

# ATM MANAGEMENT SYSTEM

PRESENTED BY: ENGR. MUHAMMAD MANSOOR



## PROBLEM STATEMENT:

- THE CURRENT ATM MANAGEMENT SYSTEM HAS SEVERAL ISSUES SUCH AS LONG QUEUES, SLOW PROCESSING, FREQUENT DOWNTIME, AND LIMITED FUNCTIONALITIES. THESE ISSUES HAVE RESULTED IN POOR CUSTOMER EXPERIENCE AND LOSS OF BUSINESS FOR THE BANKS.
- THE GOAL OF THIS PROJECT IS TO DEVELOP A NEW ATM MANAGEMENT SYSTEM THAT ADDRESSES THESE ISSUES AND PROVIDES AN IMPROVED CUSTOMER EXPERIENCE. THE SYSTEM SHOULD BE ABLE TO HANDLE A LARGE NUMBER OF TRANSACTIONS QUICKLY AND EFFICIENTLY, PROVIDE A WIDE RANGE OF FUNCTIONALITIES, AND ENSURE HIGH UPTIME AND AVAILABILITY.
- THE SYSTEM SHOULD ALSO INCLUDE FEATURES SUCH AS ADVANCED SECURITY MEASURES TO PREVENT FRAUDULENT ACTIVITIES, USER-FRIENDLY INTERFACES TO ENSURE EASE OF USE FOR CUSTOMERS, AND REAL-TIME MONITORING TO ENABLE QUICK RESPONSE TO ANY ISSUES.
- THE PROJECT WILL INVOLVE DESIGNING AND DEVELOPING THE NEW SYSTEM, INTEGRATING IT WITH THE EXISTING BANKING SYSTEMS, AND ENSURING ITS SEAMLESS OPERATION. THE SYSTEM SHOULD ALSO BE SCALABLE AND ADAPTABLE TO ACCOMMODATE FUTURE CHANGES AND ADVANCEMENTS IN TECHNOLOGY.

# SCOPE OF WORK:

<b>Release I</b> • Selection of Functionalities	<b>Release II</b> • Selection of User Stories	<b>Release III</b> • Implementation of selected functionalities using assigned tools	<b>Release IV</b> • Selection of Test cases and their implementation.
--	--	---	--

# PROJECT STRUCTURE:

Sr.No	Roles	Name
1	Engagement Manager	Jawad Naeem
2	Project Manager	Muhammad Ateeb
3	Software Engineer	Muhammad Mansoor
4	Environment	Contour Software

# STAKE HOLDERS (PRIMARY):

Sr.	Primary
1	Project Manager
2	Business Analyst
3	Developers
4	Quality Assurance Engineers
5	UI Designers
6	UX Designers
7	INTERNET

# STAKE HOLDERS (SECONDARY):

Sr.	Secondary
1	Customers
2	Banks
3	State Bank of Pakistan
4	Ministry of Finance
5	Cybersecurity Wings
6	Card Merchants
7	Other Fintech Application Providers

# FUNCTIONALITIES:

Sr. No	Functionality	Sub Functionality
1	Authentication with Login	<ul style="list-style-type: none"><li>• Card along pin is verified.</li><li>• Biometric Recognition with pin is supported.</li></ul>
2	Cash Withdrawal	<ul style="list-style-type: none"><li>• Cash Withdrawal with assigned limits</li><li>• Free of Cost at same bank ATM</li><li>• With Charges at different bank ATM</li></ul>
3	Balance Inquiry	<ul style="list-style-type: none"><li>• Free of Cost At same bank's ATM</li><li>• With Charges at a different bank ATM</li></ul>
4	Cash Deposit	<ul style="list-style-type: none"><li>• Cash Deposit with assigned limits</li><li>• Free of Cost at same banks ATMS</li></ul>

# FUNCTIONALITIES (CONT):

Sr. No	Functionality	Sub-Functionality
5	Update Pin	<ul style="list-style-type: none"><li>• Pin can be updated on same banks ATM</li></ul>
6	Maintains Transaction Records	<ul style="list-style-type: none"><li>• Transaction Records for each customer are maintained</li></ul>
7	Bill payments	<ul style="list-style-type: none"><li>• Bill Payments for Electricity, Gas , Water Telephone etc</li></ul>
8	Funds Transfer to other Accounts	<ul style="list-style-type: none"><li>• Funds Transfer to other accounts within the bank</li></ul>

# USER STORIES:

Story ID	As	Wants to	So that they can
UT-01	Customer	Log in to ATM using card	Perform a transaction
UT-02	Customer	Check balance	To view their current available balance
UT-03	Customer	Withdraw Cash	Fulfil their needs with directly going to bank 'counter.
UT-04	Customer	Deposit Cash	So amount can be deposited in bank account without specifically going to bank's counter
UT-05	Customer	Update Pin	Replace their previous insecure pin with a new one in a hassle free way.

# USER STORIES (CONT):

Story ID	As	Wants to	So that they can
UT-06	Customer	Pay bills	Avoid long queues
UT-07	Customer	Transfer funds within same bank	Transfer funds to other bank accounts within the same bank and avoid counters and deposit slip issues.
UT-08	Customer	Transfer funds to other bank	Transfer funds to other bank accounts and avoid counters and deposit slip issues.
UT-09	Customer	Login using biometric and pin	Avoid debit cards.
UT-10	Customer	Needs mini bank statement	View their transaction history

# IMPLEMENTED FUNCTIONALITIES:

- LOGIN WITH CREDENTIALS
- BALANCE INQUIRY AT SAME BANK'S ATM
- CASH WITHDRAWAL AT SAME BANK'S ATM
- CASH DEPOSIT AT SAME BRANCH
- UPDATE PIN AT SAME BRANCH
- TRANSACTION LIST

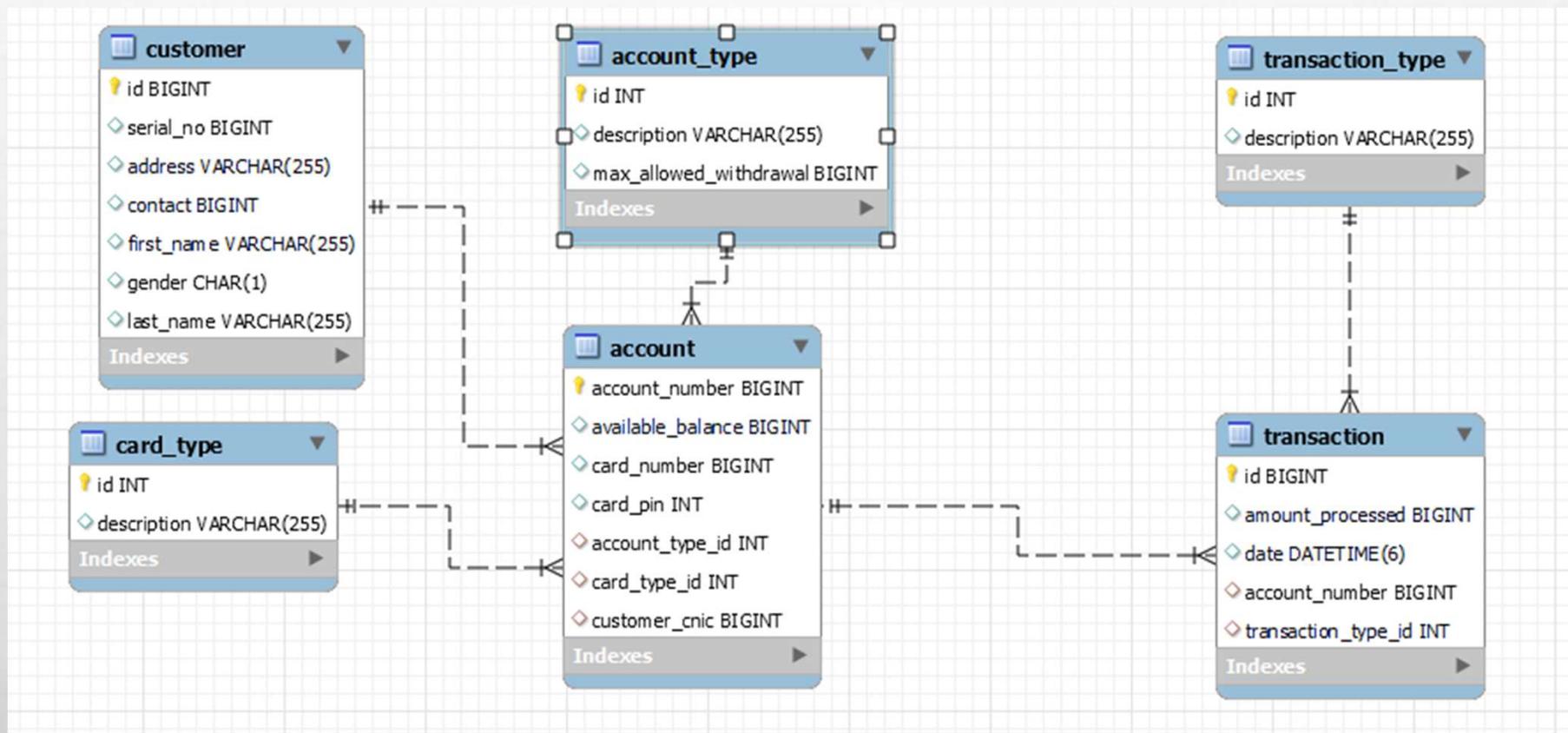
## TEST CASES:

- Login with correct credentials(Card Number and Pin)
- Login with incorrect credentials
- Balance Check
- Cash Withdrawal beyond limits
- Cash withdrawal within limits
- Cash Deposit Beyond Limits
- Cash Deposit within limits
- Update Pin by a new pin
- Update Pin with existing pin
- Implementing any of the test case without Login

# TOOLS USED:

- JAVA
- MySQL
- Spring Boot
- JPA & Hibernate
- Spring security
- Swagger
- Postman
- REST API
- Junit testing
- IntelliJ Idea Ultimate
- MySQL work bench
- Thymleaf

# ERD DIAGRAM FOR RELATIONS:



# UML DIAGRAMS:

**C**

### BalanceCheckService

«Autowired»  
AccountRepository accountRepository  
«Autowired»  
TransactionRepository transactionRepository  
«Autowired»  
TransactionTypeRepository typeRepository  
«Autowired»  
TransactionService transactionService  
«Autowired»  
AccountDataService accountDataService

□ modelData: ModelData  
□ login: Login

● getAvailableBalance(): String

**C**

### LoginService

«Autowired»  
AccountRepository accountRepository  
«Autowired»  
AccountDataService accountDataService

□ account: Account  
□ login: Login

● authenticateLogin(modelCardNumber: Long, modelCardPin: Long): String  
● exit(): void



**SWAGGER**



http://localhost:8090 - Generated server url ▾

## cash-withdrawal-controller ^

PUT /atm/withdraw-cash/{amount} ▾

## update-pin-controller ^

PUT /atm/update-pin/{pin} ▾

## cash-deposit-controller ^

PUT /atm/deposit-cash/{amount} ▾

## login-controller ^

GET /atm/login/{modelCardNumber}/{modelCardPin} ▾

## balance-check-controller ^

GET /atm/balance-inquiry ▾

# IMPLEMENTING TEST CASES:



# WRONG CREDENTIALS

Home Workspaces API Network Explore

Search Postman

+ Invite ⚙️ 🔔 🌐 Upgrade

My Workspace New Import

Overview GET http://localhost:8090/atm/login/123456789/12345 GET http://localhost:8090/atm/login/123456789/12345 PUT http://localhost:8090/atm/login/123456789/12345 PUT http://localhost:8090/atm/login/123456789/12345 + ⚙️ No Environment

Collections

APIs

Environments

Mock Servers

Monitors

Flows

History

No APIs yet

APIs define related collections and environments under a consistent schema.

Create an API

http://localhost:8090/atm/login/123456789/12345

GET http://localhost:8090/atm/login/123456789/12345

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Status: 200 OK Time: 462 ms Size: 183 B Save as Example

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize Text

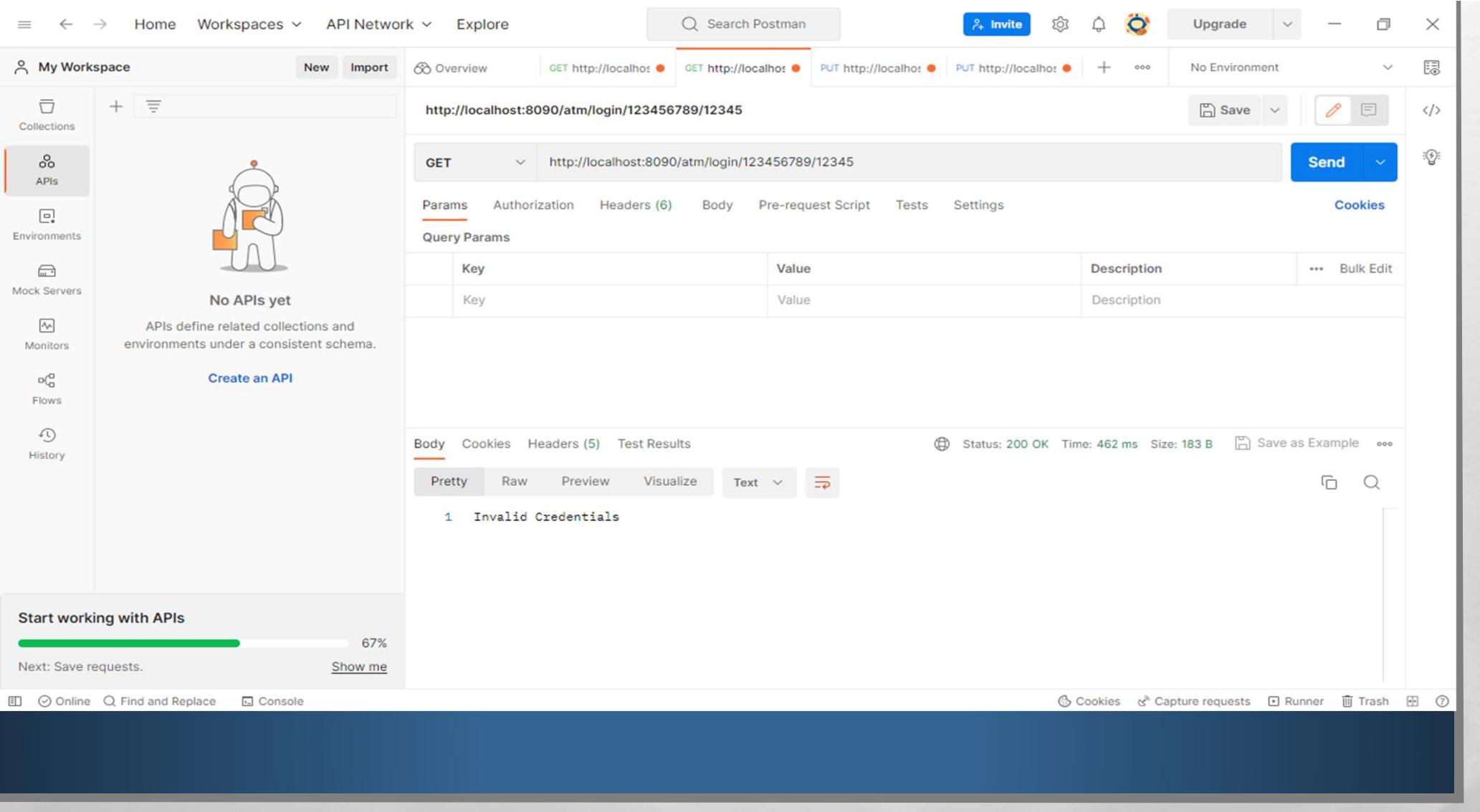
1 Invalid Credentials

Start working with APIs

67%

Next: Save requests. Show me

Online Find and Replace Console Cookies Capture requests Runner Trash



The screenshot shows the Postman application interface. The left sidebar contains navigation links: Home, Workspaces, API Network, Explore, Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main workspace is titled "My Workspace" and shows a placeholder for an API with a character icon. A message says "No APIs yet" and provides instructions: "APIs define related collections and environments under a consistent schema." Below this is a "Create an API" button. At the bottom of the sidebar, there's a progress bar labeled "Start working with APIs" at 67%, followed by "Next: Save requests." and a "Show me" link.

The central workspace displays a failed API request for "http://localhost:8090/atm/login/12345678/1234". The request method is "GET". The status bar indicates "Status: 200 OK Time: 62 ms Size: 183 B". The response body shows the error message "1 Invalid Credentials".

The top navigation bar includes links for Home, Workspaces, API Network, Explore, a search bar, and various user settings like Invite, Upgrade, and notifications.

The background of the image is a complex, abstract pattern resembling a topographic map or a microscopic view of organic tissue. It features a dense network of fine, wavy lines in shades of red, green, and black, creating a sense of depth and movement. The colors transition from deep reds and blacks in the recessed areas to bright greens and whites in the raised, textured regions. The overall effect is organic and dynamic.

**BALANCE INQUIRY**

Home Workspaces API Network Explore

Search Postman Invite Settings Notifications Upgrade No Environment

My Workspace New Import Overview GET http://localhost:8090/atm/balance-inquiry

GET http://localhost:8090/atm/balance-inquiry

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

No APIs yet

APIs define related collections and environments under a consistent schema.

Create an API

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 52 ms Size: 190 B Save as Example

Pretty Raw Preview Visualize Text

Available Balance = 525000

Start working with APIs 67% Next: Save requests. Show me

Online Find and Replace Console Cookies Capture requests Runner Trash

The screenshot shows the Postman application interface. On the left, there's a sidebar with icons for Collections, APIs (selected), Environments, Mock Servers, Monitors, Flows, and History. The main workspace is titled 'My Workspace' and shows a placeholder for an API with the message 'No APIs yet'. It includes a 'Create an API' button and a note about schema consistency. A central panel displays a request for 'http://localhost:8090/atm/balance-inquiry' with a 'GET' method. The 'Params' tab is selected, showing a table for 'Query Params' with two rows: 'Key' and 'Value'. Below this, the 'Body' tab is selected, showing the response: 'Available Balance = 525000'. The status bar at the bottom indicates a 200 OK status, 52 ms time, and 190 B size. The bottom navigation bar includes links for Online, Find and Replace, Console, Cookies, Capture requests, Runner, and Trash.

# **EXCEEDING WITHDRAWAL LIMIT**

Home Workspaces API Network Explore

Search Postman

Invite Settings Bell Upgrade

New Import Overview GET http://localhost:8080/ GET http://localhost:8080/ PUT http://localhost:8080/ PUT http://localhost:8080/ + ... No Environment

Collections

APIs

Environments

Cloud Servers

Monitors

Flows

History

No APIs yet

APIs define related collections and environments under a consistent schema.

Create an API

http://localhost:8090/atm/withdraw-cash/26000

PUT http://localhost:8090/atm/withdraw-cash/26000

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Description	... Bulk Edit
	Key	Value	Description	

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 28 ms Size: 208 B Save as Example

Pretty Raw Preview Visualize Text

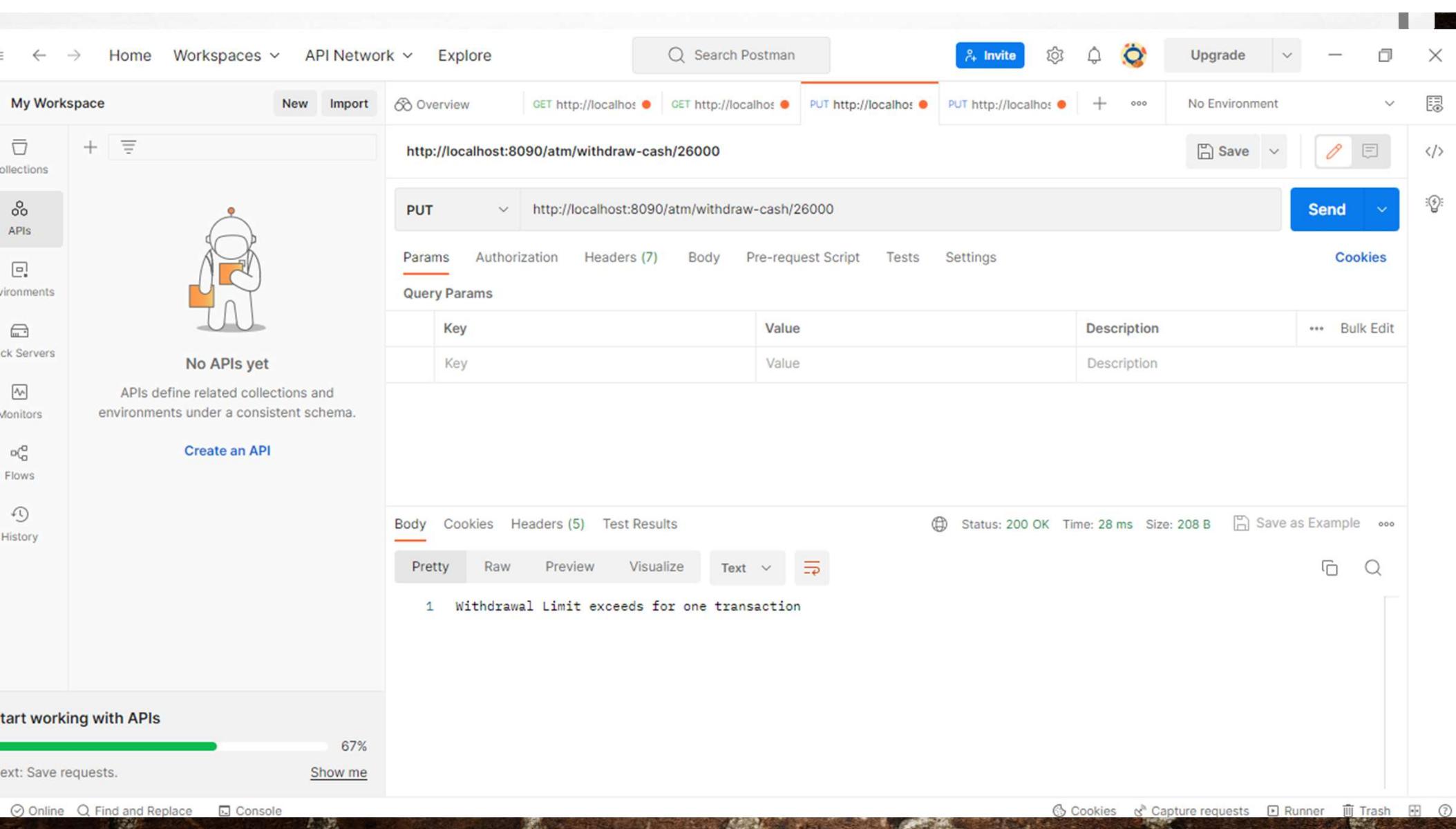
1 Withdrawal Limit exceeds for one transaction

Start working with APIs

67%

ext: Save requests. Show me

Online Find and Replace Console Cookies Capture requests Runner Trash



**SUCCESSFUL WITHDRAW**

Home Workspaces API Network Explore

Search Postman

Invite Settings Bell Upgrade

New Import Overview GET http://localhost:8090/ GET http://localhost:8090/ PUT http://localhost:8090/ PUT http://localhost:8090/ + ... No Environment

Save

PUT http://localhost:8090/atm/withdraw-cash/25000

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

No APIs yet

APIs define related collections and environments under a consistent schema.

Create an API

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 53 ms Size: 193 B Save as Example

Pretty Raw Preview Visualize Text

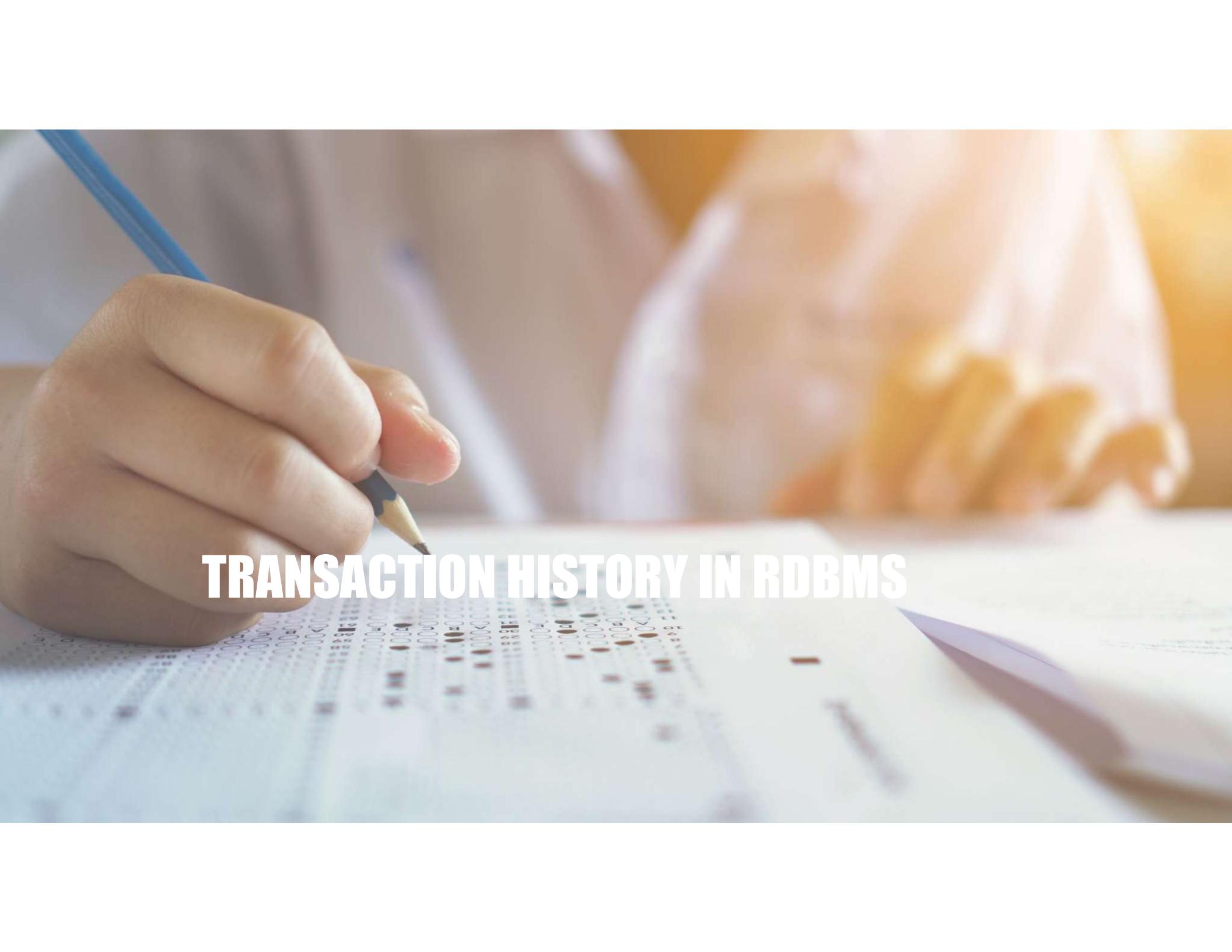
1 25000 withdrawn successfully

Start working with APIs

67%

Next: Save requests. Show me

Online Find and Replace Console Cookies Capture requests Runner Trash



# TRANSACTION HISTORY IN RDBMS

```
1 •  SELECT t.id,t.amount_processed,t.date,t.account_number,tt.description  
2   | FROM atm.transaction t,transaction_type tt  
3   | where t.transaction_type_id=tt.id;
```

< [ ]

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

	id	amount_processed	date	account_number	description
▶	1	0	2023-04-16 06:19:09.270000	1977900982903	Balance Inquiry
	2	0	2023-04-16 06:20:19.270000	1977900982903	Update Pin
	3	25000	2023-04-16 06:23:16.270000	1977900982903	Withdraw
	4	25000	2023-04-16 06:24:09.270000	1977900982903	Cash Deposit

# OUT OF SCOPE:

- Login with biometric
- Balance inquiry at different bank's ATM
- Cash withdrawal at different bank's ATM
- Cash deposit at different bank's atm
- Update pin at different bank's atm
- Bill payments
- Funds transfer to other accounts
- Further implementation of spring security
- Further implementation of frontend technologies

## FUTURE WORK:

- All the out of scope functionalities.
- Further Enhancements related to Security and Front-End



**ANY?**

**QUESTIONS**

**THANK YOU**