

Софийски университет „Свети Климент Охридски“
Факултет по математика и информатика

Тема:
“Bank Management System”

Изготвил: Нина Пламенова Вълкова

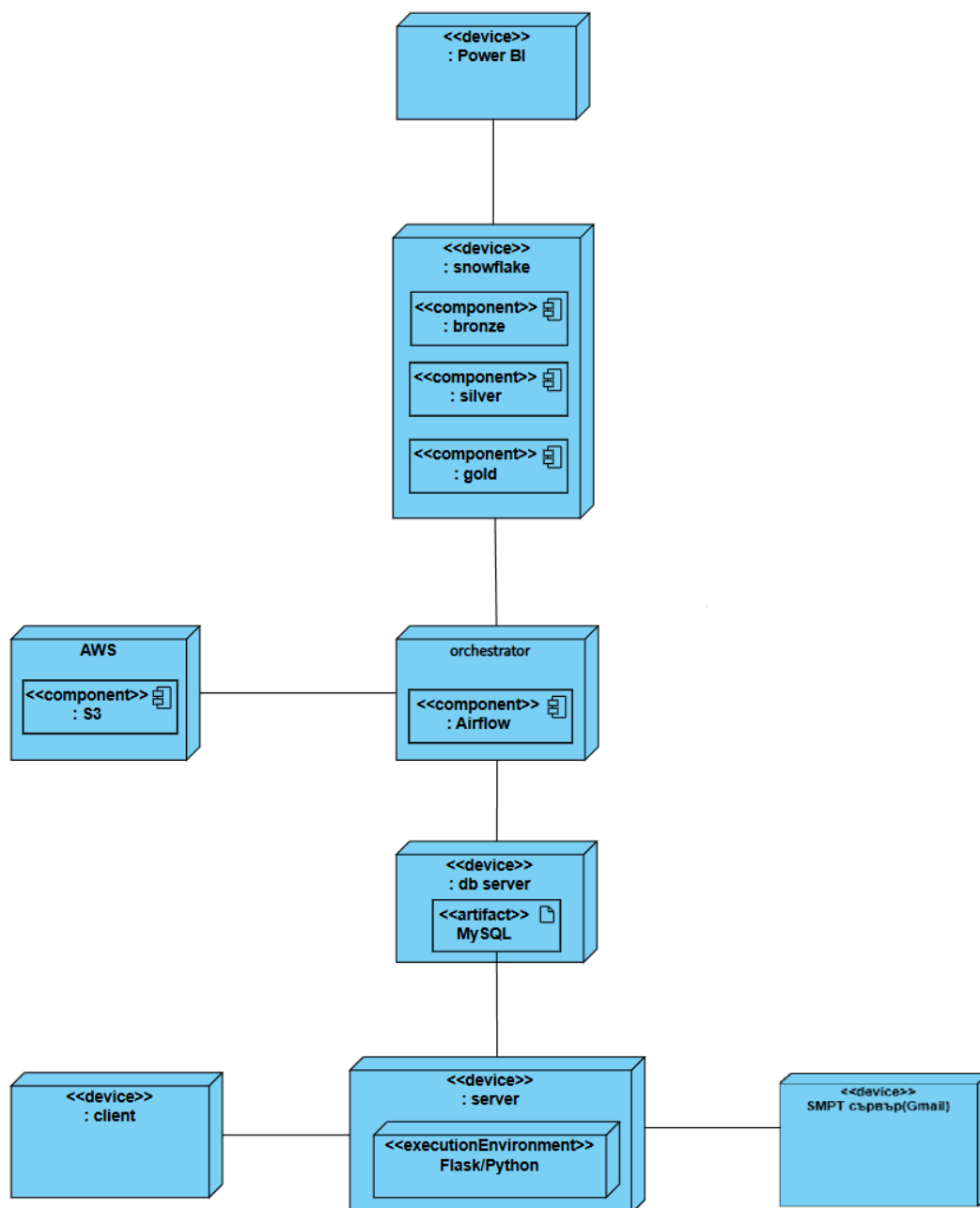
Специалност: Информационни системи

Факултетен номер: 4MI3400863

Съдържание

Софтуерна архитектура	3
Потребителски случай.....	5
База данни.....	7
Хранилище за данни	8
Бизнес анализ	12
Изглед на потребителски интерфейс	13

1. Софтуерна архитектура



Фигура 1: Софтуерна архитектура на система

Системата е разделена на два основни потока:

- Оперативен (OLTP) поток – поддържа банкови операции в реално време (вход, баланс, преводи, депозити, тегления).
- Аналитичен (OLAP) поток – извличане на данни и създаване на модел подходящ за анализ в Power BI.

Клиентът представлява потребителския интерфейс (React приложение).

Сървърът предоставя:

- Flask-RESTful ресурси.
- удостоверяване и оторизация (JWT)
- услуги (например authentication service, transaction service, email service)
- Unit of Work шаблон + Repository
- разпределяне на събития чрез message bus шаблон + handlers (напр. изпращане на email)

Цялата информация се съхранява в MySQL база данни.

Оркестраторът Apache Airflow контролира ETL/ELT процеса. Неговите отговорности са:

- дефиниране на Directed Acyclic Graph (DAG)
- извличане на данни от MySQL
- извличане на данни от S3, където са съхранени файлове (.asc формат)
- зареждане на данните в Snowflake хранилище за данни

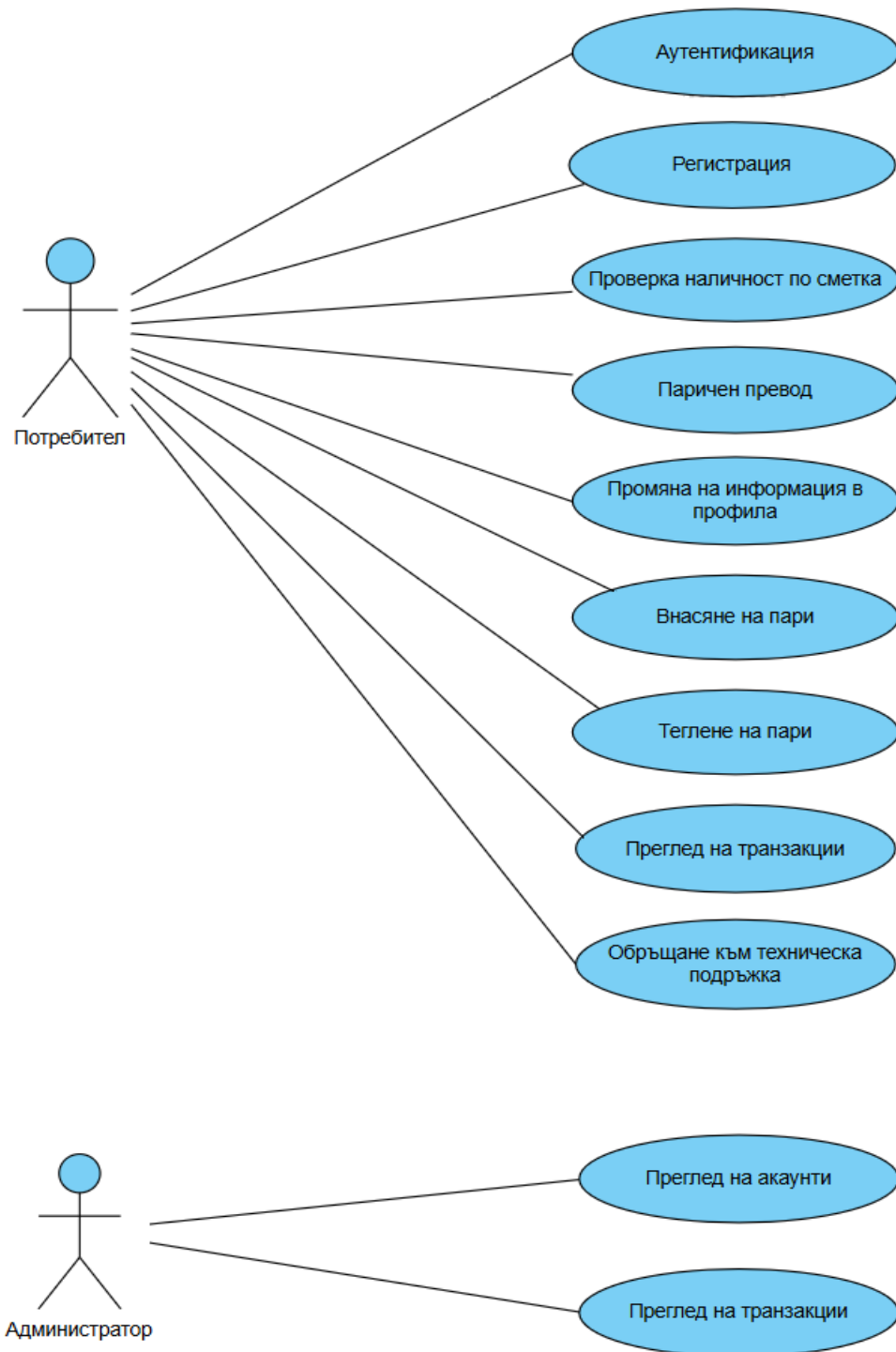
Snowflake е хранилището за данни, където се съхраняват данни.

Трите слоя представляват medallion архитектура:

- Bronze layer:
 - Данните се приемат възможно най-близо от източника (MySQL, S3).
- Silver layer:
 - Данните са валидирани и стандартизирани (типове, липсващи стойности).
- Golden layer:
 - Този слой е оптимизиран за BI инструменти.

Power BI се свързва със Snowflake и предоставя dashboard

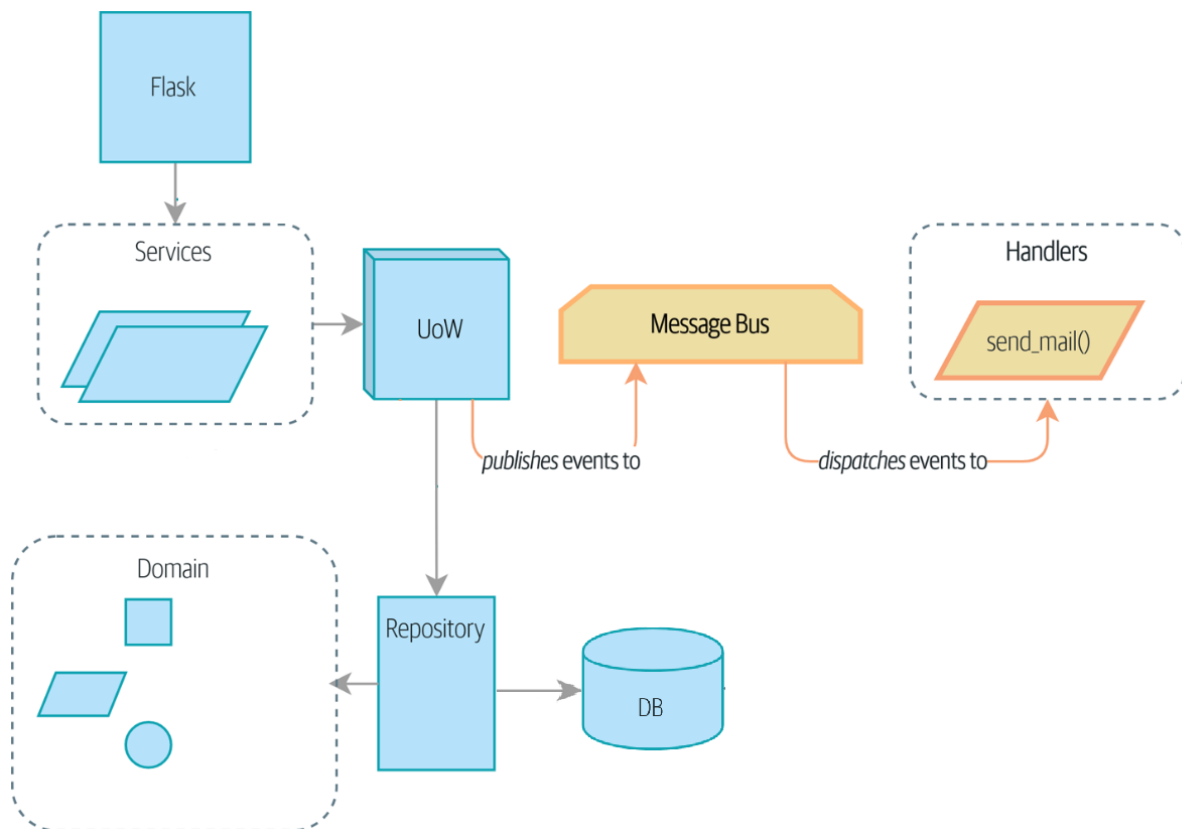
2. Потребителски случай



Фигура 2: Диаграма на потребителските случай

Системата поддържа две основни роли: потребител и администратор. След аутентификация потребителят извършва банкови операции като проверка на баланса по сметката, парични преводи, депозити, тегления, промяна на информацията в профила, преглед на история на транзакциите, обръщане към техническа поддръжка. За дейности парични преводи, депозити, тегления потребителят получава известие по email за успешно извършена операция. Администраторът се удостоверява отделно и има права за преглед на данни (потребители, акаунти, транзакции).

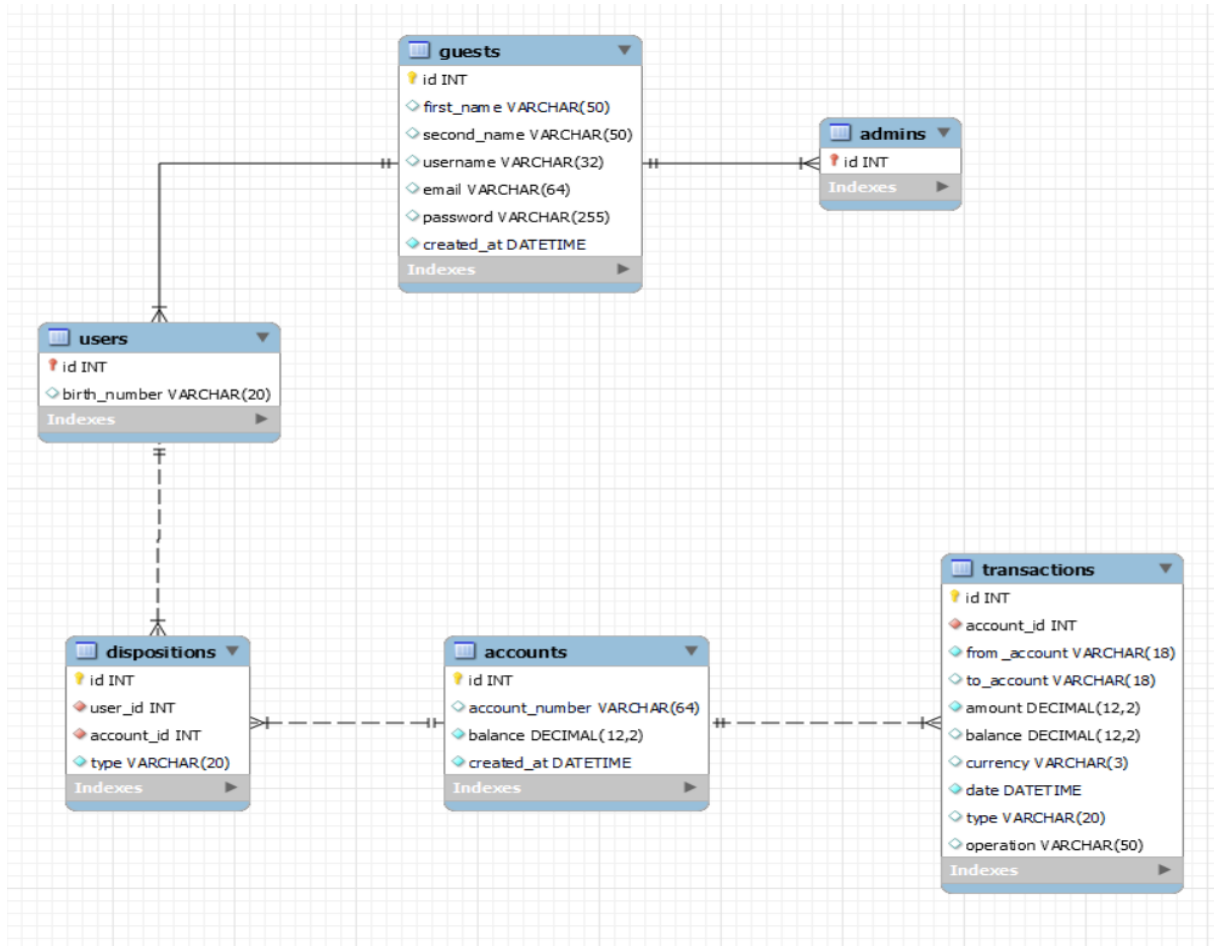
Уеб сървърът е реализиран с помощта на Flask и служи като входна точка на приложението. Той е отговорен за обработката на HTTP заявки. Message bus е система за публикуване/абониране. Handlers са абонирани да получават events, които публикуваме в шината. Events – факти за неща, които са се случили. Използвана е шина за съобщения, за отговор на събитията.



Фигура 3: Диаграма на потока

3. База данни

Цялата информация се съхранява в MySQL база данни.
Фигура 4 показва физически модел на базата данни.



Фигура 4: Физически модел на база данни

4. Хранилище за данни

Използван е Snowflake като облачна платформа за съхранение, обработка и анализ на структурирани данни.

Създаване на Star schema modeling по Кимбъл в Snowflake за набор от данни, използвайки слоеве Bronze, Silver, Golden.

Bronze layer - сурови таблици, заредени са като CSV, таблици на MySQL/приложение с база данни.

Silver layer - почистване, стандартизиране на данни, коригиране на типове данни (дати, числа), изписване с главни букви на кодове, конвертиране на празни низове в NULL, обработка на невалидни стойности, осигуряване на FK целостност.

Golden layer - създаване на таблици с измерения и факти (DateDim, DistrictDim, AccountDim, CustomerDim, CardDim, KSymbolDim, BankDim, OperationDim, TransTypeDim, LoanStatusDim; факти: TransactionFact, LoanFact, OrderFact, FactBalance; плюс Bridge_AccountCustomer).

Зареждане на Golden с правилни сурогатни ключове, FK взаимоотношения и логика isCurrent/StartDate/EndDate.

Осигуряване на достъп за бизнес анализ (BI), използвайки специална POWERBI_ROLE.

Всичко е изпълнявано с Docker + Airflow compose stack (Airflow услуги + Postgres база данни).

Избраният подход е Bronze, Silver, Golden layers с модел на Кимбъл (dim/fact) защото се постига разделение на отговорностите (Bronze/Silver/Golden) Star schema по Кимбъл е идеална за dashboards (Power BI) защото оптимизира производителността и подобрява използваемостта, като организира данните в dimension tables, fact tables.

Fact table съхраняват numerical measurements (сума, баланс, плащания)

Dimension tables съхраняват описателен контекст (дата, сметка, клиент, област, карта, банка, операция, вид транзакция). Dimension tables имат StartDate / EndDate / isCurrent по модела на Кимбъл - Kimball pattern for Slowly Changing Dimensions (Type 2).

В случая е подходящ защото банковите отчети са предимно базирани на агрегиране: общи суми, бройки, тенденции, разбивки по категории.

Прочетени са таблици с банкови данни (ACCOUNT, DISP, TRANSACTION и др.) и таблици с база данни на приложения (DATABASE_ACCOUNTS, DATABASE_USERS, DATABASE_GUESTS, DATABASE_TRANSACTIONS) – където в Silver layer е правилното място за обединяване на източници и дефиниране на приоритета – рождената дата на клиент с id 1 на приложението не

съответства с тази от файлове – за приоритет се приема тази от приложението Фигура 28. По този начин запазва Golden layer стабилен дори ако източник изчезне или се промени. Bronze layer е проектиран като област за приемане на сурови данни и служи като исторически архив.

Staging Area е областта, където данните пристигат точно както са получени. В проекта staging area е реализирана чрез схемата BRONZE_LAYER. Таблиците съхраняват данните, когато пристигат от изходните системи.

BRONZE_LAYER.DISTRICT (A1...A16) - съдържа данни за области/региони.

Оригиналният набор от данни използва имена на колони A1...A16, чиито имена са запазени в staging area.

BRONZE_LAYER.ACCOUNT (account_id, district_id, frequency, date)

BRONZE_LAYER.CLIENT (client_id, birth_number, district_id)

BRONZE_LAYER.DISP (disp_id, client_id, account_id, type)

BRONZE_LAYER.CARD (card_id, disp_id, type, issued)

BRONZE_LAYER.LOAN (loan_id, account_id, date, amount, duration, payments, status)

BRONZE_LAYER.ORDER (order_id, account_id, bank_to, account_to, amount, k_symbol)

BRONZE_LAYER.TRANSACTION (trans_id, account_id, date, type, operation, amount, balance, k_symbol, bank, account)

BRONZE_LAYER.DATABASE_ACCOUNTS (id, account_number, balance, created_at)

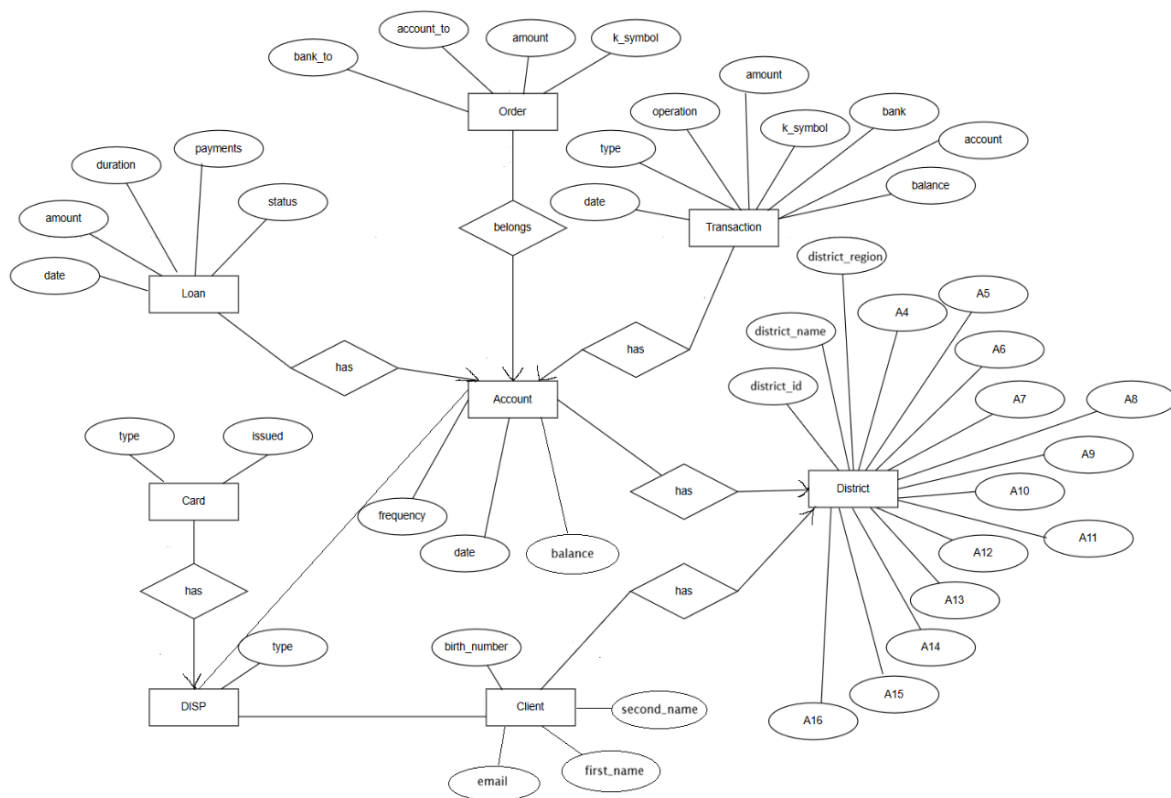
BRONZE_LAYER.DATABASE_GUESTS (id, first_name, second_name, username, email, password, created_at)

BRONZE_LAYER.DATABASE_USERS (id, birth_number)

BRONZE_LAYER.DATABASE_DISPOSITIONS (user_id, account_id, type)

BRONZE_LAYER.DATABASE_TRANSACTIONS (account_id, from_account, to_account, amount, balance, currency, date, type, operation)

В Bronze layer таблиците се съхраняват без ограничения за primary и foreign ключове, тъй като това е staging area. Ограниченията са умишлено отложени, за да се предотвратят грешки при зареждане, причинени от непълни или непоследователни изходни данни. Логическите връзки съществуват чрез колони с идентификатори (напр. account_id, client_id, district_id), но референтната цялост се прилага само след почистване и стандартизация в Silver layer. Фигура 5 показва концептуален модел на хранилището за данни.



Фигура 5: Концептуален модел на хранилището за данни

В Silver layer се извършва почистване, стандартизация, именуване подходящо за бизнеса.

1) Преименуване на колоните

-Наример в таблица DISTRICT

A1 преименована на district_id

A2 преименована на district_name

A3 преименована на district_region

2) Промяна на типове данни (низове към числов тип)

В BRONZE_LAYER.DISTRICT, колони A4–A16 се съхраняват като VARCHAR

Следователно може да съдържат:

-числови низове ('123')

-празни низове

-нечислови стойности ('?', 'n/a', 'unknown')

Извършено е валидиране и преобразуване на всяка стойност от колоните в числов тип

Обработени са нечислови стойности, например вместо '?' е зададена стойност -1

3) Стандартизиране на текстовите стойности

В SILVER_LAYER.TRANSACTION конвертирам от чешки стойности в стандартните английски категории
 PRIJEM преименован на CREDIT
 VYDAJ / VYBER преименован на DEBIT
 VKLAD преименован на CASH_DEPOSIT
 VYBER преименован на CASH_WITHDRAW
 PLATBA KARTOU преименован на CARD_PAYMENT
 При наличие на липсващи стойности като стойности само с празни пространства променя на NULL

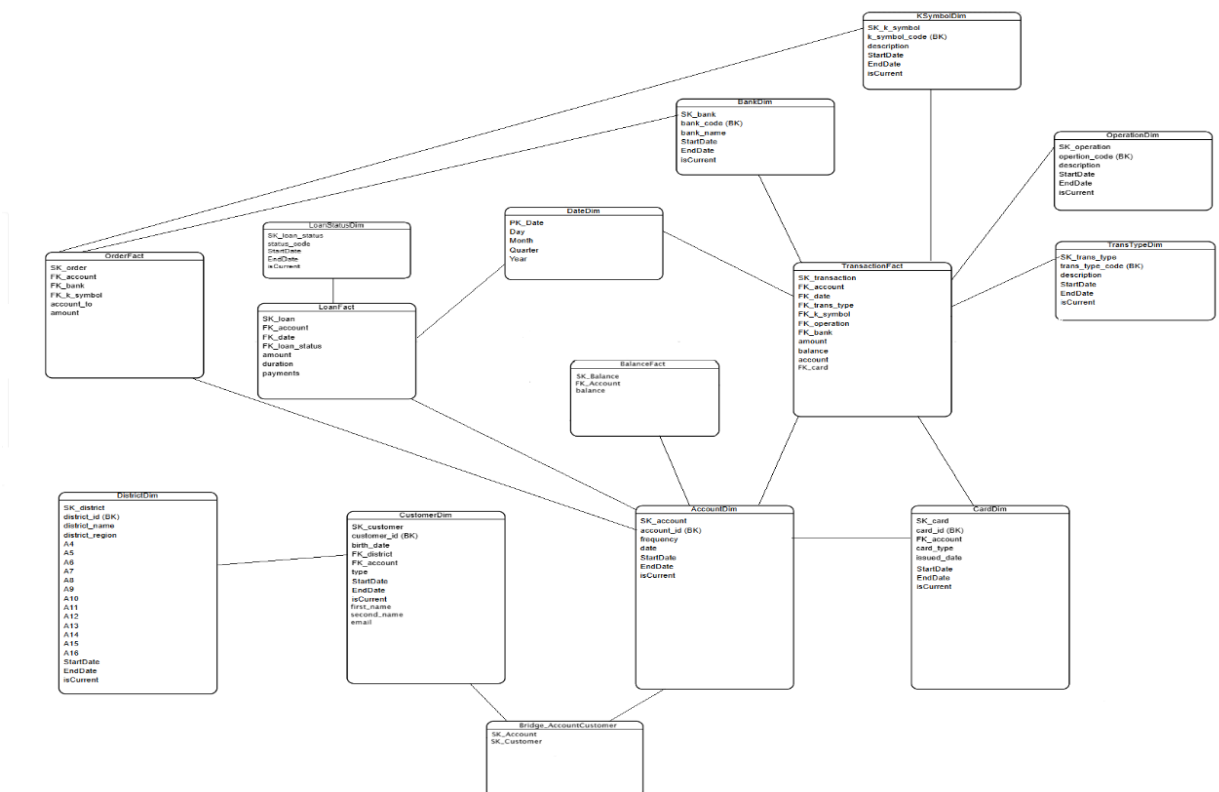
Golden Layer следва принципите на Ralph Kimball's fact table съхраняват numerical measurements (сума, баланс, плащания) и външни ключове към dimensions

В модела fact tables са TransactionFact, LoanFact, OrderFact, BalanceFact

Таблиците с измерения (dimension tables) съдържа описателни атрибути, в модела са DateDim, AccountDim, CustomerDim, DistrictDim, CardDim, BankDim, KsymbolDim, OperationDim, TransTypeDim, LoanStatusDim

Използвана е Bridge table е таблица за връзката много-към-много в случая на account/customer - един account може да има много customers и един customer може да има много accounts

Модела използва Star schema и Snowflaked dimensions в случая на District и Customer. Фигура 6 показва логически модел на хранилището за данни.



Фигура 6: Логически модел на хранилището за данни

Бизнес анализ

Power BI е свързан с Snowflake и предоставя dashboard.

Total customer – общ брой клиенти

Total account balance – общо пари по всички сметки

Loan Amount & Loan Count by Loan Status – общ брой заеми в зависимост от “status” на loan. Общо пари за всеки “status” на loan.

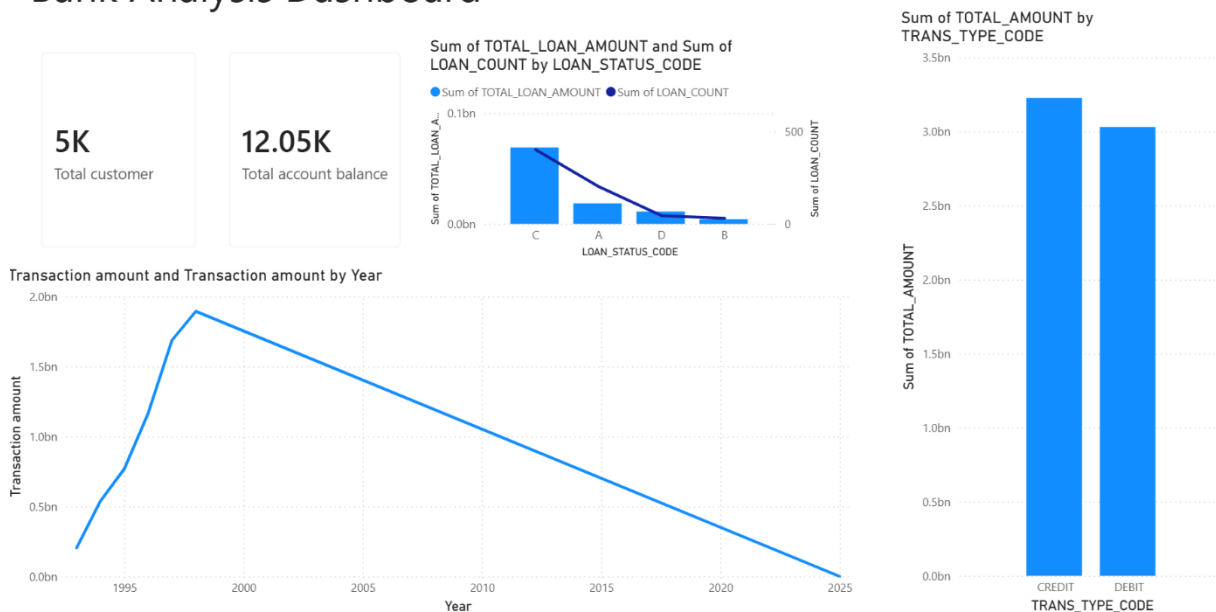
Transaction Amount by Transaction Type - общо пари в зависимост от типа транзакция CREDIT (внесени пари) е DEBIT (изтеглени пари)

Transaction Amount by Year - общо обменени пари от транзакции за година

Заради защитата на Snowflake, за да може Power BI да прочете данните е създадена роля с командата CREATE ROLE IF NOT EXISTS POWERBI_ROLE;

Фигура 7 показва бизнес анализ.

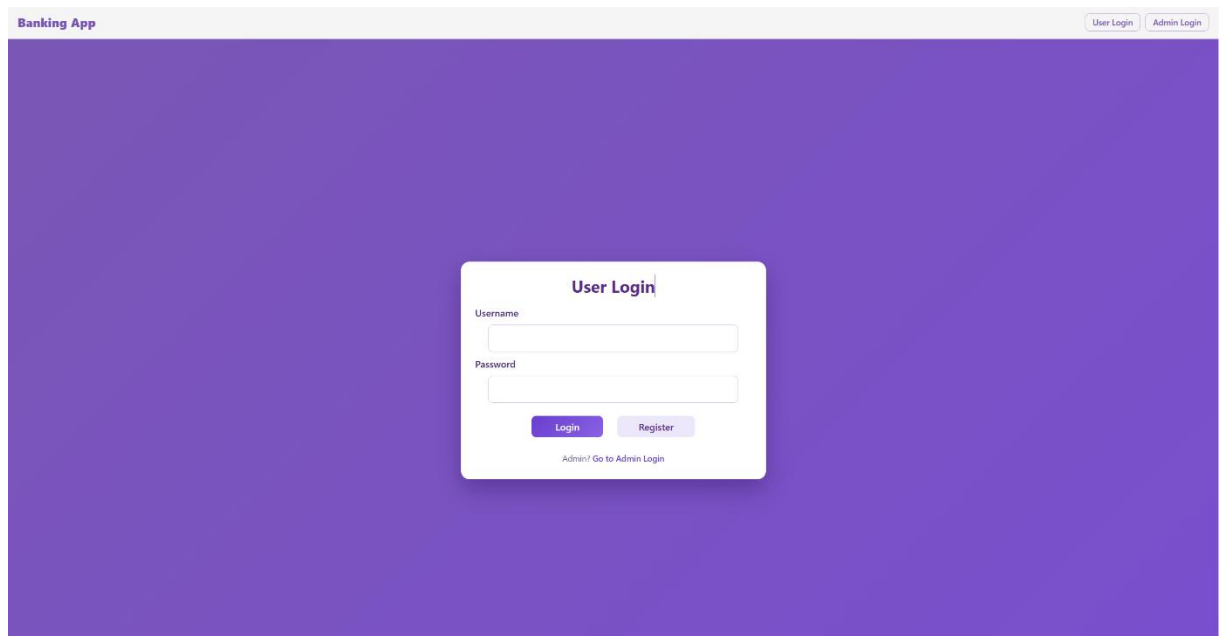
Bank Analysis Dashboard



Фигура 7: Бизнес анализ

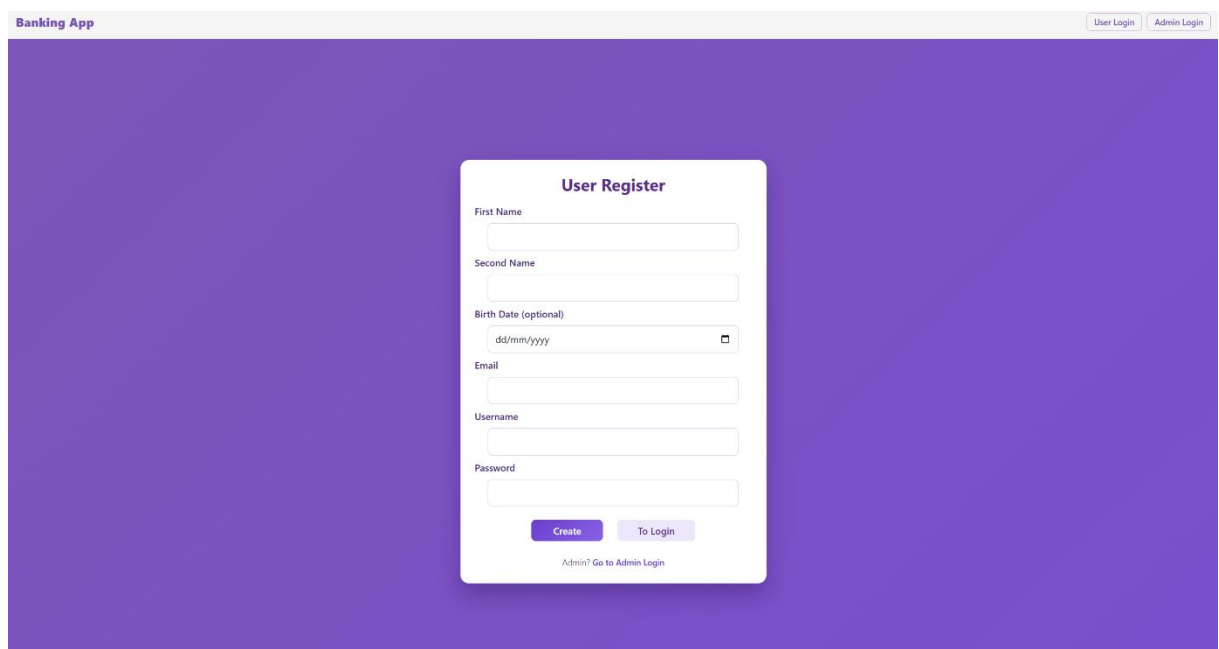
5. Изглед на потребителски интерфейс

Когато уеб приложението се стартира се показва начална страница – страница за вход . Фигура 8, фигура 9, фигура 10 показвт страниците свързани с вход и регистрация.



The image shows a web browser window titled "Banking App". In the top right corner, there are two buttons: "User Login" and "Admin Login". The main content area has a solid purple background. Centered on the page is a white rectangular form titled "User Login". Inside the form, there are two input fields: "Username" and "Password". Below these fields are two buttons: "Login" (purple) and "Register" (light purple). At the bottom of the form, there is a link that says "Admin? Go to Admin Login".

Фигура 8: Потребителски вход



The image shows a web browser window titled "Banking App". In the top right corner, there are two buttons: "User Login" and "Admin Login". The main content area has a solid purple background. Centered on the page is a white rectangular form titled "User Register". Inside the form, there are several input fields: "First Name", "Second Name", "Birth Date (optional)" (with a date picker showing "dd/mm/yyyy"), "Email", "Username", and "Password". Below these fields are two buttons: "Create" (purple) and "To Login" (light purple). At the bottom of the form, there is a link that says "Admin? Go to Admin Login".

Фигура 9: Регистрация на потребител

Banking App User Login Admin Login

Admin Login

Username

Password

Login

[User? Go to User Login](#)

Фигура 10: Вход за администратор

След успешен вход се отваря началната страница – Фигура 11, Фигура 12

Banking App Logout

Admin Dashboard

ID	First	Second	Username	Email	Birth	Created
5	Nina	Valkova	nina1	nina13valkova@gmail.com	2001-04-19	2026-01-05T22:54:44
4	Ivan	Ivanov	ivan1	ivan@test.com	2009-06-11	2026-01-05T22:53:33
3	Tommy	Lie	tommy1	tommylie@test.com	2001-05-18	2026-01-05T22:52:43
2	John	Doe	john1	john1@test.com	2000-01-01	2026-01-05T22:50:33

ID	Account Number	Balance	Created
12	NL01AAZ800000000012	1000	2026-01-05T22:44:55
11	NL01AAZ800000000011	1000	2026-01-05T22:44:55
10	NL01AAZ800000000010	1000	2026-01-05T22:44:55
9	NL01AAZ800000000009	1000	2026-01-05T22:44:55
8	NL01AAZ800000000008	1000	2026-01-05T22:44:55
7	NL01AAZ800000000007	1000	2026-01-05T22:44:55
6	NL01AAZ800000000006	1000	2026-01-05T22:44:55
5	NL01AAZ800000000005	1000	2026-01-05T22:44:55
4	NL01AAZ800000000004	900	2026-01-05T22:44:55
3	NL01AAZ800000000003	700	2026-01-05T22:44:55
2	NL01AAZ800000000002	1000	2026-01-05T22:44:55
1	NL01AAZ800000000001	1120	2026-01-05T22:44:55

ID	Account ID	From	To	Amount	Balance	Currency	Type	Operation	Date
7	4	NL01AAZ800000000004		100	900	EUR	DEBIT	CASH_WITHDRAW	2026-01-28T18:50:50
6	4	NL01AAZ800000000004		200	1000	EUR	CREDIT	CASH_DEPOSIT	2026-01-28T18:49:36

Фигура 11: Начална страница - администратор

Banking App Logout

BALANCE INQUIRY

TRANSFER

USER INFORMATION

UPDATE DETAILS

Welcome, Nina!

CASH DEPOSIT

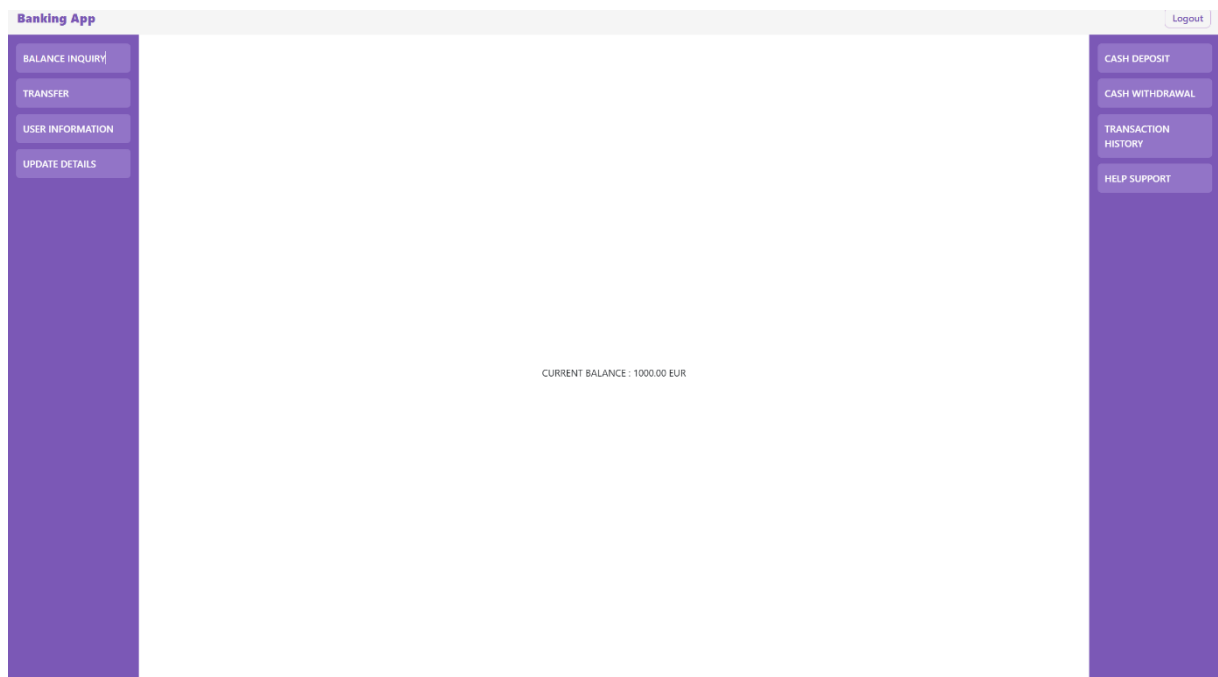
CASH WITHDRAWAL

TRANSACTION HISTORY

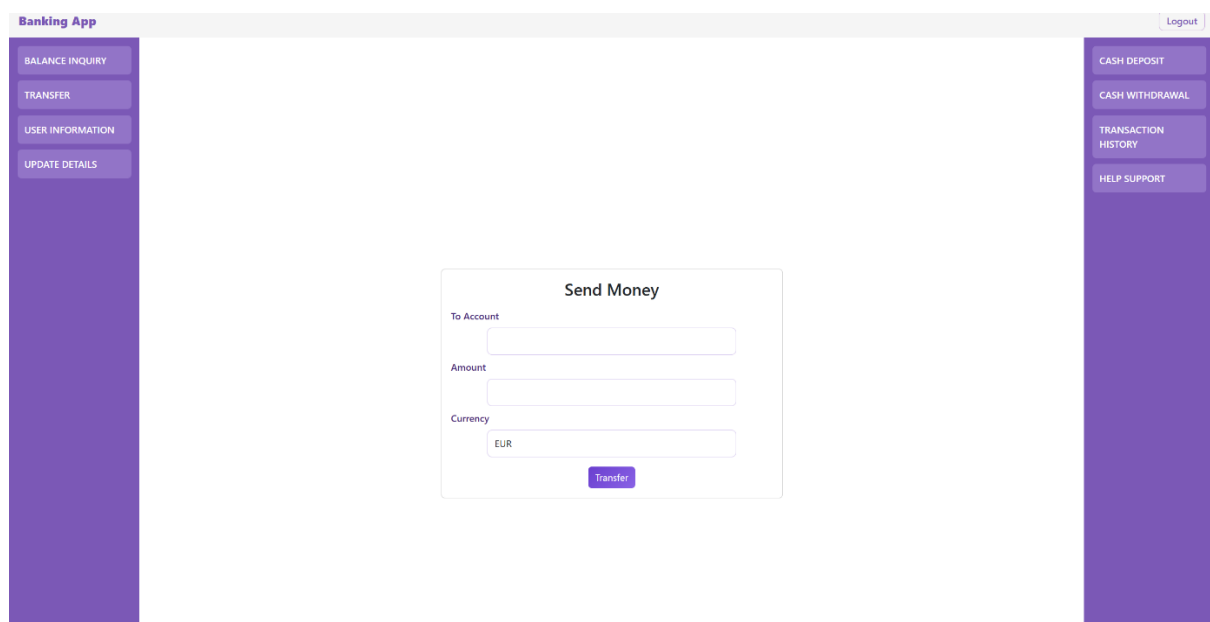
HELP SUPPORT

Фигура 12: Начална страница - потребител

Фигура 12, фигура 13, фигура 14, фигура 15, фигура 16, фигура 17, фигура 18, фигура 19 показват операциите в системата



Фигура 12: Баланс



Фигура 13: Трнсфер на сума

Banking App

Logout

BALANCE INQUIRY

TRANSFER

USER INFORMATION

UPDATE DETAILS

CASH DEPOSIT

CASH WITHDRAWAL

TRANSACTION HISTORY

HELP SUPPORT

User Information

First Name: Nina

Second Name: Valkova

Email: nina13valkova@gmail.com

Birth Number: 2001-04-19

Balance: 1000.00

Фигура 14: Информация за потребител

Banking App

Logout

BALANCE INQUIRY

TRANSFER

USER INFORMATION

UPDATE DETAILS

CASH DEPOSIT

CASH WITHDRAWAL

TRANSACTION HISTORY

HELP SUPPORT

Update User Information

First Name

Nina

Second Name

Valkova

Email

nina13valkova@gmail.com

Birth Number

2001-04-19

Update

Фигура 15: Промяна на информация за потребител

Banking App

Logout

BALANCE INQUIRY

TRANSFER

USER INFORMATION

UPDATE DETAILS

CASH DEPOSIT

CASH WITHDRAWAL

TRANSACTION HISTORY

HELP SUPPORT

Cash Deposit

Amount

Deposit

Фигура 16: Внасяне на пари

Banking App

Logout

BALANCE INQUIRY

TRANSFER

USER INFORMATION

UPDATE DETAILS

CASH DEPOSIT

CASH WITHDRAWAL

TRANSACTION HISTORY

HELP SUPPORT

Cash Withdrawal

Amount

Withdraw

Фигура 17: Изтегляне на пари

Banking App

BALANCE INQUIRY

TRANSFER

USER INFORMATION

UPDATE DETAILS

Logout

CASH DEPOSIT

CASH WITHDRAWAL

TRANSACTION HISTORY

HELP SUPPORT

Transaction History

Date: Newest → Oldest

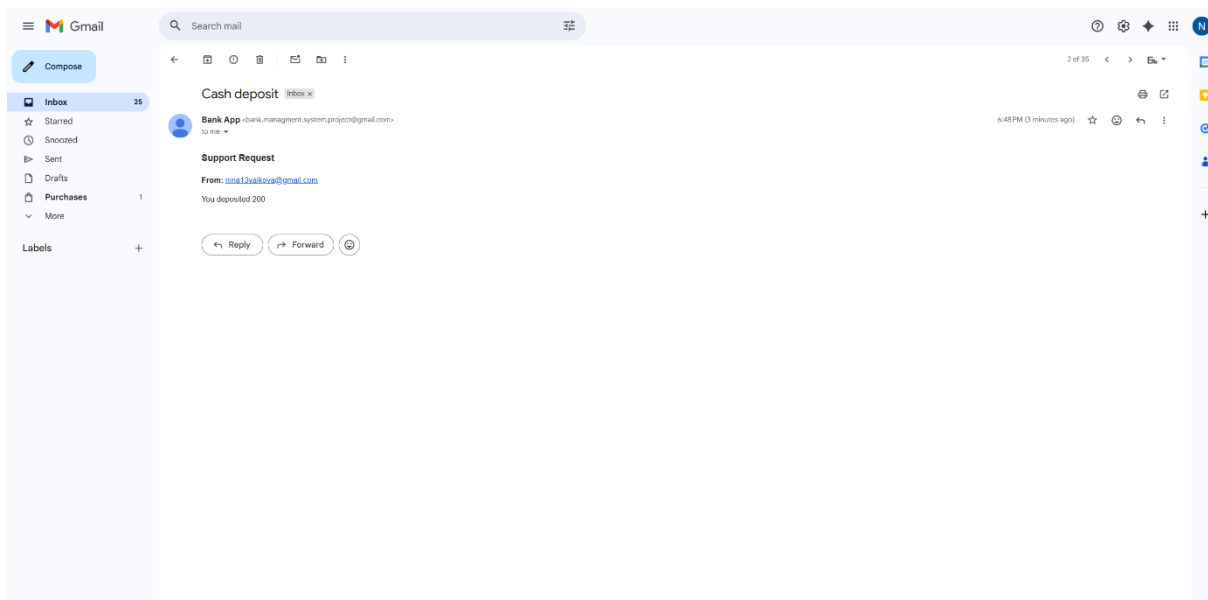
Date	Operation	Type	From	To	Amount	Balance
1/28/2026, 6:50:50 PM	CASH_WITHDRAW	DEBIT	NL01AAZB00000000004	-	100 EUR	900
1/28/2026, 6:49:36 PM	CASH_DEPOSIT	CREDIT	NL01AAZB00000000004	-	200 EUR	1000
1/24/2026, 2:16:27 PM	CASH_WITHDRAW	DEBIT	NL01AAZB00000000004	-	40 EUR	800
1/5/2026, 10:56:12 PM	TRANSFER	DEBIT	NL01AAZB00000000004	NL01AAZB00000000001	120 EUR	840
1/5/2026, 10:55:46 PM	CASH_DEPOSIT	CREDIT	NL01AAZB00000000004	-	80 EUR	960
1/5/2026, 10:55:33 PM	CASH_WITHDRAW	DEBIT	NL01AAZB00000000004	-	120 EUR	880

Фигура 18: История

Banking App							Logout
BALANCE INQUIRY						CASH DEPOSIT	
TRANSFER						CASH WITHDRAWAL	
USER INFORMATION						TRANSACTION HISTORY	
UPDATE DETAILS						HELP SUPPORT	
<div> <div>Help & Support</div> <div> <div>Your message</div> <div></div> <div>Send</div> </div> </div>							

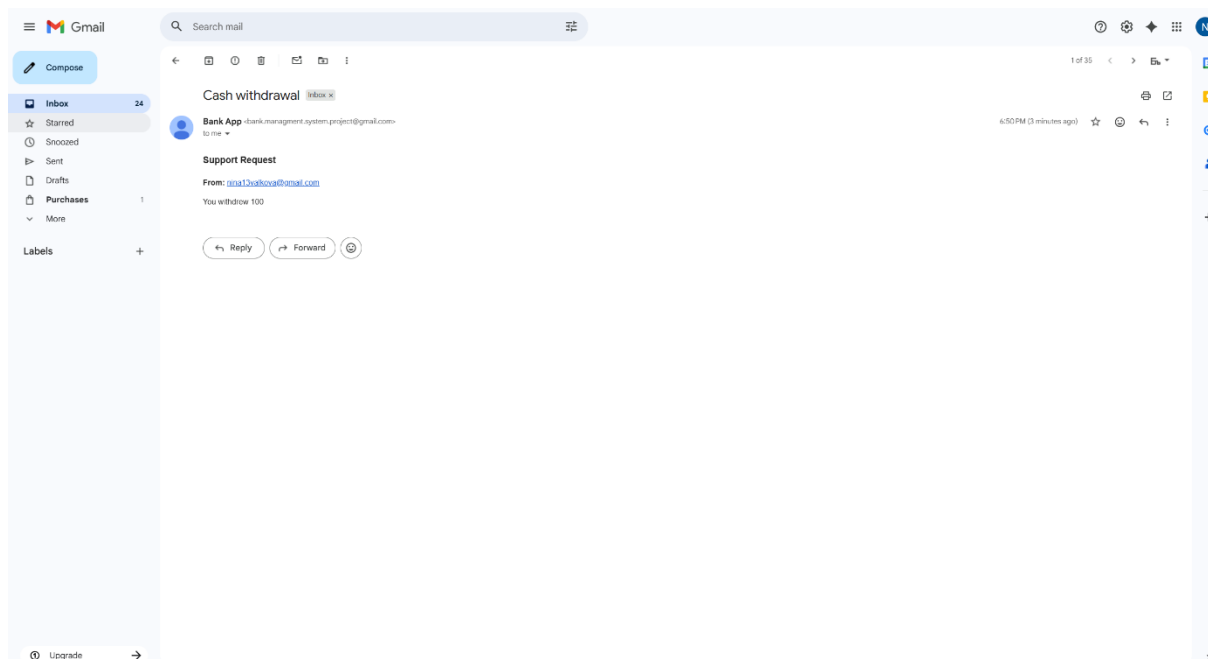
Фигура 19: Обръщане към поддръжка

При операция внасяне на пари потребителят получава известие по email за успешно извършено действие – Фигура 20



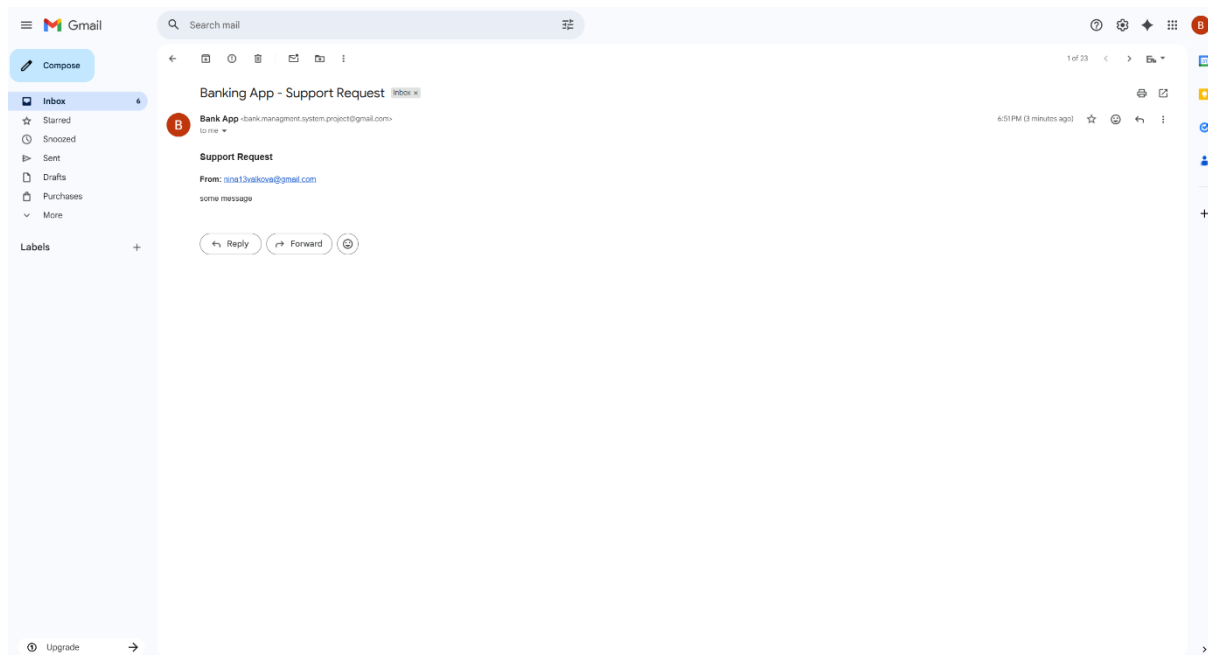
Фигура 20: Внасяне на пари - email известие

При операция теглене на пари потребителят получава известие по email за успешно извършено действие – Фигура 21



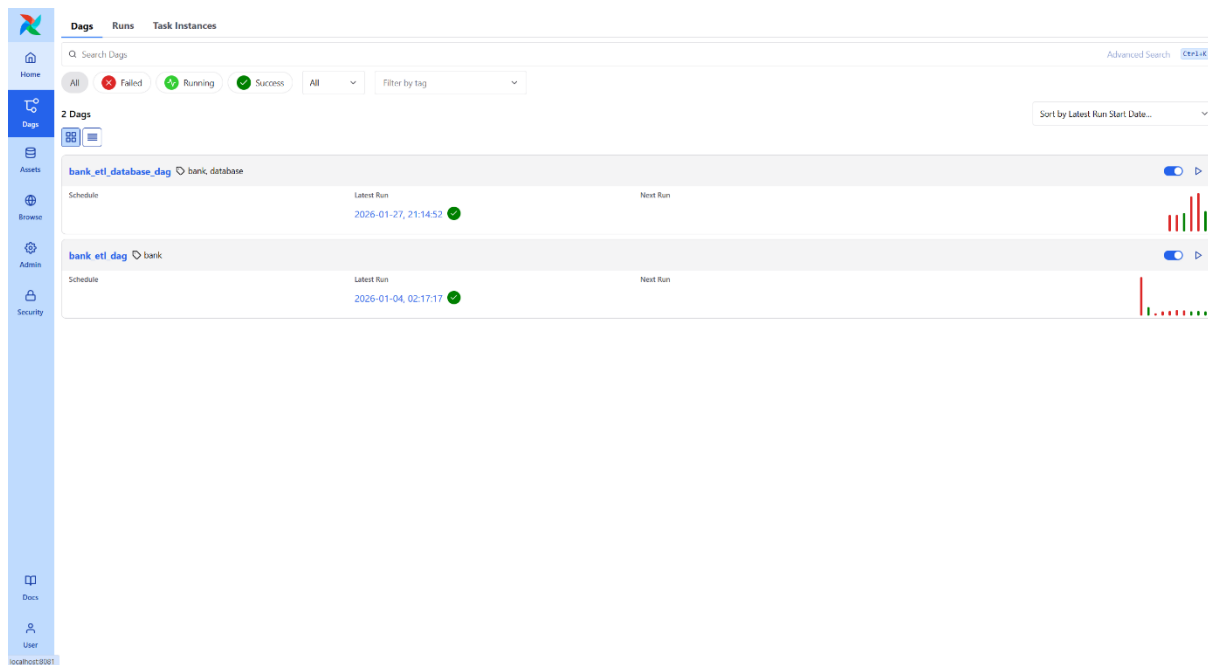
Фигура 21: Изтегляне на пари - email известие

При операция обръщане към поддръжка известие по email се получава на email на поддръжка – Фигура 22

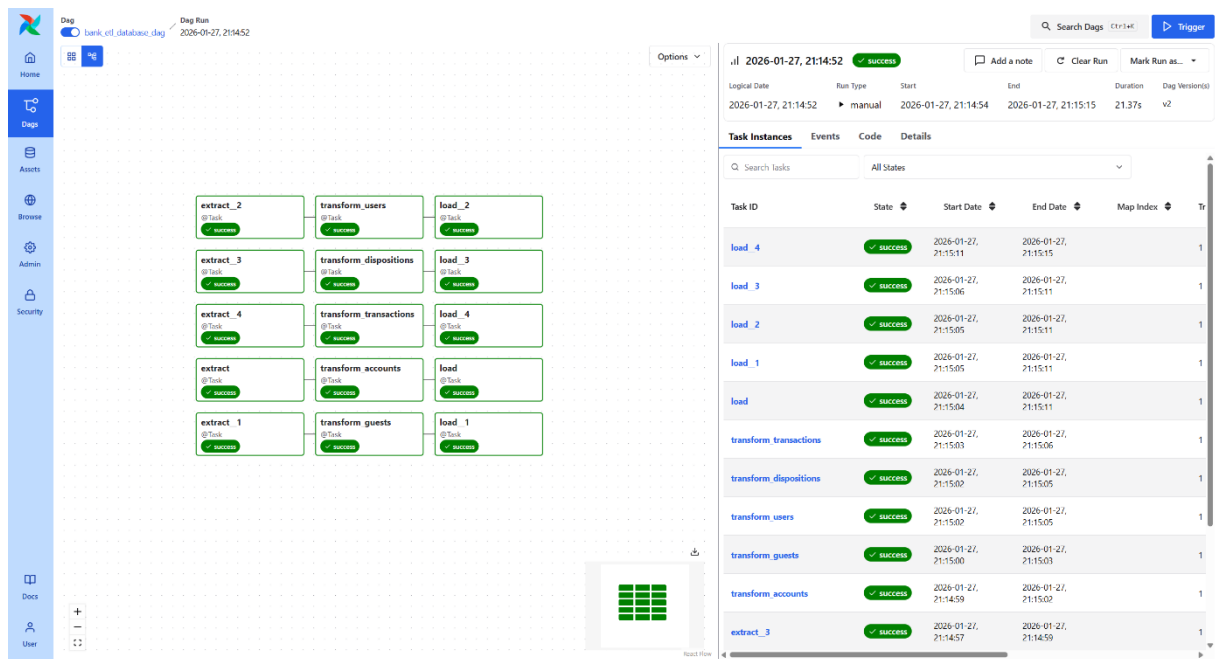


Фигура 22: Обръщане към поддръжка - email

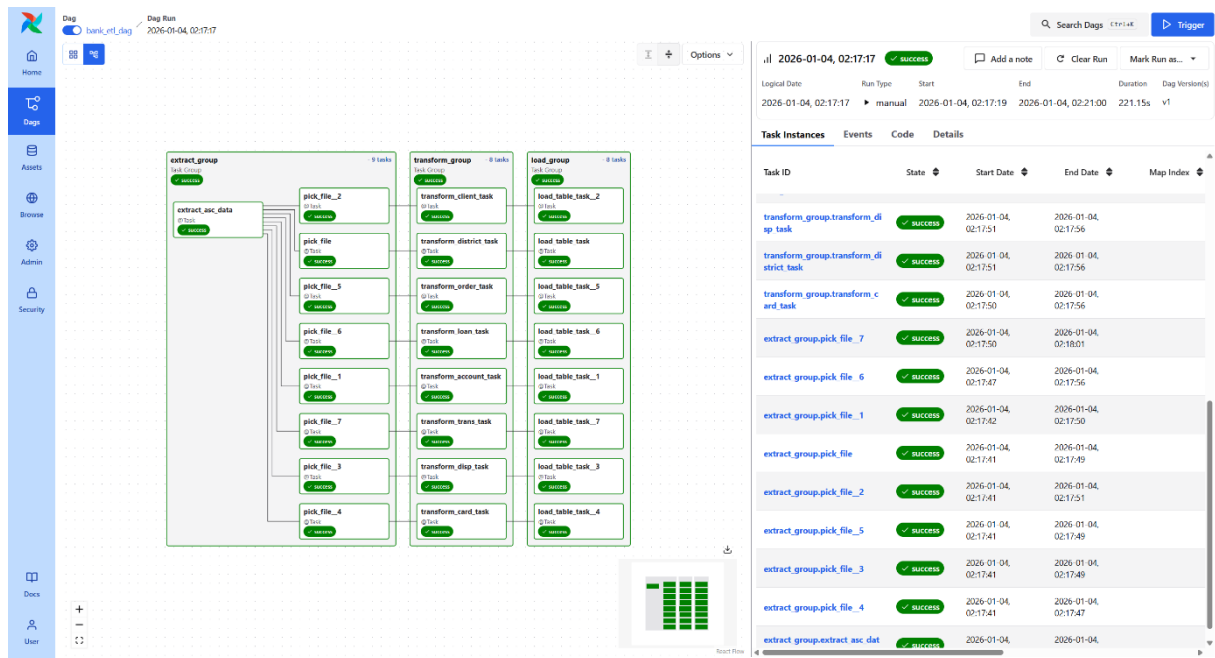
Фигура 23, фигура 24, фигура 25 показват оркестратор Apache AirFlow



Фигура 23: Apache AirFlow DAGs

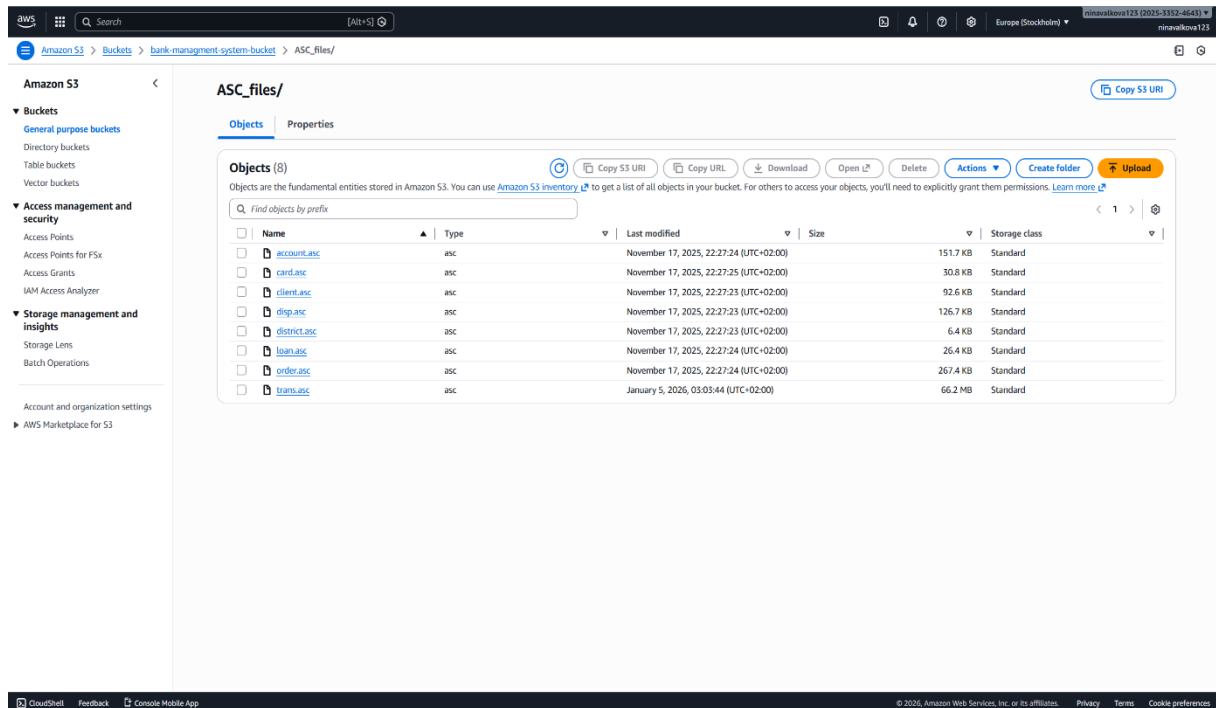


Фигура 24: Apache AirFlow Dag – bank_etl_database_dag



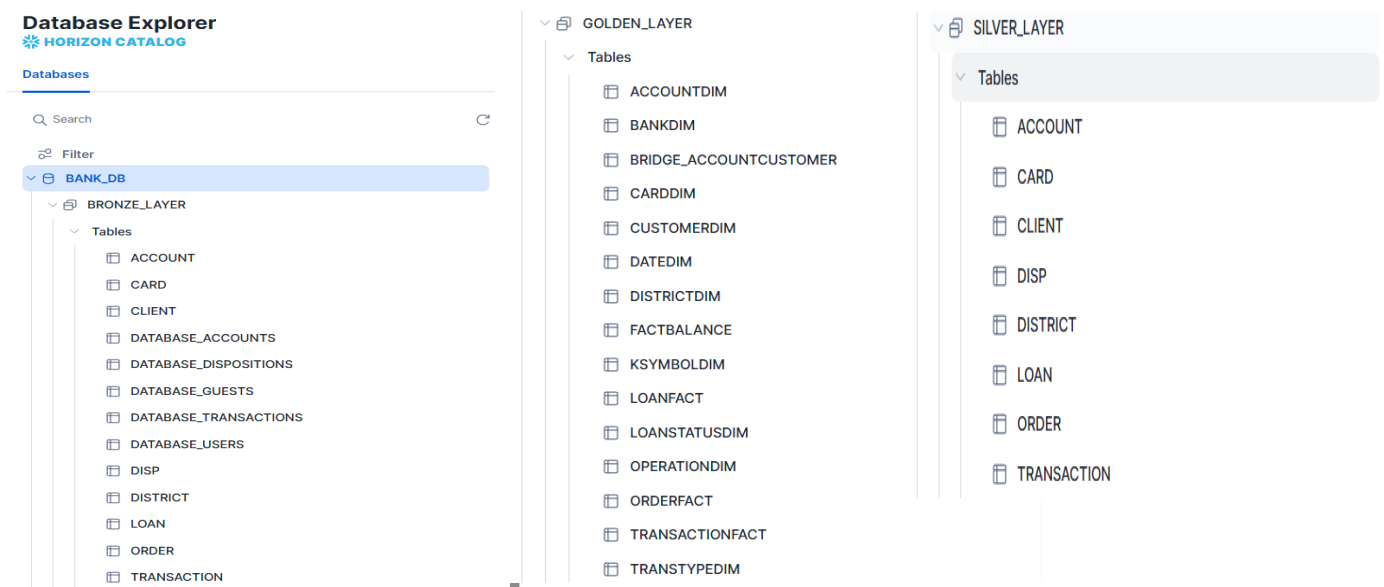
Фигура 25: Apache AirFlow Dag – bank_etl_dag

Файлове (.asc формат) са съхранени в AWS S3 - Фигура 26



Фигура 26: AWS S3

В Snowflake е създадено хранилище за данни BANK_WH и база данни BANK_DB с BANK_DB.BRONZE_LAYER, BANK_DB.SILVER_LAYER, BANK_DB.GOLDEN_LAYER - Фигура 27



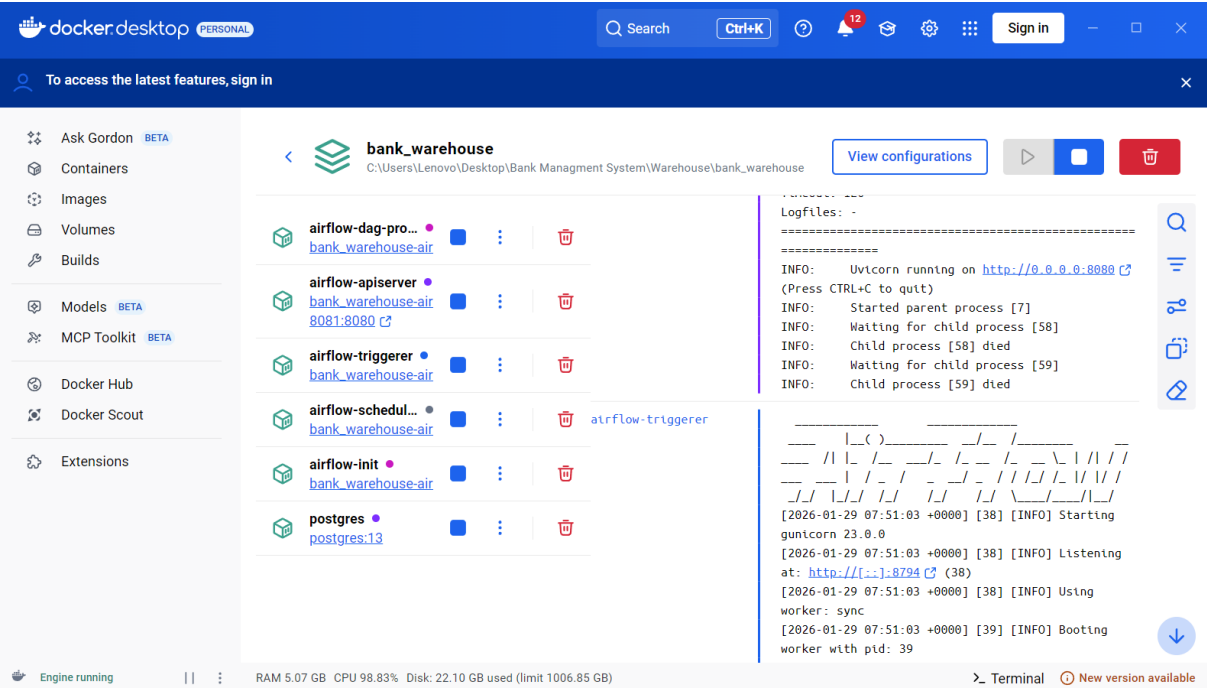
Фигура 27: BANK_DB schemas

Съхранени са данни от двата източника. Пример Фигура 28 представя BANK_DB.SILVER_LAYER.CLIENT

Results (just now)						
Table Chart						
	# CLIENT_ID	Δ FIRST_NAME	Δ SECOND_NAME	Δ EMAIL	Δ BIRTH_NUMBER	# DISTRICT_ID
1		1	null	null	1970-12-13 00:00:00.000	18
2		2	John	Doe	2000-01-01 00:00:00.000	1
3		3	Tommy	Lie	2001-05-18 00:00:00.000	1
4		4	Ivan	Ivanov	2009-06-11 00:00:00.000	5
5		5	Nina	Valkova	2001-04-19 00:00:00.000	5
6		6	null	null	1919-09-22 00:00:00.000	12
7		7	null	null	1929-01-25 00:00:00.000	15
8		8	null	null	1938-02-21 00:00:00.000	51
9		9	null	null	1935-10-16 00:00:00.000	60
10		10	null	null	1943-05-01 00:00:00.000	57
11		11	null	null	1950-08-22 00:00:00.000	57
12		12	null	null	1981-02-20 00:00:00.000	40
13		13	null	null	1974-05-29 00:00:00.000	54
14		14	null	null	1942-06-22 00:00:00.000	76
15		15	null	null	1918-06-28 00:00:00.000	21
16		16	null	null	1919-02-25 00:00:00.000	21
17		17	null	null	1934-10-13 00:00:00.000	76
18		18	null	null	1931-04-05 00:00:00.000	76
19		19	null	null	1942-12-28 00:00:00.000	47
20		20	null	null	1979-01-04 00:00:00.000	46
21		21	null	null	1952-10-29 00:00:00.000	12
22		22	null	null	1969-10-11 00:00:00.000	1
23		23	null	null	1973-05-29 00:00:00.000	1
24		24	null	null	1939-07-29 00:00:00.000	43
25		25	null	null	1939-04-23 00:00:00.000	21
26		26	null	null	1969-04-20 00:00:00.000	74
27		27	null	null	1986-03-26 00:00:00.000	54
28		28	null	null	1945-09-29 00:00:00.000	1
29		29	null	null	1951-09-11 00:00:00.000	30

Фигура 28: BANK_DB.SILVER_LAYER.CLIENT

Използван е Docker за изпълнение на Apache Airflow среда– без да се инсталира Airflow, Postgres и всички зависимости на компютъра - Фигура 29



Фигура 29: Docker