



Group Members:

Athanase Abayo

Mabinty Mambu

Olivier Kwizera

Victoria Ama Nyonator

Course Name: Web Technologies

Cohort B

Sprint 2 Deliverables Report

Lecturer: Dr. Osafo-Maafo Kwadwo

Teacher Assistant: Barbara-Marie Doh

November 14, 2025

Sprint 2 Overview (Version 2.0)

In Sprint 2, we focused on transforming SwapIt from a basic authentication prototype into a functional peer-to-peer swapping marketplace. The development effort targeted the core customer-facing modules, hardened backend APIs, improved database reliability, and introduced essential UX functionality that makes the platform feel intuitive and usable.

This sprint elevated the system from “login-only” to a **fully navigable item sharing ecosystem** capable of creating listings, initiating swap requests, managing user profiles, and maintaining persistent user actions such as wishlists and saved items.

1. Backend Enhancements

1.1 Item Listing API – Full CRUD Enablement

Sprint 2 completed the internal plumbing required to create, update, retrieve, and delete item listings.

Deliverables:

- New endpoints added under /api/items.php
- Backend logic to:
 - Accept user-submitted forms from add-listing.html
 - Validate uploaded images
 - Store images as JSON arrays
 - Insert/update/delete data using prepared statements
- Automatic population of listing metadata (date, user_id, view_count)

Key Outcomes:

- SwapIt now supports *full lifecycle management* of user listings.
- Items are cleanly tied to user accounts with foreign keys.

- Image handling upgraded with error-safe upload routines.

1.2 Swap Request Engine – v1 Transaction Workflow

A dedicated module was added to handle item-for-item swap operations. This was one of the heaviest backend components in Sprint 2.

Core Features Implemented:

- Request creation with metadata:
 - requester_id
 - owner_id
 - item_requested
 - scheduling time
 - optional message
- Status states:
 - pending → accepted/rejected → completed/cancelled
- Secure database flows using foreign keys & constraints
- Email/dashboard notifications wired at API level (frontend integration in Sprint 3)

Business Logic Added:

- Owners cannot request swaps on their own listings.
- Duplicate swap requests blocked.
- Requests log timestamps for traceability.

1.3 Messaging System – Foundation Layer

A working messaging pipeline was introduced to facilitate user conversations.

What Sprint 2 Achieved:

- Backend table design (messages)
- Message creation endpoint
- Read/unread state tracking
- Message retrieval by conversation thread (user-to-user)

Value Added:

This sets the stage for real-time messaging in Sprint 3.

1.4 Authentication Hardening

Sprint 2 extended Sprint 1's prototype by adding:

- Logout API endpoint
- Session timeout controls
- Automatic session validation on page load
- Input sanitization enforcement
- Detailed error reporting for invalid login attempts
- Secure redirects for unauthorized users

2. Frontend Enhancements

2.1 Item Listing UI

Users can now add, modify, browse, filter, and delete items.

Implemented UI Elements:

- Modern “Add Listing” page with:
 - Category dropdown
 - Multiple image upload
 - Condition selector
 - Price/location fields
 - Description box
- “Browse Items” enhanced with:
 - Price filter
 - Condition filter
 - Category filter
 - Search bar

User Experience Upgrades:

- Toast notifications for actions
- Clean grid layout for items
- Lazy loading images

2.2 Wishlist & Cart Subsystems (LocalStorage v1)

Two persistent client-side modules were finalized.

Wishlist Features:

- Add/remove with heart icon
- Real-time badge counter

- Dedicated wishlist page
- Toast updates

Cart Features:

- Quantity management
- Total price calculation
- Remove with confirmation
- Multi-page synchronization

These modules provide a smooth, intuitive shopping-like experience used later in the Swap Request flow.

2.3 Profile Management UI

A complete profile page was implemented, supporting:

- Profile picture upload
- Personal information editing
- Phone/location update
- Biography section

This enables users to build trust within the community.

2.4 Notification System

A custom notification component was packaged and deployed.

Capabilities:

- Slide-in animation

- Auto-dismiss
- Styling for success/info/warning
- Shared across modules

3. Database Upgrades (SI2025)

Sprint 2 Database Work Included:

New Tables Finalized:

- swap_requests
- saved_items
- reviews
- messages

Database Quality Improvements:

- Foreign key constraints applied
- Cascading deletes configured
- ENUM types introduced for consistent state values
- CHECK constraints for validation
- Performance indexes on:
 - user_id
 - item_id
 - request_status
 - categories

Outcome:

A robust, normalized, and scalable v2 database ready for complex queries.

4. Security Improvements

Sprint 2 placed significant emphasis on operational security.

Enhancements Completed:

- Server-side sanitization for all listing fields
- File upload restrictions (MIME + size)
- CSRF token integration into HTML forms
- Output encoding on all pages
- SQL injection prevention across all endpoints

5. UI/UX Enhancements

Design Goals Achieved:

- Smooth and human-friendly item browsing
- Intuitive dropdown navigation
- Responsive layouts across all pages
- Lightweight animations to improve user engagement
- Empty-state illustrations for wishlist/cart/messages

Sprint 2 improved SwapIt's "feel" dramatically, making it approachable and enjoyable.

6. Testing Completed

Unit Tests:

- Price validation
- Email format checks
- Password hashing + verification
- Image upload handler

Integration Tests:

- Swap request creation + status transitions
- Item CRUD via real API calls
- Authentication + session persistence

User Acceptance Tests:

- Listing creation walkthrough
- Wishlist/cart interactions
- Profile editing flow
- Messaging usability

7. Sprint 2 Deliverables Summary

Module	Status	Description
Authentication	Complete ▾	Hardened sessions, logout, auto-check
Item Listings	Complete ▾	Full CRUD + images + filters
Swap Requests	Complete ▾	Pending/accept/reject/complete cycle
Profiles	Complete ▾	Editable user info + avatar
Wishlist	Complete ▾	LocalStorage implementation
Cart	Complete ▾	Quantity, totals, persistent storage
Messaging	Complete ▾	Basic send/receive + read/unread
Notifications	Complete ▾	Unified toast system