

**EXPONILE**

Merchant Milestone (Merchant Balance Iteration)



November 27, 2023

ACME SAICO

ARCHITECTURE AND MODELING TEAM

Contents

[Document Release History 2](#_Toc152095701)

[Reviewers 2](#_Toc152095702)

[Epics 2](#_Toc152095703)

[1. Merchant balance 3](#_Toc152095704)

[2. Merchant statement 5](#_Toc152095705)

[3. ExpoNile statement 9](#_Toc152095706)

# Document Release History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Version | Issue Date | Author | Role | Comments |
| 1 | v 1.0 | 27/11/2023 | Raneem Basem | Product Owner | Initial publication |
|  |  |  |  |  |  |

# Reviewers

|  |  |  |
| --- | --- | --- |
| # | Name | Role |
| 1 | Mohamed Ali | Business Analysis Team Leader |
|  |  |  |

# Epics

|  |  |  |
| --- | --- | --- |
| # | Epic ID | Epic |
| 1 |  | Transactions |

# Merchant balance



|  |  |
| --- | --- |
|  | **Admin views merchant balance** |
| * As a/an | * Admin |
| * I want to | * View the store’s balance |
| * So that | * Check the credit, debit, and settled accounts |
|  |
| * Scenario * Given | * Admin views the store’s balances list and can view the statement for every store * Admin see the merchant Balance page |
| * When   And | * System displays list of merchant’s balance * Store Name * Store Balance (Last value in the balance column in the merchant statement)   Display the Total Balance: ∑ Balance column |
| * Then   And | * System allows admin to: * Search by store name (auto complete / not mandatory) * Filter by (not mandatory) * Store Category * balance type * Credit (Negative value) * Debit (Positive value) * Settled (Zero Value) * All (credit, debit, and equivalent) default * Print * Export to Excel sheet * View a specific merchant statement by click “view” button * System updates the total balance based on the filters and search |

# Merchant statement



|  |  |
| --- | --- |
|  | **Admin views Merchant Statement** |
| * As a/an | * Admin |
| * I want to | * View the merchant’s statement |
| * So that | * Can follow up the merchant transactions |
|  |
| * Scenario * Given   And   * When | * Admin views a specific store’s statement, search and filter to get a short list * Admin views “merchant balance” screen   Admin selected specific store by click on “view” button to display the selected store’s statement   * System displays the selected store’s statement * Transaction Date/Time * Transaction ID * Transaction Type * P/O No. * Warehouse Method * Delivery Method * Payment Method * Total Item price * Summation of item prices per store grouped by warehouse * Promo Code * Summation of the division amount of the promo code of all items grouped by warehouse * The promo code contribution per item grouped by warehouse =   [(item price / total item prices in the store) \* promo code amount]   * Order Amount (Total item price - Promo code) * Delivery Fees * If delivery method is delivery boy, system get fess from shop setup * If delivery method is agent, system get fees through integration * In case delivery agent, system calculate the order delivery fees divided on number of stores in the same order delivery agent * Total order Amount (order amount + delivery fees) (hidden column) * Wallet * The wallet contribution per item for the given store grouped by warehouse * The wallet contribution per item =   [(item price / net item prices in PO after promo code) \* wallet amount]   * Payment (Total order amount – Wallet) * ExpoNile Share * Summation of ExpoNile share for all items for the given store grouped by warehouse * Based on payment method either payment method flagged as “COD” or no “COD” flag which mean online * If ExpoNile share is percentage * ExpoNile share = ExpoNile percentage \* Order amount * If ExpoNile share less than the minimum limit, apply minimum limit * If ExpoNile share exceed the maximum limit, apply maximum limit * If ExpoNile share is Amount * ExpoNile share = ExpoNile share amount   + Debit: Reference (Table #01)   + Credit: Reference (Table #02)   + Balance = Latest Balance + Debit - Credit   + Total Orders amount: ∑ Order amount column   + Total ExpoNile share: ∑ ExpoNile share column |
| * Then   And | * System allows admin to: * Search by: (not mandatory) * Transaction id (auto complete) * P/O No. (auto complete) * Filter by: (not mandatory) * Transaction Date /time (Date/time range) * Transaction Type (select one or more from lookup) * Canceled order * Return order (claim document \ upcoming iteration) * Exchange order (claim document \ upcoming iteration) * Cash receipt (settlement document \ upcoming iteration) * Cash voucher (settlement document \ upcoming iteration) * Payment Method (select one or more from lookup) * Warehouse Method (select one or more from lookup) * Delivery Method (select one or more from lookup) * Print * Export to Excel sheet * All filters are dependent on each other * System displays "hyphen symbol" in any cell has null or zero value * System displays the previous balance as an initial balance * System displays the initial balance in the single row on the top of the statement * Just in columns aggregation, system considers the amounts between brackets as a negative value * System updates the total balance depend on the cross filters |

# ExpoNile statement



|  |  |
| --- | --- |
|  | **Merchant views ExpoNile Statement** |
| * As a/an | * Merchant |
| * I want to | * View the ExpoNile’s statement |
| * So that | * Can follow up the transactions and remaining balance |
|  |
| * Scenario * Given * When | * Merchant views the ExpoNile’s statement, search and filter to get a short list * Merchant views “ExpoNile Statement” screen * System displays the selected store’s statement * Transaction Date/Time * Transaction ID * Transaction Type * P/O No. * Warehouse Method * Delivery Method * Payment Method * Total Item price * Summation of item prices grouped by warehouse * Promo Code * Summation of the division amount of the promo code of all items grouped by warehouse * The division amount of the promo code per item =   [(item price / total item prices in the store) \* promo code amount]   * Order Amount (Total item price - Promo code) * Delivery Fees (display the collected delivery fees for the warehouse) * If delivery method is agent, system get fees through integration * If delivery method is delivery boy, system get fess from shop setup * In case delivery agent, system calculate order delivery fees divided on number of stores in the same delivery agent * Total order Amount (order amount + delivery fees) (hidden column) * Wallet * Summation of the division amount of the wallet for all items per store grouped by warehouse * The division amount of the wallet per item =   [(item price / net item prices for PO after promo code) \* wallet amount]   * Payment (Total order amount – Wallet) * ExpoNile Share * Summation of ExpoNile share for all items per store grouped by warehouse * Based on payment method either payment method flagged as “COD” or no “COD” flag which mean online * If ExpoNile share is percentage * ExpoNile share = ExpoNile percentage \* Order amount * If ExpoNile share less than the minimum limit, apply minimum limit * If ExpoNile share exceed the maximum limit, apply maximum limit * If ExpoNile share is Amount * ExpoNile share = ExpoNile share amount   + Debit: Reference (Table #02)   + Credit: Reference (Table #01)   + Balance = Latest Balance + debit - credit   + Total Orders amount: ∑ Order amount column   + Total ExpoNile share: ∑ ExpoNile share column |
| * Then   And | * System allows admin to: * Search by: (not mandatory) * Transaction id (auto complete) * P/O No. (auto complete) * Filter by: (not mandatory) * Transaction Date /time (Date/time range) * Transaction Type (select one or more from lookup) * Sales order * Canceled order * Return order (claim document \ upcoming iteration) * Exchange order (claim document \ upcoming iteration) * Cash receipt (settlement document \ upcoming iteration) * Cash voucher (settlement document \ upcoming iteration) * Payment Method (select one or more from lookup) * Warehouse Method (select one or more from lookup) * Delivery Method (select one or more from lookup) * Print * Export to Excel sheet * All filters are dependent on each other * System displays "hyphen symbol" in any cell has null or zero value * System displays the previous balance as an initial balance * System displays the initial balance in the single row on the top of the statement * Just in columns aggregation, system considers the amounts between brackets as a negative value * System updates the total balance depend on the cross filters |