viva_api_tokens
Table that holds just one entry, the currently active API bearer token for Viva.

App/Services/VivaApiTokenService
Has a method that requests a Viva API bearer token and saves it in the database. Has the client ID and client secret hardcoded inside.

packages
Each package has multiple bookings. Each booking gets a package_id when it is created which corresponds to the package it belongs. Besides that the package entries hold only a name and description, the buyer's user id, and its total price (which should be calculated every time a new booking is added to it - so when a booking is made, check its package id and add its price, when it is deleted, reduce the price). It also has a status showing if it's just a cart, or an actual purchased package.

payments
Each payment corresponds to a package. It is created when a package is bought and holds the price, the package id, the order code for viva, and the status (in progress, successful, unsuccessful). Refunding might need a rework, for now it simply holds how much of the payment has been refunded (initialized at 0).

App/HTTP/Controllers/API/VivaOrderController
The controller takes only a package id. It creates an HTTP request to the Viva API, using the total price for the package as the amount, and the user's personal information (email etc). If the request results in a 401 error, it means the token has expired, so the VivaApiTokenService requests a new one and retries. If the request is successful, a new Payment entry is created. The viva OrderCode is returned, which is a numerical code that can be appended to the following URL (which will normally be what the user is redirected to):

https://demo.vivapayments.com/web/checkout?ref=