Airline Ticket Booking and Flight Management System

A project report submitted in partial fulfillment of
The requirements of Business Process Management Course (IMMG-4108).

IPG-MBA

by

Anirudh Gautam (2019-IMG-006)
and
Kawaljeet Singh Batra (2019-IMG-027)

submitted to

Dr. Gyan Prakash

ABV-INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT
GWALIOR-474015

2022

Table of Contents

Chapter 1

INTRODUCTION

1.1 Overview

Online airline ticket booking system is one of the essential applications of E-commerce. With the development of Internet and security technology, more and more people begin to consume online, which is more convenient and personal than traditional way. The goal of this system is to make people purchase airline tickets easily. The system is written in Javascript and Python.

The Web-based airline ticket booking system, **EasyCruise**, uses client/server architecture. The customers can use Web browser to access the system and book airline tickets. Because Internet and web browsers are widely used all over the world, there is no need to train customers how to use them. The second advantage is that there is no limit on customers' operating systems, for example, they can use almost all kinds of popular operating systems such as MacOS, Linux and Window.

1.2 Technologies Used

1.2.1 Svelte

Svelte is a radical new approach to building user interfaces. Whereas traditional frameworks like React and Vue do the bulk of their work in the browser, Svelte shifts that work into a compile step that happens when you build your app. Instead of using techniques like virtual DOM diffing, Svelte writes code that surgically updates the DOM when the state of your app changes.

1.2.2 FastAPI

FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints. The key features are:

- Fast: Very high performance, on par with NodeJS and Go (thanks to Starlette and Pydantic). One of the fastest Python frameworks available.
- Fewer bugs: Reduce about 40% of human (developer) induced errors.
- Short: Minimize code duplication. Multiple features from each parameter declaration. Fewer bugs.
- Easy: Designed to be easy to use and learn. Less time reading docs.

1.2.3 PostgreSQL

PostgreSQL is a powerful, open source object-relational database system with over 35 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance. There are several benefits for choosing Postgres over traditional MySQL. PostgreSQL is an object-relational database, while MySQL is purely relational.

1.2.4 Tailwind CSS

Tailwind CSS can be used to make websites in the fastest and the easiest way. Tailwind CSS is basically a utility-first CSS framework for rapidly building custom user interfaces. It is a highly customizable, low-level CSS framework that gives you all of the building blocks you need to build bespoke designs without any annoying opinionated styles you have to fight to override. The beauty of this thing called tailwind is it doesn't impose design specification or how your site should look like, you simply bring tiny components together to construct a user interface that is unique. What Tailwind simply does is take a 'raw' CSS file, processes this CSS file over a configuration file, and produces an output.

Chapter 2

DATABASE DESIGN

In this section, we discuss about the database and its design for the airline ticket booking system. PostgreSQL, a Relation Database Management System has been used for this system. The benefit of using a relational database is the ability to interlink communication entities together through the means of joins and then being able to store and retrieve transactions in a fast and efficient manner.

2.1 PostgreSQL

PostgreSQL is a powerful, open source object-relational database system with over 35 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance. There are several benefits for choosing Postgres over traditional MySQL. PostgreSQL is an object-relational database, while MySQL is purely relational. This means PostgreSQL offers more complex data types and allows objects to inherit properties, but it also makes working with PostgreSQL more complex.

2.1.1 Comparision with MySQL

There are several benefits for choosing Postgres over traditional MySQL. PostgreSQL has a single, ACID-compliant storage engine. MySQL has support for 16 different storage engines suitable for different use cases. The default storage engine, InnoDB, provides index-organized

tables. PostgreSQL spawns a new system process with its own memory allocation for each client connection it establishes, so it requires a lot of memory on systems with a high number of client connections. MySQL uses a single process and maintains one thread (or path of execution) per connection, which works well for most applications of less than enterprise scope.

Three common database features are views, triggers, and stored procedures. PostgreSQL has more robust views, and supports materialized views, which can improve performance for complex queries. Both databases support AFTER and BEFORE triggers for SQL INSERT, UPDATE, and DELETE statements; PostgreSQL also offers an INSTEAD OF trigger, and can execute complex SQL statements in a trigger using functions. Both databases support standard SQL stored procedures, but PostgreSQL offers in addition the ability to call procedures written in languages other than SQL.

2.2 Schemas

There are three 4 major relations in the database, they are:

- **User**
- **Flight**
- **FlightSchedule**
- **Journey**

2.2.1 User Schema

The User relation, depicts the attributes that are related to the user, accessing the system through web. The schema of the User table includes the attributes like *username*: for storing the name of the user, *password* : for storing the account password for authentication, and *profilePhoto* : a link for the user's profile photo.

- id : UUID PRIMARY KEY, NOT NULL
- username : STRING
- password : STRING

2.2.2 Flight Schema

The Flight relation, depicts the attributes that are related to the aircraft, that is being drafted for use by the company. The table includes the following attributes:

- id : UUID PRIMARY KEY, NOT NULL
- flightCode : UUID
- company : STRING
- source : STRING
- destination : STRING
- ecoCapacity : INTEGER
- busCapacity : INTEGER
- execCapacity : INTEGER
- ecoPrice : FLOAT
- busPrice : FLOAT
- execPrice : FLOAT

2.2.3 FlightSchedule Schema

The Flight relation, depicts the attributes that are related to the flight schedule. Once an aircraft has been drafted, it can be scheduled for a particular date of flight. the attributes of this relation includes:

- id : UUID PRIMARY KEY, NOT NULL
- flightID : FOREIGN KEY NOT NULL
- source : STRING
- destination : STRING
- date : DATETIME
- ecoRem : INTEGER
- busRem : INTEGER
- execRem : INTEGER

2.2.4 Journey Schema

The Journey relation, depicts the attributes that are related to the journey, that a user prepares for a particular flight, on a particular date. the journey schema, is a ternary relationship, between the User schema, Flight schema and the FlightSchedule schema. The attributes of the journey relation are as follows:

- id : UUID PRIMARY KEY, NOT NULL
- flightID : FOREIGN KEY NOT NULL
- userID : FOREIGN KEY NOT NULL
- scheduleID : FOREIGN KEY NOT NULL
- metaData : JSON
- amountPayed : FLOAT

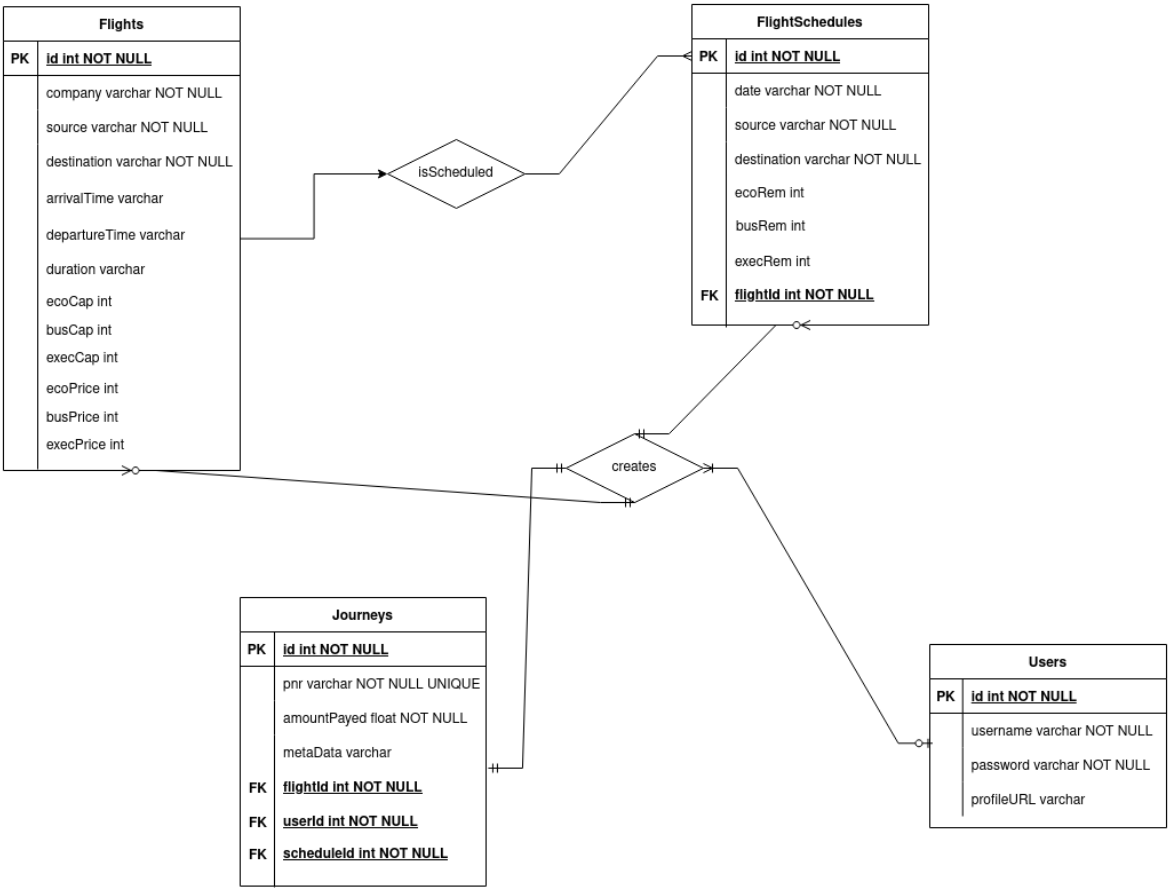

Figure 2.1: Entities and their relationships in the Database.

Chapter 3

SYSTEM PROCESSES

After looking at the database design and different entities interacting with each other, in this chapter, we will discuss about different interlinked process that are executed, when a user tries to book a ticket for a particular journey on **easycruise**. We also discuss about the importance of Business Process Management and how the overall workflow of the booking process is.

3.1 Business Process Management

Process management is a systematic approach to ensure that effective and efficient business processes are in place. It is a methodology used to align business processes with strategic goals. In contrast to project management, which is focused on a single project, process management addresses repetitive processes carried out on a regular basis. It looks at every business process, individually and as a whole, to create a more efficient organization. It analyzes current systems, spots bottlenecks, and identifies areas of improvement.

Process management is a long-term strategy that constantly monitors business processes so they maintain optimal efficiency. Implemented properly, it significantly helps boost business growth. With business processes systematically implemented, you reduce time wasted on repetitive tasks and minimize errors due to human inefficiency. It also prevents the loss of data and missed steps within a process. Moreover, it ensures that resources are used properly so your business becomes more cost-efficient.

3.2 Business Processes in Airline Ticket Booking System

3.2.1 User Authentication

User authentication is a security process that prevents unauthorized users from accessing the contents of our system. It's a login procedure where an application requests personalized passwords to give the user authorized access to it. If a user lacks the proper login rights to the network, their authentication fails. With user authentication, information inputted in the computer for verification is either approved or declined. In cases where the computer declines your request, it shows that you have either entered incorrect information or forgotten your password combination.

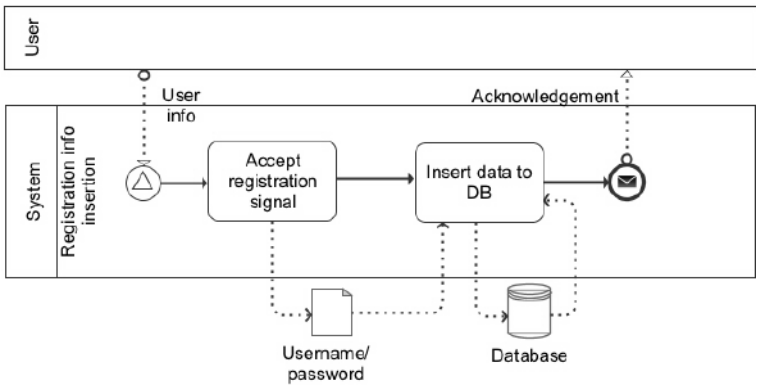

Figure 3.1: BPMN for simple user registration and login for authentication.

3.2.2 Searching Available Flights

The next process in the pipeline, when user authentication successful is to search for available flight options. The user will have to enter the source city/state, the destination city/state, the date of journey, the number of travellers in the journey and the cabin class they want to choose, whether *Economy class*, *Business class* or *Executive class*. Once, these data points have been taken as input, a request will be made to server, which in further process, query the database to determine whether any scheduled flights matching the input parameters is avaible or not. If they are available, it is going to display the list of such flights, else an error message will be returned to the user, depicting that no such flights are available.

3.2.3 Reviewing the Itinerary and Recording Passenger Information

Once the user has selected from one of the available scheduled flights, the next process is to review the journey itinerary and provide the details of the all the passengers who are travelling. This step also involved recording the contact information of the user, so that E-copy of the ticket can be send to those contact once, the booking is confirmed and the seats are allotted.

In the traveller details section, the user has to enter the name, gender, and food preferences of all the travellers. This information is then stored in the database, for further verification during the on-boarding process.

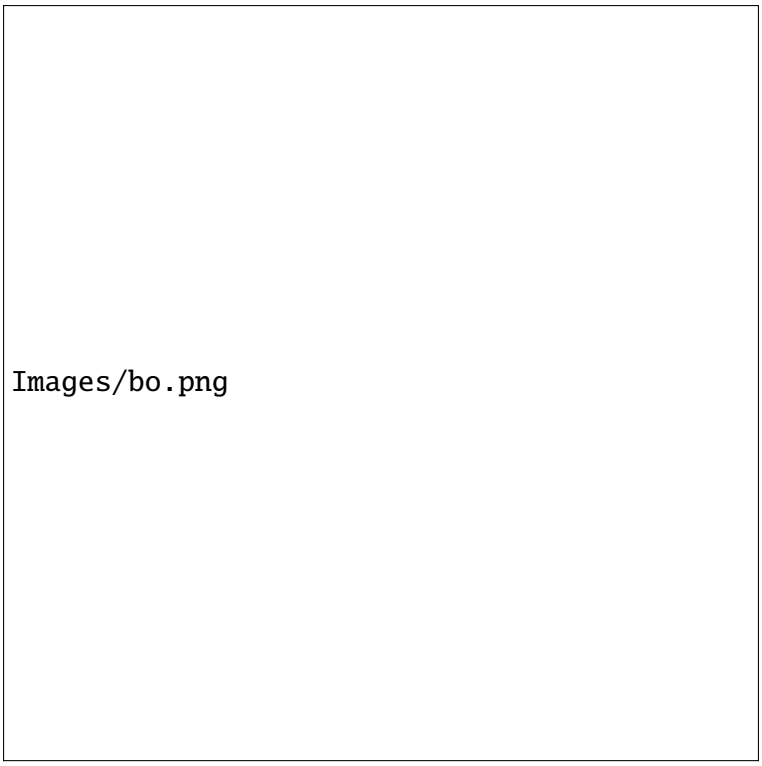

Figure 3.2: BPMN for Reviewing the itinerary and Recording the Passenger's information.

3.2.4 Payment for the tickets

The next process, once the user has reviewed the itinerary and entered the contact information and details of the passengers, the next step is the complete the payment for booking the seats in the aircraft.

The payment portal allows the user to pay the fees online for their airline tickets. There are multiple payment methods available that include:

- Unified Payment Interface (UPI)
- Net Banking
- Credit/Debit Cards

Once, the user has chosen the payment method, and entered the correct validation details, they proceed with the payment. The payment transaction is then recorded in the database, with a user transaction record. A confirmation message is triggered to the user indicating the status of payment, whether it was successful or not.

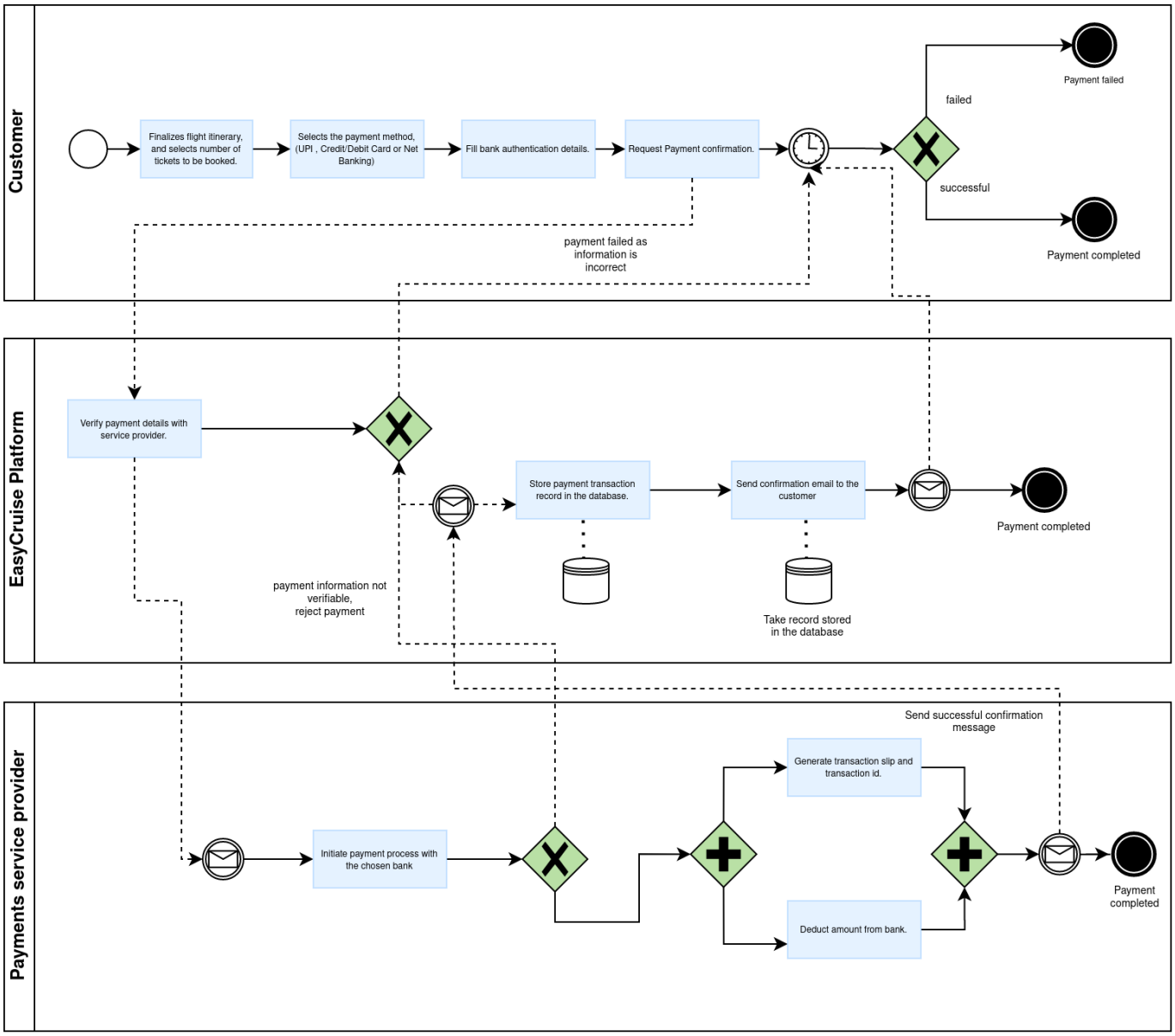

Figure 3.3: BPMN for Payment process while booking tickets.

3.2.5 Final ticket PNR generation and flight confirmation

Once the payment confirmation is complete, and the server return a 200 ok response, a unique ticket PNR *(Passenger Name Record)* is generated, that uniquely identifies a journey record, and is sent to the client.

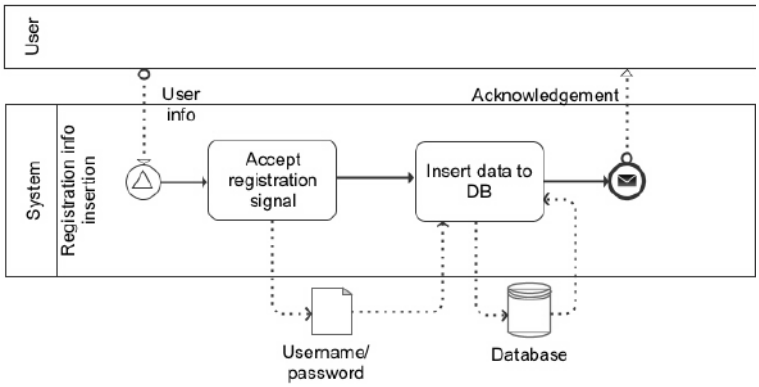

Figure 3.4: BPMN for Payment process while booking tickets.

Chapter 4

CONCLUSION AND FUTURE SCOPE

This project is an attempt to simulate a real-life Airline Ticket Booking System, that allows users to book tickets from a source to destination and also record the traveller's information. The project aims at presenting both the client side views and the admin section view where, the platform admin can monitor the statistics related to system.

There are still many improvements that can be made to make the user booking experience much more simpler and effective.

4.1 Future Innovations

Certiain innovations and improvements that can be made are:

- Improvements can be made in the seat allocation algorithm of the system. Currently, the system uses a naive allocation algorithm, by linearly allocating seats to the passengers. A custom interface, where passengers are able to select from the available seats can be designed for better user experience.
- A feature for providing user discounts while booking the tickets through discount coupons can be implemented to retain and attract more number of users to the platform.
- Implementation of JWT (Javascript Web Token) authentication can be done, over the existing session based user authentication for enhanced user security and authentication control.

APPENDIX

The code files for both the client-side and server can be found on the mentioned Github Repositories, with all the details on how to setup and test a developement server.

Client : https://github.com/Kawaljeet2001/Airline-Ticket-Booking-System-Frontend

Server : https://github.com/gautamanirudh/Airline-Ticket-Booking-System-Backend