

SwiftPass

Operational Android scanning with local records and configurable server synchronization.

OFFLINE RECORDING

CONFIGURABLE API SYNC

DEVICE ADAPTERS

Input / Output workflows
Local and company workspaces
Searchable scan history
Movement reports and CSV

Scanning needs a complete operating flow

Operators need a clear record of each action and its processing status.

Hardware spread

Scanner vendors use different APIs, trigger behavior and device services.

Offline complexity

Work must continue offline, with a visible queue for later synchronization.

Business confirmation

A local scan record and server approval represent different stages of an operation.

What SwiftPass already does

Implemented workflows for Android phones and scanner terminals.

- Warehouse Input/Output, Event Access and Asset Check-in/out workflow presets.
- Manual entry, camera scanning and device adapters through a shared interface.
- Separate local and company workspaces, with searchable Room history.
- Configurable API sync with accepted, rejected, conflict and retry states.
- Daily movement reports, CSV export and adaptive layouts for large text.

Modular Android foundation

Kotlin and Jetpack Compose, with feature modules separated from shared services.

app

Navigation, manual dependency wiring and WorkManager scheduling.

core:database

Room records, search, migrations and persistent sync queue.

core:network

Runtime REST configuration, remote login and scan receipts.

core:scanner

Provider registry, vendor engines and camera fallback.

feature modules

Login, Scan, History, Settings, Menu and movement reports.

client adaptation

Agree API contracts, business rules, branding and target-device testing.

One app, multiple scanner paths

Shared scan behavior with adapters for specific device services.

Camera

Camera scanning with manual entry available as an alternative.

Zebra

DataWedge integration. Validate the target model and profile.

Sunmi

Scanner-service adapter. Validate the target model and service.

Bluebird

Integrated connector and decode handling. Target-device QA required.

Newland

Extension point. Device integration remains future work.

Compatibility depends on the terminal model, installed services and device configuration.

Local records and company synchronization

Separate workspace storage and a persistent Room queue.

- Local Workspace records on-device without scheduling remote synchronization.
- Company Connection queues records for a configured server and retries failed sends.
- History distinguishes pending, syncing, accepted, failed, rejected and conflict results.
- Search and CSV exports use the stored records. Local recording does not imply server approval.

Configurable REST synchronization

The mobile app records operations and exchanges receipts with a compatible server.

Connector

Runtime server URL, sync/health paths and environment. Basic, Bearer or API-key auth. Company, tenant and device context.

Reconciliation

Accepted, rejected and conflict receipts update the record. Failed sends receive retry state and details.

Background processing

Foreground triggers and WorkManager reuse the queue processor. Android schedules network-constrained retries.

Customer integration requires agreed endpoints and business rules for the selected workflow.

A reusable Android layer for client workflows

Partners can adapt the existing app around their backend and operating process.

- Reuse scanning, local history and synchronization handling across client projects.
- Map the REST contract to a compatible WMS, ERP or access-control backend.
- Validate one workflow and device family before extending the deployment.
- Potential formats include licensing, white label and customer-specific deployment.

Movement reports and operational diagnostics

Visibility into the operations stored in the selected workspace.

Daily movement report

Choose a day and review grouped Input/Output counts, net scans and pending or excluded records. The report describes recorded movements, not authoritative stock availability.

CSV and diagnostics

Export scan history and movement data as CSV. The menu export can include a nonempty local error log to help investigate a problem. Diagnostics remain workspace-scoped.

Pilot scope and delivery steps

Schedule and effort follow the agreed backend, workflow and device requirements.

Scope

Choose one workflow, auth method and device family. Review the demo and define acceptance criteria.

Integration

Map endpoints and fields. Test offline recovery, server receipts and scanning on the selected terminal.

Delivery

Prepare the client APK and configuration. Agree rollout, support responsibilities and next-stage scope.

Potential customers and integration partners

Teams that need an Android scanning interface around an existing service.

- WMS / inventory / ERP product teams.
- Event-access platforms needing entry and exit recording.
- Enterprise mobility integrators.
- Rugged Android scanner and hardware resellers.
- Equipment and asset teams needing check-in/out records.

SwiftPass demo and pilot discussion

Licensing, white label or customer deployment, with terms and scope by agreement.

Review

See the scan flow, history, movement report and synchronization behavior.

Choose

Identify one workflow, backend owner and target device family.

Pilot

Agree the API mapping and acceptance tests for a controlled rollout.

Yasen Apostolov +359 899 010 308 yasenu3@gmail.com