Mobile App Architecture

Drop Mobile App

> React Native / Expo Router app in `src/drop-mobile/`.

Tech Stack

Technology	Version/Details
Framework	React Native (Expo)
Routing	Expo Router (file-based)
Fonts	DM Sans (expo-google-fonts), Fraunces (expo-google-fonts)
Navigation	Stack (root) + Tabs (main app)
State	React `useState` (no global state library)
API	Custom fetch wrapper with Bearer token auth
Storage	AsyncStorage (for auth token)

Directory Structure

```
src/drop-mobile/
├── App.js                  # Expo entry (unused — Expo Router takes over)
├── app/
│   ├── _layout.js          # Root Stack layout + font loading
│   ├── index.js            # Welcome screen
│   ├── login.js            # Login screen
│   ├── register.js         # Registration (2-step)
│   ├── history.js          # Transaction history
```

```
│   └── (tabs)/
│       ├── _layout.js        # Tab navigator (4 tabs)
│       ├── index.js          # Dashboard / Home
│       ├── send.js           # Send money
│       ├── scan.js           # QR scanner
│       └── profile.js        # Profile & settings
├── lib/
│   ├── api.js                # API client
│   └── theme.js              # Theme constants
└── assets/                   # Static assets
```

Navigation

Root Stack (`app/_layout.js`)

```
Stack.Screen: index          (Welcome — no header)
Stack.Screen: login          (Login — no header)
Stack.Screen: register       (Register — no header)
Stack.Screen: (tabs)         (Main app — no header)
Stack.Screen: send           (Send money — modal presentation)
Stack.Screen: scan           (QR scan — modal presentation)
```

Font loading happens here: `Fraunces_600SemiBold`, `Fraunces_700Bold`, `DMSans_400Regular`, `DMSans_500Medium`, `DMSans_700Bold`. App shows nothing until fonts are loaded (SplashScreen.preventAutoHideAsync).

Tab Navigator (`app/(tabs)/_layout.js`)

Tab	Label	Icon	Screen
index	Hjem	Unicode house	Dashboard
send	Send	Unicode arrow	Send money
scan	QR	Unicode QR	QR scanner
profile	Profil	Unicode person	Profile

Tab bar style: `backgroundColor: colors.white`, `borderTopColor: colors.border`, `height: 60`. Active tint: `colors.green` (#0B6E35), inactive: `colors.textLight` (#9CA3AF).

Screens

Welcome (`app/index.js`)

- Green background (`colors.green`)
- "drop." wordmark in white Fraunces font
- Headline: "Enklere betalinger. Lavere gebyrer."
- Two buttons: "Logg inn" (outline) → /login, "Opprett konto" (solid white) → /register

Login (`app/login.js`)

- White background
- Email + password fields
- "Logg inn" green button → calls `api.login(email, password)`
- On success: stores token, navigates to `/(tabs)`
- "Opprett konto" link → /register
- Pre-filled demo credentials in dev

Register (`app/register.js`)

- 2-step flow with progress dots indicator
- **Step 1:** firstName, lastName, email, phone
- **Step 2:** password, confirmPassword
- Calls `api.register(data)`, on success navigates to `/(tabs)`

Dashboard (`app/(tabs)/index.js`)

- Greeting: "Hei, {firstName}" with hand wave emoji
- Green balance card: "Total saldo" with formatted NOK amount
- Quick stats row: Sendt, Mottatt, Ventende (sent, received, pending)
- "Siste transaksjoner" section with FlatList
- Each transaction: icon (arrow up=sent, arrow down=received), name, date, amount with color coding
- "Se alle" link → /history
- Pull-to-refresh via `RefreshControl`

Send Money (`app/(tabs)/send.js`)

- 2-step flow:
- **Step 1:** Recipient name input + currency picker (5 currencies with flags)
  - Currencies: BAM (Bosnia), RSD (Serbia), PKR (Pakistan), TRY (Turkey), PLN (Poland)
- **Step 2:** Amount input (NOK) + conversion card showing:
  - Exchange rate (fetched from `api.getRate`)
  - Fee (0.5%)
  - "Mottaker far" (recipient gets) calculated amount
- "Bekreft og send" button → `api.sendRemittance(data)`
- Success: shows confirmation, "Tilbake til hjem" button

QR Scanner (`app/(tabs)/scan.js`)

- Camera placeholder (gray box with QR icon)
- "Skann QR-kode" instruction text
- "Simuler skanning" button for demo
- Nearby merchants list (hardcoded: Ahmetov Kebab, Kafe Oslo, Narvesen)
- Payment flow: merchant info → amount input → confirm → success
- Calls `api.payQR({ merchantId, amount })`

Transaction History (`app/history.js`)

- Filter tabs: Alle, Sendinger (remittance), QR (qr_payment)
- FlatList with transaction items
- Each item: direction icon, recipient/merchant name, date, amount
- Pull-to-refresh
- Filter changes trigger re-fetch via `api.getTransactions({ type })`

Profile (`app/(tabs)/profile.js`)

- User info section: initials avatar, name, email
- "Mine mottakere" section with recipients list (fetched from `api.getRecipients()`)
- Each recipient: name, country, account number
- Settings menu: Sprak, Varsler, Personvern, Vilkar
- Logout button → clears token, navigates to /index

API Client

File: `lib/api.js`

Configuration

```
const API_URL = __DEV__
  ? "http://localhost:3000/api"
  : "https://drop-app.vercel.app/api";
```

Auth Pattern

- Token stored in module-level variable `let token = null`
- All requests include `Authorization: Bearer ${token}` header
- `setToken(t)` / `getToken()` exported for auth management

Available Methods

Method	HTTP	Endpoint	Returns
`api.login(email, password)`	POST	`/auth/login`	`{ token, user }`
`api.register(data)`	POST	`/auth/register`	`{ token, user }`
`api.logout()`	POST	`/auth/logout`	—
`api.getMe()`	GET	`/auth/me`	`{ user }`
`api.getTransactions(params)`	GET	`/transactions`	`{ transactions }`
`api.sendRemittance(data)`	POST	`/transactions/remittance`	`{ transaction }`
`api.payQR(data)`	POST	`/transactions/qr-payment`	`{ transaction }`
`api.getRecipients()`	GET	`/recipients`	`{ recipients }`
`api.addRecipient(data)`	POST	`/recipients`	`{ recipient }`
`api.getMerchants()`	GET	`/merchants`	`{ merchants }`
`api.getRates()`	GET	`/rates`	`{ rates }`
`api.getRate(currency)`	GET	`/rates/{currency}`	`{ rate }`
`api.health()`	GET	`/health`	`{ status }`

Error Handling

All methods use a shared `request(endpoint, options)` function that:

- Adds auth header if token exists
- Parses JSON response
- Throws on non-OK status with `error.status` and `error.data`

Theme

File: `lib/theme.js`

Colors

```
colors: {
  green: '#0B6E35',       // Primary — matches web app
  greenDark: '#095C2C',   // Hover/pressed state
  greenLight: '#E8F5E9',  // Light green bg
  gold: '#D4A017',        // Accent
  white: '#FFFFFF',
  bg: '#FAFCF8',          // Off-white background
  card: '#FFFFFF',
  text: '#1A1A1A',        // Primary text
  textSecondary: '#6B7280', // Secondary text
  textLight: '#9CA3AF',   // Muted text
  border: '#E5E7EB',      // Borders
  error: '#EF4444',       // Errors
  success: '#10B981',     // Success/positive
}
```

Fonts

```
fonts: {
  display: 'Fraunces_700Bold',
  heading: 'Fraunces_600SemiBold',
  body: 'DMSans_400Regular',
  bodyMedium: 'DMSans_500Medium',
  bodyBold: 'DMSans_700Bold',
}
```

Spacing

```
spacing: { xs: 4, sm: 8, md: 16, lg: 24, xl: 32, xxl: 48 }
```

Border Radius

```
radius: { sm: 8, md: 12, lg: 16, full: 9999 }
```

Web App vs Mobile App Comparison

Feature	Web (`drop-app`)	Mobile (`drop-mobile`)
Auth storage	httpOnly cookie	Bearer token (in-memory)
Navigation	Next.js App Router	Expo Router (Stack + Tabs)
Send flow	4 steps	2 steps
Onboarding	4 steps (info, OTP, PIN, success)	2 steps (info, password)
QR scanner	Simulated camera UI	Simulated camera UI
Bottom nav	5 tabs (Hjem, Aktivitet, Skann, Kontoer, Profil)	4 tabs (Hjem, Send, QR, Profil)
Cards page	Yes (feature-flagged)	No
Merchant page	Yes (feature-flagged)	No
Accounts page	Yes (dedicated)	No (balance on dashboard)
History	Dedicated page with filters	Dedicated screen with filters
Feature flags	Environment variables	Not implemented
UI components	shadcn/ui (Radix)	React Native StyleSheet
State	useState + useAuth hook	useState + api module

Development Notes

- **API compatibility:** Mobile app calls the same API endpoints as the web app (same backend)
- **Dev API URL:** `http://localhost:3000/api` — requires the Next.js dev server running
- **Prod API URL:** `https://drop-app.vercel.app/api`
- **No deep linking** configured yet
- **No push notifications** implemented yet
- **No offline support** — requires network for all operations
- **No biometric auth** — login is email/password only

Revision #5
Created 2026-02-18 08:44:25 UTC by John
Updated 2026-06-27 21:54:12 UTC by John