COMPSYS 704 Project 1 Written Compendium

By Jouveer, Jason, and Phoenix

Design Approach

Our initial plan for approaching the project was to divide the core elements and work on them individually while testing them with the other components as we go.

The main components of the project include:

- Automated Bottling System - ABS
  - Conveyor
  - Rotary Table
  - Capper
  - Filler
  - Lid Loader
- Product/Purchase Order System - POS
- Environment Control System - ECS
- Safety and Access Control System - SACS

The ABS was approached as a team while the other three components (POS, ECS, and SACS) were done individually.

Automated Bottling System

These components were planned and designed individually before we developed the logic using SystemJ in the component's respective controller and plant files.

Initial Design

Conveyor

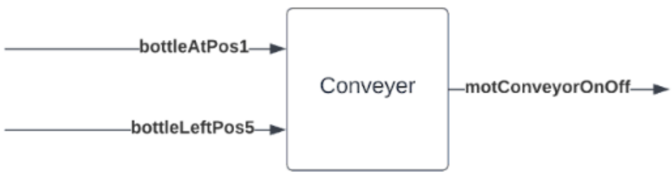


Figure 1: Conveyor Design

Rotary Table

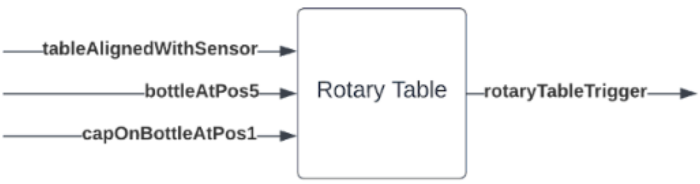


Figure 2: Rotary Table Design

Capper

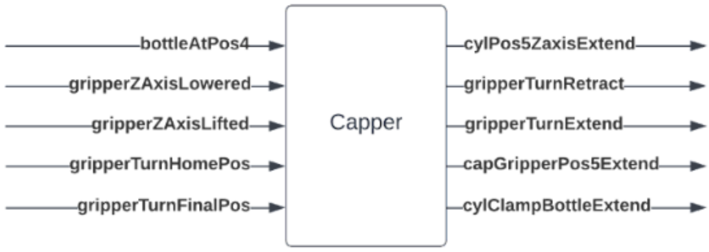


Figure 3: Capper Design

Filler

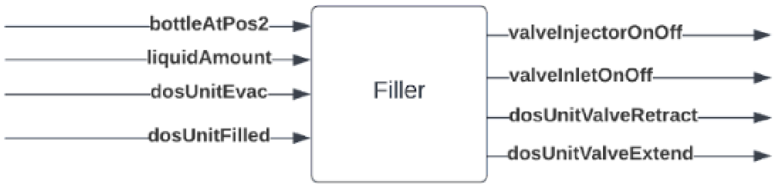


Figure 4: Filler Design

Lid Loader

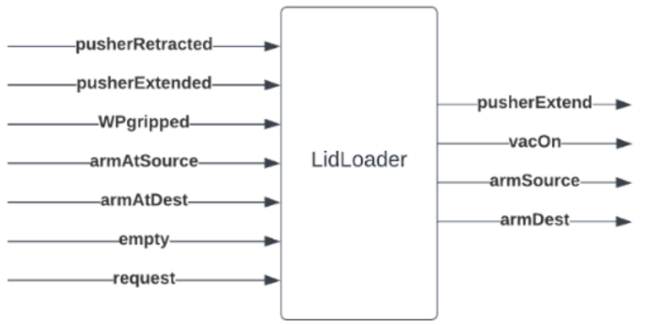
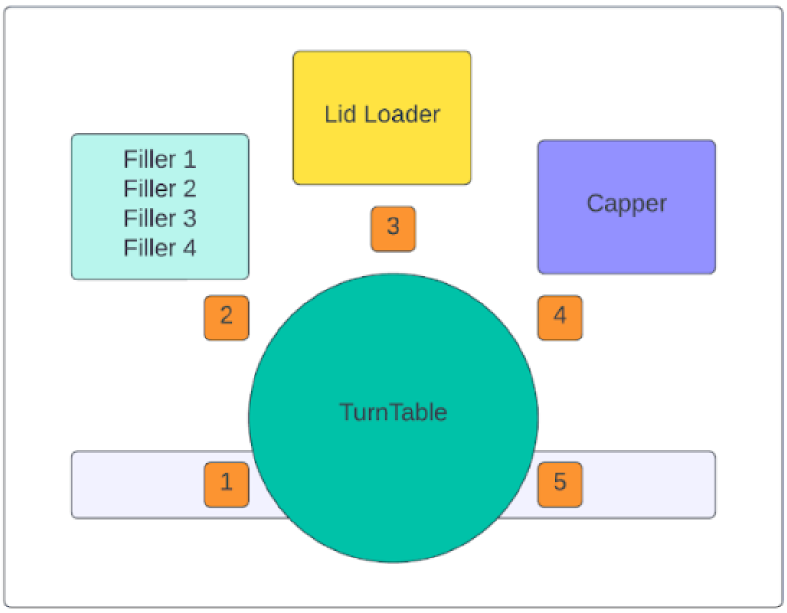


Figure 5: Lid Loader Design

GUI

Figure 6: GUI Mockup



Our approach to designing the GUI revolved around displaying all components of the ABS clearly in a logical order. At the bottom of the GUI is the conveyor belt wrapped around the turntable. Within the conveyor belt are numbered squares displaying the position of the bottle. These positions are necessary for signalling when the components should be active. The four fillers are in the top left at position two, lid loader in the middle at position three, and capper in the top right at position four. The decisions behind the design of the GUI are detailed in the design decisions part of the compendium.

Product Order System

The POS is used so that order can be placed, then given to the ABS. The ABS will take the information given by the user in the POS and schedule the manufacture of the required amount, and mix the liquids in the required proportions.
The user interface will be created in Java, like the user interface given in the last lab.

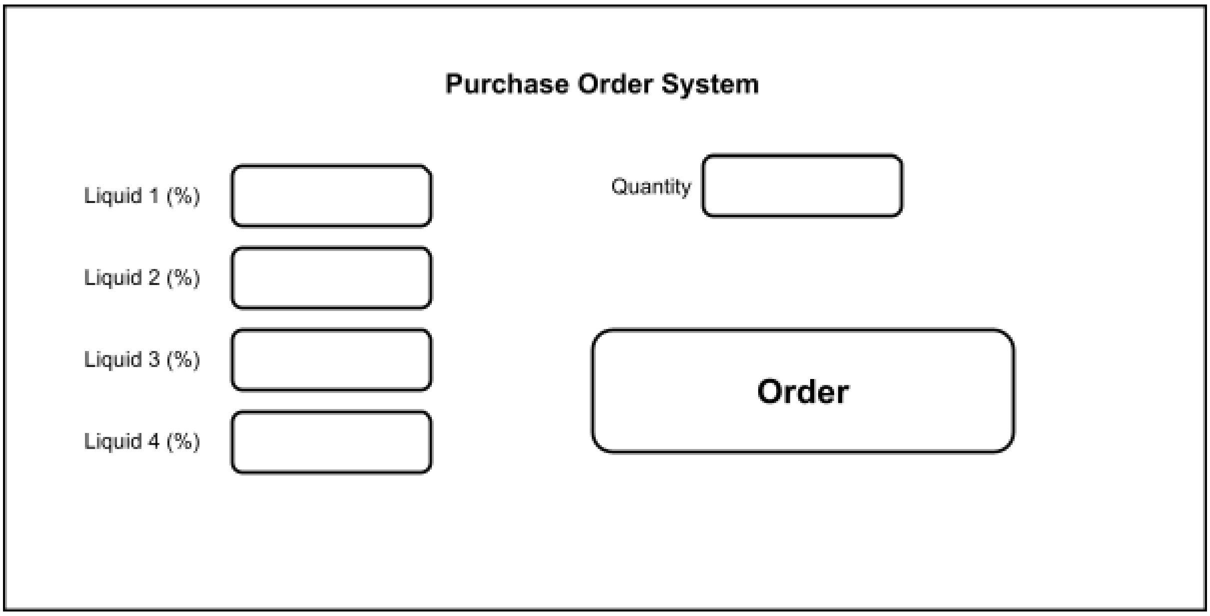


FIgure 7: Product Order System UI Mockup

Environment Control System

The ECS has been split into 3 different sections grouped 3 different sections. The ECS is used to monitor and control the temperature, humidity and lighting of the facility while taking into account, current temperature, occupancy, working hours. An additional feature the ECS has to have is a fire detection system where once detected, it will activate the fire alarm and possibly a fire suppressant system.
The ECS will only be active during working hours of the facility (Assumed to be 9am-5pm) and then will utilise 3 concurrent reactions to monitor the temperature of each section and will control heaters and cooling within said sections

Safety and Access Control System

The ACS's main goals are to allow certain cards to access certain areas and to suspend the bottling process when the card is present in zone 4, 5 or 6. Entry and exit from the facility is authorised through access cards and biometric information, the latter being optional. Along with the card system there must be a badge system that transmits the user's location.

The approach for the SACS is to develop various signals that signal where the card holder is using their card and if they have access to that zone. For the badge part of the SACS there will be signals corresponding to both the user's badge ID and the location of the badge.

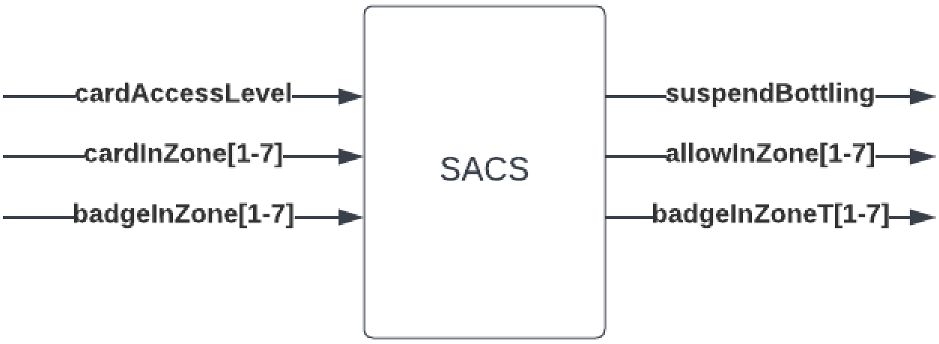


Figure 8: SACS Design

Final Design Decisions

Overall

We had decided to create the controllers to solely focus on controlling the ‘machines’ that they represent such as the Filler, Lid loader, Capper. Any other logic such as bottle tracking and communication between machines would be done when through the Plants.

Since there weren’t any plants given to us to simulate/test the logic, we decided to develop an accompanying plant file with each controller file. Separating the code into a controller and plant file for each component allowed us to manage the project easier and not affect other components while working on another.

A big decision we made was to have one central GUI that displayed all relevant information for the entire ABS rather than different UIs for each component. As seen in figure six the GUI shows all components clearly on one screen and displays where the bottle is during the process.

Automated Bottling System

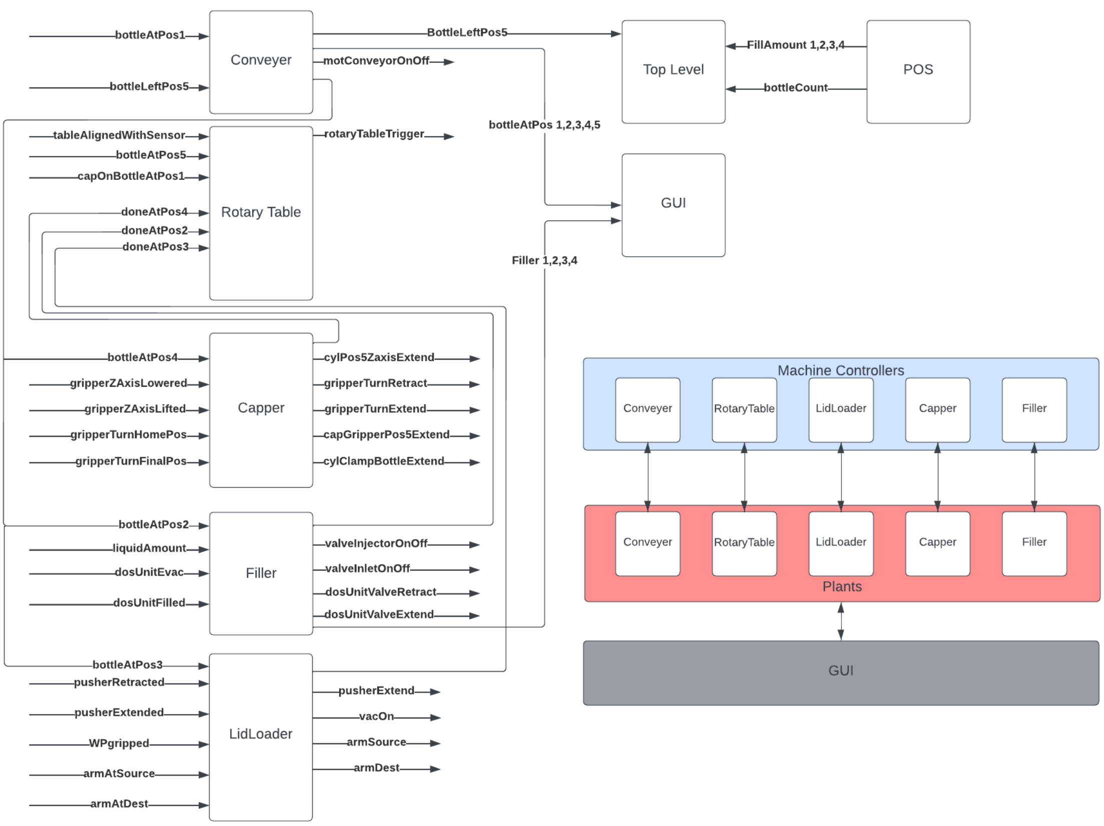


Figure 13. Overall Diagrammatic design of ABS

Conveyor

For the conveyor we decided to process one bottle at a time. This makes the bottling process simpler as we only need to track one bottle on the conveyor.

We decided to use the Conveyor Plant to keep track of the bottle positions during operation by receiving turn signals from the Rotary Table emitting bottleAtPos signals to the next machines (filler, lid loader, capper).

The ConveyorPlant also sends the bottle position information to the GUI so it can be displayed.

Rotary Table

For the rotary table's controller we decided to use four different abort statements for each position of the bottle. Within each abort statement is a trap within which sustains the rotaryTableTrigger signal, waits half a second and exits the trap (to simulate turn time of the table).

The rotary table has extra input signals for the controller so it can determine whether or not the machines have finished working and it is ok to turn through the use of doneAtPos* signals from the Filler, LidLoader and Capper.

Filler

For the filler's controller we decided to loop through the filler process four times. For the required four fillers, different liquids were implemented as 4 different fillers (filler1, filler2, filler3, filler4). We used an int signal to signify which filler would be currently processing and sent that information to the plant which then relayed the current filler to the GUI.

The plant was made to pass information from the filling machine to the GUI and to simulate the operations of the machine.

The decision for looping through the whole process 4 times because our understanding of the filler was not sufficient as the documentation for it was insufficient and there wasn't much luck in finding proper documentation on the machine.

Filler clock-domain (for a single liquid filler)

- Input signal(s)
    - bottleAtPos2 – Present when the bottle is at position 2
    - dosUnitEvac – Present when a pressure canister is at bottom
    - dosUnitFilled – Present when a pressure canister is at top
- Output signal(s)
    - valveInjectorOnOff – Turns on or off the valve injector (absence of this signal will turn off the injector)
    - valveInletOnOff – Opens the inlet valve (absence of this signal will close the inlet)
    - dosUnitValveRetract – brings the pressure canister to top
    - dosUnitValveExtend – brings the pressure canister to bottom

Lid Loader

For the lid loader we decided to simplify the operations by removing the ability to manually operate the machine through the enable button and instead made it fully automatic. It detects the signal 'bottleAtPos3' emitted by the conveyorPlant and automatically pushes a new cap, grips the workpiece and places it upon the waiting bottle and emits 'doneAtPos3'

Modifications were made to this machine such as removing the possibility of the machine not having caps within the magazine to simplify it as we had decided to create the ABS fully automatic.

Capper

For the capper we implemented the machine as stated in the brief with one added signal being 'doneAtPos4' to allow the RotaryTable to turn to its final position.

GUI

As mentioned previously, the major decision during the design process for the GUI was to encapsulate all components' tasks within one GUI rather than individual GUIs for each part of the ABS. This led us to the finalised design of the GUI where each part of the bottling process is displayed as well as the bottle's current position.

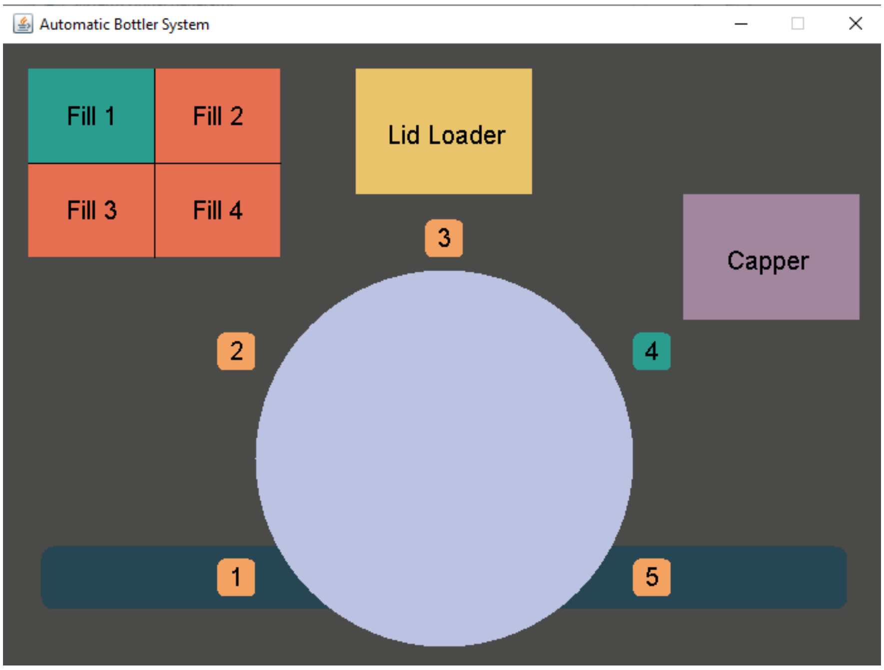


Figure 9: GUI Design

We designed the GUI using JPanel from the Java Swing package. JPanel allowed us to easily create oval and rectangle shapes to represent each of the ABS components. Depending on the state of the system we used draw functions to set the colour of the different shapes to display the current position and process the bottle is in.

Purchase Order System

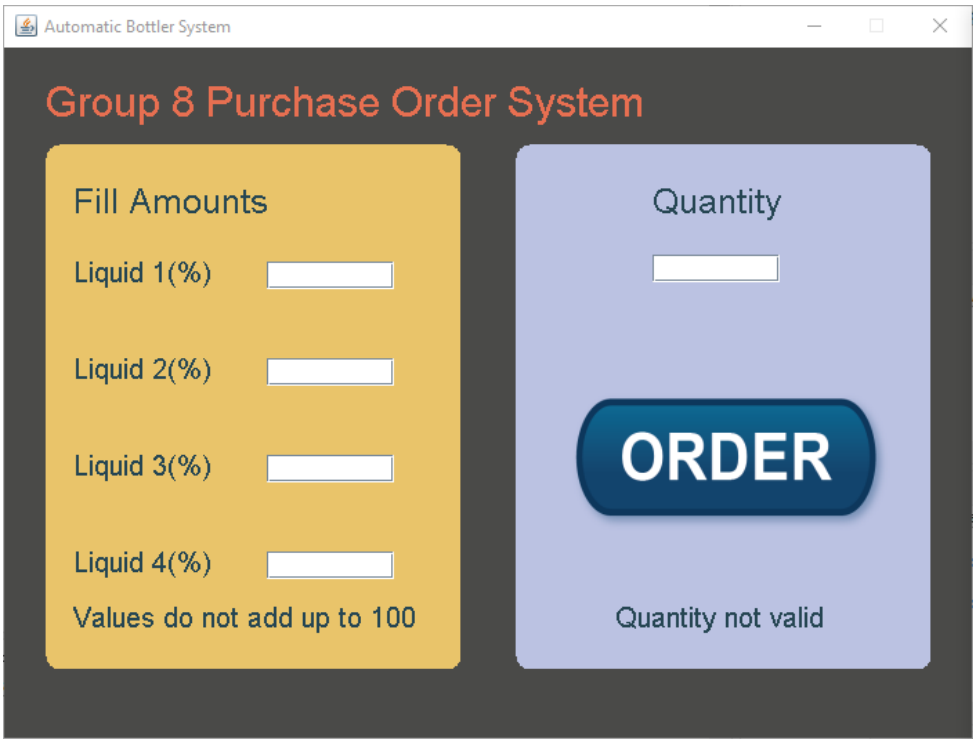


Figure 10: Purchase Order System

The purchase order system was revamped from the initial concept shown above into a better design and colour scheme. The system has checks in order to determine whether the user has entered a valid order before sending the order signal to the rest of the bottling system.

The purchase order system is running at the top level controller for the rest of the bottling system. This means that the flow of bottles into the first position is done by this system, and new bottles are stopped from being placed once the system has done all of the required quantity.

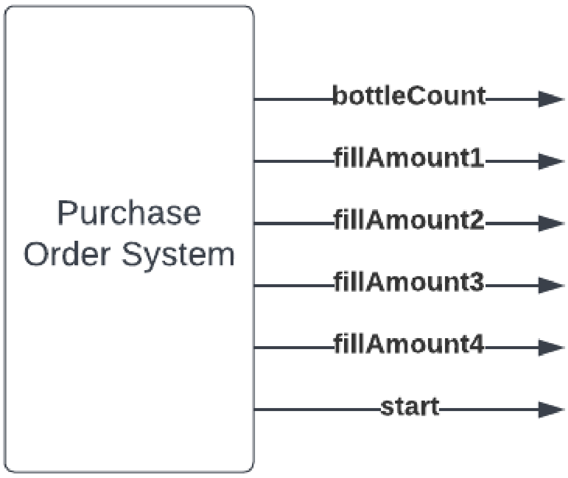


Figure 11: Purchase Order System Signals

These are the signals that were implemented for the purchase order system. The start signal was used to start the whole system when the order button was pressed. Additionally, the fill amount signals and the bottleCount signal were sustained with the appropriate information given by the user's input.

Environment Control System

The Environmental control system was successfully implemented, monitoring and controlling the Temperature, light, presence as well as being able to detect smoke coming from any of the 3 sections. The ECS's temperature control and lighting system is only active during the working hours of the day (0900 to 1700).

The Temperature control system reads the temperature of the current room and if it isn't at the desired temperature it sends a signal to activate the AC to bring the temperature to the desired level.

The lighting for the system is reliant on the time of day (working hours or not) and only activates when there is a presence detected within the section, which then turns the light on if the light level detected is lower than the set amount.
The fire detection system doesn't actively have any fire suppression system but will still sound a fire-alarm if there is any smoke detected within the facility

Working hours are simulated by simply using an integer and adding 100 to a 24 hour clock. Occupancy and fire/smoke are randomly generated within the 3 sections with a 2% and 1% chance respectively per code cycle to simulate occupancy and fire for the system.
The ECS has been split into 3 different sections grouped 3 different sections. As shown below

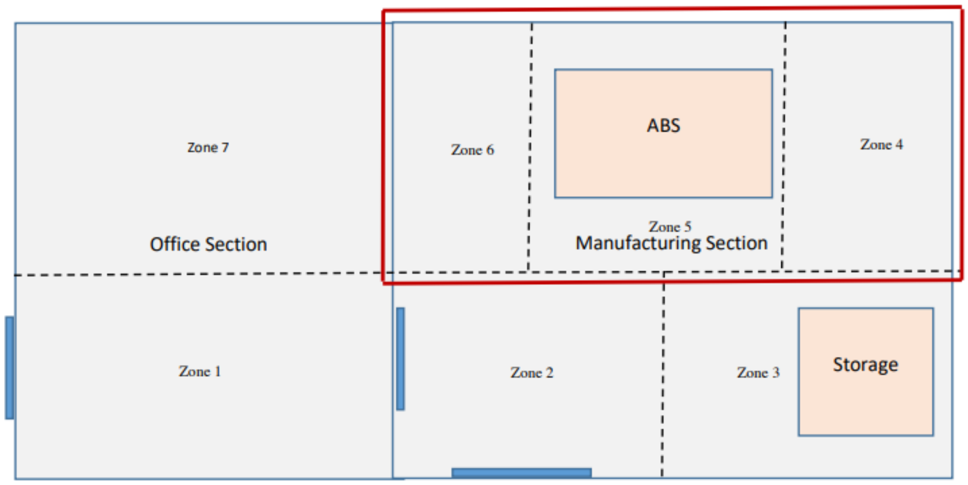


Figure 12. Manufacturing facility conceptual layout

Grouping	Zones
Section 1 (Office)	1, 7
Section 2 (Storage)	2, 3, 4
Section 3 (Manufacturing)	5, 6

Safety and Access Control System

The safety and access control system was decided to run three processes concurrently:

1. Card allowing access to zones
2. Badge location transmission
3. Bottle process suspension

The card allowing access to zones is achieved by two inputs. The first input is a signal with an integer attached which is either 1 or 2. 1 allows access to the manufacturing zones (4, 5, and 6) and 2 allows access to the office zone (7). Present statements are used to check if an input signal signalling when the personnel is attempting to enter a zone and the input signal signalling the card's access level. If the zone that is being attempted to be accessed corresponds with the card's access level, entry is granted.

The badge location transmission is achieved by constantly checking different inputs per zone to see if a personnel is active within any of the zones. If the badge is detected in any zone a signal is emitted.

If the zone entered or badge location transmitted is in the manufacturing part of the building a bottle process suspension signal is sustained to suspend all bottling processes for safety purposes.

Although there are other ways to develop the safety and access control system these are the decisions that were made.