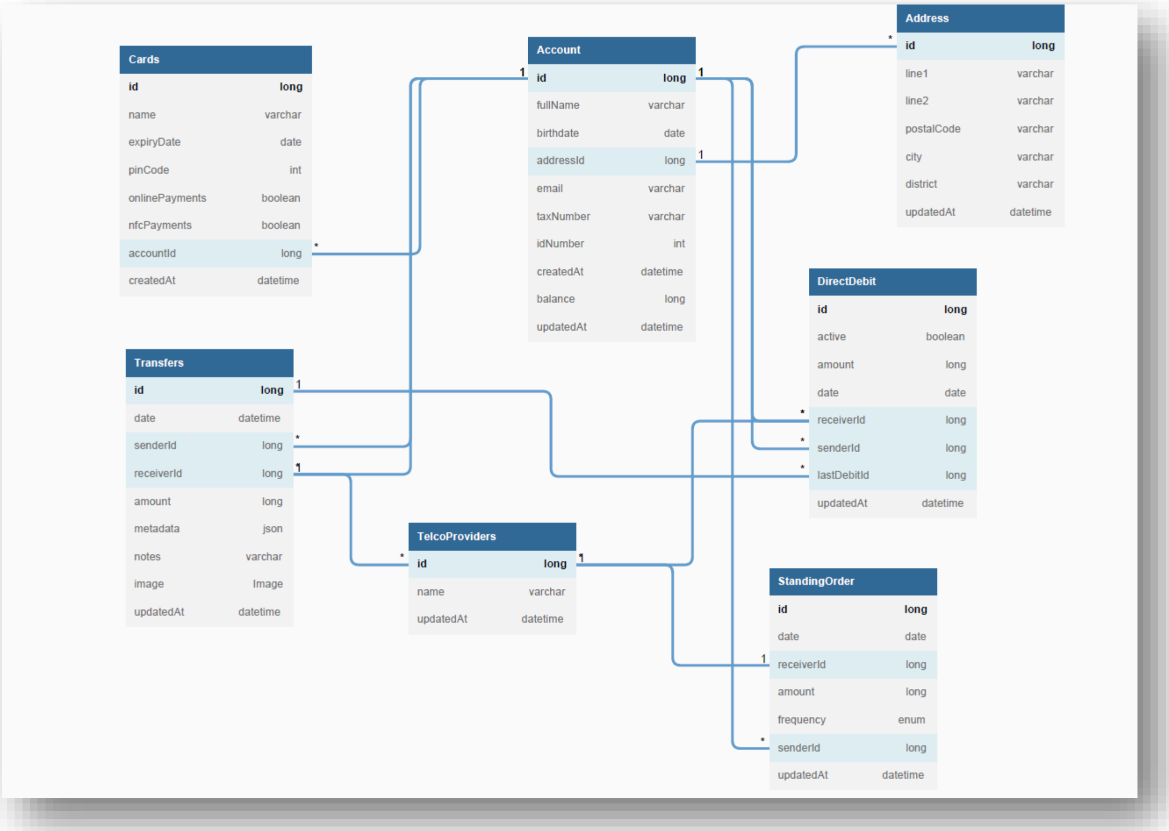
Diagram Documentation

Project-UpBank



Oleksandra Kiselova

Diagram Documentation - UpBank

Contents

Introduction

The below text serves as appended documentation to the presented ER Diagram of the use of database for the project “UpBank”. Firstly, the system requirements are presented against an explanation of use for the same, according to the database diagram. Secondly, database entities and their respective attributes are further filed with details and conditions under which the latter shall be inserted.

System Requirements

1. Create an account: To create an account, the tables Account and Cards shall be used. They shall create a distinctive ID connected to the user's full name, birthdate, tax number and ID number. The account id will generate the account's IBAN. When creating an account, the system shall automatically create two debit cards, called "Physical Card" and "Virtual Card", so will create the Cards table.

2. Close an account: The system shall use the Account table to delete the user's records. In case the balance in the account is positive, the latter must be transferred to another account. Only when the balance is zero, an account permanent deletion can proceed.

3. View account balance: The system shall use the table Account, where the attribute balance shows account’s current balance. The balance is determined by the sum of all account transactions, thus the attribute amount from table Transactions shall be implemented in the calculation of current balance.

4. **View transactions:** The system shall use the Transfers and Account tables, to get the sender and receiver data. It shall pick by the transaction’s id and shall include the transaction date, transaction amount, sender and receiver. Association of image and notes to a transaction – an image and text notes can be added in a transaction in the respective image and notes attributes.

The user shall be able to filter transactions according to date, type and sender/recipient:

- Filter by date: the user shall be able to put the starting date and an ending date to see the transactions within those dates.
- By type: (expense/income) the user shall be able to see transactions where if it is an expense the SenderId = id of the user, and if it is an income, the RecieverId=id of the user.
- By sender/recipient: the user shall be able to see transactions filtering by recipient or sender id.

4.1 Change transaction's notes: the system shall use the Transfers table, attribute notes in order to change the notes of a specific transaction.

4.2 Get a proof of transaction: the system shall use the Transfers table, pick up a transaction id and include amount, date, senderId and metadata.

5. Transfer Money: the system shall use the Transfers table and Account.

5.1 Make a bank transfer by IBAN: the system shall use the Transfers and Account tables. The transfers shall require the user to enter a valid IBAN as a destination id, which could also be stored under the account table.

5.2 Make a service payment: the system shall use the Transfers and Account table Transfers will require a reference and an entity for the payment, where the receiverId could be associated with the Account.

5.3 Make a government payment: the system shall use the Transfers and Account tables. The system shall select the respective metadata with payments to the government. It shall require an amount only, as government account will be a recieverId with an entity that shall automatically linked as an account.

5.4 Make a telco payment: the system shall use the Transfers and Telco Provider tables. The system shall select the metadata with the option mobile phone charging. A mobile phone number with maximum of nine digits, starting only with 91,92,93 or 96 shall be accepted. The receiver shall be a telecom directly recognized by the TelcoProvider table (id and name).

6. View Schedule a transfer: the system shall use the StandingOrder and Account tables. The SenderId shall be derived from the Account table; the receiverId shall be a different IBAN account, which can also be associated as a distinctive id either from the Account or TelcoProvider tables.

6.1 Schedule a transfer: the system shall use the StandingOrder and Account tables. To schedule a transfer, the user will provide a respective IBAN, an amount, a frequency, and a starting date.

6.2 Change a schedule transfer: the system shall use the StandingOrder table, so it can change the needed attributes such as the status, the receiverId or the amount of the transfer.

6.3 Delete a schedule transfer: the system shall use the StandingOrder table, to delete a scheduled transfer. The system shall erase the respective order id.

7. View Direct debits:

The system shall use the tables DirectDebit, Transfers and Account. To see the direct debits already concluded, the system shall pick up all last transactions associated with the user’s account, which have been identified as such in the transaction metadata. These shall be able to be filtered by the same filters applying for all type of transactions presented in Case use 4.

7.1 Toggle a direct debit: The system shall use the tables DirectDebit, Transfers and Account. The table DebitDirect has an active attribute, which is a Boolean (true/false). In such way, it possible to switch on or off this payment.

8. View cards: The system shall use the Cards and Account tables. The card ID is generated from the account ID, together with its name (physical or virtual). When the system is creating a card, the expiry date must be valid for 2 years after the creation of the card. These attributes shall be visible in the createdAt and expiryDate.

8.1 Toggle online payments: The system shall use the Cards and Account tables. Similar to a direct debit, the Cards entity has an active attribute, which is a Boolean (true/false). It shall be changed to switch on or off this option.

8.2 Toggle NFC payments: The system shall use the Cards and Account tables. Similar to a direct debit, the Cards entity has an active attribute, which is a Boolean (true/false). It shall be changed to switch on or off this particular option.

8.3 Change PIN: The system shall use the Cards and Account tables. The entity cards shall store the current pin under the pinCode attribute. The system shall access the card’s id, identify the name and change the pin, storing the new one. The new pin must be four digits.

9. Get Documents: The system shall use the Account table to pick up user’s account data for any type of documentation.

9.1 Get IBAN proof: The system shall use the table Account, where it shall take the id attribute. The id is linked to an IBAN and an account number, which are automatically generated on the opening of the account, however not stored in the database as separate numbers. The system shall also use the full name attribute to be able to generate the final proof of IBAN.

9.2 Get account statements: The system shall use Account and Transfers tables. It shall make use of the id, name and balance in euro from account .Id, amount in euro, date, sender, receiver and metadata from the transaction’s entity.
The system shall be able to generate automatic monthly statements when a month is concluded, by filtering starting and ending date of the last month and inserting the information of all transactions occurred in that period.

The same logic shall be applied when an account statement is requested by the user for a custom period, where the user inserts the start and end date of the desired period.

10. Change account information: The system shall use the Account and Address tables. The system shall be able to accept changes in the email, phone number and address details. Address details shall be in conformity with the fields requested (Address line 1, Address line 2 (optional), Postal code (####-###), City, District). The system shall not be accepting changes in attributes such as name and tax number for security reasons. When changes in the account table are performed, the attribute updatedAt shall be automatically changed to the date of the performed changes.

Database Entities

Account

Atribute	Data type	Description
id	long	Unique, primary key, from the id will be generated a IBAN. The type is classified as long because they can store more numbers and do not have decimal places. All values in Long are divided by 100.
fullName	varchar	The first and last name of the user.
birthdate	date	The system will ensure the user is at least 18 years old.
addressId	long	Related to an ID of the Address table.
email	varchar	Email address of the user, the system will ensure the given email is not already in use (unique).
taxNumber	varchar	The tax number of the user. Shall be able to accept all types of characters is case account holder has a foreign tax number.
idNumber	int	The identification document number of the user
createdAt	datetime	Date of creation of the account.
balance	long	The amount in euro that is available. The type is classified as long because they can store more numbers and do not have decimal places. All values in Long are divided by 100.

Cards

Atribute	Data type	Description
id	long	Unique, primary key, identifies the different cards
name	varchar	Name of the card, the system will generate two debit cards, called “Physical Card” and "Virtual Card”
expiryDate	date	The Expiry date of the card shall be an automatic + 2 years after creation date.
pincode	int	Pin code to gain access to the card. It can be changed by user.
onlinePayments	Boolean	Online payments are a boolean type, active or disabled
nfcPayments	Boolean	NFC payments are a boolean type, active or disabled
accountId	long	account ID, associated with the id of the user’s account.The type is classified as long because they can store more numbers and do not have decimal places. All values in Long are divided by 100.
createdAt	datetime	The registered date of the creation of the card.

Transfers

Atribute	Data type	Description
id	long	Unique, primary key identifies each transfer. The type is classified as long because they can store more numbers and do not have decimal places. All values in Long are divided by 100.
Date	datetime	The date and time a transfer has occurred.
senderId	long	Account id of the sender shall be user's own account. The type is classified as long because they can store more numbers and do not have decimal places. All values in Long are divided by 100.
receiverId	long	Account id of the receiver The type is classified as long because they can store more numbers and do not have decimal places. All values in Long are divided by 100.
amount	long	Amount in euro that will be transferred. The type is classified as long because they can store more numbers and do not have decimal places. All values in Long are divided by 100.
metadata	json	Is the type of transfers in json format, which can be a transfer via IBAN, payment by a reference, government payments and mobile phone top-ups. It shall store different type of data, depending on the type of payment that is performed.
notes	varchar	Associate descriptive notes, optional.
images	image	Associated descriptive image, optional

TelcoProviders

Atribute	Data type	Description
id	long	Unique, primary key, which identifies the Telco Providers. The type is classified as long because they can store more numbers and do not have decimal places. All values in Long are divided by 100.
name	varchar	The id that will be associate to the below telecoms company names: • Lycamobile GT Mobile • MEO • MEO Card • MEO Card - PT Hello /PT Card • MEO Card- TelefoneHello • MEO Escola Digital • Moche • NOS • NOS- Escola Digital • SAPO • Sapo ADSL • UZO • Via Card • Vodafone • WTF
updatedAt	datetime	Date of the last update of database.

StandingOrder

Atribute	Data type	Description
id	long	Unique, primary key, which identifies the Standing order. The type is classified as long because they can store more numbers and do not have decimal places. All values in Long are divided by 100.
recieverId	long	The system shall associate from ID of table Account or from TelcoProviders. The type is classified as long because they can store more numbers and do not have decimal places. All values in Long are divided by 100.
amount	long	The amount in euro that will be paid. The type is classified as long because they can store more numbers and do not have decimal places. All values in Long are divided by 100.
frequency	Enum	Frequency shall be classes as : daily, weekly, monthly, and annually.
senderId	long	The system shall pick up the Id from the table Account.

DirectDebit

Atribute	Data type	Description
id	long	Unique, primary key, which identifies the direct debit.
active	Boolean	Shows the active or not active status of the direct debit.
amount	long	The amount in euro connected to this debit.
receiverId	long	The system shall associate from ID of table Account or from TelcoProviders.
senderid	long	The system shall pick up the Id from the table Account.
lastDebit	long	It is related to the id of a transfer, which allows checking which transfers are made by date and when was the last.

Address

Atribute	Data type	Description
id	long	Unique, primary key, which identifies the address of the user
Line1	varchar	First line of address of the user, mandatory
line2	varchar	Second line of address of the user (optional)
postalcode	varchar	Shall allow post codes in format (####-###), but also entering different type of characters, mandatory field.
city	varchar	Shall allow entering different type of characters, mandatory field.
district	varchar	Shall allow entering different type of characters, mandatory field.