Workshop - 10

Workshop Value: 10 marks (6% of your final grade)

Learning Outcomes

Upon successful completion of this workshop, you will have demonstrated the abilities:
- to decipher and identify a problem
- to analyze and decompose a problem
- to identify the required detailed steps to solve a **larger problem** using **modularity**
- to communicate the solution to fellow peers and non-technical business persons

Workshop Grading and Promotion Policy

Workshops for this course will be assessed using the following criteria:
- Workshops must be completed during the class time to be graded
- You must successfully complete 9 workshops (if more than 9 are completed, the best 9 will be used)
- Each student is expected to be a presenter of the workshop solution at least once by the end of the term
- Workshop solutions and presentations will be evaluated using the published workshop rubrics

---

Workshop Overview

Vending machines now almost run without any human maintenance. They employ the "internet of things" (**IoT**) enabling real-time updates on stock levels, payments, and alerts for machine maintenance. Electronic payments use both swipe and "tap" technology for debit and credit cards and even accept cell phone payments using "near field communications" (**NFC**). No physical money is stored! Maintenance costs are drastically reduced with these enhancements over the older models as routine inventory and money pickups are eliminated. The only time a service provider needs to physically visit the machine, is for restocking inventory and addressing any general mechanical maintenance (which would be infrequent). **But how should this all work?**

Reminder

Always start by using **pseudocode** to develop the detailed step-by-step process. Once you are satisfied with your pseudocode solution, develop a flowchart.

Workshop Details

Applications of vending machines essentially have two main logical components to define:

1. **Hardware States** (physical interface)

2. **Software Logic** (main functional logic/controller)

States

Hardware states are triggered by the software layer that implement the overall system process. The system has 6 main states (or modes):

1. **Power-On**
2. **Power-Off**
3. **Idle/Ready** (stand-by: waiting for customer)
4. **Active** (customer interaction building an order)
5. **Payment** (final step in customer transaction)
6. **Cancel**

You define these states

For each of these states, <u>both</u> the **hardware** and **software** components will have their own set of defined processes.

Constraints

- **<u>Hardware States</u>**
  - Hardware components have only two possible states: "**enabled**" or "**disabled**"
  - For each machine state, the various hardware components should be set to complement the software logic (example: disable controls that don't apply to the current state and only enable those that should be)
- **<u>Product Selection</u>**
  - Limited to a single letter (column) button followed by a single number (row) button sequence (ex: "A" button + "9" button)
  - Products can only be added to a transaction if the requested quantity for the item is in-stock (has inventory)
- **<u>Product Inventory</u>**
  - Following payment, inventory must be adjusted in real-time (hint: local and remote data)
  - Inventory minimum stock levels must be enforced (if the quantity in-stock reaches the minimum stock level set for that product's row & column location, more products should be ordered
- **<u>Acceptable Payments</u>**
  - Credit card (tap/swipe)
  - Bank card (tap/swipe)
  - NFC phone payment
  - Phone application payment
  - **$$ NO CASH $$**

Hardware Controls/Interface (*see last page for illustrations*)

- <u>Payment Module</u>: Physical hardware for reading credit cards, debit, NFC, Tap etc.
- <u>LCD Colour Screen</u>: 10" wide X 6" high (10 cm X 15 cm)
- <u>Column/Row buttons</u>: Column-letter buttons (A-E), row-number buttons (1 – 10)
- "<u>Enter</u>" <u>button</u>: Adds a product selection to the session (transaction), or applies entered quantity
- "<u>Correct</u>" <u>button</u>: Used to backspace an entry/quantity, NOT remove an already added item
- "<u>Cancel</u>" <u>button</u>: Cancels the entire session and resets
- "<u>Pay</u>" <u>button</u>: Triggers the payment process and finalizes the session

Data Structures

Apply the following data structures in your solution:

"Product" (*"Inventory"* is simply a **collection/array** of "Product" data**)**
  slotID   // Unique location slot ID (where it is physically placed in the machine ex: "D8")
  sku   // Unique product identifier
  quantity   // Actual quantity available (physically in machine at the given slotID)
  maxQuantity   // Maximum machine qty that can be stocked for the slotID
  minQuantity   // Re-order when this qty is reached (based on: maxQuantity - quantity)
  price   // Vending machine price to charge customer per unit
  description   // Product name

"Transaction" (*"Session"* is simply a **collection/array** of "Transaction" data**)**
  slotID   // Unique location slot ID (where it is physically placed in the machine ex: "D8")
  quantity   // Requested quantity
  price   // Price per single unit quantity
  description   // Product name

Your Task

IMPORTANT: Divide the work up among your group members where possible to ensure you complete the requirements on-time. You will need to discuss the overall flow and overlapping logic periodically to ensure your solutions piece together seamlessly.

1. **Hardware**:

   - For each state, create the necessary **FLOWCHARTS** (pseudo code not required)
       o Essentially, define what controls need to be **enabled** or **disabled** per state
       o You should have 4-flowcharts in total *(this should take no more than 15 minutes)*

   Hint: the software logic layer will "call" these processes as required when changing states

2. **Software**: For each state, define how the machine should work (***pseudo code*** and ***flowchart***)
   - You should have 4 separately defined pseudo code processes
   - You should have 4 separately illustrated flowcharts

3. **Present Solution**:
   - The instructor will let you know at presentation time what states to present (there will not be time to cover the entire solution)
   - Present the assigned state solutions to the instructor or class (single student)
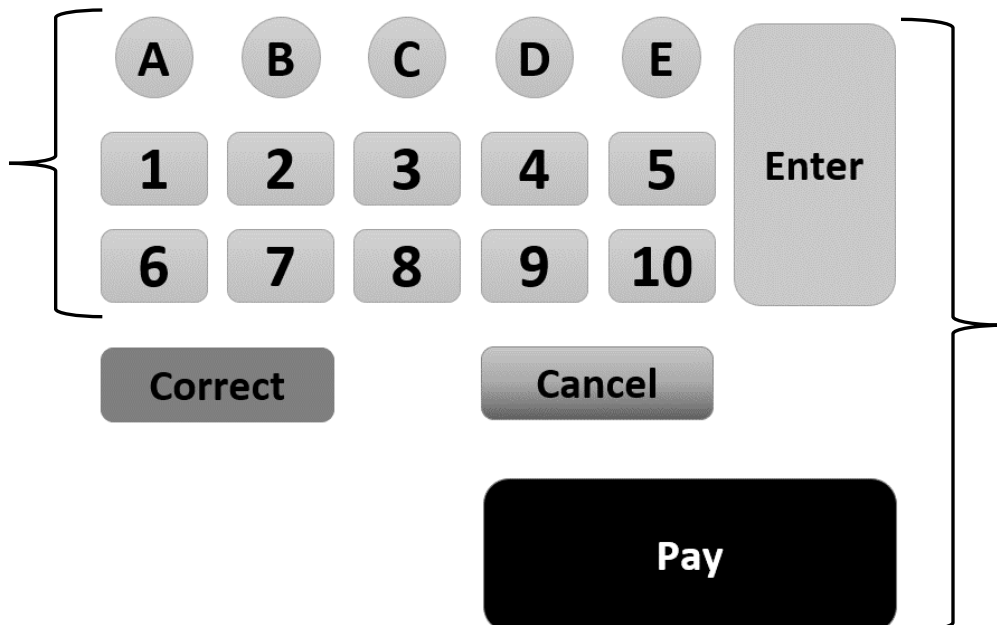
Payment Module

LED - Screen (NOT touch screen)

PRESENT CARD
EUR 5.00

Vending Machine

LCD Display Screen

Be Creative As To What Should Be Shown Here!

Column/Row Buttons

A	B	C	D	E
1	2	3	4	5
6	7	8	9	10

Enter

Correct

Cancel

Pay

Command Buttons