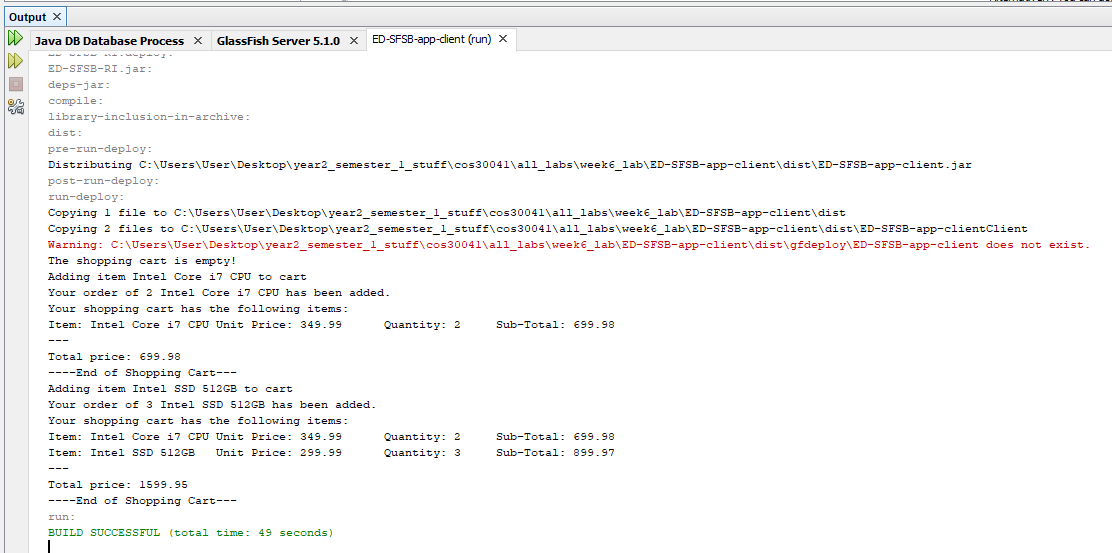
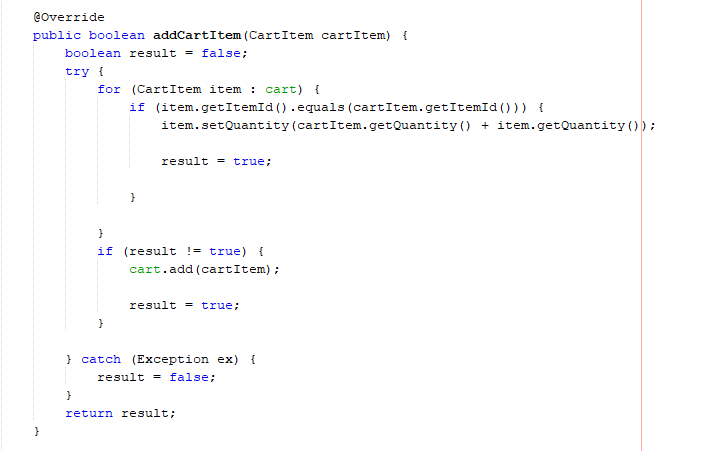
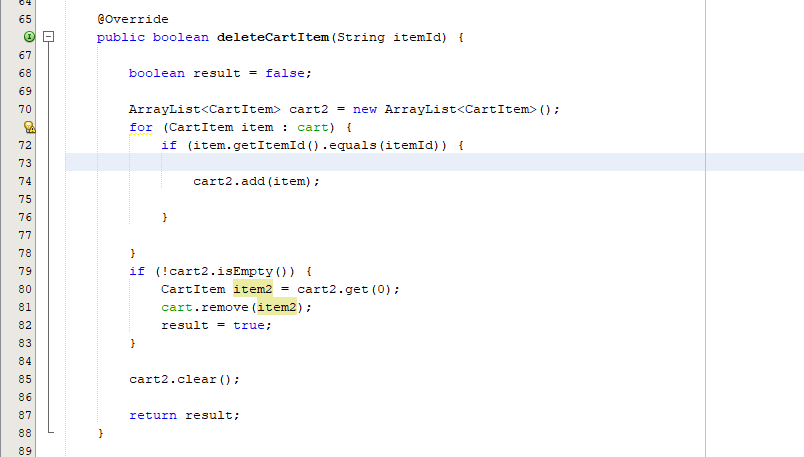
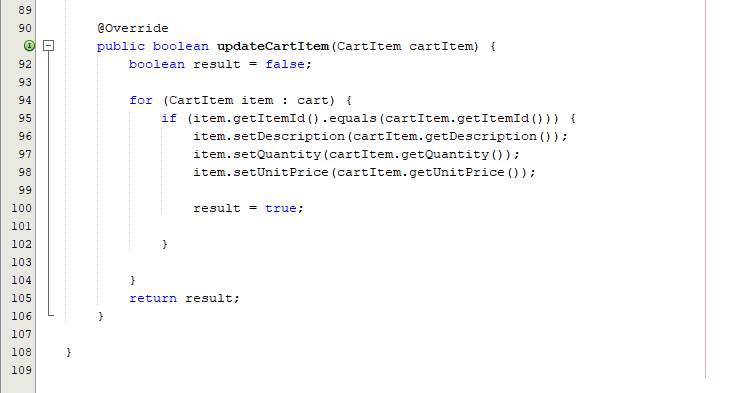
Step 1 ending part:

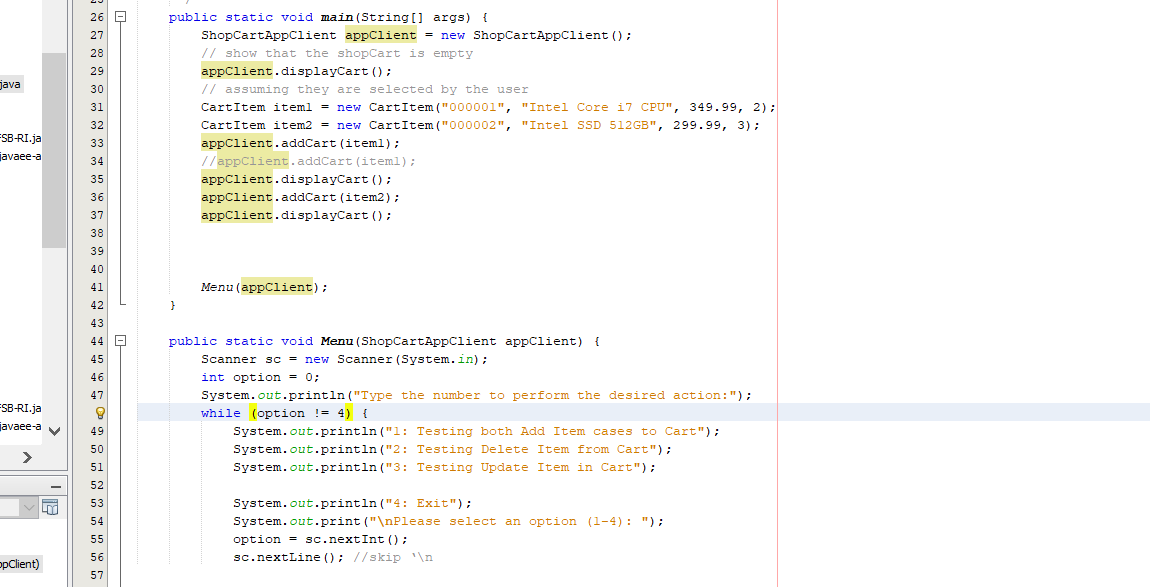


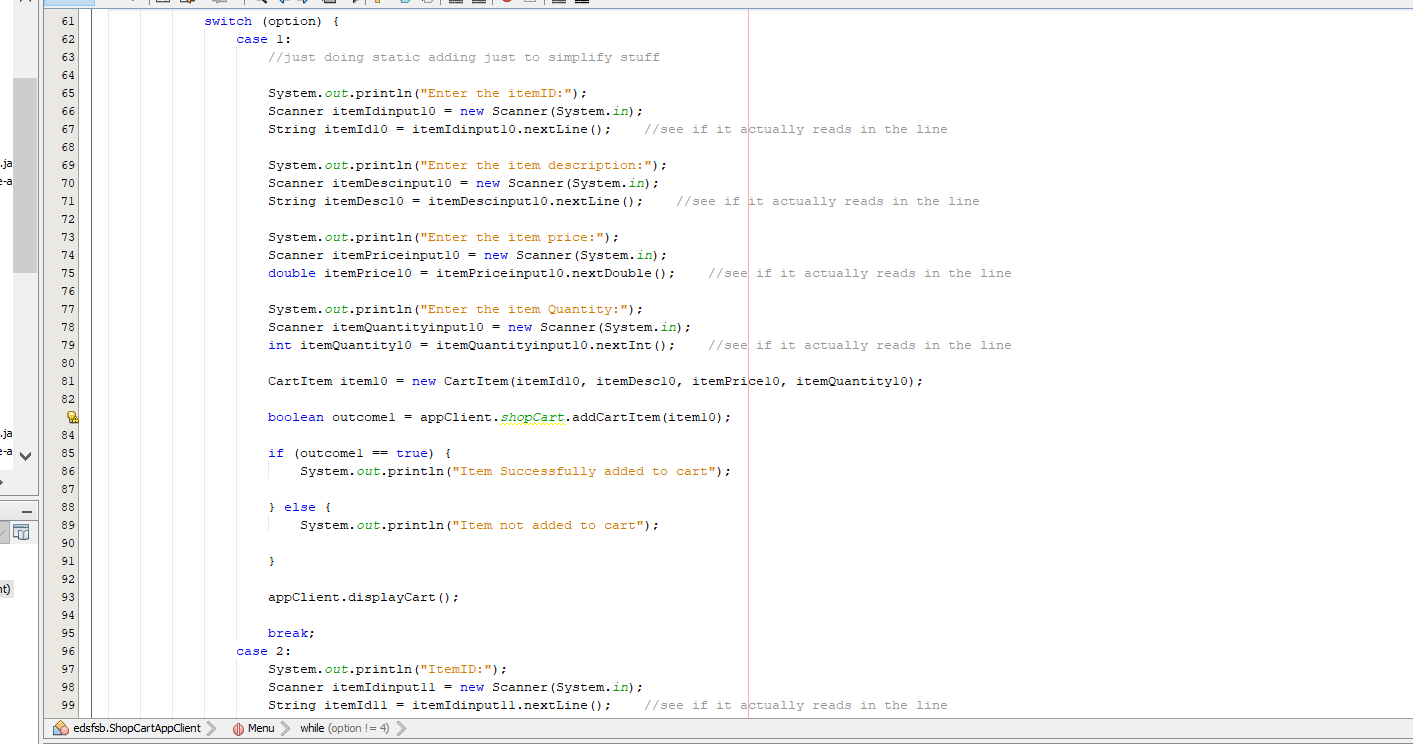
Step 2 all code parts:

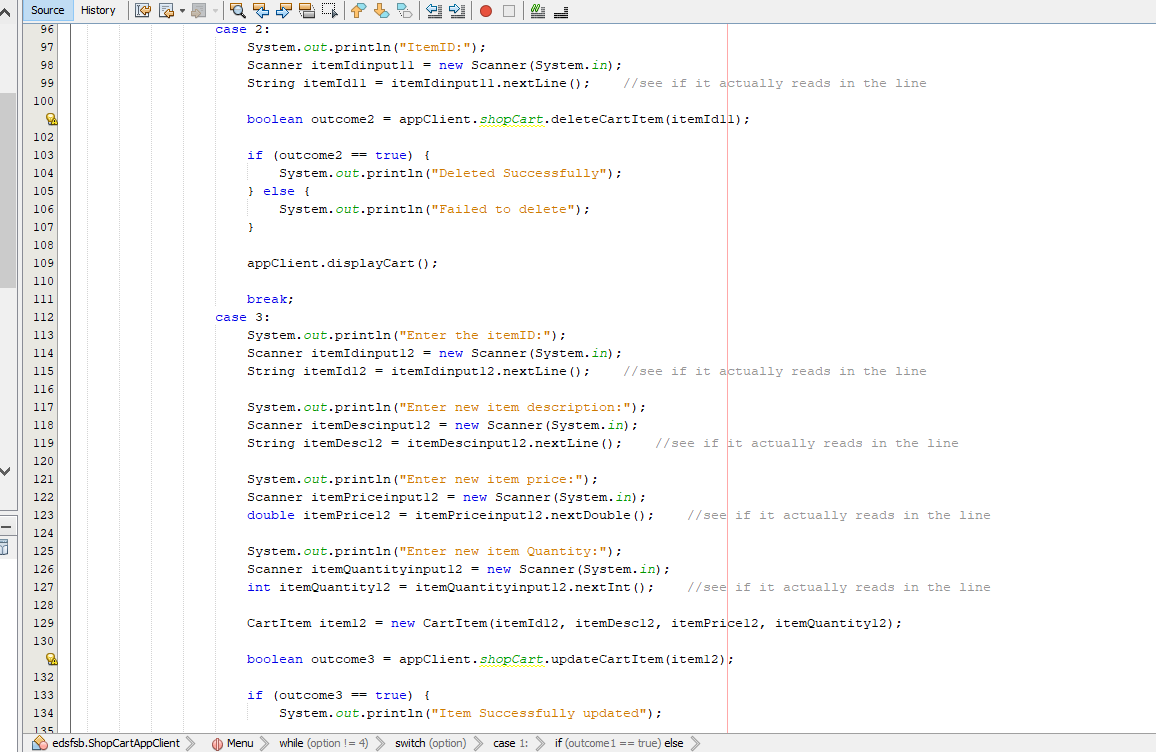


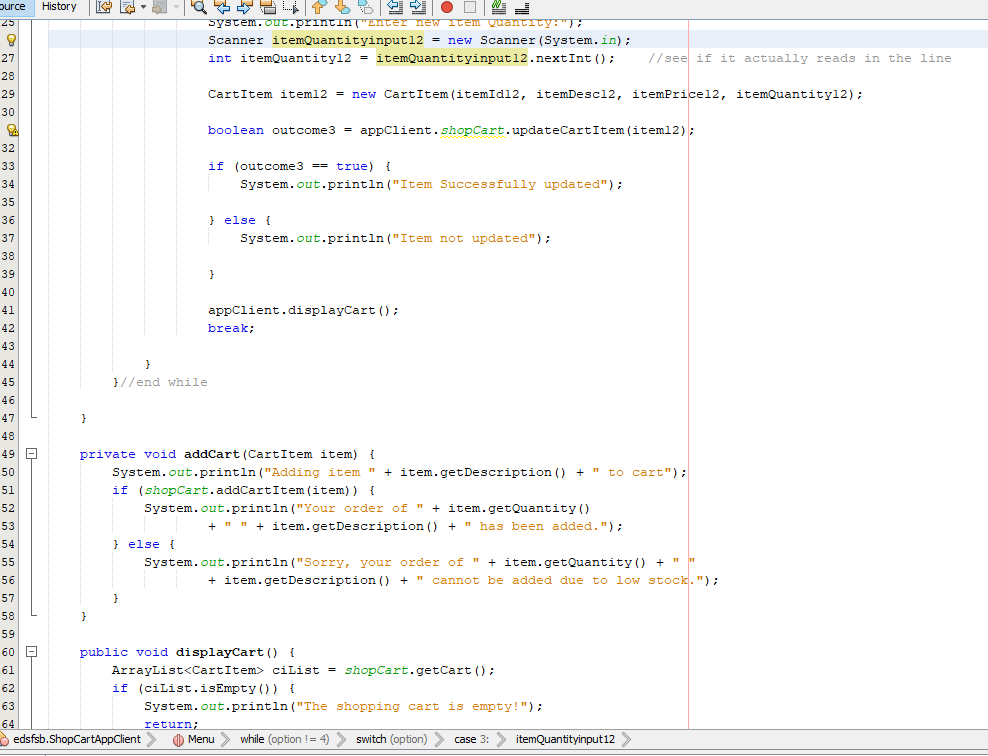


Task 3 code parts:



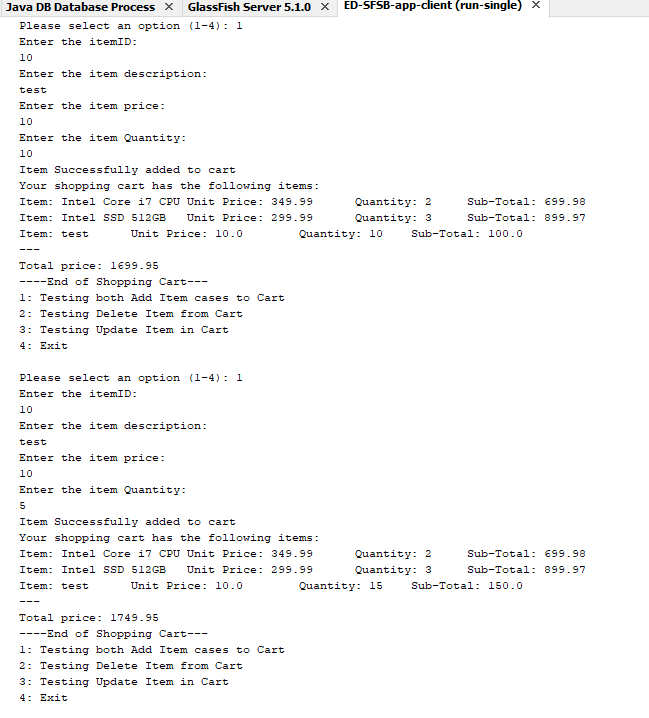




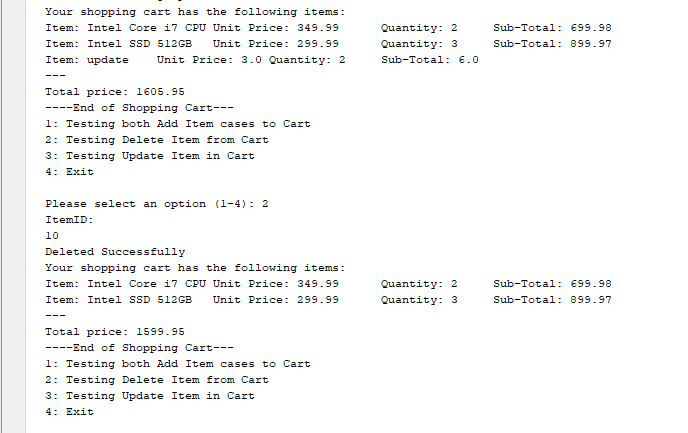


Task4:

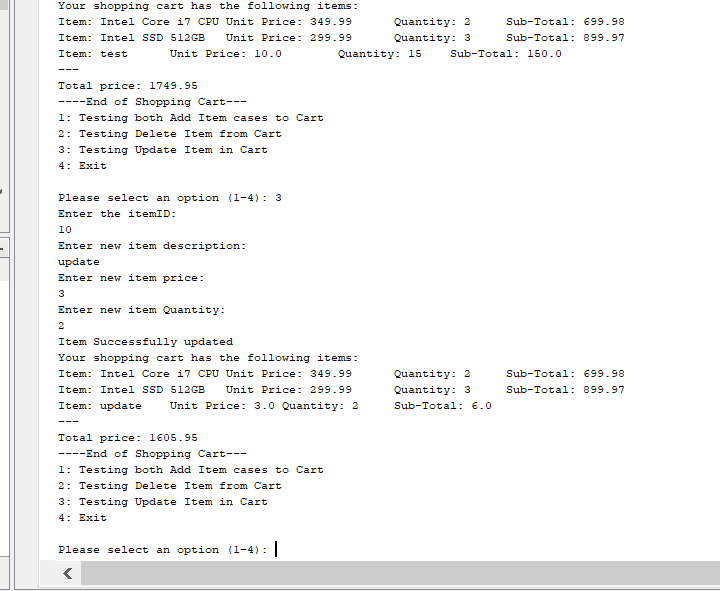
Output for adding items (both new item and existing item)



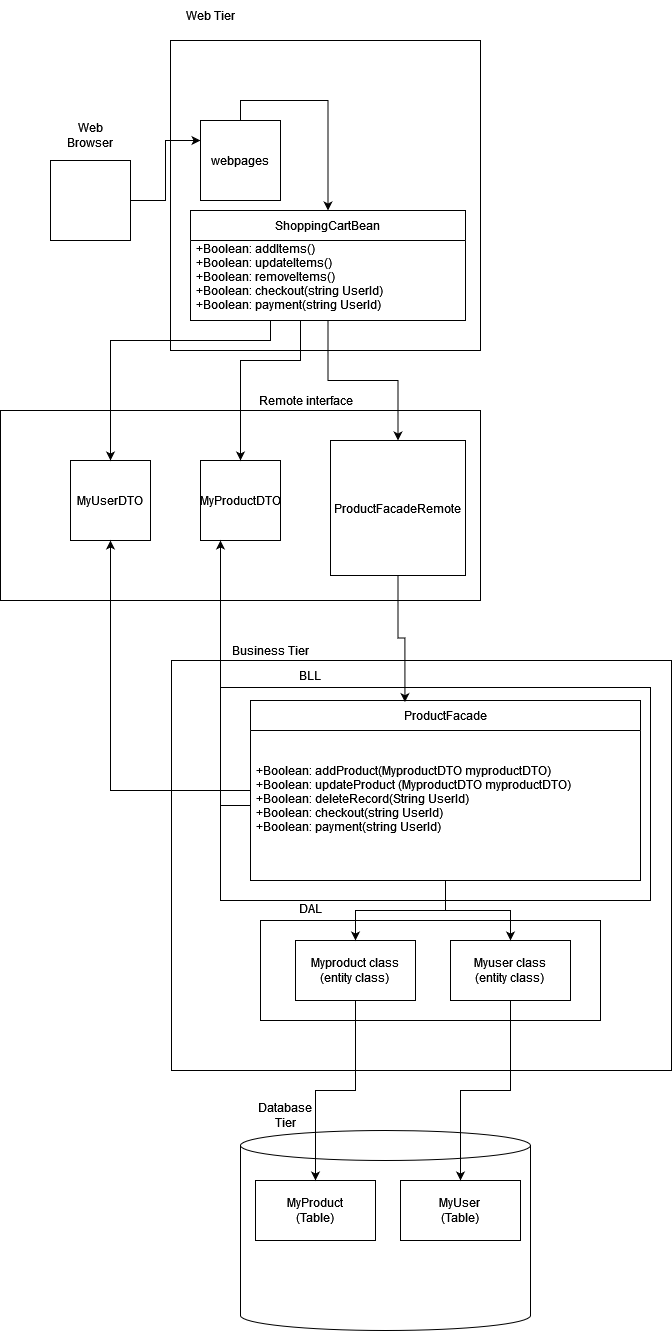
Output for deleting item:



Output for updating item:



Task 5:



Webpage: It is the UI (basically whatever the user sees)

ShoppingCartBean: This is the stateful managed bean. It basically keeps track of users and contains all the methods needed for the business logic to function

MyuserDTO: It is a DTO class that I would task to keep track of userID so that when customer comes again (assuming after certain period of time by which idle time has been triggered and browser has locked the user out), they would just input their user id and then the myuserDTO will pass it to the relevant objects (in BLL) to restore the user’s session (in UI) and bring him back to his previous page.

MyproductDTO: It is a DTO class that will keep track of all data transfer from web UI to BLL

ProductFacadeRemote: it is a remote interface class. It is used to call the ProductFacade class from ShoppingCartBean

ProductFacade: It is a Facade Class. It serves as an interface to simplify the codes and contains most of the basic logic being used in the ShoppingCartBean (except the basic SQL ones)

Myproduct class: It is an entity class. It contains the most basic knowledge needed to transfer data to and from the MyProduct table in Database

Myuser class: It is an entity class. It contains the most basic knowledge needed to transfer data to and from the MyUser table in Database

MyProduct: It is a database table that will store the product details for business purposes

MyUser: It is a database table that will store the user details for session purposes