Fundamentals of Computing II - LAB
CSCE 1102- Project
Milestone 1 - Design Report

Team#: 1

Team Email: quickstockcs2@gmail.com

Members' Info:

Member Name	ID	Email
Salma Gomaa	**900241457**	salmagomaa@aucegypt.edu
Omar mubarak	**900243846**	Mubarak@aucegypt.edu
Nour elmasry	**900223003**	nourelmasry04@aucegpt.edu
Ali gamal	**900222145**	aligamal04@aucegypt.edu

GitHub link: https://github.com/aligamaal/Group-Project.git

Core classes

Class Name /Short description	Data members	Member Functions

Item	String itemID, string name, string description, int quantity, double price, bool isActive	Public:
		
		// Pure virtual functions
		virtual void display() const = 0;
		virtual double calculateTotalValue() const = 0;
		// common functions
		void updateQuantity(int newQuantity);
		void addStock(int amount);
		void removeStock(int amount);

		bool isInStock() const;
		bool isLowStock(int threshold) const;
		// getters
		string getItemID() const;
		string getName() const;
		int getQuantity() const;
		double getPrice() const;
		// setters
		void setName(const string& name);
		void setPrice(double price);

Product : Item	**String category, string brand, double weight, string dimensions, string barcode**	**public:**
		void display() const override;
		double calculateTotalValue() const override;
		void setCategory(const string& category);
		void setBarcode(const string& barcode);
		string getCategory() const;
		string getBarcode() const;

Supply : Item	**String supplierName, string supplierContact, int reorderLevel, string lastOrderDate, bool isConsumable**	**public:**
		void display() const override;
		double calculateTotalValue() const override;
		void setSupplier(const string& supplier);
		string getSupplier() const;
		bool needsReorder() const;

Users classes and functionality of each user

Class Name /Short description	Data members	Member Functions

User	String userId, string username, string password, string email, string role, string firstName, string lastName, string dateCreated, bool isActive	public:
		User(string id, string user, string pass, string mail, string r);
		// Authentication
		bool login(string user, string pass);
		void logout();
		
		// Getters
		string getUserId() const;
		string getUsername() const;
		string getEmail() const;
		string getRole() const;
		string getFirstName() const;
		string getLastName() const;
		string getDateCreated() const;
		bool getIsActive() const;
		
		// Setters
		void setUsername(string user);
		void setPassword(string pass);
		void setEmail(string mail);
		void setFirstName(string fname);
		void setLastName(string lname);
		void setIsActive(bool active);
		
		// Utility functions
		void displayInfo() const;
		void changePassword(string newPass);
		
		// Pure virtual function
		virtual void displayMenu() const = 0;

Admin : User	String adminLevel, int maxUsers	Public:
		// User management
		void createUser(const User& user);

	Vector <String> systemPermissions	**void deleteUser(const string& userID);
		void modifyUserPermissions(const string& userID);
		void viewAllUsers() const;
		
		// Item management
		void addItem(const Item& item);
		void removeItem(const std::string& itemID);
		void modifyItem(const std::string& itemID);
		
		// System management
		void generateFullReport() const;
		void backupSystem();
		void restoreSystem();
		void viewSystemLogs() const;
		void configureSystem();**

Manager : User	**String department,
	double budgetLimit,
	string supervisorID,
	Int approvalLevel**
		**public:
		// Item management
		void addItem(const Item& item);
		void updateItem(const string& itemID);
		void viewAllItems() const;
		
		// Stock management
		void updateStock(const string& itemID, int quantity);
		void checkLowStock() const;
		void approveReorder(const string& itemID);
		
		// Reporting
		void generateInventoryReport() const;
		void generateStockReport() const;
		void generateValueReport() const;
		void viewItemHistory(const string& itemID) const;**

Employee : User	**String employeeID, string shift, string**	**// Basic item operations void searchItem(const string& keyword) const;**

	workstation, string managerID, vector <string> assignedTasks	void viewItem(const string& itemID) const;
		void viewAllItems() const;
		
		// Stock updates
		void updateQuantity(const string& itemID, int newQuantity);
		void recordStockMovement(const string& itemID, int amount, const string& type);
		
		// Basic reporting
		void generateSimpleReport() const;
		void checkItemAvailability(const string& itemID) const;
		void viewLowStockItems() const;

Maestro class(es)

List Name	Selected DS	Reasons for selecting this DS
InventorySystem	Stack	Inventory system is managed using a LIFO system (last in first out) which is represented by the stack.