

Case 1: Double Checkout

- 1) The user has a package with bookings, and presses "Checkout" to pay with Viva
 - a) An order code is generated
 - b) A Payment is created
 - c) The package status is changed to "processing_payment"
 - d) The payment link is sent to the user
- 2) Instead of paying with the first link, the user presses "Checkout" again for whatever reason
 - a) A new order code and Payment is generated

Problem: The user can pay twice (or more) for the same package.

Solution: If the package status is "processing_payment", do not generate new order codes

Problem: In that case, assume the first time the user goes to checkout, they let the order timeout, and then cannot pay at all since the status is still "processing_payment"

Solution: Time-based check? Timeout is 1800 seconds by default (or however much we set it to) so wait for that long, check if the payment went through, and if not revert package status?

Case 2: Insufficient Funds for Refund

- 1) The user wants to refund a booking
- 2) If the booking status is "confirmed", the refund goes from the marketplace account back to the user.

Problem: If the user refunds the same day as the payment, funds have not been transferred to the marketplace account yet so there might not be enough for the refund. This might not be an issue since "refund" is essentially "remove transaction". Test for it.

Solution: Tell the user to try again later?

- 3) If the booking status is "payout_confirmed", then first reverse the payout from marketplace to vendor, then refund to user.

Problem: If the vendor has moved the money from the viva wallet to an external bank, there are no funds to pay for the refund.

Solution: Mark the transaction "to be refunded", pull from funds in the marketplace account, and deduct them from the next payment to that vendor? What if the vendor has no other transactions afterwards? Alert the vendor to put money in their wallet for the refund to go through, and tell the user to try again later?