FIKI Digital Market

Use-Case-Realization Specification: Manage Inventory

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 03/November/2024 | 1.0 | Final version of the document | Trần Minh Đức |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[**1. Introduction 4**](#_Toc7229)

[1.1 Purpose 4](#_Toc24338)

[1.2 Scope 4](#_Toc4932)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc30604)

[1.4 References 4](#_Toc9036)

[1.5 Overview](#_Toc17585) 4

[**2. Flow of Events—Design 5**](#_Toc10050)

[2.1 Sequence diagram 6](#_Toc19775)

[2.2 Class diagram 10](#_Toc19558)

[**3. Derived Requirements 10**](#_Toc11530)

Use-Case-Realization Specification: Inventory Management

# Introduction

## Purpose

This document describes how the use case for Managing Inventory in the FIKI mobile application is realized within the design model, specifically through sequence diagrams and class diagrams.

## Scope

This document applies to inventory management functions within the FIKI mobile application, including stock checking, inventory updates, alerts management, and reporting features.

## Definitions, Acronyms, and Abbreviations

**Seller**: User who manages inventory through the FIKI app

**HomePage**: Main interface displaying statistics and navigation options

**StockChecking**: Interface for scanning and checking product inventory

**InventoryUpdating**: Interface for adding, removing, transferring stock and processing returns

**AlertsManagement**: Interface for managing inventory-related alerts and notifications

**ReportsManagement**: Interface for generating and viewing inventory reports

**UserController**: Handle requests and data flow of users data

**OrderController**: Handle requests and data flow of users data

**ProductController**: Handle requests and data flow of users data

**AlertController**: Handle requests and data flow of alerts data

**ReportController**: Handle requests and data flow of reports data

**Users**: Stores and manage users data operations

**Orders**: Stores and manage orders data operations

**Products**: Stores and manage products data operations

**Drafts**: Stores and manage drafts data operations

**Alerts**: Stores and manage alerts data operations

**Reports**: Stores and manage reports data operations

## References

N/A.

## Overview

This document provides a comprehensive specification of the Manage Inventory use case for the FIKI mobile application. It includes sequence and class diagrams showing the implementation of various Manage Inventory functions, including stock checking, updates, alerts, and reporting capabilities.

# Flow of Events—Design

The “Manage Inventory” workflow begins when the Seller logs into the application. Upon login, the "UserController" performs user credential validation and retrieves the user's profile from the Users table as well as the "OrderController" fetches the orders stats from the Orders table. These statistics are then returned and displayed on the Homepage View for the Seller to review.

When the Seller needs to check stock levels, they press the "Stock Check" button, prompting the system to navigate to the Stock Checking Interface. In this interface, the Seller can scan product codes or manually input product information. The system performs immediate data validation on the input, after which the "ProductController" sends a request to the Products for detailed product information. Once the Products returns the product information, the system displays this detailed product data to the Seller.

For inventory updates, the process starts when the Seller presses the "Inventory Update" button. The system navigates to the Inventory Update Interface, where the Seller can select one of four options available in the quick action menu. After the Seller submits their update information, the "ProductController" initiates a transaction to save either drafts or updates. The Products commits the transaction, and the system displays a success notification to confirm the operation's completion.

The alerts management process begins when the Seller presses "Alerts Management," causing the system to navigate to the Alerts Management Interface. The "AlertController" automatically requests alerts information from the Alerts, which queries the "alerts" table and returns all active alerts. These alerts are then displayed to the Seller, who can take various actions on them. When the Seller takes action, the system submits these alerts updates to the "AlertController", which updates the alerts information in the Alerts. The Alerts returns a success or error status, and the system shows an appropriate notification to the Seller.

For report management, the workflow starts when the Seller presses "Report Updates," and the system navigates to the Updates Reporting Screen. The "ReportController" requests all available reports from the Reports, which queries the "reports" table and returns the reports data. The system then displays all accessible reports to the Seller. When the Seller presses "Export Update," the "ReportController" initiates a request to download the reports, triggering the Reports to query and return the target report data. The system then performs the download and sends a notification. Additionally, when the Seller presses "Quick Insights," the "ReportController" requests report analysis. The Reports queries the "reports" table again, returns the reports for analysis, and the system performs the analysis and presents the results to the Seller.

## Sequence diagram

A diagram of a diagram

Description automatically generated

Figure 1. Sequence diagram of HomePage interface.

A diagram of a diagram

Description automatically generated

Figure 2. Sequence diagram of StockChecking interface.

A diagram of a computer program

Description automatically generated with medium confidence

Figure 3. Sequence diagram of InventoryUpdate interface.

A diagram of a project

Description automatically generated

Figure 4. Sequence diagram of AlertsManagement interface.

A diagram of a project

Description automatically generated

Figure 5. Sequence diagram of ReportsManagement interface.

## Class diagram

A diagram of a computer code

Description automatically generated with medium confidence

Figure 6. Class diagram of the “Manage Inventory” Use-case.

# Derived Requirements

- Real-time synchronization required for stock levels across all interfaces

- Reports must be exportable in PDF, Excel, and CSV formats

- Draft saves must persist across app sessions