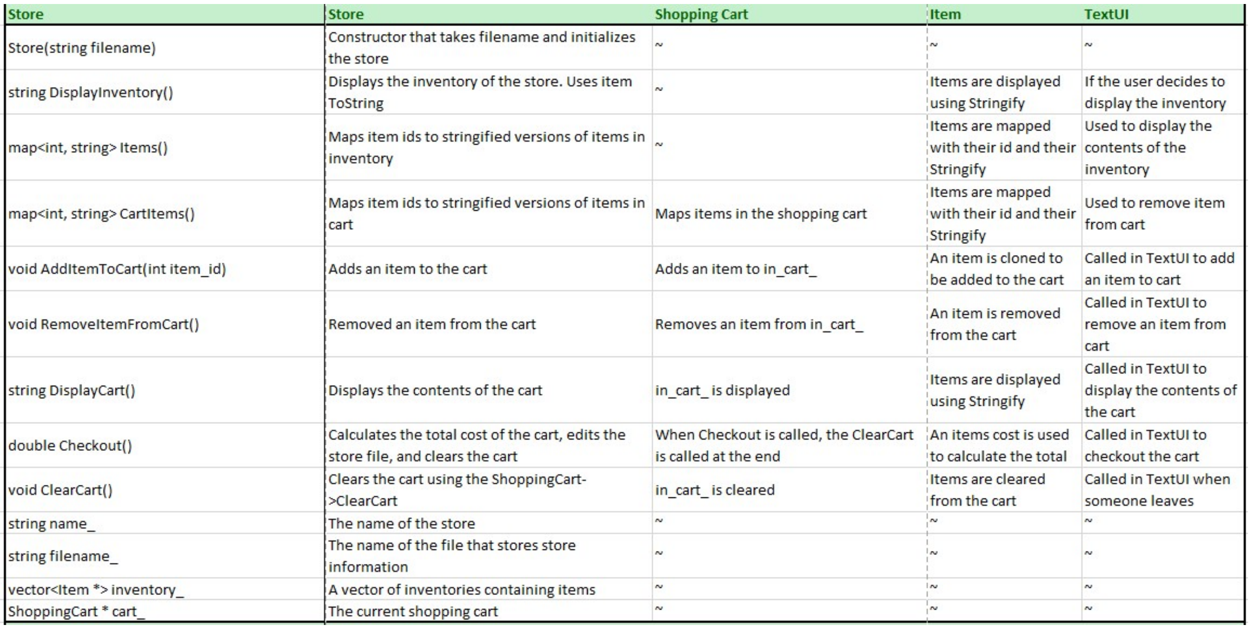
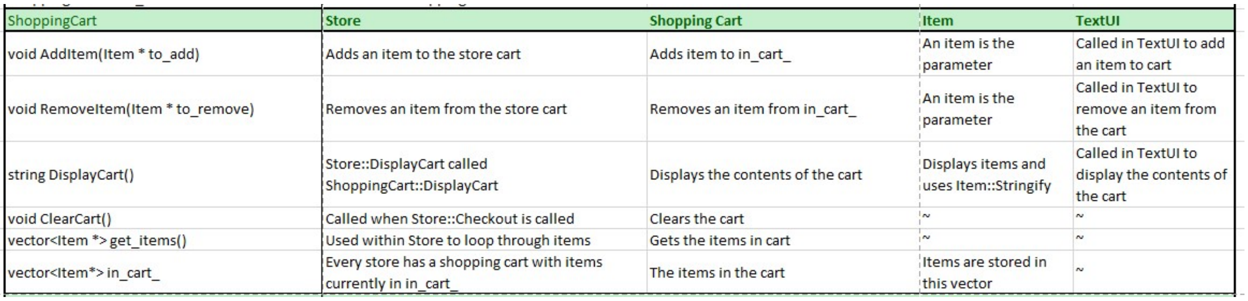
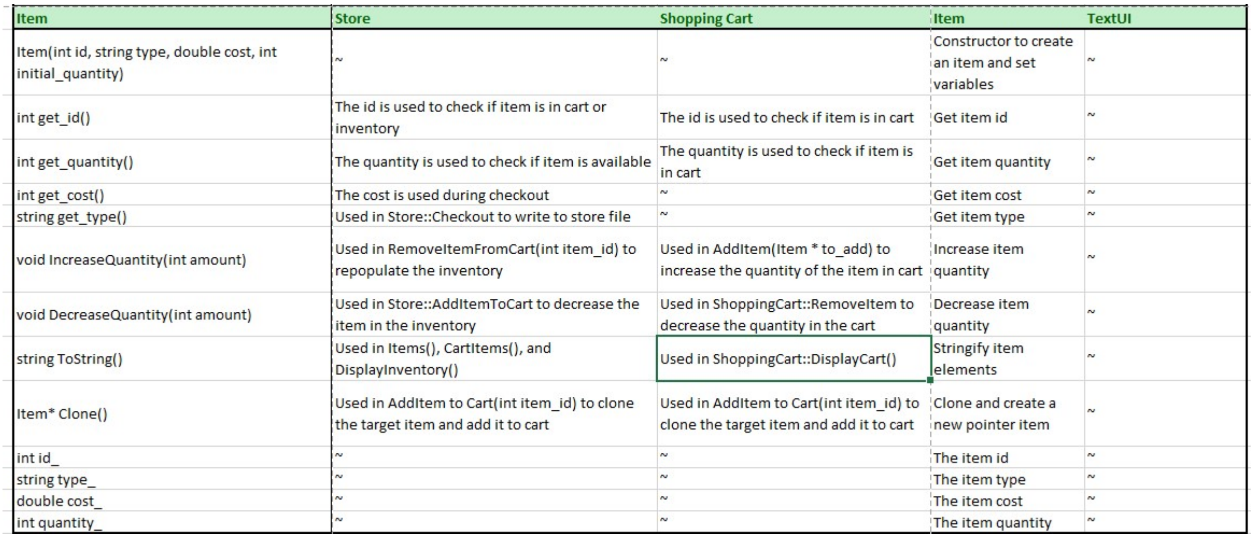


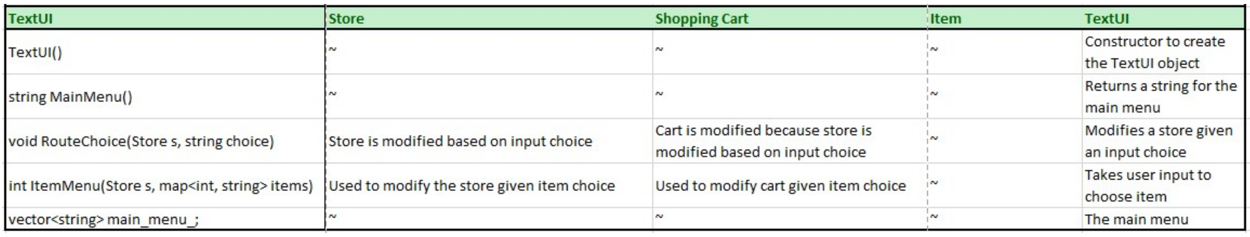
Store	Store	Shopping Cart	Item	TextUI
Store(string filename)	Constructor that takes filename and initializes the store	~	~	~
string DisplayInventory()	Displays the inventory of the store. Uses item ToString	~	Items are displayed using Stringify	If the user decides to display the inventory
map<int, string> Items()	Maps item ids to stringified versions of items in inventory	~	Items are mapped with their id and their Stringify	Used to display the contents of the inventory
map<int, string> CartItems()	Maps item ids to stringified versions of items in cart	Maps items in the shopping cart	Items are mapped with their id and their Stringify	Used to remove item from cart
void AddItemToCart(int item_id)	Adds an item to the cart	Adds an item to in_cart_	An item is cloned to be added to the cart	Called in TextUI to add an item to cart
void RemoveItemFromCart()	Removed an item from the cart	Removes an item from in_cart_	An item is removed from the cart	Called in TextUI to remove an item from cart
string DisplayCart()	Displays the contents of the cart	in_cart_ is displayed	Items are displayed using Stringify	Called in TextUI to display the contents of the cart
double Checkout()	Calculates the total cost of the cart, edits the store file, and clears the cart	When Checkout is called, the ClearCart is called at the end	An items cost is used to calculate the total	Called in TextUI to checkout the cart
void ClearCart()	Clears the cart using the ShoppingCart->ClearCart	in_cart_ is cleared	Items are cleared from the cart	Called in TextUI when someone leaves
string name_	The name of the store	~	~	~
string filename_	The name of the file that stores store information	~	~	~
vector<Item *> inventory_	A vector of inventories containing items	~	~	~
ShoppingCart * cart_	The current shopping cart	~	~	~



ShoppingCart	Store	Shopping Cart	Item	TextUI
void AddItem(Item * to_add)	Adds an item to the store cart	Adds item to in_cart_	An item is the parameter	Called in TextUI to add an item to cart
void RemoveItem(Item * to_remove)	Removes an item from the store cart	Removes an item from in_cart_	An item is the parameter	Called in TextUI to remove an item from the cart
string DisplayCart()	Store::DisplayCart called ShoppingCart::DisplayCart	Displays the contents of the cart	Displays items and uses Item::Stringify	Called in TextUI to display the contents of the cart
void ClearCart()	Called when Store::Checkout is called	Clears the cart	~	~
vector<Item *> get_items()	Used within Store to loop through items	Gets the items in cart	~	~
vector<Item*> in_cart_	Every store has a shopping cart with items currently in in_cart_	The items in the cart	Items are stored in this vector	~



Item	Store	Shopping Cart	Item	TextUI
Item(int id, string type, double cost, int initial_quantity)	~	~	Constructor to create an item and set variables	~
int get_id()	The id is used to check if item is in cart or inventory	The id is used to check if item is in cart	Get item id	~
int get_quantity()	The quantity is used to check if item is available	The quantity is used to check if item is in cart	Get item quantity	~
int get_cost()	The cost is used during checkout	~	Get item cost	~
string get_type()	Used in Store::Checkout to write to store file	~	Get item type	~
void IncreaseQuantity(int amount)	Used in RemoveItemFromCart(int item_id) to repopulate the inventory	Used in AddItem(Item * to_add) to increase the quantity of the item in cart	Increase item quantity	~
void DecreaseQuantity(int amount)	Used in Store::AddItemToCart to decrease the item in the inventory	Used in ShoppingCart::RemoveItem to decrease the quantity in the cart	Decrease item quantity	~
string ToString()	Used in Items(), CartItems(), and DisplayInventory()	Used in ShoppingCart::DisplayCart()	Stringify item elements	~
Item* Clone()	Used in AddItem to Cart(int item_id) to clone the target item and add it to cart	Used in AddItem to Cart(int item_id) to clone the target item and add it to cart	Clone and create a new pointer item	~
int id_	~	~	The item id	~
string type_	~	~	The item type	~
double cost_	~	~	The item cost	~
int quantity_	~	~	The item quantity	~



TextUI	Store	Shopping Cart	Item	TextUI
TextUI()	~	~	~	Constructor to create the TextUI object
string MainMenu()	~	~	~	Returns a string for the main menu
void RouteChoice(Store s, string choice)	Store is modified based on input choice	Cart is modified because store is modified based on input choice	~	Modifies a store given an input choice
int ItemMenu(Store s, map<int, string> items)	Used to modify the store given item choice	Used to modify cart given item choice	~	Takes user input to choose item
vector<string> main_menu_;	~	~	~	The main menu