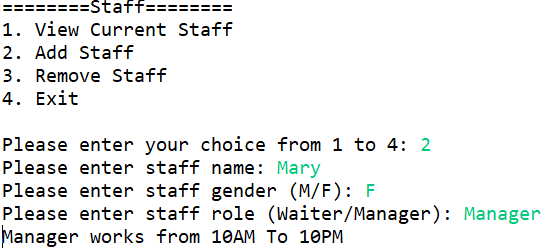
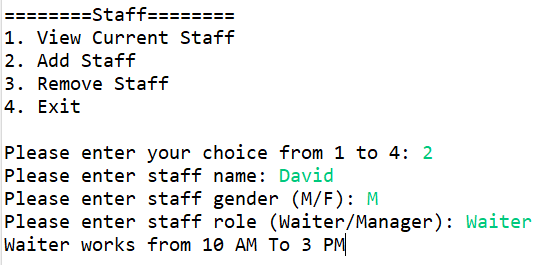
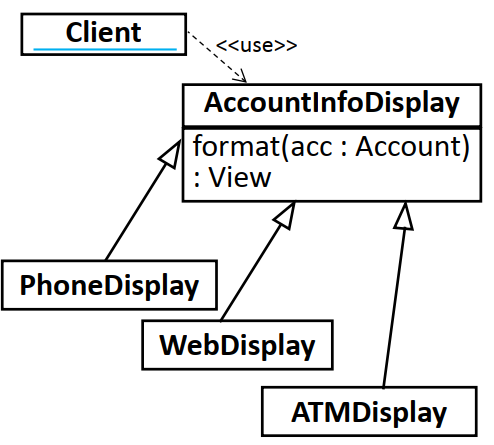
**Wut**

**--------------------------------------------------------------------------------------------------------------------------------------**

SOLID Principle

* Single Responsibility Principle
  + All our classes are tied to one entity in the system (Staff UI / Staff Control / Staff for Staff-Related)
  + All classes have a single responsibility (UIs to get user input / Control to Retrieve & Store Data)
* Open / Closed Principle
  + We have an abstract class for UI and all the different UIs just extends from this which allows us to create different UIs for different entities
* Liskov Substitution Principle
  + Any Method That Requires A Staff Class Can Take In A Waiter Object And Still Work As Pre-Conditions Of Waiter Are No More Than Staff And Post-Conditions Of Waiter Are No Weaker Than Staff
* Dependency Injection Principle
  + Higher Level Module Such As Order Depends On Staff And Not Waiter And Managers So They Do Not Depend Upon The Lower-Level Modules Of Waiter And Manager But Staff So Order Is Independent Of The Details For The Different Types Of Staff
  + Staff Have Different getWorkingHours()
*  



Reusability is high as classes can be used in other projects

* Staff can be reused in other projects that requires an employee class
* Reservation system can be reused in other restaurants by changing the reservation control to fit the conditions of the other restaurant (how long before reservation is void)

Extensibility is high as the system can be changed to fit other retail businesses

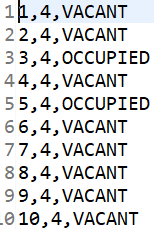
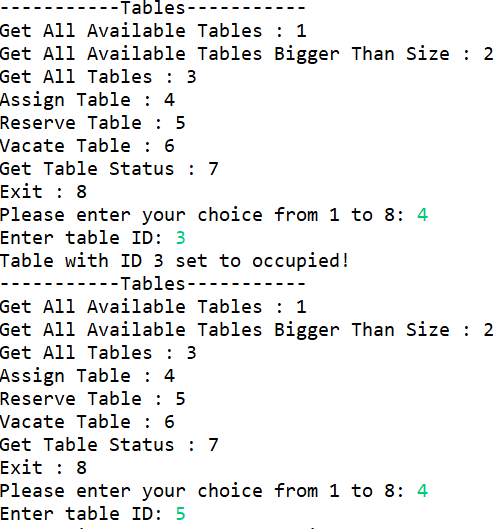
* For shops
  + Change staff roles in staff
  + Change table to counter
  + Change menu to the items the shop sells
  + Remove reservation if there is no such system
* Other restaurants
  + Change menu and setpackages to fit the store

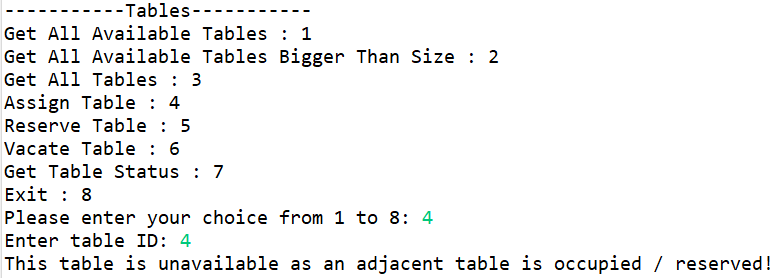
**Additional Features**

Leaving Adjacent Tables Empty – Change Reserve / Assign Table To Check For Adjacent Tables

– All Other Classes Like Invoice And Sales Are Unchanged Due To SRP

Example – Table 3 and 5 are occupied so table 4 cannot be assigned





Temperature Taking – Extend Staff To Have A Sub-Class Waiter

– Make Waiter Take Temperature

Discount For Takeaways – Duplicate All Items To Have A Takeaway Version

– All Other Classes Like Invoice And Sales Are Unchanged Due To Single Responsibility Principle