

DOCKET NO.: 2220023-00125US2

Filed on behalf of Shopify Inc.

By: Daniel Williams, Reg. No. 45,221 (Lead Counsel)
Gregory Lantier (*pro hac vice* to be filed) (Backup Counsel)
Trishan Efram, Reg. No. 74,075 (Backup Counsel)
Liv Herriot (*pro hac vice* to be filed) (Backup Counsel)
R. Gregory Israelsen, Reg. No. 72,805 (Backup Counsel)
Wilmer Cutler Pickering Hale and Dorr LLP
2100 Pennsylvania Avenue, NW
Washington, DC 20037
Email: daniel.williams@wilmerhale.com
gregory.lantier@wilmerhale.com
trishan.esram@wilmerhale.com
liv.herriot@wilmerhale.com
greg.israelsen@wilmerhale.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SHOPIFY INC.
Petitioner

v.

DKR CONSULTING LLC
Patent Owner

IPR2025-00131
U.S. Patent No. 10,846,785

**PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 10,846,785
CHALLENGING CLAIMS 1–20
UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104**

TABLE OF CONTENTS

	Page
I. MANDATORY NOTICES	1
A. Real Party-in-Interest	1
B. Related Matters.....	1
C. Counsel	3
D. Service Information	3
II. CERTIFICATION OF GROUNDS FOR STANDING	4
III. OVERVIEW OF CHALLENGE AND RELIEF REQUESTED.....	4
A. Prior Art.....	4
B. Ground.....	5
IV. OVERVIEW OF THE '785 PATENT	5
A. Alleged Invention	5
B. Prosecution History	7
C. Overlooking the Prior Art was Material Error	8
V. PRIOR ART PATENTS AND PUBLICATIONS	10
A. Dierks	10
B. Johnston.....	14
VI. CLAIM CONSTRUCTION	16
VII. LEVEL OF ORDINARY SKILL	16
VIII. SPECIFIC GROUNDS FOR UNPATENTABILITY	16
A. Ground 1: Claims 1–20 are Obvious over Dierks in view of Johnston	17
1. Independent Claim 1	17
2. Claim 2	60
3. Claim 3	62
4. Claim 4	65
5. Claim 5	67
6. Claim 6	67

Petition for *Inter Partes Review*
of U.S. Patent No. 10,846,785

7.	Independent Claim 7	68
8.	Claim 8	77
9.	Claim 9	77
10.	Claim 10	79
11.	Claim 11	81
12.	Claim 12	81
13.	Claim 13	81
14.	Independent Claim 14	82
15.	Claim 15	85
16.	Claim 16	86
17.	Claim 17	87
18.	Claim 18	87
19.	Claim 19	87
20.	Claim 20	88
IX.	CONCLUSION.....	90
X.	APPENDIX OF CLAIM LISTING.....	i

Shopify Inc. (“Petitioner” or “Shopify”) respectfully requests *inter partes* review (“IPR”) of claims 1–20 of U.S. Patent (“USP”) No. 10,846,785 (the “’785 patent”) (EX1001) pursuant to 35 U.S.C. §§311–19 and 37 C.F.R. §42.1 *et seq.*

I. MANDATORY NOTICES

A. Real Party-in-Interest

Under 37 C.F.R. §42.8(b)(1), Petitioner certifies that Shopify Inc. is the real party-in-interest.

B. Related Matters

The ’785 patent is a continuation of application No. 16/512,018, now USP No. 11,455,678 (the “’678 patent”), which is a continuation of 15/591,208, now USP No. 10,395,304, which is a continuation of application No. 13/571,293, now USP No. 9,679,298, assigned to DKR Consulting LLC (“Patent Owner” or “PO”), which is a continuation-in-part of application No. 12/852,090, now abandoned.

Shopify is concurrently filing a petition for IPR against the ’678 patent, and USP Nos. 11,157,995 (the “’995 patent”) and 11,488,237 (the “’237 patent”). The ’785 and ’995 patents both claim priority to the ’678 patent, and the ’237 patent claims priority to the ’995 patent.

As of the filing date of this petition, Shopify knows of the following co-pending litigation involving the ’785 patent, in which PO asserts it is the owner: *DKR Consulting LLC v. Shopify Inc.*, No. 2:23-cv-6904 (C.D. Cal.); *DKR*

Consulting, LLC v. Shopify Inc., No. 2024-2279 (Fed. Cir.). The earliest date of service was December 5, 2023. A Motion to Dismiss was granted on August 1, 2024, finding all claims of the '785 patent to be ineligible under 35 USC §101. EX1025, 14 (“In short, the Asserted Patents do not claim patent-eligible subject matter under Section 101 and the Alice test.”); 10 (“In sum, all the claims in these patents describe inventions that are substantially similar and linked to the same idea. The Court therefore finds that Claim 1 of each of the Asserted Patents is representative of the other claims within those patents.”). On August 26, 2024, PO noticed its appeal of the district court’s dismissal order and final judgment. EX1026. The Court of Appeals for the Federal Circuit issued a Notice of Docketing on August 30, 2024. EX1030. On October 29, 2024, PO filed its opening brief. EX1031.

The '678 patent is currently under reexamination, Control No. 90/019,476. The request for *ex parte* reexamination was filed by Unified Patents, LLC on April 9, 2024. EX1028. On May 16, 2024, the Patent Office ordered reexamination, finding a substantial new question of patentability as to claims 1–6 and 8–20 of the '678 patent. A non-final Office Action was issued on September 18, 2024. EX1029. Patent Owner petitioned for an extension of time to respond to the Office Action. The petition was dismissed, and Patent Owner failed to file a Response by the November 18, 2024 due date. EX1033.

C. Counsel

Under 37 C.F.R. §§42.8(b)(3)–(4), Shopify identifies the following lead and backup counsel, to whom all correspondence should be directed.

Lead Counsel:	Daniel Williams (Reg. No. 45,221)
Backup Counsel:	Gregory Lantier (<i>pro hac vice</i> to be filed)
	Trishan Efram (Reg. No. 74,075)
	Liv Herriot (<i>pro hac vice</i> to be filed)
	R. Gregory Israelsen (Reg. No. 72,805)

D. Service Information

E-mail:	daniel.williams@wilmerhale.com
	gregory.lantier@wilmerhale.com
	trishan.esram@wilmerhale.com
	liv.herriot@wilmerhale.com
	greg.israelsen@wilmerhale.com
Post and hand delivery:	Wilmer Cutler Pickering Hale and Dorr LLP
	2100 Pennsylvania Avenue, NW
	Washington, DC 20037
	Telephone: 202-663-6000
	Fax: 202-663-6363

Shopify consents to service by e-mail on lead and backup counsel.

II. CERTIFICATION OF GROUNDS FOR STANDING

Shopify certifies under 37 C.F.R. §42.104(a) that the patent for which review is sought is available for *inter partes* review and under 37 C.F.R. §§42.101(a)–(c) that Shopify is not barred or estopped from requesting an IPR challenging the patent claims on the grounds identified in this Petition.

III. OVERVIEW OF CHALLENGE AND RELIEF REQUESTED

Shopify requests review of claims 1–20 of the '785 patent and cancellation of these claims as unpatentable over the prior art.

A. Prior Art

The earliest claimed priority date of the '785 patent is August 6, 2010. EX1001 at 2. Shopify relies on the documents in the Table of Exhibits, including:

- USP Application Publication No. 2008/0183593 (“Dierks”) (EX1004), published on July 31, 2008, as §102(b) prior art¹;
- USP Application Publication No. 2010/0114739 to Johnston (EX1005), filed on September 2, 2009 and published on May 6, 2010, as §§102(a), 102(e) prior art.

¹ Johnston is entitled to its September 3, 2008 provisional application (No. 61/093,917) filing date, as explained by Dr. Mowry. EX1002, ¶¶54–64.

B. Ground

Under 37 C.F.R. §§42.22(a)(1) and 42.104(b)(1)–(2), Shopify requests cancellation of claims 1–20 of the '785 patent on the following ground:

Ground	Basis	Claims	References
1	§103	1–20	Dierks and Johnston

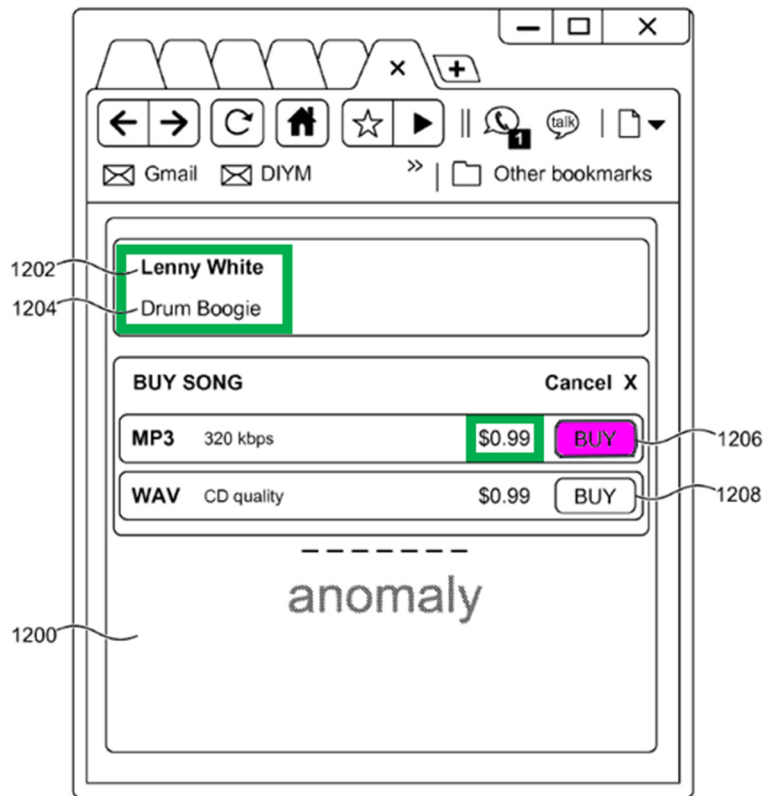
This Petition, supported by Dr. Mowry's declaration (EX1002), demonstrates that there is a reasonable likelihood Shopify will prevail with respect to cancellation of at least one of the challenged claims. *See* 35 U.S.C. §314(a).

IV. OVERVIEW OF THE '785 PATENT

A. Alleged Invention

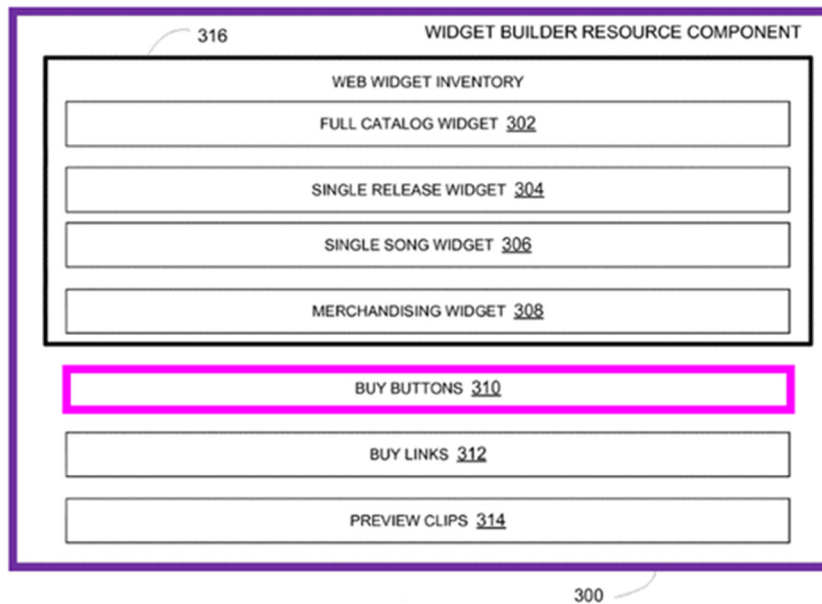
The '785 patent claims describe a system and method for generating and embedding a buy button into a web site. EX1001, claims 1, 7, and 14. Independent claims 1, 7, and 14 are similar. To the extent claims 7 and 14 include features that differ from claim 1, these features are taught by the prior art, as discussed in the grounds. *Id.* EX1002, ¶33.

Figure 12 of the '785 patent below illustrates an embedded **buy button 1206** used to purchase a song 1204 called "Drum Boogie." EX1001, 17:42–49. Figure 12 includes **information about the song**, such as name, artist, and price. *Id.* After clicking the **buy button 1206**, a checkout screen is presented. *Id.*, Fig. 13, 17:50–61. EX1002, ¶34.



EX1001, FIG. 12

The '785 patent discloses that **buy buttons** are generated using a “**widget builder resource component 300**.” EX1001, 8:40–43. As represented by Figure 3 below, the '785 patent indicates that the “**widget builder resource component 300** is used to create an inventory of ‘web widgets’ 316, **buy buttons 310**, buy links 312 and preview clips 314.” *Id.*, 7:67–8:2. EX1002, ¶35.



EX1001, FIG. 3

As the ground herein demonstrates, generating and using buy buttons was well-known in the art before the '785 patent's priority date. EX1002, ¶¶36–37.

B. Prosecution History

One Office Action issued rejecting all claims under (1) pre-AIA 35 U.S.C. §103(a) as unpatentable over USP Number 7,526,545 (“Jerome”), and (2) provisional double patenting as unpatentable over claims 8–27 of USP Application 16/512,018. EX1009, 121–127.

Applicant responded by amending independent claims 1, 7 and 14. *Id.*, 161–175. Applicant provided remarks equating the claimed “provider” to a “multimedia content distribution system 200” where the provider “executes both the embeddable buy button builder 204, as well as a transaction payment processing service 214.”

Id., 171. The payment processing system 308 of Dierks provides these same button-building and transaction payment processing services.

Applicant also argued that “*Jerome* does not disclose a system having a first server operated by an electronic commerce provider that allows the creation of embeddable buy buttons **and** where the same electronic commerce provider facilitates payment processing.” *Id.*, 172 (emphasis in original). Dierks likewise discloses these features.

C. Overlooking the Prior Art was Material Error

Johnston and USP No. 9,721,290, which is related² to Dierks, were of record in the ’785 patent’s prosecution, but the Examiner did not mention, describe, address, or apply Johnston and the ’290 patent—alone or in combination—substantively.

“If the record of the Office’s previous consideration of the art is not well-developed or *silent*, then a petitioner may show the Office erred by overlooking something persuasive. ...” *Advanced Bionics, LLC v. MED-EL Elektromedizinische Gerate GmbH*, IPR2019-01469, Paper 6 at 10 (P.T.A.B. Feb. 13, 2020)³; *see also*

² The ’290 patent is the last in a chain of continuation and divisional applications stemming from Dierks.

³ All emphasis added herein unless otherwise stated.

Amazon.com, Inc., v. M2M Solutions LLC, IPR2019-01205, Paper 14 at 16 (P.T.A.B. Jan. 27, 2020) (“The Board frequently holds that a reference that ‘was neither applied against the claims nor discussed by the Examiner’ does not weigh in favor of exercising the Board’s discretion under § 325(d) to deny a petition.”).

Dierks in view of Johnston discloses or renders obvious the precise limitations that the Examiner identified as not being taught by the closest prior art references (*see* EX1009, 198), demonstrating that the challenged claims are unpatentable. By overlooking specific teachings of Johnston and Dierks in allowing the challenged claims, the Examiner committed a material error even if PO may argue that the Examiner considered Johnston and Dierks. *See* 35 U.S.C. §325(d); *Advanced Bionics*, Paper 6 at 10.

The Board has instituted review under these exact circumstances, and Shopify respectfully requests that it do so again here. *See, e.g., Mylan Pharm. Inc., v. Merck Sharp & Dohme Corp.*, No. IPR2020-00040, Paper 21 at 18 (P.T.A.B. May 12, 2020) (instituting review even though “WO ’498 was presented to the Examiner” because the “Examiner never discussed WO ’498 or made a rejection based on it” and “gave no reasons for allowance” even though the “disclosure meets the claimed invention” and “[t]he likeliest explanation ... is that the Examiner simply overlooked [its] teachings”); *Roku, Inc. v. Universal Elecs., Inc.*, No. IPR2019-01615, Paper 12 at 12–13 (P.T.A.B. Apr. 17, 2020) (“Examiner erred by overlooking the specific

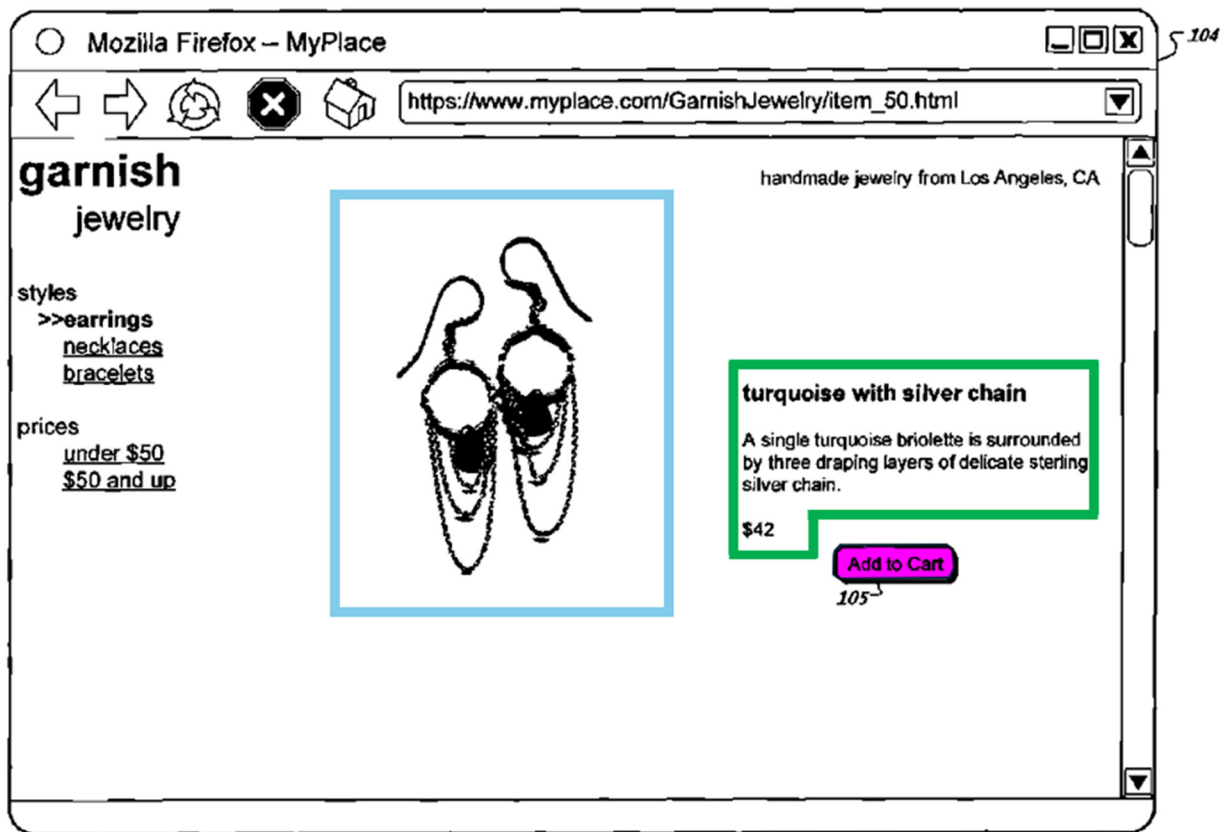
teaching of Chardon” because it taught the precise limitations that the Examiner had concluded were “not disclosed, taught, or suggested in the references applied” in allowing the claims); *see also Advanced Bionics*, Paper 6 at n. 9 (“An example of a material error may include misapprehending or overlooking specific teachings of the relevant prior art where those teachings impact patentability of the challenged claims.”).

V. PRIOR ART PATENTS AND PUBLICATIONS

The following references are pertinent to the grounds of unpatentability:

A. Dierks

Similar to the ’785 patent, Dierks discloses an “**add to cart**” button 105 (“buy button”) that a user clicks to buy the item, as shown below in Figure 1B. EX1004, ¶¶36, 37, 73. Figure 1B also shows a **textual description** and an **image representation** of the item. *Id.*, ¶36. EX1002, ¶44.



EX1004, FIG. 1B

The “**add to cart**” button 105 is analogous to the “buy button” of the ’785 patent, as described with additional detail by Dr. Mowry. For both, pressing the button leads the user to a display showing the total cost of the item. EX1001, 17:50-57; EX1004, ¶43. The user then presses an additional button to access a page for entering payment information. EX1001, 17:62–18:1; EX1004, ¶¶44, 48. EX1002, ¶¶68–72.

To create an “add to cart” button, Figure 4 of Dierks below discloses an interface for the user to enter **descriptive information** about the item for sale.

EX1004, ¶73. The user then clicks the “**Create Button Code.**” *Id.*, ¶74. EX1002, ¶45.

Mozilla Firefox - MyCheckout

https://www.mycheckout.com/Merchant/8675309/CreateInfoCode/

1. Choose a button type.

Basic Button

Button with Options

2. Choose a button style.

My web page has a White background.

3. Describe your item.

Item Name: turquoise earrings

Item Description: A single turquoise briolette is surrounded by three draping layers of delicate sterling silver chain.

Price: \$50.00 USD

Create Button Code

Insert this code for shopping cart:

```
<script type="text/javascript" src="https://www.mycheckout.com/shoppingcart.js?sellerId=8675309"></script><span id="mycheckout_status_bar"></span>
```

Insert this code for add to cart button:

```
<form action="https://www.mycheckout.com/Merchant/8675309/add" onSubmit="return false;" method="post"><input type="hidden" name="item" value="PD94bVWgdmVyc2lvbj0iMS4wIjBibmNvZGluZz0iVVRGLTglPz4NCjxJdGVtIHRhbnRwOj8vY2hiY2tvdXQuZ29vZ2x1LmNvbS9zY2h1bWEvMil+DQogIDx0dWFudGI0eT4xPC9xdWFudGI0eT4NCiAgPHVuaXQ
```

EX1004, FIG. 4

After the “Create Button Code” button is clicked, an **HTML Generator 312**, shown below in Figure 3, generates code that can be embedded into a web page to display the “**add to cart**” button, including the **description of the item**, shown above in Figure 1B. EX1004, ¶62, Fig. 3; *see also id.*, ¶¶36, 37, Fig. 4. EX1002, ¶46.

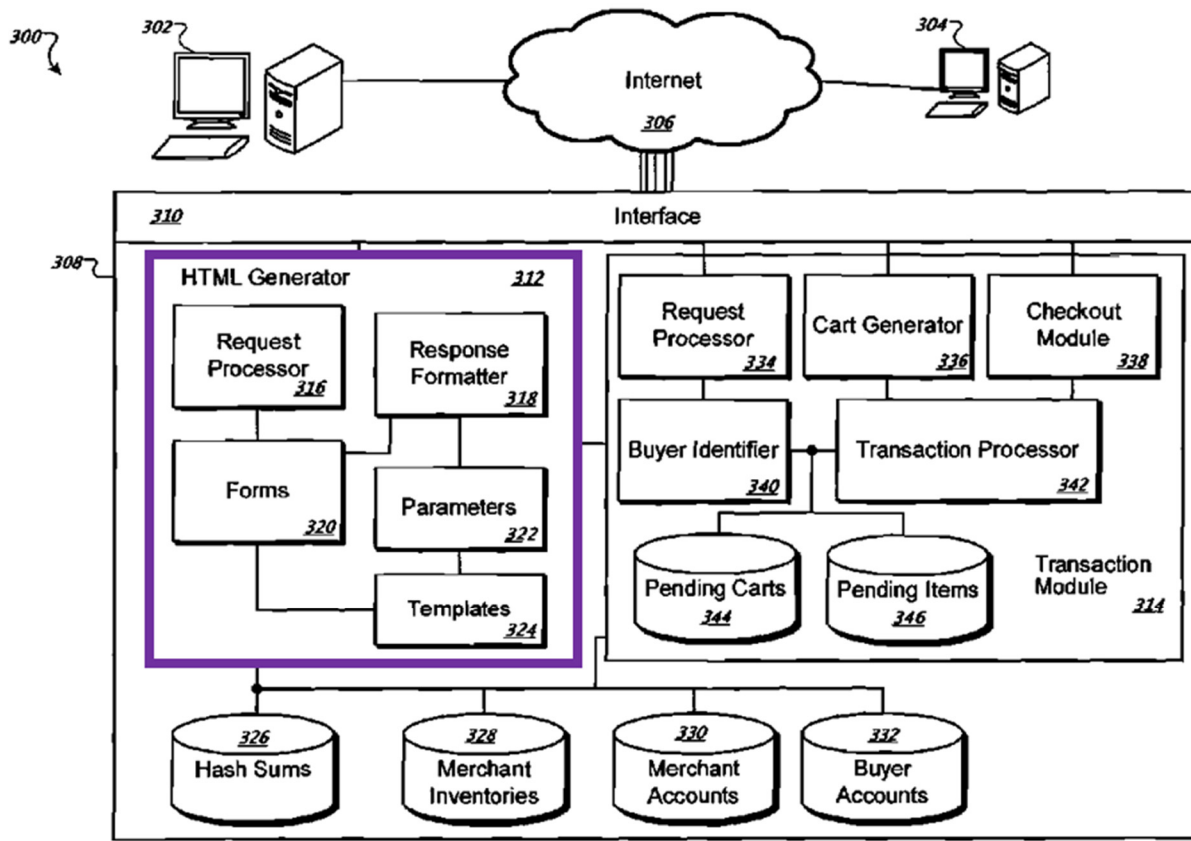


FIG. 3

EX1004, FIG. 3

Accordingly, similar to the '785 patent, Dierks discloses to generate and embed buy buttons. EX1002, ¶¶45–46.

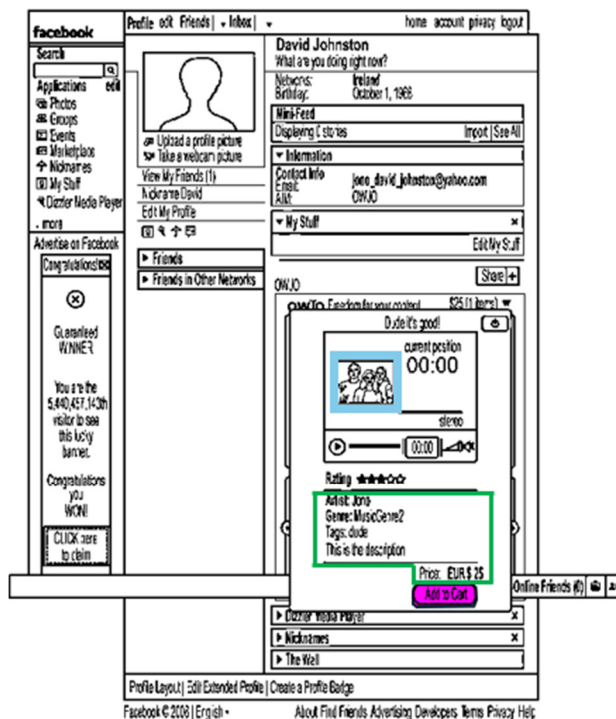
B. Johnston

Similar to Dierks, Johnston provides an interface for a user to enter product information for the purpose of generating an Internet Storefront to sell an item. EX1005, Figs. 3A–3F, ¶¶225–230. Figure 3B of Johnston below illustrates an interface for sellers to provide a **description of the item** (e.g., music) to be sold, including the price. *Id.*, ¶226. The interface prompts the seller to select an image that represents the item for sale. As shown in Figure 3B, the interface prompts the seller to “**provide a thumbnail file** that we will display on the OWJO Widget.” *Id.*, Fig. 3B; *see also id.*, ¶226. EX1002, ¶¶56–57.

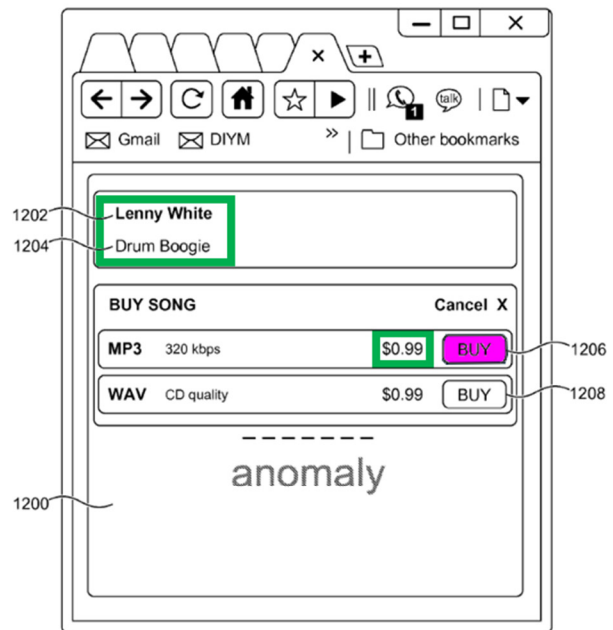
The screenshot shows a web browser window titled "Welcome CPA (Log out.)" with navigation links for Home, About us, and Site map. The main content area is titled "OWJO Freedom for your content" and includes a sidebar with "My Stuff" and "Featured" sections. The "Upload Content" section is active, showing tabs for music, videos, images, podcasts, and vodcasts. The "Music file Settings" form is highlighted with a green border and contains fields for Name of track, Artist, Tags, Genre, and a dropdown for Pop. Below this is a "Preview track" section with a "Browse..." button. The "Music file" section is highlighted with a blue border and contains a "Browse..." button. The "Thumbnail" section is highlighted with a blue border and contains a "Browse..." button. The "Set your price" section is highlighted with a green border and contains a dropdown for EUR and a text input field. An "upload" button is at the bottom right.

EX1005, FIG. 3B

Figure 4B below shows the generated Storefront including the **description of the item** and an “**Add to Cart**” button. This is similar to the product listing that the ’785 patent illustrates in Figure 12 (reproduced below alongside Johnston’s Figure 4B). EX1002, ¶58.



EX1005, FIG. 4B



EX1001, FIG. 12

As shown in Figure 4B of Johnston, the Storefront displays the **image representation** of the item for sale. EX1005, ¶¶222, 289. The **image representation** of the item for sale in Johnston is analogous to the image representation of the item for sale, e.g., earrings, discussed above in Dierks, since they both provide an image representation of an item. EX1002, ¶59.

VI. CLAIM CONSTRUCTION

The challenged claims of the '785 patent are construed “using the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. §282(b).” 37 C.F.R. §42.100(b) (Nov. 13, 2018). Each claim term should be given its plain and customary meaning as understood by one of ordinary skill in the art, in accordance with *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). If PO, seeking to avoid the prior art, offers a specific construction or interpretation, Shopify reserves the right to respond.

VII. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art (“POSITA”) at the time of the earliest claimed priority date of the '785 patent (August 6, 2010) would have had at least the equivalent of a bachelor’s degree in computer science, computer engineering, or a related subject and two or more years of experience in the field of distributed systems, transaction processing, and web-based user interfaces. Less work experience may be compensated by a higher level of education, such as a master’s degree, and vice versa. EX1002, ¶29.

VIII. SPECIFIC GROUNDS FOR UNPATENTABILITY

Under 37 C.F.R. §42.104(b)(4)–(5), the following sections (as confirmed in Dr. Mowry’s declaration, EX1002, ¶¶65–282) detail the grounds of unpatentability,

the limitations of challenged claims 1–20 of the ’785 patent, and how these claims were therefore obvious in view of the prior art.

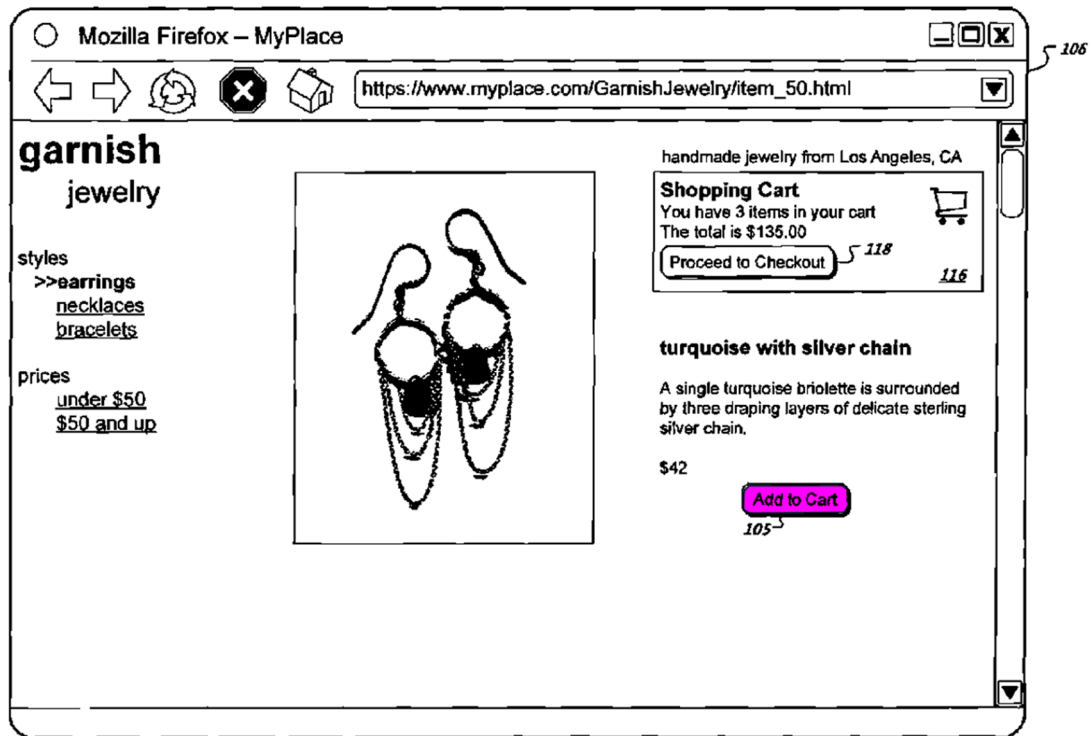
A. Ground 1: Claims 1–20 are Obvious over Dierks in view of Johnston

1. Independent Claim 1

a) 1[preamble]: “A method for embedding a buy button for an electronic commerce store, comprising:”

Dierks discloses the preamble of claim 1. EX1002, ¶73.

Figure 1B of Dierks shows a web site (“*an electronic store*”) for selling a product using an embedded “**add to cart**” button 105 (“*a buy button*”). EX1004, ¶36 (““**add to cart**’ button 105 that a purchaser may select to purchase a unit of the product.”). Dierks also discloses that the electronic store includes a checkout screen, as discussed below for element 1[g]. EX1002, ¶66.



EX1004, FIG. 1B

Dierks discloses a method for embedding the “add to cart” button 105. EX1004, ¶8 (“a computer-implemented method is disclosed.”). The “add to cart” button 105 of Dierks is analogous to the “buy button” of the ’785 patent. For both, pressing the button leads the user to a display showing the total cost of the item. The user then presses another button that brings the user to a further web page for entering payment related information. EX1002, ¶67.

In particular, after the user presses the “add to cart” button 105 in Dierks, the user is shown a shopping cart that displays the total cost of the item, as discussed below in more detail for element 1[f]. The user then presses a “proceed to checkout” button 118 that brings the user to a screen for providing payment-related

information, as discussed below for element 1[g]. A similar process occurs in the '785 patent. EX1002, ¶¶68–72.

b) 1[a]: “receiving, by an embeddable buy button builder executing on a first server operated by a buy button provider, a selection of an item from an item inventory and a representation of the item, wherein the item is to be listed for sale on the electronic commerce store; and”

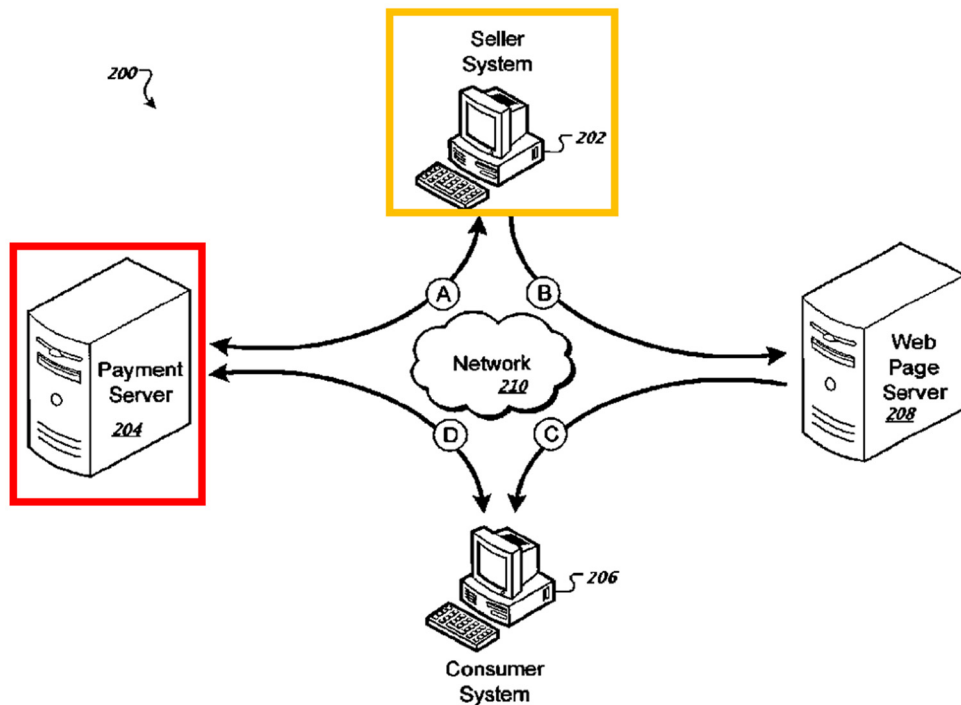
The combination of Dierks and Johnston renders obvious element 1[a]. EX1002, ¶74.

(1) “an embeddable buy button builder executing on a first server operated by a buy button provider”

Dierks discloses “an embeddable buy button builder executing on a first server operated by a buy button provider.” *Id.*, ¶75.

Dierks discloses a **payment server 204** (“a first server”) that receives information from a **seller system 202**. EX1004, Fig. 2, ¶53 (“The **seller system 202** may create a web page and ... register the seller with a **payment server 204** such as payment server 114⁴.”). EX1002, ¶76.

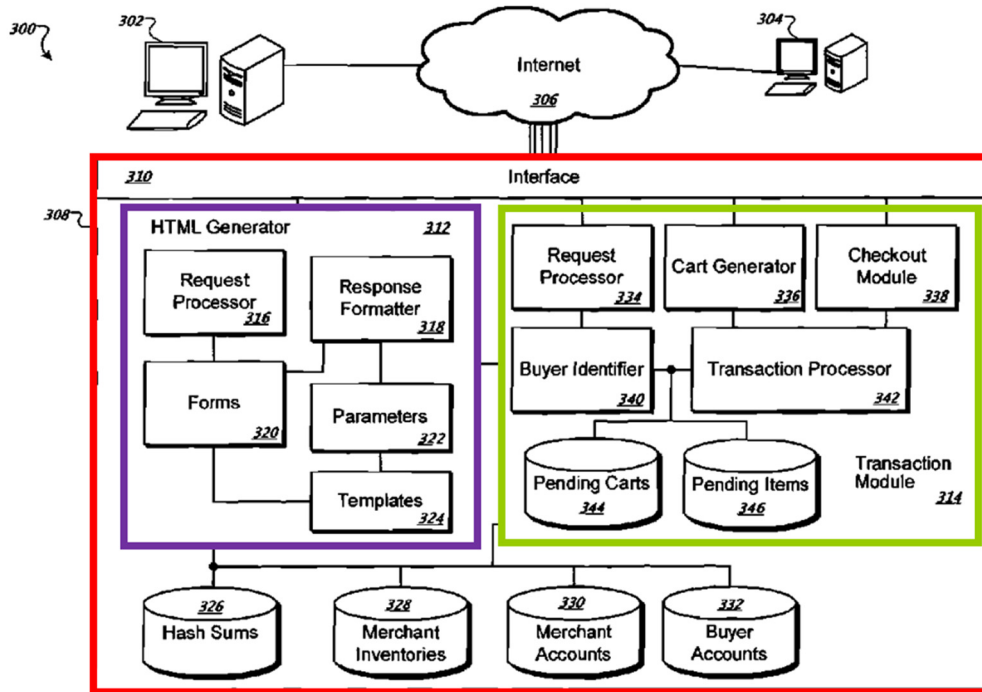
⁴ Disclosure regarding the payment server 114 applies to the payment processing system 308, as Dr. Mowry explains in his declaration. EX1002, ¶¶48–53.



EX1004, FIG. 2

Figure 3 of Dierks, below, discloses additional details of the payment server 204 with reference to the **payment processing system 308**. Like the payment server 204, Dierks discloses that the payment processing system 308 comprises a “server.” EX1004, ¶62 (“The payment processing system 308 communicates with the outside world using an interface 310, which may take the form of **one** or more **web servers** and other such hardware, or may take other appropriate forms.”). *Id.*, ¶62. A POSITA would have understood that Dierks’s reference to “**one** or more **web servers**” is pertaining to the payment processing system 308 and not the “interface” specifically. A network interface is a point where pieces of equipment or protocol layers connect, which may be in the form of a Network Interface Card. Accordingly,

a POSITA would have been taught that the payment processing system 308 of Dierks comprises at least one server (“a first server”) that includes the **HTML Generator 312** and **Transaction Module 314**, shown below. EX1002, ¶77.



EX1004, FIG. 3

The **HTML Generator 312** discloses the claimed “*embeddable buy button builder*.” As discussed below in more detail for element 1[b], the HTML Generator 312 generates an “add to cart” button (“*embeddable buy button*”) and therefore discloses the claimed embeddable buy button builder. The HTML Generator 312 is “*executing on*” the server, as claimed, because the payment processing system 308 includes the HTML Generator 312. EX1002, ¶78.

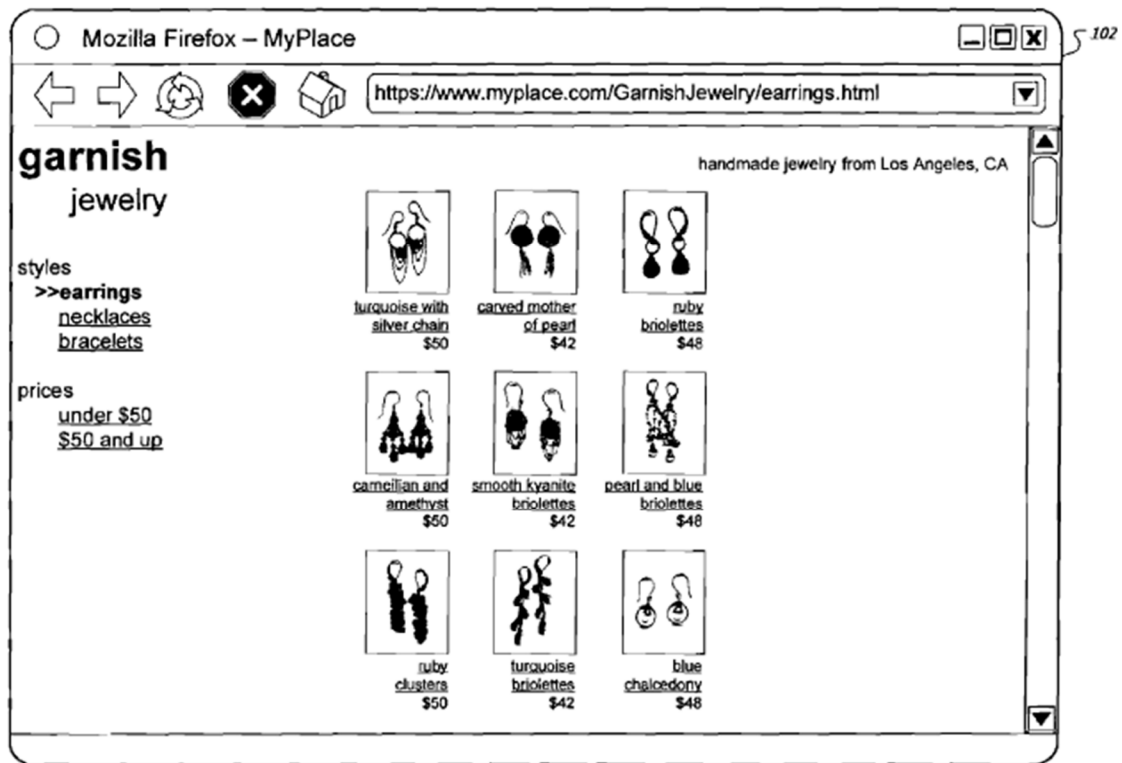
The server on which the **HTML Generator 312** (“*the embeddable buy button builder*”) executes is operated by the **payment processing system 308** (“*a buy button provider.*”). As described in Section IV.B, applicant equated a “provider” to a “multimedia content distribution system 200” where the provider “executes both the embeddable buy button builder 204, as well as a transaction payment processing service 214.” EX1009, 171. As discussed herein, the payment processing system 308 of Dierks provides these same button-building and transaction payment processing services. EX1002, ¶79.

Accordingly, Dierks discloses an **HTML Generator** (“*embeddable buy button builder*”) executing on a first server operated by the payment processing system 308 (“*buy button provider*”). EX1002, ¶80.

(2) “receiving, by the embeddable buy button builder...a selection of an item from an item inventory”

Dierks discloses “receiving, by the embeddable buy button builder a selection of an item from an item inventory.” EX1002, ¶81.

Figure 1A of Dierks discloses that the vendor sells multiple items, *i.e.*, pieces of jewelry. EX1004, ¶35 (“The display 102 ... shows **a number of** high-quality, handcrafted **jewelry pieces** (see FIG. 1A.”). EX1002, ¶82.

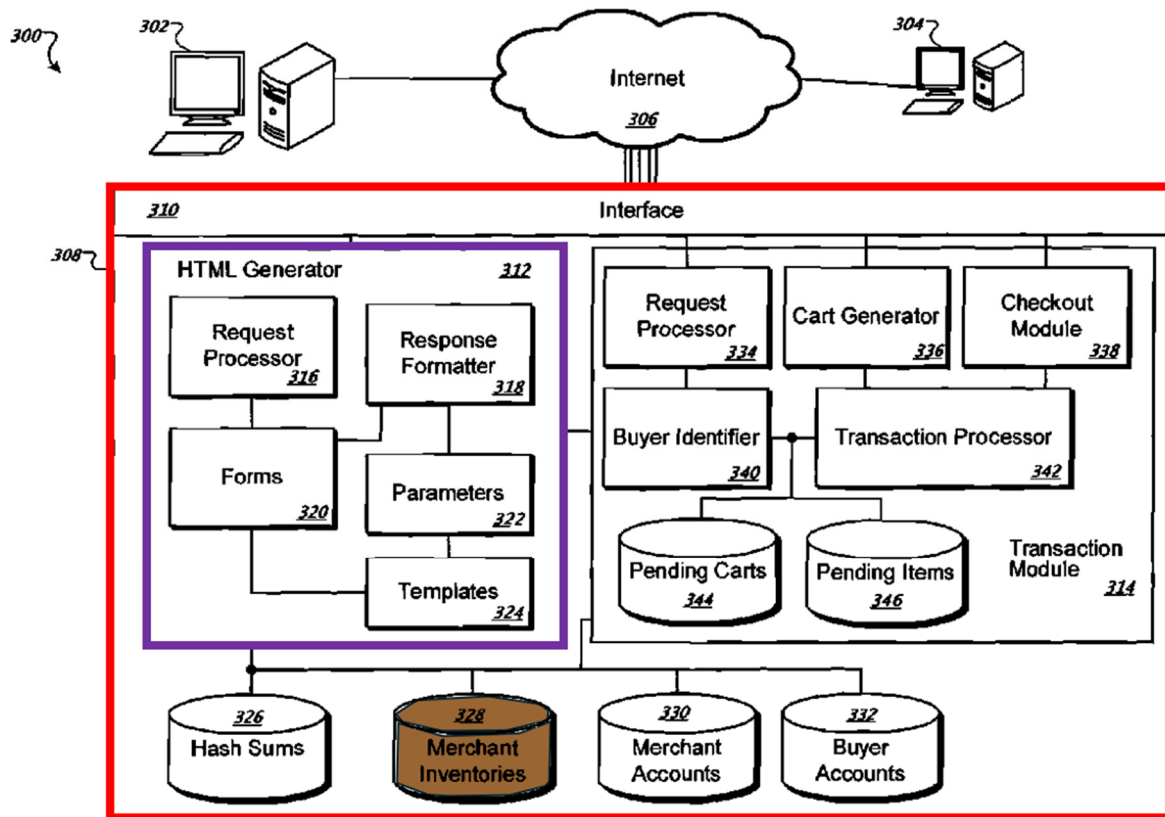


EX1004, FIG. 1A

Dierks further discloses that the vendor has an “*item inventory*,” stored in a database, that includes names and descriptions of items to be sold. EX1004, ¶53 (“[T]he seller may optionally also open and populate an **inventory database** for goods to be offered for sale by the seller. The seller in particular may identify **names and descriptions for items to be sold** ... along with prices for the items.”). Dierks discloses that a “Google Base” database can be used for storing the inventory. *Id.*, ¶41 (“where the **vendor stores** information relating to the **vendor’s inventory** using

a system associated with payment server 114⁵, such as GOOGLE BASE, the payment server 114 can also check such a **database** for availability of the selected goods.”); *id.*, ¶86 (“an **inventory of the seller** is available to the payment processing system (e.g., through Google Base).”). EX1002, ¶83.

Dierks discloses that the **payment processing system 308** communicates with a database called “**Merchant Inventories 328**,” which might be implemented using Google Base. EX1004, Fig. 3. EX1002, ¶84.



EX1004, FIG. 3

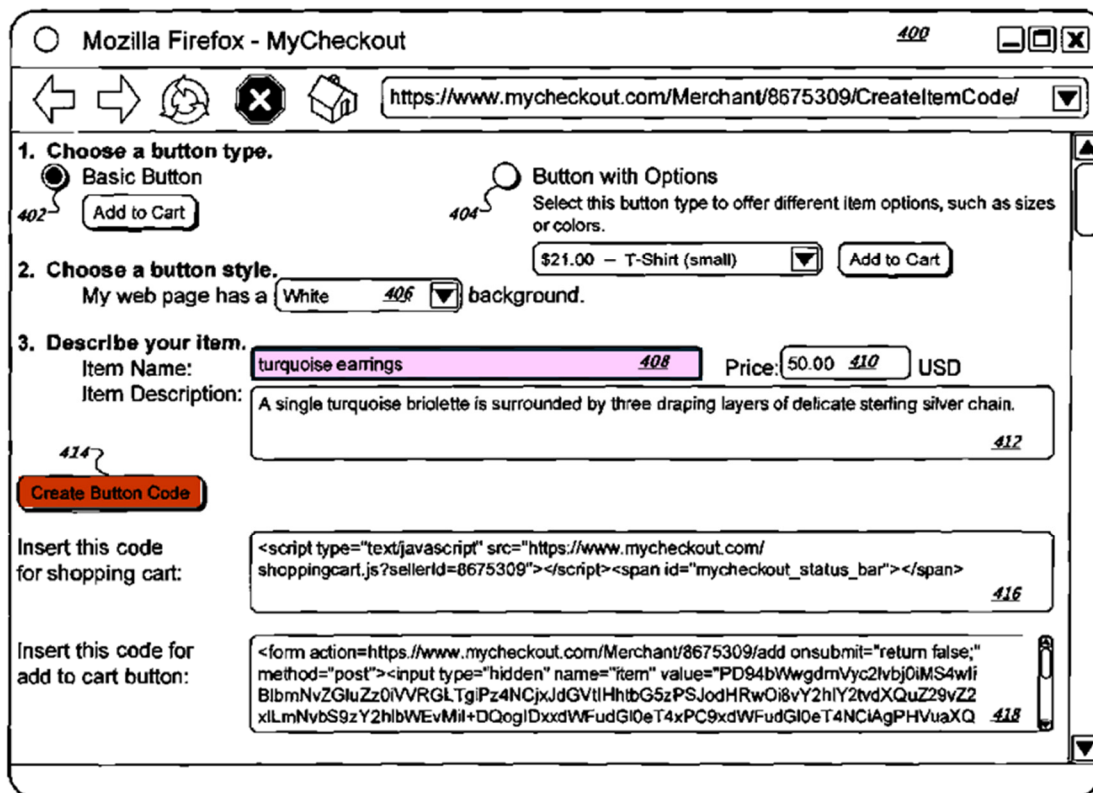
⁵ See n. 4.

A POSITA would have known that the Merchant Inventories 328 in Figure 3 represents a database because of the cylindrical structure used for 328. For example, reference numerals 326, 330, 332, 344, and 346 are shown with the same cylindrical structure and are each described respectively as a “database.” EX1004, ¶¶65, 66, 67, 68, 41. A POSITA would have understood that the Merchant Inventories database 328 could have been implemented with Google Base, because they are both disclosed as being inventory databases for vendors. Accordingly, Dierks discloses that the payment processing system 308 has access to an “*item inventory*” stored in database 328 or another database implemented with Google Base. EX1002, ¶¶85–86.

Dierks discloses that the HTML Generator 312 receives “*a selection of an item*” from the item inventory. In particular, Dierks discloses that the HTML generator “is configured to **receive parameters** regarding items to be sold by vendors and to generate code.” EX1004, ¶62; *see also id.*, ¶63 (“The **HTML generator 312** includes a number of structures for **receiving from web developers parameters** for web shopping controls and for generating HTML. ... The form may include areas in which the requester can provide information about an item the requester hopes to offer for sale, such as a description and price for the item. The **requester may enter such parameters** into the delivered form, and may submit the parameters back to payment processing system 308.”); *id.*, ¶64. These parameters

include the name of the item. *Id.*, ¶73 (“Text entry box 408 receives entry of a **product name. ...**”). EX1002, ¶87.

The seller provides the claimed “selection” of the item when entering the **item name**, as shown below in Figure 4, and pressing the “**create button code**” button 414. EX1004, ¶74 (“A **create button code** button 414 causes an HTTP request to be sent to a remote server, with the information provided in the display 400 being sent with the request. The remote server may then use the submitted information to generate code for managing a shopping cart, along with code for **generating an add to cart button.**”); *see also id.*, ¶73 (“Text entry box 408 receives entry of a **product name. ...**”); *id.*, ¶71 (“step-by-step process for **entering relevant information.**”). Thus, the seller “selects” an item from the item inventory when entering an item name that appears on the inventory list, and that selection is received by the HTML generator 312 (“the embeddable buy button builder”). EX1002, ¶88.



EX1004, FIG. 4

In particular, Dierks teaches that the item name entered in text box 408 (“turquoise earrings”) corresponds to a preexisting item name in inventory. This is because Dierks discloses that the seller may populate an inventory database for goods “to be offered for sale” (i.e., *not for sale yet*) when opening up an account. EX1004, ¶53 (“The seller may ... **open an account** into which payments from buyers may be placed. In addition, **the seller may** optionally also **open and populate an inventory database** for goods **to be offered for sale** by the seller.”); *see also id.*, ¶54. Since Dierks uses the future tense of “to be offered for sale,” Dierks teaches that the inventory list exists *before* the vendor begins entering the **item name**

in text box 408. The vendor *selects* an item listed on the item inventory because the inventory contains the items that the user has for sale. EX1002, ¶89.

Moreover, Dierks discloses monitoring and updating services that require use of the item inventory. For example, the payment processing system 308 checks the availability of items using the inventory to confirm that items are available. EX1004, ¶41 (“where the **vendor stores** information relating to the **vendor’s inventory** ..., the payment server 114 can also **check such a database for availability** of the selected goods.”). Dierks also discloses that the number of items in the inventory is decreased after an item is purchased. *Id.*, ¶86. (“where an **inventory of the seller** is available to **the payment processing system** (e.g., through Google Base), the **inventory may be decreased** for each of the **purchased items**.”). EX1002, ¶90.

A vendor needs to select an item from the inventory to avail itself of the monitoring and updating services. A POSITA would have been taught that if the vendor enters an **item name** *not from the item inventory*, the payment processing system 308 would not be able to check the availability of that item using the inventory nor update the inventory when that item is sold. EX1002, ¶91.

Accordingly, Dierks teaches receiving, by the HTML Generator 312 (“*an embeddable buy button builder*”) a selection of an item from the Merchant

Inventories database 328 and/or Google Base (i.e., “*an item inventory*”). EX1002, ¶92.

(3) “receiving, by an embeddable buy button builder ... a representation of the item, wherein the item is to be listed for sale on the electronic commerce store”

Dierks combined with Johnston teaches “receiving, by an embeddable buy button builder ... a representation of the item, wherein the item is to be listed for sale on the electronic commerce store.” EX1002, ¶93.

Dierks does not explicitly disclose that the payment processing system 308 (“*the embedded buy button builder*”) receives a “selection of ... a representation of the item.” However, because of Johnston’s teachings, a POSITA would have found this obvious. In Fig. 1B of Dierks, the vendor’s web page (part of “*the electronic commerce store*”) includes the **item name** (e.g., “turquoise with silver chain”), an **item description** (e.g., “single turquoise briolette is surrounded by three draping layers of delicate sterling silver chain”), a **price** (i.e., \$42), and an **image of the product for sale**. EX1004, ¶36. EX1002, ¶94.

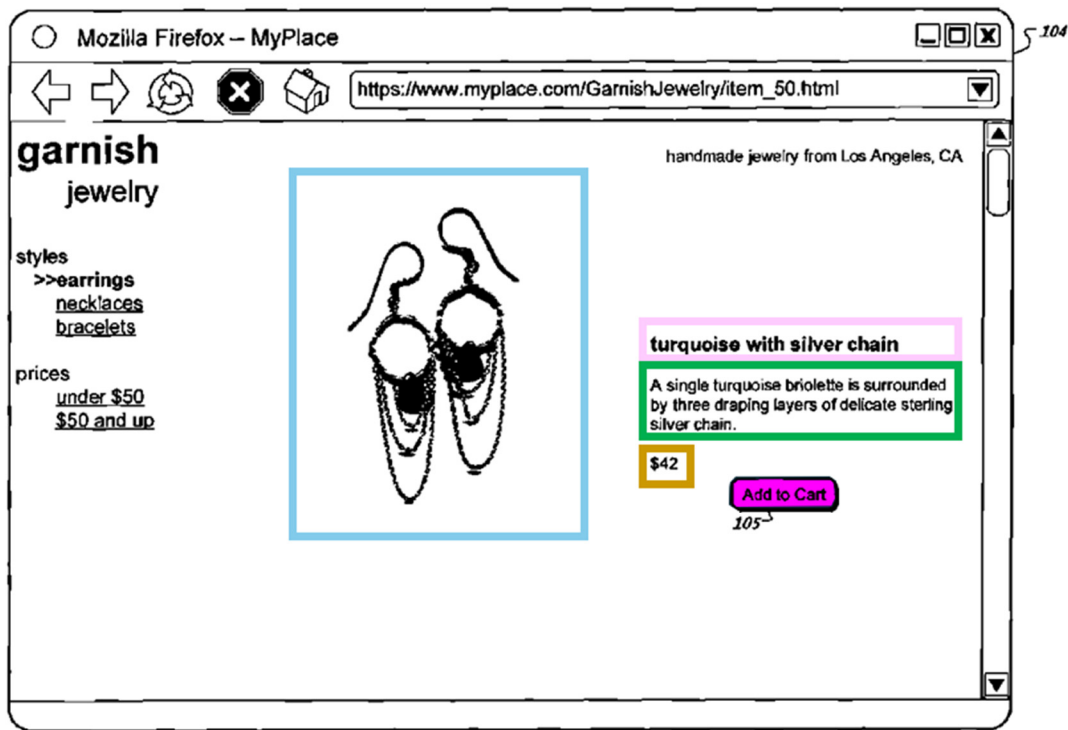


FIG. 1B

EX1004, FIG. 1B

Dierks teaches that the **item name** is selected, as described above, and discloses that the **item description** and the **price** are entered through the screen shown below in FIG 4. EX1004, ¶73 (“*text entry areas 408-412* permit the user to *enter parameters* for the product to be sold.”). EX1002, ¶95.

Mozilla Firefox - MyCheckout

https://www.mycheckout.com/Merchant/8675309/CreateItemCode/

1. Choose a button type.

☒ Basic Button 402 404

Button with Options
Select this button type to offer different item options, such as sizes or colors.

\$21.00 - T-Shirt (small)

2. Choose a button style.

My web page has a 406 background.

3. Describe your item.

Item Name: 408 Price: 410 USD

Item Description: 412

414

Insert this code for shopping cart:

```
<script type="text/javascript" src="https://www.mycheckout.com/shoppingcart.js?sellerId=8675309"></script><span id="mycheckout_status_bar"></span>
```

Insert this code for add to cart button:

```
<form action="https://www.mycheckout.com/Merchant/8675309/add" onsubmit="return false;" method="post"><input type="hidden" name="item" value="PD94bVwgdMvYc2lvbj0iMS4wIiBibmNvZGluZz0iVVRGLTgiPz4NCjxJdGVtIHhtbG5zPSJodHRwOi8vY2h1Y2tdXQuZ29vZ2xiLmNvbS9zY2hlbWEvMil+DQogIDx0dWFudGI0eT4xPC9xdWFudGI0eT4NCiAgPHVuaXQ
```

EX1004, FIG. 4

The payment processing system 308 then generates HTML used to display the item name, description, and cost—of the item to be listed for sale on the electronic commerce site—based on the selected parameters. EX1004, ¶15 (“receive parameters relating to **an item offered for sale** and to **generate HTML code**.”). EX1002, ¶96.

Dierks is silent regarding *how* the **image of the item**, shown above in Figure 1B, is provided. A POSITA would have found it obvious for the image in Dierks to be provided to the payment processing system 308 by the vendor. The image would

be received in a manner similar to how the above-noted parameters are received through the screen of Figure 4. Doing so would save vendors the trouble of inserting images into web pages themselves or spending time to have someone else insert the images, which aligns with stated goals of Dierks, *e.g.*, to assist users that have “limited computer skills” and limited “time.” EX1004, ¶35 (“the **vendor** is an individual who **works full-time at another job** and ... **has limited computer skills** ... she **does not have the time to learn** how to set up a full-blown e-commerce site.”). EX1002, ¶97.

Thus, a POSITA would have found it obvious to configure the features in Figure 4 of Dierks to include a prompt telling the vendor to upload the image. Doing so was well known before the priority date of the ’785 patent, as taught by Johnston and as discussed next. EX1002, ¶98.

(4) Combination of Dierks and Johnston

The combination of Dierks and Johnston renders obvious the “receiving, by embedded buy button builder” a “selection of ... a representation of the item,” as recited in element 1[a]. EX1002, ¶99.

First, Johnston discloses and illustrates in Figures 3A–3G, a seller “creating an OWJO Storefront via a user interface of the OWJO platform.” EX1005, ¶225. For example, in Figure 3B (reproduced below), the user interface prompts the seller to enter information associated with an item to be listed for sale. *Id.*, ¶226

(“seller can create a music track by specifying 1) the **name** of the track ... 7) a thumbnail **image** that can be displayed in the OWJO Storefront and 8) the **price** that the user would like to sell his or her music track for.”). The item for sale in Figure 3B is music. The seller can do the same for other items for sale, such as videos, images, podcasts, and vodcasts. *See id.*, ¶¶227–230, Figs. 3C–3F. EX1002, ¶100.

The screenshot shows the OWJO website's 'Upload Content' page. The top navigation bar includes 'Welcome CPA (Log out...)', 'Home', 'About us', and 'Site map'. Below this is a secondary bar with 'Upload', 'Sales', 'Purchases', 'Top Up', 'Summary', 'Publish', 'Settings', and 'Admin'. The main content area is titled 'Upload Content' and has tabs for 'music', 'videos', 'images', 'podcasts', and 'vodcasts'. The 'music' tab is selected. On the left, there's a 'My Stuff' section showing 'User: TheFirstName' and '30 (0 items)'. It lists three items: 'Running after Cars', 'Comfortably Numb', and 'Dude it's good!'. Below this is a 'Featured' section with four items: 'Running after Cars', 'Comfortably Numb', 'Dude it's good!', and 'The Podcast Title'. The right side of the page is the 'Music file Settings' form. It includes fields for 'Name of track:', 'Artist:', 'Tags:', and 'Genre:' (with a dropdown menu set to 'Pop'). There are two 'Browse...' buttons for uploading a 'Preview track' and a 'Music file'. The 'Thumbnail' section, which is highlighted with a blue border, contains a 'Browse...' button. At the bottom, there's a 'Set your price' section with a dropdown menu set to 'EUR' and a text input field with '0'. An 'upload' button is at the very bottom right.

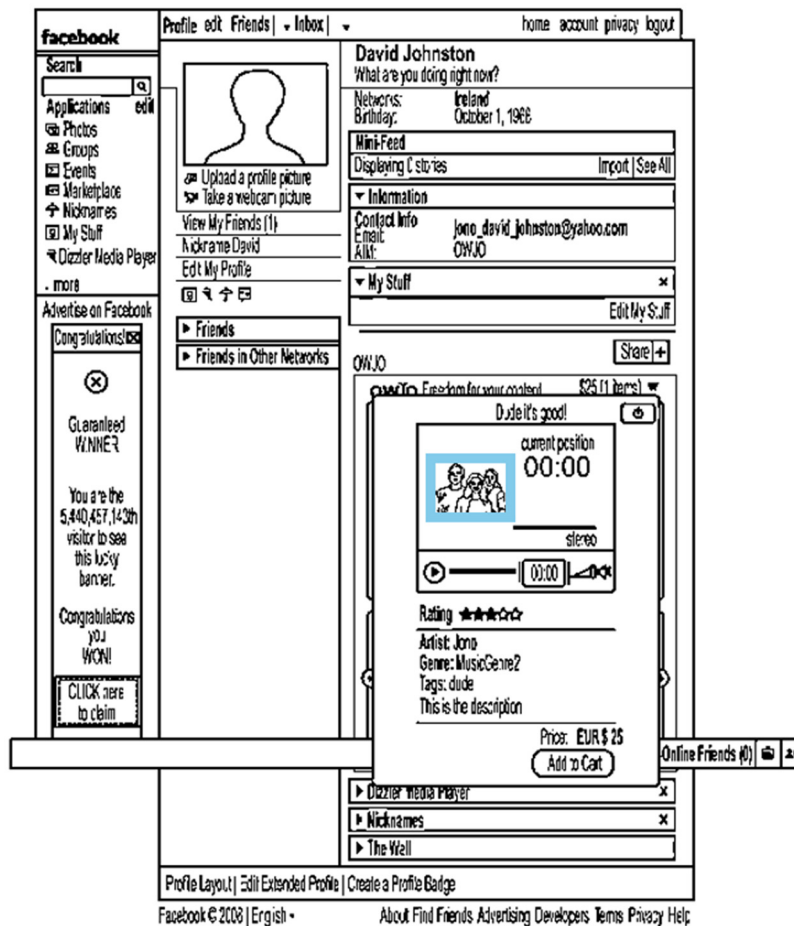
EX1005, FIG. 3B

The interface of Figure 3B prompts the seller to select the image that represents the item for sale (“*a representation of the item*”) by instructing the seller

to “provide a thumbnail file that we will display on the OWJO Widget,” as shown above in Figure 3B, by using a “browse” button. EX1005, Fig. 3B. EX1002, ¶101.

Johnston discloses that the image is displayed on the Storefront as a “*representation of the item*” being sold. EX1005, ¶226 (“a seller can create a music track by **specifying** ... a thumbnail **image** that can be **displayed in the OWJO Storefront.**”); *id.*, ¶¶222, 289. EX1002, ¶102.

Figures 4A and 4B of Johnston show the user’s Storefront with the representative image. For example, the Storefront in Figure 4B shows an **image that represents** the song for sale. EX1005, Fig. 4B, ¶68 (“FIG. 4B is an embodiment of the user interface of a Storefront to preview content.”); *cf.* EX1001, 13:6–9 (“the multimedia display files include visual representations of the **cover art** used on each **album** in the **musical catalogue.**”). EX1002, ¶103.



EX1005, FIG. 4B

Johnston discloses that the seller *selects* the image by loading the image from the user's computer or from anywhere on the internet. EX1005, ¶226 (“The ... **thumbnail** can either be **uploaded from the machine** that is executing the web browser they are using or **loaded from a Uniform Resource Identifier** anywhere on the internet.”); *see also id.* ¶¶227–230, 272. EX1002, ¶104.

Johnston discloses that the image is “received” by the system, as recited in element 1[a], and discloses that the system stores the image. For example, Johnston

discloses a storage 106 that “is a set of one or more storage services or locations **to store and retrieve content by the platform 110**. ... An artist or seller may provide and/or maintain his/her own storage 106.” EX1005, ¶118. Johnston also discloses a content acquisition service 135 that is “constructed to ingest, **receive**, register or obtain **any type and form of predetermined formats or content types, such any format used for any type and form of video, audio, documents, music, movies, etc.**” *Id.*, ¶204. Johnston also discloses a storage module that “may manage any type and form of storage and may include storage internal to the platform 110 or external to the platform 110.” *Id.*, ¶178. A POSITA would have understood that the image uploaded by the user would be stored in at least one of the storage areas disclosed in Johnston. Regardless of where the image is stored, a “*selection*” of the image is received by Johnston’s system so the image can be used for the Storefront. EX1002, ¶105.

(5) Motivation to Combine Dierks and Johnston

A POSITA would have been motivated to combine Dierks and Johnston so the vendor in Dierks can upload an image to the payment processing system 308 for use on the vendor’s web page to represent the item for sale. EX1002, ¶106.

First, Dierks and Johnston are from the same field of endeavor. Both references are directed to electronic commerce. Dierks discloses a “computer-implemented method [for] generating on a user terminal a **merchant web site**

associated with a merchant with code for a merchant web page.” EX1004, Abstract; *see also id.* ¶21. Similarly, Johnston discloses “a platform for the delivery, management and payment of **content between sellers and buyers. ... establish a store front** at one or more hosted services.” EX1005, ¶2, *see also id.* ¶21. EX1002, ¶107.

Johnston, similar to Dierks, also acknowledges the investment needed to “create a website that has full ecommerce facilities” and offers a simple solution to people that want to sell items online. EX1005, ¶4 (“If an artist wants to generate income ... they would have to 1) **create a website that has full ecommerce facilities. ... This is a significant investment** and would **deter most content creators.**”). EX1002, ¶108.

Johnston, similar to Dierks, provides a “self-service” way for a seller to establish an e-commerce store front. EX1005, ¶2 (“**methods of self-service by a seller to establish a store front... .**”). Johnston does this by allowing a user to simply provide information about an item for sale, including an image representation of the item, so that the system handles the work of preparing the e-commerce store. EX1002, ¶109.

Thus, both references are directed to electronic commerce and to creating web sites for sellers. Accordingly, a POSITA would have understood that Johnston’s teachings were directly applicable to Dierks’s system. EX1002, ¶110.

Second, a POSITA would have combined the elements taught by Dierks and Johnston—namely, a system that provides a product listing generated based on information provided from the user, as taught by Dierks, and the function of allowing users to select an image representation of the product, as taught by Johnston—using known and routine computer programming principles that allow the payment processing system 308 in Dierks to receive the image and generate HTML for displaying the image on the vendor’s web site. While Dierks discloses a “thumbnail” image, a POSITA would have understood that a larger file size image could be used, if the image needs to be a particular size or resolution. A POSITA would have understood that a “thumbnail” image is simply a scaled-down version of the original image that is used when an image is not needed to occupy an entire screen, and it has the advantage of allowing the browser to display its page more quickly. The system of Dierks would store the image for the vendor, as discussed above for Johnston (EX1005, ¶¶118, 178, 204), and generate HTML that embeds the image along with the HTML generated for the item name, description, and price, as discussed below for elements 1[b], and 1[c]. The image could be embedded with known techniques taught by Dierks, such as by using the “src” attribute containing a URL pointing to the image to embed. EX1004, ¶38 (“src=‘https://checkout.google.com/**img**/addtoCart.gif’ alt=‘Add to Cart’ type=‘**image**’”); *id.*, ¶43 (“src=‘https://checkout.google.com/**img**/gcb2.gif’ alt=

‘checkout’ type=‘**image**’ id= ‘gcb_img’’). When a user’s browser parses the HTML from the vendor’s web site, the image would be fetched and displayed. This result would have been readily predictable and recognized by a POSITA. EX1002, ¶111.

Third, configuring Dierks based on Johnston’s teachings is simply an application of a known technique (uploading an image for inclusion on a web site) to a known system ready for improvement (Dierks’s system that displays an image but does not explicitly disclose where the image comes from) with predictable results. *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 416 (2007); M.P.E.P. §2143(I)(B). Application of this known technique would have been readily achievable by a POSITA via known and routine computer programming principles to implement Johnston’s image uploading and displaying features in the system in Dierks. This configuration of Dierks would have yielded the expected, predictable result of displaying an image “*representation of the item*” for sale, while not requiring sellers to possess the technical skills needed to add such an image on their own. EX1002, ¶112.

Fourth, a POSITA would have been motivated to combine the teachings of Dierks and Johnston because allowing a user to upload an image representation of an item for sale on an e-commerce web site would have been obvious to try. *KSR*, 550 U.S. at 421; M.P.E.P. §2143(I)(E). There were only two possible known and predictable choices for generating a product listing that includes an image of the item

for sale—(1) requiring users to generate the code themselves for displaying images and (2) preparing the code for the users, so the users need not learn how to prepare the code themselves. A POSITA could have pursued these two potential solutions with a reasonable expectation of success because the analysis would have simply involved trying each one of these two solutions. *See, e.g., CRFD Research, Inc. v. Matal*, 876 F.3d 1330, 1347 (Fed. Cir. 2017) (“[A] person of ordinary skill would have two predictable choices ... providing [the] person of ordinary skill with a simple design choice” of choosing one solution or the other). Thus, a POSITA would have found it obvious to configure Dierks by incorporating Johnston’s teaching of embedding an image, representing a product, for the seller. EX1002, ¶113.

Fifth, the configuration of Dierks’s system to include the functionality of receiving an image from a user and generating HTML that displays the image would have been readily implemented by augmenting the screen shown in Figure 4 of Dierks to prompt a user to browse for and select an image to display and having the HTML Generator in Dierks generate HTML that displays the image. The HTML generator in Dierks already generates HTML that displays the item name, description, and price of the item, as discussed above. EX1004, ¶¶63–64, 73–74. Thus, the payment processing system 308 would simply be configured to store the image in one of its storage areas, and the HTML generator would simply be

configured to generate the HTML that displays the image, along with the item name, description, and price. A POSITA would have implemented this configuration using basic computer programming principles to configure Dierks's payment processing system 308 to receive and store the image so the generated HTML, which is used for the vendor's web site, displays the image when the web site is accessed. Making this configuration would have been well within the knowledge and skillset of a POSITA. EX1002, ¶114.

***(6) Reasonable Expectation of Success in Combining
Dierks and Johnston***

A POSITA would also have had a reasonable expectation of success. The combination of Dierks and Johnston represents a straightforward implementation that configures the functionality of Dierks's system to upload an image and prepare HTML that displays the image. Dierks already discloses an HTML Generator that generates HTML to display the name, description, and price of the item, and a POSITA would have been familiar with and able to implement the functionality to do the same for an uploaded image. The reasons a POSITA would have expected success parallel those that motivate this combination—including because both Dierks and Johnston are directed to electronic commerce, and both disclose product listings with image representations. *See, e.g.*, Sections VIII, X.A.1.b.4-5. EX1002, ¶115.

Accordingly, Dierks in view of Johnston renders obvious element 1[a] regarding “receiving, by embedded buy button builder” a “selection of ... a representation of the item.” For these same reasons, a POSITA would have been motivated to combine the teachings of these references for all subsequently discussed claims and limitations in this ground. EX1002, ¶¶115–116.

Further, Dierks discloses that the item is sold on the vendor’s web page with the “add to cart” button and, therefore, discloses that “the item is to be listed for sale on the electronic commerce store” as claimed. EX1004, ¶62. EX1002, ¶117.

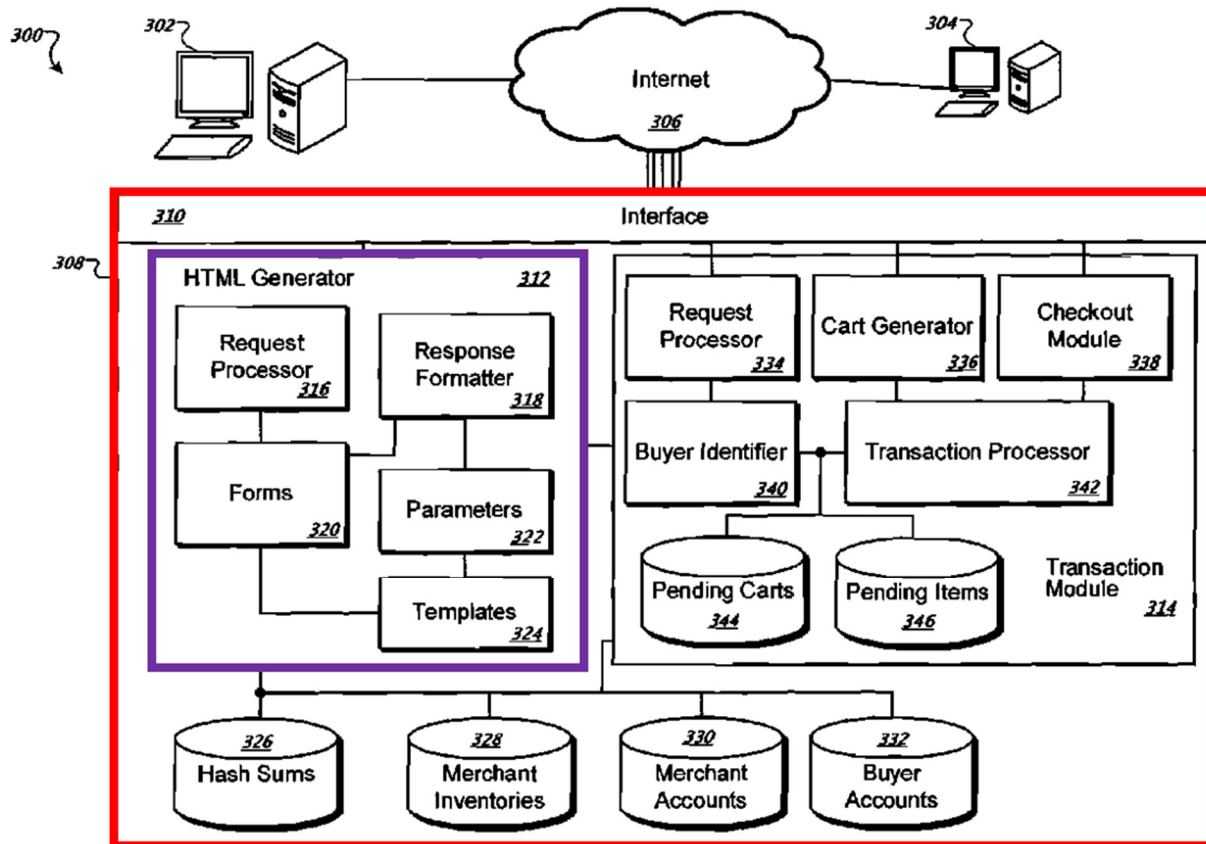
Thus, Dierks and Johnston teach receiving, by the HTML Generator 312 (“*the embeddable buy button builder*”) executing on a first server operated by a buy button provider, the name of an item in inventory (“*selection of an item from an item inventory*”) and an item description (“*representation of the item*”), wherein the item is to be listed for sale on a vendor’s web page (“*the electronic commerce store.*”). EX1002, ¶118.

c) 1[b]: “generating, by the first server, the embeddable buy button”

Dierks discloses element 1[b]. EX1002, ¶124.

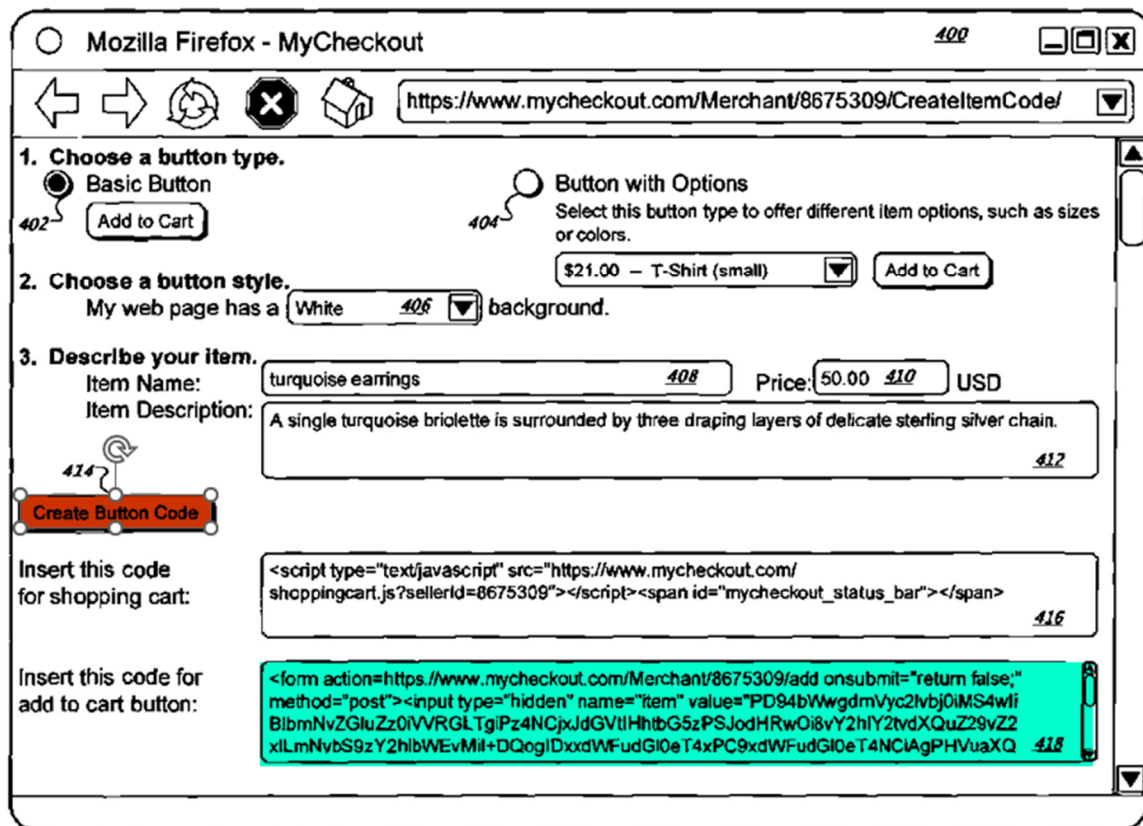
As discussed for element 1[a], Dierks discloses that the **payment processing system 308** (“*the first server*”) includes the **HTML Generator 312**, shown below, that generates the code for the “add to cart” button 105 (“*the embeddable buy*

button”). EX1004, ¶66 (“an ‘add to cart’ control, added by the HTML generator 312.”). EX1002, ¶120.



EX1004, FIG. 3

As discussed for element 1[a], the **HTML Generator 312** receives parameters that the seller enters into the screen shown below in Figure 4. EX1002, ¶121.



EX1004, FIG. 4

After the parameters are entered, the seller selects the **“Create Button Code” button 414** that causes the HTML Generator 312 to generate the “add to cart” button. EX1004, ¶74 (“A **create button code button 414** causes an HTTP request to be sent to a remote server, with the information provided in the display 400 being sent with the request. The remote server may then use the submitted information to generate code for managing a shopping cart, along with code for **generating an add to cart button.**”); *see also id.*, ¶62 (“The **HTML generator**, in this example, is configured to receive parameters regarding items to be sold by vendors and to **generate code**, such as HTML or XHTML, for the vendors to include in their web pages. ...”); Fig.

4. The code generated for the “add to cart” button is shown above in **area 418**. EX1004, ¶74 (“The code for generating an add to cart button, displayed in **area 418**, may take the form of HTML code”). EX1002, ¶122.

Accordingly, Dierks discloses “*generating*,” by the HTML Generator 312, which is part of “*the first server*,” the “add to cart” button (“*the embeddable buy button*”). *Id.*, ¶123.

d) 1[c] “wherein the embeddable buy button is configured to display the representation of the item, and a hyperlink;”

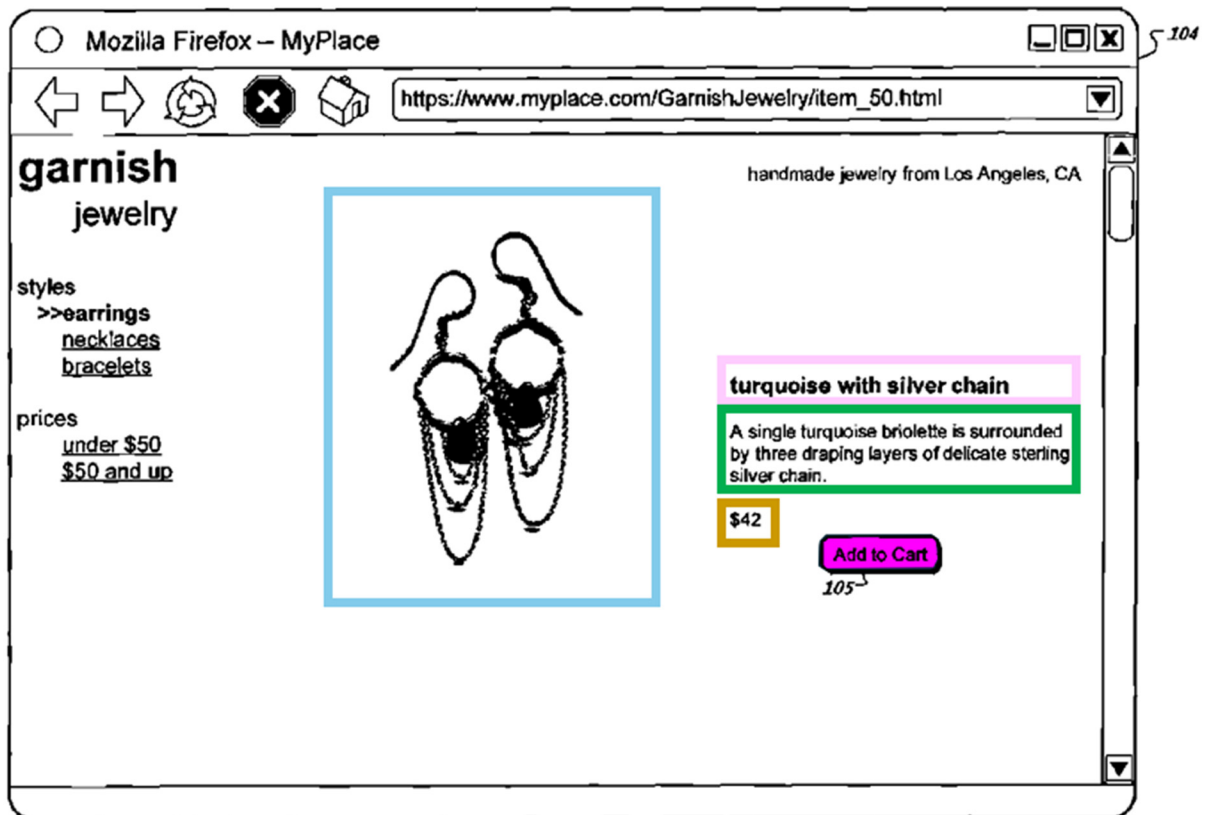
The combination of Dierks and Johnston teaches element 1[c]. EX1002, ¶131.

As discussed for element 1[a], the combination of Dierks and Johnston teaches that the HTML Generator 312 (“*the embeddable buy builder*”) receives an image representation of the item. EX1002, ¶125.

The code generated for the “add to cart” button displays more than the depiction of the button 105. The code also displays the item name, description, and price, along with the depiction of the button. In particular, Dierks discloses that the parameters entered into the screen of Figure 4 are “coded along with the button.” EX1004, ¶73 (“Text entry box 408 receives entry of a **product name**, while text entry box 410 receives a **product price**, and text entry box 412 receives a **product**

description. Such entries may be **coded along with the button**, so as to be **displayed** on the vendor's web page **with the button. ...**"). EX1002, ¶126.

Thus, the item name, description, and price are coded with the button, i.e., the generated code displays the depiction of the “**add to cart**” button, the item name (“**turquoise with silver chain**”), the **description**, and the **price**. These items are placed next to the **image of the item**, shown in Figure 1B. EX1002, ¶127.



EX1004, FIG. 1B

A POSITA would have understood that the code for the item name, description, and price are included with the code for the depiction of the “**add to**

cart” button because Figure 4, shown below, has two areas 416 and 418 that include code. Area 416 is used for a “shopping cart” feature discussed further below. This leaves area 418 to include the code for the depiction of the button 105 and the item name, description, and price. EX1002, ¶128.

Mozilla Firefox - MyCheckout 400

https://www.mycheckout.com/Merchant/8675309/CreteItemCode/

1. Choose a button type.

☒ Basic Button 402

☐ Button with Options 404
Select this button type to offer different item options, such as sizes or colors.

2. Choose a button style.

My web page has a 406 background.

3. Describe your item.

Item Name: 408 Price: 410 USD

Item Description: 412

414

Insert this code for shopping cart:

```
<script type="text/javascript" src="https://www.mycheckout.com/shoppingcart.js?sellerId=8675309"></script><span id="mycheckout_status_bar"></span>
```

Insert this code for add to cart button:

```
<form action="https://www.mycheckout.com/Merchant/8675309/add" onSubmit="return false;" method="post"><input type="hidden" name="item" value="PD94bWwgdmVyc2lvbj0iMS4wIjBibmNvZGluc2Z0IHRVVGltZG5zPSJodHRwOi8vY2h1Y2tdXQuZ29vZ2xlLnNvbS9zY2h1bWEvMil+DQogIDx0dWFudGI0eT4xPC9xdWFudGI0eT4NCiAgPHVuaXQ
```

EX1004, FIG. 4

When Dierks is configured in view of Johnston, to receive the uploaded image representation of the item, as discussed for element 1[a], it would have been obvious to generate the code for the embedded buy button by including the code for displaying the image, along with the code for displaying the name, description, and

price. This is because (1) the image representation of the item is another form of a product description, (2) the image is disclosed by Dierks as being positioned next to the name, description, and price of the item, as shown above in Figure 1B, and (3) Dierks discloses that the code for the “add to cart” button is already configured to display more than just the depiction of the button 105. EX1002, ¶129.

A POSITA would have included the code needed to display the image using known and routine computer programming principles. Therefore, the combination of Dierks and Johnston teaches that the “add to cart” button (“*the embeddable buy button*”) is configured to display the image representation of the item. *Id.*, ¶130.

Further, when the “add to cart” button 105 is pressed, it causes an HTTP request and, therefore, includes a generated hyperlink⁶, such that the generated code for the “add to cart” button is configured to display a hyperlink, i.e., display the depiction of the button that functions as a hyperlink. EX1004, ¶38 (“[s]election of the *button 105 may cause an HTTP request*, in the form of an XMLHttpRequest (XHR), to be made of payment server 114⁷, as indicated by Arrow A.”). A POSITA would have understood that a hyperlink can be presented in different forms, like an

⁶ The term “hyperlink” is not used in the ’785 patent specification outside of the claims.

⁷ See n. 4.

image, icon, text, or any visible element that, when clicked, sends an HTTP request. *See, e.g.*, EX1022, 5:26-28 (“The web page of the reseller includes a hyperlink, which is indicated by a graphic display, such a button, highlighted text, or the like.”); *id.*, 2:47-53. In the case of Dierks, the displayed hyperlink is in the form of the “add to cart” button 105 that is visible to the user and causes the HTTP request when pressed. EX1002, ¶131.

Accordingly, the combination of Dierks and Johnston teaches that the “add to cart” button, including its underlying code that is generated by the first server, (“*the embeddable buy button*”) is configured to display the image of the item (“*the representation of the item*”), and display the generated hyperlink provided by the depiction of the button. EX1002, ¶131.

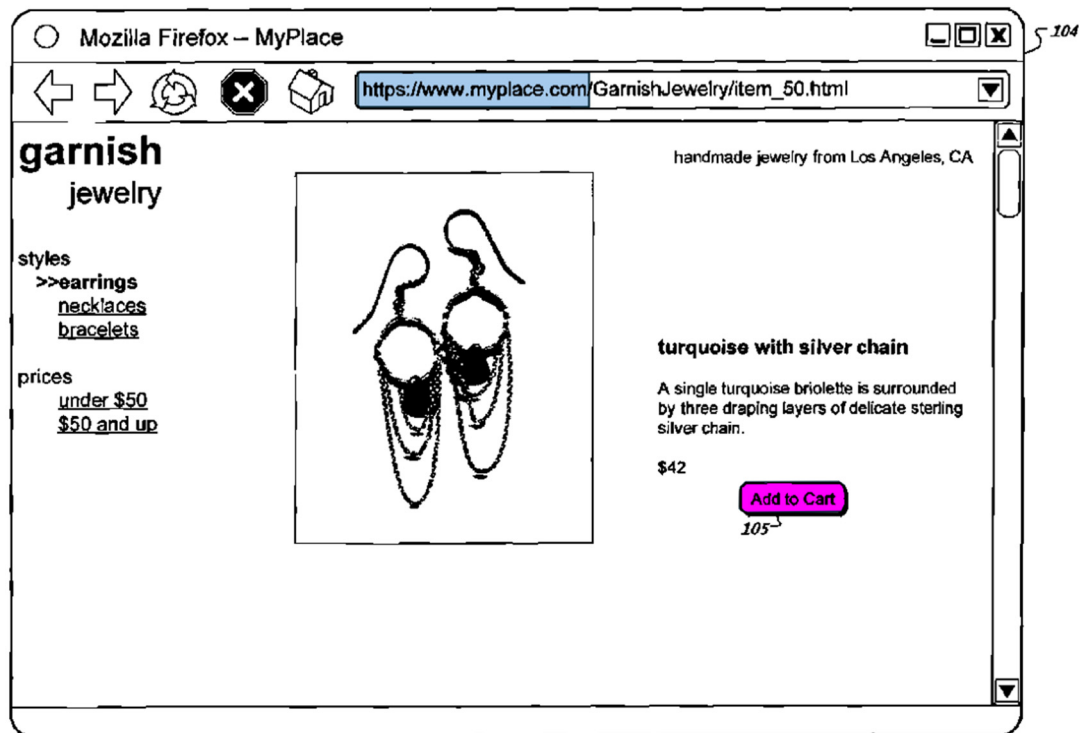
e) 1[d]: “embedding, by the buy button provider, the embeddable buy button into a web site,

Dierks discloses element 1[d]. EX1002, ¶132.

Dierks discloses that the payment processing system 308 (“*the buy button provider*”) embeds the “add to cart” button (“*the embeddable buy button*”) into a web site. The payment processing system 308 can embed the code “automatically, such as by *pushing the code* to particular spots in an *open mark-up language document* on the seller’s client device, e.g., at particular *pre-determined tags inserted in the page.*” EX1004, ¶78. Dierks also discloses that the code can be

embedded by inserting “pointers” in the web page that point to the code. *Id.*, ¶78 (“*pointers to the code* may be placed in the seller’s web page. In addition, the code incorporated into the web page may itself *point to other code*, such as Javascript code.”). EX1002, ¶133.

A POSITA would have understood that Dierks’s reference to “page” and “web page” means a “web site.” For example, Figure 1B of Dierks shows the following domain address, which is a web site: <https://www.myplace.com>. The specific web page within the web site with the “add to cart” button is https://www.myplace.com/GarnishJewelry/item_50.html. EX1004, Fig. 1B. EX1002, ¶134.



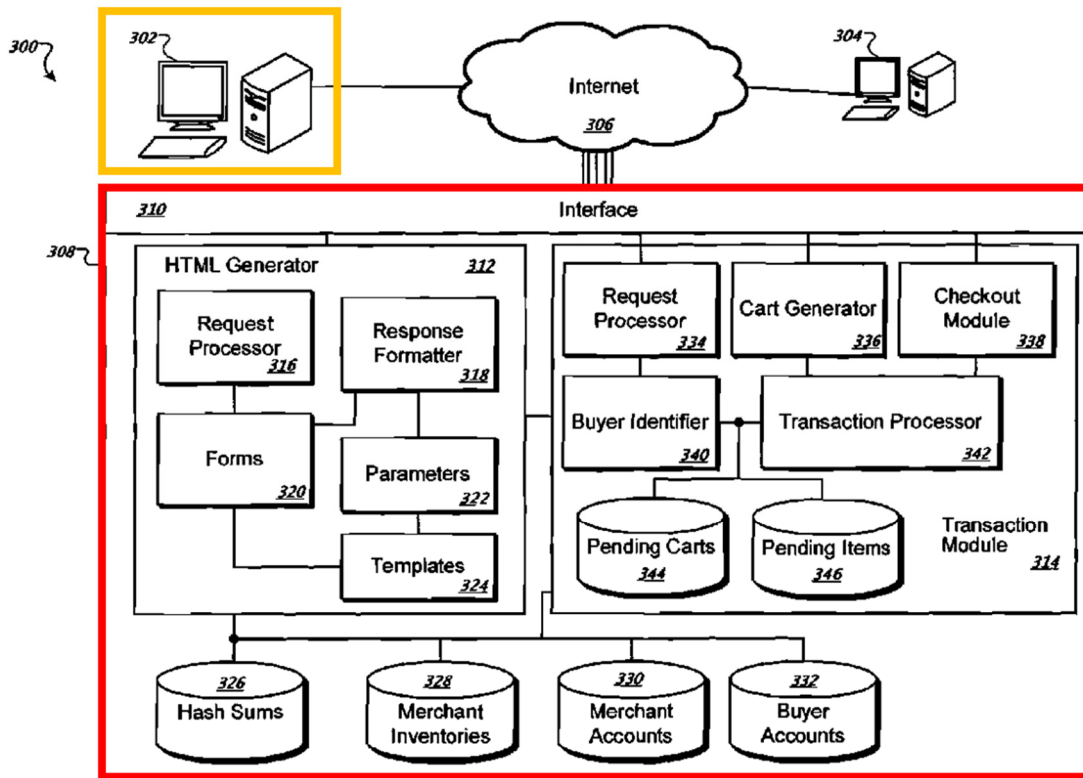
EX1004, FIG. 1B

Accordingly, the payment processing system 308 (“*the buy button provider*”) embeds the code for the “add to cart” button by inserting the code, either directly or by using pointers, into the vendor’s web site. EX1002, ¶135.

f) 1[e] wherein the web site is executed by a second server not operated by the buy button provider,”

Dierks discloses element 1[e]. *Id.*, ¶137.

Fig. 3 of Dierks discloses **server 302** that provides access to the web site. EX1004, ¶61 (“**server 302** ... may represent a web server *providing access* to one or more *web sites* that are offering goods or services for sale.”). A POSITA would have understood that these “web sites” that offer goods for sale correspond to the web site discussed for element 1[d] that includes the embedded “add to cart” button. EX1002, ¶138.



EX1004, FIG. 3

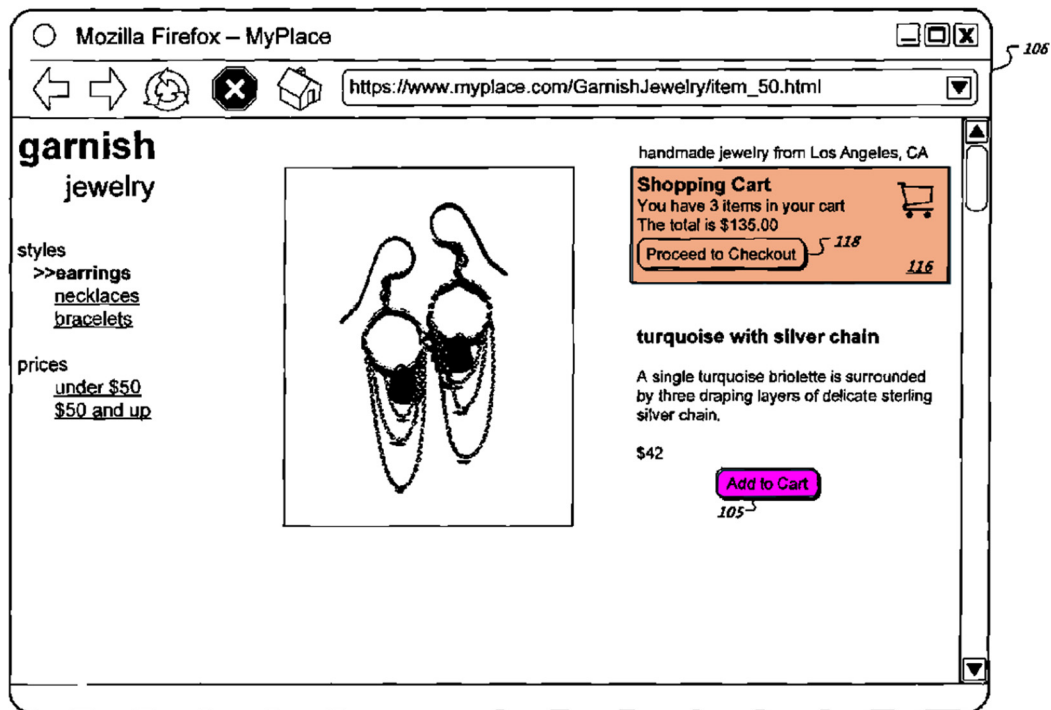
A POSITA would not have expected the **server 302** to be operated by the **payment processing system 308** (“buy button provider”) since they provide separate services and are not disclosed as being related. Accordingly, Dierks discloses that the web site is executed by the server 302 (“*a second server*”) and is *not* operated by the payment processing system 308 (“*the buy button provider*”). EX1002, ¶139.

g) 1[f]: “wherein the electronic commerce store is displayed within the web site upon selection of the hyperlink,

Dierks discloses element 1[f]. *Id.*, ¶141.

Dierks discloses a “**shopping cart**” 116 feature that forms part of the electronic commerce store because it facilitates the purchase of an item, as shown below in Figure 1C. EX1004 ¶9 (“the **shopping cart** information includes displaying a **shopping cart** total that includes shipping and tax charges.”); *id.*, ¶51 (“When the purchaser chooses to complete the purchase, their account is debited by the payment server 114⁸ in an amount **equal to the price of all the goods in the shopping cart. ...**”). EX1002, ¶142.

⁸ See n. 4.



EX1004, FIG. 1C

The **shopping cart 116** is displayed within the web site upon selection of the “**add to cart**” 105 button (“*the hyperlink*”). Pressing the “add to cart” button 100 activates the hyperlink, as described for element 1[c]. EX1004, ¶43 (“Display 106 shows the result of a **user’s selection of button 105** (see FIG. 1C). Specifically, a **shopping cart object** shown as a graphical element”). EX1002, ¶143.

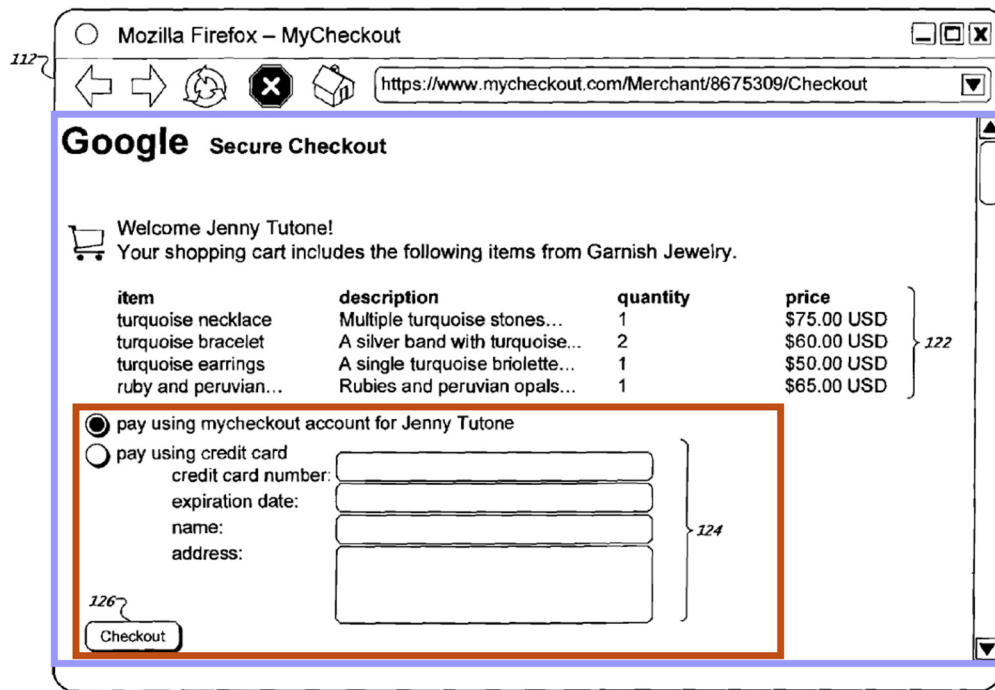
Accordingly, Dierks discloses that the shopping cart (part of “*the electronic commerce store*”) is displayed within the web site upon selection of the “add to cart” button (“*the hyperlink*”). EX1002, ¶144.

h) 1[g] the electronic commerce store configured to facilitate payment processing related to a purchase of the item, and”

Dierks discloses element 1[g]. EX1002, ¶146.

Dierks discloses a secure checkout (another part of “*the electronic commerce store*”) that facilitates payment processing related to a purchase of the item. The secure checkout is accessed when a user selects the “proceed to checkout” button 118 shown in the shopping cart 116 discussed for element 1[f]. EX1004, ¶44 (“**go to a checkout to pay** or otherwise edit entries in their shopping cart, by **selecting** the ‘**proceed to checkout**’ control button 116”); *id.*, ¶48 (“At display 112, the user/purchaser has selected a ‘**proceed to checkout**’ button. ...”). EX1002, ¶147.

The **secure checkout** is shown below as having a **payment mechanism 124** that facilitates payment processing related to the purchase of an item. EX1002, ¶148.



EX1004, FIG. 1E

The **payment mechanism 124** facilitates payment processing by providing the user with options to pay for the item. EX1004, ¶48 (“The user may also be permitted to *select a payment method*, such as through a GOOGLE CHECKOUT account or a debit or credit card or other payment mechanism 124. ... Finally, the user may confirm that the displayed information is accurate and may cause a **closing of the transaction** by selecting checkout button 120.”); *id.*, ¶70 (“The checkout module may **manage closing out of a transaction**”). Dierks also discloses that the user’s account is debited for the transaction. EX1004, ¶51 (“When the purchaser

chooses to complete the purchase, their **account is debited** by the **payment server 114**.⁹...”). EX1002, ¶¶149–150.

While Figure 1E above shows a domain address of <https://www.mycheckout.com>, Dierks discloses that the secure checkout can be *displayed with the vendor’s web site*, similar to the shopping cart 116 (part of the “*electronic commerce store*”). EX1004, ¶44 (“the **checkout** may also be **performed in the same content window** of a browser **as the vendor’s web site**, by mechanisms similar to those described above”). EX1002, ¶151.

Dierks discloses to display the checkout in the same window as the vendor’s web site “to **avoid** taking a purchaser on a ‘**round trip**’ to the payment processor’s web page and back to the vendor’s page, or opening another window or tab in a browser.” EX1004, ¶34. Dierks teaches to not put the checkout in a new window because the new window “would likely have a different interface than does the web site of the vendor, and *could thus be confusing to the purchaser*, or at least take away from the immersive aspect of the shopping transaction that is more readily available *when one company handles both the merchandise and payment processing*.” *Id.* EX1002, ¶152.

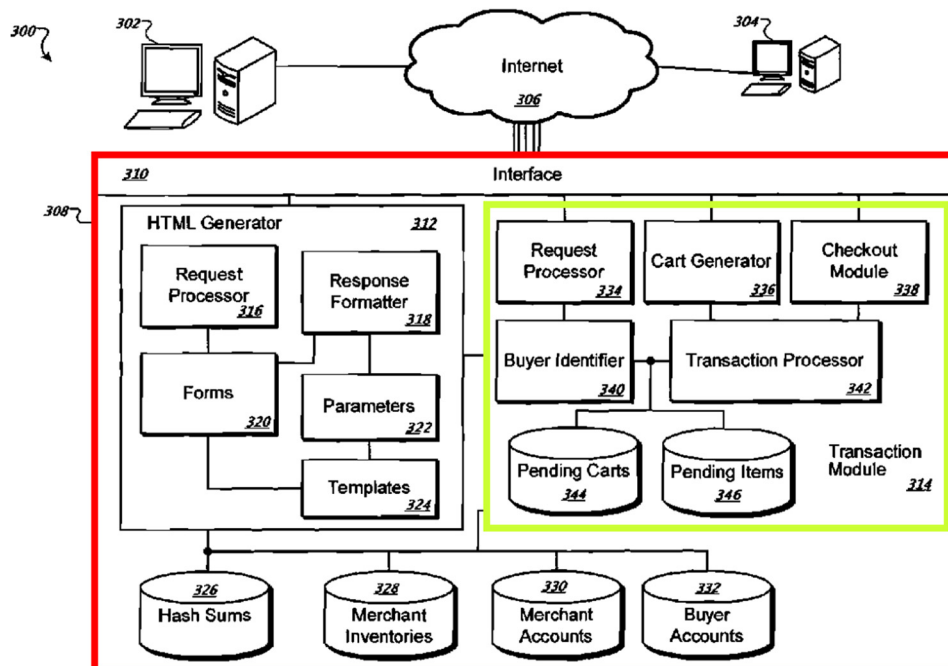
⁹ See n. 4.

i) 1[h]: “wherein the electronic commerce store is operated by the buy button provider.”

Dierks discloses element 1[h]. EX1002, ¶158.

Dierks discloses that the shopping cart and secure checkout (“the electronic commerce store”), discussed above for elements 1[f] and 1[g], is operated by the payment processing system 308 (“the buy button provider”). EX1002, ¶154.

Dierks discloses that the **payment processing system 308**, shown below in Figure 3, includes a **Transaction Module 314** having a “Transaction Processor 342,” “Pending Carts [database] 344,” “Cart Generator 336,” and a “Checkout Module 338.” EX1004, Fig. 3. EX1002, ¶155.



EX1004, FIG. 3

The transaction processor 342 operates the shopping cart by opening and populating the shopping cart. EX1004, ¶67 (“transaction processor 342 may open a new **shopping cart** for the pair, and may **populate the cart** with information associated with the selected control.”). Also, the “pending carts database 344” stores a “pending **shopping cart** table” that “tracks the transaction” and “may include fields for a ConsumerID, a MerchantID, an AnonymousID, and a CartID.” *Id.* Further, A POSITA would have understood that the “Cart Generator 336” is related to operation of the shopping cart at least because element 336 is called a “*cart*” generator. EX1004, Fig. 3; *see also id.*, ¶¶47, 49, 58,59, 67, 68, and 74. EX1002, ¶156.

Further, Dierks discloses that the “[c]**heckout module** 338 may be accessed when a purchaser decides to finish a transaction and buy or edit items in his or her shopping cart.” EX1004, ¶70. The checkout module 338 manages “**closing out of a transaction**, such as by computing a total cost, any transaction fees, taxes, and shipping/handling, in a conventional manner.” *Id.*; *see also id.*, ¶¶50, 51. EX1002, ¶157.

Accordingly, the payment processing system 308 (“*the buy button provider*”) in Dierks operates the shopping cart and secure checkout (part of “*the electronic commerce store*”) to disclose element 1[h]. EX1002, ¶158.

2. Claim 2¹⁰

Dierks discloses the additional limitation of claim 2. EX1002, ¶159.

Dierks discloses that the “add to cart” button may be generated using JavaScript. EX1004, ¶37 (“the *button 105 is generated using JavaScript* (or, e.g., Jscript or VBScript) associated with the vendor’s underlying web page that generates display 104.”). EX1002, ¶160.

A POSITA would have understood that JavaScript is used with an HTML document by employing a <script> tag at the beginning and end of the JavaScript code. This follows the following disclosure of JavaScript in Dierks used for the “add to cart” button 105. As shown, the JavaScript uses the <script> tag that allows the code to be embedded into an HTML document. EX1002, ¶161.

```
<script
  type="text/javascript"
  src="https://www.mycheckout.com/shoppingcart.js?sellerId=8675309">
</script>
<span id="mycheckout_status_bar"></span>
<script type="text/javascript"
  src="https://www.cmycheckout.com/shoppingcart.js?sellerId=8675309">
</script>
<span id="mycheckout_status_bar"></span>
```

EX1004, ¶37

¹⁰ See Appendix of Claim Listing.

The JavaScript may be “stored by the payment processor and referenced by code on the vendor web site.” EX1004, ¶37. A POSITA would have understood this to mean that JavaScript embedded into an HTML document on the vendor’s web site provides a link to additional JavaScript code stored on the “payment processor.” This additional JavaScript code is fetched from the payment processor and provided to the browser to effectuate the “add to cart” button 105. EX1002, ¶162.

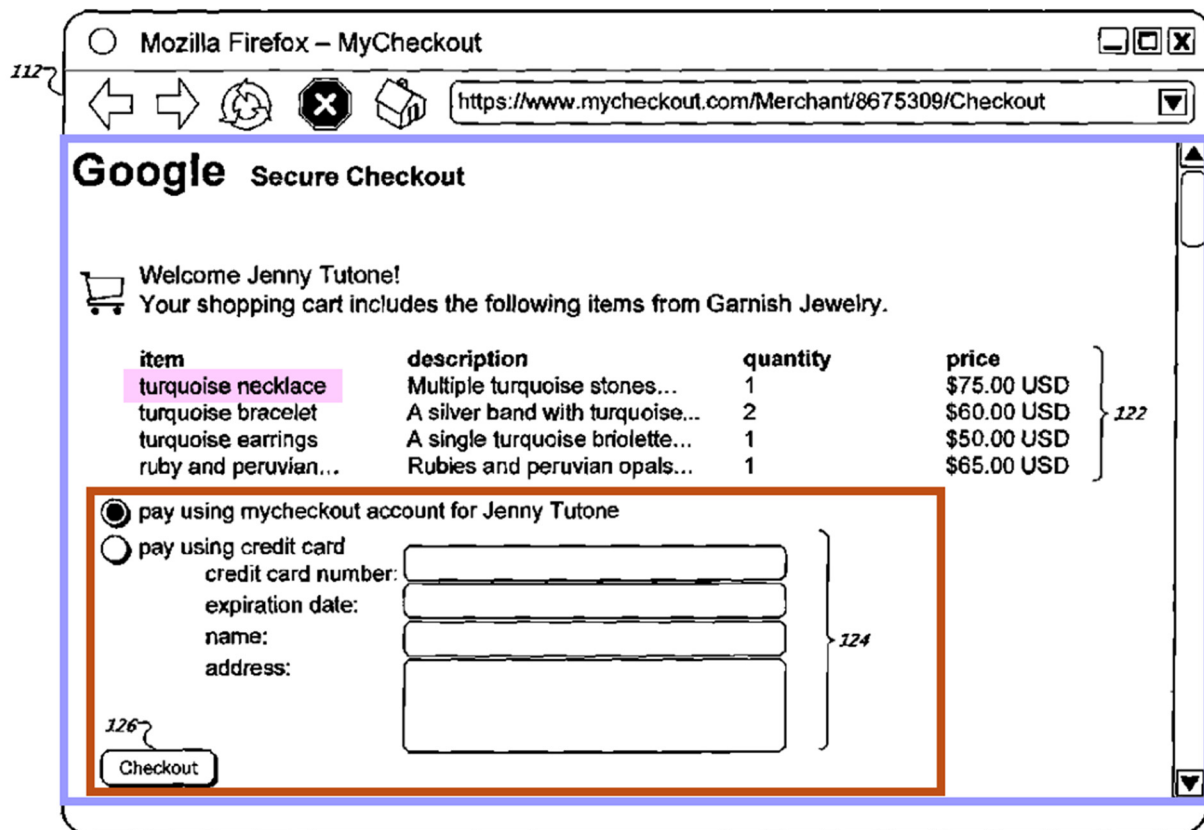
This fetching occurs after the browser parses the HTML document and reaches the “**src**” attribute, shown above, that identifies the location of additional code using the shown uniform resource locator (“URL”), i.e., <https://www.mycheckout.com/shoppingcart.js?sellerId=8675309>. A POSITA would have understood that the “**src**” attribute is used to provide a path to an external file or resource linked to the HTML document. EX1002, ¶163.

The JavaScript of Dierks is not web site dependent. It can be embedded within an HTML document of a first web site to provide the “add to cart” button 105, shown in Fig. 1C of Dierks. The JavaScript could also be embedded into an HTML document for a different web site. For each web site, the browser will react the same by accessing the URL and obtaining the additional code that establishes the “add to cart” button 105. Accordingly, the code for the “add to cart” button (“*the embeddable buy button*”) is configured to be embedded into different web sites. EX1002, ¶164.

3. Claim 3

Dierks discloses the additional limitation of claim 3. EX1002, ¶166.

As discussed for element 1[g], Dierks discloses a **secure checkout** having a **payment mechanism 124** that facilitates payment processing related to the purchase of an item. In Fig. 1E, an item selected for purchase is a “**turquoise necklace**.” EX1002, ¶167.



EX1004, FIG. 1E

Dierks discloses that the user can “**select a payment method**, such as through a GOOGLE CHECKOUT account or a debit or credit card or other payment mechanism 124.” EX1004, ¶48. As discussed for element 1[g], Dierks discloses

that the secure checkout can be displayed within the vendor's web site, similar to the shopping cart 116. *Id.*, ¶34. The secure checkout, including the **payment mechanism 124**, provides a checkout “*screen*” since it is configured as a display. *Id.*, ¶48 (“At **display 112**, the user/purchaser has selected a ‘proceed to checkout’ button.¹¹...”); *see also id.*, ¶98. EX1002, ¶168.

Thus, Dierks discloses that the **payment mechanism 124** of the **secure checkout** (part of “*the electronic commerce store*”) includes a checkout screen that allows the purchase of an item using GOOGLE CHECKOUT. Dierks discloses that GOOGLE CHECKOUT is a transaction processing service for paying a vendor. EX1004, ¶3 (“person-to-person **transactions** may be paid for through open payment systems such as **GOOGLE CHECKOUT**¹² or PAYPAL, such as when the person providing the product or service (the merchant or vendor) does not accept credit cards.”); *see also id.*, ¶¶6, 33, 37, 96; *cf.* EX1001, 18:8–12 (“In one alternative embodiment, the alternative **transaction processing service** can be any one of

¹¹ The “proceed to checkout” button is shown in Figure 1D, which the user presses to access display 112. EX1002, ¶168.

¹² A POSITA would have understood that “Google Checkout” refers to “**Google Secure Checkout**” shown in display 112 of Figure 1E because of the overlapping names. EX1002, ¶169.

Authorize.Net, **Google Checkout**, Storm Pay, PayPal, or other Internet service for transaction processing.”). EX1002, ¶169.

Further, Dierks discloses that Google Checkout (“*the transaction processing service*”) is operated by the payment processing system 308 (“*the buy button provider*”). EX1004, ¶48 (“At **display 112**, the user/purchaser has selected a ‘proceed to checkout’ button, and is redirected to a web page **operated by the payment server 114**¹³.”). EX1002, ¶170.

Moreover, as discussed for element 1[h], the payment processing system 308 of Dierks includes a Transaction Module 314 having a “[c]**heckout module 338** [that] may be accessed when a purchaser decides to *finish a transaction and buy* or edit *items* in his or her shopping cart.” EX1004, ¶70. The checkout module 338 manages “*closing out of a transaction*, such as by computing a total cost, any transaction fees, taxes, and shipping/handling, in a conventional manner.” *Id.* EX1002, ¶171.

Accordingly, Dierks discloses that the payment mechanism 124 of the secure checkout (part of the “*the electronic commerce store*”) includes a checkout screen that allows a purchase of a turquoise necklace (“*the item*”) using Google Checkout

¹³ See n. 4.

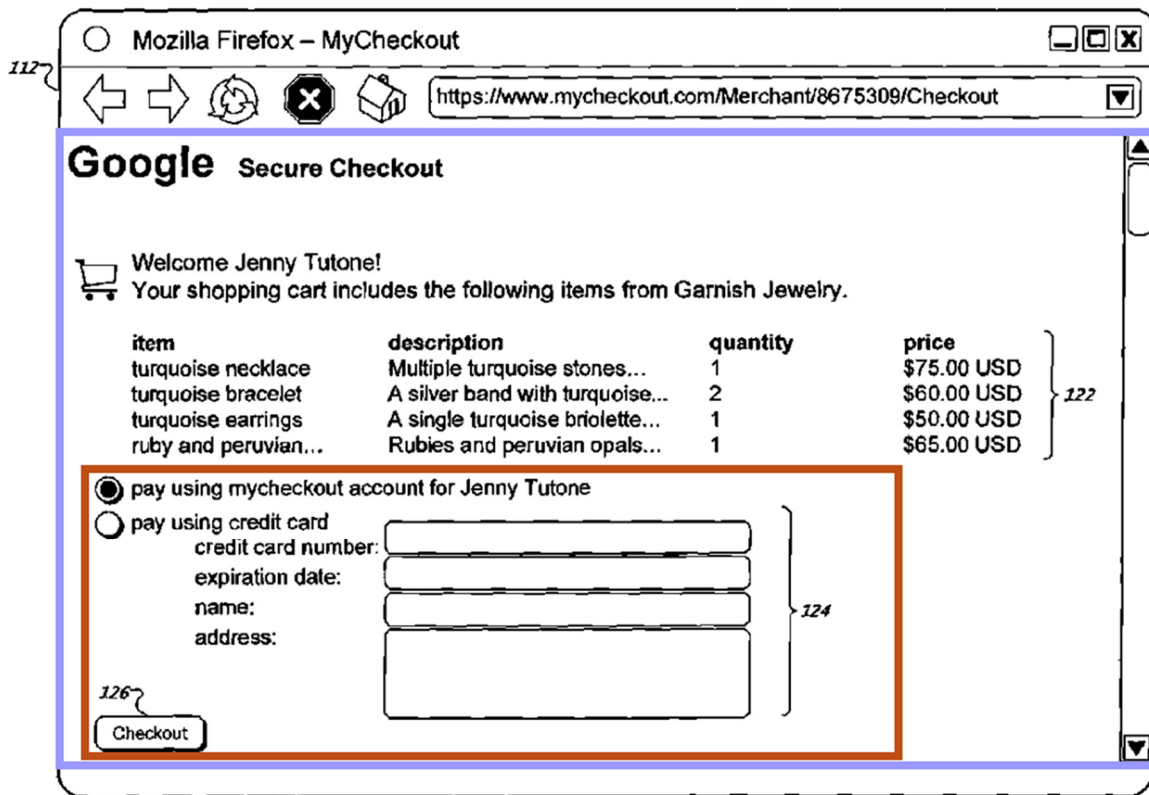
(“*a transaction processing service*”) operated by the payment processing system 308 (“*the buy button provider*”). EX1002, ¶172.

4. Claim 4

Dierks teaches claim 4. EX1002, ¶174.

As discussed for claim 3, Dierks discloses that “the electronic commerce store includes a checkout screen that allows a purchase of the item using a transaction processing service.” Dierks also discloses a transaction processing service “operated by a third-party server,” as required by claim 4. EX1002, ¶174.

As shown below in Figure 1E, Dierks discloses that the **payment mechanism 124** of the **secure checkout** allows purchases to be made using a “my checkout account” and, separately, by using a “credit card.” EX1004, Fig. 1E; *see also id.*, ¶48 (“The user may also be permitted to select a payment method, such as through a GOOGLE CHECKOUT account or a **debit or credit card** or other payment mechanism 124.”); *id.*, ¶83 (“submit a charge to a **credit card**”); ¶86 (“charge to a **credit card** of the buyer, or to a **bank account** or **other account** associated with the buyer.”). EX1002, ¶175.



EX1004, FIG. 1E

A POSITA would have understood that debit/credit cards are issued by entities that facilitate transactions, such as credit unions and banks. It was well known before the '785 patent priority date that conventional credit/debit card payment authorization systems are third-party banking institutions that issue credit or debit cards. *See, e.g.*, EX1005 (Johnston), ¶300 (“credit card vendors”); EX1006 (Babaria), ¶22 (“a third-party credit provider”). EX1002, ¶176.

Dierks does not indicate that the payment processing system 308 is a credit provider. Thus, a POSITA would have understood that Dierks’s disclosure of using a credit card as payment, instead of a “my checkout account” would use a transaction

processing service operated by a third-party organization that manages credit cards, such as a bank or credit union. It was well known before the '785 patent priority date that transaction processing services for credit cards used servers (“*third-party server*”). *See, e.g.*, EX1019, ¶34 (“The system 100 further includes **credit provider server 112** for approving transactions conducted with a **credit card**. ...”); EX1020, 2:60–3:2 (“Payment processing system 102 ... operating on one or more general purpose **server computing platforms**.”). EX1002, ¶177.

Accordingly, Dierks discloses that the secure checkout (part of the “*the electronic commerce store*”) includes a checkout screen that allows a purchase of a turquoise necklace (“*the item*”) using a transaction processing service operated by a server of a credit card company (“*the third-party server*”). EX1002, ¶178.

5. Claim 5

Dierks discloses the additional limitation of claim 5. As discussed for element 1[a], the item is turquoise earrings, i.e., a physical good. EX1002, ¶¶180–181.

6. Claim 6

Dierks and Johnston render obvious claim 1, as discussed above. Johnston discloses the additional limitation of claim 6 because the “*representation*” is an image (“*a static image*”), as discussed for element [1a]. EX1021, M.P.E.P. §2117(I) (“A ‘Markush’ claim recites a list of **alternatively** useable members.”). EX1002, ¶¶182–183.

7. Independent Claim 7

a) 7[preamble]¹⁴

Even if the preamble is limiting, Dierks discloses it for at least the same reasons discussed above for element 1[preamble]. EX1002, ¶184.

b) 7[a]

Dierks discloses element 7[a] for at least the same reasons discussed above for element 1[a]. EX1002, ¶185.

c) 7[b]

Dierks discloses element 7[b] for at least the same reasons discussed for element 1[a]. EX1002, ¶186.

As discussed for element 1[a], Dierks discloses a “Merchant Inventories” database 328 used for storing inventories of items to be offered for sale on the electronic commerce store. The “*item inventory*” discussed for element 1[a] is an “*inventory list*” because it identifies names of items offered for sale. EX1004, ¶53 (“an **inventory database** for goods to be offered for sale by the seller. The seller in particular may identify **names** and descriptions for **items to be sold**”). EX1002, ¶187.

¹⁴ See Appendix of Claim Listing.

Dierks also discloses that the database is communicatively coupled to the first server. Limitation 7[b] does not require the “database” and the “first server” to have particular structures. As discussed above in Section VIII.A.1.b, for limitation 1[a], Dierks discloses that the payment processing system 308 comprises “**one or more** web servers and other such **hardware**, or may **take other appropriate forms**.” EX1004, ¶62. EX1002, ¶188.

A POSITA would have understood that the Merchant Inventories database 328 in Dierks includes the actual data stored in the database, *i.e.*, merchant inventories data, and when the payment processing system 308 of Dierks is configured with one server, that server would have access to the merchant inventories data and is therefore communicatively coupled to the Merchant Inventories database 328 shown in Figure 3 of Dierks. EX1002, ¶189.

If PO argues that the database is external to the first server, which is not required by the claim, then such an interpretation would have also been taught because Dierks discloses that the payment processing system may comprise multiple configurations, *i.e.*, “one or more web servers and other such hardware, or may take other appropriate forms.” EX1004, ¶62. Indeed, it was well known before the priority date of Dierks to provide databases external to web servers. *See, e.g.*, EX1024 (Kossmann) Figures 2-4 showing multiple known implementations where

databases are external to and in communication with web servers. EX1024, Figs. 2–4; EX1032. EX1002, ¶190.

Moreover, Dierks teaches that the database for storing the vendor’s inventory list may be external to the payment processing server. *See, e.g.*, EX1004, ¶41 (“where the vendor stores information relating to the vendor’s inventory using a system *associated with* payment server 114, such as **GOOGLE BASE**, the payment server 114 can also *check such a database* for availability of the selected goods.”); *id.*, ¶86 (“In addition, *where an inventory of the seller is available to the payment processing system (e.g., through Google Base)*, the inventory may be decreased for each of the purchased items. ...”). EX1002, ¶191.

As noted above, Dierks refers to Google Base as being a database system that is “associated with” the payment server 114¹⁵ (¶41) and “available to the payment process system” (¶86). A database system used by the vendor that is “associated with” a server or “available to” a payment processing system suggests a database that is external to, but in communication with, the server or payment processing system. Also, the Merchant Inventories database 328 in Figure 3 of Dierks is shown as being communicatively coupled with the HTML Generator 312 and Transaction Module 314. Thus, in view of the configuration options disclosed by Dierks, a

¹⁵ *See* n. 4.

POSITA would have been taught to provide the merchant inventory list in a database that is external to and in communication with the first server. EX1002, ¶192.

d) 7[c]

Dierks and Johnston render obvious element 7[c] obvious for at least the same reasons discussed above for elements 1[a] and 7[b]. EX1002, ¶194.

Dierks’s disclosure of “an embeddable buy button builder **executing on a first server**” for element 1[a] discloses “an embeddable buy button builder **executed by the first server**,” as recited in element 7[c]. EX1002, ¶¶195–196.

Dierks’s teaching of “receiving, a selection of an item from an item inventory and a representation of the item” for element 1[a], likewise teaches “to receive a selection of an item from the inventory list and a representation of the item,” as recited in element 7[c]. While element 1[a] recites “an item inventory” and not an “inventory list,” the analysis for element 7[b] describes how the “item inventory” is an “inventory list.” EX1002, ¶197.

Further, Dierks discloses that the embeddable buy button builder is “**configured** to receive” a selection, as recited in element 7[c], for the same reasons that Dierks discloses “receiving, by an embeddable buy button builder...a selection,” as discussed for element 1[a]. EX1002, ¶198.

e) 7[d]

Dierks discloses element 7[d] for at least the same reasons discussed above for element 1[b]. As discussed for element 1[b], the HTML Generator 312 (“*the embeddable buy button builder*”) generates the code for the “add to cart” button and therefore generates the “add to cart” button (“*the embeddable buy button*”). The HTML Generator 312 is “configured” to generate the code for the “add to cart” button by virtue of the HTML Generator 312 generating the code. EX1002, ¶200.

f) 7[e]

Dierks discloses element 7[e] for at least the same reasons discussed above for element 1[d]. As discussed for element 1[d], Dierks discloses “embedding, by the buy button provider, the embeddable buy button into a web site.” Since Dierks discloses that the buy button provider embeds the embeddable buy button into a web site, the *embeddable buy button is configured to be embedded into the web site by the buy button provider*. EX1002, ¶¶201–202.

g) 7[f]

Dierks discloses element 7[f] for at least the same reasons discussed above for element 1[c]. As discussed for element 1[c], Dierks discloses “wherein the embeddable buy button is configured to display the representation of the item, and a hyperlink.” EX1002, ¶203.

The “add to cart” button (“*the embeddable buy button*”) in Dierks “*includes*” the hyperlink, as required by element 7[f]. When the “add to cart” button 105 is pressed, it causes an HTTP request and, therefore, includes a hyperlink. EX1004, ¶38 (“[s]election of the *button 105 may cause an HTTP request*, in the form of an XMLHttpRequest (XHR), to be made of payment server 114¹⁶, as indicated by Arrow A.”). EX1002, ¶204.

h) 7[g]

Dierks discloses element 7[g] for at least the same reasons discussed above for element 1[e]. EX1002, ¶206.

i) 7[h]

Dierks discloses element 7[h]. The payment processing system 308 (“*the buy button provider*”) in Dierks controls operation of the “add to cart” button (“*the embeddable buy button*”) on the web site. EX1002, ¶¶207–208.

First, as discussed for element 1[b], the payment processing system 308 includes the HTML Generator 312 that generates the code for the “add to cart” button 105. The HTML Generator 312 controls operation of the “add to cart” button 105 by virtue of generating the code. EX1002, ¶209.

¹⁶ See n. 4.

Second, Dierks discloses that “[s]election of the button 105 may cause an HTTP request, in the form of an XMLHttpRequest (XHR), **to be made of payment server 114**¹⁷.” EX1004, ¶38. Dierks then discloses that the “payment server 114 may then **parse the HTTP request** to obtain an **identifier for the purchaser**, and also to **identify the vendor and relevant payment parameters**.” *Id.*, ¶40. Thus, the payment processing system 308 is controlling operation of the “add to cart” button by acting on the HTTP request. EX1002, ¶210.

Third, Dierks discloses that the payment server 114 passes information back to the purchaser based on operation of the “add to cart” button. EX1004, ¶42 (“the **changes may be passed back to the client device of the purchaser**, as shown by Arrow B ... **the payment server may pass information** reflecting a **description of the selected item(s), their quantity, their price, and a total price of goods** in the purchaser’s shopping cart.”); *see also id.*, ¶43. EX1002, ¶¶211-212.

Accordingly, the payment processing system 308 (“*the buy button provider*”) controls operation of the “add to cart” button (“*the embeddable buy button*”) on the web site because (1) the payment processing system 308 creates the “add to cart” button and therefore controls the flow of information that occurs when the “add to cart” button is pressed, (2) the payment processing system 308 parses the

¹⁷ See n. 4.

information received after the “add to cart” button is pressed, and (3) the payment processing system 308 controls the flow of information back to the web site to update the shopping cart based on the pressing of the “add to cart” button. EX1002, ¶213.

j) 7[i]

Dierks discloses element 7[i]. As discussed for element 1[h], the electronic commerce store, including the shopping cart and secure checkout, is operated by the payment processing system 308 (“*the buy button provider*”). EX1002, ¶¶215–216.

As discussed for element 7[f], the “add to cart” button 105 includes the hyperlink and when the “add to cart” button 105 is pressed, an HTTP request is sent to the payment server 114¹⁸. EX1004, ¶38 (“[s]election of the **button 105 may cause an HTTP request** ... to be made of payment server 114”). EX1002, ¶217.

As discussed for element 7[h], selection of the “add to cart” button, and therefore selection of the hyperlink, causes the payment processing system 308 to display the shopping cart on the web site. EX1004, ¶43 (“Display 106 shows the **result of a user's selection of button 105** (see FIG. 1C). Specifically, a **shopping cart object shown. ...**”). EX1002, ¶218.

Accordingly, the payment processing system 308 controls operation of the shopping cart (part of “*the electronic commerce store*”) by sending code to display

¹⁸ See n. 4.

the shopping cart upon selection of the hyperlink. EX1004, ¶43 (“The iframe 116 [shopping cart object] is generated using the information received from the payment server 114¹⁹.”); *id.*, ¶43; Fig. 3 (“Cart Generator”). EX1002, ¶219.

The payment processing system 308 is also responsible for controlling operation by updating the shopping cart with information, including a description of the selected items, a quantity of the items, and a price of the items. EX1004, ¶42 (“the **payment server may pass information** reflecting a **description of the selected item(s), their quantity, their price, and a total price of goods** in the purchaser’s shopping cart.”). EX1002, ¶220.

Further, as discussed for element 1[h], the payment processing system 308 (“*the buy button provider*”) operates and therefore controls the secure checkout, including the “[c]heckout module 338 [that] may be accessed when a purchaser decides to finish a transaction and buy or edit items in his or her shopping cart.” EX1004, ¶70. EX1002, ¶221.

Accordingly, the payment processing system 308 of Dierks controls operation of the electronic commerce store on the web site upon selection of the hyperlink. EX1002, ¶222.

¹⁹ See n. 4.

8. Claim 8

Dierks discloses the additional limitation of claim 8. As discussed for claim 3, Dierks discloses Google Checkout (“*a transaction processing service*”) operated by the payment processing system 308 (“*the buy button provider*”). EX1002, ¶¶224-225.

Google Checkout is configured to process a payment initiated via the electronic store, such as being initiated through the display 112 of Figure 1E, by debiting the purchaser’s account. EX1004, ¶33 (“user interaction with an on-line payment system, such as GOOGLE CHECKOUT”); *id.*, ¶48 (“select a payment method, such as through a GOOGLE CHECKOUT”); ¶51 (“When the purchaser chooses to complete the purchase, their **account is debited by the payment server** 114.²⁰...”). EX1002, ¶226.

9. Claim 9

Dierks discloses the additional limitation of claim 9. EX1002, ¶228.

Dierks discloses that the payment processing system 308 (“*the buy button provider*”) is configured to execute the shopping cart (part of the “*electronic commerce store*”) and the secure checkout with payment mechanism 124 (also part of “*the electronic commerce store*”). The payment processing system 308 executes

²⁰ See n. 4.

the electronic commerce store by establishing and controlling the shopping cart and secure checkout, as discussed for element 7[i], and by updating the shopping cart after the user selects the “add to cart” button 105, as also discussed for element 7[i]. EX1002, ¶229.

Further, the shopping cart and secure checkout are executed by the payment processing system 308 when the user selects the “proceed to checkout” button 118 in the shopping cart 116, as discussed for elements 1[g] and 7[i]. EX1004, ¶44. Selecting the “proceed to checkout” button 118 causes the payment processing system 308 to execute functions, including (1) generating the secure checkout display, (2) checking that database fields have been updated, and (3) closing out the transaction. *Id.*, ¶48 (“web page operated by the payment server 114²¹... the payment server 114 may check its database fields that have been updated through the purchaser’s selection of ‘add to cart’ buttons ... permit closing out of the transaction ... changes will be reflected in associated databases managed by the payment processor.”). EX1002, ¶230.

²¹ See n. 4.

The payment processing system 308 also executes the electronic commerce store by facilitating and processing payment of the item, as discussed for claim 8. *Id.*, ¶231.

Accordingly, Dierks discloses that the payment processing system 308 (“*the buy button provider*”) executes the shopping cart and secure checkout (“*the electronic commerce store*”) by providing the aforementioned functionality. *Id.*, ¶232.

10. Claim 10

Dierks discloses the additional limitation of claim 10. EX1002, ¶234.

As discussed for element 7[d], the HTML generator 312 (“*the embeddable buy button builder*”) is configured to generate the “add to cart” button (“*the embeddable buy button.*”). A POSITA would have understood that HTML stands for hypertext-markup-language. Dierks also discloses that the HTML is “code” and is configured to be inserted into the web site. EX1004, ¶62 (“The **HTML generator** ... is configured to ... **generate code**, such as **HTML**. ...”); *id.*, ¶64 (“the **HTML generator 312 may produce mark-up code** for the user that the user can employ to provide payment processing services **on their web site.**”). EX1002, ¶235.

Figure 4 of Dierks below illustrates an example of hypertext-markup-language generated by the HTML generator 312, including **code 418** for the “**add to cart**” button.

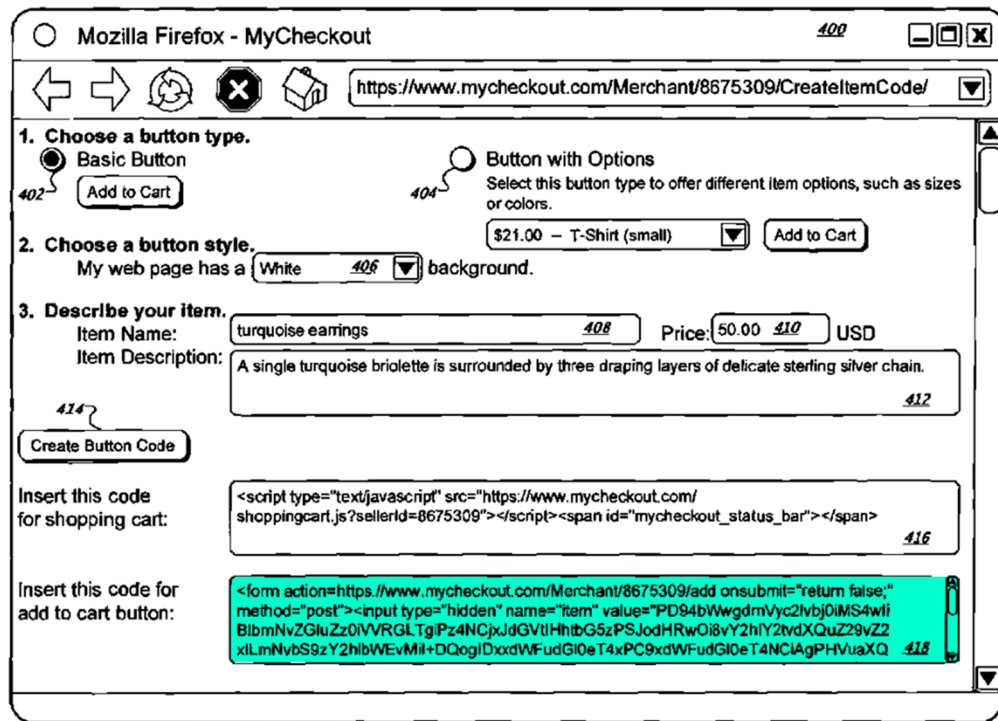


FIG. 4

EX1004, FIG. 4

EX1004, ¶74 (“code for generating an add to cart button, displayed in area 418, may take the form of HTML code ...”). EX1002, ¶236.

The hypertext-markup-language code for the “add to cart” button is configured to be inserted in the web site. EX1004, ¶74 (“The code ... **can be cut-and-pasted into the user’s web page.**”); *see also* analysis of claim 2, and elements 1[d], 7[e], that discuss how Dierks discloses to embed code into the web site. EX1002, ¶237.

Thus, Dierks discloses that the HTML Generator 312 (“*the embeddable buy button builder*”) generates hypertext-markup-language code configured to be inserted into the web site. EX1002, ¶238.

11. Claim 11

Dierks and Johnston render claim 11 obvious for at least the same reasons discussed above for claim 6. EX1002, ¶240.

12. Claim 12

Dierks discloses claim 12 for at least the same reasons discussed above for claim 2. Therefore, Dierks and Johnston render obvious claim 12. EX1002, ¶241.

13. Claim 13

Dierks discloses the additional limitation of claim 13 for at least the same reasons discussed for claims 3 and 17. EX1002, ¶242.

As discussed for claim 3, Dierks discloses that “*the electronic commerce store includes a checkout screen.*” As discussed for claim 17, Dierks discloses that “*the checkout screen is configured to facilitate the purchase of the item.*” *Id.*, ¶243.

Therefore, Dierks and Johnston render obvious claim 13. *Id.*, ¶244.

14. Independent Claim 14

a) 14[preamble]

Dierks discloses the preamble for at least the same reasons discussed above for element 1[preamble]. EX1002, ¶245.

b) 14[a]

Dierks and Johnston render obvious element 14[a] for at least the same reasons discussed above for element 1[a]. Element 14[a] differs from element 1[a] by reciting “*a selection of an item to sell on a web site*” (element 14[a]) instead of “*a selection of an item from inventory*” (element 1[a]). As discussed for element 1[a], the turquoise earrings (“*an item*”) are selected to sell on a web site. EX1004, ¶73 (“displayed on the vendor’s web page.”). EX1002, ¶¶246–247.

c) 14[b]

Dierks discloses element 14[b] for at least the same reasons discussed above for element 1[e]. EX1002, ¶248.

d) 14[c]

Dierks discloses element 14[c]. As discussed for element 1[b], Dierks discloses “*generating, by the first server, the embeddable buy button.*” As discussed for claims 2 and 10, Dierks discloses that the embeddable buy button contains JavaScript. EX1004, ¶37 (“the **button 105** is generated using **JavaScript**”). Code written using JavaScript is “software” since the code includes instructions used by a computer to execute a task. EX1002, ¶¶249–250.

Dierks discloses the following JavaScript code for the “add to cart” button (“*the embeddable buy button*”). EX1002, ¶251.

```
<script
  type="text/javascript"
  src="https://www.mycheckout.com/shoppingcart.js?sellerId=8675309">
</script>
<span id="mycheckout_status_bar"></span>
<script type="text/javascript"
  src="https://www.cmycheckout.com/shoppingcart.js?sellerId=8675309">
</script>
<span id="mycheckout_status_bar"></span>
```

EX1004, ¶37

As discussed for claim 2, A POSITA would have understood that JavaScript is used with HTML by employing the `<script>` tag at the beginning and end of the JavaScript code. The source or “**src**” attribute (shown above) is used to access JavaScript stored at the payment processor, *i.e.*, stored at the payment processing system 308. EX1004, ¶37 (“The JavaScript may be **stored by the payment processor and referenced by code on the vendor web site. ...**”); ¶78. EX1002, ¶252.

The JavaScript code shown above as being bound by the `<script>` tag is executed by the first server because the first server (1) receives an HTTP request, for additional code, that is triggered by the above shown JavaScript code, and (2) provides the web site with the additional code used to display the shopping cart.

EX1004, ¶38 (“**Selection of the button 105** may cause an **HTTP request**, in the form of an XMLHttpRequest (XHR), to be made of **payment server 114**.”²²...); *id.*, ¶43 (“Display 106 shows the result of a user’s selection of button 105 (see FIG. 1C). Specifically, a shopping cart object ... has been created on display 106.”). EX1002, ¶253.

Accordingly, Dierks discloses generating, by a first server, the “add to cart” button (“*the embeddable buy button*”) containing JavaScript code (“software”) configured to be executed by the first server on the web site. *Id.*, ¶¶254–255.

e) 14[d]

Dierks discloses element 14[d] for at least the same reasons discussed above for elements 1[d] and 7[f]. As discussed for element 1[d], Dierks discloses “embedding, by the buy button provider, the embeddable buy button into a web site.” As discussed for element 7[f], Dierks discloses that “the embeddable buy button includes a hyperlink.” EX1002, ¶¶256–257.

f) 14[e]

Dierks discloses element 14[e]. EX1002, ¶258.

As discussed for element 14[c], the “add to cart” button (“*the embeddable buy button*”) encompasses JavaScript code (“*the software*”) that is embedded in the web

²² See n. 4.

site and is configured to display the shopping cart (part of “*the electronic commerce store*”) within the web site. *Id.*, ¶259.

As discussed for element 14[d], Dierks discloses that the “add to cart” button (“*the embeddable buy button*”) includes a hyperlink. When the “add to cart” button 105 is pressed, it sends an HTTP request to the payment processing system. EX1004, ¶38. Further, as discussed for element 1[f], “*the electronic commerce store is displayed within the web site upon selection of the hyperlink.*” The shopping cart object 116 is displayed within the web site upon selection of the “add to cart” 105 button that includes the hyperlink. EX1004, ¶43. The payment processing system 308 executes the software by receiving the HTTP request and causing the shopping cart to be displayed. *Id.*, ¶¶43, 67. EX1002, ¶¶260–262.

Thus, Dierks discloses executing, by the payment processing system 308 (“*the buy button provider*”), the JavaScript code (“*the software*”) upon selection of the hyperlink, wherein the software is configured to display the shopping cart object (part of the “*electronic commerce store*”) within the web site, as required by element 14[e]. *Id.*, ¶¶263–264.

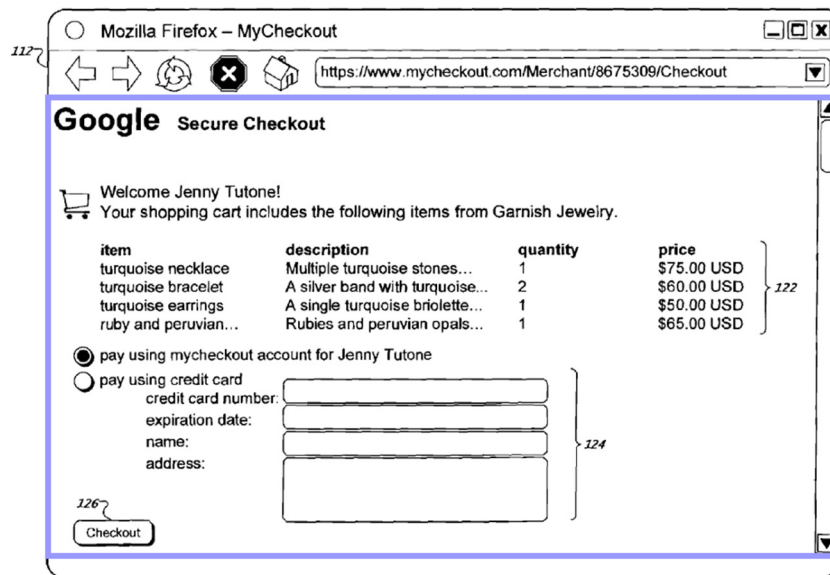
15. Claim 15

Dierks discloses the additional limitation of claim 15 for at least the same reasons discussed above for claim 2. EX1002, ¶265.

Therefore, Dierks and Johnston render obvious claim 15. *Id.*, ¶266.

16. Claim 16

Dierks discloses the additional limitation of claim 16. As discussed for claim 3, Dierks discloses that **the electronic commerce store includes a checkout screen** that allows a purchase of the item using a transaction processing service **operated by the payment processing system 308** (“*the buy button provider*”). Dierks also discloses that the payment processing system 308 operates the display 112 that includes the checkout screen shown below in Figure 1E. EX1004, ¶48 (“At **display 112**, the user/purchaser has selected a ‘proceed to checkout’ button, and is redirected to a web page **operated by the payment server 114**²³.”); *see also id.* ¶20 (“FIGS. 1A-1E show example **screen shots**”). EX1002, ¶¶267–268.



EX1004, FIG. 1E

²³ See n. 4.

17. Claim 17

Dierks discloses the additional limitation of claim 17. As discussed for claim 3, Dierks discloses that “the electronic commerce store includes a **checkout screen that allows a purchase of the item using a transaction processing service operated by the buy button provider.**” Dierks’s disclosure of a “checkout screen *that allows a purchase of the item*” also discloses that “the checkout screen is *configured to facilitate the purchase of the item.*” EX1002, ¶269.

The checkout screen of Dierks facilitates the purchase of the item using the payment mechanism 124, described for claim 3, that provides a way for the user to purchase the item. EX1004, ¶48 (“**select a payment method**, such as through a GOOGLE CHECKOUT account or a debit or credit card or other **payment mechanism 124.**”). EX1002, ¶¶270–271.

18. Claim 18

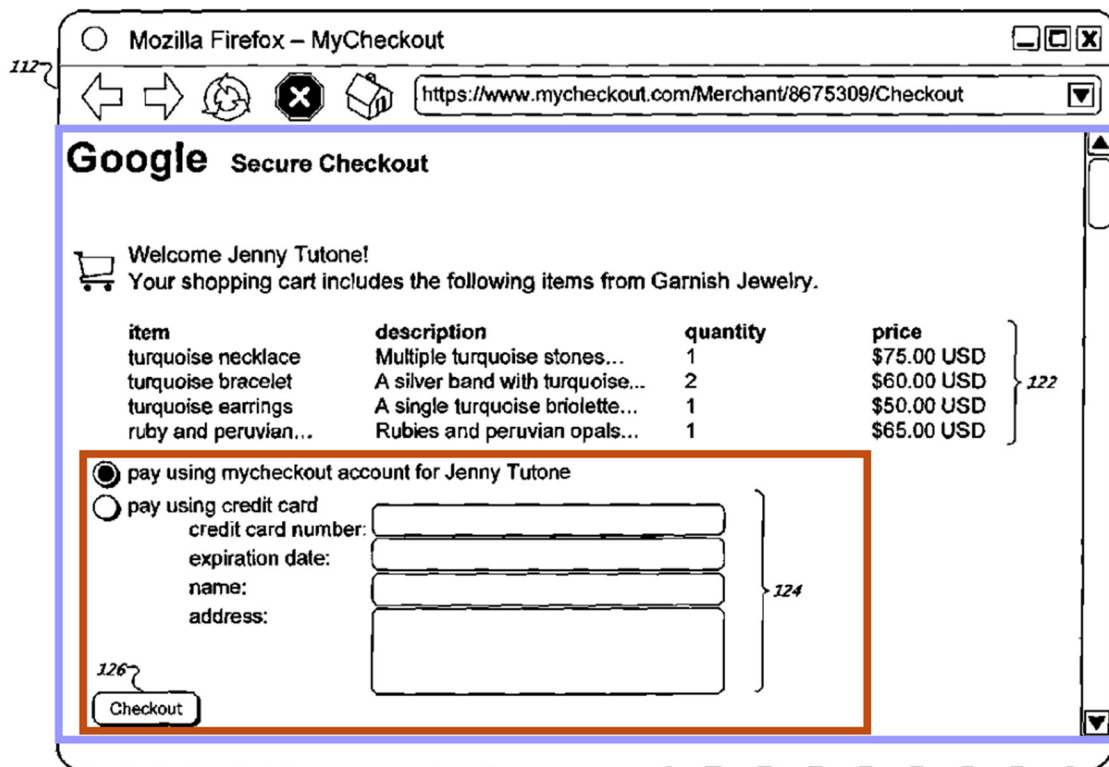
Dierks and Johnston render claim 18 obvious for at least the same reasons discussed above for claim 6. EX1002, ¶¶272–273.

19. Claim 19

Dierks discloses the additional limitation of claim 19 for at least the same reasons discussed above for claim 5. EX1002, ¶¶274–275.

20. Claim 20

Dierks discloses the additional limitation of claim 20. As discussed for claim 17, Dierks discloses a secure checkout screen configured to facilitate the purchase of the item. Dierks's **checkout screen** is shown below in Figure 1E with the **payment mechanism 124** that provides multiple payment options, including using a “mycheckout” account and a credit card. EX1002, ¶¶276–277.



EX1004, FIG. 1E

The “pay using mycheckout account” option is shown as being selected in Figure 1E. A POSITA would have understood that the option to pay using “mycheckout” does not require entering credit card information because payment

information was previously stored. This is further supported by Dierks disclosing that the payment processing system 308 stores payment information for future use. EX1004, ¶48 (“payment server 114²⁴ may also **store a preferred payment method** for the user and use that method (or present it as a default) if the user does not select another.”). EX1002, ¶278.

Dierks also discloses that the stored information may be a credit card that is accessed after the user authenticates themselves. EX1004, ¶83 (“The shopper ... **authenticates themselves** to a payment processing system ... to **associate subsequent purchase requests with the user’s account**, such as to remove money from the user’s account or **submit a charge to a credit card associated with the user.**”); *see also id.*, ¶86 (“**a charge to a credit card of the buyer**, or to a **bank account** or other account **associated with the buyer.**”). EX1002, ¶279.

Moreover, a POSITA would have understood that the previously stored information is a credit card, a debit card, or bank account information because Dierks discloses that GOOGLE CHECKOUT may be used for payment. EX1004, ¶48. It was well known before the ’785 patent priority date that GOOGLE CHECKOUT stored credit card, debit card or bank account information of users. *See, e.g.*, EX1023, 5:50–65 (“make purchases from Internet merchants 120 without disclosing

²⁴ *See* n. 4.

sensitive payment information (e.g., credit card number). ... GOOGLE CHECKOUT”). EX1002, ¶280.

Accordingly, Dierks discloses that the checkout screen allows a purchase of the item using at least one of a previously stored credit card, a debit card or bank account information. *Id.*, ¶¶281–282.

IX. CONCLUSION

Shopify requests institution of an *inter partes* review of the ’785 patent and cancellation of claims 1–20.

Respectfully Submitted,

Dated: November 22, 2024

/Daniel V. Williams/
Daniel V. Williams, Lead Counsel
Registration No. 45,221

X. APPENDIX OF CLAIM LISTING²⁵

Independent Claim 1

1[preamble]: A method for embedding a buy button for an electronic commerce store, comprising:

1[a]: receiving, by an embeddable buy button builder executing on a first server operated by a buy button provider, a selection of an item from an item inventory and a representation of the item, wherein the item is to be listed for sale on the electronic commerce store;

1[b]: generating, by the first server, the embeddable buy button,

1[c]: wherein the embeddable buy button is configured to display the representation of the item, and a hyperlink; and

1[d]: embedding, by the buy button provider, the embeddable buy button into a web site,

1[e]: wherein the web site is executed by a second server not operated by the buy button provider,

1[f]: wherein the electronic commerce store is displayed within the web site upon selection of the hyperlink,

²⁵ 37 C.F.R. §42.24(a)(1) (“The word count... does not include... appendix of exhibits or claim listing.”)

1[g]: the electronic commerce store configured to facilitate payment processing related to a purchase of the item, and

1[h]: wherein the electronic commerce store is operated by the buy button provider.

Dependent Claim 2

The method of claim 1, wherein the embeddable buy button is configured to be embedded into different web sites.

Dependent Claim 3

The method of claim 1, wherein the electronic commerce store includes a checkout screen that allows a purchase of the item using a transaction processing service operated by the buy button provider.

Dependent Claim 4

The method of claim 1, wherein the electronic commerce store includes a checkout screen that allows a purchase of the item using a transaction processing service operated by a third-party server.

Dependent Claim 5

The method of claim 1, wherein the item is a physical good or service.

Dependent Claim 6

The method of claim 1, wherein the representation is selected from a group consisting of a static image, a video file, and a multimedia file.

Independent Claim 7

7[preamble]: A system for embedding buy button for an electronic commerce store, comprising:

7[a]: a first server operated by a buy button provider;

7[b]: a database communicatively coupled to the first server, the database configured to store an inventory list of items capable of being offered for sale on the electronic commerce store;

7[c]: an embeddable buy button builder executed by the first server, wherein the embeddable buy button builder is configured to receive a selection of an item from the inventory list and a representation of the item,

7[d]: the embeddable buy button builder further configured to generate the embeddable buy button,

7[e]: wherein the embeddable buy button is configured to be embedded into a web site by the buy button provider,

7[f]: wherein the embeddable buy button includes a hyperlink; and

7[g]: a second server configured to execute the web site,

7[h]: wherein the buy button provider controls operation of the embeddable buy button on the web site, and

7[i]: wherein the buy button provider controls operation of the electronic commerce store on the web site upon selection of the hyperlink.

Dependent Claim 8

The system of claim 7, further comprising a transaction processing service operated by the buy button provider, the transaction processing service configured to process a payment initiated via the electronic commerce store.

Dependent Claim 9

The system of claim 7, wherein the buy button provider is configured to execute the electronic commerce store.

Dependent Claim 10

The system of claim 7, wherein the embeddable buy button builder generates hypertext-markup-language code that is configured to be inserted into the web site.

Dependent Claim 11

The system of claim 7, wherein the representation is selected from a group consisting of a static image, a video file, and a multimedia file.

Dependent Claim 12

The system of claim 7, wherein the embeddable buy button is configured to be embedded into different web sites.

Dependent Claim 13

The system of claim 7, wherein the electronic commerce store includes a checkout screen is configured to facilitate the purchase of the item.

Independent Claim 14

14[preamble]: A method for embedding a buy button for an electronic commerce store, comprising:

14[a]: receiving, by embeddable buy button builder executing on a first server operated by a buy button provider, a selection of an item to sell on a web site and a representation of the item,

14[b]: wherein the web site is executed by a second server not operated by the buy button provider;

14[c]: generating, by a first server, the embeddable buy button containing software configured to be executed by the first server on the web site;

14[d]: embedding, by the buy button provider, the embeddable buy button into the web site, wherein the embeddable buy button includes a hyperlink; and

14[e]: executing, by the buy button provider, the software upon selection of the hyperlink, wherein the software is configured to display the electronic commerce store within the web site.

Dependent Claim 15

The method of claim 14, wherein the embeddable buy button is configured to be embedded into different web sites.

Dependent Claim 16

The method of claim 14, wherein the electronic commerce store includes a checkout screen is operated by the buy button provider.

Dependent Claim 17

The method of claim 16, wherein the checkout screen is configured to facilitate the purchase of the item by a transaction processing service operated by the buy button provider.

Dependent Claim 18

The method of claim 14, wherein the representation is selected from a group consisting of a static image, a video file, and a multimedia file.

Dependent Claim 19

The method of claim 14, wherein the item is a physical good or a service.

Dependent Claim 20

The method of claim 14, wherein checkout screen allows a purchase of the item using at least one of a previously stored credit card, a debit card or bank account information.

TABLE OF EXHIBITS

Exhibit	Description
1001	U.S. Patent No. 10,846,785
1002	Declaration of Todd Mowry
1003	U.S. Patent No. 6,868,403 (“Wiser”)
1004	U.S. Patent Application Pub. No. 2008/0183593 (“Dierks”)
1005	U.S. Patent Application Pub. No. 2010/0114739 (“Johnston”)
1006	U.S. Patent Application Pub. No. 2010/0250382 (“Babaria”)
1007	<i>Reserved</i>
1008	U.S. Patent Application Pub. No. 2008/0255962 (“Chang”)
1009	’785 Patent File History
1010	U.S. Provisional Application No. 61/093,917 (“Johnston-Provisional”)
1011	U.S. Patent No. 9,324,098 (“Agrawal”)
1012	U.S. Patent No. 7,565,308 (“Bollay”)
1013	U.S. Patent Application Pub. No. 2008/0228595 (“Hill”)
1014	<i>Reserved</i>
1015	<i>Reserved</i>
1016	<i>Reserved</i>
1017	<i>Reserved</i>

Petition for *Inter Partes* Review
of U.S. Patent No. 10,846,785

Exhibit	Description
1018	<i>Reserved</i>
1019	U.S. Patent Application Pub. No. US 2007/0045405 (“Rothschild”)
1020	U.S. Patent No. 8,046,288 (“LeRoux”)
1021	MPEP §2117(I)
1022	U.S. Patent No. 9,203,971 (“Narain”)
1023	U.S. Patent No. 8,069,088 (“Foulser”)
1024	D. Kossmann et al., “An evaluation of alternative architectures for transaction processing in the cloud,” <i>Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data</i> , June 2010
1025	Order Granting Motion to Dismiss dated August 1, 2024
1026	Plaintiff DKR Consulting, LLC’s Notice of Appeal to the United States Court of Appeals for the Federal Circuit (dated August 26, 2024)
1027	<i>Reserved</i>
1028	Request for <i>Ex Parte</i> Reexamination of ’678 Patent, Control No. 90/019,476
1029	Non-Final Office Action in <i>Ex Parte</i> Reexamination of ’678 Patent
1030	United States Court of Appeals for the Federal Circuit, Notice of Docketing (dated August 30, 2024)
1031	Plaintiff-Appellant DKR Consulting, LLC’s Opening Brief (dated October 29, 2024)
1032	ACM Digital Library for Kossmann

Petition for *Inter Partes Review*
of U.S. Patent No. 10,846,785

Exhibit	Description
1033	Decision on Petition in <i>Ex Parte</i> Reexamination of '678 Patent

CERTIFICATE UNDER 37 CFR § 42.24(d)

Under the provisions of 37 CFR § 42.24(d), the undersigned hereby certifies that the word count for the foregoing Petition for *Inter Partes Review* totals 13,916, which is less than the 14,000 words allowed under 37 CFR § 42.24(a)(1)(i).

Respectfully submitted,

Dated: November 22, 2024

/Daniel V. Williams/
Daniel V. Williams
Reg. No. 45,221

CERTIFICATE OF SERVICE

I hereby certify that on November 22, 2024, I caused a true and correct copy
of the foregoing materials:

- Petition for *Inter Partes* Review of U.S. Patent No. 10,846,785 under 35 U.S.C. § 312 and 37 C.F.R. § 42.104
- Exhibit List
- Exhibits for Petition for *Inter Partes* Review of U.S. Patent No. 10,846,785 (Exhibits 1001–1006, 1008–1013, 1019–1026, 1028–1033)
- Power of Attorney
- Fee Authorization
- Word Count Certification Under 37 CFR § 42.24(d)

to be served via Federal Express on the following correspondent of record as listed
on USPTO Patent Center:

Maulin Shah
Envision IP LLC
2020 Progress Court
Suite 100-158
Raleigh, NC 27608

with a courtesy copy via electronic mail on the following counsel and litigation
counsel of record:

Ronald M. Daignault (rdaignault@daignaultiyer.com)
Chandran B. Iyer (cbiyer@daignaultiyer.com)
Jason S. Charkow (jcharkow@daignaultiyer.com)
Austin Ciuffo (aciuffo@daignaultiyer.com)

DATED: November 22, 2024

/Daniel V. Williams/
Daniel V. Williams
Reg. No. 45,221