

## User Stories and Backend Stories for Arifpay Web App UI Screens

---

### 1. Dashboard Overview (Image 1)

**WebApp User Story:** As an admin, I want to see an overview of branches, POS, clerks, and weekly transaction performance, so that I can monitor key operational metrics at a glance.

**Backend Story:**

- Create an API to fetch counts for branches, POS terminals, and clerks.
  - Create an API to aggregate weekly transaction data by day.
  - Create an API to retrieve the most used payment methods with counts and amounts.
  - Implement caching for dashboard data for faster loading.
- 

### 2. Payments Screen (Image 2)

**WebApp User Story:** As a merchant or admin, I want to access different wallet and payment options, So that I can easily process transactions and see recent activities.

**Backend Story:**

- Integrate with Telebirr, CBE Birr, Kacha, and MPESA APIs.
  - Create a service to generate dynamic QR codes.
  - Create CRUD APIs for card, QR, and cash transactions.
  - Create an API to retrieve recent transactions filtered by method.
- 

### 3. Transactions List (Image 3)

**WebApp User Story:** As an admin, I want to view and filter all transaction history, So that I can monitor activity and export data.

**Backend Story:**

- Create an endpoint to fetch transactions with filters (date, method, status, clerk).
- Implement pagination and export functionality (CSV/Excel).
- Secure API with role-based access control.

---

## 4. QR Management (Image 4)

**WebApp User Story:** As a merchant, I want to manage static and dynamic QR codes by store, so that I can assign and reuse them easily.

**Backend Story:**

- Create endpoints for static and dynamic QR code creation and retrieval.
- Assign QR codes to stores with appropriate metadata.
- Store QR codes securely with download URLs.

---

## 5. Inventory Management (Image 5)

**WebApp User Story:** As a store manager, I want to track inventory status, top-selling items, and low-stock items, so that I can manage stock effectively.

**Backend Story:**

- Implement product, category, and unit models.
- Create APIs to track sales, reorder levels, and current stock.
- Create reports for on-demand and low-stock items.

---

## 6. Store Management (Image 6)

**WebApp User Story:** As an admin, I want to add, edit, disable, or delete store branches, So that I can manage store structure efficiently.

**Backend Story:**

- Create CRUD endpoints for branch/store management.
- Attach QR code download functionality to store data.
- Handle soft-delete and reactivation workflows.

---

## 7. Events Management (Image 7)

**WebApp User Story:** As an event organizer, I want to create, edit, and display events, So that I can manage campaigns and public-facing activities.