SecureCash Access System

Overall Description

The SecureCash company manages several banks in Poland. Principally, the company is responsible for handling large sums of cash.

Each SecureCash centre has some zones. All centres require a system that controls access to their zones and keeps track of people in each zone for security reasons. Access to a zone is through a door, and to get through the Door an access card is required.

Each Door connects a source zone to a destination zone and works in one direction only. The employee would insert a card into a card reader at the Door, and the card information read. The system would then check whether the card is allowed through the Door to the destination zone.

A card is only allowed through a Door if it meets the conditions for entry. If these conditions are met, the system opens the Door and allows access to the destination zone. It also updates its records to show that a card (and the employee using it) has left the source zone and entered the destination zone.

Zones

Each Zone has a name and a maximum number of employees (capacity) that can be present in the Zone at any one time. Each Zone must maintain a list of all cards/employees currently in the Zone. These lists are updated whenever members enter or leave a zone, and it should always be possible to say how many employees are in the Zone and who they are.

Principally, all SecureCash centres have the following zones:

- Secure Zone
  - accessible only to Secure employees
  - max capacity: 2 employees
- Operations Zone
  - accessible to Operations and Secure employees
  - max capacity: 5 employees
- Transaction Zone
  - accessible to all SecureCash employees
  - max capacity: 7 employees
- Outside Zone
  - the default zone for all SecureCash employees who are currently not at work
  - max capacity: unlimited

cont...

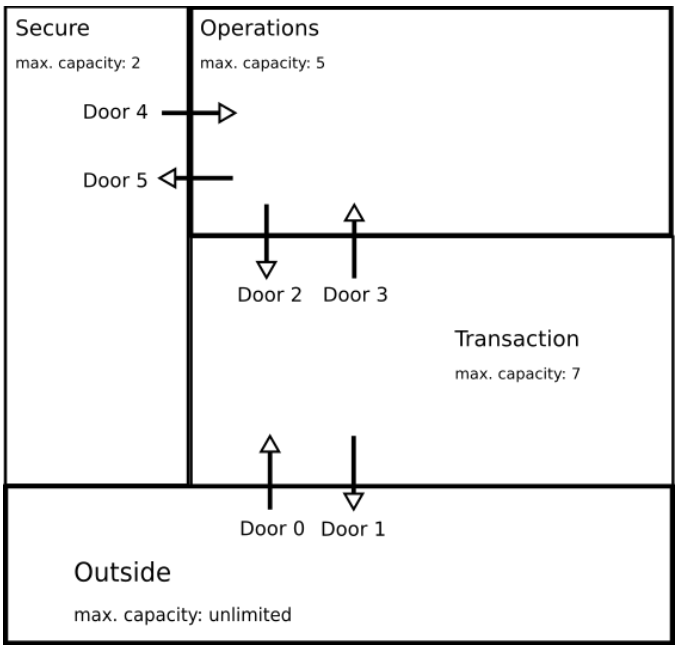
Cards

An employee of a SecureCash centre is only allowed to have one type of card. All cards include a card number and name. Employees cannot use cards of one centre at another centre. The following are the different types of cards for SecureCash employees.

- Managers
  - Always have access to all zones without any restrictions.
  - Card numbers less than 100
- Secure employees
  - Have access to Secure, Operation and Transaction zones
  - Card numbers from 100 to 200
- Operation employees
  - Have access to Operation and Transaction zones
  - Card numbers from 201 to 500
- Transaction employees
  - Have access to Transaction zones
  - Card numbers from 501 to 999
- Janitors
  - Have access to all centre zones, on condition that at least one other employee who is not a Janitor is present in the Zone.
  - Card numbers from 1000 and over

Assignment instructions

1. Create a new **MediBank SecureCash Centre** which contains classes/objects representing the following **Zones** and **Doors** as follows:



cont...

2. Create also some **MediBank SecureCash** centre employee **Cards**:

No.	Name	Type
22	Ivan	Manager
123	Seth	Secure
107	Debra	Secure
186	Gilbert	Secure
230	Vera	Operations
412	Lucy	Operations
254	Janet	Operations
665	Ramon	Transactions
725	Tracey	Transactions
1032	Lewis	Janitor

> For testing purposes you may have to invent some more of your own.

3. Create a **Tester class** that creates the MediBank SecureCash centre object. Through a number of methods, you should simulate the whole system and test that the system is working according to the specification, ex.
   - Move an employee card from the Outside zone to the Transaction zone.
   - Move a card from the Transactions zone to the Operations Zone.
   - No more than 2 employees are in the Secure Zone
   - Door will not allow entrance to a Janitor if there is no other employee in the Zone.
   - No Transaction employee is allowed in the Operations Zone.
   - Etc… **Important**: All test results should be output to the console.

4. Interacting with the system through a GUI: Apart from being able to run the tests through the terminal, the implementation should also enable a user to interact with the system using a basic Graphical User Interface accessible through the browser.

   a) A crude, yet acceptable GUI, can be achieved by using the standard HTML elements combined with some basic CSS styles and the use of the console. Alternatively, if you wish to enhance the look & feel of your GUI, you may also include and use any other external libraries/frameworks which you consider helpful. (Bootstrap, React, etc…)

5. After completing the project, fill in the Project Submission Form, ZIP your project and submit it on Moodle.