Test Case Template: Drop — Fintech Payment App

Test Case Template: Drop — Fintech Payment App

> **Project:** Drop — Remittance + QR Payments **Version:** 1.0 **Date:** 2026-02-23
> **Author:** John (AI Director) **Status:** Approved **Reviewers:** Alem Bašić (CEO)

Document History

Version	Date	Author	Changes
0.1	2026-02-23	John	Initial test case template with Drop-specific examples

---

1. Test Case ID Format & Naming Convention

Format: `TC-{MODULE_CODE}-{SEQUENCE}`

Part	Description	Example
`TC`	Test Case prefix (always TC)	`TC`
`MODULE_CODE`	2-4 letter module abbreviation	`AUTH`, `REM`, `QR`, `SEC`, `DB`, `PERF`
`SEQUENCE`	3-digit zero-padded number	`001`, `042`, `100`

Drop Module Codes:

- `AUTH` — Authentication & Onboarding
- `REM` — Remittance (Send Money)
- `QR` — QR Payments (Consumer + Merchant)
- `SEC` — Security & Input Validation
- `DB` — Database Compliance (no balance, no CVV)
- `PERF` — Performance Benchmarks
- `RATE` — Exchange Rates API
- `SMK` — Smoke Tests (critical path subset)

Examples:

- `TC-AUTH-001` — Authentication module, registration test
- `TC-REM-015` — Remittance module, fee calculation test
- `TC-DB-001` — DB compliance, no balance column test

Related requirement ID: Link to `AC-{ID}` (acceptance criteria) or `FR-{ID}` (functional requirement)

---

2. Test Suite Organization

Suite ID	Suite Name	Module	Test Cases	Owner
TS-AUTH	Authentication	auth.test.ts	TC-AUTH-001 to TC-AUTH-020	Builder + Validator
TS-REM	Remittance	transactions.test.ts	TC-REM-001 to TC-REM-020	Builder + Validator
TS-QR	QR Payments	transactions.test.ts	TC-QR-001 to TC-QR-010	Builder + Validator
TS-SEC	Security / Validation	validation.test.ts, middleware.test.ts	TC-SEC-001 to TC-SEC-030	Builder + Validator
TS-DB	DB Compliance	db.test.ts	TC-DB-001 to TC-DB-010	Builder + Validator
TS-RATE	Exchange Rates	rates.test.ts	TC-RATE-001 to TC-RATE-005	Builder + Validator
TS-PERF	Performance	api-benchmarks.test.ts	TC-PERF-001 to TC-PERF-010	Builder + Validator
TS-E2E	E2E User Journeys	user-flows.spec.ts, full-flows.spec.ts	TC-E2E-001 to TC-E2E-020	Builder + Validator
TS-CHAOS	Input Chaos	input-chaos.spec.ts	TC-CHAOS-001 to TC-CHAOS-030	Builder + Validator
TS-SMK	Smoke Tests	user-flows.spec.ts (subset)	TC-SMK-001 to TC-SMK-005	Builder + Validator

Suite ID	Suite Name	Module	Test Cases	Owner
TS-REG	Regression Suite	api-endpoints.test.ts	All high-priority TCs	Builder + Validator

3. Individual Test Case Format

Test Case: TC-{MODULE}-{SEQ}

Field	Value
ID	TC-{MODULE}-{SEQ}
Title	{CONCISE_DESCRIPTION}
Description	{FULL_DESCRIPTION} — 1-3 sentences explaining what is being verified and why
Module / Feature	{MODULE_NAME}
Priority	{PRIORITY} — Critical / High / Medium / Low
Type	{TYPE} — Functional / Regression / Smoke / Boundary / Negative / Performance / Security / Compliance
Requirement	{AC_ID} / {FR_ID}
Automation Status	Automated / Manual / Planned
Automation ID	{test_file_path}:{test_name}

Preconditions

1. {PRECONDITION_1}
2. {PRECONDITION_2}
3. {PRECONDITION_3}

Test Data

Field	Value	Notes
{Field}	`{value}`	Stored in Vaultwarden "Drop UAT"

Test Steps

Step	Action	Expected Result
1	{ACTION_1}	{EXPECTED_1}
2	{ACTION_2}	{EXPECTED_2}

Step	Action	Expected Result
3	{ACTION_3}	{EXPECTED_3}

Expected Final Result

{OVERALL_EXPECTED_RESULT}

Post-conditions

- {POSTCONDITION_1}
- {POSTCONDITION_2}

---

4. Priority Definitions

Priority	Definition	Drop Examples
Critical	Core functionality; financial invariant; auth; Pass-through model	Login, remittance, db.test.ts no-balance assertion
High	Important feature; significant user impact if broken	Exchange rates, merchant registration, KYC gate
Medium	Standard feature; workaround exists	Feature flags, notification preferences
Low	Minor feature, cosmetic, or edge case	Error message wording, UI sorting

---

5. Test Case Type Definitions

Type	Description
Functional	Verifies a Drop feature works as specified
Regression	Verifies previously working functionality still works after code change
Smoke	Fast check of most critical paths (subset for quick confidence — 5 tests)
Boundary	Tests at the edges of valid input (e.g., exactly 18 years old; amount = 100 NOK min)
Negative	Tests invalid input, error conditions, unauthorized access
Performance	Verifies response time, throughput under defined load (api-benchmarks.test.ts)
Security	Verifies access controls, injection resistance, bcrypt, JWT

Type	Description
Compliance	Verifies regulatory requirements (PCI-DSS no CVV; GDPR no excess data; AML transaction limits)

---

6. Batch Test Execution Template

Test Run: {RUN_ID} **Environment:** Staging (https://drop-staging.fly.dev/) **Build / Version:** v{VERSION} **Tester:** Validator Agent + Builder Agent **Date:** {DATE}

Test Case ID	Title	Priority	Result	Actual Result / Notes	Defect ID
TC-AUTH-001	User registers successfully	Critical			
TC-AUTH-002	Under-18 rejected	Critical			
TC-AUTH-003	Login with valid credentials	Critical			
TC-REM-001	Remittance fee = 0.5%	Critical			
TC-QR-001	QR payment fee = 1%	Critical			
TC-DB-001	No balance column in users table	Critical			
TC-DB-002	No card_number/cvv in cards table	Critical			

Summary:

- Total: {TOTAL}
- Passed: {PASSED}
- Failed: {FAILED}
- Blocked: {BLOCKED}
- Pass rate: {PASS_RATE}%

---

7. Test Execution Log Format

Timestamp	Test Case ID	Tester	Environment	Build	Result	Duration	Notes
{TIMESTAMP}	TC-AUTH-001	Validator	staging	v0.5.0	Pass	1.2s	

---

8. Defect Linking

Defect format: `BUG-{ID}` (in Mission Control)

Test Case	Defect ID	Severity	Status	Fixed In
TC-{MODULE}-{SEQ}	BUG-{ID}	{SEVERITY}	Open / Fixed / Verified	v{VERSION}

Defect fields required:

- Steps to reproduce (reference test case ID)
- Expected vs actual behavior
- Environment + build version
- Screenshot / screen recording (for E2E failures)
- Severity and priority
- Vitest / Playwright error output

Example Test Cases — Drop Specific

Test Case: TC-AUTH-001

Field	Value
ID	TC-AUTH-001
Title	User registers successfully with valid Norwegian phone and age ≥ 18
Description	Verifies that a new user can complete 3-step registration: email+DOB → OTP → PIN. Tests the core Drop onboarding business process.
Module / Feature	Authentication — User Registration
Priority	Critical
Type	Functional

Field	Value
Requirement	AC-001, FR-001, BR-001
Automation Status	Automated
Automation ID	`src/drop-app/__tests__/auth.test.ts:successful registration`

Preconditions

1. Fresh email address not previously registered in Drop
2. Norwegian phone number (+47)
3. DOB indicating age ≥ 18 (e.g., born 20 years ago)

Test Data

Field	Value	Notes
Email	`e2e-fresh-{timestamp}@test.alai.no`	Unique per test run
Password	`TestDrop123!`	≥ 8 chars
Phone	`+4712345678`	Norwegian format
DOB	20 years ago from test date	Age = 20 years
First Name	`Amir`	Unicode safe
Last Name	`Hasić`	With diacritics

Test Steps

Step	Action	Expected Result
1	POST `/api/auth/register` with valid payload	201 Created; user in DB; no password hash in response
2	POST `/api/auth/verify-otp` with correct 6-digit OTP	200; user proceeds to PIN step
3	POST `/api/auth/setup-pin` with valid 4-digit PIN	200; account activated; JWT httpOnly cookie set
4	GET `/api/auth/me` with JWT cookie	200; user object returned; no password hash

Expected Final Result

Account created and activated. JWT httpOnly cookie set. User redirected to dashboard. Password hash NEVER appears in any API response. No balance column in user record.

Post-conditions

- User exists in DB with kyc_status = 'pending' (mock) or 'approved' (auto in dev mode)

- Audit log entry created for registration event
- Test cleanup: delete user in `afterEach`

Notes / Edge Cases

- Related: TC-AUTH-002 (under-18 rejection)
- Unicode test: Bosnian diacritics in name must be stored correctly (AC-083)

Test Case: TC-AUTH-002

Field	Value
ID	TC-AUTH-002
Title	Under-18 registration rejected with Norwegian error message
Description	Verifies that a user born less than 18 years ago receives a 422 error in Norwegian: "Du må være minst 18 år".
Module / Feature	Authentication — Age Validation
Priority	Critical
Type	Negative / Compliance
Requirement	AC-004, FR-001, BR-002
Automation Status	Automated
Automation ID	`src/drop-app/__tests__/auth.test.ts:under-18 rejected`

Preconditions

1. Registration form accessible

Test Data

Field	Value	Notes
DOB	Today minus 17 years	Age = 17 years old
Email	`underaged@test.alai.no`	

Test Steps

Step	Action	Expected Result
1	POST `/api/auth/register` with DOB indicating age < 18	422 Unprocessable Entity
2	Check response body	Error message contains "Du må være minst 18 år"

Step	Action	Expected Result
3	Check DB	No user record created

Expected Final Result

422 response with Norwegian age validation error. No account created.

Test Case: TC-DB-001

Field	Value
ID	TC-DB-001
Title	Users table has NO balance column — pass-through model invariant
Description	Verifies that the pass-through model architectural invariant is enforced: Drop NEVER stores user balances. Balance is always read from the bank via AISP.
Module / Feature	Database Compliance
Priority	Critical
Type	Compliance
Requirement	NF-AC-020, NFR-COMP05, ADR-003 (pass-through model)
Automation Status	Automated
Automation ID	`src/drop-app/__tests__/db.test.ts:users table has no balance column`

Preconditions

1. Database initialized with current schema

Test Steps

Step	Action	Expected Result
1	Query SQLite schema: `PRAGMA table_info(users)`	Column list returned
2	Assert `balance` not in column list	Test passes — no balance column exists

Expected Final Result

Test passes. DB schema has no `balance` column in `users` table.

Notes / Edge Cases

- This test must NEVER be skipped or disabled
- Any migration adding a balance column must be immediately reverted as it violates ADR-003
- Related: TC-DB-002 (no card_number/cvv), TC-DB-003 (FK constraints enabled)

Test Case: TC-REM-001

Field	Value
ID	TC-REM-001
Title	Remittance fee calculated correctly at 0.5%
Description	Verifies that the remittance fee is exactly 0.5% of the transaction amount (not 0.5% of total debit).
Module / Feature	Remittance — Fee Calculation
Priority	Critical
Type	Functional / Boundary
Requirement	AC-030, AC-031, FR-020
Automation Status	Automated
Automation ID	`src/drop-app/__tests__/transactions.test.ts:fee calculation`

Test Steps

Step	Action	Expected Result
1	POST `/api/transactions/remittance` with amount=1000, currency=RSD	201 Created
2	Check response: fee field	fee = 5 NOK (exactly 0.5% of 1000)
3	POST with amount=2000	201; fee = 10 NOK
4	POST with amount=99 (below minimum)	400 "Amount must be between 100 and 50000 NOK"
5	POST with amount=50001 (above maximum)	400 validation error

Expected Final Result

Fee = amount × 0.005. Minimum 100 NOK; maximum 50,000 NOK per transaction.

Related Documents

- Test Strategy
- Test Plan
- E2E Test Plan
- Acceptance Criteria

Approval

Role	Name	Date	Signature
Author	John (AI Director)	2026-02-23	Approved (AI)
QA Lead	Validator Agent	2026-02-23	Approved (AI)
AI Director (John)	John	2026-02-23	Approved
CEO (Alem)	Alem Bašić	TBD	

Revision #5
Created 2026-02-23 12:06:03 UTC by John
Updated 2026-05-31 20:03:27 UTC by John