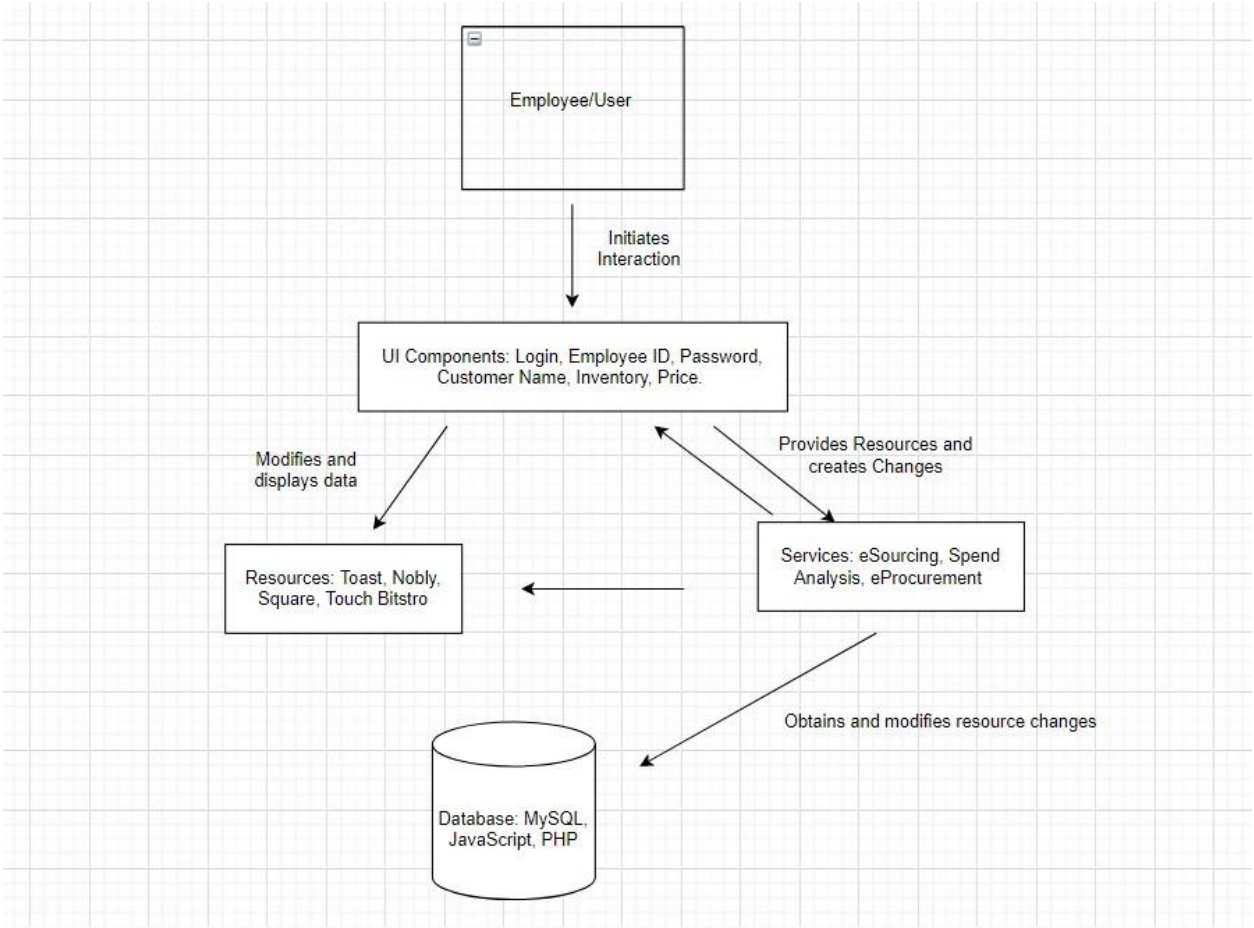
Ty Davidson

7/11/2021

Four Leaf Service Layer

I was able to create a blueprint diagram of the service layer and how I intend for it to work. This diagram includes the user/employee utilizing the UI components. When utilizing and modifying the components, it will respond to the services and resources. Everything will be updated and obtained in the database.



To create a user profile, the employee will use a POST request. Name and EmployeeID will be the required login credentials. The user will use a POST request to create customer profile as well. The user will then use a GET method to acquire the customer's purchase history. The POST method will then be used when the customer purchases an item. Finally PUT will be used to update inventory and customer's purchase history. As astretch feature I would utilize GET to count the remaining inventory available in the system. And I would use PUT to update the inventory count after every purchase.

Endpoint	Operation/Data	Request Type	Request	Response	Purpose
Data					
	name/employee ID	GET	/user/login	Login { 'success' boolean message string }	Logs the user into the system
	required field check	GET	/app/validate	Validate { 'success' bolean message string }	The required fields are populated
	ERROR wrong requirements	GET	/app/validate/ERROR	Error { 'Failed' Enter Requirements boolean message string }	Re-enter suggested requirements
Data Creation					
	create user	POST	/user/create	CreateUser { success boolean message string }	Creates the user with employee ID
	create customer profile	POST	/user/customerprofile	CreateCustomerProfile { success boolean message string }	Creates customer profile with name
	item purchase	POST	/user/customerprofile/itempurchase	CreateItemPurchaseHistory { success boolean message string }	Create record of item purchase
	history log record	POST	/user/customerprofile/history/create	CreateHistoryLog { success boolean message string }	Creates history of customer purchases
Data Update					
	inventory update	PUT	/user/inventory/update	UpdateInventory { quantity item puchase integer }	Update inventory data
	inventory count	PUT	/user/inventory/count	CountInventory { quantity items remain integer }	Update Count remaining inventory data
	history log record	PUT	/user/customerprofile/history/update	UpdateCustomerHistory { success boolean message string }	Update history log
Data					
	customer profile data	GET	/user/customerprofile/purchasehistory	GetCustomerData { UserID: integer, CustomerID: integer, Name: string Email: string, Purchase History: string }	Get's the customer profile data
	relationships based on history	GET	/user/customerprofile/history/relationships	GetCustomerProfileRelationship { UserID: integer, CustomerID: integer, Employee ID: integer, Purchase history: string, Inventory: integer}	Gets the customer's relationship data
	history log record	GET	/user/customerprofile/history	GetCustomerHistory { UserID: integer, CustomerID: integer, History: string, Purchase item: string }	Gets customer Profile history
	inventory count data	GET	/user/invenotry/count	GetCountinventory { UserID: integer, History: string, Purchase item: string, inventory count: integer}	Gets inventory remaining count