SDLC Case Study Worksheet - Alpha IMS

Project Title: Alpha IMS
Team Name: Team Alpha
Team Members and Roles: Shaheer Ahmed (Developer, PM), Hasan (Marketing Expert), Hunain (Business Analyst), Mauzam and Sarim (Supporter), Hamza Aijaz (Docs Helper)

---

1. Requirements Phase

Functional Requirements:

1. **User Authentication and Authorization System**: Secure login with role-based access for administrators and employees.
2. **Product Inventory Management**: Manage product details including adding, updating, deleting, and viewing inventory.
3. **Sales Transaction Processing**: Process sales, update inventory, and maintain transaction history.
4. **Real-time Dashboard and Analytics**: Display key performance indicators, sales trends, and alerts in real-time.
5. **Automated Inventory Alerts and Reporting**: Generate low stock alerts and comprehensive sales/inventory reports.

Non-Functional Requirements:

1. **Performance and Scalability**: Ensure fast response times and ability to scale with increased user load and data.
2. **Security and Data Protection**: Implement robust security measures for data transmission, authentication, and access control.

---

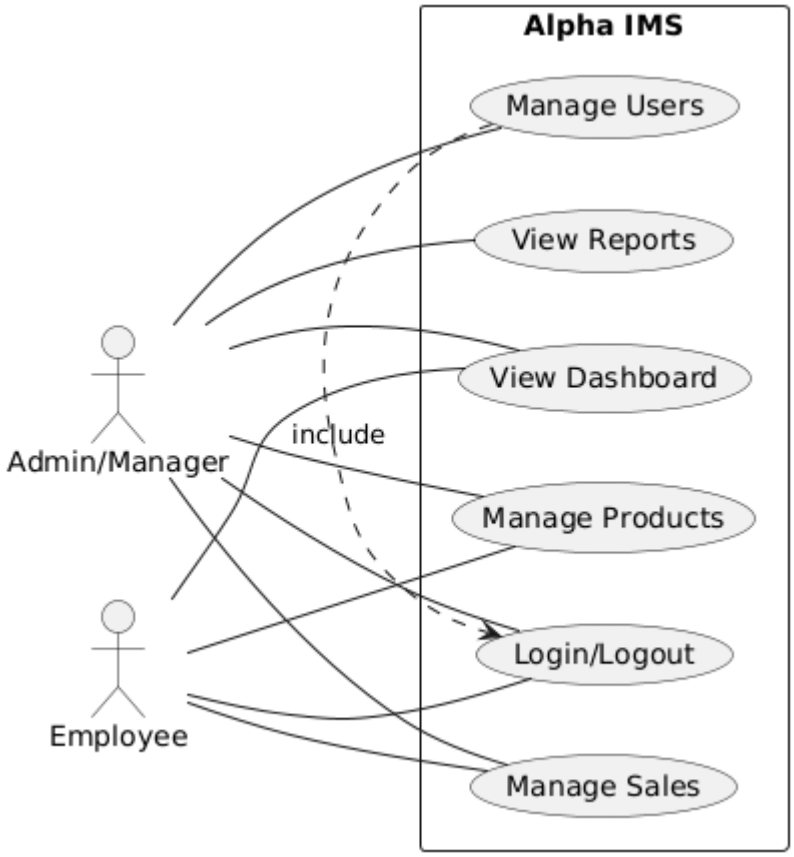
2. Design Phase

Work Breakdown Structure (WBS)

The Alpha IMS project is structured into seven main work packages: Project Management, Requirements Analysis, Design, Development, Testing, Deployment, and Maintenance. Each package is further broken down into detailed tasks across three levels to ensure comprehensive project execution and control.

UML Use Case Diagram

This diagram illustrates the primary interactions between system users (Admin/Manager and Employee) and the Alpha IMS functionalities, such as managing products, processing sales, viewing dashboards, and generating reports. It highlights role-based access control within the system.



3. Backend Design

Firebase/Firestore Integration

Alpha IMS uses Google Firebase, specifically Firestore, as its NoSQL document database. This provides real-time synchronization, scalability, and robust security. The database structure includes collections for Users, Products, Sales, and Inventory Transactions.

Tool Used: Google Firebase Console with Firestore Database

Database Collections Structure

Users Collection

```
/users/{userId}
{
  email: string,
  firstName: string,
  lastName: string,
  password: string (hashed),
  role: string,
  ...
}
```

Products Collection

```
/products/{productId}
{
  name: string,
  category: string,
  price: number,
  quantity: number,
  ...
}
```

Sales Collection

```
/sales/{saleId}
{
  productId: string,
  quantity: number,
  totalAmount: number,
  saleDate: timestamp,
  ...
}
```

Inventory Transactions Collection

```
 /inventoryTransactions/{transactionId}
{
  productId: string,
  type: string,
  quantity: number,
  ...
}
```

Security Rules Implementation

Firebase Security Rules enforce data protection and access control, ensuring users can only access authorized data based on their roles.

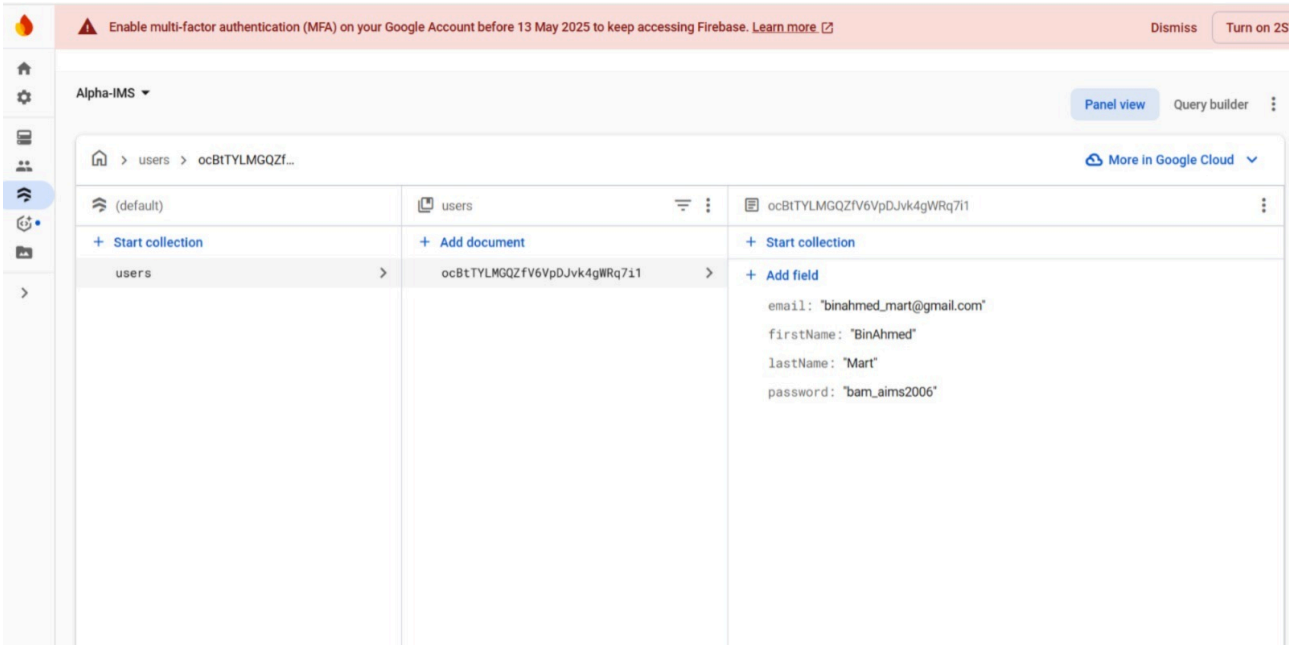
```
 rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /users/{userId} {
      allow read, write: if request.auth != null && request.auth.uid == userId;
    }
    match /products/{productId} {
      allow read: if request.auth != null;
      allow write: if request.auth != null &&

get(/databases/$`(database)/documents/users/`$(request.auth.uid)).data.role ==
"admin";
    }
    match /sales/{saleId} {
      allow read: if request.auth != null;
      allow create: if request.auth != null;
      allow update, delete: if request.auth != null &&

get(/databases/$`(database)/documents/users/`$(request.auth.uid)).data.role ==
"admin";
    }
  }
}
```

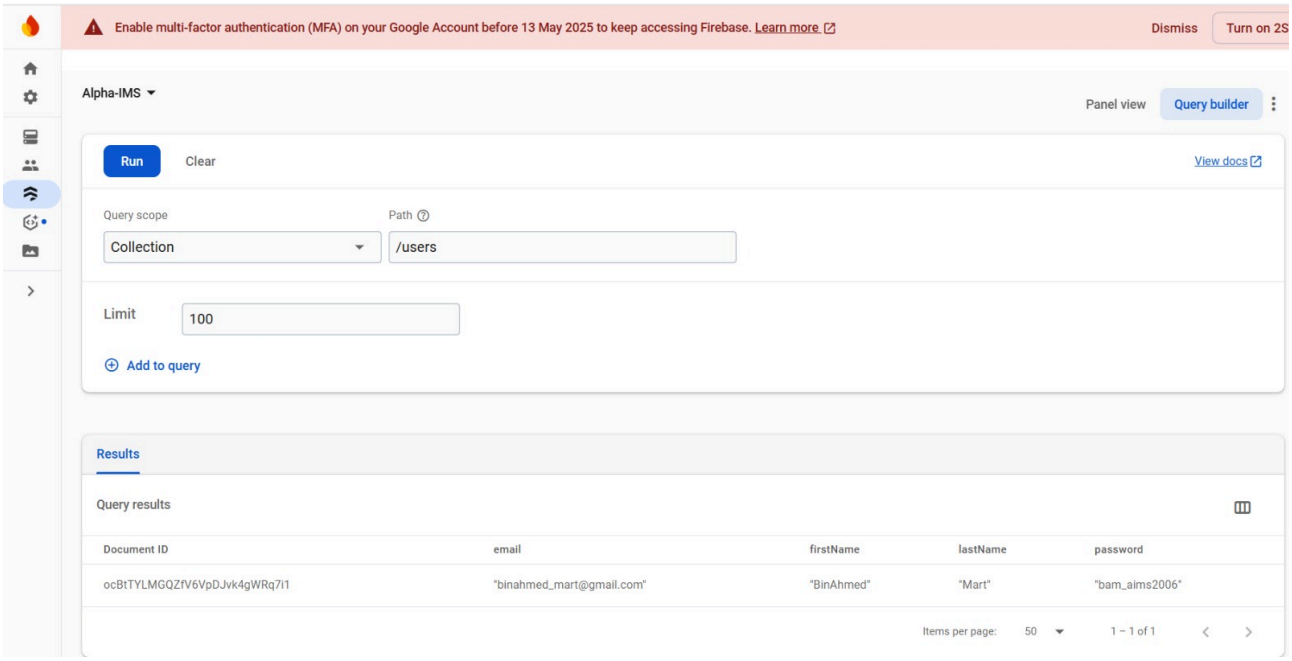
API Architecture

Cloud Functions handle server-side logic and API endpoints for authentication, product management, and sales management.

Real-time Data Synchronization

Firestore's real-time listeners provide immediate updates across all connected clients, ensuring live dashboard and inventory synchronization.


Enable multi-factor authentication (MFA) on your Google Account before 13 May 2025 to keep accessing Firebase. Learn more
Dismiss
Turn on 2S
Alpha-IMS
Panel view
Query builder
users > ocBtTYLMGQZf...
More in Google Cloud
(default)
users
ocBtTYLMGQZfV6VpDJvk4gWRq7i1
Start collection
Add document
Start collection
users
ocBtTYLMGQZfV6VpDJvk4gWRq7i1
Add field
email: "binahmed_mart@gmail.com"
firstName: "BinAhmed"
lastName: "Mart"
password: "bam_aims2006"


Enable multi-factor authentication (MFA) on your Google Account before 13 May 2025 to keep accessing Firebase. Learn more
Dismiss
Turn on 2S
Alpha-IMS
Panel view
Query builder
Run
Clear
View docs
Query scope
Collection
Path
/users
Limit
100
Add to query
Results
Query results
Document ID
email
firstName
lastName
password
ocBtTYLMGQZfV6VpDJvk4gWRq7i1
"binahmed_mart@gmail.com"
"BinAhmed"
"Mart"
"bam_aims2006"
Items per page:
50
1 – 1 of 1

4. Development Phase

Pseudocode for Key System Functions

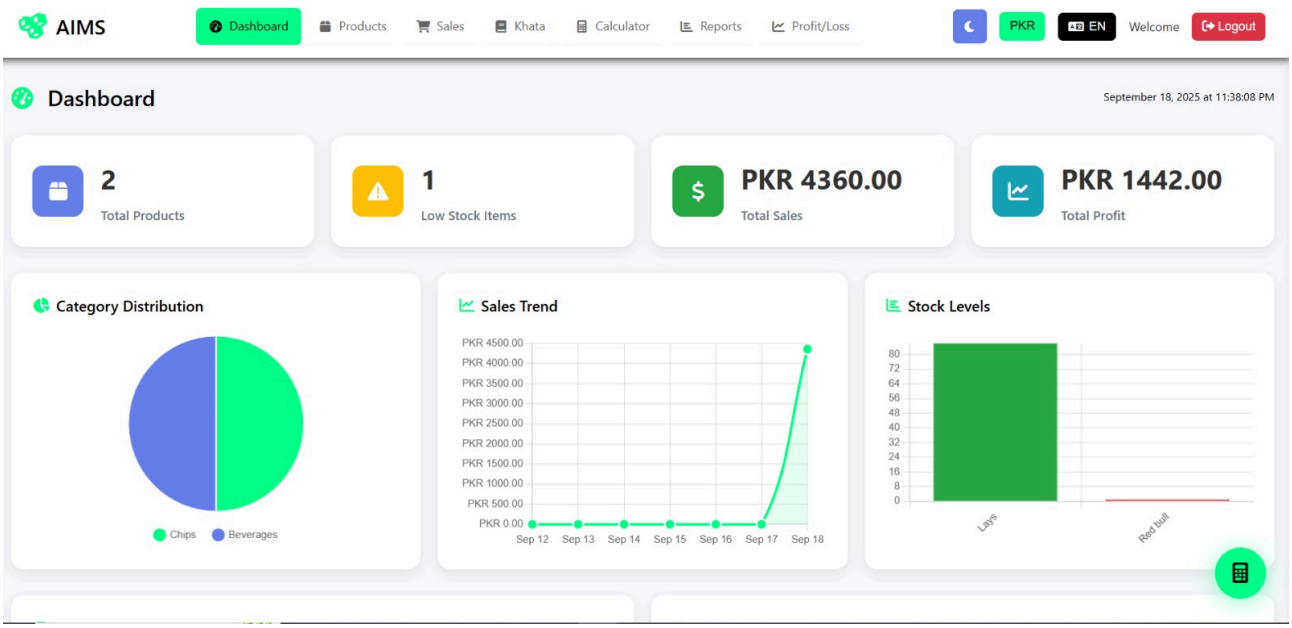
Function: Process Sale Transaction

```
 FUNCTION processSaleTransaction
INPUT(s): productId, quantity, customerId, employeeId, paymentMethod
PROCESS (steps):
    1. BEGIN TRANSACTION
    2. VALIDATE input parameters
    3. FETCH product details from products collection
    4. CHECK inventory availability
    5. CALCULATE sale amounts
    6. CREATE sale record
    7. UPDATE product inventory
    8. LOG inventory transaction
    9. CHECK for low stock alert
    10. COMMIT TRANSACTION
OUTPUT: saleId, totalAmount, profit
END FUNCTION
```

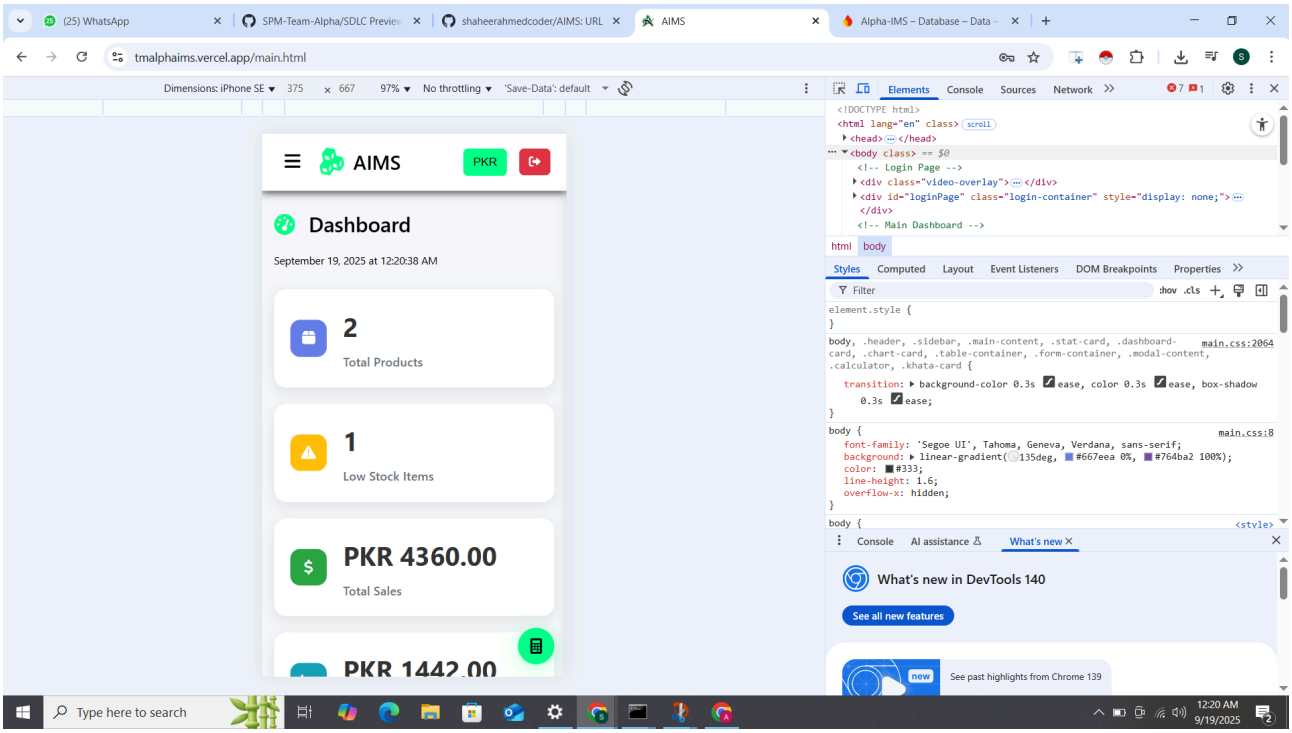
Frontend Design Implementation

The Alpha IMS frontend features a modern, responsive design optimized for both desktop and mobile devices, utilizing a professional green and white color scheme.

Desktop Interface Design: The desktop version provides a comprehensive dashboard with sidebar navigation and interactive charts for key performance indicators.



Mobile Responsive Design: The mobile interface adapts desktop functionality into a touch-friendly format with optimized navigation and card-based layouts.



Key Design Features: - Clean, minimalist interface with intuitive navigation - Responsive design supporting various devices - Real-time data updates with visual indicators

---

5. Testing Phase

Test Cases for Alpha IMS

Test Case ID	Description	Input(s)	Expected Output	Result (Pass/Fail)
TC001	User Login with Valid Credentials	Email: "admin@alphaims.com", Password: "admin123"	Successful login, redirect to dashboard, display user name and role	Pass
TC002	Process Sale Transaction with Sufficient Stock	Product ID: "PROD001", Quantity: 5	Sale recorded, inventory updated, total amount calculated.	Pass
TC003	Attempt Sale with Insufficient Stock	Product ID: "PROD002", Quantity: 100 (when only 10 available)	Error message "Insufficient stock available", no changes.	Pass

Additional Test Scenarios

Functional, non-functional, and integration testing will cover user authentication, product management, dashboard updates, report generation, security, performance, and cross-browser compatibility.

---

6. Reflection

1. Which SDLC phase was the most challenging? Why?

The Design Phase was most challenging due to balancing functional and non-functional requirements, especially in designing the Firebase/Firestore database for

real-time synchronization and creating an intuitive user interface.

2. Which SDLC model (Waterfall, Agile) best fits this project? Why?

Agile SDLC best fits Alpha IMS due to its flexibility, iterative nature, and emphasis on continuous feedback, which is crucial for adapting to evolving inventory management needs and leveraging cloud-based platforms like Firebase.

3. How did you determine functional and non-functional requirements?

Requirements were determined through stakeholder interviews, business process analysis, and industry research. Functional needs focused on core operations, while non-functional aspects addressed performance, scalability, and security based on system constraints and expectations.

---

7. Questions and Answers

Technical Questions

Q1: Why was Firebase/Firestore chosen as the backend database for Alpha IMS?

A1: Chosen for real-time synchronization, robust security rules, and scalable NoSQL document structure, aligning with dynamic inventory management needs.

Q2: How does the system handle concurrent access to inventory data?

A2: Firestore's atomic transactions and optimistic locking ensure data consistency. Real-time listeners provide immediate updates, preventing conflicts.

Q3: What security measures are implemented to protect sensitive business data?

A3: Firebase Auth for secure login, Firestore Security Rules for access control, HTTPS for data in transit, and encryption at rest protect sensitive data.

Functional Questions

Q4: How does the low stock alert system work?

A4: Monitors product quantities against predefined minimum levels. Alerts are triggered automatically when stock falls below the threshold, notifying administrators.

Q5: What reporting capabilities does Alpha IMS provide?

A5: Offers real-time sales reports, inventory status, and profit/loss analysis with visual charts, supporting data-driven decision-making.

Business Questions

Q6: How does Alpha IMS support business scalability and growth?

A6: Firebase's serverless architecture provides automatic scaling. Modular design allows feature expansion, and multi-user support accommodates growing teams.

Q7: What are the key benefits of implementing Alpha IMS for inventory management?

A7: Benefits include real-time tracking, automated alerts, comprehensive reporting, and user-friendly interfaces, leading to improved efficiency and reduced costs.

This document serves as the complete SDLC case study worksheet for the Alpha IMS project, demonstrating the application of software development lifecycle principles in creating a comprehensive inventory management solution.