

SERVER AND METHOD FOR MULTI-MERCHANT ORDERS

TECHNICAL FIELD

[001] Various aspects of this disclosure relate to a server configured for multi-merchant orders. Various aspects of this disclosure relate to a method for multi-merchant orders. Various aspects of this disclosure relate to a non-transitory computer-readable medium storing computer executable code for multi-merchant orders. Various aspects of this disclosure relate to a computer executable code for multi-merchant orders.

BACKGROUND

[002] Current delivery systems and applications allow users to order items from a single merchant in a single order. If a user wishes to order items from other merchants, the user has to place a separate order for those items from the other merchants.

[003] This process is more time-consuming for a user who wants to order items from multiple merchants as multiple orders must be made even though the other merchants may be situated near each other, for e.g., in the same shopping area or in the same food court. This may also lead to higher costs for the user as multiple drivers will be assigned to these orders.

[004] Further, for the delivery provider, since multiple drivers will be assigned to these orders for the user, the cost to serve each user may be higher. On top of that, during peak hours, fulfillment rate for delivery service is lower than non-peak hours as the demand additional drivers is usually higher than the supply of available drivers. This may lead to a higher number of unsatisfied customers.

SUMMARY

[005] Therefore, there may be a need to provide a system and/or application which is able to allow users to effectively order from multiple merchants or stores in a single order. There may also be a need for the system and/or application to effectively and accurately select one or more merchants that the user can order from. There may also be a need to improve user experience, and/or decrease user costs, and/or improve profitability through higher basket size.

[006] Various embodiments may provide a server configured for multi-merchant orders is disclosed. The server may include one or more processor(s) and a memory having

instructions stored therein. The instructions when executed by the one or more processor(s), may cause the one or more processor(s) to receive in real time, a first merchant data from a user device of a user when the user device is displaying information from a first merchant. The one or more processor(s) may also determine at least one suitable merchant to display on the user device of the user using the first merchant data and historical data in a database. The one or more processor(s) may further display information of the at least one suitable merchant on the user device of the user. The one or more processor(s) may also receive a user order from the user, wherein the user order includes a first order data regarding a first item from the first merchant, and a second order data regarding a second item from a second merchant. The one or more processor(s) may further send the user order to a service provider device of a service provider for the service provider to pick up the first item from the first merchant, and the second item from the second merchant, and to deliver the first item and the second item to the user at a same time.

[007] According to various embodiments, the one or more processor(s) may be configured to determine the at least one suitable merchant based on at least one of: a location of the first merchant, a category of items sold by the first merchant, a preparation time of the first merchant, and user preferences.

[008] According to various embodiments, the one or more processor(s) may be configured to determine the at least one suitable merchant by: determining the location of the first merchant through the first merchant data or through the historical data in the database. The one or more processor(s) may be configured to search the historical data in the database for at least one potential merchant, wherein the at least one potential merchant has a location that is within a distance threshold from the location of the first merchant. The one or more processor(s) may also be configured to set the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[009] According to various embodiments, the one or more processor(s) may be configured to determine the at least one suitable merchant by: determining the location of the first merchant through the first merchant data or through the historical data in the database. The one or more processor(s) may be configured to search the historical data in the database for at least one potential merchant, wherein a time taken to travel between the at least one potential merchant and the first merchant is within a time threshold. The one or more processor(s) may also be configured to set the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[0010] According to various embodiments, the one or more processor(s) may be configured to determine the at least one suitable merchant by: determining the category of items sold by the first merchant through the first merchant data or through the historical data in the database. The one or more processor(s) may be configured to search the historical data in the database for at least one potential merchant that is selling a similar category of items as the category of items sold by the first merchant. The one or more processor(s) may also be configured to set the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[0011] According to various embodiments, the one or more processor(s) may be configured to determine the at least one suitable merchant by: determining the category of items sold by the first merchant through the first merchant data or through the historical data in the database. The one or more processor(s) may be configured to search the historical data in the database for at least one potential merchant that is selling a complementary category of items as the category of items sold by the first merchant. The one or more processor(s) may also be configured to set the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[0012] According to various embodiments, the one or more processor(s) may be configured to determine the at least one suitable merchant by: determining the preparation time of the first merchant through the first merchant data. The one or more processor(s) may be configured to search the historical data in the database for at least one potential merchant with a historical preparation time that is within a preparation time threshold compared to the preparation time of the first merchant. The one or more processor(s) may also be configured to send a preparation time information request to a device of the at least one potential merchant. The one or more processor(s) may further be configured to receive a preparation time of the at least one potential merchant from the device of the at least one potential merchant. The one or more processor(s) may also be configured to determine if the preparation time of the at least one potential merchant is within the preparation time threshold compared to the preparation time of the first merchant. The one or more processor(s) may further be configured to set the at least one potential merchant as the at least one suitable merchant to display on the user device of the user when the preparation time of the at least one potential merchant is within the preparation time threshold.

[0013] According to various embodiments, the one or more processor(s) may be configured to determine the at least one suitable merchant by: searching the historical data

in the database for the user preferences. The one or more processor(s) may be configured to search the historical data in the database for at least one potential merchant that matches the user preferences. The one or more processor(s) may be configured to set the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[0014] According to various embodiments, prior to receiving the user order from the user, the one or more processor(s) may be configured to receive from the user device of the user, information regarding a virtual cart of the user. The virtual cart may include information regarding potential items to be checked out. The one or more processor(s) may be configured to receive the user order from the user when the user checks out items from the virtual cart.

[0015] According to various embodiments, the one or more processor(s) may be configured to receive from a second user device of a second user, information regarding an item that the second user adds to the virtual cart using the second user device. The one or more processor(s) may also be configured to update the virtual cart displayed on the user device of the user.

[0016] Various embodiments may provide a method for multi-merchant orders. The method may include using one or more processor(s) of a server to receive in real time, a first merchant data from a user device of a user when the user device is displaying information from a first merchant. The one or more processor(s) may determine at least one suitable merchant to display on the user device of the user using the first merchant data and historical data in a database. The one or more processor(s) may display information of the at least one suitable merchant on the user device of the user. The one or more processor(s) may receive a user order from the user, wherein the user order includes a first order data regarding a first item from the first merchant, and a second order data regarding a second item from a second merchant. The one or more processor(s) may send the user order to a service provider device of a service provider for the service provider to pick up the first item from the first merchant, and the second item from the second merchant, and to deliver the first item and the second item to the user at a same time.

[0017] According to various embodiments, the method may include using the one or more processor(s) to: determine the at least one suitable merchant based on at least one of: a location of the first merchant, a category of items sold by the first merchant, a preparation time of the first merchant, and user preferences.

[0018] According to various embodiments, the method may include using the one or more processor(s) to determine the at least one suitable merchant by: determining the location of the first merchant through the first merchant data or through the historical data in the database. The method may include searching the historical data in the database for at least one potential merchant, wherein the at least one potential merchant has a location that is within a distance threshold from the location of the first merchant. The method may include setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[0019] According to various embodiments, the method may include using the one or more processor(s) to determine the at least one suitable merchant by: determining the location of the first merchant through the first merchant data or through the historical data in the database. The method may include searching the historical data in the database for at least one potential merchant, wherein a time taken to travel between the at least one potential merchant and the first merchant is within a time threshold. The method may include setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[0020] According to various embodiments, the method may include using the one or more processor(s) to determine the at least one suitable merchant by: determining the category of items sold by the first merchant through the first merchant data or through the historical data in the database. The method may include searching the historical data in the database for at least one potential merchant that is selling a similar category of items as the category of items sold by the first merchant. The method may include setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[0021] According to various embodiments, the method may include using the one or more processor(s) to determine the at least one suitable merchant by: determining the category of items sold by the first merchant through the first merchant data or through the historical data in the database. The method may include searching the historical data in the database for at least one potential merchant that is selling a complementary category of items as the category of items sold by the first merchant. The method may include setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[0022] According to various embodiments, the method may include using the one or more processor(s) to determine the at least one suitable merchant by: determining the preparation time of the first merchant through the first merchant data. The method may include searching the historical data in the database for at least one potential merchant with a historical preparation time that is within a preparation time threshold compared to the preparation time of the first merchant. The method may also include sending a preparation time information request to a device of the at least one potential merchant. The method may include receiving a preparation time of the at least one potential merchant from the device of the at least one potential merchant. The method may further include determining if the preparation time of the at least one potential merchant is within the preparation time threshold compared to the preparation time of the first merchant. The method may also include setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user when the preparation time of the at least one potential merchant is within the preparation time threshold.

[0023] According to various embodiments, the method may include using the one or more processor(s) to determine the at least one suitable merchant by: searching the historical data in the database for the user preferences. The method may also include searching the historical data in the database for at least one potential merchant that matches the user preferences. The method may further include setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[0024] According to various embodiments, the method may include prior to receiving the user order from the user, using the one or more processor(s) to: receive from the user device of the user, information regarding a virtual cart of the user, wherein the virtual cart includes information regarding potential items to be checked out. The method may also include receiving the user order from the user when the user checks out items from the virtual cart.

[0025] According to various embodiments, the method may include using the one or more processor(s) to: receive from a second user device of a second user, information regarding an item that the second user adds to the virtual cart using the second user device. The method may include updating the virtual cart displayed on the user device of the user.

[0026] Various embodiments may provide a non-transitory computer-readable medium storing computer executable code including instructions for multi-merchant orders according to the various embodiments disclosed herein.

[0027] Various embodiments may provide a computer executable code including instructions for multi-merchant orders according to the various embodiments disclosed herein.

[0028] To the accomplishment of the foregoing and related ends, the one or more embodiments include the features hereinafter fully described and particularly pointed out in the claims. The following description and the associated drawings set forth in detail certain illustrative features of the one or more aspects. These features are indicative, however, of but a few of the various ways in which the principles of various aspects may be employed, and this description is intended to include all such aspects and their equivalents.

BRIEF DESCRIPTION OF THE DRAWINGS

[0029] The invention will be better understood with reference to the detailed description when considered in conjunction with the non-limiting examples and the accompanying drawings, in which:

[0030] FIG. 1 shows a flowchart of a method for multi-merchant orders according to various embodiments.

[0031] FIG. 2 shows a schematic diagram of a communication system configured for multi-merchant orders according to various embodiments.

[0032] FIG. 3 shows a flow chart of a communication system configured for multi-merchant orders according to various embodiments.

[0033] FIG. 4 shows an exemplary area with potential suitable merchants within a distance threshold from a first merchant, according to various embodiments.

[0034] FIG. 5 shows a chart depicting exemplary driver waiting time at different merchants, according to various embodiments.

[0035] FIG. 6 shows a backend system of a communication system for multi-merchant order, according to various embodiments.

[0036] FIGS. 7A to 7N show flowcharts illustrating different backend processes when users use an application configured for multi-merchant orders, according to various embodiments.

[0037] FIG. 8 shows a flowchart of various order states, according to various embodiments.

[0038] FIGS. 9A to 9B show flowcharts illustrating different backend processes, according to various embodiments.

[0039] FIGS. 10A to 10F show flowcharts illustrating different backend processes, according to various embodiments.

DETAILED DESCRIPTION

[0040] The following detailed description refers to the accompanying drawings that show, by way of illustration, specific details and embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention. Other embodiments may be utilized and structural, and logical changes may be made without departing from the scope of the invention. The various embodiments are not necessarily mutually exclusive, as some embodiments can be combined with one or more other embodiments to form new embodiments.

[0041] Embodiments described in the context of one of the systems or server or methods or computer program are analogously valid for the other systems or server or methods or computer program and vice-versa.

[0042] Features that are described in the context of an embodiment may correspondingly be applicable to the same or similar features in the other embodiments. Features that are described in the context of an embodiment may correspondingly be applicable to the other embodiments, even if not explicitly described in these other embodiments. Furthermore, additions and/or combinations and/or alternatives as described for a feature in the context of an embodiment may correspondingly be applicable to the same or similar feature in the other embodiments.

[0043] The word “exemplary” is used herein to mean “serving as an example, instance, or illustration.” Any embodiment or design described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other embodiments or designs

[0044] In the context of various embodiments, the articles “a”, “an”, and “the” as used with regard to a feature or element include a reference to one or more of the features or elements.

[0045] As used herein, the term “and/or” includes any and all combinations of one or more of the associated listed items.

[0046] The terms “at least one” and “one or more” may be understood to include a numerical quantity greater than or equal to one (e.g., one, two, three, four, [...], etc.). The term “a plurality” may be understood to include a numerical quantity greater than or equal to two (e.g., two, three, four, five, [...], etc.).

[0047] The words “plural” and “multiple” in the description and the claims expressly refer to a quantity greater than one. Accordingly, any phrases explicitly invoking the aforementioned words (e.g. “a plurality of [objects]”, “multiple [objects]”) referring to a quantity of objects expressly refers more than one of the said objects. The terms “group (of)”, “set [of]”, “collection (of)”, “series (of)”, “sequence (of)”, “grouping (of)”, etc., and the like in the description and in the claims, if any, refer to a quantity equal to or greater than one, i.e. one or more. The terms “proper subset”, “reduced subset”, and “lesser subset” refer to a subset of a set that is not equal to the set, i.e. a subset of a set that contains less elements than the set.

[0048] The term “data” as used herein may be understood to include information in any suitable analog or digital form, e.g., provided as a file, a portion of a file, a set of files, a signal or stream, a portion of a signal or stream, a set of signals or streams, and the like. Further, the term “data” may also be used to mean a reference to information, e.g., in form of a pointer. The term data, however, is not limited to the aforementioned examples and may take various forms and represent any information as understood in the art.

[0049] The term “processor” or “controller” as, for example, used herein may be understood as any kind of entity that allows handling data, signals, etc. The data, signals, etc. may be handled according to one or more specific functions executed by the processor or controller.

[0050] A processor or a controller may thus be or include an analog circuit, digital circuit, mixed-signal circuit, logic circuit, processor, microprocessor, Central Processing Unit (CPU), Graphics Processing Unit (GPU), Digital Signal Processor (DSP), Field Programmable Gate Array (FPGA), integrated circuit, Application Specific Integrated Circuit (ASIC), etc., or any combination thereof. Any other kind of implementation of the respective functions, which will be described below in further detail, may also be understood as a processor, controller, or logic circuit. It is understood that any two (or more) of the processors, controllers, or logic circuits detailed herein may be realized as a single entity with equivalent functionality or the like, and conversely that any single processor, controller, or logic circuit detailed herein may be realized as two (or more) separate entities with equivalent functionality or the like.

[0051] The term “system” (e.g., a drive system, a position detection system, etc.) detailed herein may be understood as a set of interacting elements, the elements may be, by way of example and not of limitation, one or more mechanical components, one or more electrical

components, one or more instructions (e.g., encoded in storage media), one or more controllers, etc.

[0052] A “circuit” as user herein is understood as any kind of logic-implementing entity, which may include special-purpose hardware or a processor executing software. A circuit may thus be an analog circuit, digital circuit, mixed-signal circuit, logic circuit, processor, microprocessor, Central Processing Unit (“CPU”), Graphics Processing Unit (“GPU”), Digital Signal Processor (“DSP”), Field Programmable Gate Array (“FPGA”), integrated circuit, Application Specific Integrated Circuit (“ASIC”), etc., or any combination thereof. Any other kind of implementation of the respective functions which will be described below in further detail may also be understood as a “circuit.” It is understood that any two (or more) of the circuits detailed herein may be realized as a single circuit with substantially equivalent functionality, and conversely that any single circuit detailed herein may be realized as two (or more) separate circuits with substantially equivalent functionality. Additionally, references to a “circuit” may refer to two or more circuits that collectively form a single circuit.

[0053] As used herein, “memory” may be understood as a non-transitory computer-readable medium in which data or information can be stored for retrieval. References to “memory” included herein may thus be understood as referring to volatile or non-volatile memory, including random access memory (“RAM”), read-only memory (“ROM”), flash memory, solid-state storage, magnetic tape, hard disk drive, optical drive, etc., or any combination thereof. Furthermore, it is appreciated that registers, shift registers, processor registers, data buffers, etc., are also embraced herein by the term memory. It is appreciated that a single component referred to as “memory” or “a memory” may be composed of more than one different type of memory, and thus may refer to a collective component including one or more types of memory. It is readily understood that any single memory component may be separated into multiple collectively equivalent memory components, and vice versa. Furthermore, while memory may be depicted as separate from one or more other components (such as in the drawings), it is understood that memory may be integrated within another component, such as on a common integrated chip.

[0054] FIG. 1 shows a flowchart of a method for multi-merchant orders according to various embodiments.

[0055] According to various embodiments, the method 100 of determining an advanced booking fee for an advance booking may be provided. In some embodiments, the method

100 may include a step 102 of using one or more processor(s) of a server to receive in real time, a first merchant data from a user device of a user when the user device is displaying information from a first merchant. The method 100 may include a step 104 of using the one or more processor(s) to determine at least one suitable merchant to display on the user device of the user using the first merchant data and historical data in a database. The method 100 may include a step 106 of using the one or more processor(s) to display information of the at least one suitable merchant on the user device of the user. The method 100 may include a step 108 of using the one or more processor(s) to receive a user order from the user, wherein the user order includes a first order data regarding a first item from the first merchant, and a second order data regarding a second item from a second merchant. The method 100 may include a step 110 of using the one or more processor(s) to send the user order to a service provider device of a service provider for the service provider to pick up the first item from the first merchant, and the second item from the second merchant, and to deliver the first item and the second item to the user at a same time.

[0056] Steps 102 to 110 are shown in a specific order, however other arrangements are possible. Steps may also be combined in some cases. Any suitable order of steps 102 to 110 may be used.

[0057] FIG. 2 shows a schematic diagram of a communication system configured for multi-merchant orders according to various embodiments.

[0058] According to various embodiments, the communication system 200 may include a server 210, and/or a user device 220, and/or a service provider device 240, and/or a merchant device 250.

[0059] In some embodiments, the server 210 and the user device 220 may be in communication with each other through communication network 230. The server 210 and the service provider device 240 may also be in communication with each other through communication network 230. The server 210 and the merchant device 250 may also be in communication with each other through communication network 230. Even though FIG. 2 shows lines connecting the server 210, the user device 220, the service provider device 240, and the merchant device 250 to the communication network 230, in some embodiments, the server 210, the user device 220, the service provider device 240, and the merchant device 250 may not be physically connected to each other, for example through a cable. Instead, the server 210, the user device 220, the service provider device 240, and the merchant device

250 may be able to communicate wirelessly through communication network 230 by internet communication protocols or through a mobile cellular communication network.

[0060] In various embodiments, the server 210 may be a single server as illustrated schematically in FIG. 2, or have the functionality performed by the server 210 distributed across multiple server components. The server 210 may include one or more server processor(s) 212. The various functions performed by the server 210 may be carried out by the one or more server processor(s) 212. In some embodiments, the various functions performed by the server 210 may be carried out across the one or more server processor(s). In other embodiments, each specific function of the various functions performed by the server 210 may be carried out by specific server processor(s) of the one or more server processor(s).

[0061] In some embodiments, the server 210 may include a memory 214. The server 210 may also include a database. The database may be in the memory 214. The memory 214 and the database may be one component or may be separate components. The memory 214 of the server may include computer executable code defining the functionality that the server 210 carries out under control of the one or more server processor 212. The database and/or memory 214 may include historical data of a plurality of merchants, e.g., locations of the plurality of merchants and/or category of items sold by the plurality of merchants, and/or preparation time of the plurality of merchants, and/or user preferences of a plurality of users. The memory 214 may include or may be a computer program product such as a non-transitory computer-readable medium.

[0062] According to various embodiments, a computer program product may store the computer executable code including instructions for multi-merchant orders according to the various embodiments. The computer executable code may be a computer program. The computer program product may be a non-transitory computer-readable medium. The computer program product may be in the communication system 100 and/or the server 210.

[0063] In some embodiments, the server 210 may also include an input and/or output module allowing the server 210 to communicate over the communication network 230. The server 210 may also include a user interface for user control of the server 210. The user interface may include, for example, computing peripheral devices such as display monitors, user input devices, for example, touchscreen devices and computer keyboards.

[0064] In various embodiments, the user device 220 may include a user device memory 222 and a user device processor 224. The user device memory 222 may include computer

executable code defining the functionality the user device 220 carries out under control of the user device processor 224. The user device memory 222 may include or may be a computer program product such as a non-transitory computer-readable medium. The user device 220 may also include an input and/or output module allowing the user device 220 to communicate over the communication network 230. The user device 220 may also include a user interface for the user to control the user device 220. The user interface may be a touch panel display. The user interface may include a display monitor, a keyboard or buttons.

[0065] In various embodiments, the service provider device 240 may include a service provider device memory 242, and a service provider device processor 244. The service provider device memory 242 may include computer executable code defining the functionality the service provider device 240 carries out under control of the service provider device processor 244. The service provider device memory 242 may include or may be a computer program product such as a non-transitory computer-readable medium. The service provider device 240 may also include an input and/or output module allowing the service provider device 240 to communicate over the communication network 230. The service provider device 240 may also include a user interface for the user to control the service provider device 240. The user interface may be a touch panel display. The user interface may include a display monitor, a keyboard or buttons.

[0066] In various embodiments, the merchant device 250 may include a merchant device memory 252, and a merchant device processor 254. The merchant device memory 252 may include computer executable code defining the functionality the merchant device 250 carries out under control of the merchant device processor 254. The merchant device memory 252 may include or may be a computer program product such as a non-transitory computer-readable medium. The merchant device 250 may also include an input and/or output module allowing the merchant device 250 to communicate over the communication network 230. The merchant device 250 may also include a user interface for the merchant to control the merchant device 250. The user interface may be a touch panel display. The user interface may include a display monitor, a keyboard or buttons.

[0067] In various embodiments, the communication system 200 may include a plurality of user devices, and/or a plurality of service provider devices, and/or a plurality of merchant devices. For the sake of brevity, duplicate descriptions of features and properties are omitted. It will be understood that any features and property described herein for the user device 220, and/or the service provider device 240, and/or the merchant device 250 may apply to the

plurality of user devices, and/or the plurality of service provider devices, and/or the plurality of merchant devices.

[0068] In various embodiments, the server 210 may be configured for multi-merchant orders. In some embodiments, a user may order items from multiple merchants through a user order, for example, the user order may contain a first order data regarding a first item from a first merchant, and a second order data regarding a second item from a second merchant. In some embodiments, the user may use the user device 220 to order items from multiple merchants. In some embodiments, the server 210 may receive the user order over the communication network 230 from the user device 220.

[0069] In various embodiments, a service provider, e.g., a delivery driver, may pick up the user order which may contain items from multiple merchants from the multiple merchants. For example, the service provider may pick up the first item from the first merchant, and the second item from the second merchant. In some embodiments, the service provider may deliver the user order which may contain items from multiple merchants to the user at the same time. In some embodiments, the service provider may use the service provider device 240 to receive the user order from the server 210 through the communication network 230.

[0070] In various embodiments, after receiving the user order, the server 210 may relay the user order to the merchants. In some embodiments, the user order may include a first order data regarding a first item from a first merchant, and a second order data regarding a second item from a second merchant. In some embodiments, the server 210 may send the first order data regarding the first item to the first merchant through a first merchant device. The server 210 may also send the second order data regarding the second item to the second merchant through a second merchant device. In some embodiments, the first merchant and/or the second merchant may prepare the first item and/or the second item on the user order. The first merchant and/or the second merchant may send a preparation time to the server 210. In some embodiments, the server may send the preparation time of the first merchant and/or the second merchant to the service provider through the service provider device 240. In some embodiments, the server may send the preparation time of the first merchant and/or the second merchant to the user through the user device 220.

[0071] FIG. 3 shows a flow chart 300 of a communication system configured for multi-merchant orders according to various embodiments.

[0072] In various embodiments, the communication system may include a server. The server may include one or more processor(s). The one or more processor(s) may receive in real time, a first merchant data 302 from a user device of a user when the user device is displaying information from a first merchant. In some embodiments, the first merchant may be a primary merchant. In some embodiments, the first merchant data 302 may include a merchant identification (ID). The first merchant data 302 may also include a location of the first merchant, and/or a category of items sold by the first merchant, and/or a preparation time of the first merchant, and/or a capacity of the first merchant. In another embodiment, the location of the first merchant, and/or the category of items sold by the first merchant, and/or the preparation time of the first merchant, and/or the capacity of the first merchant may be stored in a database in the communication system. The sever may access the database to retrieve the location of the first merchant, and/or the category of items sold by the first merchant, and/or the preparation time of the first merchant, and/or the capacity of the first merchant after receiving the first merchant data from the user device.

[0073] In some embodiments, the one or more processor(s) may determine from the location of the first merchant, a list of nearby merchants. The nearby merchants may be potential merchants that the server may display on the user device of the user. In some embodiments, the one or more processor(s) may determine from the location of the first merchant, a map 304 of nearby merchants. The map 304 may cover areas within a distance threshold from the location of the first merchant. The distance threshold may be between 30 metres (m) to 100 m, for e.g., 50 m, from the location of the first merchant. In some embodiments, merchants with locations that are within the distance threshold from the location of the first merchant may be in the map 304, and may be included in the list of nearby merchants. The list of nearby merchants may be sent to a merchant recommendation engine 306 in the server.

[0074] In some embodiments, the map 304 may cover areas within a time threshold from the location of the first merchant. In other words, a time taken to travel between the potential merchant and the first merchant is within the time threshold. The time threshold may be any time below 5 minutes, for e.g., 3 minutes, from the location of the first merchant.

[0075] In some embodiments, merchants with locations that are within the time threshold from the location of the first merchant may be in the map 304, and may be included in the list of nearby merchants. The list of nearby merchants may be sent to a merchant recommendation engine 306 in the server.

[0076] In various embodiments, the server may retrieve from the database (i.e., a data lake 308), historical data such as historical booking patterns. The historical data may also include merchants with locations that are within a distance threshold from the location of the first merchant, and/or merchants with locations that are within a time threshold from the location of the first merchant, and/or merchants with a similar category of items as the category of items sold by the first merchant, and/or merchants with a complementary category of items as the category of items sold by the first merchant, and/or merchants with historical preparation times that are within a preparation time threshold compared to the preparation time of the first merchant, and/or user preferences.

[0077] In various embodiments, the server 306 may provide smart recommendations of merchants to the user by displaying at least one merchant in addition to the first merchant that the user can order from in a single order. The merchant recommendation engine 306 of the server may determine at least one suitable merchant to display on the user device of the user using the first merchant data and the historical data in the database.

[0078] In various embodiments, the one or more processor(s) of the server may be configured to determine at least one suitable merchant based on at least one of: a location of the first merchant, a category of items sold by the first merchant, a preparation time of the first merchant, and user preferences.

[0079] In various embodiments, the one or more processor(s) of the server may be configured to determine at least one suitable merchant by determining the location of the first merchant through the first merchant data or through the historical data in the database. The server may search the historical data in the database for at least one potential merchant, wherein the at least one potential merchant has a location that is within a distance threshold from the location of the first merchant. The server may set the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[0080] In various embodiments, the one or more processor(s) of the server may be configured to determine at least one suitable merchant by determining the location of the first merchant through the first merchant data or through the historical data in the database. The server may search the historical data in the database for at least one potential merchant, wherein a time taken to travel between the at least one potential merchant and the first merchant is within a time threshold. The server may set the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[0081] In various embodiments, the one or more processor(s) of the server may be configured to determine the category of items sold by the first merchant through the first merchant data or through the historical data in the database. The server may search the historical data in the database for at least one potential merchant that is selling a similar category of items as the category of items sold by the first merchant. The server may set the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[0082] In various embodiments, the one or more processor(s) of the server may be configured to determine the category of items sold by the first merchant through the first merchant data or through the historical data in the database. The server may search the historical data in the database for at least one potential merchant that is selling a complementary category of items as the category of items sold by the first merchant. The server may set the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[0083] In various embodiments, the one or more processor(s) of the server may be configured to search the historical data in the database for the user preferences. The server may search the historical data in the database for at least one potential merchant that matches the user preferences. The server may set the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

[0084] In various embodiments, the one or more processor(s) of the server may be configured to determine the preparation time of the first merchant through the first merchant data. The server may search the historical data in the database for at least one potential merchant with a historical preparation time that is within a preparation time threshold compared to the preparation time of the first merchant. The server may send a preparation time information request to a device of the at least one potential merchant. The server may receive a preparation time of the at least one potential merchant from the device of the at least one potential merchant. The server may determine if the preparation time of the at least one potential merchant is within the preparation time threshold compared to the preparation time of the first merchant. The server may set the at least one potential merchant as the at least one suitable merchant to display on the user device of the user when the preparation time of the at least one potential merchant is within the preparation time threshold.

[0085] In various embodiment, a recommendation model 310 may be built based on the historical data in the database. In some embodiments, a feature used in the recommendation

model 310 may be merchants with a similar category of items as the category of items sold by the first merchant. For example, if the first merchant sells fast food, the server may recommend another merchant that also sells fast food. In some embodiments, a feature used in the recommendation model 310 may be merchants with a complementary category of items as the category of items sold by the first merchant. Data of the merchants with a complementary category of items may be obtained using historical order patterns. For example, if the first merchant sells food, the server may recommend another merchant that sells beverages, if the server determines through historical order patterns the category of “beverages” complements the category of “food”. In some embodiments, a feature used in the recommendation model 310 may be user preferences. The user preferences may include a category of item which the user prefers, for e.g., a cuisine type. The user preferences may also include a maximum waiting time the user will wait for an order. In some embodiments, a feature used in the recommendation model 310 may be merchant capacity, and/or merchant preparation time, and/or merchant basket size.

[0086] In various embodiments, the recommendation model 310 may be machine learning models at the user order level to predict how likely an order is made in another merchant when an order is made in the primary (first) merchant within a time window. The machine learning models may predict which other merchant the user would be more likely to book when the first merchant is selected. Modeling choices may vary from logistic regression, RandomForest classification or ensemble models (such xgboost).

[0087] In various embodiments, the recommendation model 310 may be built using a recommendation system to link merchants through user-level bookings, such as Non-negative Matrix Factorization. The recommendation model 310 may predict the similarities between merchant according to the input features provided, e.g., cuisine type. In an embodiment, when the recommendation model 310 receives a signal of a user checking the first merchant, other related merchants may be recommended and may be ranked according to the predicted similarity score.

[0088] In various embodiments, the type of model selections may be based on merchant distributions (e.g., the density of merchants in an area) at the city level.

[0089] In various embodiments, the data from recommendation model 310, may be sent to a data processing software 312, e.g., Spark software, for processing the data and updating the merchant recommendation engine 306. The merchant recommendation engine 306 may be updated at regular intervals, e.g., every 5 minutes.

[0090] In various embodiments, the server may display n most relevant merchants on the user device in a user application (app) 314. The number n may be any suitable integer, for e.g., 3 merchants. In some embodiments, the recommendation model 310 may determine a list of suitable merchants based on the first merchant data and the historical data in the database. The server may display only n most relevant merchants on the user device even though the list of suitable merchants may be more than n . If the list of suitable merchants is equal or less than n , the server will display all the merchants in the list of suitable merchants.

[0091] In an exemplary embodiment, the server may determine that merchant 1 is 51.9% suitable, merchant 2 is 25.3% suitable, merchant 3 is 9.2% suitable, and merchant 4 is 4.8% suitable. If n is set to 3 merchants, the server will display merchant 1, merchant 2 and merchant 3 on the user device, leaving out merchant 4. On the other hand, if n is set to 5, the server will display merchant 1, merchant 2, merchant 3, and merchant 4 on the user device.

[0092] In various embodiments, the server may receive a user order 316 from the user. The user order may include a first order data regarding a first item from the first merchant, and a second order data regarding a second item from a second merchant. The second merchant may be a merchant that the server displayed on the user device. In some embodiments, the server may store the user order in the database 308 as historical data.

[0093] In various embodiments, prior to receiving the user order from the user, the server may be configured to receive from the user device of the user, information regarding a virtual cart of the user. In some embodiments, the virtual cart may include information regarding potential items to be checked out. In some embodiments, the server may be configured to receive the user order from the user when the user checks out items from the virtual cart.

[0094] In various embodiments, the server may be configured to receive from a second user device of a second user, information regarding an item that the second user adds to the virtual cart using the second user device. The server may be configured to update the virtual cart displayed on the user device of the user. In some embodiments, when the user checks out items from the virtual cart, the virtual cart may include items added by the second user.

[0095] In various embodiments, the server may send the user order to a service provider device of a service provider through a driver app 318. The server may send instructions to the service provider to pick up the first item from the first merchant, and the second item from the second merchant. The server may also send instructions to the service provider to deliver the first item and the second item to the user at a same time.

[0096] In various embodiments, a merchant travel order engine 328 of the server may send merchant preparation information such as the preparation time of the first merchant and/or the second merchant to the service provider through the driver app 318. The service provider may use the merchant preparation information to decide which merchant to pick up the items from first.

[0097] In various embodiments, a travel history of the service provider such as the service provider travel time and/or time spent waiting for merchants may be sent to a database (i.e., data lake 322) to be stored as historical data. The historical data may be sent to a driver travel model 324, which may be a machine learning model. In some embodiments, the driver travel model may use the historical data to determine a service provider for the user order.

[0098] In various embodiments, the data from the driver travel model 324, may be sent to a data processing software 326, e.g., Spark software, for processing the data and updating the merchant travel order engine 328. The merchant travel order engine 328 may be updated at regular intervals, e.g., every 5 minutes.

[0099] FIG. 4 shows an exemplary area 400 with potential suitable merchants within a distance threshold from a first merchant, according to various embodiments.

[00100] As shown in FIG. 4, a first merchant 402 has a first location in the area 400. A boundary 404 depicting a first distance threshold is shown. The first distance threshold may be any suitable distance, for e.g., within 50 m from the first merchant 402. In some embodiments, there may be at least one potential suitable merchant 406 that is within the distance threshold from the first merchant is shown. In some embodiments, there may be a plurality of potential suitable merchants 406. In some embodiments, the server may determine at least one potential suitable merchant 406 to display on the user device.

[00101] FIG. 5 shows a chart 500 depicting exemplary driver waiting time at different merchants, according to various embodiments.

[00102] As shown in chart 500, different merchants may have different driver waiting times. In some embodiments, the server may indicate the sequence of item collections for the service provider. This may have an advantage of shortening the service provider waiting time for multi-merchant orders.

[00103] In various embodiments, a model (e.g., a machine learning model) may be built to predict waiting time based on the historical data of merchant-level features. The historical data of merchant-level features may include merchant cuisine types, basket sizes, time. In

some embodiments, the server may instruct the service provider to collect the item from the merchant with the shortest predicted waiting time first, and collect the item from the merchant with the longest predicted waiting time last. For example, if a user orders from merchant A and merchant B, since the waiting time for merchant A is shorter than merchant B, the server may instruct the user to collect the item from merchant A first before collecting the item from merchant B. As another example, if a user orders from merchant B and merchant C, since the waiting time for merchant C is shorter than merchant B, the server may instruct the user to collect the item from merchant C first before collecting the item from merchant B.

[00104] FIG. 6 shows a backend system 600 of a communication system for multi-merchant order, according to various embodiments.

[00105] The steps shown in the flowchart of FIG. 6 is shown in a specific order, however other arrangements are possible. Steps may also be combined in some cases. Any suitable order of steps may be used.

[00106] The backend system 600 illustrates the overall architecture of various backend systems working together to deliver a multi merchant group order created by a user through an application. In various embodiments, users, drivers, and merchants may get notified about the order on their applications through a push notification.

[00107] As shown in FIG. 6, there may be an Eater application 602 (i.e., the eater app) configured for multi-merchant orders. There may also be a customer experience (CE) portal 604. The CE portal 604 may view the order status for an ongoing order and may perform certain actions e.g., cancel an order, refund an amount to the user, and identify the driver assigned for a specific order. There may be a merchant application 606 (i.e., the merchant app) configured for multi-merchant orders. There may be a driver application 608 (i.e., the driver app) configured for multi-merchant orders. In various embodiments, the eater app 602, the merchant app 606, and/or the driver app 608 may be a same or different application on the respective user device of each of the eater (i.e., the user), the merchant and the driver. In various embodiment, the backend system may include a business portal 610 to assist in business matters, e.g., configuring delivery fee multipliers, and merchant commission ratio.

[00108] As shown in FIG. 6, the backend system may include an application programming interface (API) gateway 612. The API gateway 612 may be an API management tool that is between a client and a collection of backend services. The API gateway 612 may act as a common security layer to protect all the backend applications from cyber security threats.

[00109] As shown in FIG. 6, the backend system may include a PAX API 614, a CE API 624, a merchant API 628, a driver API 640 and a business API 644. In various embodiments, the PAX API 614, the CE API 624, the merchant API 628, the driver API 640 and/or the business API 644 may communicate with the eater app 602, the CE portal 604, the merchant app 606, the driver app 608, and/or the business portal 610 through the API gateway 612. In various embodiments, the PAX API 614, the CE API 624, the merchant API 628, the driver API 640 and/or the business API 644 may communicate with other backend applications to fulfil the requests.

[00110] In various embodiments, the eater app 602 may communicate with the PAX API 614 through the API gateway 612. The PAX API 614 may call a recommendation system 616 to show user specific merchants. The user specific merchants may be based on user preference, e.g., user preferred cuisines previously stored in the database of the server. The user specific merchants may also include other merchants chosen based on e.g., distance or any other possible choosing method. In various embodiments, the recommendation system 616 may call on a database to load user preferred cuisines previously stored in the database. The recommendation system may load merchants based on cuisine by calling a food dataservice module 612 in the backend system 600.

[00111] In various embodiments, the PAX API 614 may call a cart manager module 620 to manage the items chosen by the user. In various embodiments, when the user selects the cart on the eater app 602, the cart manager 620 may show the items chosen by the user in the cart. In various embodiments, the CE portal 604 may call the CE API 624 to assist in with the real time update of an order manager 622.

[00112] In various embodiments, when the user checks out the items in the cart, the order manager 622 may call on a preparation API 626. The preparation API 626 may pass the order to a merchant API 628. The merchant API 628 may pass the order to the merchant app 606 through the API gateway 612. The merchant may receive and may accept the order from the user. The order from the user may include the items ordered by the user, the details of a delivery person (i.e., a drive) and/or a time or date that the user to prepare the items for the order.

[00113] In various embodiments, when the user checks out the items in the cart, the order manager 622 may call on a cashier module 630. The cashier module 630 may call on a payment gateway 632. The payment gateway 632 may include various payment methods such as credit cards or any other electronic payment methods.

[00114] In various embodiments, when the user checks out the items in the cart, the order manager 622 may call on a delivery API 634. The delivery API 634 may call on a driver allocation module 638. In various embodiments, the driver app 608 may communicate with a driver API 640 through the API gateway 612. The driver API 640 may communicate with a driver capability module 640. The driver capability module may assess whether the driver is a suitable fit based on various requirements, e.g., whether the distance of the driver from the merchant and/or whether the driver is currently fulfilling another order. In various embodiments, if the driver capability module 642 assesses that the driver is suitable, the driver capability module 642 may call on the driver allocation module 638. In various embodiments, the driver allocation module 638 may allocate the driver to the order. In various embodiments, the delivery API 634 may call on an order batching module 636, the order batching module 636 may determine whether there are any orders with similar pickup locations and/or dropoff locations as the order made by the user. The order batching module 636 may include one or more other orders with similar pickup locations and/or dropoff locations with the order from the user and may call on the driver allocation module 638 to assign multiple orders to the same driver.

[00115] FIGS. 7A to 7N show flowcharts illustrating different backend processes when users use an application configured for multi-merchant orders, according to various embodiments.

[00116] The steps shown in the flowcharts of FIGS. 7A to 7N are shown in a specific order, however other arrangements are possible. Steps may also be combined in some cases. Any suitable order of steps may be used.

[00117] A flowchart 700 of FIG. 7A shows a backend process when a user enters the application (i.e., the Eater App) configured for multi-merchant orders. At step 701, the application may call a user or passenger (PAX) application programming interface (API) when opening the application to load a home page. At step 702, the PAX API may call a recommendation system to show user specific merchants. The user specific merchants may be based on user preference, e.g., user preferred cuisines previously stored in the database of the server. At step 703, the recommendation system may call on the database to load user preferred cuisines previously stored in the database. At step 703, the recommendation system may load merchants based on cuisine by calling a food dataservice module in the server.

[00118] A flowchart 705 of FIG. 7B shows a backend process when a user clicks on a first merchant in the application. At step 706, the application may call the PAX API. At step 707, the PAX API may call the recommendation system to get the nearby merchants allowed for multi-merchant ordering. The details for determining suitable merchants for multi-merchant ordering can be found in FIGS. 1 to 5 and the description regarding FIG. 1 to 5. At step 707, the PAX API may call the food dataservice module to load information regarding the suitable merchants which have been selected to be displayed on the user device. The information may include merchant details with menu items.

[00119] A flowchart 709 of FIG. 7C shows a backend process when a user creates a group and shares a link of the group. At step 710, the application may call the PAX API to create a group order. At step 711, the PAX API may check if a group already exists for the user. The PAX API may create the group if no group with any merchant exists already for that user. If there is an existing group, the PAX API may return an error to cancel the existing group. After the PAX API creates the group, the PAX API may return a unique sharing link for this group to the user. The user may share the link to with other users via any suitable platform such as other social platforms like messaging platforms like whatsapp, wechat etc. The other user who join the group may be considered as members of the group.

[00120] A flowchart 712 of FIG. 7D shows the different group states when a user creates a group. At step 713, the user may create the group. A state 714 of the group is now “created”. At step 715, all members in the group may proceed to add items to a virtual cart, and may proceed to confirm the added items in the virtual cart. A state 716 of the group is now “confirmed”. At step 717, the user which may be the group creator, may place the order. The order may include the items that the other members in the group added in the virtual cart. After the user places the order, the state 718 of the group is now “completed”. Alternatively, at the “confirmed” state 716, at step 719, the user may cancel the group. A state 720 of the group is now “cancelled”. Alternatively, at the “created” state 714, at step 721, the user may cancel the group. A state 720 of the group is now “cancelled”.

[00121] A flowchart 721 of FIG. 7E shows the member states in the group. At step 722, a member may join the group. A state 723 of the member is now “joined”. At step 724, the member in the group may proceed to add items to a virtual cart, and may proceed to confirm the added items in the virtual cart. A state 725 of the virtual cart is now “confirmed”. At step 726, the member may leave the group. A state 727 of the member is now “left”. At step 728, the member may rejoin the group. A state 723 of the member is now “joined”. At step 729,

the user (i.e., the group creator) may kick out the member from the group. A state 730 of the member is now “kicked”. The group may contain the following information in Table 1:

Table 1	
Field	Description
ID	unique id for the group used for sharing with the members
CREATOR	id of the group creator
MEMBERS	id of the members joined the group
STATES	states of all the members in the group
MERCHANT	primary merchant id on which the group was created
DELIVERY_LOCATION	the delivery location of the group creator

[00122] In various embodiments, the link may be a deep link to the application. The deep link may open to a specific page in the application when the link is clicked. In some embodiments, on clicking the link, the application will open. After the user signs in, the application may open a view where the user may have an option to select the items from the selected merchant including the nearby merchants. After selecting the items from all the merchants, the user may verify and may confirm the items.

[00123] A flowchart 731 of FIG. 7F shows a backend process when members join the group. After the group creator (i.e., user) shares the link to other members, members may join the group to add items to the virtual cart. At step 732, on clicking the shared link received from the group creator, the application may show the group, and may show an option to join the group. On clicking the join option, the application calls the PAX API. At step 733, the PAX API may validate if the group exists and if the group exists, it may add the member to the group.

[00124] A flowchart 734 of FIG. 7G shows a backend process when members leave the group. In an embodiment, a member who joined the group may leave the group any time before the group order has been placed. At step 735, when a member clicks on leave group, the application may call the PAX API. At step 736, the PAX API may validate if the group exists and if the member belongs to the group, and may call a food-cart module to delete all the items added by this member. At step 737, the food-cart module may update the virtual cart. At step 738, the PAX API may validate if the group exists and if the member belongs to the group. If member belongs to the group, the PAX API may remove the member from

the group. All the items added by this member from the group order may be removed by calling the food-cart module.

[00125] A flowchart 739 of FIG. 7H shows a backend process when members gets kicked out by the group creator. At step 740, the application of the group creator may call the PAX API. At step 741, the PAX API may validate if the member to be kicked belongs to the group and may call the food-cart module to delete the items added by the kicked out member. At step 742, the food-cart module may update the virtual cart. At step 743, the PAX API may update the member state in the group.

[00126] A flowchart 744 of FIG. 7I shows a backend process when the group creator cancels the group. The group creator may cancel the group if the group order has not been placed. At step 745, when the group creator clicks on cancel the group, the application may call the PAX API. At step 746, the PAX API may call the food-cart module to delete all the items from the virtual cart if any items added before. At step 747, the food-cart module may clear the items from the cart. At step 748, the PAX API may delete the group from its database.

[00127] A flowchart 749 of FIG. 7J shows a backend process when a member views the merchants to add items to the virtual cart. At step 750, to view merchants, the application may call the PAX API. At step 751, the PAX API may fetch group details to validate the member. At step 752, the PAX API may use the group details to call the recommendation system to fetch all the nearby merchants to be recommended for the group order. At step 753, the PAX API may fetch the merchant's details, and/or menu items from food data service and may return the response to the application.

[00128] A flowchart 754 of FIG. 7K shows a backend process when a member adds items to the virtual cart. At step 755, the application may call the PAX API. At step 756, the PAX API may validate the member states in the group (i.e., if members belong to the group). If member is not a part of the group, it may return the error. At step 757, the PAX API may call the food-cart module to create and/or update the virtual cart with the selected items. At step 758, the food-cart module may update the items in the virtual cart for all members of the group.

[00129] A flowchart 759 of FIG. 7L shows a backend process when a member confirms the items in the virtual cart. At step 760, the application may call the PAX API after the member may confirm the items in the virtual cart. At step 761, the PAX API may update the member state in the group to "CONFIRMED".

[00130] A flowchart 762 of FIG. 7M shows a backend process when the members and/or the group creator view the items added by the members to the virtual cart. At step 763, the application may call the PAX API to retrieve the virtual cart details. At step 764, the PAX API may retrieve the group information. At step 765, the PAX API may retrieve the virtual cart details based on the group from the food-cart module. At step 766, the food-cart module may retrieve the virtual cart details from the database.

[00131] A flowchart 767 of FIG. 7N shows a backend process when the group creator places the order. At step 768, after items have been added by the members to the virtual cart, the user may check out the virtual cart, e.g., by clicking on order. The application may call the PAX API to create the order. Once the order is placed successfully, the PAX API may poll for the order status, and details continuously until order is completed, for e.g, when the items are delivered. At step 769, the PAX API may call an order manager to create the order. At step 770, the order manager may fetch the virtual cart for the order from the food-cart module. At step 771, the order manager may hold the money from the application wallet and/or a credit card and/or any suitable paying method by calling the cashier and registering the order. Alternatively, if a debit card is used, the order manager may deduct the amount instead of holding. The order manager may notify the user through the application that the order is placed successfully.

[00132] FIG. 8 shows a flowchart 800 of various order states, according to various embodiments.

[00133] The steps shown in the flowchart 800 are shown in a specific order, however other arrangements are possible. Steps may also be combined in some cases. Any suitable order of steps may be used.

[00134] At state 801, the user order may be created after the user checks out the items in the virtual cart, and the state of the order may be changed to “created”. At step 802, the merchant may accept the user order. At state 803, after the merchant accepts the order, the state of the order may be changed to “confirmed”. At step 804, a service provider (i.e., a driver) may be allocated by the server to deliver the items to the user. At state 805, before the driver collects the items from the merchants, the state of the order may be “pickup”. At step 806, the service provider may collect the order from the merchant, or the plurality of merchants in the case of a multi-merchant order. At state 807, after the service provider collects the order, the state of the order may be changed to “dropoff”. At step 808, the service provider may deliver the items to the user. In the case of a multi-merchant order, the service

provider may deliver the items from the plurality of merchants to the user at the same time. At state 809, after the service provider delivers the order, the state of the order may be changed to “completed”. In some embodiments, at state 801 of the order being “created”, at step 810, the user may cancel the order or the merchant may reject the order, resulting in a state 811 of the order being changed to “cancelled”. In some embodiments, in the case of a multi-merchant order, when a merchant of the plurality of merchants that the user ordered from rejects the order, the server may inform the user through the user device. The user may have an option of cancelling the whole order or to proceed with the order without the item from the merchant who rejected. In some embodiments, after the merchant or the plurality of merchant accept the order, and the state of the order is at the state 803 of “confirmed”, at step 812, the server may fail to allocate a service provider to deliver the items to the user. This may be due to lack of available service providers. When the server fails to allocate a service provider, a state 813 occurs, and the state of the order may be changed to “unallocated”.

[00135] FIGS. 9A to 9B show flowcharts illustrating different backend processes, according to various embodiments.

[00136] The steps shown in the flowcharts of FIGS. 9A to 9B are shown in a specific order, however other arrangements are possible. Steps may also be combined in some cases. Any suitable order of steps may be used.

[00137] A flowchart 900 of FIG. 9A shows a backend process when merchants are notified about the order. At step 901, after the order is created, an order manager may notify a preparation API. At step 902, the preparation API may notify all the different merchants involved about the order via a merchant API, and may wait for all the merchants to confirm the order. At step 903, the merchant API may notify a merchant app on a merchant device of the merchant about the order.

[00138] A flowchart 904 of FIG. 9B shows a backend process when merchants review and confirm the order. At step 905, after the merchant app is notified about the order, the merchant may review the items in the order, and may confirm the order by calling the merchant API. At step 906, the merchant API may notify the preparation API. At step 907, once the preparation API receives confirmation from all the merchants, the preparation API may notify the order manager with the preparation time for each merchant. At step 908, the application may keep polling for the order status and details. At step 909, the PAX API may fetch the order status with details such as the merchant accepting the order.

[00139] FIGS. 10A to 10F show flowcharts illustrating different backend processes, according to various embodiments.

[00140] The steps shown in the flowcharts of FIGS. 10A to 10F are shown in a specific order, however other arrangements are possible. Steps may also be combined in some cases. Any suitable order of steps may be used.

[00141] A flowchart 1000 of FIG. 10A shows a backend process when a driver (i.e., a service provider) is assigned for order delivery. At step 1001, after the order is accepted by the merchants, the order manager may notify deliver task with the order details including preparation time of each merchant for order delivery. At step 1002, a delivery API may notify driver allocation to assign a driver for the order and may keep polling for driver details. At step 1003, the driver allocation may fetch all available drivers around the location along with their storage capacities. At step 1004, the driver allocation may broadcast the task to the available drivers and may wait for driver to accept the task. At step 1005, the driver API may notify driver about the task details.

[00142] A flowchart 1006 of FIG. 10B shows a backend process when a driver accepts the order delivery. At step 1007, after the driver receives the task, driver may accept the task and may notify driver API. At step 1008, the driver API may notify the driver allocation about driver confirmation. At step 1009, the delivery API may fetch driver details from the driver allocation. At step 1010, the delivery API may notify the order manager about driver details. At step 1011, the application may poll for order status and details. At step 1012, the PAX API may fetch status and details having driver details and driver location map.

[00143] A flowchart 1013 of FIG. 10C shows a backend process of a driver routing to merchant location. At step 1014, the driver app may call the driver API for on-going tasks. At step 1015, the driver API may fetch on-going task from the driver-allocation. At step 1016, the driver API may fetch the delivery task details from the delivery API which may contain the information of the sequence in which driver should visit the merchants to collect the items. The delivery API may decide the routing based on preparation time of the merchants for the order.

[00144] A flowchart 1017 of FIG. 10D shows a backend process when the driver pickup items from the merchant. At step 1018, after the driver arrives at the merchant location, the driver may verify the item details and may collect them. Once the driver indicates that he has collected the items, for e.g., clicking collect on the driver app, the driver app may call the driver API. At step 1019, the driver API may notify the delivery API about the items

collected from the merchant. At step 1020, after all the items are collected from all the merchants, the delivery API may notify the order manager that the order has been picked up from all the merchants. A notification may be sent to the user through the user application on the user device that the order has been picked up. At step 1021, the user application may poll for order status and details. At step 1022, the PAX API may fetch the order status and details which may include information that the order has been picked up by the driver.

[00145] A flowchart 1023 of FIG. 10E shows a backend process when the driver arrives at user location. At step 1024, on arriving at user location, the driver may indicate that he has arrived at user location, for e.g., by clicking “Arrived” the on driver app, which may call the driver API. At step 1025, the driver API may notify the delivery API that the driver has arrived at user location. At step 1026, the user may receive a notification that the driver has arrived with the items ordered. At step 1027, the application may poll for order status. At step 1028, the PAX API may fetch order status and details. The order status and details may include information that the driver has arrived at user location.

[00146] A flowchart 1029 of FIG. 10F shows a backend process when the driver delivers the order. At step 1030, after the items are delivered to the user, the driver may indicate that he has complemented the order, for e.g., by clicking on “complete” in driver app, which may call the driver API. At step 1031, the driver API may notify the delivery API that the task is delivered. At step 1032, the delivery API may inform the driver allocation that the driver is available for next task. At step 1033, the delivery API may update the order status as completed in the order manager. At step 1034, once the user order is completed, the order manager may call cashier to deduct the amount from the user. At step 1035, the cashier may deduct money from the payment method chosen by the user while creating the order. At step 1036, the application may poll for order status. At step 1037, the PAX API may fetch order status and details. The order status and details may include information that the order is completed.

[00147] While the invention has been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the scope of the invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced.

CLAIMS

1. A server configured for multi-merchant orders, comprising:
 - one or more processor(s); and
 - a memory having instructions stored therein, the instructions, when executed by the one or more processor(s), cause the one or more processor(s) to:
 - receive in real time, a first merchant data from a user device of a user when the user device is displaying information from a first merchant;
 - determine at least one suitable merchant to display on the user device of the user using the first merchant data and historical data in a database;
 - display information of the at least one suitable merchant on the user device of the user;
 - receive a user order from the user, wherein the user order comprises a first order data regarding a first item from the first merchant, and a second order data regarding a second item from a second merchant; and
 - send the user order to a service provider device of a service provider for the service provider to pick up the first item from the first merchant, and the second item from the second merchant, and to deliver the first item and the second item to the user at a same time;
 - wherein prior to receiving the user order from the user, the one or more processor(s) is configured to receive from the user device of the user, information regarding a virtual cart of the user;
 - wherein the virtual cart comprises information regarding potential items to be checked out;
 - wherein the one or more processor(s) is configured to receive from a second user device of a second user, information regarding an item that the second user adds to the virtual cart using the second user device, and update the virtual cart displayed on the user device of the user; and
 - wherein the one or more processor(s) is configured to receive the user order from the user when the user checks out items from the virtual cart.
2. The server of claim 1, wherein the one or more processor(s) is configured to determine the at least one suitable merchant based on at least one of: a location of the first

merchant, a category of items sold by the first merchant, a preparation time of the first merchant, and user preferences.

3. The server of claim 2, wherein the one or more processor(s) is configured to determine the at least one suitable merchant by:

determining the location of the first merchant through the first merchant data or through the historical data in the database;

searching the historical data in the database for at least one potential merchant, wherein the at least one potential merchant has a location that is within a distance threshold from the location of the first merchant; and

setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

4. The server of claim 2, wherein the one or more processor(s) is configured to determine the at least one suitable merchant by:

determining the location of the first merchant through the first merchant data or through the historical data in the database;

searching the historical data in the database for at least one potential merchant, wherein a time taken to travel between the at least one potential merchant and the first merchant is within a time threshold; and

setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

5. The server of claim 2, wherein the one or more processor(s) is configured to determine the at least one suitable merchant by:

determining the category of items sold by the first merchant through the first merchant data or through the historical data in the database;

searching the historical data in the database for at least one potential merchant that is selling a similar category of items as the category of items sold by the first merchant; and

setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

6. The server of claim 2, wherein the one or more processor(s) is configured to determine the at least one suitable merchant by:

determining the category of items sold by the first merchant through the first merchant data or through the historical data in the database;

searching the historical data in the database for at least one potential merchant that is selling a complementary category of items as the category of items sold by the first merchant; and

setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

7. The server of claim 2, wherein the one or more processor(s) is configured to determine the at least one suitable merchant by:

determining the preparation time of the first merchant through the first merchant data;

searching the historical data in the database for at least one potential merchant with a historical preparation time that is within a preparation time threshold compared to the preparation time of the first merchant;

sending a preparation time information request to a device of the at least one potential merchant;

receiving a preparation time of the at least one potential merchant from the device of the at least one potential merchant;

determining if the preparation time of the at least one potential merchant is within the preparation time threshold compared to the preparation time of the first merchant; and

setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user when the preparation time of the at least one potential merchant is within the preparation time threshold.

8. The server of claim 2, wherein the one or more processor(s) is configured to determine the at least one suitable merchant by:

searching the historical data in the database for the user preferences;

searching the historical data in the database for at least one potential merchant that matches the user preferences; and

setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

9. A method for multi-merchant orders comprising:
 - using one or more processor(s) of a server to:
 - receive in real time, a first merchant data from a user device of a user when the user device is displaying information from a first merchant;
 - determine at least one suitable merchant to display on the user device of the user using the first merchant data and historical data in a database;
 - display information of the at least one suitable merchant on the user device of the user;
 - receive a user order from the user, wherein the user order comprises a first order data regarding a first item from the first merchant, and a second order data regarding a second item from a second merchant; and
 - send the user order to a service provider device of a service provider for the service provider to pick up the first item from the first merchant, and the second item from the second merchant, and to deliver the first item and the second item to the user at a same time;
 - wherein prior to receiving the user order from the user, the one or more processor(s) is configured to receive from the user device of the user, information regarding a virtual cart of the user;
 - wherein the virtual cart comprises information regarding potential items to be checked out;
 - wherein the one or more processor(s) is configured to receive from a second user device of a second user, information regarding an item that the second user adds to the virtual cart using the second user device, and update the virtual cart displayed on the user device of the user; and
 - wherein the one or more processor(s) is configured to receive the user order from the user when the user checks out items from the virtual cart.

10. The method of claim 9, comprising:
 - using the one or more processor(s) to:

determine the at least one suitable merchant based on at least one of: a location of the first merchant, a category of items sold by the first merchant, a preparation time of the first merchant, and user preferences.

11. The method of claim 10, comprising:

using the one or more processor(s) to determine the at least one suitable merchant by:

determining the location of the first merchant through the first merchant data or through the historical data in the database;

searching the historical data in the database for at least one potential merchant, wherein the at least one potential merchant has a location that is within a distance threshold from the location of the first merchant; and

setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

12. The method of claim 10, comprising:

using the one or more processor(s) to determine the at least one suitable merchant by:

determining the location of the first merchant through the first merchant data or through the historical data in the database;

searching the historical data in the database for at least one potential merchant, wherein a time taken to travel between the at least one potential merchant and the first merchant is within a time threshold;

setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

13. The method of claim 10, comprising:

using the one or more processor(s) to determine the at least one suitable merchant by:

determining the category of items sold by the first merchant through the first merchant data or through the historical data in the database;

searching the historical data in the database for at least one potential merchant that is selling a similar category of items as the category of items sold by the first merchant; and

setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

14. The method of claim 12, comprising:

using the one or more processor(s) to determine the at least one suitable merchant by:

determining the category of items sold by the first merchant through the first merchant data or through the historical data in the database;

searching the historical data in the database for at least one potential merchant that is selling a complementary category of items as the category of items sold by the first merchant; and

setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

15. The method of claim 10, comprising:

using the one or more processor(s) to determine the at least one suitable merchant by:

determining the preparation time of the first merchant through the first merchant data;

searching the historical data in the database for at least one potential merchant with a historical preparation time that is within a preparation time threshold compared to the preparation time of the first merchant;

sending a preparation time information request to a device of the at least one potential merchant;

receiving a preparation time of the at least one potential merchant from the device of the at least one potential merchant;

determining if the preparation time of the at least one potential merchant is within the preparation time threshold compared to the preparation time of the first merchant; and

setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user when the preparation time of the at least one potential merchant is within the preparation time threshold.

16. The method of claim 10, comprising:

using the one or more processor(s) to determine the at least one suitable merchant by:

- searching the historical data in the database for the user preferences;
- searching the historical data in the database for at least one potential merchant that matches the user preferences; and
- setting the at least one potential merchant as the at least one suitable merchant to display on the user device of the user.

17. A non-transitory computer-readable medium storing computer executable code comprising instructions for multi-merchant orders according to any one of claims 1 to 16.

18. A computer executable code comprising instructions for multi-merchant orders according to any one of claims 1 to 17.