## 📅 \*\*Phase 2: Core Banking Services (Weeks 4-7)\*\*

### \*\*Week 4: Wallet Service Implementation\*\* (Dec 30, 2024 - Jan 5, 2025)

\*\*Day 22 - Mon, Dec 30\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Financial transaction principles, ACID properties

- \*\*Development\*\*:

- [ ] Implement `POST /api/wallets/create` endpoint

- [ ] Add wallet validation (unique per user, currency support)

- [ ] Create wallet activation/deactivation

- [ ] Implement wallet balance inquiry `GET /api/wallets/{userId}/balance`

- \*\*Success Check\*\*: Users can create and query wallets via API

\*\*Day 23 - Tue, Dec 31\*\* (4h dev + 1h learning) \*New Year's Eve\*

- \*\*Learning\*\*: Database transactions, @Transactional management

- \*\*Development\*\*:

- [ ] Implement balance update operations with transaction locking

- [ ] Add concurrent access handling with optimistic locking

- [ ] Create balance change audit trail

- [ ] Implement rollback mechanisms for failed transactions

- \*\*Success Check\*\*: Balance updates are atomic and thread-safe

\*\*Day 24 - Wed, Jan 1\*\* (4h dev) \*New Year's Day\*

- \*\*Development\*\*:

- [ ] Implement internal transfer `POST /api/wallets/transfer/internal`

- [ ] Add transfer validation (sufficient funds, same currency)

- [ ] Create transfer status tracking (PENDING, COMPLETED, FAILED)

- [ ] Implement idempotency keys for duplicate transfer prevention

- \*\*Success Check\*\*: Users can transfer funds between wallets

\*\*Day 25 - Thu, Jan 2\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Event-driven architecture for financial systems

- \*\*Development\*\*:

- [ ] Implement transaction history `GET /api/wallets/{userId}/transactions`

- [ ] Add pagination and filtering for transaction queries

- [ ] Create transaction event publishing

- [ ] Implement real-time balance updates

- \*\*Success Check\*\*: Complete transaction history with real-time updates

\*\*Day 26 - Fri, Jan 3\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Financial reporting basics

- \*\*Development\*\*:

- [ ] Implement daily transaction summaries

- [ ] Add wallet activity reports

- [ ] Create admin endpoints for wallet management

- [ ] Implement wallet suspension/restoration

- \*\*Success Check\*\*: Comprehensive wallet reporting and admin controls

\*\*Day 27 - Sat, Jan 4\*\* (4h dev)

- \*\*Development\*\*:

- [ ] Add currency conversion support

- [ ] Implement multi-currency wallet operations

- [ ] Create exchange rate service integration (simulated)

- [ ] Add cross-currency transfer validation

- \*\*Success Check\*\*: Multi-currency wallet operations functional

\*\*Day 28 - Sun, Jan 5\*\* (4h dev)

- \*\*Weekly Review & Polish\*\*:

- [ ] Test high-concurrency scenarios

- [ ] Implement circuit breaker for wallet operations

- [ ] Add comprehensive error handling

- [ ] Performance test wallet APIs

- \*\*Week 4 Deliverable\*\*: ✅ Full wallet management with internal transfers

---

### \*\*Week 5: Payment Method Service\*\* (Jan 6 - Jan 12, 2025)

\*\*Day 29 - Mon, Jan 6\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: PCI DSS compliance basics, tokenization principles

- \*\*Development\*\*:

- [ ] Design payment\_methods table schema

- [ ] Implement `POST /api/payment-methods/cards` endpoint

- [ ] Add card validation (Luhn algorithm, expiry date)

- [ ] Create card tokenization service

- \*\*Success Check\*\*: Cards can be tokenized and stored securely

\*\*Day 30 - Tue, Jan 7\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Encryption techniques, secure data storage

- \*\*Development\*\*:

- [ ] Implement AES encryption for sensitive card data

- [ ] Add key management system for encryption keys

- [ ] Create secure card data retrieval

- [ ] Implement card data masking in responses

- \*\*Success Check\*\*: Card data encrypted at rest and properly masked

\*\*Day 31 - Wed, Jan 8\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Bank account verification processes

- \*\*Development\*\*:

- [ ] Implement `POST /api/payment-methods/bank-accounts`

- [ ] Add bank account validation (routing number, account number)

- [ ] Create simulated bank account verification flow

- [ ] Implement bank account tokenization

- \*\*Success Check\*\*: Bank accounts can be linked and verified

\*\*Day 32 - Thu, Jan 9\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Payment method lifecycle management

- \*\*Development\*\*:

- [ ] Implement `GET /api/payment-methods/user/{userId}`

- [ ] Add default payment method selection

- [ ] Create payment method update/delete endpoints

- [ ] Implement payment method status tracking

- \*\*Success Check\*\*: Users can manage multiple payment methods

\*\*Day 33 - Fri, Jan 10\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Risk management in payment methods

- \*\*Development\*\*:

- [ ] Implement payment method verification flows

- [ ] Add suspicious activity detection

- [ ] Create payment method usage limits

- [ ] Implement payment method expiration handling

- \*\*Success Check\*\*: Comprehensive payment method security

\*\*Day 34 - Sat, Jan 11\*\* (4h dev)

- \*\*Development\*\*:

- [ ] Add webhook for payment method events

- [ ] Implement payment method analytics

- [ ] Create admin endpoints for payment method management

- [ ] Add bulk operations for payment methods

- \*\*Success Check\*\*: Complete payment method management system

\*\*Day 35 - Sun, Jan 12\*\* (4h dev)

- \*\*Weekly Review & Polish\*\*:

- [ ] Security audit of payment method storage

- [ ] Performance test payment method APIs

- [ ] Implement rate limiting for payment method operations

- [ ] Document PCI compliance measures

- \*\*Week 5 Deliverable\*\*: ✅ Secure payment method management with tokenization

---

### \*\*Week 6: Central Payment Processor Service\*\* (Jan 13 - Jan 19, 2025)

\*\*Day 36 - Mon, Jan 13\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Card network communication protocols

- \*\*Development\*\*:

- [ ] Design payment processor database schema

- [ ] Implement `POST /api/processor/authorize` endpoint

- [ ] Create authorization request formatting

- [ ] Add basic card network simulation

- \*\*Success Check\*\*: Authorization requests can be processed

\*\*Day 37 - Tue, Jan 14\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Authorization vs Settlement flows

- \*\*Development\*\*:

- [ ] Implement authorization hold mechanism

- [ ] Add authorization expiration handling

- [ ] Create authorization reversal endpoint

- [ ] Implement authorization status tracking

- \*\*Success Check\*\*: Authorization holds work correctly

\*\*Day 38 - Wed, Jan 15\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Settlement and capture processes

- \*\*Development\*\*:

- [ ] Implement `POST /api/processor/settle` endpoint

- [ ] Add batch settlement processing

- [ ] Create settlement reconciliation

- [ ] Implement settlement failure handling

- \*\*Success Check\*\*: Settlement process completes successfully

\*\*Day 39 - Thu, Jan 16\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Refund and dispute management

- \*\*Development\*\*:

- [ ] Implement `POST /api/processor/refund` endpoint

- [ ] Add partial refund capability

- [ ] Create refund status tracking

- [ ] Implement refund reversal process

- \*\*Success Check\*\*: Full refund functionality operational

\*\*Day 40 - Fri, Jan 17\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Fraud detection basics

- \*\*Development\*\*:

- [ ] Implement basic fraud checks (velocity, amount patterns)

- [ ] Add transaction scoring system

- [ ] Create fraud rule engine

- [ ] Implement manual review workflow

- \*\*Success Check\*\*: Basic fraud detection flags suspicious transactions

\*\*Day 41 - Sat, Jan 18\*\* (4h dev)

- \*\*Development\*\*:

- [ ] Add processor webhook for async notifications

- [ ] Implement processor health monitoring

- [ ] Create processor performance metrics

- [ ] Add circuit breaker for processor operations

- \*\*Success Check\*\*: Processor is resilient and monitored

\*\*Day 42 - Sun, Jan 19\*\* (4h dev)

- \*\*Weekly Review & Polish\*\*:

- [ ] End-to-end payment flow testing

- [ ] Load test processor service

- [ ] Implement retry mechanisms for failed processing

- [ ] Document processor API specifications

- \*\*Week 6 Deliverable\*\*: ✅ Central payment processor with card network simulation

---

### \*\*Week 7: Transaction Service & Reporting\*\* (Jan 20 - Jan 26, 2025)

\*\*Day 43 - Mon, Jan 20\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Event sourcing patterns

- \*\*Development\*\*:

- [ ] Design transaction event schema

- [ ] Implement transaction event storage

- [ ] Create transaction event publishing

- [ ] Add transaction event consumers

- \*\*Success Check\*\*: All financial events are properly recorded

\*\*Day 44 - Tue, Jan 21\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: CQRS pattern implementation

- \*\*Development\*\*:

- [ ] Implement `GET /api/transactions/user/{userId}`

- [ ] Add transaction query with filtering

- [ ] Create transaction read model

- [ ] Implement transaction search optimization

- \*\*Success Check\*\*: Fast transaction queries with complex filtering

\*\*Day 45 - Wed, Jan 22\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Financial reporting requirements

- \*\*Development\*\*:

- [ ] Implement `GET /api/transactions/reports/daily`

- [ ] Add transaction summary endpoints

- [ ] Create revenue reporting

- [ ] Implement transaction analytics

- \*\*Success Check\*\*: Comprehensive transaction reporting

\*\*Day 46 - Thu, Jan 23\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Audit trail compliance

- \*\*Development\*\*:

- [ ] Implement immutable transaction logging

- [ ] Add audit trail for all financial operations

- [ ] Create audit report generation

- [ ] Implement audit trail verification

- \*\*Success Check\*\*: Complete audit trail for compliance

\*\*Day 47 - Fri, Jan 24\*\* (4h dev + 1h learning)

- \*\*Learning\*\*: Data export and integration

- \*\*Development\*\*:

- [ ] Implement transaction data export

- [ ] Add webhook for transaction events

- [ ] Create real-time transaction streaming

- [ ] Implement transaction reconciliation

- \*\*Success Check\*\*: Transaction data available for external systems

\*\*Day 48 - Sat, Jan 25\*\* (4h dev)

- \*\*Development\*\*:

- [ ] Add transaction dispute management

- [ ] Implement transaction correction workflow

- [ ] Create transaction archiving strategy

- [ ] Add transaction data retention policies

- \*\*Success Check\*\*: Complete transaction lifecycle management

\*\*Day 49 - Sun, Jan 26\*\* (4h dev)

- \*\*Weekly Review & Polish\*\*:

- [ ] Test transaction consistency across services

- [ ] Implement transaction replay capability

- [ ] Performance test transaction queries

- [ ] Document transaction API specifications

- \*\*Week 7 Deliverable\*\*: ✅ Comprehensive transaction service with reporting

---

## 🎯 \*\*End of Phase 2 Review\*\* (Jan 27, 2025)

\*\*Status Check\*\*:

- [ ] Full wallet management with internal transfers

- [ ] Secure payment method management

- [ ] Central payment processor operational

- [ ] Comprehensive transaction tracking and reporting

- [ ] Basic fraud detection implemented

\*\*Phase 2 Complete!\*\* ✅ Core banking services are now operational!