

PAYMENT METHODS AND APPARATUSES, ELECTRONIC DEVICES, AND READABLE MEDIA

TECHNICAL FIELD

[0001] One or more embodiments of this specification relate to the field of internet
5 of things technologies, and in particular, to payment methods and apparatuses, electronic devices, and readable media.

BACKGROUND

[0002] Currently, after a target user purchases a product, an order QR code can be generated. The target user can scan the order QR code to pay for an order of the product.

10 [0003] If a target user cannot perform payment for various reasons when purchasing a product, the target user can send an order QR code corresponding to an order of the product to another user. The other user performs payment on behalf of the target user by scanning the order QR code.

[0004] In this case, if there are a plurality of other users, it is possible that the
15 plurality of other users duplicately pay for the order. Other users that perform duplicate payments need to perform refund processes after a system completes processing of the order. Alternatively, when the order fails to be paid for by the first user that completes payment, the target user needs to perform a refresh operation. After the system refreshes the order QR code, another user can then perform payment by using a new order QR
20 code.

[0005] It can be seen that in existing payment methods, when users make third-party payments on behalf of others, the QR code of the order cannot be reused until the order is successfully paid for, and the QR code of the order cannot prevent duplicate payments.

SUMMARY

[0006] One or more embodiments of this specification provide payment methods and apparatuses, electronic devices, and readable media. When another user needs to pay for an order on behalf of a target user, a payment method is provided that allows the QR code of the order to be reused until the order is successfully paid for, while
5 avoiding duplicate payments. This helps reduce system pressure and improve user experience.

[0007] To implement the above-mentioned objective, one or more embodiments of this specification provide a payment method, including the following: A payment instruction sent by a first user by scanning an order QR code is received, where the
10 order QR code is generated based on a first order corresponding to a target product, and the first order is generated based on a purchase operation performed by a second user for the target product; the first order is set to an occupied state, where a payment instruction of a user other than the first user is rejected to be executed when the first
15 order is in the occupied state; a second order of the first user is created based on an order identifier of the first order in the occupied state; and a payment operation is performed based on the second order.

[0008] One or more embodiments of this specification further provide a payment apparatus, including: a receiving module, configured to receive a payment instruction
20 sent by a first user by scanning an order QR code, where the order QR code is a QR code generated based on a first order corresponding to a target product, and the first order is generated based on a purchase operation performed by a second user for the target product; a setting module, configured to set the first order to an occupied state, where a payment instruction of a user other than the first user is rejected to be executed
25 when the first order is in the occupied state; a creation module, configured to create a second order of the first user based on an order identifier of the first order in the occupied state; and a third-party payment module, configured to perform a payment operation based on the second order.

[0009] The payment methods, apparatuses, electronic devices, and readable media provided in the one or more embodiments of this specification have at least the
30 following beneficial effects: In the embodiments of this specification, first, a payment instruction sent by a first user by scanning an order QR code is received, where the order QR code is generated based on a first order corresponding to a target product, and

the first order is generated based on a purchase operation performed by a second user for the target product; then, the first order is set to an occupied state, where a payment instruction of a user other than the first user is rejected to be executed when the first order is in the occupied state; then, a second order of the first user is created based on an order identifier of the first order in the occupied state; and finally, a payment operation is performed based on the second order. It can be seen that in the embodiments of this specification, after the payment instruction is received, the order is locked, so that another user cannot pay for the first order when the first user pays for the first order. In addition, because the payment operation is performed based on the second order created based on the first order, the first order is not directly processed, and even if the first user fails to perform payment, only the second order is affected, and the first order is not affected, so that the order QR code can be further reused, thereby achieving technical effects of preventing duplicate payments while ensuring that the order QR code can be reused before the first order of the target product is successfully paid for.

BRIEF DESCRIPTION OF DRAWINGS

[0010] To describe the technical solutions in the embodiments of this specification or in the existing technology more clearly, the following briefly describes the accompanying drawings needed for describing the embodiments or the existing technology. Clearly, the accompanying drawings in the following descriptions merely show some embodiments of this specification, and a person of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.

[0011] FIG. 1 is a schematic diagram illustrating a structure of a system of a payment method according to one or more embodiments of this specification;

[0012] FIG. 2 is a schematic flowchart illustrating a payment method according to one or more embodiments of this specification;

[0013] FIG. 3 is another schematic flowchart illustrating a payment method according to one or more embodiments of this specification;

[0014] FIG. 4 is still another schematic flowchart illustrating a payment method according to one or more embodiments of this specification;

[0015] FIG. 5 is still another schematic flowchart illustrating a payment method according to one or more embodiments of this specification;

[0016] FIG. 6 is still another schematic flowchart illustrating a payment method according to one or more embodiments of this specification;

5 [0017] FIG. 7 is a lane diagram illustrating a payment method according to one or more embodiments of this specification;

[0018] FIG. 8 is a schematic diagram illustrating a structure of a payment apparatus according to one or more embodiments of this specification; and

[0019] FIG. 9 is a schematic diagram illustrating a structure of an electronic device
10 according to one or more embodiments of this specification.

DESCRIPTION OF EMBODIMENTS

[0020] To make the objectives, technical solutions, and advantages of one or more embodiments of this specification clearer, the following clearly and comprehensively describes the technical solutions in one or more embodiments of this specification with
15 reference to specific embodiments of this specification and the corresponding accompanying drawings. Clearly, the described embodiments are merely some but not all of embodiments of this specification. All other embodiments obtained by a person of ordinary skill in the art based on embodiments of this specification without creative efforts shall fall within the protection scope of one or more embodiments of this
20 specification.

[0021] The following describes in detail the technical solutions provided in the embodiments of this specification with reference to the accompanying drawings.

[0022] In the existing technology, when a buyer purchases a product, for example, shops online or performs consumption in a brick-and-mortar store, a related system, for
25 example, a payment system or a code platform, can generate an order QR code based on an order of the product after determining that the buyer purchases the product. The order QR code is a form of QR code, and can include an amount and product information. When the order QR code is scanned to perform a payment operation, the order can be paid for without requesting a user to enter the amount. When the buyer
30 needs third-party payment, the buyer can forward the order QR code to another user. The other user scans the order QR code to perform third-party payment.

[0023] For example, the order QR code is sent to a group chat, so that other users in the group chat can perform third-party payment. In this case, in the group chat, if payment time of user A is too long and user B also performs payment, it is possible that user A also successfully performs payment after user B successfully performs payment, that is, the order of the product is duplicately paid for.

[0024] Currently, in case of duplicate payments, the payment system needs to initiate a refund operation for user A or user B when performing unified settlement. Therefore, the payment system needs to perform an extra process, and further needs to determine a refund target user, resulting in a waste of system resources. In addition, because both user A and user B successfully perform payments, it is possible that both user A and user B initiate refund applications. It further needs manual coordination to complete accurate refunding. Clearly, duplicate payments bring extra overheads to the system and increases labor costs, and user experience is very poor.

[0025] In addition, if a payment operation of user A is earlier than a payment operation of another user but fails to perform payment due to an insufficient balance or another factor, because the transaction is unique, in this case, the order QR code corresponding to the order of the product becomes invalid. When another user wants to perform third-party payment, the other user cannot continue to use the order QR code to complete payment. Therefore, the buyer needs to actively refresh the order QR code and then send a new order QR code to the third-party payer. Generation of the new order QR code brings extra overheads to the related system, for example, the payment system or the code platform, and the order QR code used to pay for the order of the product cannot be duplicately used.

[0026] In view of this, a payment method should be provided, so that duplicate payments can be further prevented while an order QR code can be duplicately used, to reduce system overheads, reduce extra user operations, and improve user experience.

[0027] FIG. 1 is a schematic diagram illustrating a structure of a system of a payment method according to one or more embodiments of this specification.

[0028] User 20 can establish a communication connection to code platform 10 by scanning order QR code 30. Certainly, there can be a plurality of users 20, and code platform 10 can be further connected to a unified payment system. The unified payment system can be a system that can perform payment by using a plurality of payment platforms.

[0029] FIG. 2 is a schematic flowchart illustrating a payment method according to one or more embodiments of this specification.

[0030] As shown in FIG. 2, the payment method provided in the one or more embodiments of this specification can include the following steps.

5 **[0031]** Step 200: Receive a payment instruction sent by a first user by scanning an order QR code. The order QR code is generated based on a first order corresponding to a target product, and the first order is generated based on a purchase operation performed by a second user for the target product. In the one or more embodiments of this specification, the order QR code can correspond to the first order. When the
10 purchase operation performed by the second user for the target product is obtained, a process of generating the first order based on the target product and generating the order QR code corresponding to the first order based on the first order is triggered. The second user can purchase the target product online, for example, by logging in to a shopping website by using a browser, or certainly, can purchase the target product offline, for
15 example, in a self-service purchase method. It can be understood that the target product can be a physical product such as an electronic device, furniture, a home appliance, a daily article, or fuel, or can be a virtual product such as a game card, a prop, equipment, or a value-added service. Implementations are not limited here, provided that the order QR code can be generated based on the first order after the first order is generated based
20 on the target product.

[0032] In some embodiments, the order QR code is generated by a code platform, and the code platform can be a part of a payment system, or can be a separate platform system.

25 **[0033]** The order QR code can include information about the target product, an amount of the target product, an order number of the order corresponding to the target product, etc. Specific included information can be set based on an actual need.

[0034] The first user can scan the order QR code by using applications of different payment platforms, to implement third-party payment for the second user. Certainly, if the order QR code is scanned by using different applications, payment platforms
30 corresponding to the applications can also be different.

[0035] Step 202: Set the first order to an occupied state. A payment instruction of a user other than the first user is rejected to be executed when the first order is in the occupied state. In the one or more embodiments of this specification, after the payment instruction is received, the first order can be set to the occupied state. Only an occupant

can perform a payment operation on the first order in the occupied state, and a payment instruction of a user other than the first user is rejected to be executed when the first order is in the occupied state. For example, after the first user locks the first order, only the first user can perform a payment operation on the first order in a locking process.

- 5 In this case, if another user, for example, a third user, attempts to pay for the first order, because the first order is in the occupied state, a payment operation of the third user is not performed.

[0036] In the one or more embodiments of this specification, the order can be set to the occupied state by locking a data row in a database or by setting a parameter, for
10 example, setting a non-occupiable time point.

[0037] In the one or more embodiments of this specification, the setting the first order to an occupied state includes: locking a data row, in a database, that the first order is located in.

[0038] The setting the first order to an occupied state includes: setting a non-
15 occupiable time point to a sum of a current time point and a predetermined non-occupiable time parameter.

[0039] In the one or more embodiments of this specification, the setting the first order to an occupied state can include at least two different methods.

[0040] In some embodiments, the data row, in the database, that the first order is
20 located in can be locked. For example, the data row can be locked by using a database row lock. The database row lock allows only one user to perform an operation at one time. When the first user obtains the database row lock to lock the data row, other users cannot obtain the database row lock and cannot perform a data row locking operation, that is, cannot implement an operation of setting the first order to the occupied state.
25 This approach can achieve exclusivity, especially in situations with a high level of concurrency and very short time intervals between operations.

[0041] For example, in the third-party payment scenario in the one or more
embodiments of this application, the first user and the third user separately obtain the database row lock at an interval of 50 ms, to lock the data row of the first order in the
30 database, to complete the operation of setting the first order to the occupied state. Because the first user sends the payment instruction at a time point earlier than the third user sends the payment instruction, when the third user attempts to obtain the database row lock, the third user is made to wait, or error information can be directly returned to block the payment operation of the third user.

[0042] In this method, the database needs to be occupied. However, because response time of the database is short, locking can be performed provided that the database row lock is not used, without use of other settings or algorithms or a worry about a network fluctuation problem.

5 **[0043]** Another implementation is setting the non-occupiable time point instead of occupying the database. The non-occupiable time point is used to represent a time point that a payment instruction of a user other than the first user can be executed at. The non-occupiable time point indicates that currently, it is possible that another user sets occupation time and is performing a payment operation. For example, the non-occupiable time point is null or zero by default. When a time point that the payment instruction sent by the first user is received at is not earlier than the non-occupiable time point, the non-occupiable time point can be set to the sum of the current time point and the predetermined non-occupiable time parameter. For example, if the current time point is 12:00:00, and the predetermined non-occupiable time parameter is 2 seconds, 10 the non-occupiable time point is set to 12:00:02. The non-occupiable time parameter can be set based on an actual need. The predetermined non-occupiable time parameter can be used to represent duration for processing a payment instruction. Certainly, the predetermined non-occupiable time parameter can be set with reference to use duration of the database row lock, can be set based on actual duration, or can be updated based 15 on a subsequent need. 20

[0044] In the method of setting the non-occupiable time point, the database does not need to be occupied, so that resources of the database can be saved. However, because an algorithm operation needs to be performed for parameter setting, it is possible that setting fails due to network fluctuation. However, setting can be accelerated with constant improvement of hardware performance and network performance, to prevent a setting failure. 25

[0045] In the one or more embodiments of this specification, regardless of a method, the order can be set to the occupied state, so that another user cannot perform a payment operation, to make a data preparation for preventing duplicate payments.

30 **[0046]** Step 204: Create a second order of the first user based on an order identifier of the first order in the occupied state. In the one or more embodiments of this specification, the second order of the first user can be created based on the order identifier of the first order that is set to the occupied state. The order identifier can be obtained by matching an order code in the payment instruction, and the order code can

enable the first user, the first order, and the second order to have an association relationship.

5 [0047] In the one or more embodiments of this specification, different users can correspond to different second orders. The first order can correspond to a plurality of second orders. Even if the second order fails to be paid for or an error occurs, a status of the first order is not affected. Therefore, a status of the order QR code corresponding to the first order is not affected, so that the order QR code can be duplicately used before payment of the first order is completed, and the order QR code corresponding to the first order does not need to be actively or passively refreshed.

10 [0048] After the second order is created, a data table of the second order can be correspondingly created in the database. A data table of the first order can correspond to data tables of a plurality of second orders. The data table of the first order and the data table of the second order have an association relationship, for example, have a same order code. Each second order can correspond to a different user identifier to distinguish between different users. Certainly, the data table of the second order can further have other information related to the first order and the user. Specific implementations are not limited here.

[0049] Step 206: Perform a payment operation based on the second order.

20 [0050] In the one or more embodiments of this specification, when the second order is generated, the payment operation can be performed based on the second order, to complete a third-party payment operation performed by the first user for the second user.

25 [0051] In the one or more embodiments of this specification, the payment operation can be completed based on the second order created based on the identifier of the first order in the occupied state. When the first order is in the occupied state, even if the second order fails to be paid for, the order QR code corresponding to the first order can be duplicately used. In addition, because only one user can successfully pay for the first order in the occupied state, duplicate payments can be prevented while it is ensured that the order QR code can be duplicately used.

30 [0052] In the above-mentioned one or more embodiments, the process of setting the first order to the occupied state is described. In the one or more embodiments of this specification, before the first order is set to the occupied state, it can be further determined whether the first order is in the occupied state.

[0053] FIG. 3 is another schematic flowchart illustrating a payment method according to one or more embodiments of this specification.

[0054] On the basis of the above-mentioned one or more embodiments, before the setting the first order to an occupied state in step 202, the method further includes the following step. Step 300: Receive a payment instruction sent by a first user by scanning an order QR code.

[0055] Step 302: Determine whether a first order is in an occupied state. Step 304 is performed if it is determined that the first order is not in the occupied state.

[0056] Step 304: Set the first order to the occupied state.

[0057] Step 306: Create a second order of the first user based on an order identifier of the first order in the occupied state.

[0058] Step 308: Perform a payment operation based on the second order.

[0059] For step 300, step 304, step 306, and step 308, references can be made to step 200, step 202, step 204, and step 206 described in the above-mentioned one or more embodiments, and details are omitted here for simplicity.

[0060] The determining whether the first order is in an occupied state can specifically include: determining whether a data row, in a database, that the first order is located in is locked; and determining that the first order is not in the occupied state if it is determined that the data row is not locked.

[0061] The determining whether the first order is in an occupied state can specifically include: determining whether a current time point is earlier than a non-occupiable time point, where the non-occupiable time point is used to represent a time point that is set when a payment instruction is received at a previous time; and determining that the first order is not in the occupied state if it is determined that the current time point is not earlier than the non-occupiable time point.

[0062] In the one or more embodiments of this specification, the determining whether the first order is in an occupied state can be determining whether the data row, in the database, that the first order is located in is locked, or can be determining whether the current time point is earlier than the non-occupiable time point.

[0063] In the one or more embodiments of this specification, the non-occupiable time point is the time point that is set when the payment instruction is received at the previous time. It can be seen from the above-mentioned one or more embodiments that the non-occupiable time point can be set based on a receiving time point of the payment instruction when the payment instruction is received at the previous time, to indicate

that a user occupies the first order and is performing a payment operation on the first order.

5 **[0064]** In the one or more embodiments of this specification, if the data row is locked, that is, a database row lock has been obtained by a user, it means that the user is performing a payment operation on the first order. Therefore, it should be determined that the first order is in the occupied state. Otherwise, if the data row in the database is not locked, it is determined that the first order is not in the occupied state, and a subsequent operation can be performed.

10 **[0065]** In the one or more embodiments of this specification, before the first order is set to the occupied state, it can be determined whether the first order has been set to the occupied state, and the first order can be set to the occupied state if the first order has not been set to the occupied state.

15 **[0066]** In the one or more embodiments of this specification, prompt information indicating that the first order is being processed is sent to the first user if it is determined that the first order is in the occupied state.

[0067] In the one or more embodiments of this specification, if the first order is in the occupied state, the prompt information indicating that the first order is being processed, for example, prompt information "being processed", can be sent to the first user, and a subsequent operation process of paying for the first order can be prevented.

20 **[0068]** In the one or more embodiments of this specification, the determining that the first order is in the occupied state can specifically include: determining whether the data row, in the database, that the first order is located in is locked after a predetermined waiting time interval if it is determined that the data row is locked; and determining that the first order is in the occupied state if the data row, in the database, that the first order is located in is locked after the predetermined waiting time interval.

25 **[0069]** In the one or more embodiments of this specification, if the first order is in the occupied state, it is also possible to wait for a predefined time interval. The predetermined time interval can be set based on an actual need, for example, 2 seconds. The predetermined time interval can be a time period that is calculated after the payment instruction is received and that is occupied by generating a payment code corresponding to the second order and then sending the payment code to a unified payment platform, and receiving a message, for example, an acceptance message, returned by the unified payment platform. The predetermined time interval can be adjusted based on an actual need.

[0070] If the data row is still locked after the wait for the predetermined time interval, it means that it is possible that payment processing of a previous user is not completed due to a system problem or network fluctuation, and it is determined that the first order is still in the occupied state.

5 **[0071]** In the one or more embodiments of this specification, to further avoid duplicate payments, it can be further determined whether the current time point is earlier than a non-payable time point.

[0072] FIG. 4 is still another schematic flowchart illustrating a payment method according to one or more embodiments of this specification.

10 **[0073]** Step 400: Receive a payment instruction sent by a first user by scanning an order QR code.

[0074] Step 402: Set a first order to an occupied state.

[0075] Step 404: Determine whether a current time point is earlier than a non-payable time point. Step 406 can be performed if the current time point is not earlier
15 than the non-payable time point.

[0076] Step 406: Create a second order of the first user based on an order identifier of the first order in the occupied state.

[0077] Step 408: Perform a payment operation based on the second order.

[0078] In the one or more embodiments of this specification, for step 400, step 402, step 406, and step 408, references can be respectively made to step 200, step 202, step 204, and step 206 in the above-mentioned one or more embodiments, and details are
20 omitted here for simplicity.

[0079] In the one or more embodiments of this specification, to further avoid duplicate payments, before the second order of the first user is created based on the order identifier of the first order in the occupied state, it can be further determined
25 whether the current time point falls within a time period of a payment operation performed by another user.

[0080] In the one or more embodiments of this specification, step 404 can be a determining process performed after a database row lock is unlocked. Therefore, in step
30 404, before it is determined whether the current time point is earlier than the non-payable time point, it can be further determined whether a data row, in a database, corresponding to the first order at the current time point has been unlocked. Step 404 is performed if the data row has been unlocked. Otherwise, step 404 is performed after

the data row is unlocked, or error prompt information, for example, prompt information "the order is being processed", is returned to the first user.

[0081] In the one or more embodiments of this specification, an extreme case in a payment process further needs to be considered. For example, after the first user locks the data row of the first order in the database, if the first order is being paid for by another user, for example, when a third user initiates a checkout process to perform a payment operation, in this case, although the data row of the first order in the database has been unlocked, that is, the database row lock has been released, the subsequent step of creating the second order should not be performed, and the second order can be created only when the current time point is not earlier than the non-payable time point.

[0082] It can be understood that a user cannot be in a payment state for long time. Therefore, the non-payable time point can also guarantee another user will be unable to perform payment when a user suspends a payment operation for long time.

[0083] Based on this, in the one or more embodiments of this specification, it can be further determined whether the current time point is earlier than the non-payable time point. In the one or more embodiments of this specification, the non-payable time point can be a time point after a predetermined payment time interval after a time point that the payment instruction is received at, or can be a time point after a predetermined payment time interval after a time point that a message, for example, an acceptance message, sent by a unified payment platform is received at. Certainly, the non-payable time point can be alternatively a time point after a predetermined payment time interval after a time point that step 404 is triggered at.

[0084] The non-payable time point can be set based on an actual need. For example, duration of a payment operation performed by a user from payment page redirecting to payment completion usually does not exceed 3 minutes. Therefore, the non-payable time point can be set to the third minute after the time point that the payment instruction is received at, so that the user can successfully perform the payment operation. For example, after 12:00, if the predetermined payment time interval is 3 minutes, the non-payable time point is 12:03.

[0085] Certainly, if the non-payable time point needs to be subsequently changed, the predetermined payment time interval can be further modified. For example, if the predetermined payment time interval is modified to 1 minute, the non-payable time point can be 12:01. That is, the non-payable time point in the one or more embodiments

of this specification can be set and updated based on an actual need, and is a parameter that can reflect duration of a payment operation of a user.

5 **[0086]** In the one or more embodiments of this specification, it is determined that the first order is not in a payment process when it is determined that the current time point is not earlier than the non-payable time point. The process of step 406 can be performed.

[0087] It can be understood that step 404 can be alternatively a process performed before step 402, and an execution sequence of step 404 can be adjusted based on an actual need.

10 **[0088]** On the basis of the above-mentioned one or more embodiments, prompt information can be returned to the first user if it is determined that the first order is in the payment process.

[0089] In the one or more embodiments of this specification, the method further includes: determining that the first order is in the payment process if it is determined
15 that the current time point is earlier than the non-payable time point; and sending, to the first user, prompt information indicating that the first order is being paid for.

[0090] In the one or more embodiments of this specification, if it is determined that the first order is in the payment process, that is, a second order corresponding to the first order is being processed by the unified payment platform, prompt information, for
20 example, prompt information "the order is being paid for", can be sent to the first user, and a subsequent process can be interrupted, for example, the process of generating the second order of the first user is not performed.

[0091] In the one or more embodiments of this specification, to further avoid duplicate payments, it can be further determined whether the first order is in the
25 payment process, and a prompt can be given to the user.

[0092] In the one or more embodiments of this specification, it can be further determined whether the first order is in a payable state, for example, whether the first order has been paid for.

[0093] FIG. 5 is still another schematic flowchart illustrating a payment method
30 according to one or more embodiments of this specification.

[0094] Step 500: Receive a payment instruction sent by a first user by scanning an order QR code.

[0095] Step 502: Determine whether a first order is in a payable state. Step 504 can be performed if the first order is in the payable state.

[0096] Step 504: Set the first order to an occupied state.

[0097] Step 506: Create a second order of the first user based on an order identifier of the first order in the occupied state.

[0098] Step 508: Perform a payment operation based on the second order.

5 **[0099]** The method further includes: sending, to the first user, a prompt message indicating that the first order has been paid for if it is determined that the first order is not in the payable state.

[0100] In the one or more embodiments of this specification, for step 500, step 504, step 506, and step 508, references can be made to step 200, step 202, step 204, and step 10 206 in the above-mentioned one or more embodiments, and details are omitted here for simplicity.

[0101] In the one or more embodiments of this specification, it can be further determined whether the first order is in the payable state. The payable state can indicate that a payment operation can be performed on the first order. It is determined that the 15 first order is not in the payable state if it is determined that the first order has been paid for; or it is determined that the first order is in the payable state if it is determined that the first order is not paid for. The second order of the first user can be created based on the order identifier of the first order in the occupied state when the first order is in the payable state.

20 **[0102]** A prompt message indicating that the first order has been paid for is sent to the first user if it is determined that the first order is not in the payable state.

[0103] Prompt information "has been paid for" can be sent to the user when the first order has been paid for. For example, it can be determined, by using a field, in a data row that the first order is located in, indicating whether the first order is paid for, 25 whether the first order has been paid for. Alternatively, it can be determined, by using an order status of the first order, whether the first order has been paid for. The order status can be an attribute parameter corresponding to the order, and it can be determined, by using the attribute parameter, whether the first order has been successfully paid for.

[0104] In the one or more embodiments of this specification, it is determined that 30 the first order is not in the payable state if the first order has been paid for, and a subsequent payment process of step 504 to step 508 cannot be performed on the first order that is not in the payable state. The subsequent payment process of step 504 to step 508 can be performed on the first order in the payable state.

[0105] In the one or more embodiments of this specification, to save system resources, it can be further determined whether the first order is in the payable state.

[0106] Certainly, it can be alternatively determined, in the following method, whether the first order is in the payable state.

5 **[0107]** Before the creating a second order of the first user based on an order identifier of the first order in the occupied state, the method further includes: determining that the first order is in a payable state if it is determined that a transaction corresponding to the first order is in an unclosed state; or determining that the first order is not in the payable state if it is determined that the transaction corresponding to the first order is in a closed state.

[0108] The method further includes: sending, to the first user, a prompt message indicating that the first order cannot be paid for if it is determined that the first order is not in the payable state.

15 **[0109]** In the one or more embodiments of this specification, it means that the transaction of the first order is invalid if the transaction corresponding to the first order is in the closed state. For example, the transaction corresponding to the first order is in the closed state if a certain flash sale activity has expired. Alternatively, the transaction corresponding to the first order should also be in the closed state if the second user actively cancels the transaction for the first order. A transaction status change can be
20 alternatively caused by other cases, and no enumeration is made here.

[0110] In the one or more embodiments of this specification, a prompt message "cannot be paid for", for example, "the order cannot be paid for" or "the transaction of the order has been closed", can be sent to the first user if it is determined that the transaction of the first order is in the closed state.

25 **[0111]** In the one or more embodiments of this specification, an execution sequence of step 502 can be set based on an actual need, to reduce unnecessary overheads of the system.

[0112] The above-mentioned one or more embodiments describe the process of performing the payment operation based on the second order. The following describes
30 the process in detail.

[0113] FIG. 6 is still another schematic flowchart illustrating a payment method according to one or more embodiments of this specification.

[0114] Step 600: Receive a payment instruction sent by a first user by scanning an order QR code.

[0115] Step 602: Set a first order to an occupied state.

[0116] Step 604: Create a second order of the first user based on an order identifier of the first order in the occupied state.

[0117] Step 606: Send a payment code of the second order to a unified payment platform. The unified payment platform can generate payment redirect page address information based on the payment code of the second order and payment platform information of an application used by the first user to scan the order QR code, and send the payment redirect page address information to the application, so that the first user performs the payment operation in a payment interface popped up by the application based on the payment redirect page address information.

[0118] In the one or more embodiments of this specification, for an execution process of step 600, step 602, and step 604, references can be made to the execution process of step 200, step 202, and step 204 in the above-mentioned one or more embodiments, and details are omitted here for simplicity.

[0119] In the one or more embodiments of this specification, payment operations of a plurality of payment platforms can be supported.

[0120] To implement payment operations of a plurality of payment platforms, in the one or more embodiments of this specification, the payment code of the second order can be sent to the unified payment platform. The payment code can be generated based on the second order, and can include related payment information such as an amount of a target product and an identifier of the first user.

[0121] After the payment code is sent to the unified payment platform, the unified payment platform can establish a communication connection to the application used by the first user to scan the order QR code, and generate a payment redirect page corresponding to a payment platform based on the platform information of the payment platform of the application, so that the first user performs the payment operation based on the payment redirect page address information by using the application, for example, initiates a checkout process, and then performs an operation, for example, enters a payment confirmation password, in a pop-up interface.

[0122] In the one or more embodiments of this specification, different payment redirect pages can be generated based on different payment platforms, to implement payment of different payment platforms.

[0123] Certainly, the function of the unified payment platform can be alternatively implemented as a module of an execution body in the one or more embodiments of this application.

[0124] In the one or more embodiments of this specification, the method can
5 further include: receiving an acceptance message returned by the unified payment platform; and unlocking a data row, in a database, that the first order is located in.

[0125] In the one or more embodiments of this specification, the timing for
unlocking the data row can be set to being after acceptance of the unified platform. Due
to a limitation of the database, if a data row lock is occupied for long time, it is possible
10 that problems such as slow database running, redundant operation logs, and database crash occur. Therefore, locking time should not be too long, for example, should be 3 seconds or 50 milliseconds. To ensure the payment operation on the first order, the data row can be unlocked after the unified platform receives the payment code.

[0126] In the one or more embodiments of this specification, the method can
15 further include: updating a non-payable time point based on a predetermined payment time parameter after an acceptance message returned by a unified payment platform is received.

[0127] In the one or more embodiments of this specification, the non-payable time
point can be updated when the acceptance message returned by the unified payment
20 platform is received. The predetermined payment time parameter can be set with reference to the setting method of the predetermined payment time interval in the above-mentioned one or more embodiments. The predetermined payment time parameter can represent duration of a payment operation of a user for an order.

[0128] In the one or more embodiments of this specification, the method can
25 further include: setting the first order to a paid state after it is determined that the first user successfully performs payment.

[0129] In the one or more embodiments of this specification, alternatively, a
payment status of the unified payment platform can be asynchronously queried. If it is
determined that the second order is successfully paid for, a payment status of the second
30 order is modified to paid, and simultaneously the first order is set to the paid state, to complete payment of the first order. Certainly, alternatively, after payment is completed, the unified payment platform actively sends a payment result to the execution body, for example, a code platform, in this application, and then the execution body determines,

based on the received payment result, whether the first user successfully performs payment.

5 **[0130]** In the one or more embodiments of this specification, the method can further include: obtaining a corresponding cancelation time point that the first user cancels the payment operation at if it is determined that the first user cancels the payment operation; and setting the non-payable time point to the cancelation time point if it is determined that the cancelation time point is earlier than the non-payable time point.

10 **[0131]** In the one or more embodiments of this specification, if the first user cancels the payment operation, to enable another user to perform a payment operation on the first order as soon as possible, the non-payable time point can be updated based on the cancelation time point that the first user cancels the payment operation at, so that another user can pay for the first order as soon as possible after the first user fails to perform payment.

15 **[0132]** FIG. 7 is a lane diagram illustrating a payment method according to one or more embodiments of this specification.

20 **[0133]** Referring to FIG. 7, based on the payment method provided in the embodiments of this application, in actual use, the following process can be included: An execution body, for example, a code platform, in the embodiments of this specification receives, at 10:00:00, a payment instruction, of a first order, sent by a first user by scanning an order QR code of a target product; obtains a database row lock, and locks an order row; creates a second payment order corresponding to the first order; generates a payment code of the second order, and sends the payment code to a unified payment platform; receives, at 10:00:30, an acceptance notification returned by the unified payment platform, and adjusts a non-payable time point to 10:02; and unlocks the order row.

25 **[0134]** The unified payment platform returns a payment redirect page address based on an application of the first user, and the first user performs a payment operation based on a redirected interface.

30 **[0135]** After payment succeeds, the unified payment platform saves a payment result. After it is determined that the payment result is that payment succeeds, a payment status of the second order is modified to paid, and a payment status of the first order is modified to paid. The first user completes payment.

[0136] If a payment instruction sent by a third user by scanning the order QR code is received at 10:01, and the database row lock of the first order cannot be obtained, prompt information "being paid for" is fed back to the third user.

[0137] The third user sends a payment instruction at 10:03. If it is determined that the data row of the first order is not locked, and further determined that the first order is in a paid state, a prompt message "the order has been paid for" is returned to the third user.

[0138] It can be seen that in the one or more embodiments of this specification, duplicate payments can be prevented while the order QR code can be duplicately used.

[0139] Based on the same idea, the embodiments of this specification further provide an apparatus corresponding to the above-mentioned method. FIG. 8 is a schematic diagram illustrating a structure of a payment apparatus corresponding to FIG. 2 according to one or more embodiments of this specification.

[0140] As shown in FIG. 8, the payment apparatus can include: a receiving module 81, configured to receive a payment instruction sent by a first user by scanning an order QR code, where the order QR code is a QR code generated based on a first order corresponding to a target product, and the first order is generated based on a purchase operation performed by a second user for the target product; a setting module 82, configured to set the first order to an occupied state, where a payment instruction of a user other than the first user is rejected to be executed when the first order is in the occupied state; a creation module 83, configured to create a second order of the first user based on an order identifier of the first order in the occupied state; and a third-party payment module 84, configured to perform a payment operation based on the second order.

[0141] FIG. 9 is a schematic diagram illustrating a structure of an electronic device corresponding to FIG. 2 according to one or more embodiments of this specification. As shown in FIG. 8, an electronic device 900 can include at least one processor 910 and a memory 930 communicatively connected to the at least one processor. The memory 930 stores instructions 920 that can be executed by the at least one processor 910. The instructions are executed by the at least one processor 910, so that the at least one processor 910 can receive a payment instruction sent by a first user by scanning an order QR code, where the order QR code is generated based on a first order corresponding to a target product, and the first order is generated based on a purchase operation performed by a second user for the target product; set the first order to an

occupied state, where a payment instruction of a user other than the first user is rejected to be executed when the first order is in the occupied state; create a second order of the first user based on an order identifier of the first order in the occupied state; and perform a payment operation based on the second order.

5 **[0142]** The embodiments in this specification are described in a progressive way. For same or similar parts of the embodiments, references can be made to the embodiments mutually. Each embodiment focuses on a difference from other embodiments. Particularly, the payment apparatus shown in FIG. 7 is briefly described because of being basically similar to the method embodiments. For related parts,
10 references can be made to some descriptions in the method embodiments.

[0143] In the 1990s, whether a technical improvement is a hardware improvement (for example, an improvement to a circuit structure, such as a diode, a transistor, or a switch) or a software improvement (an improvement to a method process) can be clearly distinguished. However, as technologies develop, current improvements to
15 many method processes can be considered as direct improvements to hardware circuit structures. A designer usually programs an improved method process into a hardware circuit, to obtain a corresponding hardware circuit structure. Therefore, a method process can be improved by using a hardware entity module. For example, a programmable logic device (PLD) (for example, a field programmable gate array
20 (FPGA)) is such an integrated circuit, and a logical function of the PLD is determined by a user through device programming. The designer performs programming to "integrate" a digital system to a PLD without requesting a chip manufacturer to design and produce an application-specific integrated circuit chip. In addition, nowadays, instead of manually manufacturing an integrated circuit chip, this type of programming
25 is mostly implemented by using "logic compiler" software. The software is similar to a software compiler used to develop and write a program. Original code needs to be written in a particular programming language for compilation. The language is referred to as a hardware description language (HDL). There are many HDLs, such as the advanced Boolean expression language (ABEL), the Altera hardware description
30 language (AHDL), Confluence, the Cornell university programming language (CUPL), HDCal, the java hardware description language (JHDL), Lava, Lola, MyHDL, PALASM, and the ruby hardware description language (RHDL). The very-high-speed integrated circuit hardware description language (VHDL) and Verilog are most commonly used at present. A person skilled in the art should also understand that a

hardware circuit that implements a logical method process can be readily obtained once the method process is logically programmed by using some described hardware description languages and is programmed into an integrated circuit.

[0144] A controller can be implemented by using any appropriate method. For example, the controller can be a microprocessor or a processor, or a computer-readable medium that stores computer-readable program code (such as software or firmware) that can be executed by the microprocessor or the processor, a logic gate, a switch, an application-specific integrated circuit (ASIC), a programmable logic controller, or an embedded microprocessor. Examples of the controller include but are not limited to the following microprocessors: ARC 625D, Atmel AT91SAM, Microchip PIC18F26K20, and Silicone Labs C8051F320. The memory controller can also be implemented as a part of the control logic of the memory. A person skilled in the art also knows that, in addition to implementing the controller by using only the computer-readable program code, logic programming can be performed on method steps to enable the controller to implement the same function in forms of the logic gate, the switch, the application-specific integrated circuit, the programmable logic controller, the embedded microcontroller, etc. Therefore, the controller can be considered as a hardware component, and an apparatus included in the controller for implementing various functions can also be considered as a structure in the hardware component. Alternatively, the apparatus for implementing various functions can even be considered as both a software module implementing the method and a structure in the hardware component.

[0145] The system, apparatus, module, or unit illustrated in the above-mentioned embodiments can be specifically implemented by using a computer chip or an entity, or can be implemented by using a product having a certain function. A typical implementation device is a computer. Specifically, for example, the computer can be a personal computer, a laptop computer, a cellular phone, a camera phone, a smartphone, a personal digital assistant, a media player, a navigation device, an e-mail device, a game console, a tablet computer, a wearable device, or any combination of these devices.

[0146] For ease of description, when the above-mentioned apparatus is described, the apparatus is divided into various units based on functions for separate description. Certainly, when this application is implemented, the functions of the units can be implemented in one or more pieces of software and/or hardware.

[0147] A person skilled in the art should understand that the embodiments of this specification can be provided as methods, systems, or computer program products. Therefore, this application can use a form of hardware only embodiments, software only embodiments, or embodiments with a combination of software and hardware.

5 Moreover, this application can use a form of a computer program product that is implemented on one or more computer-usable storage media (including but not limited to a disk memory, a CD-ROM, an optical memory, etc.) that include computer-usable program code.

[0148] This specification is described with reference to the flowcharts and/or block diagrams of the method, the device (system), and the computer program product according to the embodiments of this specification. It should be understood that computer program instructions can be used to implement each process and/or each block in the flowcharts and/or the block diagrams and a combination of a process and/or a block in the flowcharts and/or the block diagrams. These computer program
10 instructions can be provided for a general-purpose computer, a dedicated computer, an embedded processor, or a processor of another programmable data processing device to generate a machine, so the instructions executed by the computer or the processor of the other programmable data processing device generate an apparatus for implementing a specific function in one or more processes in the flowcharts and/or in one or more
15 blocks in the block diagrams.

[0149] These computer program instructions can be stored in a computer-readable memory that can instruct the computer or the other programmable data processing device to work in a specific way, so the instructions stored in the computer-readable memory generate an artifact that includes an instruction apparatus. The instruction
20 apparatus implements a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

[0150] These computer program instructions can be loaded onto the computer or another programmable data processing device, so that a series of operations and steps are performed on the computer or the other programmable device, thereby generating
25 computer-implemented processing. Therefore, the instructions executed on the computer or the other programmable device provide steps for implementing a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

[0151] In a typical configuration, a computing device includes one or more central processing units (CPU), input/output interfaces, network interfaces, and memories.

[0152] The memory can include a non-persistent memory, a random access memory (RAM), a non-volatile memory, and/or another form in a computer-readable medium, for example, a read-only memory (ROM) or a flash memory (flash RAM). The memory is an example of the computer-readable medium.

[0153] The computer-readable medium includes persistent, non-persistent, movable, and unmovable media that can store information by using any method or technology. The information can be a computer-readable instruction, a data structure, a program module, or other data. Examples of the computer storage medium include but are not limited to a phase change random access memory (PRAM), a static random access memory (SRAM), a dynamic random access memory (DRAM), another type of random access memory (RAM), a read-only memory (ROM), an electrically erasable programmable read-only memory (EEPROM), a flash memory or another memory technology, a compact disc read-only memory (CD-ROM), a digital versatile disc (DVD) or another optical storage, a cassette magnetic tape, a magnetic tape/magnetic disk storage, another magnetic storage device, or any other non-transmission medium. The computer storage medium can be configured to store information that can be accessed by a computing device. Based on the definition in this specification, the computer-readable medium does not include transitory media such as a modulated data signal and carrier.

[0154] It should be further noted that the terms "include" and "comprise" or their any other variants are intended to cover a non-exclusive inclusion, so that a process, method, product or device that includes a list of elements not only includes those elements but also includes other elements that are not expressly listed, or further includes elements inherent to such process, method, product or device. Without more constraints, an element preceded by "includes a ..." does not preclude the existence of additional identical elements in the process, method, product, or device that includes the element.

[0155] A person skilled in the art should understand that the embodiments of this application can be provided as methods, systems, or computer program products. Therefore, this application can use a form of hardware only embodiments, software only embodiments, or embodiments with a combination of software and hardware. Moreover, this application can use a form of a computer program product that is

implemented on one or more computer-usable storage media (including but not limited to a disk memory, a CD-ROM, an optical memory, etc.) that include computer-usable program code.

5 **[0156]** This application can be described in the general context of computer-executable instructions executed by a computer, for example, a program module. Generally, the program module includes a routine, a program, an object, a component, a data structure, etc. executing a specific task or implementing a specific abstract data type. This application can be alternatively practiced in distributed computing environments. In these distributed computing environments, tasks are performed by
10 remote processing devices that are connected through a communications network. In the distributed computing environments, the program module can be located in local and remote computer storage media including storage devices.

15 **[0157]** The above-mentioned descriptions are merely embodiments of this application, and are not intended to limit this application. A person skilled in the art can make various changes and variations to this application. Any modification, equivalent replacement, improvement, etc. made without departing from the spirit and principle of this application shall fall within the scope of the claims in this application.

CLAIMS

What is claimed is:

1. A payment method, comprising:

receiving a payment instruction sent by a first user by scanning an order QR code,
5 wherein the order QR code is generated based on a first order corresponding to a target product, and the first order is generated based on a purchase operation performed by a second user for the target product;

setting the first order to an occupied state, wherein a payment instruction of a user other than the first user is rejected to be executed when the first order is in the occupied
10 state;

creating a second order of the first user based on an order identifier of the first order in the occupied state; and

performing a payment operation based on the second order.

2. The payment method according to claim 1, before the setting the first order to
15 an occupied state, further comprising:

determining whether the first order is in the occupied state; and

performing the step of setting the first order to the occupied state if it is determined that the first order is not in the occupied state.

3. The payment method according to claim 2, wherein the determining whether
20 the first order is in the occupied state specifically comprises:

determining whether a data row, in a database, that the first order is located in is locked; and

determining that the first order is not in the occupied state if it is determined that the data row is not locked.

25 4. The payment method according to claim 2, wherein the determining whether the first order is in the occupied state specifically comprises:

determining whether a current time point is earlier than a non-occupiable time point, wherein the non-occupiable time point is used to represent a time point that is set when a payment instruction is received at a previous time; and

30 determining that the first order is not in the occupied state if it is determined that the current time point is not earlier than the non-occupiable time point.

5. The payment method according to claim 1, wherein the setting the first order to an occupied state comprises:

locking a data row, in a database, that the first order is located in.

6. The payment method according to claim 1, wherein the setting the first order to an occupied state comprises:

5 setting a non-occupiable time point to a sum of a current time point and a predetermined non-occupiable time parameter, wherein the non-occupiable time point is used to represent a time point that a payment instruction of a user other than the first user can be executed at.

10 7. The payment method according to claim 2, wherein prompt information indicating that the first order is being processed is sent to the first user if it is determined that the first order is in the occupied state.

8. The payment method according to claim 3, wherein the determining whether the first order is in the occupied state further comprises:

15 determining whether the data row, in the database, that the first order is located in is locked after a predetermined waiting time interval if it is determined that the data row is locked; and

 determining that the first order is in the occupied state if the data row, in the database, that the first order is located in is locked after the predetermined waiting time interval.

20 9. The payment method according to any one of claims 1 to 8, before the creating a second order of the first user based on an order identifier of the first order in the occupied state, further comprising:

 determining whether the current time point is earlier than a non-payable time point; and

25 determining that the first order is not in a payment process if it is determined that the current time point is not earlier than the non-payable time point, wherein

 the operation of creating the second order of the first user based on the order identifier of the first order in the occupied state can be performed when the first order is not in the payment process.

10. The payment method according to claim 9, further comprising:

30 determining that the first order is in the payment process if it is determined that the current time point is earlier than the non-payable time point; and

 sending, to the first user, prompt information indicating that the first order is being paid for.

11. The payment method according to claim 1, before the creating a second order of the first user based on an order identifier of the first order in the occupied state, further comprising:

5 determining that the first order is in a payable state if it is determined that the first order has been paid for; or

determining that the first order is not in the payable state if it is determined that the first order is not paid for, wherein

the second order of the first user can be created based on the order identifier of the first order in the occupied state when the first order is in the payable state.

10 12. The payment method according to claim 1, before the creating a second order of the first user based on an order identifier of the first order in the occupied state, further comprising:

determining that the first order is in a payable state if it is determined that a transaction corresponding to the first order is in an unclosed state; or

15 determining that the first order is not in the payable state if it is determined that the transaction corresponding to the first order is in a closed state.

13. The payment method according to claim 11, further comprising:

sending, to the first user, a prompt message indicating that the first order has been paid for if it is determined that the first order is not in the payable state.

20 14. The payment method according to claim 12, further comprising:

sending, to the first user, a prompt message indicating that the first order cannot be paid for if it is determined that the first order is not in the payable state.

15. The payment method according to claim 1, wherein the performing a payment operation based on the second order specifically comprises:

25 sending a payment code of the second order to a unified payment platform, wherein the unified payment platform can generate payment redirect page address information based on the payment code of the second order and payment platform information of an application used by the first user to scan the order QR code, and send the payment redirect page address information to the application, so that the first user
30 performs the payment operation in a payment interface popped up by the application based on the payment redirect page address information.

16. The payment method according to claim 15, further comprising:

receiving an acceptance message returned by the unified payment platform; and unlocking a data row, in a database, that the first order is located in.

17. The payment method according to claim 9, further comprising:
updating the non-payable time point based on a predetermined payment time parameter after an acceptance message returned by a unified payment platform is received.

5 18. The payment method according to claim 1, further comprising:
 setting the first order to a paid state after it is determined that the first user successfully performs payment.

 19. The payment method according to claim 1, further comprising:
 obtaining a corresponding cancelation time point that the first user cancels the
10 payment operation at if it is determined that the first user cancels the payment operation;
 and

 setting the non-payable time point to the cancelation time point if it is determined that the cancelation time point is earlier than the non-payable time point.

 20. The payment method according to claim 1, wherein the order identifier of the
15 first order has a mapping relationship with an order identifier of the second order.

 21. A payment apparatus, comprising:

 a receiving module, configured to receive a payment instruction sent by a first user by scanning an order QR code, wherein the order QR code is a QR code generated based on a first order corresponding to a target product, and the first order is generated
20 based on a purchase operation performed by a second user for the target product;

 a setting module, configured to set the first order to an occupied state, wherein a payment instruction of a user other than the first user is rejected to be executed when the first order is in the occupied state;

 a creation module, configured to create a second order of the first user based on
25 an order identifier of the first order in the occupied state; and

 a third-party payment module, configured to perform a payment operation based on the second order.

 22. An electronic device, comprising:

 at least one processor; and

30 a memory communicatively connected to the at least one processor, wherein
 the memory stores instructions that can be executed by the at least one processor, and the instructions are executed by the at least one processor, so that the at least one processor can:

receive a payment instruction sent by a first user by scanning an order QR code, wherein the order QR code is a QR code generated based on a first order corresponding to a target product, and the first order is generated based on a purchase operation performed by a second user for the target product;

5 set the first order to an occupied state, wherein a payment instruction of a user other than the first user is rejected to be executed when the first order is in the occupied state;

 create a second order of the first user based on an order identifier of the first order in the occupied state; and

10 perform a payment operation based on the second order.

23. A computer-readable medium, storing computer-readable instructions, wherein the computer-readable instructions can be executed by a processor to implement the payment method according to any one of claims 1 to 20.