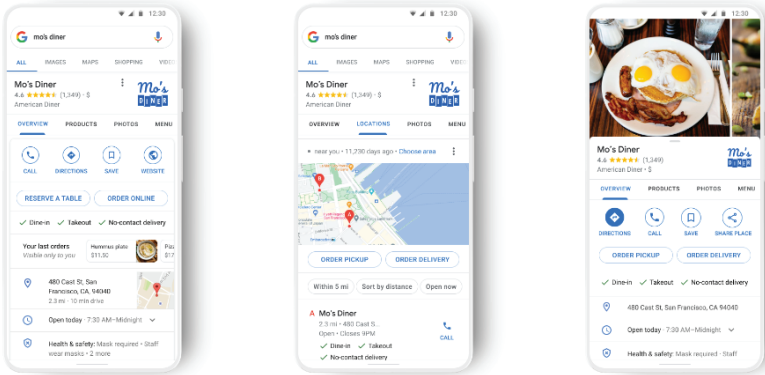
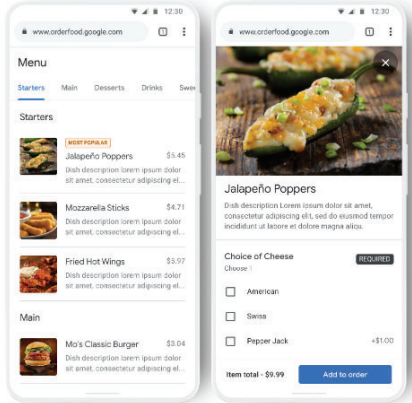
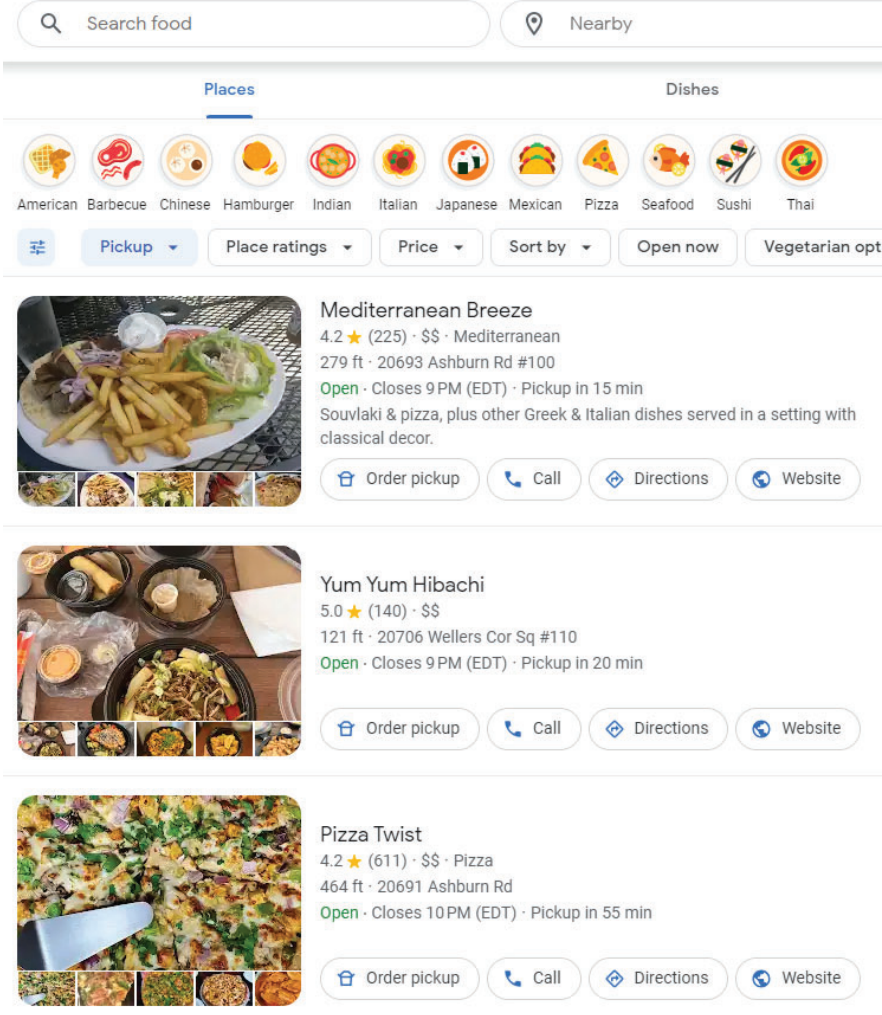


**EXHIBIT E**

**EXHIBIT E**  
**(Page 1)**

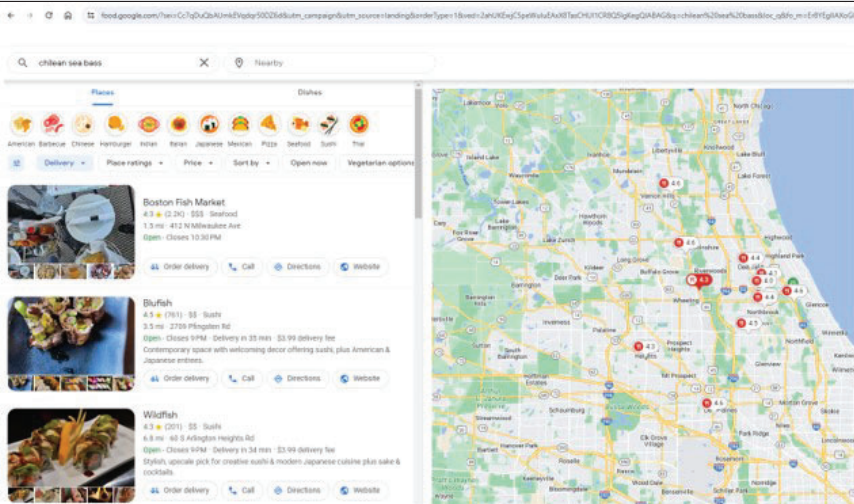
<b>ELEMENT(S)</b>	<b>INDEPENDENT CLAIM 1: 11,038,090</b>	<b>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</b>
<p>Preamble, [1a]</p>	<p>A computer-implemented method for providing a searchable aggregated data structure for a networked application, the method comprising:</p>	<p>The Accused Product/Service is a computer-implemented method for providing a searchable aggregated data structure for a networked application for users to select their food choices as shown under the “Overview and Eligibility” on Google Food Developer Site:</p> <p>Overview and Eligibility <span>📄</span> <span>Send feedback</span></p> <p>Ordering End-to-End lets partners take food orders from an end user and process that order for fulfillment with restaurants in their network.</p> <p>Ubiquitous and optimized for conversion</p> <p>Google makes ordering food available across mobile and desktop platforms through Google Search and Maps.</p>  <p>Flexible ordering experience</p> <p>Additional features such as order ahead, menu search, suggested related items, popular items, and reorder provide a flexible and enriched user experience.</p> <p>The user selects their food choices ⇄</p> <p>The user can browse through multiple menu sections and options to select their food choices, and add items to their cart along the way. They can customize their order with add-ons or special instructions, repeat previous orders, and add suggested or popular items. Then, they make their food choices and add them to their cart.</p> 

**EXHIBIT E**  
**(Page 2)**

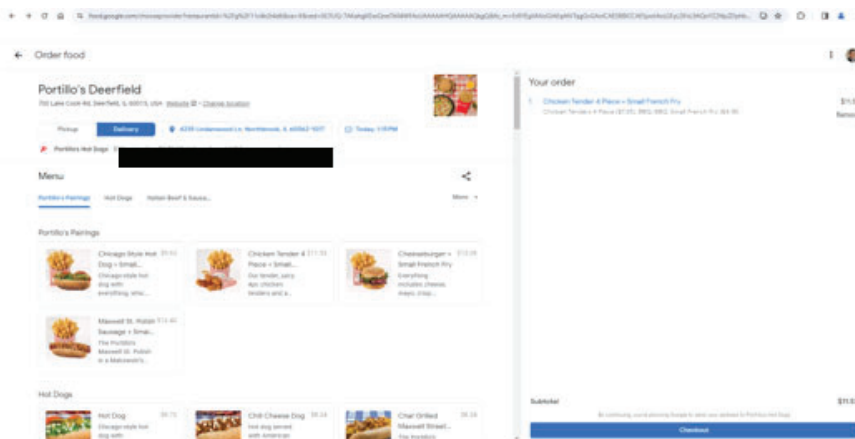
<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 1: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
		 <p>The screenshot shows the Google Food interface with a search bar and filters. Three restaurant listings are visible:</p> <ul style="list-style-type: none"> <li><b>Mediterranean Breeze:</b> 4.2 stars (225 reviews), \$\$ price, Mediterranean cuisine. Address: 279 ft · 20693 Ashburn Rd #100. Status: Open. Closes 9 PM (EDT). Pickup in 15 min. Description: Souvlaki &amp; pizza, plus other Greek &amp; Italian dishes served in a setting with classical decor.</li> <li><b>Yum Yum Hibachi:</b> 5.0 stars (140 reviews), \$\$ price, Hibachi/Japanese cuisine. Address: 121 ft · 20706 Wellers Cor Sq #110. Status: Open. Closes 9 PM (EDT). Pickup in 20 min.</li> <li><b>Pizza Twist:</b> 4.2 stars (611 reviews), \$\$ price, Pizza cuisine. Address: 464 ft · 20691 Ashburn Rd. Status: Open. Closes 10 PM (EDT). Pickup in 55 min.</li> </ul> <p>Each listing includes a main image, a row of smaller images, and buttons for 'Order pickup', 'Call', 'Directions', and 'Website'.</p>
		<p>Google Food allows a user to search food and menus for delivery from its member restaurants in a given location, including by common menu item, and with an “Order Delivery” button, as shown below for “chilean sea bass”:</p>

**EXHIBIT E**  
**(Page 3)**

<b>ELEMENT(S)</b>	<b>INDEPENDENT CLAIM 1: 11,038,090</b>	<b>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</b>
-------------------	--	--



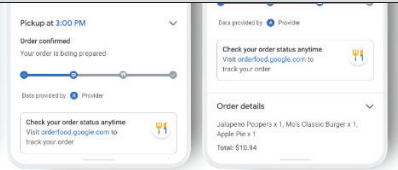
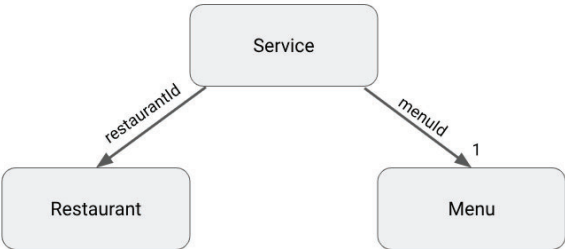
Using Google Food’s computer-implemented method for providing a searchable aggregated data structure for a networked application, a user can access a particular restaurant menu and order delivery directly through Google Food, as shown below for Portillo’s in Deerfield, Illinois:



See (<https://developers.google.com/actions-center/verticals/ordering/e2e/overview>)

[1b]	acquiring by a processor source data from a plurality of delivery service computers associated with a plurality of food or beverage delivery services over a communication network, the	Google Food acquires source data, by a processor, from a plurality of delivery service computers associated with a plurality of food or beverage delivery services over a communication network, the acquired source data being in a plurality of formats, where the acquired source data includes, for each of the plurality of food or beverage delivery services, data representing multiple source menu items, under the heading “Integration” step on the Google Food Developer Site (also shown below):
------	---	---

**EXHIBIT E**  
**(Page 4)**

<b>ELEMENT(S)</b>	<b>INDEPENDENT CLAIM 1: 11,038,090</b>	<b>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</b>
	<p>acquired source data being in a plurality of formats, where the acquired source data includes, for each of the plurality of food or beverage delivery services, data representing multiple source menu items, wherein said acquiring source data comprises one or more of:</p>	<div style="text-align: right; margin-bottom: 10px;">  </div> <p><b>Integration</b></p> <p>For your food business to connect with Google's Ordering End-to-End support, you must integrate with Google systems and APIs. There are three processes you need to implement to connect with Ordering End-to-End:</p> <ol style="list-style-type: none"> <li><b>1. Provide restaurant, menu, and service data feeds to Google.</b></li> </ol> <p>The first step of the integration process is to create and host data feeds about your restaurant, menu, and service. These feeds provide details about restaurant name, location, service hours, menu items and sections, delivery areas, and more. Google ingests your data feeds and uses them to present your menu and services to users. You can update these feeds regularly and even incorporate incremental changes in real time.</p> <p>To implement these processes, the Ordering End-to-End integration has two main components: Inventory feeds and fulfillment actions.</p> <ul style="list-style-type: none"> <li><b>Inventory feeds</b> These feeds use a relational inventory schema to supply Google with up-to-date information about a restaurant, the services it provides, and the items in its menu.</li> </ul> <p>The structure of your Ordering End-to-End data feed is defined by the relational inventory schema. The Ordering End-to-End data feed consists of the following top-level entities:</p> <ul style="list-style-type: none"> <li><b>Restaurant entities:</b> Which restaurants you service.</li> <li><b>Service entities:</b> Timing, location, and conditions of your service.</li> <li><b>Menu entities:</b> Details of each restaurant's menu.</li> </ul> <p>The following diagram shows how Service, Restaurant, and Menu entities represent a single restaurant:</p> <div style="text-align: center; margin: 10px 0;">  </div> <p><i>Figure 1: Overall relationship of Ordering End-to-End data feed entities: Service, Restaurant, and Menu.</i></p> <p><b>General guidelines</b></p> <ul style="list-style-type: none"> <li><b>Restaurants per file:</b> Each data file should represent one restaurant with its related Service and Menu entities. Use filenames that can help you search a file for a restaurant.</li> <li><b>Data file format:</b> Data files need to be formatted in newline-separated JSON files (ndjson format).</li> <li><b>DateTime and Time values:</b> For properties that require a DateTime or Time value, use the formats specified in DateTime and Time formats. For example, 2017-05-01T06:30:00+05:30 for DateTime and T08:08:00+05:30 for Time.</li> <li><b>IDs:</b> Use an @id property to identify all unique entities within an entity type. The maximum length is 300 characters. An @id is a unique identifier of the entity of that type, but across entities, IDs can overlap. For example, assume you define a Service entity with the @id property set to a16. You cannot create another Service entity with an @id of a16. However, you can use a16 as the @id value of a Menu entity.</li> <li><b>ID Generation:</b> Keep your ids stable - do not use UUIDs or otherwise change/randomize ids between feed uploads. This allows easier support for entity-related issues.</li> <li><b>Null values:</b> Do not use the value null in place of objects. If an object is optional, you must omit it from your feed.</li> </ul> <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/overview">https://developers.google.com/actions-center/verticals/ordering/e2e/overview</a>)</p>
[1c]	employing an application programming interface to interface with the plurality of	<p>Google Food employs an application programming interface (API) to interface with the plurality of delivery service computers, as shown under the heading “Integration” step on Google Food Developer Site (also shown below):</p>

**EXHIBIT E**  
**(Page 5)**

<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 1: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
	<p>delivery service computers, or</p>	<p align="center"><b>Integration</b></p> <p>For your food business to connect with Google's Ordering End-to-End support, you must integrate with Google systems and APIs. There are three processes you need to implement to connect with Ordering End-to-End:</p> <ol style="list-style-type: none"> <li><b>1. Provide restaurant, menu, and service data feeds to Google.</b> The first step of the integration process is to create and host data feeds about your restaurant, menu, and service. These feeds provide details about restaurant name, location, service hours, menu items and sections, delivery areas, and more. Google ingests your data feeds and uses them to present your menu and services to users. You can update these feeds regularly and even incorporate incremental changes in real time.</li> <li><b>2. Handle order fulfillment.</b> After a user is ready to order, Google lets them review and modify their cart details before their order is processed and submitted. As part of the Ordering End-to-End integration process, you create a webhook URL that validates and receives the orders from Google. You process online payments through a Google Pay participating processor.</li> <li><b>3. Support order updates.</b> To provide post-order experiences on Google's surfaces, you send updates to a Google API. Google then shows the information to your customer. These include the order status, estimated fulfillment time, customer service information, and other changes that might impact their order. Users who order food can view the state of their purchases in Google.</li> </ol> <p>Google Food also employs an Async Order Update API to interface with the plurality of delivery service computers for order updates and for users to check their order status, shown under the heading "Integration" step on Google Food Developer Site (also shown below):</p> <div align="center" data-bbox="630 1081 1442 1297"> <pre> graph LR     Google[Google] -- "AsyncOrderUpdateRequestMessage" --&gt; Webhook[Partner webhook]     Webhook -- "AsyncOrderUpdateResponseMessage" --&gt; Google     </pre> <p>The diagram shows a bidirectional flow between Google (green box) and a Partner webhook (yellow box). The outgoing message is labeled "AsyncOrderUpdateRequestMessage" and contains two possible states: "state = 'CONFIRMED' label = 'Accepted by restaurant'" or "state = 'REJECTED' label = 'reason'". The incoming message is labeled "AsyncOrderUpdateResponseMessage".</p> </div> <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/overview">https://developers.google.com/actions-center/verticals/ordering/e2e/overview</a>)</p>
<p>[1d]</p>	<p>scraping data from the plurality of delivery service computers;</p>	<p>Google Food scrape data from the plurality of delivery service computers by "ingesting" the data feeds, as shown below under the "Integration" step on Google Food Developer Site:</p> <p align="center"><b>Integration</b></p> <p>For your food business to connect with Google's Ordering End-to-End support, you must integrate with Google systems and APIs. There are three processes you need to implement to connect with Ordering End-to-End:</p> <ol style="list-style-type: none"> <li><b>1. Provide restaurant, menu, and service data feeds to Google.</b> The first step of the integration process is to create and host data feeds about your restaurant, menu, and service. These feeds provide details about restaurant name, location, service hours, menu items and sections, delivery areas, and more. Google ingests your data feeds and uses them to present your menu and services to users. You can update these feeds regularly and even incorporate incremental changes in real time.</li> </ol>

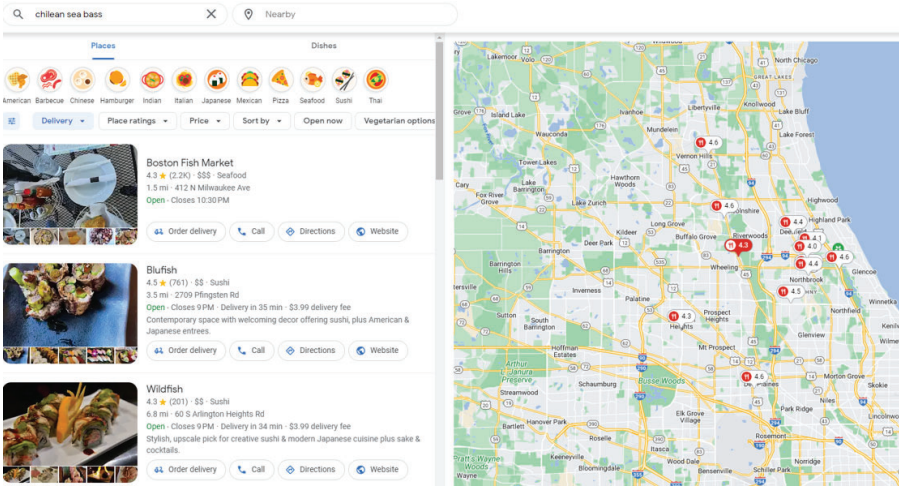
**EXHIBIT E**  
**(Page 6)**

<b><u>ELEMENT(S)</u></b>	<b><u>INDEPENDENT CLAIM 1: 11,038,090</u></b>	<b><u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u></b>
		<p>Update your data feeds</p> <p>After your data feeds are connected, Google checks for updates once each hour, but we only ingest all data feeds when the <code>marker.txt</code> or <code>sitemap.xml</code> files have been modified. We expect that you update your data feeds once a day to prevent stale inventory.</p> <p>To specify that the data feeds have been modified and are ready for batch ingestion, update the <code>last-modified</code> object metadata field of the <code>marker.txt</code> file (For GCP and S3) or the <code>last-modified</code> response header of the <code>sitemap.xml</code> file. Google uses these values to determine how fresh a data feed is.</p> <p>As the batch feed is being ingested,</p> <ul style="list-style-type: none"> <li>• New entities that don't exist in your current Ordering End-to-End inventory and don't have any errors would be inserted.</li> <li>• Entities already present in the inventory that don't have any errors on ingestion and either have a <code>dateModified</code> more recent than their current entry or in the case of not having a <code>dateModified</code> the feed ingestion start time is more recent than the current entry they would be updated, otherwise they would be marked as stale.</li> <li>• Entities that were part of a previous feed that are no longer included in the batch feed being processed would be deleted, provided there are no file level errors in the feed.</li> </ul> <p>The timestamp or the <code>last-modified</code> response header must be updated only after all of the data feed files are generated and updated. Limit the batch jobs that update your data feeds to run only once a day. Alternatively, have a gap of at least three hours between each batch job. If you don't take these steps, Google might fetch stale files.</p> <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/overview">https://developers.google.com/actions-center/verticals/ordering/e2e/overview</a>)</p>
[1e]	mapping by the processor the acquired service data according to a predetermined data format to provide formatted data;	<p>Google Food maps the acquired data according to a predetermined data format to provide formatted data. As shown below, Google Food maps the acquired service data (data feed) using relational inventory schema, under the heading “Integration” step on Google Food Developer Sit.</p> <ul style="list-style-type: none"> <li>• <b>Inventory feeds</b></li> </ul> <p>These feeds use a <a href="#">relational inventory schema</a> (<code>/actions-center/verticals/ordering/e2e/reference/feeds/relational-inventory-schema</code>) to supply Google with up-to-date information about a restaurant, the services it provides, and the items in its menu.</p> <p>Inventory feeds</p> <p>Inventory feeds are how a user discovers and orders from a desired restaurant. The <a href="#">relational inventory schema</a> defines the structure of your Ordering End-to-End inventory. An Ordering End-to-End inventory feed consists of the following main entities:</p> <ul style="list-style-type: none"> <li>• <code>Restaurant</code></li> <li>• <code>Service</code></li> <li>• <code>Menu</code></li> </ul> <p>The mapping of the ingested data is formatted for the Ordering End-to-End data feeds (Food Catalog Specification) using the JSON data schema.</p>

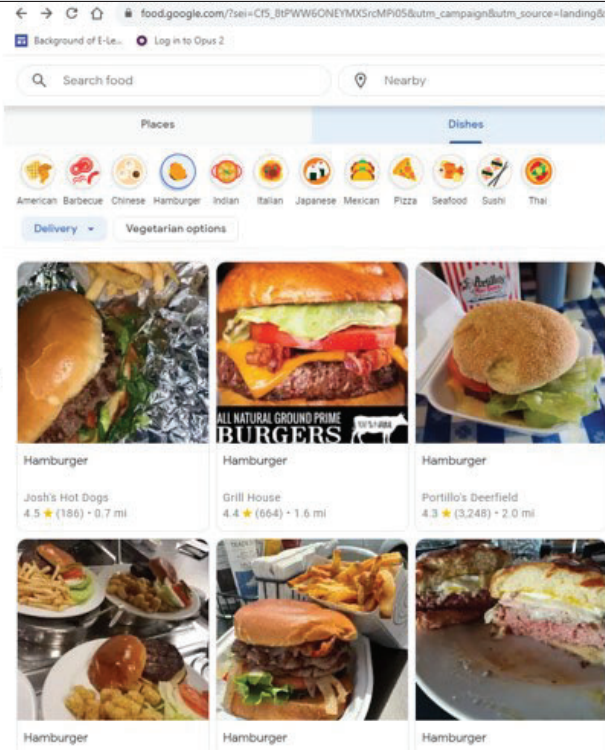
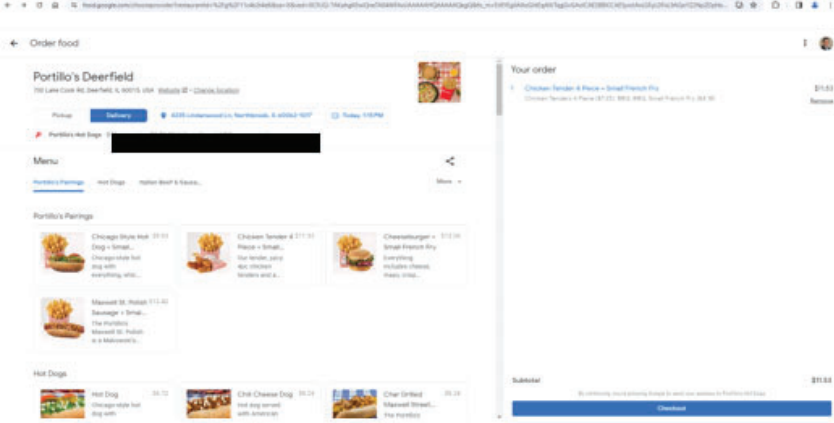
**EXHIBIT E**  
**(Page 7)**

<b><u>ELEMENT(S)</u></b>	<b><u>INDEPENDENT CLAIM 1: 11,038,090</u></b>	<b><u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u></b>
		<p style="text-align: center;"><b>Relational inventory schema</b> <span style="float: right;"><a href="#">Send feedback</a></span></p> <p>This page describes the format for the Ordering End-to-End data feeds (Food Catalog Specification) you provide to Google. For a machine-readable version of this information, you can <a href="#">download the JSON schema</a>.</p> <p><b>General requirements</b></p> <p>Entities must be structured to be on one line per entity in the feeds (entities are separated by newline characters). For readability purposes, JSON examples on this page do not follow that structure. However, you must follow that structure when sending your feeds. For example, a menu entity must be structured like the following code:</p> <pre style="background-color: #f0f0f0; padding: 5px;">{"@type": "Menu", "name": "Coffee Shop A", "@id": "1535"}</pre> <p>Each 'Restaurant' entity may have two Service entities (one each for the 'DELIVERY' and 'TAKEOUT' service types). Each 'Service' entity may only have one 'Menu' entity.</p> <p>Any sub-entities can be reused across multiple restaurants.</p> <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/overview">https://developers.google.com/actions-center/verticals/ordering/e2e/overview</a>)</p>
[1f]	linking by the processor the formatted data to common sources of the source menu items, each of the common sources being represented by source identification data uniquely identifying a respective one of the sources, such that at least one food or beverage delivery service is linked to each common source and its source menu items;	<p>Google Food links the formatted data, by the processor, to common the formatted data to common sources of the source menu items, each of the common sources being represented by source identification data uniquely identifying a respective one of the sources, such that at least one food or beverage delivery service is linked to each common source and its source menu items, under the heading “Integration” step on Google Food Developer Site. Google Food links the formatted data by restaurant, service and menu items by using the “Inventory Feed” component of its Ordering End-to-End Integration system (part of the Google Food Developer Site).</p> <p><b>Inventory feeds</b></p> <p>Inventory feeds are how a user discovers and orders from a desired restaurant. The <a href="#">relational inventory schema</a> defines the structure of your Ordering End-to-End inventory. An Ordering End-to-End inventory feed consists of the following main entities:</p> <ul style="list-style-type: none"> <li>• <a href="#">Restaurant</a></li> <li>• <a href="#">Service</a></li> <li>• <a href="#">Menu</a></li> </ul> <p><b>Menu</b></p> <p>The Menu entity describes a menu that a restaurant offers to its customers. A Menu entity must be defined within a Restaurant entity through the Restaurant's <code>menuId</code> property.</p> <p>Other objects that define the menu (such as description, image, add-ons, and nutrition information) include <code>MenuItem</code> and <code>MenuItemOffer</code>, as well as the optional objects <code>MenuSection</code>, <code>Availability</code>, and <code>MenuItemOption</code>.</p> <p><b>Menu sections</b></p> <p>You can use <code>MenuSection</code> objects to organize multiple <code>MenuItem</code> objects into logical categories. This approach is useful for restaurants that have multiple menus (like breakfast, lunch, and dinner). Add each menu as a separate <code>MenuSection</code>.</p> <p>In addition, Google Food offers merchant matching/linking services in order to ensure that inventory is shown to users in the right locations, under the Partner Portal of the Google Food Developer Site.</p>

**EXHIBIT E**  
**(Page 8)**

<b>ELEMENT(S)</b>	<b>INDEPENDENT CLAIM 1: 11,038,090</b>	<b>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</b>
		<p>Ordering End-to-End Actions Center Merchant Matching and Testing <span style="float: right;">Send feedback</span></p> <p>Match your inventory</p> <p>The Inventory Viewer is your endpoint to view all of the inventory that has been submitted to the Actions Center along with its matching status.</p> <p>What is matching?</p> <p>Matching is how you ensure that inventory is shown to users in the right locations. A Match is the connection between the Match Input, information you provide that are used for matching (typically information like merchant name, address, geo coordinate, phone number, and url), and the Match Output (typically a Google listing such as a business profile).</p> <p>All inventory that shares the same Match Input also share the same Match Output. This means that when a match is changed for one piece of inventory, all other inventory that uses that Match Input is updated as well. When changing the match the Actions Center lists all the inventory that would be impacted by the change.</p> <p>Within the Inventory Viewer the Matched column displays the match status (yes or no) on each piece of inventory.</p> <p><b>Note:</b> You can use filters, such as an entity ID or match status, to locate inventory.</p> <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/partner-portal/inventory/merchant-matching">https://developers.google.com/actions-center/verticals/ordering/e2e/partner-portal/inventory/merchant-matching</a>)</p>
[1g]	<p>identifying by the processor common menu items among the source menu items in the formatted data, and, for each identified common menu item, associating the source menu items with a master menu item;</p>	<p>Google Food identifies, by the processor, common menu items among the source menu items in the formatted data, and, for each identified common menu item, associating the source menu items with a master menu item. For instance, Google Food allows a user to search food and menus for delivery from its member restaurants in a given location, including by common menu item, and with an “Order Delivery” button, as shown here for “chilean sea bass”:</p>  <p>Google Food also groups restaurants by restaurant type, as shown here for “hamburger”:</p>

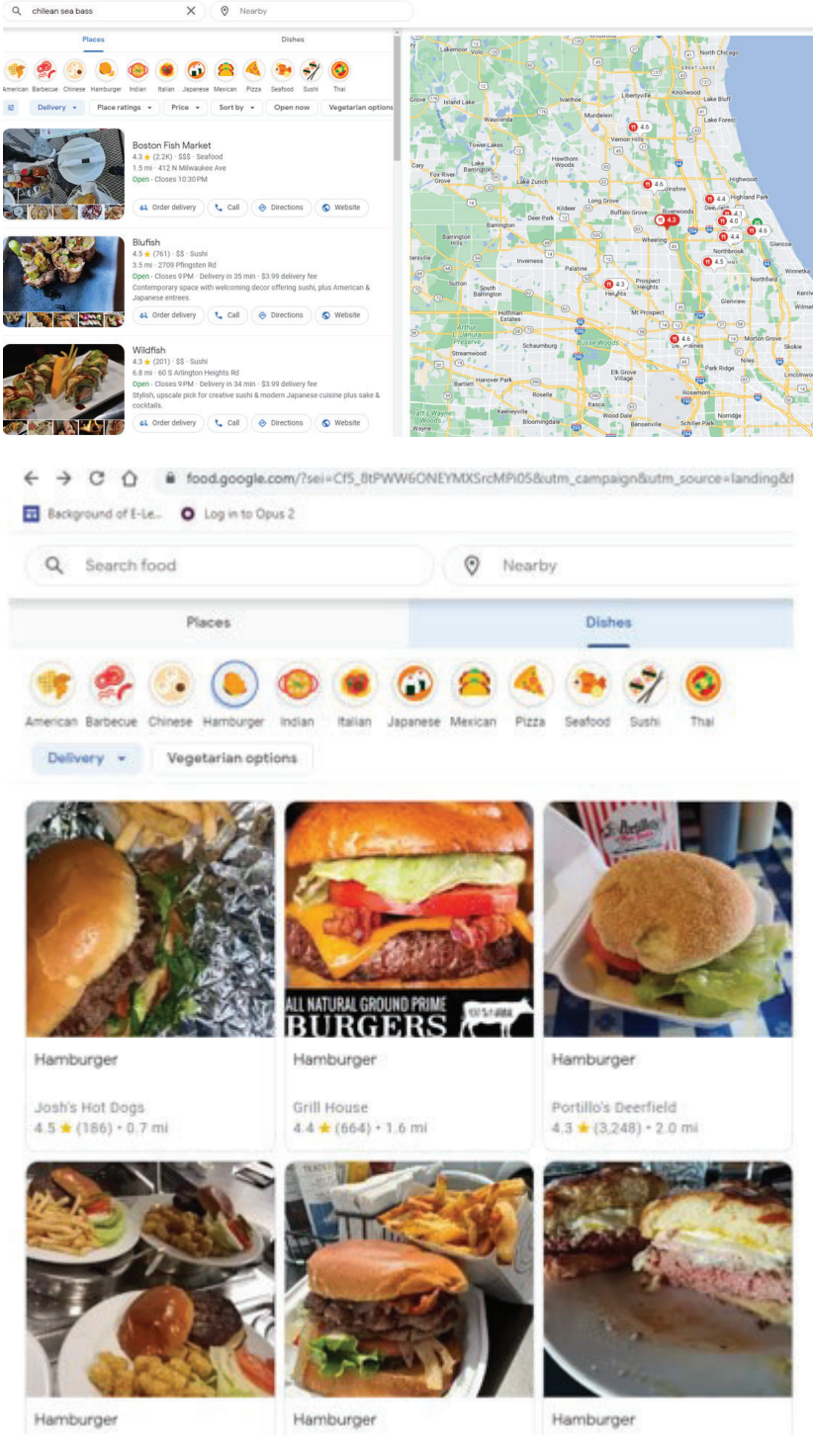
**EXHIBIT E**  
(Page 9)

<b>ELEMENT(S)</b>	<b>INDEPENDENT CLAIM 1: 11,038,090</b>	<b>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</b>
		 <p>In addition, a user can access a particular restaurant menu and order delivery directly through Google Food, as shown here for Portillo's in Deerfield, Illinois:</p>  <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/overview">https://developers.google.com/actions-center/verticals/ordering/e2e/overview</a>)</p>
[1h]	combining by the processor the linked data and the master menu items into a	See above charts illustrating how Google Food combines the linked data and the master menu items into a master data set as shown in the "Inventory Feeds" and "Relational Inventory Schema" under the heading "Integration" step on Google Food Developer Site for its Ordering End-to-End Integration System (also shown below).

**EXHIBIT E**  
**(Page 10)**

<b><u>ELEMENT(S)</u></b>	<b><u>INDEPENDENT CLAIM 1: 11,038,090</u></b>	<b><u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u></b>
	master data set; and	<p>Inventory feeds</p> <p>Inventory feeds are how a user discovers and orders from a desired restaurant. The <a href="#">relational inventory schema</a> defines the structure of your Ordering End-to-End inventory. An Ordering End-to-End inventory feed consists of the following main entities:</p> <ul style="list-style-type: none"> <li>• <a href="#">Restaurant</a></li> <li>• <a href="#">Service</a></li> <li>• <a href="#">Menu</a></li> </ul> <p>Batch feeds</p> <p>Google uses a daily batch feed of your inventory feeds to make Ordering End-to-End available to users. To keep your inventory up to date, you must update your batch feeds at least once a day for Google to fetch. It takes about two hours for your inventory to be updated by a batch.</p> <p>Incremental Updates API</p> <p>You can send time-sensitive updates of your inventory to Google. The Incremental Updates API lets you update and delete entities in your inventory in almost real time. Incremental updates are processed in no more than five minutes. This is primarily intended for updates that you can't foresee, such as emergency restaurant closures or removal of an out-of-stock item. If your change doesn't need to be reflected immediately, use the batch feeds instead.</p> <p style="text-align: center;"> <b>Relational inventory schema</b> <span style="float: right;"><a href="#">Send feedback</a></span> </p> <p>This page describes the format for the Ordering End-to-End data feeds (Food Catalog Specification) you provide to Google. For a machine-readable version of this information, you can <a href="#">download the JSON schema</a>.</p> <p><b>General requirements</b></p> <p>Entities must be structured to be on one line per entity in the feeds (entities are separated by newline characters). For readability purposes, JSON examples on this page do not follow that structure. However, you must follow that structure when sending your feeds. For example, a menu entity must be structured like the following code:</p> <pre style="background-color: #f0f0f0; padding: 5px;">{"@type": "Menu", "name": "Coffee Shop A", "@id": "1535"}</pre> <p>Each "Restaurant" entity may have two Service entities (one each for the "DELIVERY" and "TAKEOUT" service types). Each "Service" entity may only have one "Menu" entity.</p> <p>Any sub-entities can be reused across multiple restaurants.</p> <p><b>Menu</b></p> <p>The Menu entity describes a menu that a restaurant offers to its customers. A Menu entity must be defined within a Restaurant entity through the Restaurant's <code>menuId</code> property.</p> <p>Other objects that define the menu (such as description, image, add-ons, and nutrition information) include <code>MenuItem</code> and <code>MenuItemOffer</code>, as well as the optional objects <code>MenuSection</code>, <code>Availability</code>, and <code>MenuItemOption</code>.</p> <p><b>Menu sections</b></p> <p>You can use <code>MenuSection</code> objects to organize multiple <code>MenuItem</code> objects into logical categories. This approach is useful for restaurants that have multiple menus (like breakfast, lunch, and dinner). Add each menu as a separate <code>MenuSection</code>.</p> <p>Google Food, by combining the linked data and the master menu items into a master data set (a) allows a user to search food and menus for delivery from its member restaurants in a given location, including by common menu item, and with an "Order Delivery" button for "chilean sea bass" and/or (b) groups restaurants by restaurant type, as shown below for "hamburger":</p>

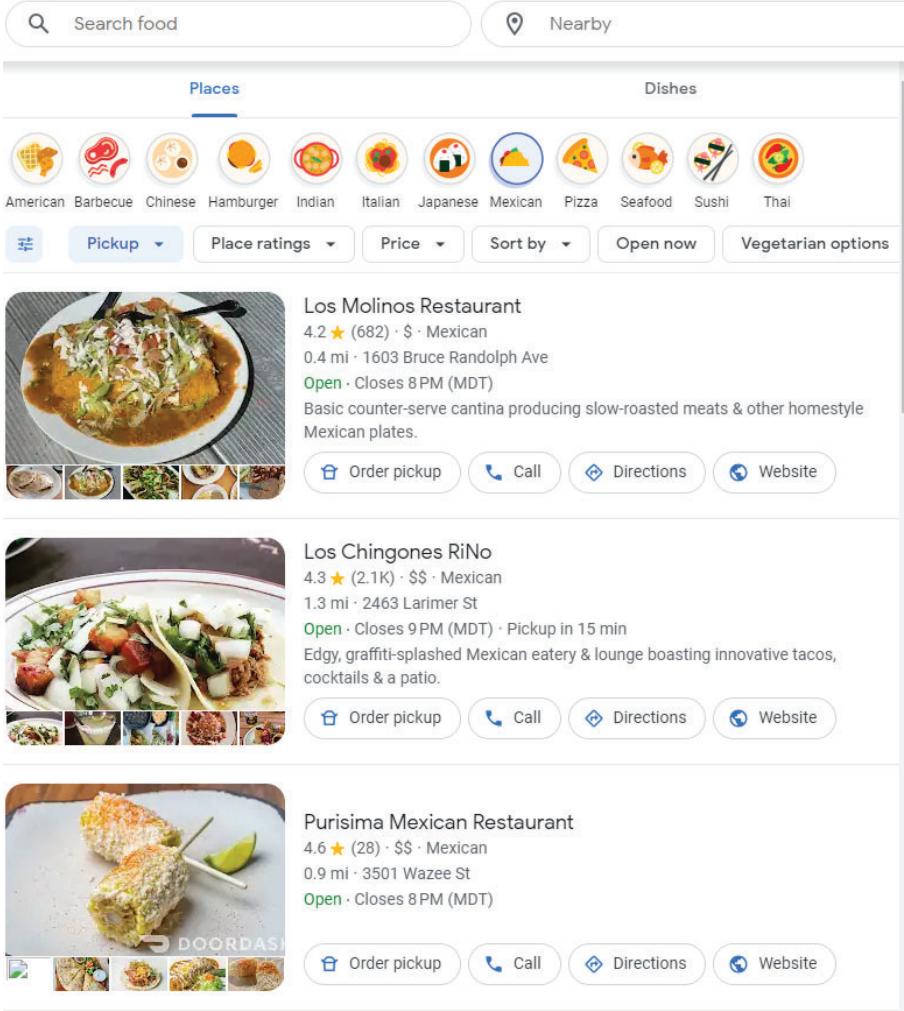
**EXHIBIT E**  
**(Page 11)**

<b>ELEMENT(S)</b>	<b>INDEPENDENT CLAIM 1: 11,038,090</b>	<b>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</b>
		 <p>The screenshot displays a Google Food search interface. At the top, a search bar contains 'chilean sea bass' and a location filter for 'Nearby'. Below the search bar, there are tabs for 'Places' and 'Dishes'. The 'Dishes' tab is active, showing a grid of food items. The top row of the grid includes three restaurant listings: Boston Fish Market, Blufish, and Wildfish. Below these, there is a search bar for 'Search food' and a 'Nearby' filter. The main content area shows a grid of hamburger images with their respective restaurant names and ratings: Josh's Hot Dogs (4.5 stars, 186 reviews, 0.7 mi), Grill House (4.4 stars, 664 reviews, 1.6 mi), and Portillo's Deerfield (4.3 stars, 3,248 reviews, 2.0 mi). The bottom row of the grid also shows three hamburger images, each labeled 'Hamburger'.</p>
[1i]	importing by the processor the master data set and the source identification data	See above charts illustrating how Google Food imports the master data set and the source identification data into the searchable aggregated data structure, under the “Inventory Feeds” and “Relational Inventory Schema” sections of the “Integration”

**EXHIBIT E**  
**(Page 12)**

<b><u>ELEMENT(S)</u></b>	<b><u>INDEPENDENT CLAIM 1: 11,038,090</u></b>	<b><u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u></b>
	<p>into the searchable aggregated data structure.</p>	<p>step on Google Food Developer Site for its Ordering End-to-End Integration System (also shown below).</p> <p><b>Inventory feeds</b></p> <p>Inventory feeds are how a user discovers and orders from a desired restaurant. The <a href="#">relational inventory schema</a> defines the structure of your Ordering End-to-End inventory. An Ordering End-to-End inventory feed consists of the following main entities:</p> <ul style="list-style-type: none"> <li>• <a href="#">Restaurant</a></li> <li>• <a href="#">Service</a></li> <li>• <a href="#">Menu</a></li> </ul> <p><b>Batch feeds</b></p> <p>Google uses a daily batch feed of your inventory feeds to make Ordering End-to-End available to users. To keep your inventory up to date, you must update your batch feeds at least once a day for Google to fetch. It takes about two hours for your inventory to be updated by a batch.</p> <p><b>Incremental Updates API</b></p> <p>You can send time-sensitive updates of your inventory to Google. The Incremental Updates API lets you update and delete entities in your inventory in almost real time. Incremental updates are processed in no more than five minutes. This is primarily intended for updates that you can't foresee, such as emergency restaurant closures or removal of an out-of-stock item. If your change doesn't need to be reflected immediately, use the batch feeds instead.</p> <p><b>Relational inventory schema</b> <span style="float: right;"><a href="#">Send feedback</a></span></p> <p>This page describes the format for the Ordering End-to-End data feeds (Food Catalog Specification) you provide to Google. For a machine-readable version of this information, you can <a href="#">download the JSON schema</a>.</p> <p><b>General requirements</b></p> <p>Entities must be structured to be on one line per entity in the feeds (entities are separated by newline characters). For readability purposes, JSON examples on this page do not follow that structure. However, you must follow that structure when sending your feeds. For example, a menu entity must be structured like the following code:</p> <pre style="background-color: #f0f0f0; padding: 5px;">{"@type": "Menu", "name": "Coffee Shop A", "id": "1535"}</pre> <p>Each 'Restaurant' entity may have two Service entities (one each for the 'DELIVERY' and 'TAKEOUT' service types). Each 'Service' entity may only have one 'Menu' entity.</p> <p>Any sub-entities can be reused across multiple restaurants.</p>

**EXHIBIT E**  
**(Page 13)**

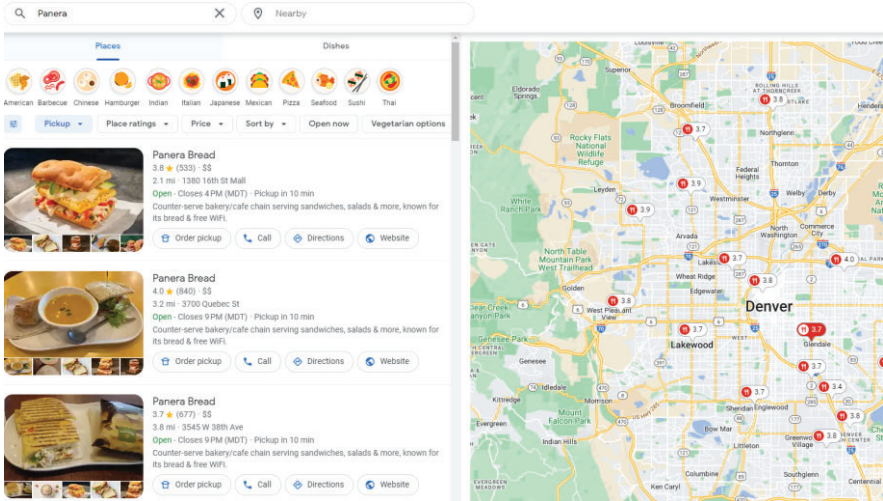
<b>ELEMENT(S)</b>	<b>INDEPENDENT CLAIM 1: 11,038,090</b>	<b>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</b>
		 <p>The screenshot shows a Google Food search interface. At the top, there is a search bar with the text "Search food" and a location filter set to "Nearby". Below the search bar, there are two tabs: "Places" (selected) and "Dishes". Under the "Places" tab, there is a row of food category icons: American, Barbecue, Chinese, Hamburger, Indian, Italian, Japanese, Mexican, Pizza, Seafood, Sushi, and Thai. Below the icons are several filter buttons: "Pickup", "Place ratings", "Price", "Sort by", "Open now", and "Vegetarian options".</p> <p>Three restaurant listings are visible:</p> <ul style="list-style-type: none"> <li><b>Los Molinos Restaurant:</b> 4.2 stars (682 reviews), \$ price range, Mexican cuisine. Address: 1603 Bruce Randolph Ave. Status: Open, closes 8 PM (MDT). Description: Basic counter-serve cantina producing slow-roasted meats &amp; other homestyle Mexican plates. Buttons: Order pickup, Call, Directions, Website.</li> <li><b>Los Chingones RiNo:</b> 4.3 stars (2.1K reviews), \$\$ price range, Mexican cuisine. Address: 2463 Larimer St. Status: Open, closes 9 PM (MDT), pickup in 15 min. Description: Edgy, graffiti-splashed Mexican eatery &amp; lounge boasting innovative tacos, cocktails &amp; a patio. Buttons: Order pickup, Call, Directions, Website.</li> <li><b>Purissima Mexican Restaurant:</b> 4.6 stars (28 reviews), \$\$ price range, Mexican cuisine. Address: 3501 Wazee St. Status: Open, closes 8 PM (MDT). Buttons: Order pickup, Call, Directions, Website.</li> </ul>

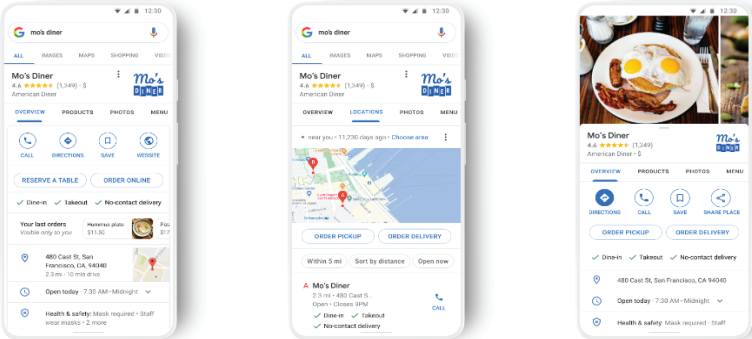
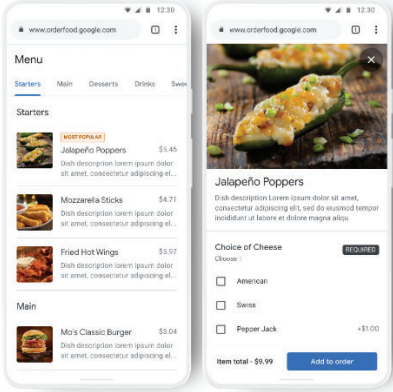
<u>ELEMENT(S)</u>	<u>DEPENDENT CLAIM 3: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>																		
[3a]	The method of claim 1, wherein said mapping the acquired data comprises: aliasing fields of the acquired data from formats used by the delivery service computers to respective fields of the predetermined data format:	<p>See above, claim chart for independent claim 1 illustrating Google Food aliasing fields of the acquired data from formats used by the plurality of delivery service computers to respective fields of the predetermined data format. under the “Integration” step on Google Food Developer Site. In addition Google Food meet this limitation, since the restaurant identifier data comprise one or more of restaurant name data, restaurant location data, or restaurant identification code, as shown below.</p> <p>Restaurant data</p> <p>Restaurant (required)</p> <p>A required entity to implement. Describes a restaurant.</p> <p>The following table lists the properties for the <code>Restaurant</code> type:</p> <table border="1"> <thead> <tr> <th>Property</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><code>@type</code></td> <td><code>Const</code></td> <td>Required. Value: <code>Restaurant</code></td> </tr> <tr> <td><code>@id</code></td> <td><code>String</code></td> <td>Required. A unique identifier of the restaurant or delivery provider. Example: <code>restaurant_1</code></td> </tr> <tr> <td><code>name</code></td> <td><code>String</code></td> <td>Required. Name of the restaurant. Example: <code>Foo</code></td> </tr> <tr> <td><code>description</code></td> <td><code>String</code></td> <td>A description of the restaurant. Example: <code>Best seafood in town</code></td> </tr> <tr> <td><code>url</code></td> <td><code>Url</code></td> <td>The URL that represents the restaurant. The restaurant domain is preferred over the aggregator domain. Example: <code>http://www.provider.com/someresaurant</code></td> </tr> </tbody> </table>	Property	Type	Description	<code>@type</code>	<code>Const</code>	Required. Value: <code>Restaurant</code>	<code>@id</code>	<code>String</code>	Required. A unique identifier of the restaurant or delivery provider. Example: <code>restaurant_1</code>	<code>name</code>	<code>String</code>	Required. Name of the restaurant. Example: <code>Foo</code>	<code>description</code>	<code>String</code>	A description of the restaurant. Example: <code>Best seafood in town</code>	<code>url</code>	<code>Url</code>	The URL that represents the restaurant. The restaurant domain is preferred over the aggregator domain. Example: <code>http://www.provider.com/someresaurant</code>
Property	Type	Description																		
<code>@type</code>	<code>Const</code>	Required. Value: <code>Restaurant</code>																		
<code>@id</code>	<code>String</code>	Required. A unique identifier of the restaurant or delivery provider. Example: <code>restaurant_1</code>																		
<code>name</code>	<code>String</code>	Required. Name of the restaurant. Example: <code>Foo</code>																		
<code>description</code>	<code>String</code>	A description of the restaurant. Example: <code>Best seafood in town</code>																		
<code>url</code>	<code>Url</code>	The URL that represents the restaurant. The restaurant domain is preferred over the aggregator domain. Example: <code>http://www.provider.com/someresaurant</code>																		

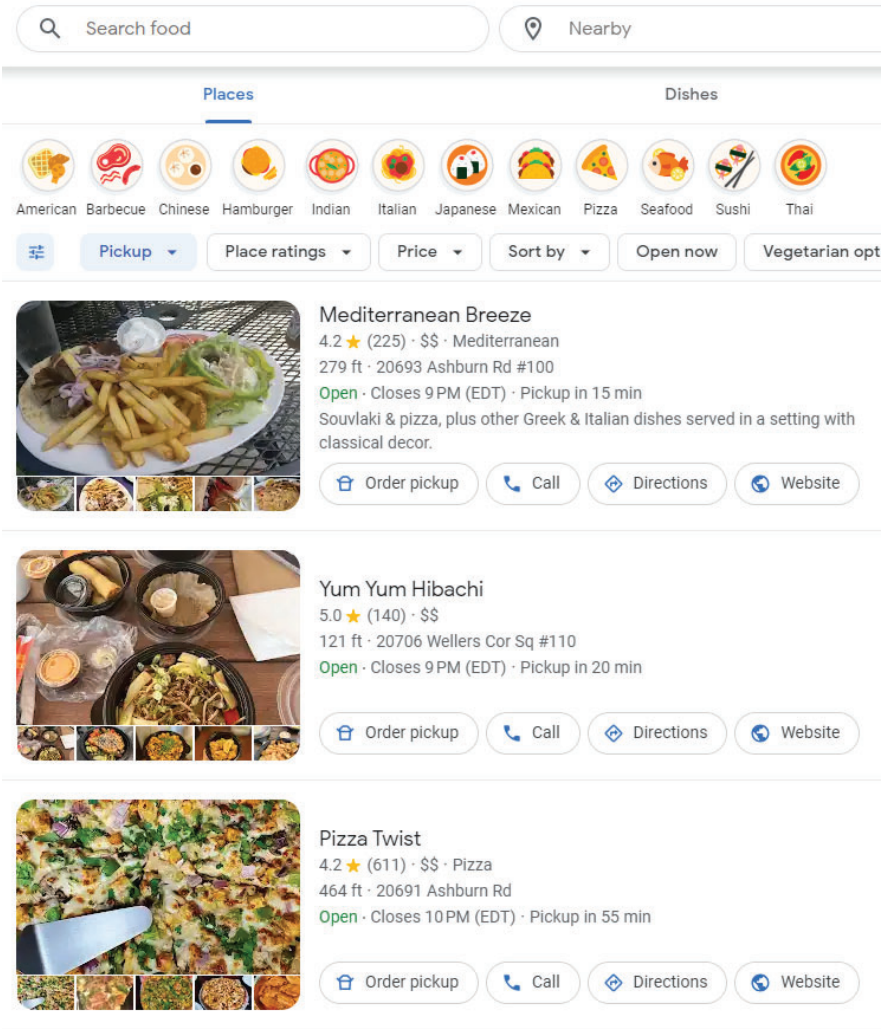
**EXHIBIT E**  
**(Page 15)**

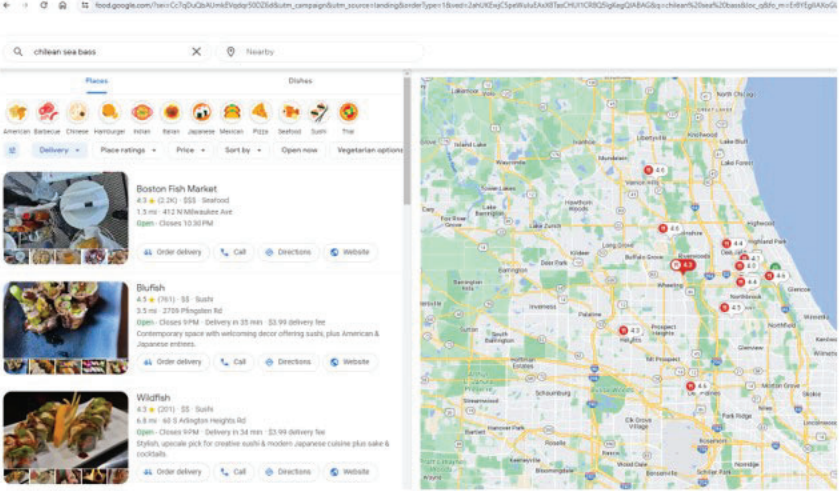
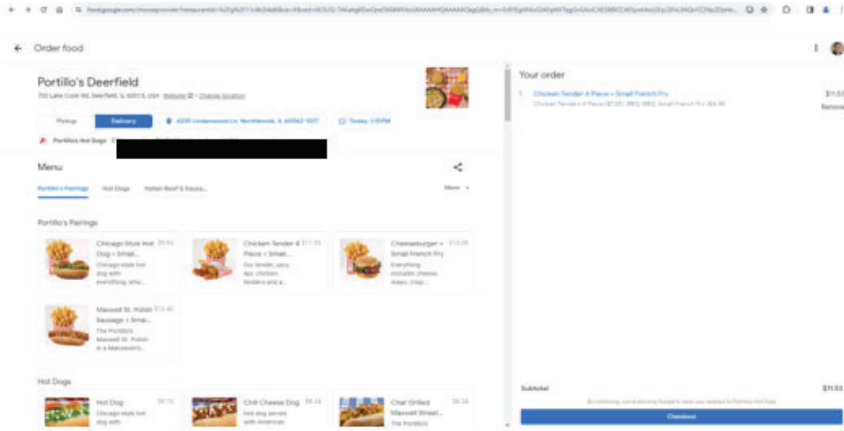
<b><u>ELEMENT(S)</u></b>	<b><u>DEPENDENT CLAIM 3: 11,038,090</u></b>	<b><u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u></b>
		<p>sameAs      <i>Url</i>      The official website for the restaurant. Example: <code>http://www.provider2.com/somerestaurant</code></p> <hr/> <p>telephone      <i>String</i>      Telephone number of the restaurant. Example: <code>+12345665898</code></p> <hr/> <p>streetAddress      <i>String</i>      Required. The street address of the restaurant. Example: <code>12345 Bar Avenu</code></p> <hr/> <p>addressLocality      <i>String</i>      Required. The locality or city. Example: <code>San Francisco</code></p> <hr/> <p>addressRegion      <i>String</i>      Required. The region or state. Example: <code>CA</code></p> <hr/> <p>postalCode      <i>String</i>      Required. The postal code. Example: <code>94124</code></p> <hr/> <p>addressCountry      <i>String</i>      Required. Two-letter ISO 3166-1 alpha-2 country code. Example: <code>US</code></p> <hr/> <p>latitude      <i>Number</i>      Latitude in degrees. Values are restricted to the range <code>[-90, 90]</code>. The precision should be at least 5 decimal places. Example: <code>35.7392607</code></p> <hr/> <p>longitude      <i>Number</i>      Longitude in degrees. Values are restricted to the range <code>[-180, 180]</code>. The precision should be at least 5 decimal places. Example: <code>-120.3895522</code></p> <p>Sample code from Google Food Developer Site implemented by Google Food to meet this limitation:</p> <div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <p style="margin: 0;"><u>Example</u></p> <pre style="margin: 0;">{   "@type": "Restaurant",   "@id": "10824",   "name": "Pronto Wood Fired Pizzeria",   "url": "https://www.provider.com/pronto-wood-fired-pizzeria",   "telephone": "+16503659978",   "streetAddress": "2560 El Camino Real",   "addressLocality": "Palo Alto",   "addressRegion": "CA",   "postalCode": "94061",   "addressCountry": "US",   "latitude": 37.472842,   "longitude": -122.217144 }</pre> </div>

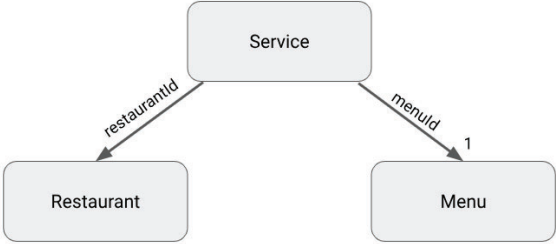
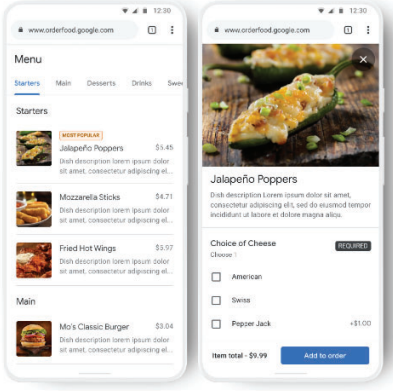
<u>ELEMENT(S)</u>	<u>DEPENDENT CLAIM 4: 11,037,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
[4a]	The method of claim 1, wherein the source identification data comprise one or more of name data, location data, or an identification code.	See above, claim chart for independent claim 1, where the source identification data comprise one or more of name data, location data, or an identification code.

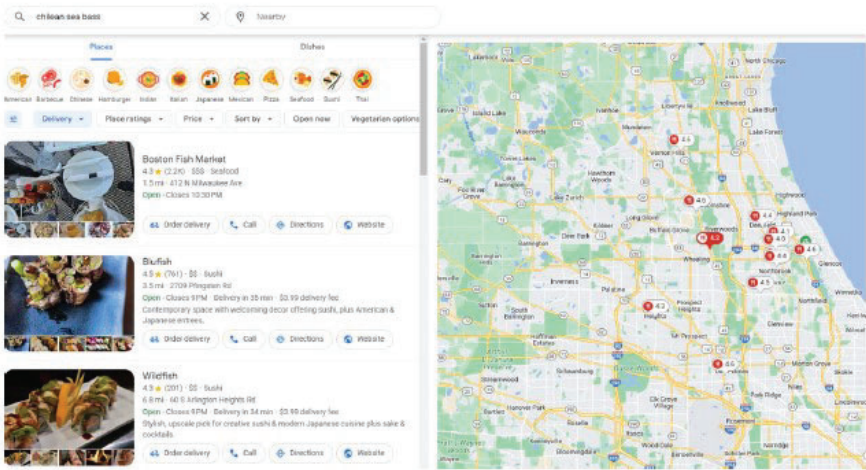
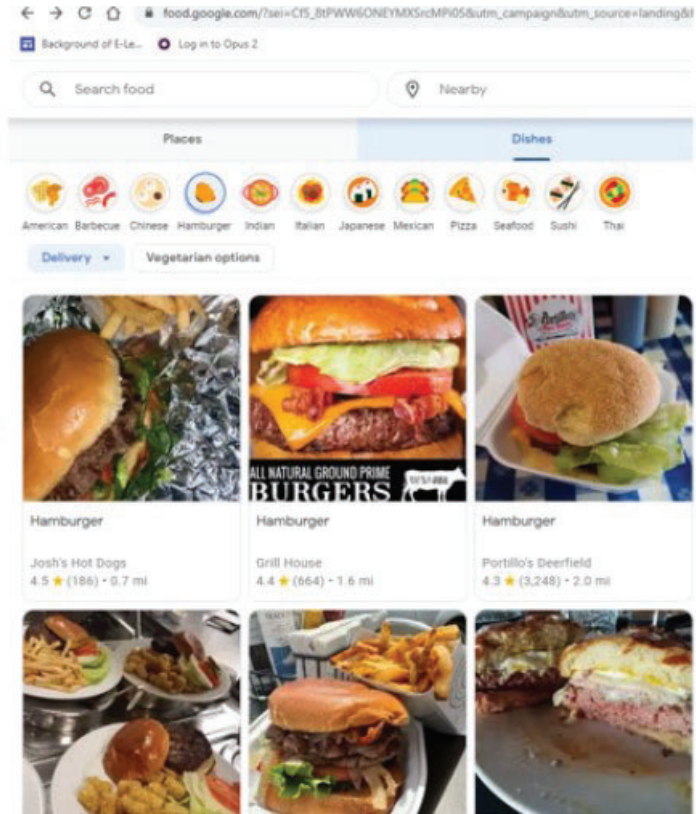
<u>ELEMENT(S)</u>	<u>DEPENDENT CLAIM 5: 11,037,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
[5a]	<p>The method of claim 4, wherein said linking comprises: analyzing the source identification data to determine the common sources; and linking the formatted data associated with like common sources to a master data object.</p>	<p>See above, claim chart for independent claim 1, showing how Google Food links by analyzing the source identification data to determine the common sources; and linking the formatted data associated with like common sources to a master data object, a under the “Integration” step on Google Food Developer Site for its Ordering End-to-End Integration System, as shown below.</p>  <p>The screenshot shows a Google Food search for 'Panera Bread'. The search results list three locations: 1. 1380 16th St Mall (3.8 rating, 533 reviews), 2. 3700 Quebec St (4.0 rating, 840 reviews), and 3. 2545 W 38th Ave (3.7 rating, 1073 reviews). Each listing includes a pickup button, call button, directions, and website link. To the right is a map of the Denver area with red location pins corresponding to the search results.</p>

<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 12: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
<p>Preamble, [12a]</p>	<p>A system for providing an interactive food or beverage ordering service accessible by a user computing device, the system comprising:</p>	<p>The Accused Product/Service is a system, an Ordering End-to-End Integration System as shown on the Google Food Developer Site, for providing an interactive food or beverage ordering service accessible by a user computing device, as shown below</p> <p><b>Overview and Eligibility</b> <span>📌</span> <span>Send feedback</span></p> <p>Ordering End-to-End lets partners take food orders from an end user and process that order for fulfillment with restaurants in their network.</p> <p>Ubiquitous and optimized for conversion</p> <p>Google makes ordering food available across mobile and desktop platforms through Google Search and Maps.</p>  <p><b>Flexible ordering experience</b></p> <p>Additional features such as order ahead, menu search, suggested related items, popular items, and reorder provide a flexible and enriched user experience.</p> <p>The user selects their food choices ⇄</p> <p>The user can browse through multiple menu sections and options to select their food choices, and add items to their cart along the way. They can customize their order with add-ons or special instructions, repeat previous orders, and add suggested or popular items. Then, they make their food choices and add them to their cart.</p> 

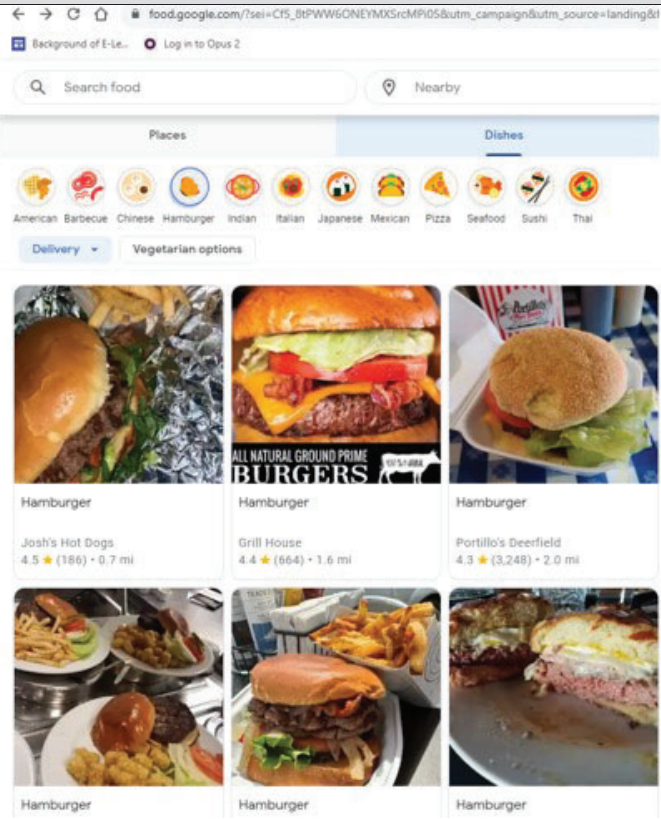
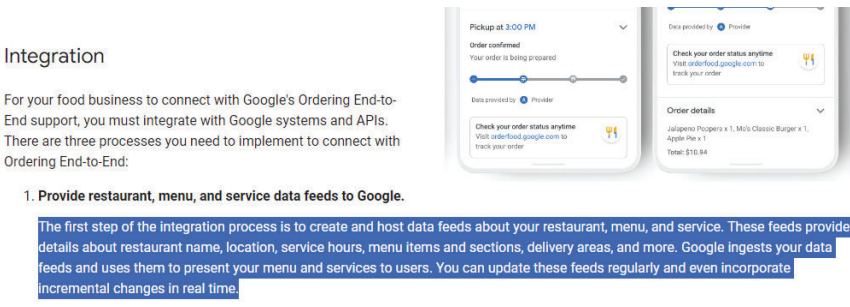
<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 12: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
		 <p>The screenshot displays the Google Food app interface. At the top, there is a search bar with the text "Search food" and a "Nearby" filter. Below the search bar, there are two tabs: "Places" (selected) and "Dishes". Under the "Places" tab, there is a row of icons representing various cuisines: American, Barbecue, Chinese, Hamburger, Indian, Italian, Japanese, Mexican, Pizza, Seafood, Sushi, and Thai. Below the icons, there are several filter buttons: "Pickup", "Place ratings", "Price", "Sort by", "Open now", and "Vegetarian opt".</p> <p>The main content area shows three restaurant listings:</p> <ul style="list-style-type: none"> <li><b>Mediterranean Breeze</b>: 4.2 ★ (225) · \$\$ · Mediterranean. 279 ft · 20693 Ashburn Rd #100. Open · Closes 9 PM (EDT) · Pickup in 15 min. Souvlaki &amp; pizza, plus other Greek &amp; Italian dishes served in a setting with classical decor. Buttons: Order pickup, Call, Directions, Website.</li> <li><b>Yum Yum Hibachi</b>: 5.0 ★ (140) · \$\$ · Hibachi. 121 ft · 20706 Wellers Cor Sq #110. Open · Closes 9 PM (EDT) · Pickup in 20 min. Buttons: Order pickup, Call, Directions, Website.</li> <li><b>Pizza Twist</b>: 4.2 ★ (611) · \$\$ · Pizza. 464 ft · 20691 Ashburn Rd. Open · Closes 10 PM (EDT) · Pickup in 55 min. Buttons: Order pickup, Call, Directions, Website.</li> </ul> <p>At the bottom of the screenshot, there is a caption: "The system allows a user to search food, beverages and menus for delivery from its member restaurants in a given location, including by common menu item, and with an "Order Delivery" button, as shown below for "chilean sea bass":"</p>

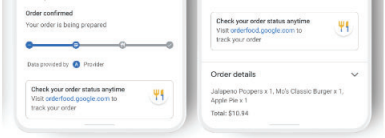

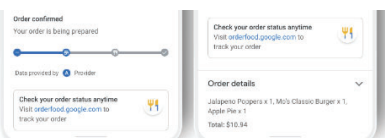
<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 12: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
		 <p>Using Google Food’s system for providing an interactive food or beverage ordering service accessible by a user computing device, a user can access a particular restaurant menu and order delivery directly through Google Food, as shown below for Portillo’s in Deerfield, Illinois:</p>  <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/overview">https://developers.google.com/actions-center/verticals/ordering/e2e/overview</a>)</p>
[12b]	a data acquisition and processing module comprising a processor, memory, and computer-readable instructions stored on a non-transitory medium that are executable by the processor to acquire source data from a	Google Food comprises a data acquisition and processing module (which evidently includes a processor, memory accessible to the processor, and a set of computer-readable instructions stored on a non-transitory medium that are executable by the processor) in order to acquire source data from a plurality of delivery service computers associated with a plurality of food or beverage delivery services and provide a master data set of formatted data, wherein the master data set includes, for each of the plurality of food or beverage delivery services, data representing multiple menu items linked to identification data uniquely identifying sources of the menu items delivered by the plurality of food or beverage delivery services as shown on Google Food Developer Site:



<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 12: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
	<p>plurality of delivery service computers associated with a plurality of food or beverage delivery services and provide a master data set of formatted data, wherein the master data set includes, for each of the plurality of food or beverage delivery services, data representing multiple menu items linked to identification data uniquely identifying sources of the menu items delivered by the plurality of food or beverage delivery services; and</p>	<p><b>Integration</b></p> <p>For your food business to connect with Google's Ordering End-to-End support, you must integrate with Google systems and APIs. There are three processes you need to implement to connect with Ordering End-to-End:</p> <ol style="list-style-type: none"> <li>1. <b>Provide restaurant, menu, and service data feeds to Google.</b></li> </ol> <p>The first step of the integration process is to create and host data feeds about your restaurant, menu, and service. These feeds provide details about restaurant name, location, service hours, menu items and sections, delivery areas, and more. Google ingests your data feeds and uses them to present your menu and services to users. You can update these feeds regularly and even incorporate incremental changes in real time.</p> <p><b>Inventory feeds</b></p> <p>Inventory feeds are how a user discovers and orders from a desired restaurant. The relational inventory schema defines the structure of your Ordering End-to-End inventory. An Ordering End-to-End inventory feed consists of the following main entities:</p> <ul style="list-style-type: none"> <li>• <b>Restaurant</b></li> <li>• <b>Service</b></li> <li>• <b>Menu</b></li> </ul> <p>The structure of your Ordering End-to-End data feed is defined by the relational inventory schema. The Ordering End-to-End data feed consists of the following top-level entities:</p> <ul style="list-style-type: none"> <li>• <b>Restaurant entities:</b> Which restaurants you service.</li> <li>• <b>Service entities:</b> Timing, location, and conditions of your service.</li> <li>• <b>Menu entities:</b> Details of each restaurant's menu.</li> </ul> <p>The following diagram shows how Service, Restaurant, and Menu entities represent a single restaurant:</p>  <p><i>Figure 1: Overall relationship of Ordering End-to-End data feed entities: Service, Restaurant, and Menu.</i></p> <p>The user selects their food choices ⇔</p> <p>The user can browse through multiple menu sections and options to select their food choices, and add items to their cart along the way. They can customize their order with add-ons or special instructions, repeat previous orders, and add suggested or popular items. Then, they make their food choices and add them to their cart.</p>  <p>Google Food provides a master data set of formatted data, wherein the master data set includes, for each one of the plurality of food delivery computers, data representing multiple menu items provided by multiple restaurants. For instance, Google Food allows a user to search food and menus for delivery from its member</p>

<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 12: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
		<p>restaurants in a given location, including by common menu item, and with an “Order Delivery” button, as shown here for “chilean sea bass”:</p>  <p>Google Food also groups restaurants, within its master data set, by restaurant type, as shown here for “hamburger”:</p>  <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/overview">https://developers.google.com/actions-center/verticals/ordering/e2e/overview</a>)</p>
[12c]	a website database accessible to the processor and	Google Food comprises a website database accessible to the processor and configured to receive updated data from the master data set, the master data set

<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 12: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
	<p>configured to receive updated data from the master data set, the master data set representing the multiple menu items provided by one or more of the plurality of food or beverage delivery services;</p>	<p>representing the multiple menu items provided by one or more of the plurality of food or beverage delivery services.</p> <p>As shown below, the first step of the integration process is to create and host data feeds about restaurants, menus, and services. These data feeds provide details about restaurant name, location, service hours, menu items and sections, delivery areas, and more. Next, Google Food ingests these data feeds and uses them to present menu and services to end-users. These data feeds are updated regularly and even incremental changes to the data feed is incorporated in real time within the Google Food system.</p> <div data-bbox="584 625 695 651" style="text-align: center;"> <p>Integration</p> </div> <div data-bbox="584 672 990 751"> <p>For your food business to connect with Google's Ordering End-to-End support, you must integrate with Google systems and APIs. There are three processes you need to implement to connect with Ordering End-to-End:</p> </div> <div data-bbox="609 766 998 787"> <p>1. Provide restaurant, menu, and service data feeds to Google.</p> </div> <div data-bbox="625 798 1445 882" style="border: 1px solid black; padding: 5px;"> <p>The first step of the integration process is to create and host data feeds about your restaurant, menu, and service. These feeds provide details about restaurant name, location, service hours, menu items and sections, delivery areas, and more. Google ingests your data feeds and uses them to present your menu and services to users. You can update these feeds regularly and even incorporate incremental changes in real time.</p> </div> <div data-bbox="1031 588 1445 756"> </div> <p>Further, as noted under the “Integration” step on Google Food Developer Site (and shown below), Google food is configured with specific sub-systems and modules such as Inventory Feeds, Batch Feeds and Incremental Updates API for receiving the most recent data from the master data set time-sensitive updates of the data.</p> <p>To implement these processes, the Ordering End-to-End integration has two main components: Inventory feeds and fulfillment actions.</p> <ul style="list-style-type: none"> <li>• <b>Inventory feeds</b></li> </ul> <p>These feeds use a relational inventory schema to supply Google with up-to-date information about a restaurant, the services it provides, and the items in its menu.</p> <p>Google Food provides master data set representing the multiple menu items provided by each one of the plurality of food or beverage delivery services for the multiple restaurants for end-to-end ordering, as shown below:</p> <div data-bbox="592 1354 1453 1816"> </div>

<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 12: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
		 <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/overview">https://developers.google.com/actions-center/verticals/ordering/e2e/overview</a>)</p>
[12d]	<p>wherein the data acquisition and processing module further comprises a plurality of modules in the form of a computer-readable instructions stored on a non-transitory medium that are executable by the processor including:</p>	<p>Google Food comprises a data acquisition and processing module (which evidently comprises a plurality of modules in the form of a computer-readable instructions stored on a non-transitory medium that are executable by the processor) in order to acquire and map the source data from a plurality of delivery service computers.</p>  <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/overview">https://developers.google.com/actions-center/verticals/ordering/e2e/overview</a>)</p>
[12e]	<p>an application programming interface configured to interface the data acquisition and processing module with the</p>	<p>Google Food comprises an application programming interface (API) to interface the data acquisition and processing module with the plurality of delivery service computers, as shown under the heading “Integration” step on Google Food Developer Site (also shown below):</p>

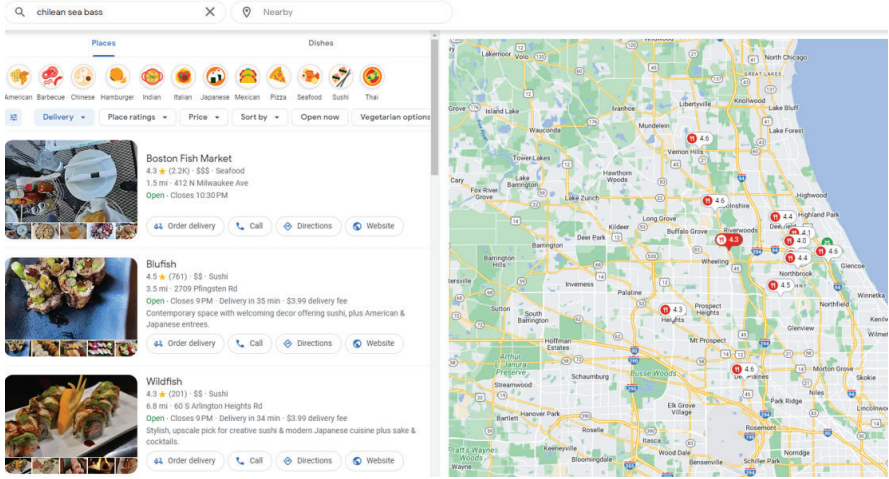
<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 12: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
	<p>plurality of delivery service computers;  <b>[DON'T SEE THE "OR" IN THIS LIMITATION- IT IS PROBABLY IN THE 683- Will check]</b></p>	<p><b>Integration</b></p> <p>For your food business to connect with Google's Ordering End-to-End support, you must integrate with Google systems and APIs. There are three processes you need to implement to connect with Ordering End-to-End:</p>  <ol style="list-style-type: none"> <li><b>1. Provide restaurant, menu, and service data feeds to Google.</b>                      The first step of the integration process is to create and host data feeds about your restaurant, menu, and service. These feeds provide details about restaurant name, location, service hours, menu items and sections, delivery areas, and more. Google ingests your data feeds and uses them to present your menu and services to users. You can update these feeds regularly and even incorporate incremental changes in real time.</li> <li><b>2. Handle order fulfillment.</b>                      After a user is ready to order, Google lets them review and modify their cart details before their order is processed and submitted. As part of the Ordering End-to-End integration process, you create a webhook URL that validates and receives the orders from Google. You process online payments through a <a href="#">Google Pay participating processor</a>.</li> <li><b>3. Support order updates.</b>                      To provide post-order experiences on Google's surfaces, you send updates to a Google API. Google then shows the information to your customer. These include the order status, estimated fulfillment time, customer service information, and other changes that might impact their order. Users who order food can view the state of their <a href="#">purchases in Google</a>.</li> </ol> <p>Google Food also employs an Async Order Update API to interface with the plurality of delivery service computers for order updates and for users to check their order status, shown under the heading "Integration" step on Google Food Developer Site (also shown below):</p>  <pre>                     graph LR                         Google[Google] -- "AsyncOrderUpdateRequestMessage" --&gt; Webhook[Partner webhook]                         Webhook -- "AsyncOrderUpdateResponseMessage" --&gt; Google                     </pre>
[12f]	<p>an extraction module configured to extract the source data from the plurality of delivery service computers as raw files by scraping data from one or more of the plurality of delivery service computers;</p>	<p>Google Food comprises an extraction module configured to extract the source data from the plurality of delivery service computers as raw files by scraping data from one or more of the plurality of delivery service computers using JSON data feed schemes for its Ordering and end-to-end integration system, as shown in the "Integration," "Inventory Feeds" and "Relational Inventory Schema" sections under the "Integration" step on the Google Food Developer Site.</p> <p><b>Integration</b></p> <p>For your food business to connect with Google's Ordering End-to-End support, you must integrate with Google systems and APIs. There are three processes you need to implement to connect with Ordering End-to-End:</p>  <ol style="list-style-type: none"> <li><b>1. Provide restaurant, menu, and service data feeds to Google.</b>                      The first step of the integration process is to create and host data feeds about your restaurant, menu, and service. These feeds provide details about restaurant name, location, service hours, menu items and sections, delivery areas, and more. Google ingests your data feeds and uses them to present your menu and services to users. You can update these feeds regularly and even incorporate incremental changes in real time.</li> </ol> <p><b>Inventory feeds</b></p> <p>Inventory feeds are how a user discovers and orders from a desired restaurant. The <a href="#">relational inventory schema</a> defines the structure of your Ordering End-to-End inventory. An Ordering End-to-End inventory feed consists of the following main entities:</p> <ul style="list-style-type: none"> <li>• <a href="#">Restaurant</a></li> <li>• <a href="#">Service</a></li> <li>• <a href="#">Menu</a></li> </ul>

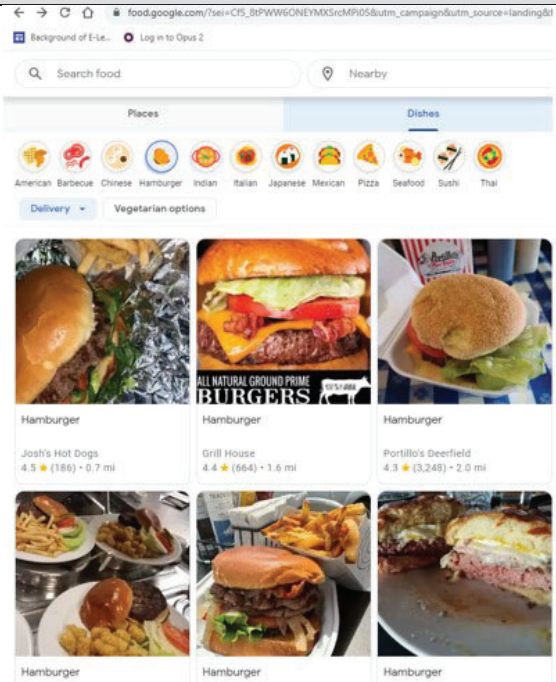
<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 12: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
		<p><b>Relational inventory schema</b>  <span data-bbox="1328 260 1451 287">Send feedback</span></p> <p>This page describes the format for the Ordering End-to-End data feeds (Food Catalog Specification) you provide to Google. For a machine-readable version of this information, you can <a href="#">download the JSON schema</a>.</p> <p><b>General requirements</b></p> <p>Entities must be structured to be on one line per entity in the feeds (entities are separated by newline characters). For readability purposes, JSON examples on this page do not follow that structure. However, you must follow that structure when sending your feeds. For example, a menu entity must be structured like the following code:</p> <pre data-bbox="592 533 992 554">{ "@type": "Menu", "name": "Coffee Shop A", "@id": "1535" }</pre> <p>Each 'Restaurant' entity may have two Service entities (one each for the 'DELIVERY' and 'TAKEOUT' service types). Each 'Service' entity may only have one 'Menu' entity.</p> <p>Any sub-entities can be reused across multiple restaurants.</p> <p><b>Generate client libraries</b>  <span data-bbox="1328 716 1451 743">Send feedback</span></p> <p>A machine-readable version of the Ordering End-to-End data feed and fulfillment API definitions are made available to generate client source code and validate the structure of JSON data. This allows you to spend more time in developing application capabilities and business logic required for the integration.</p> <p>In this example, we use <a href="#">quicktype</a> CLI to generate an easy-to-use client library.</p> <p><b>Download the JSON Schemas</b></p> <p>These machine-readable versions of data feeds and APIs are needed for code generation and validation.</p> <ul data-bbox="605 1003 821 1083" style="list-style-type: none"> <li>• <a href="#">Ordering End-to-End data feeds</a></li> <li>• <a href="#">Checkout and Submit Order API</a></li> <li>• <a href="#">Real-time updates</a></li> </ul> <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/overview">https://developers.google.com/actions-center/verticals/ordering/e2e/overview</a>)</p> <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/reference/feeds/relational-inventory-schema">https://developers.google.com/actions-center/verticals/ordering/e2e/reference/feeds/relational-inventory-schema</a>)</p> <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/reference/tools/generate-client-libraries">https://developers.google.com/actions-center/verticals/ordering/e2e/reference/tools/generate-client-libraries</a>)</p>

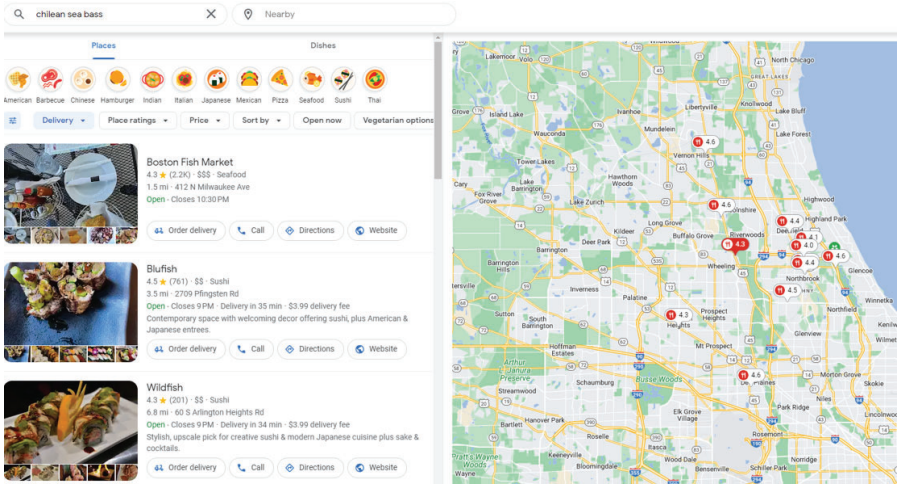
<p>[12g]</p>	<p>a mapping module configured to convert the raw files to a standardized format to provide formatted data;</p>	<p>Google Food comprises a mapping module configured to convert the raw files to a standardized format to provide formatted data. under the “Integration” step on Google Food Developer Site. Google Food maps the acquired raw files using relational inventory schema, as shown below:</p> <ul style="list-style-type: none"> <li> <b>Inventory feeds</b> <p>These feeds use a <a href="#">relational inventory schema (/actions-center/verticals/ordering/e2e/reference/feeds/relational-inventory-schema)</a> to supply Google with up-to-date information about a restaurant, the services it provides, and the items in its menu.</p> <p>Inventory feeds</p> <p>Inventory feeds are how a user discovers and orders from a desired restaurant. The <a href="#">relational inventory schema</a> defines the structure of your Ordering End-to-End inventory. An Ordering End-to-End inventory feed consists of the following main entities:</p> <ul style="list-style-type: none"> <li><a href="#">Restaurant</a></li> <li><a href="#">Service</a></li> <li><a href="#">Menu</a></li> </ul> <p>The mapping module is configured to convert the raw files to a standardized format to provide formatted data. The data is formatted for the Ordering End-to-End data feeds (Food Catalog Specification) using the JSON data schema.</p> <p>Relational inventory schema <span>🔍</span> <span>Send feedback</span></p> <p>This page describes the format for the Ordering End-to-End data feeds (Food Catalog Specification) you provide to Google. For a machine-readable version of this information, you can <a href="#">download the JSON schema</a>.</p> <p>General requirements</p> <p>Entities must be structured to be on one line per entity in the feeds (entities are separated by newline characters). For readability purposes, JSON examples on this page do not follow that structure. However, you must follow that structure when sending your feeds. For example, a menu entity must be structured like the following code:</p> <pre> {"@type": "Menu", "name": "Coffee Shop A", "id": "1535"}                     </pre> <p>Each 'Restaurant' entity may have two Service entities (one each for the 'DELIVERY' and 'TAKEOUT' service types). Each 'Service' entity may only have one 'Menu' entity.</p> <p>Any sub-entities can be reused across multiple restaurants.</p> </li> </ul>
--------------	---	--

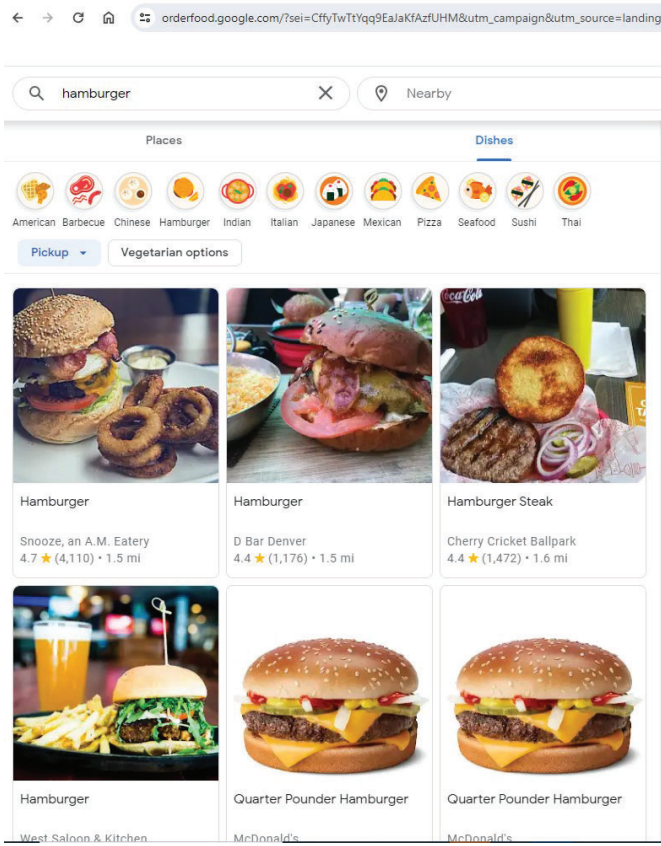
<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 12: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>																		
		<p>Restaurant data</p> <p>Restaurant (required)</p> <p>A required entity to implement. Describes a restaurant.</p> <p>The following table lists the properties for the Restaurant type:</p> <table border="1" data-bbox="609 436 1461 898"> <thead> <tr> <th>Property</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>@type</td> <td>Const</td> <td>Required. Value: Restaurant</td> </tr> <tr> <td>@id</td> <td>String</td> <td>Required. A unique identifier of the restaurant or delivery provider. Example: restaurant_1</td> </tr> <tr> <td>name</td> <td>String</td> <td>Required. Name of the restaurant. Example: Foo</td> </tr> <tr> <td>description</td> <td>String</td> <td>A description of the restaurant. Example: Best seafood in town</td> </tr> <tr> <td>url</td> <td>Url</td> <td>The URL that represents the restaurant. The restaurant domain is preferred over the aggregator domain. Example: http://www.provider.com/somerestaurant</td> </tr> </tbody> </table> <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/overview">https://developers.google.com/actions-center/verticals/ordering/e2e/overview</a>)</p>	Property	Type	Description	@type	Const	Required. Value: Restaurant	@id	String	Required. A unique identifier of the restaurant or delivery provider. Example: restaurant_1	name	String	Required. Name of the restaurant. Example: Foo	description	String	A description of the restaurant. Example: Best seafood in town	url	Url	The URL that represents the restaurant. The restaurant domain is preferred over the aggregator domain. Example: http://www.provider.com/somerestaurant
Property	Type	Description																		
@type	Const	Required. Value: Restaurant																		
@id	String	Required. A unique identifier of the restaurant or delivery provider. Example: restaurant_1																		
name	String	Required. Name of the restaurant. Example: Foo																		
description	String	A description of the restaurant. Example: Best seafood in town																		
url	Url	The URL that represents the restaurant. The restaurant domain is preferred over the aggregator domain. Example: http://www.provider.com/somerestaurant																		
[12h]	a linking module configured to perform record linkage on the formatted data according to the identification data that uniquely identifies sources; and	<p>Google Food comprises a linking module configured to perform record linkage on the formatted data according to the identification data that uniquely identifies sources under the “Integration” step on Google Food Developer Site for its Ordering End-to-End Integration System. Google Food links the formatted data by restaurant, service and menu items by using the “Inventory Feed” component of its Ordering End-to-End Integration system, as shown below from the Google Food Developer Site.</p> <p>Inventory feeds</p> <p>Inventory feeds are how a user discovers and orders from a desired restaurant. The relational inventory schema defines the structure of your Ordering End-to-End inventory. An Ordering End-to-End inventory feed consists of the following main entities:</p> <ul style="list-style-type: none"> <li>• Restaurant</li> <li>• Service</li> <li>• Menu</li> </ul> <p>Menu</p> <p>The Menu entity describes a menu that a restaurant offers to its customers. A Menu entity must be defined within a Restaurant entity through the Restaurant’s menuId property.</p> <p>Other objects that define the menu (such as description, image, add-ons, and nutrition information) include MenuItem and MenuItemOffer, as well as the optional objects MenuSection, Availability, and MenuItemOption.</p> <p>Menu sections</p> <p>You can use MenuSection objects to organize multiple MenuItem objects into logical categories. This approach is useful for restaurants that have multiple menus (like breakfast, lunch, and dinner). Add each menu as a separate MenuSection.</p> <p>In addition, Google Food offers merchant matching/linking services in order to ensure that inventory is shown to users in the right locations, under the Partner Portal of the Google Food Developer Site.:</p>																		

<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 12: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
		<p>Ordering End-to-End Actions Center Merchant Matching and Testing <span style="float: right; border: 1px solid #ccc; padding: 2px 5px; font-size: 0.8em;">Send feedback</span></p> <p>Match your inventory</p> <p>The Inventory Viewer is your entrypoint to view all of the inventory that has been submitted to the Actions Center along with its matching status.</p> <p>What is matching?</p> <p>Matching is how you ensure that inventory is shown to users in the right locations. A Match is the connection between the Match Input, information you provide that are used for matching (typically information like merchant name, address, geo coordinate, phone number, and url), and the Match Output (typically a Google listing such as a business profile).</p> <p>All inventory that shares the same Match Input also share the same Match Output. This means that when a match is changed for one piece of inventory, all other inventory that uses that Match Input is updated as well. When changing the match the Actions Center lists all the inventory that would be impacted by the change.</p> <p>Within the Inventory Viewer the Matched column displays the match status (yes or no) on each piece of inventory.</p> <div style="background-color: #e0f0ff; padding: 5px; border: 1px solid #add8e6;"> <p>★ <b>Note:</b> You can use filters, such as an entity ID or match status, to locate inventory.</p> </div> <p>See (<a href="https://developers.google.com/actions-center/verticals/ordering/e2e/partner-portal/inventory/merchant-matching">https://developers.google.com/actions-center/verticals/ordering/e2e/partner-portal/inventory/merchant-matching</a>)</p>
[12i]	a menu combining module configured to combine multiple source menus from linked sources into the master data set.	<p>Google Food comprises a menu combining module configured to combine multiple source menus from linked sources into the master data set as shown in the “Inventory Feeds” and “Relational Inventory Schema” sections under the “Integration” step on Google Food Developer Site for its Ordering End-to-End Integration System, as shown in the above-charts and also illustrated below.</p> <p>Inventory feeds</p> <p>Inventory feeds are how a user discovers and orders from a desired restaurant. The <i>relational inventory schema</i> defines the structure of your Ordering End-to-End inventory. An Ordering End-to-End inventory feed consists of the following main entities:</p> <ul style="list-style-type: none"> <li>• <a href="#">Restaurant</a></li> <li>• <a href="#">Service</a></li> <li>• <a href="#">Menu</a></li> </ul> <p>Batch feeds</p> <p>Google uses a daily batch feed of your inventory feeds to make Ordering End-to-End available to users. To keep your inventory up to date, you must update your batch feeds at least once a day for Google to fetch. It takes about two hours for your inventory to be updated by a batch.</p> <p>Incremental Updates API</p> <p>You can send time-sensitive updates of your inventory to Google. The Incremental Updates API lets you update and delete entities in your inventory in almost real time. Incremental updates are processed in no more than five minutes. This is primarily intended for updates that you can't foresee, such as emergency restaurant closures or removal of an out-of-stock item. If your change doesn't need to be reflected immediately, use the batch feeds instead.</p>

<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 12: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
		<p><b>Relational inventory schema</b> <span>📄</span> <span>Send feedback</span></p> <p>This page describes the format for the Ordering End-to-End data feeds (Food Catalog Specification) you provide to Google. For a machine-readable version of this information, you can <a href="#">download the JSON schema</a>.</p> <p><b>General requirements</b></p> <p>Entities must be structured to be on one line per entity in the feeds (entities are separated by newline characters). For readability purposes, JSON examples on this page do not follow that structure. However, you must follow that structure when sending your feeds. For example, a menu entity must be structured like the following code:</p> <pre> {"@type": "Menu", "name": "Coffee Shop A", "@id": "1535"}                     </pre> <p>Each 'Restaurant' entity may have two Service entities (one each for the 'DELIVERY' and 'TAKEOUT' service types). Each 'Service' entity may only have one 'Menu' entity.</p> <p>Any sub-entities can be reused across multiple restaurants.</p> <p><b>Menu</b></p> <p>The Menu entity describes a menu that a restaurant offers to its customers. A Menu entity must be defined within a Restaurant entity through the Restaurant's menuId property.</p> <p>Other objects that define the menu (such as description, image, add-ons, and nutrition information) include MenuItem and MenuItemOffer, as well as the optional objects MenuSection, Availability, and MenuItemOption.</p> <p><b>Menu sections</b></p> <p>You can use MenuSection objects to organize multiple MenuItem objects into logical categories. This approach is useful for restaurants that have multiple menus (like breakfast, lunch, and dinner). Add each menu as a separate MenuSection.</p> <p>For instance, the menu combining module in Google Food is configured to combine multiple source menus from linked sources into the master data set — to combine multiple source menus from linked restaurants into the master data set — in order to (a) allow a user to search food and menus for delivery from its member restaurants in a given location, including by common menu item, and with an “Order Delivery” button for “chilean sea bass” and/or (b) group restaurants by restaurant type, as shown below for “hamburger”:</p>  <p>The screenshot shows a Google Food search interface. The search bar contains 'chilean sea bass'. Below the search bar, there are tabs for 'Places' and 'Dishes'. Under 'Places', there are icons for various cuisines: American Barbecue, Chinese, Hamburger, Indian, Italian, Japanese, Mexican, Pizza, Seafood, Sushi, and Thai. There are also filters for 'Delivery', 'Place ratings', 'Price', 'Sort by', 'Open now', and 'Vegetarian options'. Three restaurant listings are visible: Boston Fish Market (4.3 rating, 2.2K reviews, Seafood), Bluffish (4.5 rating, 761 reviews, \$5-\$8, Sushi), and Wildfish (4.3 rating, 201 reviews, \$5-\$8, Sushi). To the right of the listings is a map showing the search area with red location pins and ratings for various restaurants.</p>

<u>ELEMENT(S)</u>	<u>INDEPENDENT CLAIM 12: 11,038,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
		 <p>The screenshot shows a Google Food search interface. At the top, there is a search bar with the text "Search food" and a "Nearby" filter. Below the search bar, there are tabs for "Places" and "Dishes". Under the "Dishes" tab, there is a row of food category icons including American Barbecue, Chinese, Hamburger, Indian, Italian, Japanese, Mexican, Pizza, Seafood, Sushi, and Thai. Below the icons, there are filters for "Delivery" and "Vegetarian options". The main content area displays a grid of six hamburger listings, each with a photo of the burger and the restaurant name. The listings are: Josh's Hot Dogs (4.5 stars, 186 reviews, 0.7 mi), Grill House (4.4 stars, 664 reviews, 1.6 mi), Portillo's Deerfield (4.3 stars, 3,248 reviews, 2.0 mi), and three other unnamed listings.</p>

<u>ELEMENT(S)</u>	<u>DEPENDENT CLAIM 13: 11,037,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
[13a]	The system of claim 12, wherein the website database comprises the master data set in a searchable format.	<p>See above, claim chart for independent claim 1, showing the website database comprises the master data set in a searchable format, as shown below.</p>  <p>The screenshot displays a search for 'chilean sea bass' on the Google Food platform. The search results are organized into two main sections: 'Places' and 'Dishes'. The 'Places' section lists three restaurants: Boston Fish Market (4.3 rating, \$2.2K-\$55, Seafood), Blufish (4.5 rating, \$8-\$20, Sushi), and Wildfish (4.3 rating, \$5-\$20, Sushi). Each listing includes a photo, address, hours, and options for delivery, call, directions, and website. The 'Dishes' section shows various food items with icons. To the right, a map of the Chicago area shows the locations of these restaurants marked with red pins and their respective ratings.</p>

<u>ELEMENT(S)</u>	<u>DEPENDENT CLAIM 14: 11,037,090</u>	<u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u>
[14a]	The system of claim 12, further comprising: a data warehouse configured to store the provided master data set in a searchable format; wherein the data warehouse is accessible by the website database to receive data.	<p>See above, claim chart for independent claim 1, showing data acquisition which is, upon information and belief, stored in a data warehouse (data storage) configured to store the provided master data set in a searchable format; wherein the data warehouse is accessible by the website database to receive data. The data is searchable as shown below.</p>  <p>The screenshot shows a Google Food search for 'hamburger'. The search results are categorized under 'Dishes' and include a variety of options such as American Barbecue, Chinese, Hamburger, Indian, Italian, Japanese, Mexican, Pizza, Seafood, Sushi, and Thai. There are filters for 'Pickup' and 'Vegetarian options'. The results display several items with images and restaurant information:</p> <ul style="list-style-type: none"> <li><b>Hamburger</b> at Snooze, an A.M. Eatery (4.7 stars, 4,110 reviews, 1.5 mi)</li> <li><b>Hamburger</b> at D Bar Denver (4.4 stars, 1,176 reviews, 1.5 mi)</li> <li><b>Hamburger Steak</b> at Cherry Cricket Ballpark (4.4 stars, 1,472 reviews, 1.6 mi)</li> <li><b>Hamburger</b> at West Saloon &amp; Kitchen</li> <li><b>Quarter Pounder Hamburger</b> at McDonald's</li> <li><b>Quarter Pounder Hamburger</b> at McDonald's</li> </ul>

<b><u>INDEPENDENT CLAIM 15: 11,037,090</u></b>	<b><u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u></b>
An apparatus for providing a searchable aggregated data structure for a networked application, the apparatus comprising:	See, above claim charts for independent claims 1 and 12, showing Google Food constitute an apparatus for providing a searchable aggregated data structure for a networked application. See, e.g., elements [1a] and [1i] in claim chart 1.
a processor;	See, above claim charts for independent claims 1 and 12. See, e.g., elements [1b], [1e] – [1i] and [12b] and [12c] in claim charts 1 and 12.
a memory accessible by the processor; and	See, above claim charts for independent claims 1 and 12. See, e.g., element [12b] in claim chart 12.
a set of computer-readable instructions stored on a non-transitory medium accessible by the processor, the instructions being executable by the processor to perform a method comprising:	See, above claim charts for independent claims 1 and 12. See, e.g., elements [12b] and [12d] in claim chart 12.
acquiring source data from a plurality of delivery service computers associated with a plurality of food or beverage delivery services over a communication network, the acquired source data being in a plurality of formats, where the acquired source data includes, for each of the plurality of food or beverage delivery services, data representing multiple source menu items, and wherein said acquiring source data comprises one or more of:	See, above claim charts for independent claims 1 and 12. See, e.g., elements [1b], [12b] and [12f] in claim charts 1 and 12.
employing an application programming	See, above claim charts for independent claims 1 and 12. See, e.g., elements [1c] and [12e] in claim charts 1 and 12.

<b><u>INDEPENDENT CLAIM 15:</u></b> <b><u>11,037,090</u></b>	<b><u>ACCUSED PRODUCTS/SERVICES/GOOGLE FOOD</u></b>
interface to interface with the plurality of delivery service computers, or	
scraping data from the plurality of delivery service computers;	See, above claim charts for independent claims 1 and 12. See, e.g., elements [1d] and [12f] in claim charts 1 and 12.
mapping the acquired source data according to a predetermined data format to provide formatted data;	See, above claim charts for independent claims 1 and 12. See, e.g., elements [1e], [3a] and [12g] in claim charts 1, 3 and 12.
linking the formatted data to common sources of the source menu items represented by source identification data such that at least one food or beverage delivery service is linked to each common source and its source menu items;	See, above claim charts for independent claims 1 and 12. See, e.g., elements [1f], [5a] and [12f] in claim charts 1, 5 and 12.
identifying common menu items among the source menu items in the formatted data, and, for each identified common menu item, associating the source menu items with a master menu item;	See, above claim charts for independent claims 1 and 12. See, e.g., elements [1f] and [1g] in claim chart 1.
combining the linked data and the master menu items into a master data set; and	See, above claim charts for independent claims 1 and 12. See, e.g., elements [1h] and [12i] in claim charts 1 and 12.
importing the master data set and the source identification data into the searchable aggregated data structure.	See, above claim charts for independent claims 1 and 12. See, e.g., element [1i] in claim chart 1.

