats booked, source and destination route chosen, payment details and the username of the customer who made the booking.
2. bus_route – The available routes our platform currently provides buses on, as well as the timings of the trip, duration, and fare. Is linked with the bus type and driver for that trip.
3. bus_type – The types of buses available.
4. user – stores the unique username and role of every user.
5. city – The cities available for trips, that are used in our routes as the source and destination cities.
6. driver - stores the driver details, like name, phone number, license number etc.
7. admin – stores the admin details.
8. customer – stores the customer details.

README

We have provided a zip file of our project code which includes:

1. main.java – our main code
2. BusTypeDistributionChart.java – java code for visualization
3. BusBookingChart.java – java code for visualization
4. create.sql – our database create code
5. functions.sql – the functions used in our database
6. trigger.sql – the triggers used in our database
7. procedure.sql – the procedures used in our database
8. dump.sql - The SQL database dump that includes our schema, tuples, procedures, triggers and functions.

The process of running our project:

- Run our SQL dump (dump.sql) – this will create the database for you, insert the data, and run our procedures, triggers and functions to attach to it.
- Follow the steps given below-

Steps to run the project using IntelliJ IDE on MacOS

1. Go to File > Project Structure > Libraries
2. Click on the '+' symbol.
3. Select 'From Maven'.
4. Type **'mysql:mysql-connector-java:LATEST'** in the search bar and add to the repository where the Main.Java file is stored. Type '**org.jfree:jfreechart:LATEST'** in the search bar and add to the repository where the Main.Java file is stored.
5. Run the Main.java file and proceed following console prompts to provide inputs and perform functions.

Technical Specifications

- Java 11 programming language for backend code
- MySQL database and workbench for database management
- Java Database Connectivity (JDBC) API for connectivity to the MySQL database
- MySQL connector - mysql-connector-j-8.2.0.jar (included with the project zip file)
- JetBrains DataGrip for Database Vizualisation (as presented in class)
- JFreeChart for creating bar graph and pie char for data visualization.

User Flow

The functions a user can perform depends on their role.
There are 2 roles in our system – Admin and Customer, and their functions depend on their role.

Once the welcome message appears, the user is asked whether they are a new user or not. If they are new, they are asked whether they want to sign up as an Admin or a Customer.
Then they are asked to create a user profile.
The user profile includes the following:

1) A unique username
2) First name
3) Last name
4) Email ID
5) Phone number

These attributes are used to create a user profile. Once the user profile is created, the user is shown options based on the role they picked:

Admin can:

1) Add driver/city/route/bus details
2) Edit driver/city/route/bus details
3) View driver/city/route/booking/bus details
4) Delete driver/city/route/bus details

5) Visualize Bus Type Distribution
6) Visualize Bus Booking Chart
7) View user information
8) Edit user information
9) Quit

An admin has the capabilities of manipulating logistics information and ensuring that there are always correct routes (trips) available for users to book. They can also view and edit their own profile information.

Customer can:

1) Create new booking
2) Edit existing booking
3) View existing booking
4) Cancel existing booking
5) View user information
6) Edit user information
7) Quit

A customer can manipulate their trip bookings, by picking the pre-determined route (trip) they want to go on, the number of seats they want to book and payment method to want to use to pay the total fare that is shown to them. They can also view and edit their own profile information.

The CRUD operations are reflected in the database once the user inputs are accepted and the processing is correctly performed.

Bonus Work Done

1. Visualization of the data.
2. Application supports multiple user roles.
3. Complicated schema – user data pull requires multi-joins, or many tables (> 10) due to the complexity of the data domain.
4. Interesting queries that can be used for analysis or visualization of the data.

(report continued..)

Additional features implemented after the class presentation

1. **Data visualization option in the admin menu**
   Sample charts:

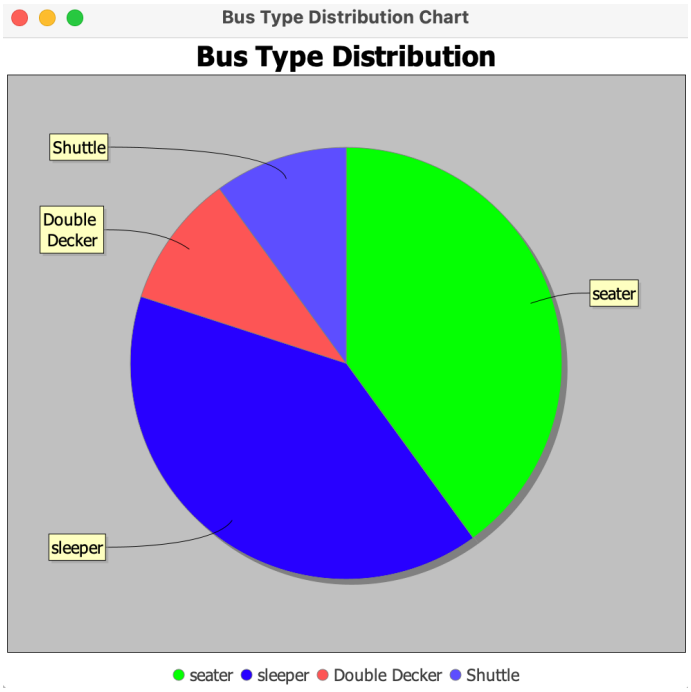


Fig 1: A pie chart showing the distribution and the count of the types of buses available in our application.

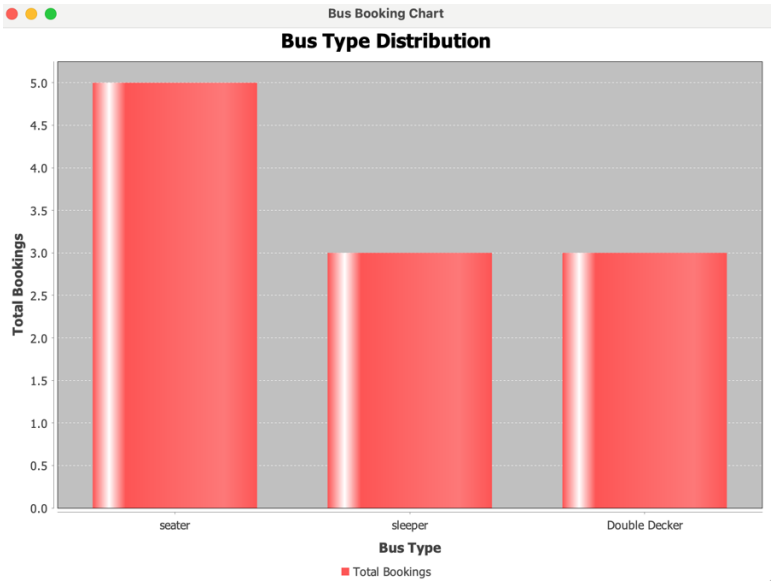
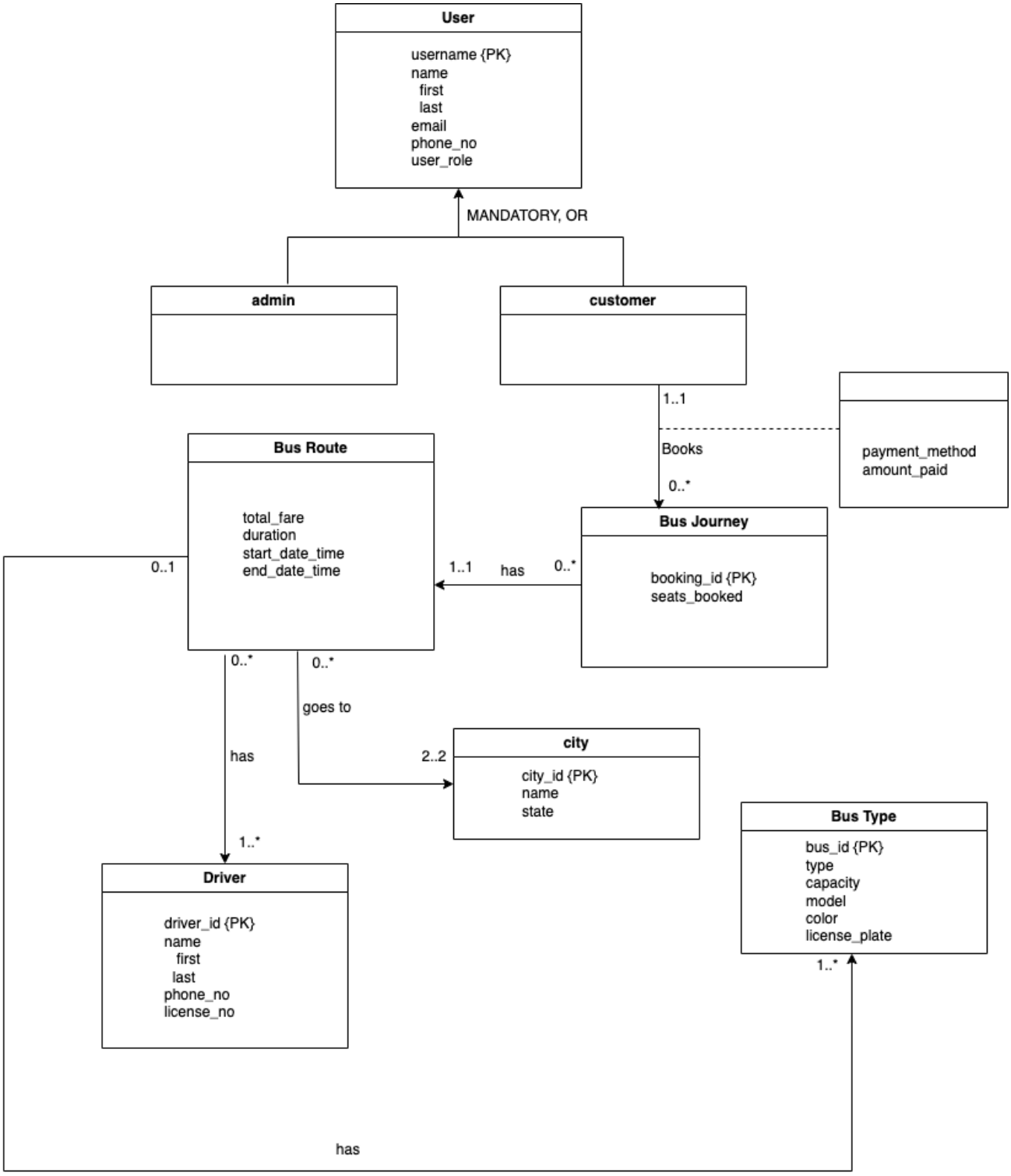
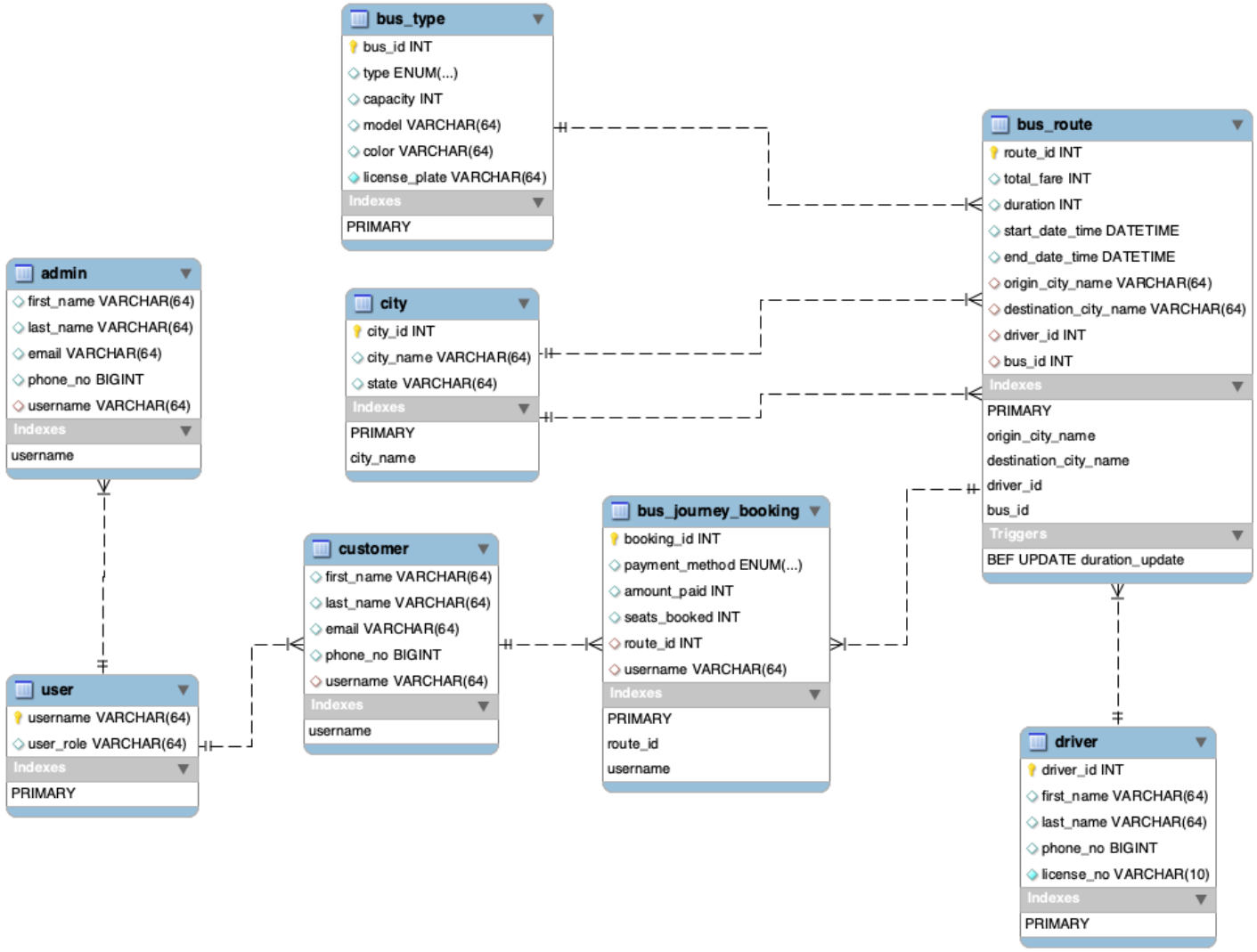


Fig 2: A bar graph showing the number of bookings for a particular bus type.

Conceptual Design

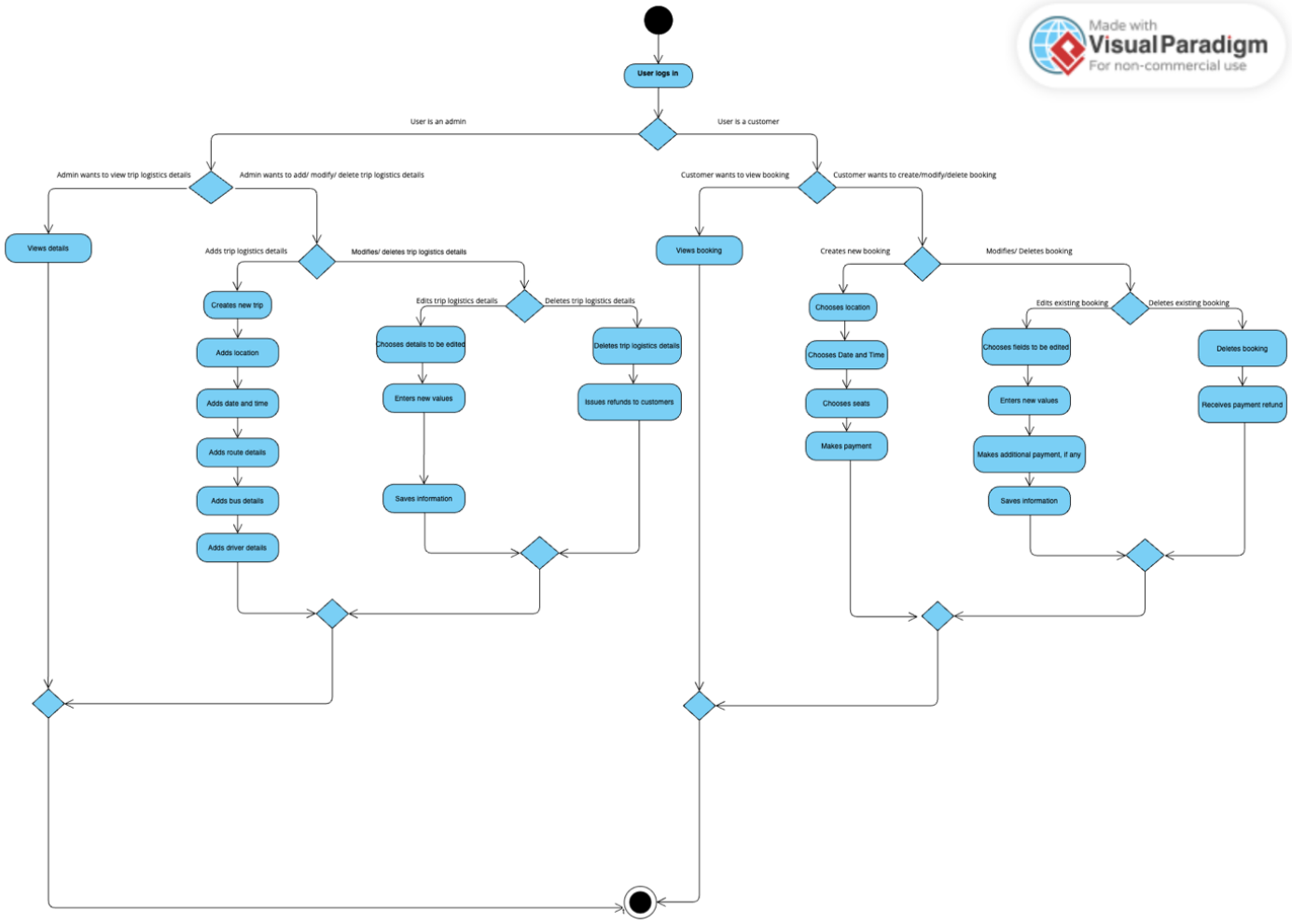


Logical Design



(report continued..)

Activity Diagram



Lessons Learned

1. **Technical expertise gained:**
   During the course of this project, we gained valuable information related to various faucets of database management. We were able to works upwards, right from the inception of our idea to the execution of a well-planned database management and query retrieval system. We implemented MySQL procedures, triggers, and functions to contribute to the smooth running of CRUD operations. We also learnt a lot about error handling by tackling all the various scenarios when a user may provide an incorrect input to the system. We used the "JFreeChart" library to enable data visualization capabilities in our project.

2. **Insights:**
   We gained various insights, including time management insights, data domain insights etc, which helped us evolve as both programmers as well as members of a team. We were able to work cohesively, accept contrary ideas and work towards a common goal. Since the scope of the domain we chose was vast, we had to delegate tasks between us to ensure the timely completion of all the elements. This also resulted in us exploring the possible designs in-depth and gaining more clarity about the functionalities involved in the bus ticket booking process. We also learned the value of a good initial plan, wherein we design our UML and activity diagrams in advance so our code can reflect what we wish for it to do.

3. **Realized or contemplated alternative design / approaches to the project:**
   We went through a lot of different possible implementations and multiple iterations while finalizing our design. We considered other entities and attributes but decided against them after careful contemplation. For example, we initially considered including a booking status functionality, but decided that once payment has been made, and we provide the option to edit the booking as well cancel it, the status does not add a lot to our existing data. Thus, we excluded it from our final design. We had also considered making it a Python based application but keeping the pros and cons as well as our proficiency in various programming languages, we settled on Java.

Future scope

1. **Planned uses of the database:**
   This database can be used for any transportation system management portal. It can be easily adapted to fit different scenarios and serve as an efficient organization method. We can also reuse it for an employee management system as we have segregated the activities that can be performed based on the roles.

2. **Potential areas for added functionality:**
   a. Adding an interactive GUI is a potential area of advancement. This will make it easier for our users to interact with our system and won't leave room for error with the text-based commands.
   b. Adding password-based authentication will increase the security of our system. It will help us verify the identity of our users.
   c. We can integrate a payment gateway and payment authentication mechanism to ensure that the payments are successfully received and/or refunded after creating or cancelling a booking.