

Project 2 Implementation.

Due date: Monday November 18. One per group. No extension possible on due date.

This is to be done on Github, one per group. The assignment requires you to create a FSM-based userinterface for the warehouse program. Start with the code developed for the Project 1 and replace the old user interface (UserInterface.java file) with the set of java files needed for the FSM-based version. Use the code for the Library FSM as a template.

NOTE: *The assignment is worth 120 points (text-only version), but you can earn up to 250 points (fully graphical version)*

Basic version of FSM-based interface (120 points. text only; no graphics).

The new UI should have 4 states: the OpeningState (or LoginState) will provide the option of going to ClientMenuState(user must provide client ID), ClerkMenuState or ManagerMenuState.

1. In the ClientMenuState, the Context has stored the ClientID for the current client; all operations are for that ClientID. The state will have operations for the following:
 - (a) Show client details.
 - (b) Show list of products (with price).
 - (c) Show client transactions
 - (d) Add item to client's wishlist.
 - (e) Display client's wishlist.
 - (f) Place an order.
 - (g) Logout. System exits to clerk or login state, depending on how we started (which has to be remembered in the context).
2. In the ClerkMenuState, we have the following operations:
 - (a) Add a client.
 - (b) Show list of products (with available quantity and price).
 - (c) Show list of clients
 - (d) Show list of clients with outstanding balance.
 - (e) Record payment from a client
 - (f) Become a client. The actor will be asked to input a ClientID; if valid, this ID will be stored in Context, and the system transitions to the ClientMenuState.
 - (g) Logout. System exits, depending on how we started, which has to be remembered in the context.
3. In the ManagerMenuState, we have the following operations:
 - (a) Add a product
 - (b) Display the waitlist for a product.
 - (c) Receive a shipment.
 - (d) Become a Clerk
 - (e) Logout.

IMPORTANT NOTES:

Use a matrix-based implementation for the FSM, centralized in the context.

Follow the LibraryFSM example in the samplecode.

No changes are needed to the backend. All we are doing is replacing the `UserInterface` class with the text-based “FSM interface”.

Graphical version of FSM-based interface. (250 points)

The requirements are the same as above. To earn 250 points all the interaction should take place through graphical components.

What to submit:

1. **Completion report (in D2L, one per group).** This should describe how the work was shared, and starid of the Github account where the code can be found.
2. **Code in Github.** In the Github repository of the starid specified above, create a folder called `WarehouseFSM` and place all the code in there. Code should be compiled and tested.
3. **Video demonstration of software.** *This should be done strictly as follows:*
 - (a) login in to your minnstate zoom account
 - (b) start a meeting, with screen share and audio; record to the cloud.
 - (c) while in the meeting, open you github repository and compile and run the program.
 - (d) demonstrate all aspects of you interface going through all the sequences described in Item 4 below. Please speak clearly, stating what you are doing.
 - (e) close the recording, the meeting and zoom session; after a little while you should be able to see the public link to the video in your zoom account.
 - (f) copy the link to the video, put it in a word/text file and place it in the dropbox in D2L (a separate dropbox will be provided for the link)
4. **Sequences to demonstrate for testing:**

Login as clerk, create five clients: C1 through C5; print/display all clients (should show the credit/debit balance for each). Logout.

Login as manager, create five products: P1 through P5, with quantities 10, 20, 30, 40, 50 respectively and unit prices \$1, \$2, \$3, \$4, \$5 respectively.

Become clerk. Print/display all products (should show price and qty for each)

Become client C1. Add to C1's wishlist: 5 each of P1, P3 and P5. Display C1's wishlist

Logout to clerk, Become C2, add 7 each of P1, P2 and P4 to wishlist. Show C2's wishlist

Logout to clerk, manager and back to login.

Login as C3. Add to C3's wishlist: 6 each of P1, P2 and P5. Print C3's wishlist. Logout

Login as C2. Place order for C2, buy everything (if available) in wishlist. Logout

Login as clerk, Print/display all clients (should show the credit/debit balance for each)

Become C3. Place order for C3, buy everything (if available) in wishlist. Logout to clerk.

Print/display all clients; display all clients with outstanding balance. Logout.