backup link

Database & Table Schema

Group 30

User(user_id, name, joined_date)

Account(account_num, owner, balance, account_type, currency)

Foreign keys: owner refers User, currency refers Currency

Currency(currency, symbol)

ExchangeRate(from_currency, to_currency, rate)

Foreign keys: from_currency and to_currency refer Currency

Transaction(id, date, from_account_num, to_account_num, description, amount, from_balance, to_balance)

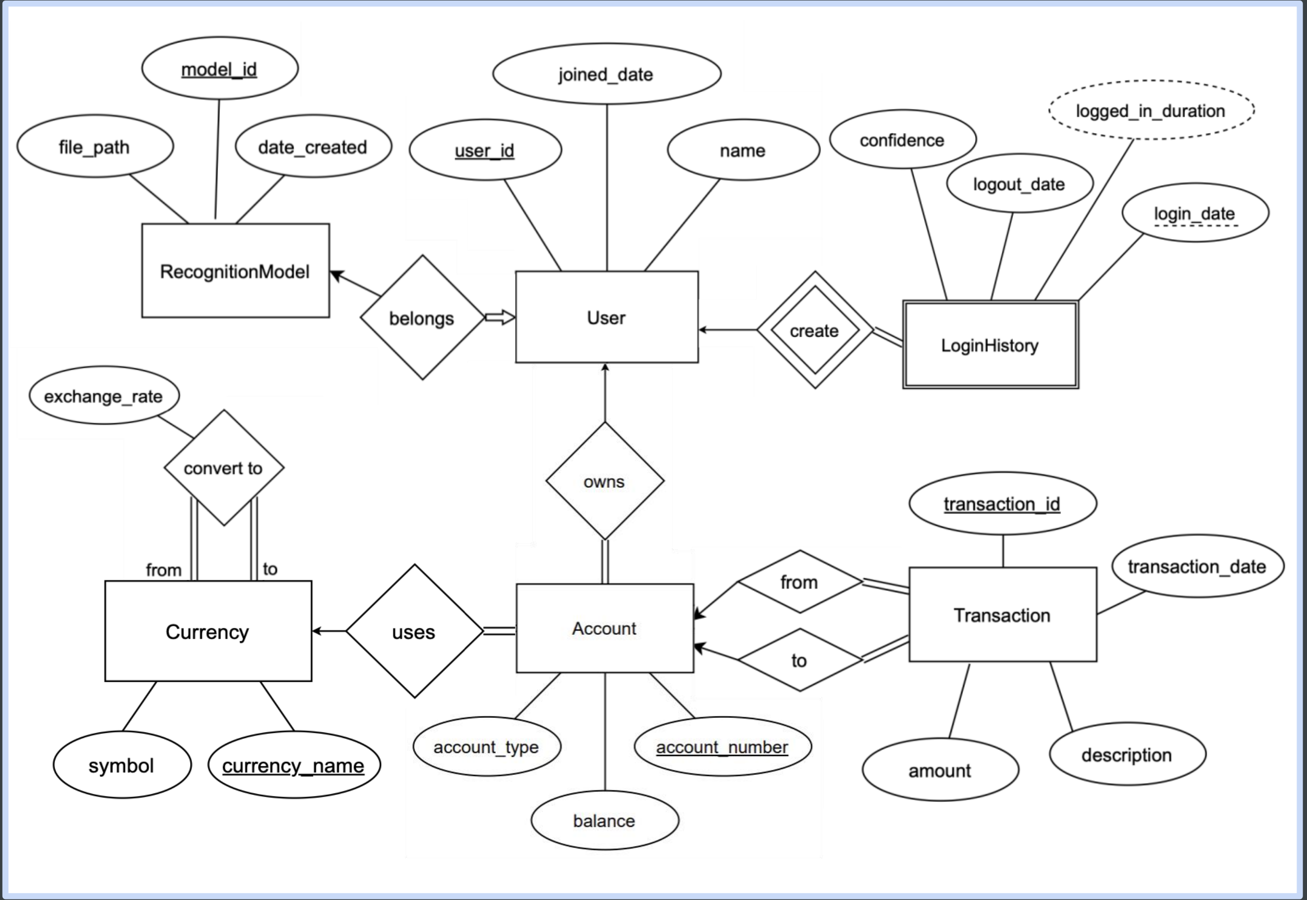
Foreign keys: from_account_num and to_account_num refer Account

Model(model_id, file_path, date_created, user)

Foreign keys: user refers User

LoginHistory(login_date, user_id, confidence, logout_date)

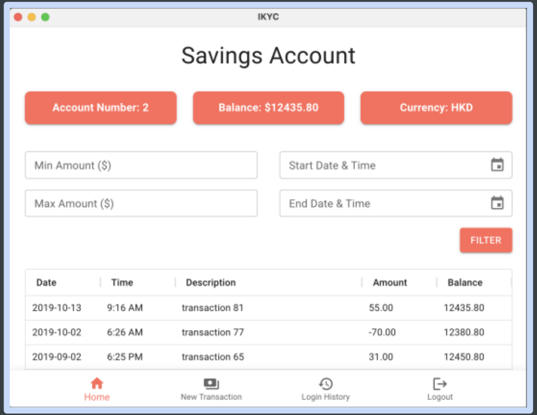
Foreign keys: user_id refers User



Software & UI Design and Implementation

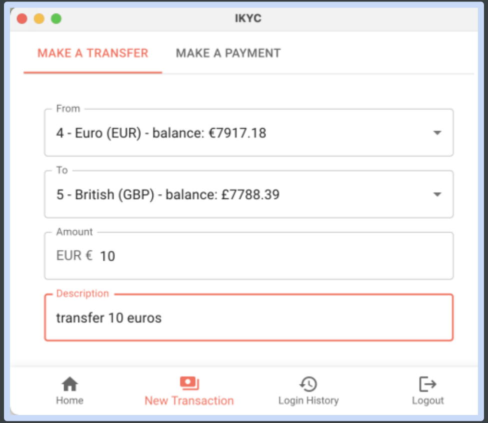
Highlighted Functionalities

- Specify username when logging in
- Sort transactions by date & amount
- Create transactions:
  - Transfer between user's own accounts
  - Transfer to other user's account
  - Support for currency conversion
- Responsive, scalable GUI



Savings Account

Date	Time	Description	Amount	Balance
2019-10-13	9:16 AM	transaction 81	55.00	12435.80
2019-10-02	6:26 AM	transaction 77	-70.00	12380.80
2019-09-02	6:25 PM	transaction 65	31.00	12450.80



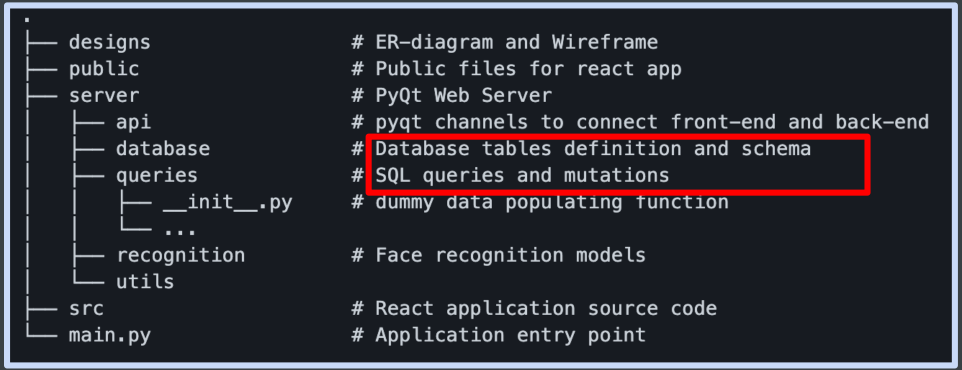
From: 4 - Euro (EUR) - balance: €7917.18

To: 5 - British (GBP) - balance: £7788.39

Amount: EUR € 10

Description: transfer 10 euros

Technology Used	Purpose
React.js	UI framework
SQLAlchemy Core	MySQL connector and schema creation
PyQt	GUI library
PyQt WebEngine	Embed www content into QT applications



```
.
├── designs              # ER-diagram and Wireframe
├── public               # Public files for react app
├── server               # PyQt Web Server
│   ├── api              # pyqt channels to connect front-end and back-end
│   ├── database         # Database tables definition and schema
│   ├── queries          # SQL queries and mutations
│   │   ├── __init__.py  # dummy data populating function
│   │   └── ...
│   ├── recognition      # Face recognition models
│   └── utils
├── src                  # React application source code
└── main.py              # Application entry point
```

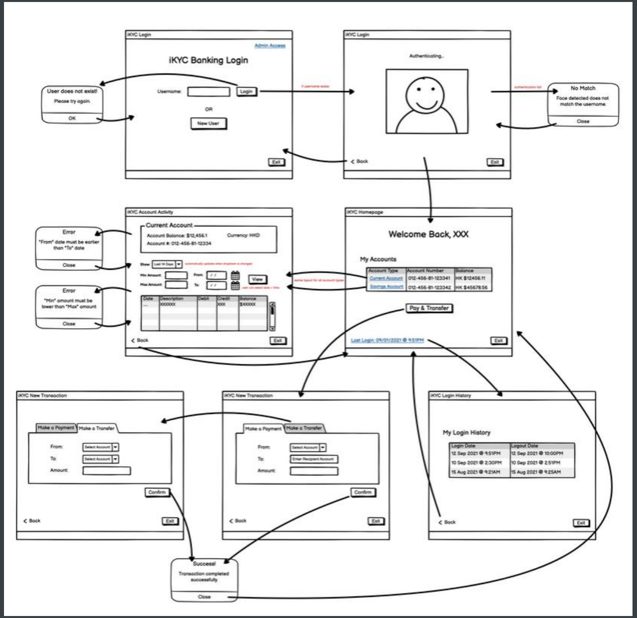
Development Process & Contributions

Software, Database & UI Design

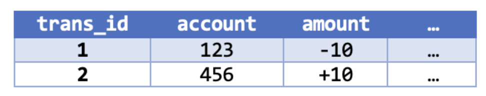
Frontend Development

Backend Development

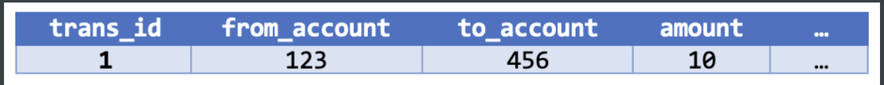
- ER diagram & schema
- Wireframe prototype
- React app setup



- **Problem**: redundancy when each transaction stored as 2 records
- **Solution**: refactor transaction table to store each transaction as one record



trans_id	account	amount	...
1	123	-10	...
2	456	+10	...



trans_id	from_account	to_account	amount	...
1	123	456	10	...

Group Member	Contribution
[redacted]	ER diagram, front & backend code
[redacted]	UI prototype, front & backend code
[redacted]	Back-end feature design, demo video
[redacted]	Back-end feature design, demo video
[redacted]	Back-end feature design, user story

Github repo: [redacted]