Project Title: EDGE Online
Team: Team 3.5
Subject: Esports hardware Online Store

Databases:

Key:

Integer
“String”
$ Double
~Boolean~
Optional variables*
Primary Key

Provider

- **#id**
- “prov_name” (name of provider)
- “prov_addr”* (address of provider)
- “prov_logo”* (image file of the logo)
- “prov_link”* (linking to provider’s site)

Item

- **LONG id**
- “item_name” (Name)
- “item_cat”* (Category)
- $item_price
- “item_img” (Image of item)
- $item_rating* (numerical rating)
- #prov_id (Foreign key using Provider: id)
- #item_stock (This is hidden from the user)

Order

· **LONG id**

· LONG item_id (Foreign key using Item: id)

· #amount

· #orderlist_id (corresponds to OrderLine: id, is not a foreign key)

. ~complete~ (Tells if this order is completed, initially false)

OrderLine (This holds most payment operations to reduce data usage since there are fewer of these than orders. We can get Orders based on orderlist_id by using this method)

· **#id**

· “user_name” (name of the user, assigned at order submission)

· “user_addr” (user address, assigned at order submission)

· #order_stage (code representing the stage of the OrderLine, initially 10, will detail later)

. #user_card_num (number of the card, assigned at order submission)

. #user_card_sec (security code of the card, assigned at order submission)

. DATE user_card_expdate (expiration date of the card, assigned at order submission, date format <mm/yy>)

. #auth_code (authorization code, assigned at order submission)

Order_stage codes

10: before submission
11: submitted, in process of shipping
12: order shipped
20: order expired as one or more items are unavailable (before and at submission)
21: order expired due to termination by administrator*

Object Assignment notes:

Provider is assigned by the admin or pre-assigned by the program.

Item is assigned by the admin or pre-assigned by the program.

Order is assigned after an item is added to the cart. This must be assigned to an OrderLine, creating a new OrderLine when necessary.

OrderLine is assigned after the first item is added to the cart, cookies will ensure that any item added to the cart by that user will have the same orderlist_id, including the first item that will be added.

DTOs:

ProviderDTO: DTO Corresponding to Provider Data.
ItemDTO: DTO Corresponding to Item Data.
OrderDTO: DTO Corresponding to Order Data.
OrderLineDTO: DTO Corresponding to OrderLine Data

CartDTO: DTO Keeping track of the number of items and the total, along with keeping track of the orders that correspond to the OrderLine this DTO is responsible for.
CheckoutDTO: DTO that keeps track of the checkout information like the user's name, address, and card information.
OrderConfirmationDTO: DTO that has the auth_code, the id of the OrderLine, and the final total of the order.
ItemSelectionDTO: DTO similar to the MovieSelectionDTO.

Services:

ItemService: Uses Item Repository, Provider Repository, and Item Mapper

List<ItemDTO> findAllFilterable (String searchTerm, String itemCat, String provName, Boolean inStockOnly)

- This is a high level search operation for the sites search functionality.
- By default, the acts as a findAll for the items in stock with the parameters (“”,””,””,True)
- SearchTerm will find all items that contains the given string
- itemCat finds items in that category
- provName finds items from that provider
- inStockOnly determines if items that are out of stock are included or not.

ItemDTO findOne (Long id)

ItemDTO save (ItemDTO item)

ItemDTO update (ItemDTO item, Long id)
Void delete (Long id)
List<ProviderDTO> findAllProviders ()

OrderService: Uses Order Repository, OrderLine Repository, and Item Repository

* OrderService is used for both Order and orderLine functionalities as this is where the functions that define the relationship between these two databases is defined.

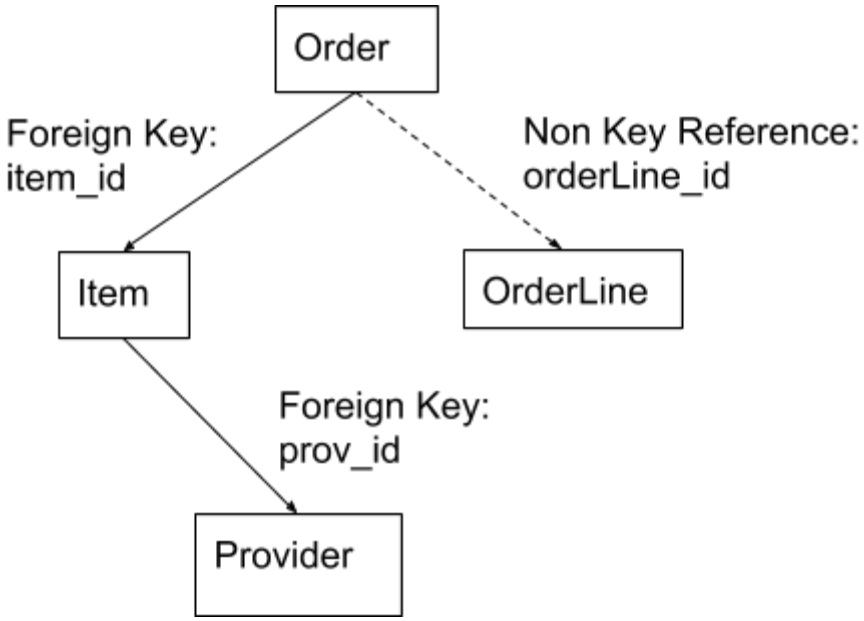
List<OrderDTO> fetchAllOrders ()
List<OrderDTO> fetchOrdersInLine (Integer orderLineId)
- This defines the relationship between between Order and OrderLine. The OrderLineId corresponds to the id of the Order List, but not actually a foreign key as the Orders never would need any data about the OrderLine besides the id.

OrderConfirmationDTO submitOrders (OrderLineDTO orderLine, CartDTO cart)
- These operations will only be performed on orderlists with order_stage code of 10, returning an exception otherwise.
- Whenever this is called, fetchOrdersInLine will be performed with the OrderLine's id. Then the Order is checked to see if the item can be found in the Item Database. If there exists an order which has an item_id that cannot be found in the Item Database. Then the order_stage is set to 20 and the auth_code is null. Otherwise, the order_stage is set to 11.

OrderLineDTO completeOrders (OrderLineDTO orderLine)
- This will complete orders if there is enough in stock. Otherwise, the order remains incomplete. The OrderLine's order_stage code will change to 12 if all orders are complete.
- The operations will only be performed when the OrderLine used has the order_stage code of 11, returning an exception otherwise

Void terminateOrders (OrderLineDTO orderLine)
- This will change the order_stage to 21 and delete all orders sharing the same orderlist_id. This does not delete the OrderLine object.
- This operation will only be performed with the ObjectLists having order_stage code of 10, creating an exception for those that do not have the proper code.

Diagram of Database relationships:



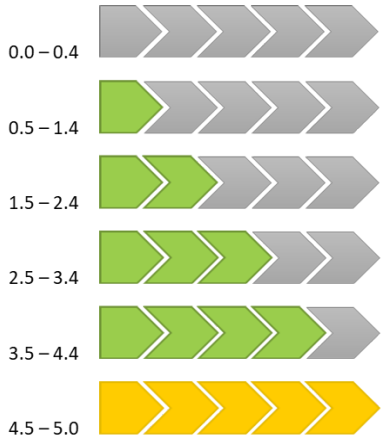
Data of OrderLine is independent of Order. OrderService provides the means of the relationship between these two databases.

Important Aspects:

- When the admin changes the price of the item, the change will affect any order_stage 10 orderlines.
- The orderline id should never be changed. A terminated orderlist is removed if that session creates a new orderlist to use.

Front End Details:

Ratings:



The six rating bar images can be found in the image folder of the project.

Backend Functions of the program:

- Admin can add, update, delete, and get items and providers. The admin can get order lists, along with being able to terminate those that have not been submitted yet. The admin can get the orders corresponding to the orderlist id.
- Item's inventory updates after orders are completed
- The orders are shipped when all orders on the orderlist are complete, like fitting multiple items in one package or something similar.
- Orders can be fetched based on the orderlist id.
- Provider data can be used by the Items
- OrderLines have order_stage code 10 before submission, 11 while being filled after submission, and 12 when it is shipped.

- During submission, all orders are checked for validity, and the order_stage code will be 20 if any order is invalid.
- The CartDTO tracks the number of items, the total, the corresponding orderlist id, and a list of the order ids of orders with the specific orderlist id.
- CheckoutDTO Holds the user's name, address, and card information.
- ItemSelection holds information of selected items.
- Items can be filtered by whether the name contains the specified search phrase or not.
- Orders reference the id of the orderline instead of being added to the orderline.

Web app functionality aspects:

- Hovering over the provider's name shows a tooltip displaying the logo, address, and name. Clicking on that name links to the provider's website.
- Each item displays the name, image, categories, provider, price, an input box for the quantity, and an add button.
- If the item is out of stock, the add button is greyed out and has 'out of stock' instead of 'add'. Also, if the quantity is larger than the inventory, they add button is greyed out, and has 'max ###' instead of 'add'. The button is greyed out if the quantity is less than 1.
- Items can be removed from the shopping cart, hence deleting that order.
- Quantity can also be changed from the shopping cart page.
- The admin pages are easy to use
- The admin cannot terminate submitted orderlists.
- The catalog page initially will display all items that are in stock
- The items can be filtered by the catagory, the provider, and if in stock. All of these use drop down options instead of typing these in.
- The items are searchable in that they can be filtered by those containing the keyword in the name.
- The navbar has a button linking to the shopping cart, which also displays the number of items in the cart if there is at least one item in the cart.
- The checkout button on the cart page cannot be selected if there are no items in the cart or if there is an item with the quantity of the order exceeding the item's stock.
- The checkout page lists the orders at the top of the page, followed by the total cost of these orders.
- In the checkout, the user's name must have at least three characters
- In the checkout, the user's address formats like how addresses are usually formatted.
- In the checkout, the card number is split into four integer input boxes, with the user_card_num value being in the format of "XXXX XXXX XXXX XXXX".
- In the checkout, the card security code must be a three digit integer
- In the checkout, the card expiration date is split into two double digit integer input boxes
    - The first box, the month, must have a value between 01 - 12.
    - The second box, the year, must have a value between 19 - 39
    - The date format is mm/yy

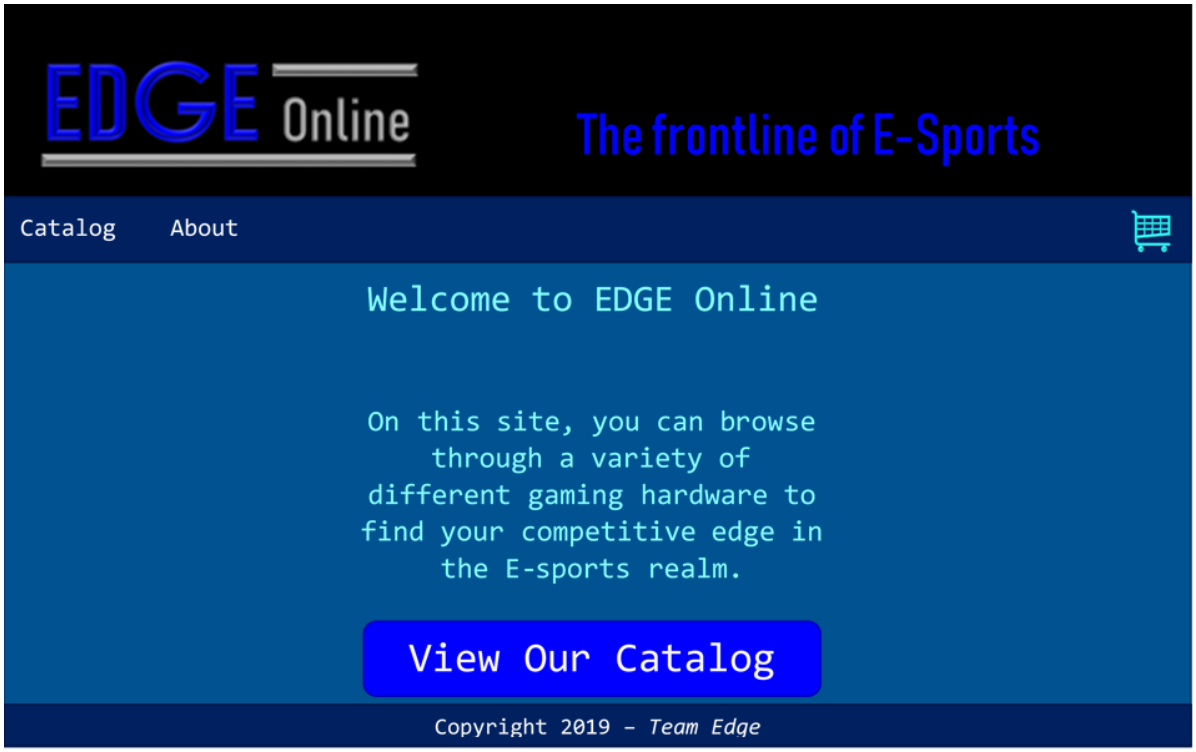
    - The valid date range is from 11/19 to 11/39, assuming that there is no card that can have more than a 20 year validity span
- The submit button is greyed out until the requirements are met.
- The submit button will link to the submit page if the results of the order submission has an authorization code. Otherwise, the button links to the order expiration page.
- Clicking on the cart button, the checkout button, or the submit button when the corresponding orderlist has an order_stage code of 21 will link to the order expiration page.
- The order expiration page has a different message based on the order_stage code of the orderlist.
- The receipt page includes the authorization code, a list of the orders, the total cost, the user's name and address, and a button to return to the catalog page.

Front end protocols

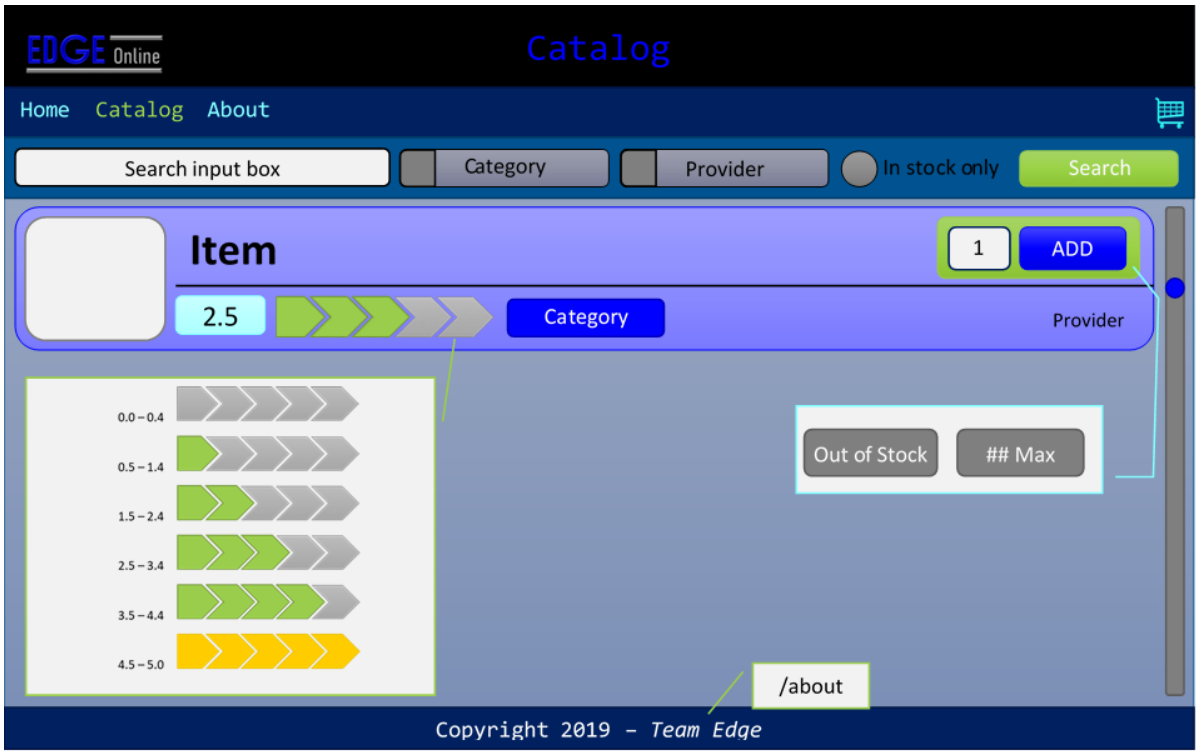
- A header box, a footer, and a nav bar is on every page
- The header box contains the site's logo, the title, and the slogan of this site.
- The navbar has the following: homepage link, catalog link, and a button linking to the cart.
- The footer has the following: copyright statement with the company name linking to the about page.
- The navbar and footer are the same color.
- The catalog page includes search operations at the top of that page
- The admin pages have a special navbar and footer that both have links to each of the admin pages: catalog/admin, orderlist/admin, provider/admin, and about/admin, along with a link to the home page to exit this mode.
- The admin pages have a different color scheme than the regular pages to help differentiate the two.
- The home page helps to introduce the users to the site, giving a brief introduction followed by a link to the catalog.
- The bottom of the home page has a section that references all of the providers that the site uses.

Wireframe Collections

- Home Page:

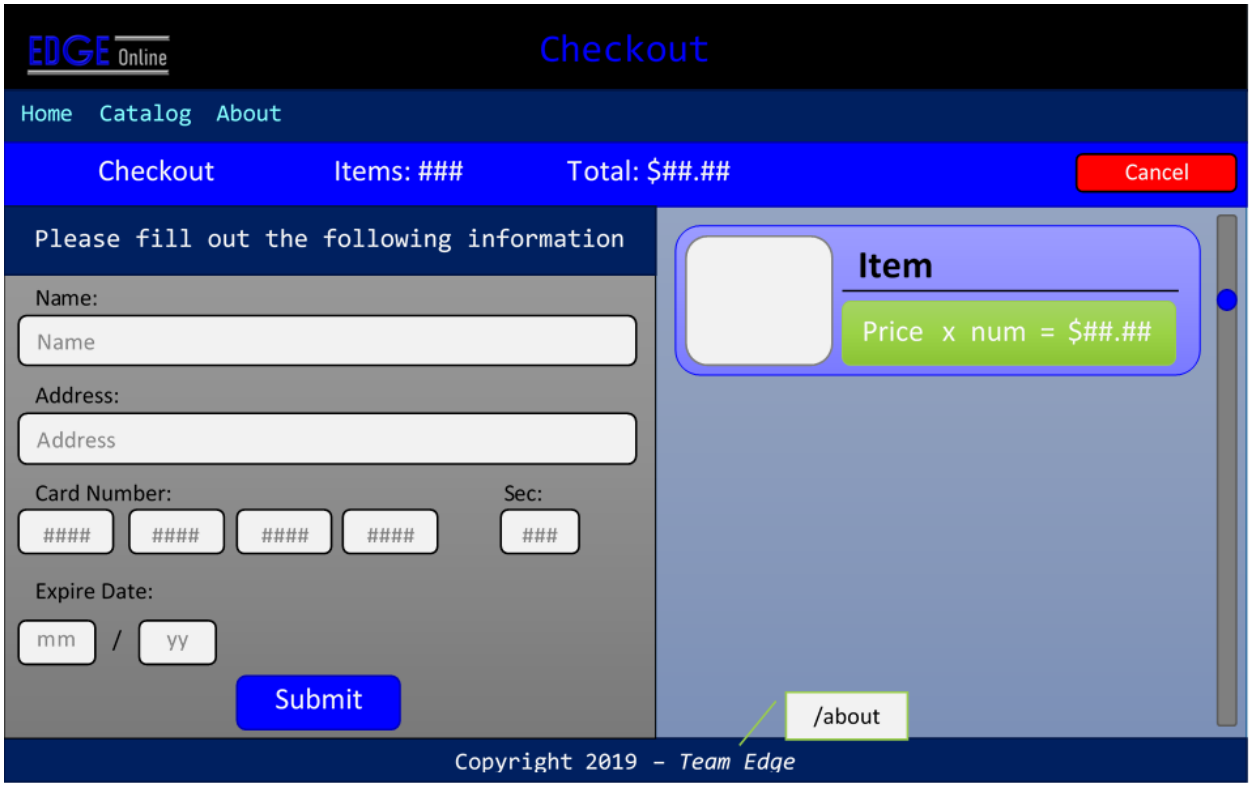


- Catalog Page:

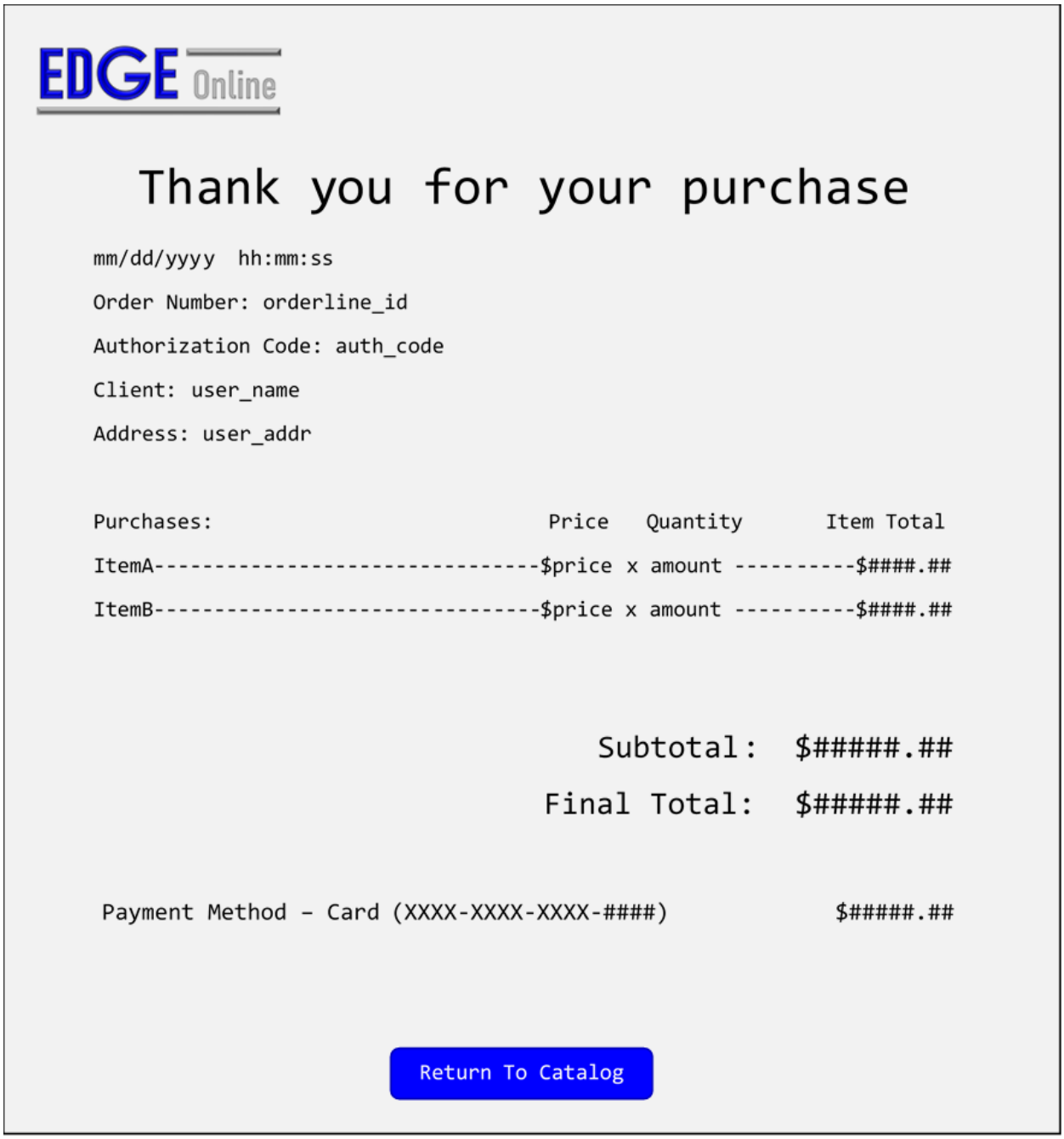


- Cart Page:

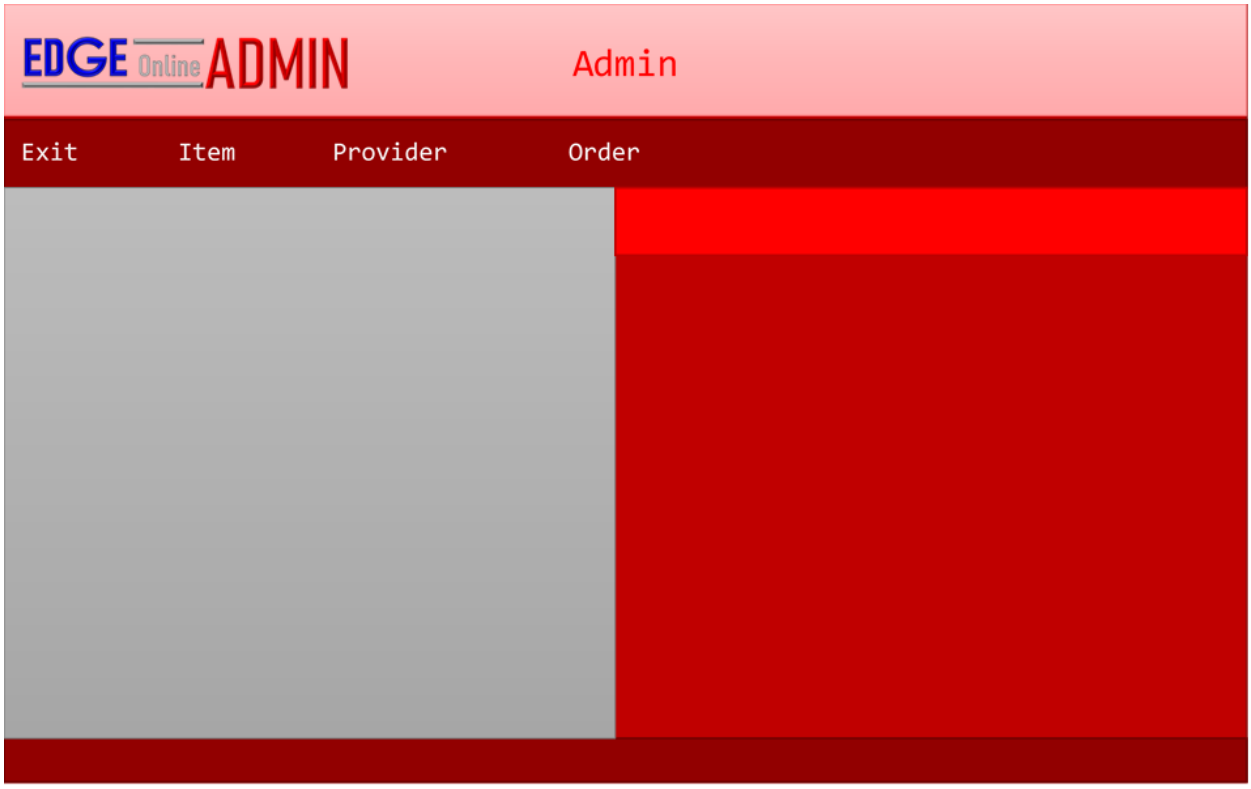
- Checkout Page:



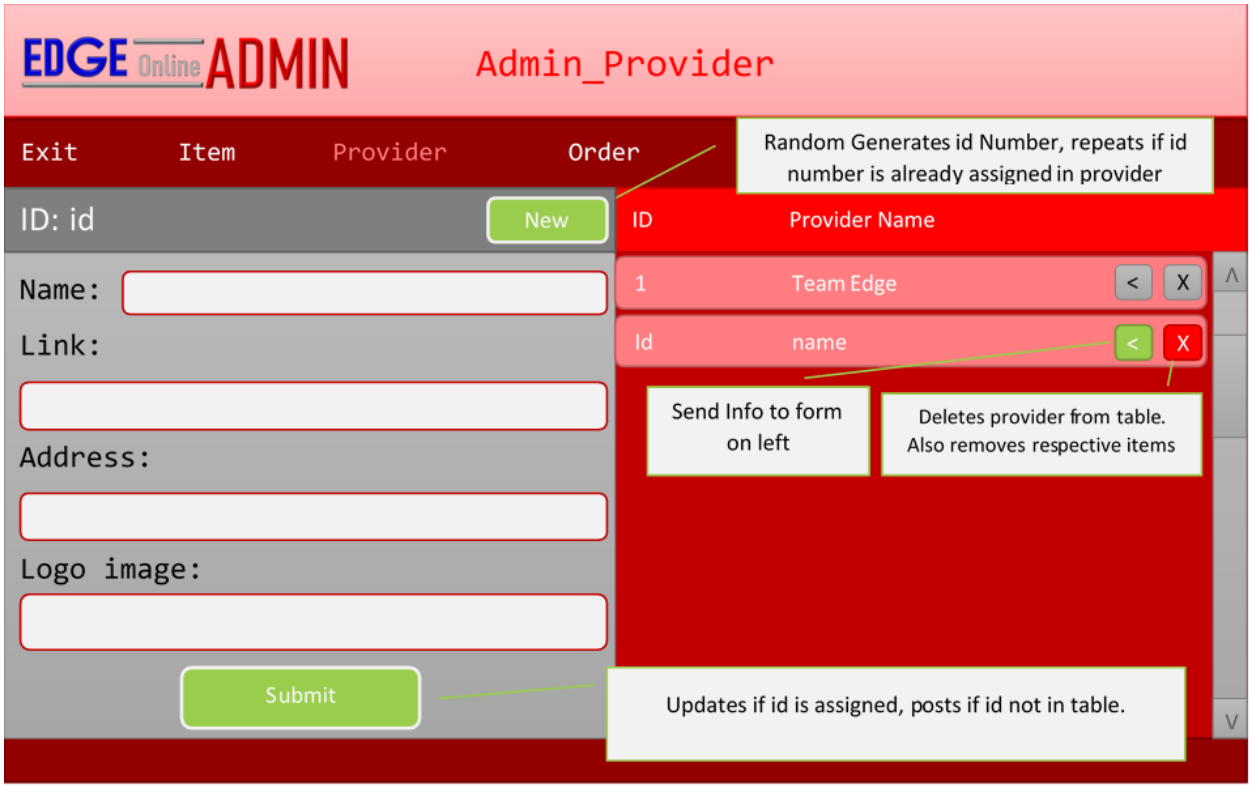
- Receipt / confirmation



- Admin Landing Page



- Admin Provider Page



- Admin Item Page