

# MoneyBee Money Transfer System - Case Study

---

## 1. Company and System Overview

---

MoneyBee is a money transfer application developed by "HappyCash Inc." and used in 200+ branches across the country. The system enables branch employees to efficiently manage customer transactions.

HappyCash is a fintech company operating in Turkey, processing thousands of money transfer transactions daily. Services such as money sending, receiving, and basic customer management are offered through the MoneyBee application.

## 2. Current Situation and Problems

---

MoneyBee currently operates as a monolithic application. All features are in the same codebase and use a single database.

**Main problems encountered:**

- System slows down during peak hours
- Entire system needs to be restarted for small updates
- Confusion when different teams work on the same code
- When one feature breaks, the entire system is affected
- Outages due to dependency on external services

## 3. Business Requirements

---

HappyCash wants to modernize the MoneyBee system. The plan is to redesign the system's core features as separate services.

### 3.1 Auth Module

A simple authentication module to manage API access.

**Requirements:**

- API Key based authentication
- Every request must be authenticated
- Rate limiting (maximum 100 requests per minute)

### 3.2 Customer Service

Service managing sender and receiver information in money transfer transactions.

**Customer information:**

- Name, Surname
- National ID Number
- Phone number
- Date of birth
- Customer type (Individual/Corporate)

**Business Rules:**

- KYC verification must be performed with KYC Verification Service for new customer registration
- Customers who fail KYC verification should not be registered in the system
- Transfer Service should be notified when customer status changes (active/passive/blocked)
- Pending transfers of blocked customers should be automatically cancelled

### 3.3 Transfer Service

The main service where money transfer transactions are performed.

**For each transaction:**

- Sender customer (must be verified from Customer Service)
- Receiver customer (must be verified from Customer Service)
- Amount sent and currency
- Transaction fee
- Transaction code (code for the receiver to withdraw money)
- Transaction status (PENDING, COMPLETED, CANCELLED, FAILED)

**Business Rules:**

- Risk check must be performed with Fraud Detection Service before each transfer
- Exchange Rate Service must be used for transfers between different currencies
- Fees for cancelled transactions must be refunded
- Daily transfer limit per customer: 10,000 TRY

## 4. Technical Requirements

---

### 4.1 Mandatory Technologies

- .NET Core 8.0 (C#)
- RESTful API design
- Docker support
- JWT or API Key Authentication
- OpenAPI/Swagger documentation
- Data Persistence (Database or cache - your choice)

## 4.2 External Services

Ready-made services that are MANDATORY to use in your system:

### Fraud Detection Service

- Docker Image: `bpnpay/fraud-service:latest`
- Description: Performs fraud risk control in every money transfer

### KYC Verification Service

- Docker Image: `bpnpay/kyc-service:latest`
- Description: Performs customer identity verification

### Exchange Rate Service

- Docker Image: `bpnpay/exchange-rate-service:latest`
- Description: Provides current exchange rates between different currencies

## 5. Business Rules

---

### 5.1 Transfer Operations

- Fraud Detection Service must be called for every transfer
- Transfers with HIGH risk score should be automatically rejected
- Transfers with LOW risk score should be directly approved
- For transfers over 1,000 TRY, regardless of fraud check result, a 5-minute waiting period for transaction approval must be applied
- Daily transfer limit: 10,000 TRY

### 5.2 Customer Operations

- KYC verification is mandatory for new customer registration
- Customers under 18 years old are not accepted
- National ID Number must pass algorithm validation
- Tax number is mandatory for corporate customers

### 5.3 Foreign Exchange Operations

- Exchange Rate Service must be used for transfers between different currencies
- Transfers in currencies other than TRY should be possible

## 6. Sample Scenarios

---

### Scenario 1: Money Sending

1. Sender customer information is entered (new registration if not exists)
2. Receiver information is entered
3. Amount to be sent and currency are entered
4. If currency is not TRY, exchange rate is retrieved from Exchange Rate Service
5. Fraud check is performed (Fraud Detection Service)
6. Based on risk assessment:
  - **LOW:** Transaction is directly approved
  - **HIGH:** Transaction is automatically rejected and customer is notified
7. Transaction is approved and transaction code is generated

### Scenario 2: Money Receiving

1. Customer comes to branch with transaction code
2. Transaction code is entered into the system
3. Transaction details are displayed
4. Customer identity is verified (through Customer Service)
5. Money is delivered and transaction is closed

### Scenario 3: Transaction Cancellation

1. Customer wants to cancel the transaction
2. Query is made with transaction code
3. If money has not been received yet, it is cancelled
4. Transaction fee is refunded

## Scenario 4: Foreign Exchange Transfer

1. Customer wants to send 100 USD
2. USD/TRY rate is retrieved from Exchange Rate Service
3. TRY equivalent is calculated
4. Fraud check is performed based on TRY value
5. If transaction is approved, both USD and TRY values are recorded

## 7. Success Criteria

---

### Functional Requirements

- ✓ 3 separate microservices (Auth, Customer, Transfer)
- ✓ Inter-service communication
- ✓ Integration of external services
- ✓ Implementation of all business rules
- ✓ Data persistence (transfer information must not be lost)

## 8. Delivery

---

**Delivery Time:** 5 business days

**Delivery Format:** GitHub Repository (public)

**Repository must include:**

- Source code
- README.md (installation and usage instructions)
- API documentation
- Postman collection or Swagger UI

## 9. Tips

---

### ⚠ Important Notes:

- External services may not always be available
- Fraud service may respond slowly
- KYC service may sometimes return errors
- Pay attention to race conditions
- Transaction idempotency is important

## 10. Questions

---

For technical questions: [kuygur@bpn.com.tr](mailto:kuygur@bpn.com.tr)

---

Good luck! 🍀

HappyCash Tech Team

© 2025 HappyCash Inc. - Confidential