Software Design and Analysis

Project Phase-2

Topic: Inventory Management System(IMS)

Group Members:

Bisma Amir	23L-0659
Aleena Qayyum	23L-0981
Aleena Orazi	23L-0820
Mehrmah Khan	23L-0623

Mehrmah’s part:

Identifier	UC-01
Name	Login
Description	This use case allows a user (Admin, Manager, or Staff) to securely log into the system with role-based access.
Priority	High
Actors	Admin, Manager, Staff
Pre-condition(s)	User account must exist in the system.
Post-condition(s)	User is logged in with correct role and activity logged.Triggers Activity logging for audit purposes.

Typical Course of Action

S#	**Actor Action**	**System Response**
1	Actor enters username and password.	System verifies that fields are not empty.
2	Actor presses 'Login'.	System validates credentials against user database.
3		System checks user role and permissions.
4		System creates a session, logs activity, and redirects to user-specific dashboard.

Alternate Course(s) of Action

S#	**Actor Action**	**System Response**
1a	Actor leaves username or password empty.	System displays 'Fields cannot be empty'.
2a	Actor enters wrong credentials.	System displays 'Invalid username or password'.

2b	Account is locked due to multiple failed attempts.	System denies access and displays 'Account locked, contact admin'.
3a	Database error while validating credentials.	System displays 'Login service unavailable'.
4a	User presses cancel.	System clears fields and returns to login page.

Identifier	UC-02	
Name	Add Product	
Description	This use case allows the Staff/Manager to add a new product into the inventory with all details validated before storage.	
Priority	High	
Actors	Staff, Manager	
Pre-condition(s)	User must be authenticated and logged in.	
Post-condition(s)	Product is added into the inventory with a unique Product ID.	
	Typical Course of Action	
S#	**Actor Action**	**System Response**
1	Actor selects 'Add Product' option from the main menu.	System displays the Add Product form with empty fields.
2	Actor enters product name.	System verifies that the name field is not empty and does not contain invalid characters.
3	Actor enters product category.	System checks if the category exists in the database; otherwise prompts for a valid category.
4	Actor enters quantity and price.	System validates that quantity and price are numeric and within acceptable limits.

5	Actor enters supplier details.	System checks if supplier exists; if not, prompts for valid supplier or new supplier registration.
6	Actor presses 'Save'.	System validates all fields, generates a unique Product ID, saves product details to the database, and displays confirmation message.
Alternate Course(s) of Action		
S#	**Actor Action**	**System Response**
2a	Actor leaves product name empty.	System displays 'Product name is required'.
3a	Actor enters invalid category.	System displays 'Invalid category'.
4a	Actor enters negative or zero quantity or invalid price.	System displays 'Enter valid values'.
5a	Actor enters supplier not in system.	System displays 'Invalid supplier'.
6a	Actor presses cancel.	System discards input and returns to home page.
6b	Database error occurs while saving.	System displays 'Unable to save product. Please try again later'.

Identifier	UC-03
Name	Update Product Details
Description	This use case allows the Staff/Manager to search and update details of an existing product.
Priority	High
Actors	Staff, Manager
Pre-condition(s)	Product must exist in the system.

Post-condition(s)	Product details are updated successfully in the inventory.		
Typical Course of Action			
S#	**Actor Action**	**System Response**	
1	Actor searches for product using name or ID.	System retrieves and displays product details.	
2	Actor updates product name, category, quantity, price, or supplier details.	System verifies each modified field for correctness.	
3	Actor presses 'Update'.	System validates all fields, saves updated details, logs the change, and displays confirmation message.	
Alternate Course(s) of Action			
S#	**Actor Action**	**System Response**	
1a	Actor enters invalid or non-existing product ID.	System displays 'Product not found'.	
2a	Actor leaves mandatory field empty.	System displays 'Field cannot be empty'.	
2b	Actor enters invalid data format (e.g., text in price field).	System highlights the invalid field and prompts correction.	
3a	Actor cancels update.	System discards changes and returns to product details view.	
3b	Database error occurs during update.	System displays 'Unable to update product. Please try again later'.	

Identifier	UC-04
Name	Delete Product
Description	This use case allows the Manager to delete or deactivate a product from the inventory system.
Priority	Medium
Actors	Manager
Pre-condition(s)	Product must exist in the system.
Post-condition(s)	Product is removed or marked as inactive.

Typical Course of Action

S#	**Actor Action**	**System Response**
1	Manager searches for and selects product to delete.	System displays product details with delete option.
2	Manager presses 'Delete/Deactivate'.	System prompts for confirmation to avoid accidental deletion.
3	Manager confirms deletion.	System checks dependencies, deletes/deactivates product, updates inventory, and shows confirmation message.

Alternate Course(s) of Action

S#	**Actor Action**	**System Response**
1a	Product does not exist.	System displays 'Invalid product'.
2a	Manager cancels delete at confirmation prompt.	System aborts deletion and returns to product details page.
3a	Product is linked to sales/purchase records.	System displays 'Cannot delete product, it is in use'.
3b	System error during deletion.	System displays 'Deletion failed. Please try

		again later'.

Identifier	UC-05
Name	Maintain Backup
Description	This use case allows the Manager to create and store secure backups of system data.
Priority	High
Actors	Manager
Pre-condition(s)	Manager logged in with admin rights.
Post-condition(s)	Backup file is created, encrypted, and securely stored.

Alternate Course(s) of Action

S#	**Actor Action**	**System Response**
1	Manager selects 'Maintain Backup' option.	System displays available backup options.
2	Manager chooses backup location and type (full or incremental).	System checks availability of selected location and prepares backup process.
3	Manager confirms backup action.	System creates backup, encrypts the file, logs activity in audit trail, and shows success message with backup ID.

Typical Course of Action

S#	**Actor Action**	**System Response**
2a	Insufficient storage space in chosen location.	System displays 'Not enough storage available'.
2b	Manager cancels the backup.	System aborts backup process and returns

		to main menu.
3a	Backup process fails due to system error.	System displays 'Backup failed, please try again' and logs the error.
3b	Network issue occurs while storing backup remotely.	System retries storing backup and notifies user if failure persists.

Identifier	UC-06	
Name	Create Unique Product ID	
Description	This use case ensures that every new product receives a unique system-generated Product ID.	
Priority	High	
Actors	System	
Pre-condition(s)	Add Product form has been successfully submitted.	
Post-condition(s)	Each product has a unique Product ID/SKU stored in the system.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Actor submits product details.	System validates all entered data.
2		System generates a unique Product ID following predefined format (e.g., PROD-XXXX).
3		System stores product with unique ID in database.
4		System displays confirmation with Product ID to user.
Alternate Course(s) of Action		

S#	Actor Action	System Response
2a	Duplicate ID is generated due to system error.	System regenerates Product ID until unique one is assigned.
3a	Database error while storing product.	System retries or displays 'Unable to store product'.

Identifier	UC-7	
Name	Encrypt Sensitive Data	
Description	This use case ensures all sensitive data like passwords, financial details, and backup files are encrypted before storage.	
Priority	High	
Actors	System	
Pre-condition(s)	Sensitive data is being stored or transmitted.	
Post-condition(s)	Data is encrypted with secure algorithms and stored safely.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Actor triggers login, backup, or financial transaction process.	System identifies sensitive fields for encryption.
2		System applies encryption algorithm to sensitive fields.
3		Encrypted data is stored in database or backup file.
4		System logs encryption activity in audit log.

Alternate Course(s) of Action		
S#	**Actor Action**	**System Response**
2a	Encryption algorithm fails or crashes.	System retries encryption and logs error.
3a	Database/storage unavailable.	System queues encrypted data and retries when available.
4a	Audit log write fails.	System retries logging and sends alert to admin.

Identifier	UC-8
Name	Search Product
Description	This use case allows Staff/Manager to search for products by different criteria in the inventory.
Priority	High
Actors	Staff, Manager
Pre-condition(s)	Products must exist in the system.
Post-condition(s)	Matching products are displayed with details.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Actor enters product name/ID/category in search bar.	System validates input for correct format.
2	Actor presses 'Search'.	System queries database for matching results.

3		System displays results in list/table format with product details.
4		Actor may refine search by applying filters (price, category, supplier).
5		System refreshes results based on applied filters.
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1a	Actor leaves search field empty.	System displays 'Please enter search criteria'.
2a	No match found in database.	System displays 'No products found'.
2b	Actor enters invalid characters.	System displays 'Invalid input'.
3a	System error while fetching results.	System displays 'Search service unavailable'.

Bisma UC’s:

Identifier	UC-9
Name	Record Purchase Order
Description	This use case allows the Manager to create and record a purchase order with a supplier.
Priority	High
Actors	Manager
Pre-condition(s)	Manager is authenticated and logged in.
Post-condition(s)	Purchase order is saved in the system. Update Inventory/Stock Levels use case is triggered to update stock quantities. Triggers Activity logging for audit purposes.

Typical Course of Action

S#	**Actor Action**	**System Response**
1	Manager shall select the “Record Purchase Order” button.	Manager shall be directed to purchase order entry form.
2	The manager shall enter the Product of the purchase order.	
3	The manager shall enter the quantity of products in purchase order.	
4	The manager shall enter the total price of the purchase order.	
5	The manager shall enter the supplier of the purchase order.	
6	The manager shall enter the date of receiving the purchase order.	
7	The manager shall press the submit button.	The system saves purchase order. System invokes **Update Inventory** use case to increase stock for ordered products. System invokes log user activity case.

		System generates a unique Purchase Order ID and displays confirmation to Manager.
Alternate Course(s) of Action		
2a	The manager may enter a product that does not exist in the inventory.	System shall display message “Please enter a valid product for purchase order”.
2b	The manager may forget to enter Product.	System will display “Product Field empty”.
3a	The manager may enter an invalid/negative quantity.	System will display “Enter a valid quantity”.
3b	The manager may forget to enter Quantity.	System will display “Quantity Field empty”.
4a	The manager may enter an invalid/ negative price.	System will display “Enter a valid price”.
4b	The manager may forget to enter Price.	System will display “Price Field empty”.
5a	the manager may enter a supplier that does not exist in the system .	System will display “Enter a valid supplier”.
5b	The manager may forget to enter the supplier.	System will display “Supplier field empty”.
6a	The manager may enter invalid date	System will display “Enter a valid date”.
7a	The manager may press the cancel button.	System discards the order data and redirects to home page.

Identifier	UC-10	
Name	Supplier Return	
Description	This use case allows the Manager to record the return of defective or excess products back to a supplier.	
Priority	Medium	
Actors	Manager	
Pre-condition(s)	Manager is authenticated and logged in. Relevant supplier and product details exist in the system. Returned products are already in stock.	
Post-condition(s)	Supplier return record is saved in the system. Update Inventory/Stock Levels is invoked to reduce stock. Return transaction is logged for audit and reporting.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Manager shall select Supplier Return option.	Manager shall be directed to supplier return form.
2	Manager shall enter the product to be returned.	
3	Manager shall enter the supplier to whom product is to be returned.	
4	Manager shall enter the quantity of return.	
5	Manager shall enter the reason for return.	
6	Manager shall enter the date for return	
7	Manager shall press the submit button	System shall save the supplier return record. System shall invoke Update Inventory to decrease stock for returned products.

		System shall invoke log user activity. System shall generate a unique Return ID and display confirmation to the Manager.
Alternate Course(s) of Action		
2a	The manager may enter a product that does not exist in the inventory.	System shall display message “Please enter a valid product for purchase order”.
2b	The manager may forget to enter Product.	System will display “Product Field empty”.
3a	The manager may enter a supplier that does not exist in the system .	System will display “Enter a valid supplier”.
3b	The manager may forget to enter the supplier.	System will display “Supplier field empty”.
4a	The manager may enter an invalid/negative quantity.	System will display “Enter a valid quantity”.
4b	The manager may forget to enter Quantity.	System will display “Quantity Field empty”.
5a	User may forget to enter the reason for return.	System will display “Reason Field empty”.
6a	The manager may enter invalid date	System will display “Enter a valid date”.
7a	The manager may press the cancel button.	System discards the return data and redirects to home page.

Identifier	UC-11
Name	Adjust Stock Levels
Description	This use case allows the Manager to manually adjust stock levels in the system to correct discrepancies caused by stock audits, damaged items, or data entry errors.
Priority	Medium
Actors	Manager
Pre-condition(s)	Manager is authenticated and logged in. Product exists in the system.
Post-condition(s)	Stock levels are updated based on adjustment. Adjustment is logged in audit records.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Manager shall select the Adjust Stock levels option.	Manager shall be directed to Adjust stock levels form.
2	Manager shall enter the product whose stock level is to be changed.	
3	Manager shall select the type of adjustment (increase/decrease).	
4	Manager shall enter the quantity of update.	
5	Manager shall enter the reason for update.	
6	Manager shall press the submit button	System shall invoke Update Inventory to increase/decrease stock for the product. System shall invoke log user activity. System shall display confirmation to the Manager.
Alternate Course(s) of Action		
2a	The manager may enter a product that does not exist in the inventory.	System shall display message “Please enter a valid product for purchase order”.

2b	The manager may forget to enter Product.	System will display “Product Field empty”.
4a	The manager may enter an invalid quantity.	System will display “Enter a valid quantity”.
4b	The manager may forget to enter Quantity.	System will display “Quantity Field empty”.
5a	User may forget to enter the reason for update.	System will display “Reason Field empty”.
6a	The manager may press the cancel button.	System discards the update stocks data and redirects to home page.

Identifier	UC-12	
Name	Record Sales Order	
Description	This use case describes the process to record a sales order from a customer.	
Priority	High	
Actors	Staff	
Pre-condition(s)	Staff is authenticated and logged in. Relevant product(s) exist in the system. Sufficient stock is available for the requested products.	
Post-condition(s)	Sales order is saved in the system. Update Inventory/Stock Levels use case is triggered to decrease stock for sold items. Triggers sales transaction Activity logging for audit purposes.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Staff shall select the “Record Sales Order” button.	Staff shall be directed to sales order entry form.
2	Staff shall enter the Products of the sales order.	
3	Staff shall enter the quantities of product(s) in sales order.	Check the stock level of the product.

4	Staff shall enter the total price of the sales order.	
5	Staff shall enter the customer of the purchase order. If a customer is a new customer display form to add a new customer. If an existing customer just adds to the record.	
6	Staff shall enter the date of the sales order.	
7	Staff shall press the submit button.	The system saves sales order. System invokes Update Inventory use case to decrease stock for ordered products. System invokes log user activity. System generates a unique Sales Order ID and displays confirmation to staff.
Alternate Course(s) of Action		
2a	Staff may enter a product that does not exist in the inventory.	System shall display message “Please enter a valid product for purchase order”.
2b	Staff may forget to enter Product.	System will display “Product Field empty”.
3a	Staff may enter quantities greater than the available stock.	System will display “Insufficient stock available for the selected product(s).”.
3b	Staff may forget to enter Quantity.	System will display “Quantity Field empty”.
4a	Staff may enter an invalid/ negative price.	System will display “Enter a valid price”.
4b	Staff may forget to enter Price.	System will display “Price Field empty”.
5a	Staff may forget to enter the customer.	System will display “Customer field empty”.
6a	Staff may enter invalid date	System will display “Enter a valid date”.
7a	Staff may press the cancel button.	System discards the order data and redirects to home page.

Identifier	UC-13	
Name	Customer Return	
Description	This use case describes the process to record the return of sold products by a customer due to defects, damages, or other reasons.	
Priority	Medium	
Actors	Staff	
Pre-condition(s)	Staff is authenticated and logged in. Customer and related sales order exist in the system. Returned products were previously sold.	
Post-condition(s)	Customer return is recorded successfully. Update Inventory/Stock Levels use case is triggered to increase stock quantities. Return transaction is logged in the audit record.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Staff shall select the “Customer Return” option.	Staff shall be directed to the Customer Return form.
2	Staff shall enter customer and sales order details.	System retrieves matching sales record and verifies eligibility for return.
3	Staff shall select products to return	
4	Staff shall enter the quantity of return.	System validates that returned quantity does not exceed originally sold quantity.
5	Staff shall enter the reason for return.	
6	Staff shall enter the date of return.	
7	Staff shall press the submit button.	The system saves customer return.

		System invokes Update Inventory use case to increase stock for returned products. System invokes log user activity. System generates a unique Return ID and displays confirmation message.
Alternate Course(s) of Action		
2a	Staff may enter a sales order or customer record that does not exist	System shall display message “No matching sales record found for this customer or order”.
2b	Staff may forget to enter sales order or customer record.	System will display “Sales order or customer details Field empty”.
3a	Staff may enter product that does not exist in relevant sales order	System will display “Return product does not exist in the specified sales order”.
3b	Staff may forget to enter Product(s).	System will display “Product(s) Field empty”.
4a	Staff may enter quantity that exceeds quantity sold.	System will display “Return quantity cannot exceed originally sold quantity.”
4b	Staff may forget to enter Quantity.	System will display “Quantity Field empty”.
5a	Staff may forget to enter the reason for return.	System will display “Reason field empty”.
6a	Staff may enter an invalid date	System will display “Enter a valid date”.
7a	Staff may press the cancel button.	System discards the return data and redirects to home page.

Identifier	UC-14
Name	Update inventory/Stock Levels
Description	This use case updates the stock quantities of products in the system whenever inventory-affecting transactions occur, such as purchases, sales, returns, or manual adjustments. The system modifies stock records accordingly.
Priority	High
Actors	System (triggered automatically from other use cases)
Pre-condition(s)	Transaction details (purchase, sale, return, or adjustment) are available. Product(s) involved in the transaction exist in the database.
Post-condition(s)	Stock levels are updated accurately in the inventory. Stock change is logged with product, quantity, details, and timestamp. Low-stock notifications are generated if applicable.

Typical Course of Action

S#	Actor Action	System Response
1		System shall receive a trigger from another use case (e.g., Purchase, Sale, Return, Adjustment).
2		System shall retrieve the affected product and its current stock level.
3		System shall calculate new stock quantity based on transaction type (increase or decrease).
4		System shall update stock levels in the inventory database.
5		System shall trigger log inventory changes to maintain the record of change for audit. System shall generate low stock notification if stock falls below the minimum threshold.

Alternate Course(s) of Action

2a		If system could not find the product and its stock level in inventory system shall display message “No matching product or stock record found to update inventory”.
4a		If the inventory update fails, system rolls back the transaction and displays database error.

Identifier	UC-15	
Name	Log Inventory Changes	
Description	This use case records every inventory modification in an audit log which ensures accountability, traceability, and data integrity for all inventory operations.	
Priority	Medium	
Actors	System(triggered automatically by Update Inventory/Stock Levels)	
Pre-condition(s)	A stock change has occurred (addition, deduction, or adjustment).	
Post-condition(s)	A log entry is created in the inventory audit log.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
3		System shall detects an update to stock/inventory levels.
4		System creates a log entry containing Product ID, old quantity, new quantity, update reason, timestamp, and user ID.
5		System shall add the log entry in audit log record.
Alternate Course(s) of Action		
5a		If audit log entry fails system displays “Failed to record change in audit log”.

Identifier	UC-16	
Name	Log User Activity	
Description	This use case ensures that all critical user actions (login, add product, update product, delete, sales, purchase, returns, etc.) are logged for accountability and monitoring.	
Priority	Medium	
Actors	System (triggered by user actions),Triggered by Manager and staff	
Pre-condition(s)	User must be logged into the system.	
Post-condition(s)	User activity record is created and added to log.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
3		System shall detects a user action that triggers log user activities.
4		System generates a log entry with User ID, action performed, timestamp.
5		System stores log entry in the user activity log.
Alternate Course(s) of Action		
5a		If the audit log entry fails system displays “Failed to record user action in audit log”.

Aleena Qayyum part
USE CASES:
View User Activity Log, Manage backup data, recover data, view backup data, view inventory audit log, Batch/lot tracking, warehouse stock tracking

Identifier	UC-17	
Name	Manage Backup Data	
Description	Responsible for the backup of all data	
Priority	High	
Actors	Admin	
Pre-condition(s)	Admin should be logged-in to the system.	
Post-condition(s)	Backup of data is successfully created.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Admin selects Backup Data option from menu.	System will show a page for backup with options differential or full backup.
2	Admin selects Full Backup.	System starts creating a full backup of all data.
		System confirms: “Full backup completed successfully.” and logs the event.
2a	Admin selects Differential Backup.	System creates a backup of changes since the last full backup.
		System confirms: “Differential backup completed successfully.” and logs the event.
Alternate Course(s) of Action		
	Actor Action	**System Response**
2a		If insufficient storage space, system shows: *“Not enough storage.*
		If backup is interrupted (power/network failure), system attempts auto-retry. If retry fails, system notifies admin: *“Backup incomplete. Please try again.”*

		If backup file is corrupted during creation, system shows: *"Backup failed due to file error."* and logs the failure.

Identifier	UC-18	
Name	View Backup Data	
Description	Shows the backup data	
Priority	Low	
Actors	Admin	
Pre-condition(s)	Admin should be logged-in to the system.	
Post-condition(s)	Admin has viewed the backup data.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Admin selects View Backup Data option from menu.	System will show a page with backup data.
Alternate Course(s) of Action		
	Actor Action	**System Response**
1a		There is no data that has been backed up by Admin so the system will show error "No data is backed up."

Identifier	UC-19
Name	Recover Data
Description	After data loss, Admin can restore data from local server.
Priority	Low
Actors	Admin
Pre-condition(s)	Admin should be logged-in to the system and atleast one backup exists.
Post-condition(s)	Data has been successfully restored into system.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Admin navigates to Recover Data.	System displays list of available backup files.
2	Admin selects a backup file to restore.	System asks for confirmation “Are you sure”? Yes/No
3a	Admin selects “Yes”	System begins recovery process, .
		System shows message “Data recovered successfully.”
		System log the event in audit log.
Alternate Course(s) of Action		
2a	**Actor Action**	**System Response**
3a	Admin selects “No”	System displays list of available backup files.
		If Backup file is corrupted. Error message “Backup file is corrupted" displays.

Identifier	UC-20	
Name	View Inventory Audit Log	
Description	This use case allows authorized users to view an audit log of inventory operations.	
Priority	Low	
Actors	Admin and Manager	
Pre-condition(s)	User is logged into the system with permission to view logs.	
Post-condition(s)	Audit log entries are displayed.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User navigates to Audit Log section.	System displays page of audit Log.
2	User selects view audit log option.	System displays the list of audit log entries with details.

Alternate Course(s) of Action		
	Actor Action	**System Response**
2a	System fails to retrieve data	System shows error “Unable to load audit logs at this time.”

Identifier	UC-21
Name	Batch/Lot Tracking
Description	This use case allows users to track products by their batch or lot number.
Priority	Medium
Actors	Admin
Pre-condition(s)	User is logged in and batch information exists in the system
Post-condition(s)	Batch details are displayed or updated and this activity is logged in the system.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User navigates to Batch Tracking page.	System displays search options (batch number, product ID, date, supplier, expiry).
2	User enters batch details	System displays matching batch records.
3	User selects a batch from the list.	System shows detailed batch info.
4	User updates something.	System updates the information and keep the record in audit log.
Alternate Course(s) of Action		
	Actor Action	**System Response**
2a	User enters invalid or non-existing batch number.	System shows message: “No batch records found.”
3a	System fails to retrieve batch data (database issue).	System displays:“Unable to load batch information”

Identifier		UC-22	
Name		View User Activity Log	
Description		This use case enables Admins or Managers to view logs of user activities within the system.	
Priority		Medium	
Actors		Admin	
Pre-condition(s)		Admin is logged into system.	
Post-condition(s)		System displays the requested activity logs.	
Typical Course of Action			
S#	**Actor Action**		**System Response**
1	User navigates to Activity Log *page*		System displays filter options (date range, user ID, activity type).
2	User enters filter criteria		System retrieves logs based on criteria.
3	User selects a log entry from the list.		System displays details (user, time, activity, affected data).
Alternate Course(s) of Action			
	Actor Action		**System Response**
2a	No log entries match the search.		System shows: “No activity records found.”

Identifier	UC-23
Name	Warehouse stock track
Description	This allows Admins to check available stock in different warehouses.The system displays product details, quantities and batch information for each warehouse.
Priority	Medium
Actors	Admin
Pre-condition(s)	Admin is logged into system.
Post-condition(s)	System displays stock details for the selected warehouse.
Typical Course of Action	

S#	Actor Action	System Response
1	User navigates to *Warehouse Management → Stock Tracking*.	System shows a list of available warehouses.
2	User selects a specific warehouse.	System displays all stock items in that warehouse (item name, quantity, batch ID, expiry if applicable).
3	User applies filters(by product category, batch lot)	System updates the stock list accordingly.
4	User views details of a stock item.	System shows item details (supplier, stock received date, current quantity).
Alternate Course(s) of Action		
	Actor Action	**System Response**
2a	User selects a warehouse with no stock.	System displays:“No stock available in this warehouse.”
2b	System fails to load stock data due to error.	System displays: *“*Unable to fetch stock details. Please try again later.*”*

Identifier	UC-24	
Name	Add Categories	
Description	Add new product categories to the system	
Priority	High	
Actors	Admin	
Pre-condition(s)	The user should be logged-in to the system.	
Post-condition(s)	New category will be added to the system	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User shall click "Add Category" button from categories management page	System displays "Add New Category" form
2	User shall enter unique category code (CAT_ID)	System validates category code format and displays “Category found”
3	User shall enter category name	
4	User shall enter category description	
5	User shall sets category status (Active/Inactive)	
6	User shall click "Save Category" button	System saves category and displays "Category added successfully" message
Alternate Course(s) of Action		
2a	User may enter duplicate category code	System will display “Category code already exists. Please use a unique code”
2b	User may leave category id empty	System will display “Category id is required”
3a	User may enter duplicate category name	System will display “Category name already exists”
3b	User may leave category name empty	System will display “Category name is required”
6a	User may clicks "Cancel".	System returns to categories management page without saving
6b	Database connection may fail during save	System will display “Unable to save category. Please try again”.

Identifier	UC-25	
Name	View Stock Availability Report	
Description	Generate current stock levels and availability status report	
Priority	High	
Actors	Admin	
Pre-condition(s)	The user should be logged-in to the system.	
Post-condition(s)	Stock availability report generated and displayed; user can export report	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User shall click "Stock Availability Report" from reports menu	System displays stock report page
2	User shall select Category Type from dropdown	
3	User shall click "Generate Stock Report"	System generates report with unique Report ID and displays stock availability data
Alternate Course(s) of Action		
2a	User may leave Category Type unselected	System will display “ Please select a Category Type”
2b	User may not select any category	System will display “Category not selected"

Identifier	UC-26
Name	View Sales vs Stock Report
Description	Generate comparative analysis of sales performance against stock levels
Priority	Medium
Actors	Admin
Pre-condition(s)	The user should be logged-in to the system.
Post-condition(s)	Comparative report generated with unique Report ID

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User shall click "Sales vs Stock Report" option	System displays stock report page
2	User shall select Category Type from dropdown	
3	User shall click "Generate Comparison Report"	System generates report with unique Report ID and displays comparison report
Alternate Course(s) of Action		
2a	User may not select any category	System will display “Category not selected"

Identifier	UC-27	
Name	View Slow & Fast Moving Items Report	
Description	Generate comparative analysis of sales performance against stock levels	
Priority	Medium	
Actors	Admin	
Pre-condition(s)	The user should be logged-in to the system.	
Post-condition(s)	Comparative report generated with unique Report ID	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User shall click "Sales vs Stock Report" option	System displays stock report page
2	User shall select Category Type from dropdown	
3	User shall click "Generate Comparison Report"	System generates report with unique Report ID and displays comparison report
Alternate Course(s) of Action		
2a	User may not select any category	System will display “Category not selected"

Identifier	UC-28	
Name	View Supplier Performance Report	
Description	Generate comparative analysis of sales performance against stock levels	
Priority	Medium	
Actors	Admin	
Pre-condition(s)	The user should be logged-in to the system.	
Post-condition(s)	Supplier performance report generated with a unique Report ID.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User shall click "Supplier Performance Report" option	System displays supplier performance report page
2	User shall select supplier name	
3	User shall select supplier name or date range from dropdown	System generates report with unique Report ID and displays supplier performance details
Alternate Course(s) of Action		
2a	User may not select any supplier	System will display “Selection not made”

Identifier	UC-29	
Name	View User Activities Log	
Description	Allows the admin to view a detailed record of all user activities performed within the system.	
Priority	Medium	
Actors	Admin	
Pre-condition(s)	The user should be logged-in to the system.	
Post-condition(s)	User activity log is displayed with date, time, and action details.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User shall click "View User Activities Log" option	System displays the activity log page
2	User shall select a specific user or date range from dropdown	
3	User shall click "View Log"	System displays user activity log with all recorded actions
Alternate Course(s) of Action		
2a	User may not select any user	System displays “Selection not made” message

Identifier	UC-30
Name	Generate Stock Availability Report
Description	Generate detailed report showing current stock levels, availability status
Priority	Medium
Actors	Admin
Pre-condition(s)	The user should be logged-in to the system.

Post-condition(s)	Stock availability report data generated and ready for display.		
Typical Course of Action			
S#	**Actor Action**	**System Response**	
1		System retrieves all product inventory records from database	
2		System calculates available quantities, reorder levels, and stock differences	
3		System generates the Stock Availability Report data with complete inventory analysis	
		System sends the report data to the "View Stock Availability Report" use case for display to user	
Alternate Course(s) of Action			
1a		If no inventory records found System returns empty dataset with "No inventory data available" status	

Identifier	UC-31	
Name	Generate Sales vs Stock Report	
Description	Generate comparative analysis report between sales performance and inventory stock levels	
Priority	Medium	
Actors	Admin	
Pre-condition(s)	The user should be logged-in to the system.	
Post-condition(s)	Sales vs stock comparative data generated	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1		System retrieves sales transaction data

2		System retrieves inventory stock level data
3		System correlates sales data with stock levels and calculates performance metrics
4		System generates comparative analysis data for sales vs stock report
5		System sends the analyzed data to display module for user presentation
Alternate Course(s) of Action		
1a		If No sales records foundSystem returns empty dataset with "No sales data available" status
2a		If No inventory data available System returns empty dataset with "No inventory data available" status

Identifier	UC-32	
Name	Generate Supplier Performance Report	
Description	Evaluate and score supplier performance based on delivery, quality, and service metrics	
Priority	Medium	
Actors	Admin	
Pre-condition(s)	The user should be logged-in to the system,supplier transaction and performance data must exist	
Post-condition(s)	Supplier performance evaluation completed suppliers scored and ranked based on performance metrics	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1		System retrieves supplier transaction history and performance records
2		System calculates delivery timeliness, product quality, and service metrics

3		System generates performance scores and rankings for each supplier
4		System sends performance evaluation data to display module for user access

Identifier	UC-33	
Name	Generate Report for Slow & Fast Moving Items	
Description	Analyze and categorize inventory items based on their movement velocity and turnover rates	
Priority	Medium	
Actors	Admin	
Pre-condition(s)	The user should be logged-in to the system,supplier transaction and performance data must exist	
Post-condition(s)	Items categorized into slow-moving and fast-moving groups; velocity analysis completed with turnover metrics	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1		System retrieves item movement history and sales data for analysis period
2		System categorizes items into slow-moving, medium-moving, and fast-moving based on thresholds
3		System generates movement velocity analysis report data
4		System sends categorized item data to display module for user review