Views:

1. Product List:

Guest Main Menu (A) or Product List Based on Specific Requirements

* 1. Guest Main Menu (A)
  2. Filter/SortBy Product Results (C)
  3. Specific Category Page (G)
  4. Search Product Results (I)
  5. Admin Removal of Item (#2.1)
  6. Marketplace – list of all your products and can add to marketplace (#8)
  7. Versions:
     1. Guest Access: Add to Cart
     2. User Access: Add to Cart/ Add to Wishlist
     3. Admin Access: Remove/ Edit (#1)
     4. In Account Page - Admin Marketplace: Add to marketplace (#8 – got to #9)
     5. In Account Page - Admin Specific Marketplaces: products currently added to marketplace – remove/ edit option (#11)

1. Product View (C2D)
2. Checkout Page (E)
3. Sign Up (K)
4. Login In (L)
5. Account Page (N) – Account Information (N)(#6)
6. Wishlist (O)
7. Change Password (P) (#6)
8. Edit Product (#3)
   1. Also used to add specific requirements for marketplace (#11)
9. Add Product (#7)
10. View Stats for specific marketplaces (#13)

Pop Up View:

1. Filter/SortBy (B)
2. Shopping Cart (D/ D.1/ D.2)
3. Navigation Bar (F)
4. Search Engine (H)
5. Login/ SignUp Options (J)
6. Add to Wishlist (M)
7. Approve Removal (#2)
8. Add Feature (#4)
9. Remove Feature (#5)
10. Marketplace – add to marketplace (#9)
11. Marketplace requirements notification (#10) (#11)
12. Marketplace – product added successfully (#12)

**Guest POV:**

A – L

**User POV:**

M – P

**Admin POV:**

#1 – #13

|  | **Guest Access** |  |
| --- | --- | --- |
| A | A picture containing graphical user interface  Description automatically generated(A) | **View: Main Menu (A) – main page**  Options:   * Navigation Bar (will be shown with transition between guest to user access),   + Will use Javascript to handle button presses, pressing the leftmost button displays different categories. This will require GET requests from the server to retrieve the data. * Search Option (will be shown with transition between guest to user access),   + It will allow a general search over product names, it may be best to organize data to achieve proficient searches.   + The search option must send a post request containing the data entry to work properly * Account Option (will be shown with transition between guest to user access),   + Javascript backend must be capable of knowing if the user is authenticated or not, if so then this option’s frontend output changes   + Javascript should determine the authentication state of the client and then show them a view based on their profile specifications   + This will theoretically only need a get request, but we may need to also post our client’s ip address and local port. * Shopping Cart Option (Go to E),   + Opens modal using bootstrap * Sort/Filter Option (Go to B),   + Opens modal using bootstrap * AddToCart Option (Go to D, D.1, D.2),   + Opens modal using bootstrap * ViewProductPage Option (Go to c2D)   + Renders product page using javascript and the DOM. |
| B | Text, letter  Description automatically generated (B) | **Pop Up View: Filter/SortBy (B)**  Options:   * Reset Option (Go to A),   + Resets the variables to their default positions * Filter/Sort Option (Go to C)   + Post above arguments to be processed by the server. It’ll then display the results shown in C |
| C | Graphical user interface, Excel  Description automatically generated (C) | **View: Filter/SortBy Result Page (C)**  Options:   * Same options as Page (A), * Click on image to view product page (Go to C2D)   + Displays focused product page that will require a get request for the products data |
| D | Graphical user interface, text, application, chat or text message  Description automatically generated (D) | **Pop Up View: Shopping Cart (D)**  Options:   * Exit (Go to previous view – this example C), * Check Out (Go to E) |
| C2D | Graphical user interface  Description automatically generated with medium confidence (C2D) | **View: Product View – image was clicked on (C2D)**  Options:   * Add to Cart (automatically adds to cart)   + Post to the server to update clients cart * ShoppingCart (Go to D.1) |
| D.1 | Graphical user interface, text, application, chat or text message, website  Description automatically generated (D.1) | **Pop Up View: Shopping Cart (D.1)**  Options:   * Exit, * Check Out Check Out (Go to E), * Increase Quantity (Go to D.2), * Decrease Quantity   Will require post requests in order to update the state for persistent data. Checking out is better defined at E. |
| D.2 | Graphical user interface, text, application, chat or text message, website  Description automatically generated (D..2) | **Pop Up View: Shopping Cart (D.2)**  Options:   * Exit, * Check Out (Go to E), * Increase Quantity, * Decrease Quantity |
| E | A picture containing graphical user interface  Description automatically generated (E) | **View: Checkout Page (E)**  Options:  Pay (Sends you to PayPal),  This will invoke Paypal’s API and process the checkout, we will need to provide product variables here. This step will require a little additional research in order to be implemented. |
|  | **Shows Guest Access And Then Transitions to Regular User Access** |  |
|  | **Guest Access** |  |
| A again | A picture containing text, screenshot, clock  Description automatically generated(A) | **View: Main Menu (A) – main page**  Options:   * Navigation Bar (Go to F), * Search Option (Go to H), * Account Option (Go to J), * Shopping Cart Option (Go to E), * Sort/Filter Option (Go to B), * AddToCart Option (Go to D, D.1, D.2), * ViewProductPage Option (Go to D)   These features were defined in the preceding A section |
| F | A screenshot of a computer  Description automatically generated with low confidence (F) | **Pop Up View: Navigation Bar (F)**  Options:   * All Products (Go to A), * List of Different Product Categories (Go to G), * NavBar (To hide nav bar)   Selecting an option will invoke a get request to the server in order to retrieve the products in their respective categories. The rest is handled in bootstrap. |
| G | Graphical user interface  Description automatically generated (G) | **View: Specific Category Page (G)**  Options:   * Same options as Page (A), * Search Option (Go to H) |
| H | A picture containing graphical user interface  Description automatically generated (H) | **Pop Up View: Search Engine (H)**  Options:   * Search Button (Go to I) , * Exit (By clicking background)   Technicalities:   * Javascript backend must be capable of knowing if the user is authenticated or not, if so then this open is made visible * It will allow a general search over product names, it may be best to organize data to achieve proficient searches. * The search option must send a post request containing the data entry to work properly |
| I | Graphical user interface  Description automatically generated (I) | **View: Search Result Page (I)**  Options:   * Same options as Page(A), * Account Option (Go to J) |
| J | Graphical user interface, application, chat or text message  Description automatically generated (J) | **Pop Up View: LogIn/Login Options (J)**  Options:   * Sign Up (K), * Login (L), Exit   Both button processes will trigger the dom to re render separate html frontends. |
| K | Table  Description automatically generated with medium confidence (K) | **View: Sign Up View (K)**  Option:   * Sign Up   This page is incredibly important, it will need to send post requests to the server to find out if the data we are using is not only correctly formatted, but also if it currently exists in memory. We’re going to have to use the DOM to communicate that information to the client’s frontend. |
| L | Chart  Description automatically generated  (L) | **View: Login View (L)**  Option:   * Login   Send a post request to the server in order to validate the login and let the server know we have an active session we need to keep track of. In order to maintain the active session, we need to periodically make sure the client’s still there, and we also need to reap idle connections if they aren’t. It’s okay to have multiple windows, so we must identify each session on its ip:src port pair. |
|  | **Regular User Access** |  |
| User’s  Version  Of  I,A,C | Graphical user interface  Description automatically generated | **View: User’s View of Products**  Options:   * Same Options as Page (A), * Add to Wishlist (+) (Go to M),   + Opens modal using bootstrap displaying wish lists to add the product to * Account Page (Go to N) |
| M | Graphical user interface, text, application, chat or text message  Description automatically generated (M) | **Pop Up View: Add to Wishlist (M)**  Options:   * Create New List (M.1) , * Add to Existing List (M.2), * Exit   This will post new data to the server under the user who activated it. It will create a new list that they can then add products to. |
| M.1 | Graphical user interface, text, application, chat or text message  Description automatically generated (M.1) | **Pop Up View: Add to Wishlist (M.1)**  Options:   * Enter Wishlist Name and click Create (i.e. My Favorite) – then close pop up |
| M.2 | Graphical user interface, text, application, chat or text message  Description automatically generated (M.2) | **Pop Up View: Add to Wishlist (M.2)**  Options:   * Existing Wishlists (adds item to preexisting list), * Create New List |
| N | Graphical user interface, application  Description automatically generated(N) | **View: Account Page (N) – automatically on account info view (N)**  Options:   * Account Info - Can change account info + save (updates N view), * Wishlist (Go to O), * Change Password (Go to P), * Log Out (returns to page A)   This will require a post request because we need to identify the client before we can actually retrieve the data we need to display here. We will need to post the required arguments and get a response showing account information. |
| O | Graphical user interface  Description automatically generated with medium confidence (O) | **View: Clicked Wishlist in NavBar 🡪 Clicked Specific Wishlist (O)**  Options:   * Remove (Update O view), * Product Page (Click on image – Go to C2D)   This will require a post request because we need to identify the client before we can actually retrieve the data we need to display here. We will need to post the required arguments and get a response showing wishlist information. |
| P | Graphical user interface, application  Description automatically generated (P) | **View: Change Password (P)**  Options:   * Same as Page N, * Save changed password   A post request will need to send the entered data over to be verified by the server, which should then let the client know what to render using the DOM. |
|  | **Admin Access** |  |
| 1 | A picture containing text, clock  Description automatically generated #1 | **View: Admin Main Menu #1**  Options:   * Navigation Bar (Go to F), * Search Option (Go to H), * Remove (Go to #2), * Edit (Go to #3), * Account Option (Go to #6), * Marketplace implementation #8   Aside from the general features already described, we can now remove products from the view if we want. This should update the storefront’s presentation itself in persistent memory. It’ll require a post request indicating the product we want gone. |
| 2 | Graphical user interface, text, application, chat or text message  Description automatically generated #2 | **Pop Up View: Approve removal #2**  Options:   * Remove (Go to #2.1), * Exit (Close Pop Up) |
| 2.1 | Graphical user interface, Excel  Description automatically generated with medium confidence  #2.1 Result from removal |  |
| 3 | Graphical user interface  Description automatically generated with medium confidence #3 | **View: Edit Product #3**  Options:   * Edit text input, * Add Feature (Go to #4), * Remove Feature (Go to #5), * Save all changes: text input changes, added features, and/or removed features (Updates #3 view with new changes)   In order to display this screen we will need to do a get request that pulls all the data for us to present. Products will need a mutable data structure that maintains its available features. A dictionary seems suitable, we will post any added/removed/modified features to the server. |
| 4 | Graphical user interface, application  Description automatically generated #4 | **Pop Up View: Add Feature #4**  Options:   * Exit (close pop up), * Fill in feature information, * Add another feature (Go to #4.1), * Save feature return to previous page by closing pop up   This is a representation of the feature we plan to add using a post request defined above. |
| 4.1 | Graphical user interface, application, website  Description automatically generated  #4.1 adding >1 feature | We will maintain a list of features client sided before forwarding that to the server with a post request. |
| 5 | Graphical user interface, text, application  Description automatically generated #5 | **Pop Up View: Remove Feature #5**  Options:   * List of Features click (X) to remove feature, * Exit (close pop up)   Send post requests that remove product features from persistent data. |
| 6 | Graphical user interface, application, table  Description automatically generated #6 | **View: Account Option #6**  Options:   * Account Info (Go to N slightly adjusted will not have cart option at top), * Change Password (Go to P slightly adjusted will not have cart option at top), * Add Product (Go to #7), |
| 7 | Text, letter  Description automatically generated#7  Text, letter  Description automatically generated #7 | **View: Add Product #7**  Options:   * Fill in Product Name, * Price, * Quantity, * Category, * Number of features based on quantity and quality, * Provide name of feature and value, * Add Product (adds product and reloads #7), * Log Out (Go to Page A)   This screen will allow us to create a base product to then add it to a category on the store. It will use a post request providing all the arguments you see here and then process the data into persistent memory. |
| 8 | **Marketplace Feature #8**  Shape, polygon  Description automatically generated#8 | **View: Marketplace – list of all your products #8**  Options:   * Add – add product to a market place (Go to #9), * Click on specific market place (Go to #11/ #11.1), * click on specific market place’s statistics (Go to #13) |
| 9 | Graphical user interface, application  Description automatically generated #9 | **Pop Up View: Marketplaces #9**  Options:   * Exit, * Add (+) – add product to marketplace (if there is no (+) then it means it already has been added Go to #10)   When adding a product into a marketplace we will need to invoke their API in order to actually send it over. This is probably going to be done through a post request, if that’s a possibility. This may actually require a key input field from the admin. |
| 10 | Graphical user interface, text, application  Description automatically generated | **Pop Up View: Marketplace requirements notification #10**  Options:   * Exit   If the product’s import into a marketplace doesn’t meet requirements indicated to us by the marketplace, we will display this prompt in the form of a modal communicating that response. |
| 11 | Shape, polygon  Description automatically generated #11 | **View: Specific products added to marketplace #11 – no notifications**  Options:   * remove product from marketplace (Go to #2), * edit product (Go to #3)   In addition to updating our persistent data, we need to also remove the product using the marketplace’s API. This will be done through a minimum of two post requests to both servers indicating that we are removing a product. |
| 11.1 | Graphical user interface, application  Description automatically generated  Shape, polygon  Description automatically generated#11.1 | **View: Specific products added to market place #11.1 – notification pops up – requirements not met**  Options:   * Same as #11, * After #3/4 (make and save changes) Go to #12 |
| 12 | Graphical user interface, text, application  Description automatically generated #12 | **Pop Up View: Success – Added to Marketplace #12**  Options:   * Exit |
| 13 | Text  Description automatically generated #13Text  Description automatically generated #13 | **View: Statistics #13**  Options: None  This is a page that will pull data from a marketplace about a posted product and display it to the admin. It will be done using their available API, likely through post requests. |