VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

ELECTRONIC COMMERCE (CO3027)

Mini-project

VNlook - A mobile travel app supporting online ticket purchases

---

Advisor: Nguyễn Thanh Bình
Students: Nguyễn Lê Nhật Dương - 1952638
Nguyễn Minh Tâm - 1952968
Lương Duy Hưng - 1952747
Huỳnh Phước Thiện - 1952463

HO CHI MINH CITY, DECEMBER 2022

Contents

1 Introduction

1.1 Problem statement

Tourism plays a vital role in promoting Vietnam's economy. The extensive investment and constant rapid development of infrastructure and transport networks aim at bringing more accessibility to major attractions in the country. Vietnam has a long history of influences from the Chinese and the French, and its unique cultural traditions are still maintained. Together with the stunning landscape and breathtaking sceneries, these factors attract over 10 million visitors annually with revenues estimated to be at 3.7 billion USD, according to Vietnam's Ministry of Culture, Sports and Tourism. That is the reason why there are so many travel applications nowadays. Most travel apps support tourists in buying airplane tickets and booking hotels but do not support them in buying tourist attractions tickets. Imagine if you have to wait in a long queue on a shiny day to buy a ticket to Vinpearl Safari, it costs lots of time and effort, not only for you but also for the receptionist.

1.2 Solution/Idea statement

Therefore, we decide to build a mobile app supporting online ticket purchases at any tourist attraction in Vietnam. This travel app will not only show tourist attractions' information but also support tourists to buy attraction tickets by using an e-wallet or mobile banking. After completing the transaction, the app will send to the tourist a QR code and they just need to show the QR code at the ticket booth to join in.

1.3 Survey

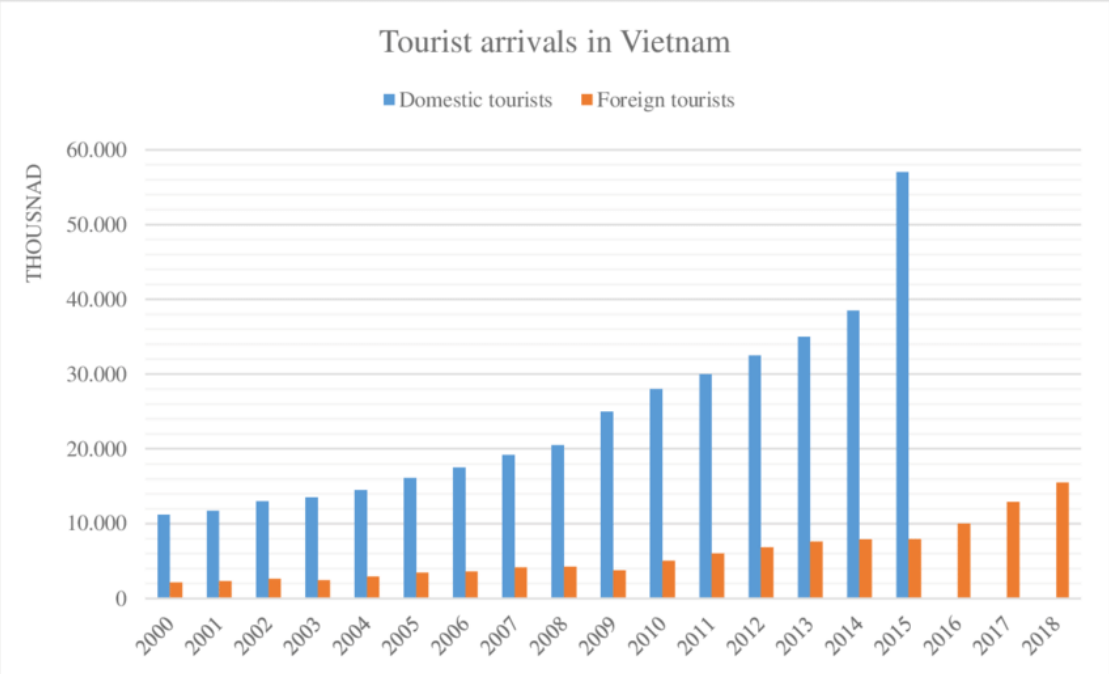

Figure 1: Domestic and International tourists in Vietnam in the period 2000-2018

The recent trends about international visitors in Vietnam are constantly growing according to the updated data of Ministry of Culture, Sport and Tourism, the International Tourist Arrivals (ITA) in the 2018 were about 15.5 million people while at the beginning of 2000 there were about two million (fig.1). Considering also the domestic visitors (about 57 million in 2015; VNAT, 2019), it is quite easy to understand the relevance of touristic movement for this country. Indeed, tourism ensures a direct and indirect contribution the 9.4% of Vietnam's GDP in 2017 and about 7.6% of total employment (World Travel Tourism Council, 2018).

2 Business model

2.1 Business model canvas

2.1.1 Key partner

Our key partner is undoubtedly tourist attractions in Vietnam as they are the motivation for the application.

Our application is dedicated to contactless payment so digital wallet companies and corporations are our favourite partners. Momo will be our biggest partner as their company size is enormous and their popularity among the community is huge. In the future, we will plan on cooperating with others such as Zalopay.

2.1.2 Key activities

Our application facilitates customers to buy tickets to tourist attractions in Vietnam without waiting in a long line. When a purchase is committed, a QR code to a specific location will be sent to their mobile phones.

2.1.3 Key resources

In terms of our profit, we get a commission from attractions at a specific rate. This commission rate varies in negotiation and depends on many factors: location of tourist attraction, tourism carrying capacity, the popularity of tourist attraction, etc.
In regard to humans, the project consists of 3 teams:

- **Developers** who are in charge of programming and maintaining the application.
- **Marketers** who are in charge of promoting the product and figuring out the strategies that can boost sales and revenue.
- **Online supporters** who are in charge of helping users troubleshoot their issues.

2.1.4 Value proposition

- VNlook is an all-in-one tickets counter for tourists who are unfamiliar with the locals or ones who look for contactless payment.
- The application will provide tourists from many regions to buy tickets beforehand, getting rid of lining up in the long queue, especially in the hot summer.
- It also provides users with ease of payment which is connected with the majority of e-wallets available on the market.
- Foreigners can utilize this application to buy tickets regardless of language barriers.

2.1.5 Customer relationships

- Point accumulation and exchange vouchers for regular customers.
- Customer support is available 24/7 to resolve any issue from customers.

2.1.6 Channels

These are the platforms that our application is available:

- iOS Smartphones
- Android Smartphones

2.1.7 Customer segment

The application aims for the following types of customers:

- **Local people** but from other regions who want to explore new places and attractions.
- **Foreign customers** who are from other countries with or without knowledge about local languages.
- **Immature and individual tour guides** who have a casual tour and want to buy tickets beforehand.

2.1.8 Cost structures

These are the cost that we include in our tickets:

- Developing cost
- Marketing and advertising cost
- Third-party payment cost
- App hosting cost

2.2 Revenue stream

- Commission from attraction per unit of ticket
- Cost optimization on development and marketing
- Third-party payment cost
- App hosting cost

3 Pricing model

For this application business, we will apply different pricing models to each of our customer groups - Normal users and Premium users

In terms of *Normal user*, **Demand pricing** model will be applied for the following reasons:

1. Because not many customers are willing for paying an extra amount to become a Premium user, this model will keep attracting consumers, even the one-time buyer.
2. Because most of them are not regular customers, so when the tourist attraction's capacity has reached a threshold (50% for example), an additional fee will be added to the ticket price.

3. With respect to tickets, the majority of the price is fixed and public on the Internet, so other pricing models with the intention to increase that fixed price permanently will cause unwanted effects.
4. This pricing model has currently been applied by Grab - a transportation and delivery e-commerce app and has been running smoothly over years.

In terms of *Premium user*, **Premium pricing** model will be applied for the following reasons:

1. For the Premium type of customers, we need a special way to treat them and make them feel valued.
2. This pricing scheme will become another source of profit made by the application.
3. This plan scheme will help regular customer save money compared to Normal customer scheme

4 Requirements

4.1 Functional requirements

Some of the functional requirements of the desired mobile app are listed below:

- The application shall allow users to see famous landscape information.
- The application shall categorize all the famous landscapes into regions so as to help users search efficiently.
- The application shall allow users to buy online tickets using an e-wallet or mobile banking.
- The application shall alert users if payment is failed.

4.2 Non-functional requirements

Some of the non-functional requirements of the desired mobile app are listed below:

- The application's response time shall be less than 5 seconds.
- The application UI must be user-friendly. Everything must be done in 4-6 steps.
- The application should check if payment is successful or not within 5 seconds after users confirm.
- The application should allow accidental downtime for at most 5 minutes.
- The uptime rate of the mobile app should be above 90% of the time.
- The application should allow only authenticated users to purchase tickets & edit profiles.

5 Mockup design

You can view our mockup design by the following link: Mockup design

5.1 Get Started

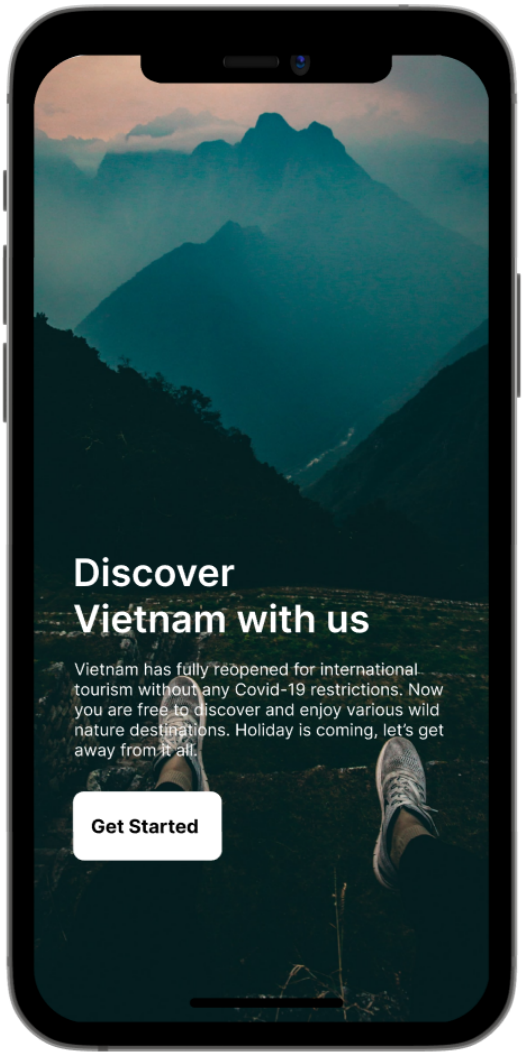

Figure 2: Get Started Screen

5.2 Login And Register

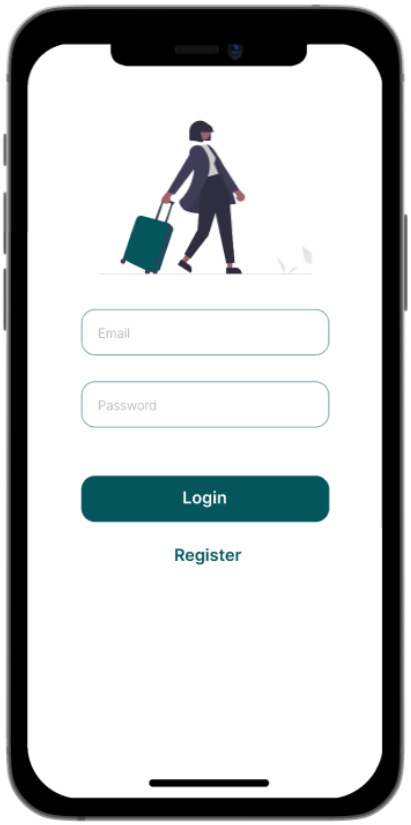

Figure 3: Login screen

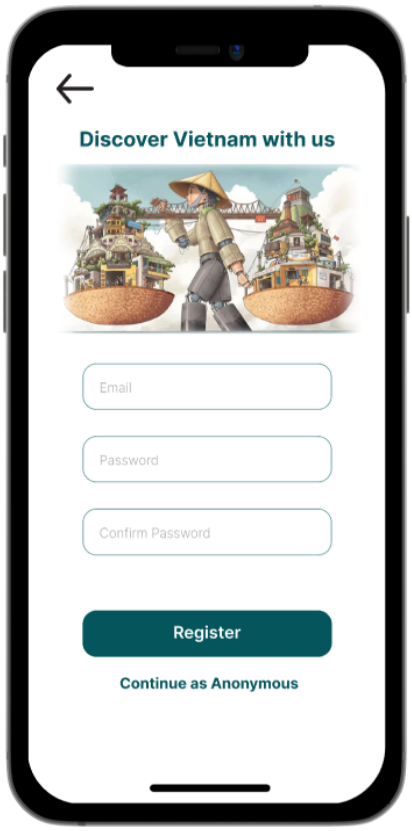

Figure 4: Register screen

5.3 Home Page

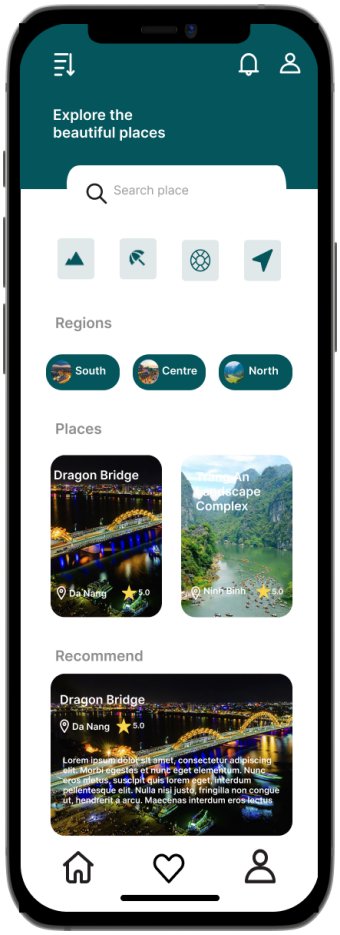

Figure 5: Home Page

5.4 Regions

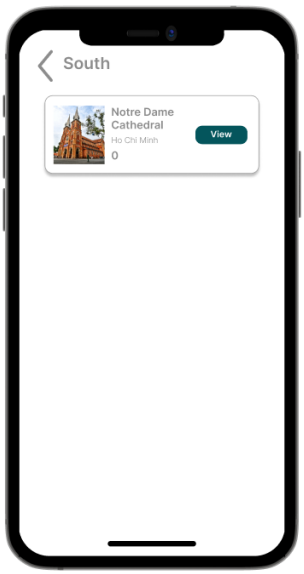

Figure 6: South

Figure 7: Centre

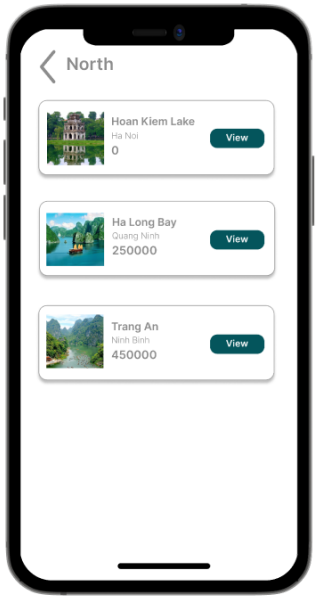

Figure 8: North

5.5 Places Board

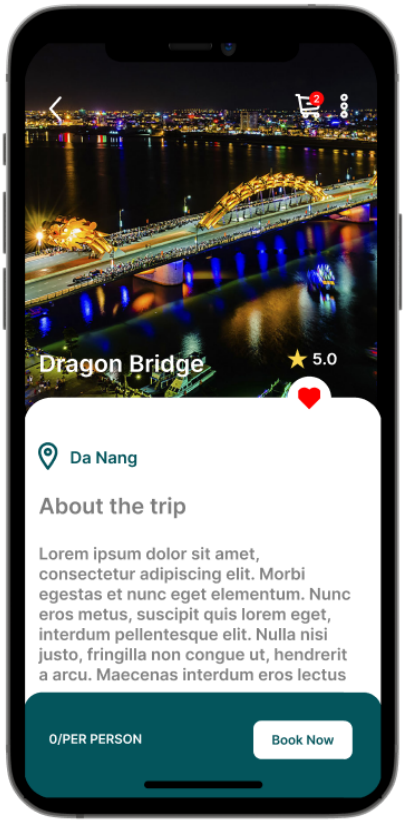

Figure 9: Da Nang

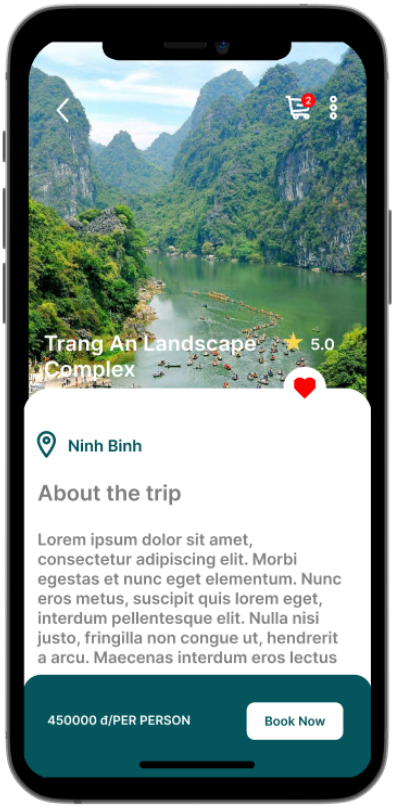

Figure 10: Ninh Binh

5.6 Item Cart

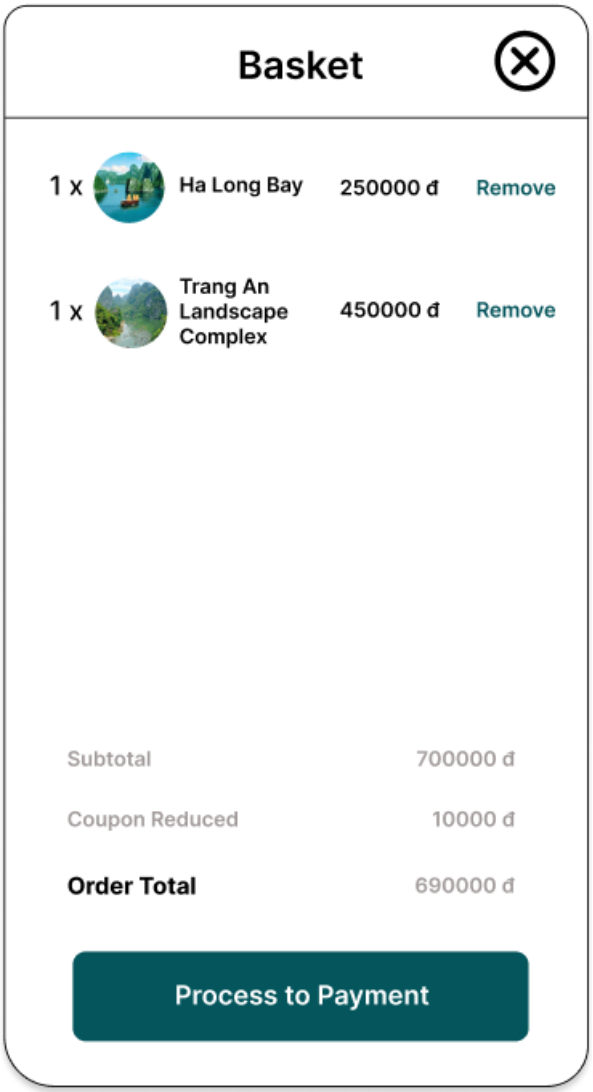

Figure 11: Item Cart

5.7 Payment Information

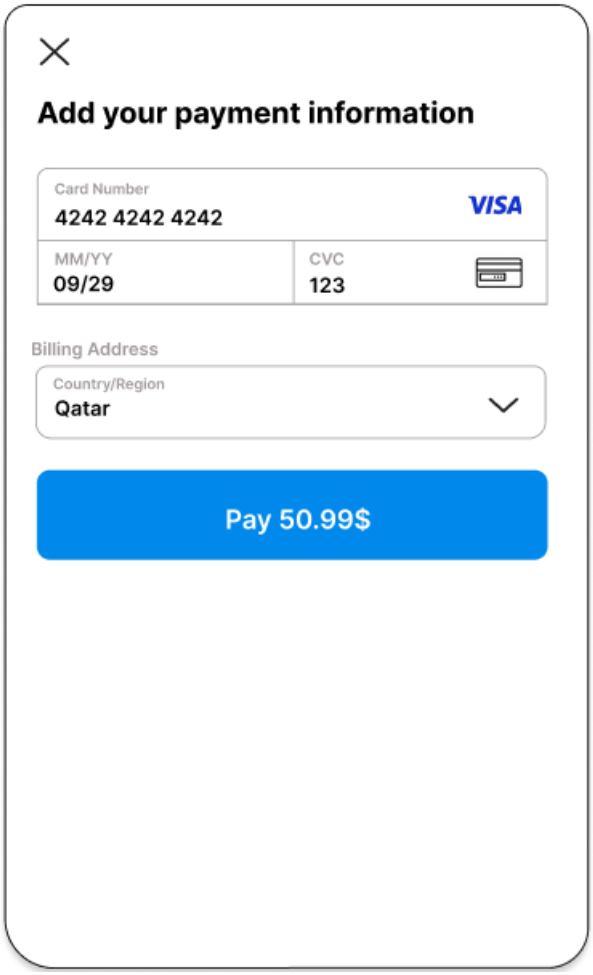

Figure 12: Add Payment Information

5.8 Successful Payment

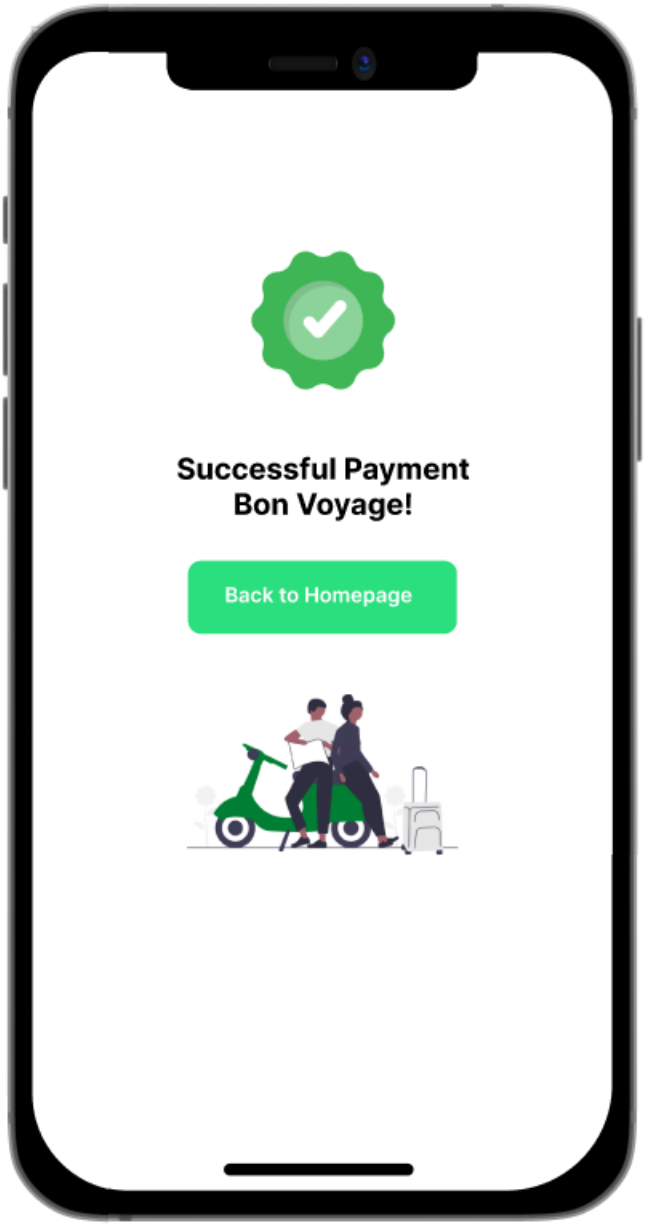

Figure 13: Successful Payment

6 Technology stack

6.1 Front-end

6.1.1 React Native - Client-side code accessible via application pages

React Native is an open-source framework, meaning it's free for public use. It contains resources like pre-built components, libraries, and reference material. Like the framework of a home, React Native provides the basic structure of an application. Developers can then customize it to suit their needs rather than build the application from the ground up.
Some of the advantages that make us choose React Native can be listed as follows:

- **Code reusability**: no need to create separate codes for different platforms
- **Cost efficiency**: many pre-built components that help to fasten the development process
- **Large community support**

6.1.2 Redux - A predictable state container for JavaScript apps

Redux is simply a store to store the state of the variables in your app. Redux creates a process and procedures to interact with the store so that components will not just update or read the store randomly. Similar to the bank. It does not mean because you have money in the bank that you can go anytime, open the vault, and take money. You have to go through certain steps to withdraw money.

Some of the advantages that make us choose Redux can be listed as follows:

- **Predictable**: Redux helps you write applications that behave consistently, run in different environments (client, server, and native), and are easy to test.
- **Centralized**: Centralizing your application's state and logic enables powerful capabilities like undo/redo, state persistence, and much more.
- **Debuggable**: The Redux DevTools make it easy to trace when, where, why, and how your application's state changed. Redux's architecture lets you log changes, use "time-travel debugging", and even send complete error reports to a server.
- **Flexible**: Redux works with any UI layer, and has a large ecosystem of add-ons to fit your needs.

6.2 Backend

6.2.1 Sanity.io - A seamless tool for flexible content management

Sanity.io is the platform for structured content. With Sanity.io you can manage your text, images, and other media with APIs. You can also use the open-source single-page application Sanity Studio to quickly set up an editing environment that you can customize. With Sanity.io you have access to a bunch of APIs, libraries, and tooling that helps you leverage the benefits of having all your content available as a single source of truth.

Some of the advantages that make us choose Sanity can be listed as follows:

- Sanity.io has a real-time datastore for structured content and supporting APIs for assets, user management, and more.

- Sanity Studio is a user interface for managing content. It's an open-source React Single Page Application that you can customize and host wherever you want.
- There are also SDKs, libraries, and tools that let you query your content and integrate it with websites, services, and other applications; wherever you need content.

As we have said earlier, Sanity has provided a bunch of APIs for users to create the database so users will not have difficulty managing the database. Here is the Sanity administration display of our application.

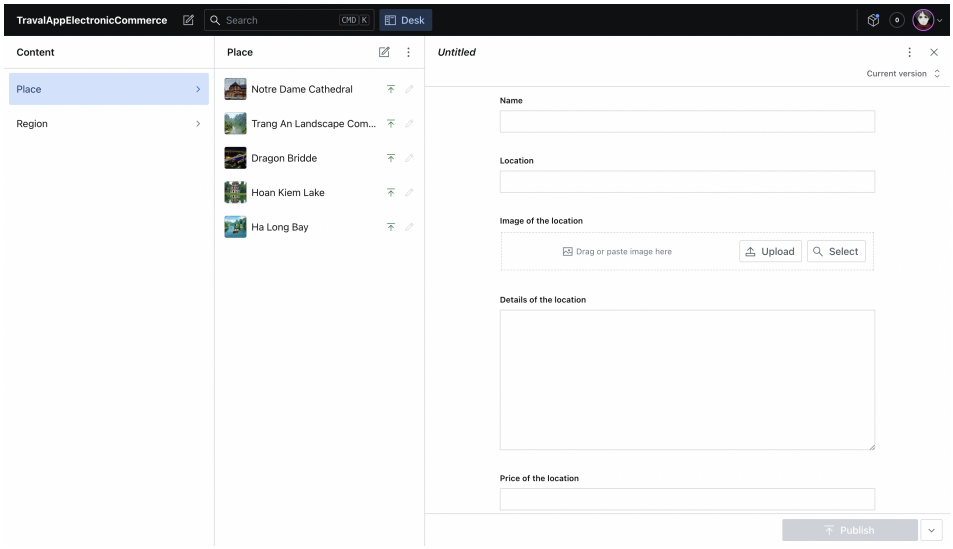

Figure 14: Create new record in Sanity

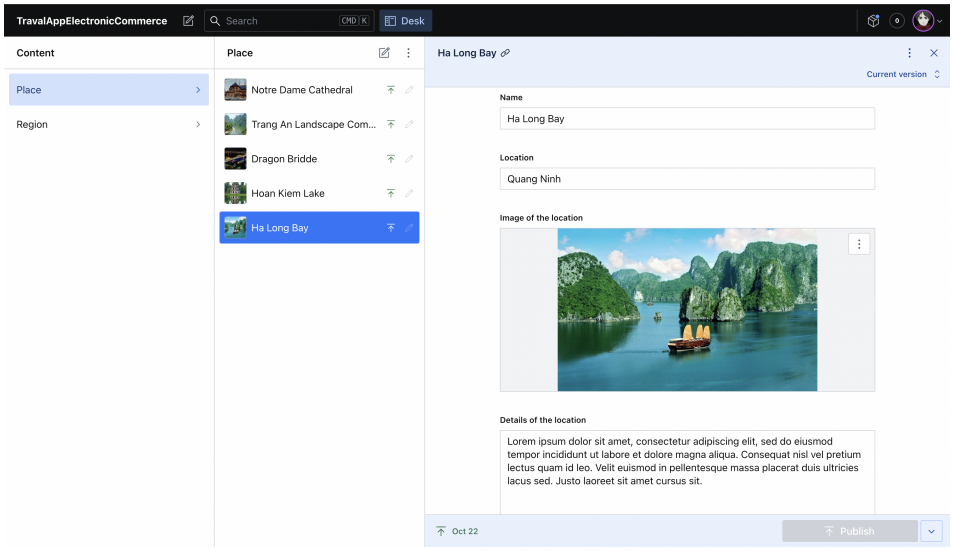

Figure 15: View record in Sanity

6.2.2 Stripe - Payment processing platform

Stripe is a suite of APIs powering online payment processing and commerce solutions for internet businesses of all sizes. Accept payments and scale faster. Stripe is without a doubt one of the most popular online payment gateways you can get.

Some of the advantages that make us choose Stripe can be listed as follows:

- **Close to the metal**: From direct integrations with card networks and banks to checkout flows in the browser, Stripe operates on and optimises at every level of the financial stack.
- **Fastest-improving platform**: Stripe releases hundreds of features and improvements each year to help users stay ahead of industry shifts.
- **Battle-tested reliability**: Stripe's systems operate with 99.99%+ uptime and are highly scalable and redundant. It is also certified to the highest compliance standards.
- **Intelligent optimisations**: Stripe's machine learning models train on billions of data points and help increase revenue across conversion, fraud, revenue recovery, and more.

How does our payment work?

To build a Stripe integration that supports all of our customers, this is what goes on behind the scenes of a card payment.

1. **Create payment intent**
   This front-end would call Stripe server with the payment amount and other necessary information, the back-end will return to the front-end the payment intent information including the customer's information and Stripe secret key.
2. **Checking card details**
   Stripe checks that the details provided are formatted correctly (for example, the expiry date is not in the past). There's no guarantee that the card itself is valid yet.
3. **Customer authentication**
   Some banks, especially in regulated regions like Europe and India, may prompt the customer to authenticate a purchase (for example, by texting the customer a code to enter on the bank's website).
4. **Authorization**
   The bank checks for sufficient funds and, if successful, holds the amount on the customer's account to guarantee it for the merchant.
5. **Capture**
   The money moves from the issuing bank to the merchant's account.

Payment results:

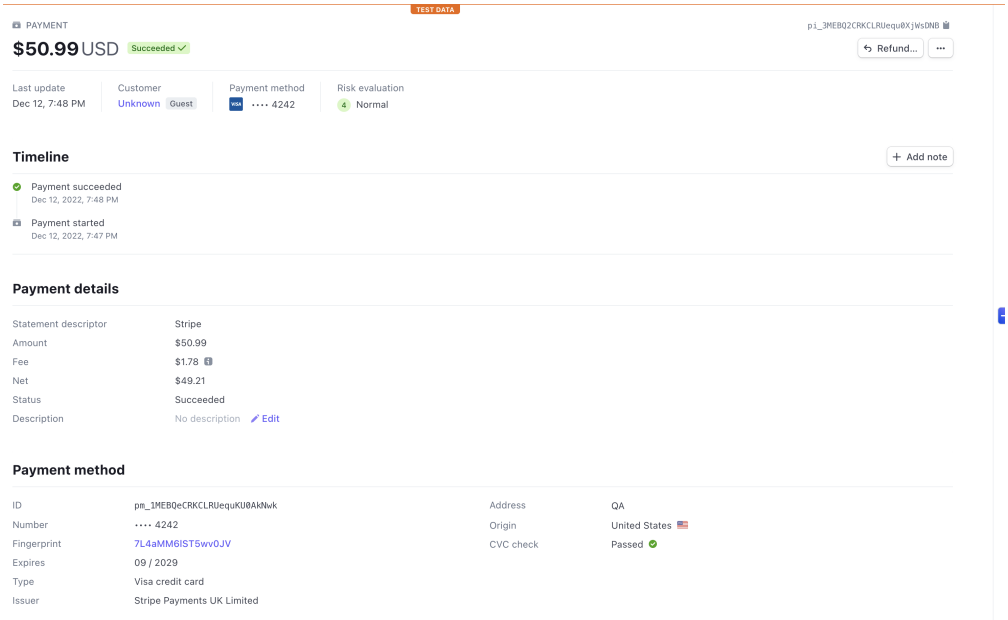

Figure 16: Payment detail saved in my Stripe account

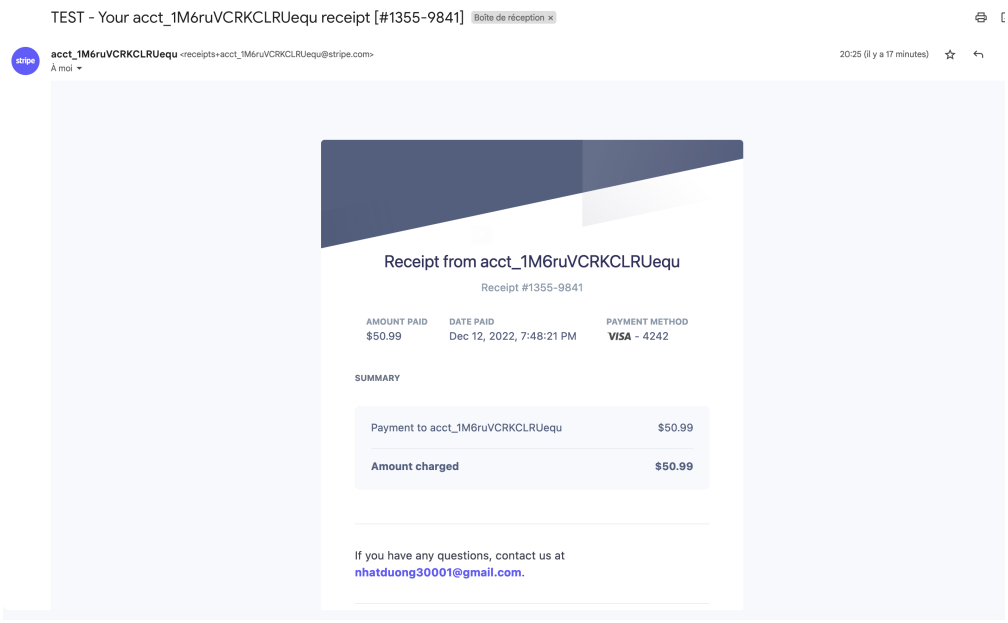

Figure 17: Receipt send to my Email account

7 Conclusion

From this project, we have learned about how to build a mobile app using React Native and integrate a payment gateway into the app. We have also studied how to apply the appropriate business and pricing model to our application. Furthermore, we have tried to apply a new content management system like Sanity.io into our application to act as a database server. When putting them all together, we were able to build a travel app that can support tourists to purchase tickets. This project has taught us many things about applying E-commerce to real applications and we hope to be able to develop this app more in future.