Project #33
Warehouse Manager

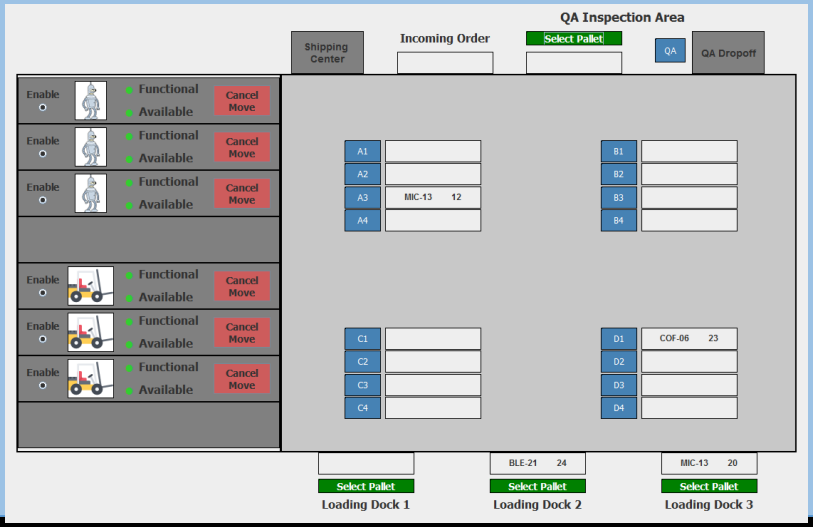
Dan Prendergast
Jon Magiera
Max Hollingsworth

CSCI 5448
Object-Oriented Analysis & Design

University of Colorado **Boulder**

Warehouse Management System Overview

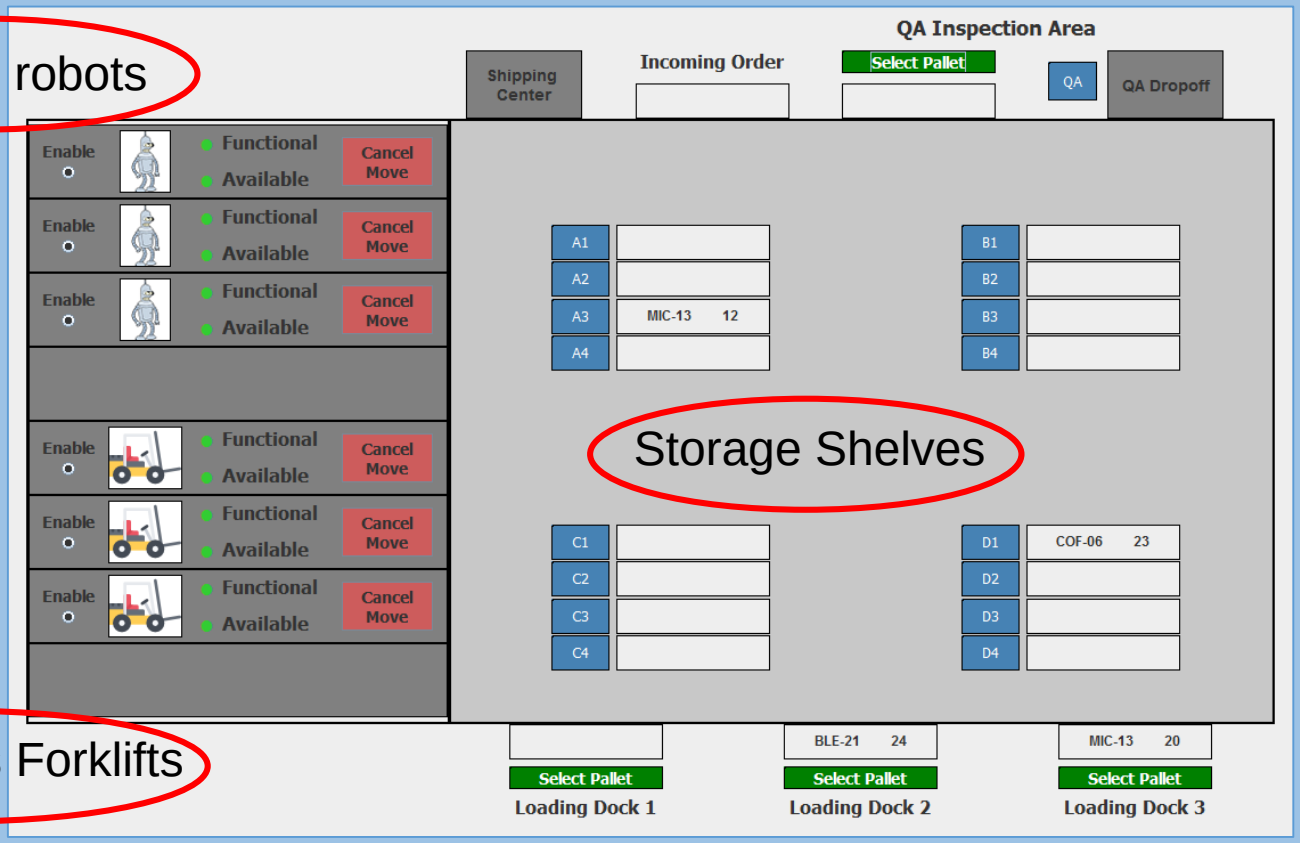
- System allows warehouse personnel to manage warehouse inventory and operations
- Manage the movement of products in a warehouse using automated forklifts/robots
- Track inventory as pallets arrive and product orders are fulfilled
- Database back-end keeps track of inventory



Warehouse Details

Shipping Center

Retrieval robots



Autonomous Forklifts

Loading Docks

Use Case ID:	UC-1
Use Case Name:	Move Pallet
Description:	Warehouse Operator directs autonomous forklift to move a pallet from its current location to a specified destination.

Actors:	Warehouse Operator (primary), Loading Dock Supervisor, QA Inspector, Autonomous Forklift
Pre-conditions:	• Pallet awaiting movement at the loading dock or the QA Inspection Area. • System displays current warehouse state (shelves, loading docks, robot status). • At least one autonomous forklift (AF) is operational and "available" for tasking.
Post-conditions:	• AF is "available" for additional tasking. • Products on pallet are added to warehouse inventory when placed on shelves.
Frequency of Use:	Near continuous (every few minutes)

Flow of Events:

	Actor Action	System Response
1	Loading Dock Supervisor or QA Inspector scans QR code on pallet.	System displays pallet and relevant information (product ID and quantity) at its current location.
2	Warehouse Operator (WO) selects pallet to be moved.	
3	WO selects pallet destination.	• System sends command to AF to move pallet from its current location to selected destination. • System indicates AF is "busy".
4	AF moves pallet from its current location to the destination.	• System indicates that AF is "available".

Variations:

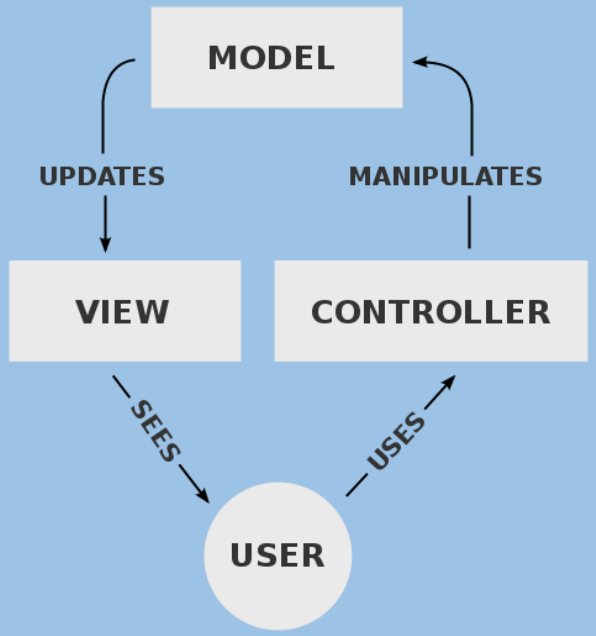
<u>UC-1a (Move Pallet from Truck to Shelf):</u>

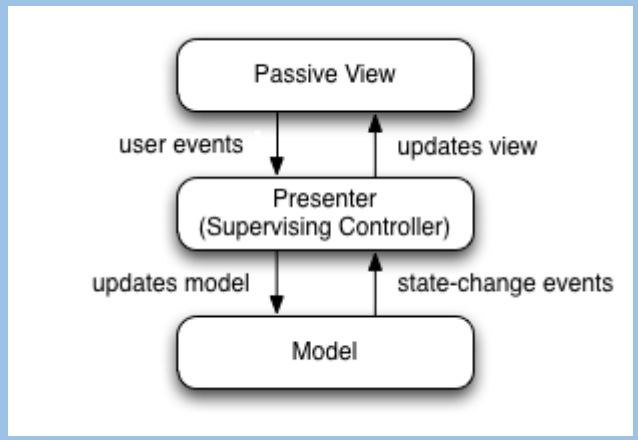
1. Loading Dock Supervisor scans QR code on pallet.
3. WO selects an empty shelf as the pallet's destination.
4. Actor: AF moves pallet from the truck to the selected shelf.
   System: In addition to system response listed above…
   - System adds products on pallet to warehouse inventory.
   - System displays products on the shelf.

Use Case ID:	UC-08
Use Case Name:	Fetch Product from Shelves
Description:	The Management System is notified of an order and a Retrieval Robot is sent to move the product from the shelf to the Packing Center.

Actors:	Packing Center, Retrieval Robot, Order System		
Pre-conditions:	An order is placed in the Order System. Order System divides orders into products and quantities. Product must be in stock. At least one Retrieval Robot must be idle.		
Post-conditions:	The correct quantity of the product is in the Packing Center awaiting shipment.		
Frequency of Use:	Potentially non-stop.		
Flow of Events:		**Actor Action**	**System Response**
	1	Order System notifies Management System of product and quantity to move.	System alerts the RetBot of the product and quantity. System updates UI to show RetBot is Busy.
	2	RetBot notifies System and Packing Center that product has been moved.	System updates the UI to show that RetBot is idle. System updates model and database with new inventory count.
	3	Packing Center makes product ready for shipment, then ships product.	

Model View Presenter vs. Model View Controller





Observer Design Pattern

Uses many-to-one dependency between objects so that if one object is modified, all of its dependents are updated automatically. -Gang of Four

```
public class ModelObserver implements Observer
{
    private WarehouseController warehouseController;

    public ModelObserver(WarehouseController warehouseController)
    public WarehouseController getWarehouseController()
    public void setWarehouseController(WarehouseController warehouseController)
    public void addObserver()
    {
        ArrayList<Pallet> pallets = warehouseController.getWarehouse().getPallets();
        for (Pallet pallet : pallets)
        {
            pallet.addObserver(this);
            pallet.indicateChange();
            System.out.println("pallet observed");
        }
        ArrayList<Forklift> forklifts = warehouseController.getWarehouse().getForklifts();
        for (Forklift forklift : forklifts)
        {
            forklift.addObserver(this);
            forklift.indicateChange();
            System.out.println("forklift observed");
        }
        ArrayList<RetBot> retbots = warehouseController.getWarehouse().getRetBots();
        for (RetBot retbot : retbots)
        {
            retbot.addObserver(this);
            retbot.indicateChange();
            System.out.println("retbot observed");
        }
    }
```

Link To Demo

Warehouse manager demo video