ATM Management System

Mitchel Baker
917679066

Github handle: mitchthebaker

Milestone Version	Date
M1V1	10/05/2021
M2V1	11/02/2021

1. Table of Contents

2. Section I: Project Description

ATM's, or automated teller machines, can be found everywhere in our day to day lives in locations ranging from banks, to corner stores, to even bars and nightclubs where you need instant access to your bank account for withdrawing cash. In an ideal world, ATM's function as expected, but inevitably, what happens when there are computer malfunctions or service repairs? ATM's, if not monitored frequently enough, may cease to operate or run out of cash during busy periods leaving customers with no other options. It is also common for ATM's to decline a customer's card when inserted, or when displaying incorrect notifications about a customer's bank information. Another huge problem, amounting to over $1 billion lost per year, are ATM skimmers who place their own magnetic strips over ATM card readers in order to read sensitive data from customers. Clearly, the current state of ATM machines have not only security issues but also limitations as a service when looking at its long term viability as a whole.

Since the main functionality of ATM machines revolves around the deposit and withdrawal of fiat currency from one's bank account, then it is imperative for these features to be maintained to the fullest with minimal complications. In the case that an ATM is close to depleting its cash reserves, the ATM admin monitoring the machine shall be notified in advance of this issue in order to provide cash refills when necessary. Credit card fraud, especially related to ATM skimming, can be prevented by simply removing the common magnetic stripe readers for verifying a user's bank account. A much more secure solution to credit card fraud is requiring ATM/Debit cards to be scanned using contactless NFC technology. Removing magnetic stripe readers kills two birds with one stone- the first being computer malfunctions from incorrect card reads, and the second being a stronger, secure, and more reliable way of connecting to a customer's sensitive bank information by using NFC, or near-field communications to provide contactless solutions for customers.

You may be asking, what makes our ATM different from other ATM machines out there? Our ATM will incorporate the use of modern context-aware IoT sensors to provide a more secure transaction experience for customers. These sensors will prevent credit card skimmers from adding unrecognized devices by shutting off access to critical portions of the system and by alerting ATM admins about significant issues which arise. We will also be combining IoT temperature sensors internally into our ATM machines to maintain the hardware's integrity. These sensors will prevent common computer malfunctions from occurring and improve upon the overall efficiency of the ATM system. We will also employ the use of fingerprint scanners for collecting biometric data, which will create a stronger authentication process as well. Our ATM machines are not only focused on improving security practices. Our customers will also be able to buy and sell, deposit and withdraw, plus borrow and lend the top cryptocurrencies straight from our ATM machines. All they have to do is set up a portfolio internally and we'll take care of the rest. Customers can use leading crypto exchanges for creating crypto transactions, and connect with open lending protocols to lend their cryptocurrency holdings to earn interest.

Overall, our ATM machine takes a modern approach towards solving the main problems customers face when interacting with standard ATM machines. Due to the ongoing need to either deposit or withdraw cash from a physical location, ATM machines will not be going away anytime soon. Therefore, it is of utmost importance for us to guarantee that a customer's data is safe and never compromised.

3. Section II: Use Cases

Title	ATM runs out of cash
Actors	ATM Machine, Authenticated user, ATM admin, Card reader device, Fingerprint scanner, ATM display
Description	Ben is looking to withdraw some cash, in dollars, from his Charles Schwab checking account. He walks up to the ATM and places his phone next to the contactless card reader device, which then scans his banking information stored in Apple Pay. After his banking information is scanned, Ben is prompted by the ATM display to securely login into his account using the biometric fingerprint scanner. After Ben's session is verified, he is then redirected to his bank details which displays the accounts he has linked to the ATM's database. Ben then taps on his checking account, chooses the amount to withdraw, and hits confirm. To Ben's surprise, the ATM display shows an error explaining that the machine is out of cash, and that an ATM operator should be arriving shortly in order to take care of the issue. If the amount of cash requested to withdraw from the machine is greater than the total remaining amount of cash in the ATM, then this is the case where an ATM operator will be notified immediately. The ATM operator will also be notified in advance, prior to the ATM's cash balance reaching $0, with the intent of restocking the total amount of cash in the ATM before a user can deplete it.

Title	Transfer funds to depleted checking account
Actors	ATM machine, Authenticated user, Checking account, Savings account, Card reader device, Fingerprint scanner, Keypad
Description	Jill is going about her weekend shopping for errands and grocery shopping. At this point, Jill has purchased about $300 worth of merchandise using her debit card. However, she doesn't realize that she's actually overdrafted the value of her checking account since she originally only had $200 to spend. Unknowingly, Jill walks up to our ATM machine to withdraw cash from her checking in order to fuel another shopping spree. After Jill scans her bank credentials using Apple wallet and logins using the fingerprint scanner, she is then shown a display of the accounts she's linked to the ATM machine. Jill selects her checking account, and then taps on "Withdraw" to take out $400. After doing so, Jill gets a notification from the ATM machine saying she does not have sufficient funds in her checking account to withdraw. The ATM also notifies her that she has overdrafted her checking account by $100, and will be charged a $35 overdraft fee if she does not replenish the checking account with funds. Jill is prompted to either cancel the transaction or transfer funds from one of her other accounts saved in the ATM's database. In order to fix the issue, Jill decides to follow the ATM's advice and transfer funds from the savings account she's linked back into her depleted checking account. Jill has a few thousand dollars in her savings account, so her transfer of $500 into her checking account goes through. As a result, her checking account's outstanding balance is -$100 + $500 = $400. Since Jill fixed the issue when she was prompted to do so, she was not charged the overdraft fee.

Title	Deposit a check into ATM to add into savings
Actors	ATM Machine, Authenticated user, Card reader device, Fingerprint scanner, Savings account, ATM display, Accounts, Deposit slot
Description	Tom has been working 60 hours a week plus overtime for the past few months. He has just received his paycheck in the mail, which he'd like to deposit into his savings account. Tom walks up to our ATM machine, uses his phone which has Google Pay setup to scan his bank account credentials, and then verifies his session with the ATM machine using the fingerprint scanner. After Tom is successfully logged into the ATM machine, he is then prompted by the ATM display to either withdraw or deposit funds. He selects Deposit funds, which redirects him to a screen asking if he would like to deposit either cash or a check. Tom selects the deposit check option. Tom is then prompted to choose which type of account he'd like to transfer his check into. Tom responds to the ATM display accordingly and chooses his savings account. After his selection, the ATM machine then communicates to the deposit slot to open so Tom can insert his check. The deposit slot closes and holds his check internally for processing after Tom inserts his check. Once the slot closes, the ATM display then notifies Tom that his check was received successfully, and that his savings account will be updated with his deposited amount within the next few business days.

Title	Add a crypto asset as an account
Actors	ATM Machine, Authenticated user, Regular account, Portfolio, Crypto asset account, Crypto exchange partner
Description	John is looking to add a new crypto account to his ATM portfolio. He would like to do so because he wants to keep track of his crypto holdings and have the ability to deposit/withdraw in the future. After scanning his smartphone and providing authorization to his account, the ATM machine then prompts John with a few options: to either Deposit/Withdraw funds or open Crypto Portfolio section. He chooses the crypto portfolio section, which redirects him to a screen explaining that no portfolio could be found linked to John's primary bank account. The ATM machine then asks John if he'd like to create a new crypto portfolio. John selects "Yes", which takes him to a separate portfolio creation screen. He is asked if he'd like to use the personal information already stored in the database, such as his first and last name, email, and phone number, in order to set up his crypto portfolio. He again selects "Yes," which finalizes the creation of his new crypto portfolio. Now that John's crypto portfolio has been created, it is now necessary for him to add a new account which pertains to a specific crypto asset. He is given a rundown of how to go about this after creating his portfolio. Now, John selects the "Create a new crypto account" option, which then redirects him to a screen asking him to choose a crypto asset. A list of the top cryptocurrencies are provided to him, such as Bitcoin, Ethereum, Cardano, and Chainlink. John opts to use Bitcoin as his first crypto account. Lastly, the ATM machine asks John if he'd like to add a new bitcoin account with a native segwit address. John selects "Yes," once again, which finalizes the creation of his new Bitcoin account inside his crypto portfolio.

Title	Withdraw crypto asset as fiat
Actors	ATM machine, ATM display, Bank account, mobile wallet, card reader device, ATM account, fingerprint scanner, Authenticated user, Portfolio, Crypto asset account, Exchange rates, ATM transaction, Withdrawal transaction, Cash dispenser
Description	Jane has authenticated access to her bank account by scanning her mobile wallet and successfully verified access to her ATM account by using the fingerprint scanner. Jane has already configured her ATM account to support crypto assets. Therefore, after verifying her ATM account, she selects the “Portfolio” option on the ATM display to view her Crypto asset accounts which are composed of the following currencies: Bitcoin, Ethereum, and Cardano. After doing so, the ATM display provides Jane with the options of exchanging, lending, or depositing/withdrawing her crypto assets. The Ethereum 2.0 update was recently completed, which caused the price of Ethereum to jump to upwards of 25%. Therefore, Jane has seen this as an opportunity to convert some of her Ethereum holdings to dollars so she has extra cash on the side to use for future investments. Jane chooses the “Exchange” option, which redirects her to a menu requesting to specify the asset she’d like to exchange and also the asset she’d like to receive in return. Jane uses the ATM display to select Ethereum as the asset to exchange, and then selects dollars as the asset she’ll receive in return. She then inputs 1 Ethereum as the amount to exchange which displays $2,995 after the amount is compared to the exchange rates the ATM database has on file. Jane is satisfied with the exchange, and hits “Convert” to finalize the exchange. After she taps on “Convert,” a new withdrawal transaction is created which is also added into ATM transactions. After ATM transactions are updated, the transaction is then sent to the cash dispenser for Jane to receive her converted Ethereum in dollars.

Title	An ATM admin performs a routine security checkup on an ATM machine
Actors	ATM admin, ATM machine, Security media content, Camera footage, Context-aware sensor data, Temperature sensor data, Region, kubeCDN
Description	Mike is an ATM admin who is working remotely from his home. His main role as an admin is to monitor a few of the ATM machines which are in the same region as he is. He performs routing checks on the security media content provided to him in order to ensure that the ATM machines are secure and operating smoothly. Each ATM machine he monitors is contained within a single region. Each region also hosts from the cloud a Kubernetes CDN, which has the responsibility of distributing the security media content it is storing. Since Mike has a unique access code provided to him and the correct access permissions to in order to access an ATM machine's internal devices and data, he can download camera footage, temperature sensor data, and context-aware sensor data from each of the ATM machines he is monitoring. After the downloads are complete, he looks over the security media content and ensures that everything is up to the security standards specified to him. He sees from the camera footage that customers are using the ATM machines properly with no issues. Mike also checks the context-aware sensor data, which indicates to him that there are no card skimming devices or other unauthorized devices in proximity with the ATM machines.

Title	Lending crypto assets to an open lending protocol
Actors	ATM machine, ATM account, Checking account, ATM Transaction, Transaction action, Exchange rate, Currency, Portfolio, Crypto account, Open lending protocol, Crypto loan, Crypto loan payment
Description	Earlier in the week, Sarah received a direct deposit into her checking account from her employer. Sarah prefers to hold stable coins on the side as opposed to holding cash, so she logs in to her ATM account and chooses to “Convert” from dollars to DAI. She is shown the exchange rate from $1 to 1 DAI, and is prompted that the exchange is possible since both Dollars and DAI are currencies supported by the ATM machine she is using. After she finalizes the exchange, it is created as an ATM transaction and finalized as a transaction action. Lastly, the transaction action confirms that Sarah’s crypto account provides valid permissions before the ATM transaction is confirmed. After the ATM transaction is confirmed, Sarah’s crypto account is then filled with the converted dollar amount in DAI token. Sarah sees the value in lending her DAI since she can earn compounding interest as opposed to just holding it on the side with no rewards. As a result, Sarah connects to Compound, one of the open lending protocols provided by our ATM machines, in order to lend her 2000 DAI. After she decides to lend the amount, the open lending protocol Compound creates a crypto loan with an interest rate of 6%, of type lend, and the date the loan was created. Since Sarah is lending this amount and not borrowing, she does not have to worry about a payment plan, the amount due, or amount remaining attributes.

4. Section III: Database Requirements (Business Rules)

1. General User
    - 1.1. A general user shall have at least one user authentication method
    - 1.2. A general user shall have at least one bank account
    - 1.3. A general user shall interact with at most one fingerprint scanner
    - 1.4. A general user shall have one unique user id
    - 1.5. A general user shall have one email
    - 1.6. A general user shall have one full name
    - 1.7. A general user shall have one first name
    - 1.8. A general user shall have one last name
    - 1.9. A general user shall have an is_authenticated attribute to indicate if the general user is authenticated or not

2. User Authentication Method
    - 2.1. A user authentication method shall have one general user
    - 2.2. A user authentication method is a mobile wallet
    - 2.3. A user authentication method is a debit/ATM card
    - 2.4. A user authentication method shall have a user auth id
    - 2.5. A user authentication method shall have a type
    - 2.6. A user authentication method shall indicate if it can be used internationally

3. Authenticated User
    - 3.1. An authenticated user shall create a session with zero or one ATM machines
    - 3.2. An authenticated user shall be authenticated by one and only one fingerprint scanner at a time
    - 3.3. An authenticated user shall login to at most one ATM account
    - 3.4. An authenticated user shall interact with one ATM display at a time
    - 3.5. An authenticated user shall interact with one keypad at a time
    - 3.6. An authenticated user shall optionally access one or many checking account(s) linked to their ATM account
    - 3.7. An authenticated user shall optionally access one or many savings account(s) linked to their ATM account
    - 3.8. An authenticated user is an ATM admin
    - 3.9. An authenticated user shall have a unique user authorization token
    - 3.10. An authenticated user shall have an authenticated at date
    - 3.11. An authenticated user shall indicate if it is an admin

4. Session
    4.1. A session shall have one unique session id
    4.2. A session shall belong to one and only one authenticated user
    4.3. A session shall have one expiration date
    4.4. A session shall have session data

5. ATM Admin
    5.1. An ATM admin shall maintain one or many ATM machines
    5.2. An ATM admin shall monitor many devices
    5.3. An ATM admin shall refill at least one cash dispenser if it is empty
    5.4. An ATM admin shall refill at least one receipt printer with paper and ink if necessary
    5.5. An ATM admin shall be able to download zero or many security media content
    5.6. An ATM admin is an authenticated user
    5.7. An ATM admin shall have a unique admin id
    5.8. An ATM admin shall have a unique access code (for administering IT support for various devices)
    5.9. An ATM admin shall have access permissions

6. Mobile Wallet
    6.1. A mobile wallet shall link to one bank account
    6.2. A mobile wallet shall scan one and only one card reader device at a time
    6.3. A mobile wallet shall have a unique mobile wallet id
    6.4. A mobile wallet shall have a card number
    6.5. A mobile wallet shall have an expiration date
    6.6. A mobile wallet shall have a cvc
    6.7. A mobile wallet shall have a zip

7. Debit/ATM Card
    7.1. A debit/ATM card shall link to one bank account
    7.2. A debit/ATM card shall be owned by only one general user
    7.3. A debit/ATM card shall scan one and only one card reader device at a time
    7.4. A debit/ATM card shall have a unique debit/ATM card id
    7.5. A debit/ATM card shall have a card number
    7.6. A debit/ATM card shall have an expiration date
    7.7. A debit/ATM card shall have a cvc
    7.8. A debit/ATM card shall have a zip

8. ATM Machine
    - 8.1. An ATM machine shall be maintained by one or many ATM admins
    - 8.2. An ATM machine shall have zero or many devices at a time
    - 8.3. An ATM machine shall create a session with zero or one authenticated user
    - 8.4. An ATM machine shall belong to one region
    - 8.5. An ATM machine shall have one ADA headphone jack
    - 8.6. An ATM machine shall have one speaker
    - 8.7. Many ATM machines shall have many languages
    - 8.8. An ATM machine shall have a unique TID (terminal id number)
    - 8.9. An ATM machine shall have a status to indicate whether it is on/off
    - 8.10. An ATM machine shall have access permissions

9. Devices
    - 9.1. A device shall have at one and only one ATM machine
    - 9.2. A device shall be monitored by many ATM admins
    - 9.3. A device is zero or many I/O devices
    - 9.4. A device is zero or many authentication devices
    - 9.5. A device is zero or many transaction devices
    - 9.6. A device is zero or many security devices
    - 9.7. A device shall have a unique device id
    - 9.8. A device shall belong to a unique TID it belongs to
    - 9.9. A device shall have a type

10. I/O Device
    - 10.1. An I/O device is a device
    - 10.2. An I/O device is zero or one ATM display
    - 10.3. An I/O device is zero or one keypad
    - 10.4. An I/O device is zero or one speaker
    - 10.5. An I/O device is zero or one ADA headphone jack
    - 10.6. An I/O device shall have a unique I/O device id
    - 10.7. An I/O device shall have a status
    - 10.8. An I/O device shall have access permissions

11. Authentication Device
    - 11.1. An authentication device is a device
    - 11.2. An authentication device is zero or one card reader device
    - 11.3. An authentication device is zero or one fingerprint scanner
    - 11.4. An authentication device shall have a unique authentication device id
    - 11.5. An authentication device shall have a status
    - 11.6. An authentication device shall have access permissions

12. Transaction Device
    - 12.1. A transaction device is a device
    - 12.2. A transaction device is a deposit/check slot
    - 12.3. A transaction device is a cash dispenser
    - 12.4. A transaction device is a receipt printer
    - 12.5. A transaction device shall have a unique transaction device id
    - 12.6. A transaction device shall have a status
    - 12.7. A transaction device shall have access permissions

13. Security Device
    - 13.1. A security device is a device
    - 13.2. A security device is zero or one camera
    - 13.3. A security device is zero or one IoT context-aware sensor
    - 13.4. A security device is zero or one IoT temperature sensor
    - 13.5. A security device shall have a unique security device id
    - 13.6. A security device shall have a status
    - 13.7. A security device shall have access permissions

14. ATM Display
    - 14.1. An ATM display is an I/O device
    - 14.2. An ATM display shall interact with one authenticated user at a time
    - 14.3. An ATM display shall have a unique display id
    - 14.4. An ATM display shall have a status
    - 14.5. An ATM display shall indicate if it has a malfunction

15. Keypad
    - 15.1. A keypad is an I/O device
    - 15.2. A keypad shall interact with one authenticated user at a time
    - 15.3. A keypad shall have a unique keypad id
    - 15.4. A keypad shall have a status
    - 15.5. A keypad shall indicate if it has a malfunction

16. Speaker
    - 16.1. A speaker is an I/O device
    - 16.2. A speaker shall have one ATM machine
    - 16.3. A speaker shall have a unique speaker id
    - 16.4. A speaker shall have a status
    - 16.5. A speaker shall indicate if it has a malfunction

17. ADA Headphone Jack
    17.1. An ADA headphone jack is an I/O device
    17.2. An ADA headphone jack shall have one ATM machine
    17.3. An ADA headphone jack shall have a unique ADA jack id
    17.4. An ADA headphone jack shall have a status
    17.5. An ADA headphone jack shall indicate if it has a malfunction

18. Card Reader Device
    18.1. A card reader device is an authentication device
    18.2. A card reader device shall scan at most one user authentication method at a time
    18.3. A card reader device shall verify the existence of one and only one bank account at a time
    18.4. A card reader device shall have a unique card reader id
    18.5. A card reader device shall have a status
    18.6. A card reader device shall indicate if it has a malfunction

19. Fingerprint Scanner
    19.1. A fingerprint scanner is an authentication device
    19.2. A fingerprint scanner shall interact with at most one general user
    19.3. A fingerprint scanner shall authenticate at most one authenticated user at a time
    19.4. A fingerprint scanner shall have a unique scanner id
    19.5. A fingerprint scanner shall have a status
    19.6. A fingerprint scanner shall indicate if it has a malfunction

20. Receipt Printer
    20.1. A receipt printer is a transaction device
    20.2. A receipt printer shall contain at least 1 receipt rolls at a time
    20.3. A receipt printer shall contain at least 1 ink cartridges at a time
    20.4. A receipt printer shall print many transaction actions
    20.5. A receipt printer shall have a unique printer id
    20.6. A receipt printer shall record the quantity of receipt rolls it has stored internally
    20.7. A receipt printer shall record the quantity of ink cartridges it has stored internally
    20.8. A receipt printer shall have a status
    20.9. A receipt printer shall indicate if it has a malfunction

21. Receipt Roll Paper
    - 21.1. Receipt roll paper shall be stored in at most one receipt printer at a time
    - 21.2. Receipt roll paper shall be 3” wide
    - 21.3. Each receipt roll paper shall be given a unique roll id
    - 21.4. Each receipt roll paper shall have a used at date
    - 21.5. Each receipt roll paper shall have a length

22. Ink Cartridges
    - 22.1. An ink cartridge shall be stored in at most one receipt printer at a time
    - 22.2. An ink cartridge shall be given a unique cartridge id
    - 22.3. An ink cartridge shall have a manufacturer
    - 22.4. An ink cartridge shall have a color

23. Deposit/check Slot
    - 23.1. A deposit/check slot is a transaction device
    - 23.2. A deposit/check slot shall process many transaction actions
    - 23.3. A deposit/check slot shall have a deposit slot id
    - 23.4. A deposit/check slot shall have a status
    - 23.5. A deposit/check slot shall indicate if it has a malfunction

24. Cash Dispenser
    - 24.1. A cash dispenser is a transaction device
    - 24.2. A cash dispenser shall process many transaction actions
    - 24.3. A cash dispenser shall have a dispenser id
    - 24.4. A cash dispenser shall have a status
    - 24.5. A cash dispenser shall indicate if it has a malfunction

25. Camera
    - 25.1. A camera is a security device
    - 25.2. A camera shall upload many camera footage
    - 25.3. A camera shall have a unique camera id
    - 25.4. A camera shall have a status to indicate if it is on/off
    - 25.5. A camera shall indicate if it has a malfunction

26. Camera Footage
    - 26.1. A camera footage is a security media content
    - 26.2. A camera footage shall be uploaded by one camera
    - 26.3. A camera footage shall have a unique footage_id
    - 26.4. A camera footage shall have a created at date
    - 26.5. A camera footage shall have an ended at date
    - 26.6. A camera footage shall have a duration

27. IoT Context-Aware Sensor
    - 27.1. An IoT context-aware sensor is a security device
    - 27.2. An IoT context-aware sensor shall upload many context-aware sensor data
    - 27.3. An IoT context-aware sensor shall have a unique context sensor id
    - 27.4. An IoT context-aware sensor shall have a status to indicate if it is on/off
    - 27.5. An IoT context-aware sensor shall indicate if it has a malfunction

28. Context-Aware Sensor Data
    - 28.1. A context-aware sensor data is a security media content
    - 28.2. A context-aware sensor data shall be uploaded by one IoT context-aware sensor
    - 28.3. A context-aware sensor data shall have a unique context data id
    - 28.4. A context-aware sensor data shall have a ATM violation status
    - 28.5. A context-aware sensor data shall have a device origin
    - 28.6. A context-aware sensor data shall have a description

29. IoT Temperature Sensor
    - 29.1. An IoT temperature sensor is a security device
    - 29.2. A IoT temperature sensor shall upload many temperature sensor data
    - 29.3. An IoT temperature sensor shall have a unique temperature sensor id
    - 29.4. An IoT temperature sensor shall have a status to indicate if it is on/off
    - 29.5. An IoT temperature sensor shall indicate if it has a malfunction

30. Temperature Sensor Data
    - 30.1. A temperature sensor data is a security media content
    - 30.2. A temperature sensor data shall be uploaded by one IoT temperature sensor
    - 30.3. A temperature sensor data shall have a temperature data id
    - 30.4. A temperature sensor data shall have an internal error status
    - 30.5. A temperature sensor data shall have a device origin
    - 30.6. A temperature sensor data shall have a description

31. ATM Account
    - 31.1. An ATM account shall be logged into by at most one authenticated user
    - 31.2. An ATM account shall link to at least one bank account
    - 31.3. An ATM account shall have zero or one portfolio
    - 31.4. An ATM account shall file zero or many fraud claims
    - 31.5. An ATM account shall create zero or many ATM transactions
    - 31.6. An ATM account shall have zero or many account notifications
    - 31.7. An ATM account shall have a unique ATM account id
    - 31.8. An ATM account shall have a unique user id
    - 31.9. An ATM account shall have a created at date

32. Region
    - 32.1. A region shall contain zero or many ATM machines
    - 32.2. A region shall host many kubeCDN’s
    - 32.3. A region shall have a unique region id
    - 32.4. A region shall have a geographic location
    - 32.5. A region shall have a timezone

33. Kubernetes Content Distribution Network
    - 33.1. A kubeCDN shall belong to one region
    - 33.2. A kubeCDN shall distribute many security media content
    - 33.3. A kubeCDN shall have a region id
    - 33.4. A kubeCDN shall have a CDN id
    - 33.5. A kubeCDN shall have a description

34. Security Media Content
    - 34.1. A security media content is camera footage, context-aware sensor data, and temperature sensor data
    - 34.2. A security media content shall be stored in one kubeCDN
    - 34.3. A security media content shall be downloaded by many ATM admins
    - 34.4. A security media content shall have a media content id
    - 34.5. A security media content shall have a content type
    - 34.6. A security media content shall have access permissions

35. Bank
    - 35.1. A bank shall operate one or many bank account(s)
    - 35.2. A bank shall partner with many crypto exchanges
    - 35.3. A bank shall have a unique bank id
    - 35.4. A bank shall have a name
    - 35.5. A bank shall have an address

36. Bank Account
    - 36.1. A bank account shall be operated by one bank
    - 36.2. A bank account shall have one and only one general user
    - 36.3. A bank account is a checking account
    - 36.4. A bank account is a savings account
    - 36.5. A bank account shall link to at least one mobile wallet or debit/ATM card
    - 36.6. A bank account’s existence shall be verified by one and only one card reader at a time
    - 36.7. A bank account shall grant permission for a transaction action to be processed
    - 36.8. A bank account shall have a unique bank account id
    - 36.9. A bank account shall have a bank id
    - 36.10. A bank account shall have a routing number

37. Checking Account
    - 37.1. A checking account is a bank account
    - 37.2. A checking account shall provide permission for a transaction action before it is processed
    - 37.3. A checking account shall have a unique checking account id
    - 37.4. A checking account shall have a balance
    - 37.5. A checking account shall have access permissions

38. Savings Account
    - 38.1. A savings account is a bank account
    - 38.2. A savings account shall provide permission for a transaction action before it is processed
    - 38.3. A savings account shall have a unique savings account id
    - 38.4. A savings account shall have a balance
    - 38.5. A savings account shall have access permissions

39. Crypto Account
    - 39.1. Many crypto accounts shall belong to one and only one portfolio
    - 39.2. A crypto account shall share crypto account info with at least one crypto exchange
    - 39.3. A crypto account shall provide permission for a transaction action to process
    - 39.4. A crypto account shall connect with zero or many open lending protocols
    - 39.5. A crypto account shall have zero or many crypto loans
    - 39.6. A crypto account shall have a unique crypto account id
    - 39.7. A crypto account shall have an asset type
    - 39.8. A crypto account shall have a balance
    - 39.9. A crypto account shall have a public key
    - 39.10. A crypto account shall have access permissions

40. Portfolio
    - 40.1. A portfolio shall have one and only one account
    - 40.2. A portfolio shall have zero or many crypto asset accounts
    - 40.3. A portfolio shall access ATM transaction info for zero or many ATM transactions
    - 40.4. A portfolio shall be able to find zero or many exchange rates
    - 40.5. A portfolio shall have one portfolio id
    - 40.6. A portfolio shall have one ATM account id
    - 40.7. A portfolio shall record the total number of crypto accounts
    - 40.8. A portfolio shall have a created at date

41. Crypto Exchange
    - 41.1. Many crypto exchanges shall partner with many banks
    - 41.2. A crypto exchange shall share crypto account info with at least one crypto account
    - 41.3. A crypto exchange shall have a unique exchange id
    - 41.4. A crypto exchange shall have a provider name
    - 41.5. A crypto exchange shall indicate if it is operational

42. Currency
    - 42.1. A currency shall be included in zero or many exchange rates
    - 42.2. A currency shall have a currency id
    - 42.3. A currency shall have currency name
    - 42.4. A currency shall have a circulation amount

43. Exchange Rate
    - 43.1. An exchange shall include zero or many currencies
    - 43.2. An exchange rate shall be found by zero or many portfolios
    - 43.3. An exchange rate shall have a exchange rate id
    - 43.4. An exchange rate shall have a trading date
    - 43.5. An exchange rate shall have a source currency
    - 43.6. An exchange rate shall have a target currency
    - 43.7. An exchange rate shall have a closing rate
    - 43.8. An exchange rate shall have an average rate

44. ATM Transaction
    - 44.1. An ATM transaction shall access ATM transaction info from zero or many portfolios
    - 44.2. An ATM transaction shall be finalized by one transaction action
    - 44.3. An ATM transaction shall have a unique transaction id
    - 44.4. An ATM transaction shall have a transaction type (withdrawal, transfer, etc)
    - 44.5. An ATM transaction shall have an amount

45. Transaction Actions
    - 45.1. A transaction action shall finalize zero or one ATM transaction
    - 45.2. A transaction action shall be processed by zero or one cash dispenser
    - 45.3. A transaction action shall be processed by zero or one deposit/check slot
    - 45.4. A transaction action shall be printed by one receipt printer
    - 45.5. A transaction action shall gain permission from zero or one bank account prior to being processed
    - 45.6. A transaction action shall gain permission from zero or one crypto account prior to being processed
    - 45.7. A transaction action shall have a transaction action id
    - 45.8. A transaction action shall have a is verified attribute
    - 45.9. A transaction action shall have a processing device
    - 45.10. A transaction action shall have a fee

46. Open Lending Protocols
    - 46.1. An open lending protocol shall connect to zero or many crypto accounts
    - 46.2. An open lending protocol shall be able to create zero or many crypto loans
    - 46.3. Many open lending protocols shall collect crypto loan payments
    - 46.4. An open lending protocol shall have a unique protocol id
    - 46.5. An open lending protocol shall have a provider name
    - 46.6. An open lending protocol shall indicate if it is operational

47. Crypto Loan
    - 47.1. A crypto loan shall be created by at least one open lending protocol
    - 47.2. A crypto loan shall belong to one and only one crypto account
    - 47.3. A crypto loan shall have zero or many crypto loan payments
    - 47.4. A crypto loan shall have a unique crypto loan id
    - 47.5. A crypto loan shall have a payment plan (monthly, annual, etc.)
    - 47.6. A crypto loan shall have a date it was created
    - 47.7. A crypto loan shall have a loan type (loan or borrow)
    - 47.8. A crypto loan shall have an asset type
    - 47.9. A crypto loan shall have a interest rate expressed in decimal form (9% would be 0.09)
    - 47.10. A crypto loan shall have a total amount due
    - 47.11. A crypto loan shall have the total amount remaining

48. Crypto Loan Payment
    - 48.1. A crypto loan payment shall be attributed to at most one crypto loan
    - 48.2. A crypto loan payment shall have a unique crypto loan payment id
    - 48.3. A crypto loan payment shall have the amount paid
    - 48.4. A crypto loan payment shall have the date it was paid

49. Account Notifications
    - 49.1. A notification shall have one and only ATM account
    - 49.2. A notification shall have a unique notification id
    - 49.3. A notification shall have a created date
    - 49.4. A notification shall have a message

50. Fraud Claim
    - 50.1. Many fraud claims shall belong to at most one ATM account
    - 50.2. A fraud claim shall have a unique claim id
    - 50.3. A fraud claim shall have a description
    - 50.4. A fraud claim shall have a created at date
    - 50.5. A fraud claim shall have a resolved at date

51. Languages
    - 51.1. A language shall be applied to many ATM machines
    - 51.2. A language shall have a unique language id
    - 51.3. A language shall have a country
    - 51.4. A language shall have a name
    - 51.5. A language shall be marked as supported or not supported
    - 51.6. A language shall have an added at date

5. Section IV: Detailed List of Main Entities, Attributes and Keys

1. General User (Strong)
    a. user_id: key, numeric
    b. email: alphanumeric
    c. full name: composite, multivalue, alphanumeric
    d. first name: multivalue, alphanumeric
    e. last name: multivalue, alphanumeric
    f. is_authenticated: numeric

2. User Authentication Method (Weak)
    a. user_auth_id: weak key, numeric
    b. type: composite, alphanumeric
    c. is_international: numeric

3. Authenticated User (Weak)
    a. user_auth_token: weak key, numeric
    b. authenticated_at: composite, date
    c. is_admin: numeric

4. Session (Weak)
    a. session_id: weak key, numeric
    b. expires_at: composite, date
    c. session_data: alphanumeric

5. ATM Admin (Weak)
    a. admin_id: weak key, numeric
    b. access_code: numeric
    c. access_permissions: numeric

6. Mobile Wallet (Strong)
    a. wallet_id: key, numeric
    b. card_number: alphanumeric
    c. exp_date: composite, date
    d. cvc: numeric
    e. zip: numeric

7. Debit/ATM Card (Strong)
    a. card_id: key, numeric
    b. card_number: numeric
    c. exp_date: composite, date
    d. cvc: numeric
    e. zip: numeric

8. ATM Machine (Strong)
    a. TID: key, numeric
    b. status: multivalue, numeric
    c. access_permissions: numeric

9. Devices (Strong)
    a. device_id: key, numeric
    b. TID: key, numeric
    c. type: multivalue, alphanumeric

10. I/O Device (Weak)
    a. IO_device_id: weak key, numeric
    b. status: multivalue, numeric
    c. access_permissions: numeric

11. Authentication Device (Weak)
    a. auth_device_id: weak key, numeric
    b. status: multivalue, numeric
    c. access_permissions: numeric

12. Transaction Device (Weak)
    a. transaction_device_id: weak key, numeric
    b. status: multivalue, numeric
    c. access_permissions: numeric

13. Security Device (Weak)
    a. security_device_id: weak key, numeric
    b. status: multivalue, numeric
    c. access_permissions: numeric

14. ATM Display (Weak)
    a. display_id: weak key, numeric
    b. status: multivalue, numeric
    c. has_malfunction: numeric

15. Keypad (Weak)
    a. keypad_id: weak key, numeric
    b. status: multivalue, numeric
    c. has_malfunction: numeric

16. Speaker (Weak)
    a. speaker_id: weak key, numeric
    b. status: multivalue, numeric

    c. has_malfunction: numeric
17. ADA Headphone Jack (Weak)
    a. ada_jack_id: weak key, numeric
    b. status: multivalue, numeric
    c. has_malfunction: numeric

18. Card Reader Device (Weak)
    a. card_reader_id: weak key, numeric
    b. status: multivalue, numeric
    c. has_malfunction: numeric

19. Fingerprint Scanner (Weak)
    a. scanner_id: weak key, numeric
    b. status: multivalue, numeric
    c. has_malfunction: numeric

20. Receipt Printer (Weak)
    a. printer_id: weak key, numeric
    b. paper_qty: multivalue, numeric
    c. ink_cartridge_qty: multivalue, numeric
    d. status: multivalue, numeric
    e. has_malfunction: numeric

21. Receipt Roll Paper (Strong)
    a. roll_id: key, numeric
    b. used_at: composite, date
    c. length: numeric

22. Ink Cartridge (Strong)
    a. cartridge_id: key, numeric
    b. manufacturer: alphanumeric
    c. color: alphanumeric

23. Deposit/Check Slot (Weak)
    a. deposit_slot_id: weak key, numeric
    b. status: multivalue, numeric
    c. has_malfunction: numeric

24. Cash Dispenser (Weak)
    a. dispenser_id: weak key, numeric
    b. status: multivalue, numeric
    c. has_malfunction: numeric

25. Camera (Strong)
    a. camera_id: key, numeric
    b. status: multivalue, numeric
    c. has_malfunction: numeric

26. Camera Footage (Weak)
    a. footage_id: weak key, numeric
    b. created_at: composite, date
    c. ended_at: composite, date
    d. duration: numeric

27. IoT Context-Aware Sensor (Strong)
    a. context_sensor_id: key, numeric
    b. Status: multivalue, numeric
    c. has_malfunction: numeric

28. Context-Aware Sensor Data (Weak)
    a. context_data_id: weak key, numeric
    b. atm_violation_status: numeric
    c. device_origin: alphanumeric
    d. description: alphanumeric

29. IoT Temperature Sensor (Strong)
    a. temp_sensor_id: key, numeric
    b. status: numeric
    c. has_malfunction: numeric

30. Temperature Sensor Data (Weak)
    a. temp_data_id: weak key, numeric
    b. internal_error_status: numeric
    c. device_origin: alphanumeric
    d. description: alphanumeric

31. ATM account (Weak)
    a. atm_account_id: weak key, numeric
    b. user_id: key, numeric
    c. created_at: composite, date

32. Region (Strong)
    a. region_id: key, numeric
    b. location: multivalue, alphanumeric
    c. timezone: multivalue, time

33. Kubernetes Content Distribution Network (Weak)
    a. region_id: key, numeric
    b. cdn_id: weak key, numeric
    c. cdn_description: alphanumeric

34. Security Media Content (Strong)
    a. media_content_id: key, numeric
    b. content_type: multivalue, alphanumeric
    c. access_permissions: numeric

35. Bank (Strong)
    a. bank_id: key, numeric
    b. name: multivalue, alphanumeric
    c. address: composite, alphanumeric

36. Bank Account (Weak)
    a. bank_account_id: weak key, numeric
    b. bank_id: key, numeric
    c. routing_number: numeric

37. Checking Account (Weak)
    a. checking_account_id: weak key, numeric
    b. balance: multivalue, numeric
    c. access_permissions: numeric

38. Savings Account (Weak)
    a. savings_account_id: weak key, numeric
    b. balance: multivalue, numeric
    c. access_permissions: numeric

39. Crypto Account (Weak)
    a. crypto_account_id: weak key, numeric
    b. asset_type: multivalue, alphanumeric
    c. balance: multivalue, numeric
    d. public_key: numeric
    e. access_permissions: numeric

40. Portfolio (Strong)
    a. portfolio_id: key, numeric
    b. atm_account_id: weak key, numeric
    c. num_crypto_accts: numeric
    d. created_at: composite, date

41. Crypto Exchange (Strong)
    a. exchange_id: key, numeric
    b. provider: alphanumeric
    c. is_operational: numeric

42. Currency (Weak)
    a. currency_id: weak key, numeric
    b. currency_name: alphanumeric
    c. circulation_amt: numeric

43. Exchange Rates (Strong)
    a. exchange_rate_id: key, numeric
    b. trading_date: composite, date
    c. source_currency: weak key, numeric
    d. target_currency: weak key, numeric
    e. closing_rate: multivalue, numeric
    f. average_rate: multivalue, numeric

44. ATM Transaction (Strong)
    a. transaction_id: key, numeric
    b. type: composite, alphanumeric
    c. amount: numeric

45. Transaction Actions (Strong)
    a. transaction_action_id: key, numeric
    b. printer_id: weak key, numeric
    c. is_verified: multivalue, numeric
    d. processing_device: multivalue, alphanumeric
    e. fee: numeric

46. Open Lending Protocols (Strong)
    a. protocol_id: key, numeric
    b. provider_name: alphanumeric
    c. is_operational: numeric

47. Crypto Loan (Strong)
    a. crypto_loan_id: key, numeric
    b. payment_plan: multivalue, alphanumeric
    c. created_at: composite, date
    d. loan_type: multivalue, alphanumeric
    e. asset_type: multivalue, alphanumeric
    f. interest_rate: multivalue, numeric
    g. total_amount: alphanumeric
    h. total_amount_remaining: alphanumeric

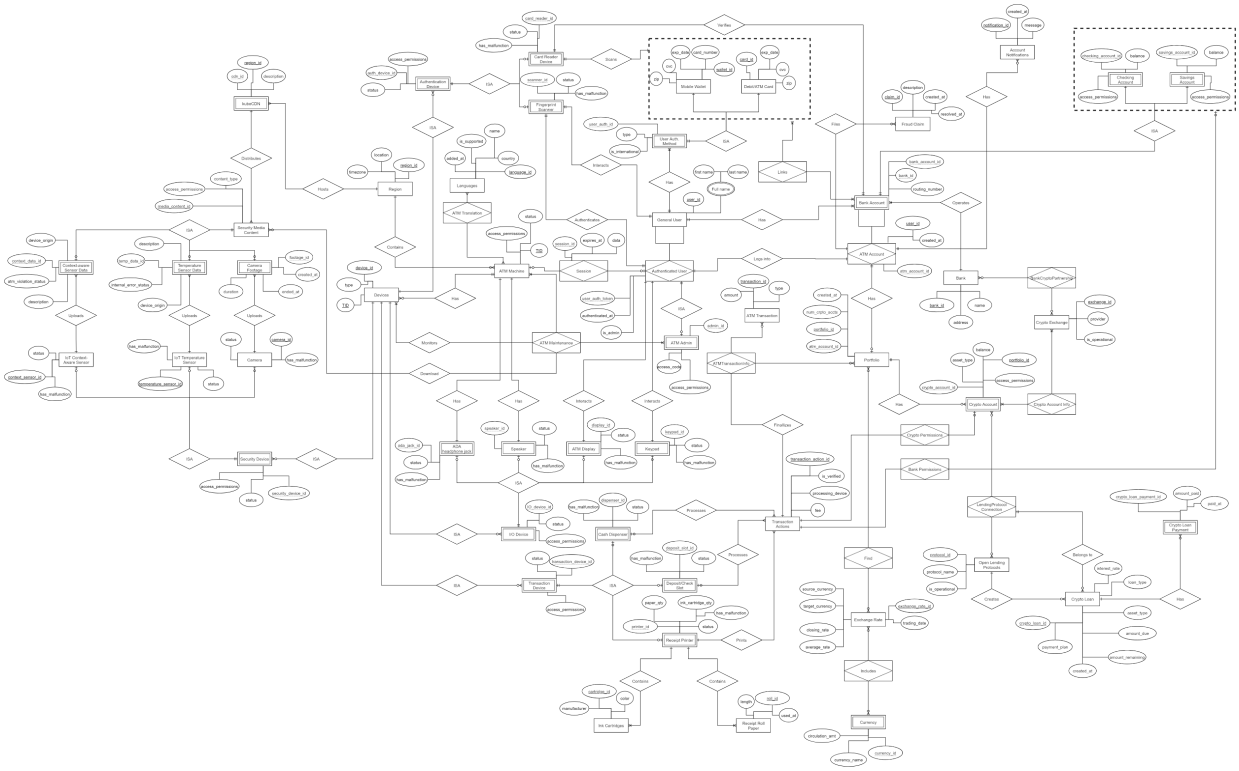
48. Crypto Loan Payment (Weak)
    a. crypto_loan_payment_id: weak key, numeric
    b. amount_paid: numeric
    c. paid_at: composite, date

49. Account Notifications (Strong)
    a. notification_id: key, numeric
    b. created_at: composite, date
    c. message: alphanumeric

50. Fraud Claim (Strong)
    a. claim_id: key, numeric
    b. description: alphanumeric
    c. created_at: composite, date
    d. resolved_at: composite, date

51. Languages (Strong)
    a. language_id: key, numeric
    b. country: multivalue, alphanumeric
    c. name: multivalue, alphanumeric
    d. is_supported: numeric
    e. added_at: composite, date

6. Section V: Entity Relationship Diagram (ERD)



7. Section VI: Testing Table

Rule	Entity A	Relation	Entity B	Cardinality	Pass/Fail	Error Description
1	General user	Interacts	ATM Machine	1-to-1	Fail	The general user needs to be authenticated first before accessing the ATM machine.
2	General user	Interacts	ATM display	1-to-1	Fail	Requirement is too vague, the relation is “interacts” here but for what reason? The user’s action here cannot be implied.
3	General user	Interacts	Keypad	1-to-1	Fail	Requirement is too vague, the general user will use the keypad to input amounts to withdraw/deposit/etc. but these are separate database requirements with their own respective actions.
4	General user	Have	Debit/ATM card or smartphone	1-to-M	Fail	First problem is that we don’t need a smartphone here, what we care about is a user’s mobile wallet. Therefore, a smartphone should be updated with a mobile wallet. Next, an aggregation should be created between the general user’s authentication method (card or mobile wallet) in case a general user has more than one authentication method.
5	General user	Belongs	Bank	1-to-M	Fail	A general user should not belong to a bank, but instead the general user should have a bank account which is linked to a bank.
6	General user	Authenticates	Bank account	1-to-M	Fail	A general user is not the entity authenticating a bank account, the user’s authentication method (mobile wallet or card) is what is scanned by the card reader device.
7	Authenticated user	Logs in	ATM account	1-to-1	Pass	None, except an authenticated user must have an auth token prior to logging in.
8	Fingerprint scanner	Verifies	Authenticated user	1-to-1	Fail	A fingerprint scanner should authenticate a general user; the verification is done when a user payment method scans a card reader to verify their bank account.
9	Authenticated user	Has	Bank account	1-to-M	Fail	It is assumed that a general user has a bank account prior to using our ATM machine, therefore we don’t need a relationship indicating an authenticated user has a bank account.
10	Authenticated user	Access	Checking account/ Savings account	1-to-M	Fail	An authenticated user, by nature of logging into their ATM account, shall have access to their many bank accounts. We have an ISA relation between bank account and checking account/savings account so this requirement is not necessary.

11	ATM admin	Monitors	ATM Machine	M-to-N	Fail	The relation here is correct but we should create an associative table so that ATM admins can monitor not only the ATM machine itself but also all of its internal devices.
12	ATM admin	Refills	Cash Dispenser	1-to-M	Pass	None, an ATM admin is responsible for refilling many cash dispensers.
13	ATM admin	Monitor	Cameras	1-to-M	Pass	None, except the footage uploaded from cameras should be stored in a self-hosted CDN.
14	ATM admin	ISA	Authenticated user	1-to-1	Pass	None
15	Mobile wallet	Linked	Bank account	M-to-N	Fail	It does not make sense for a mobile wallet to link to many bank accounts, we could have one or many mobile wallets linked to one bank account but not linked to many bank accounts.
16	ATM machine	Connects	ATM display	1-to-1	Fail	There has to be a better way of organizing how an ATM connects to each internal device
17	ATM machine	Connects	Keypad	1-to-1	Fail	Group keypad into a I/O device table
18	ATM machine	Connects	Card reader device	1-to-1	Fail	Group card reader device into an authentication device table
19	ATM machine	Connects	Fingerprint scanner	1-to-1	Fail	Group fingerprint scanner device into an authentication device table
20	ATM machine	Connects	Receipt printer	1-to-1	Fail	Group receipt printer device into an I/O device table
21	ATM machine	Connects	Speaker	1-to-1	Fail	Group speaker device into an I/O device table
22	ATM machine	Connects	Deposit/ check slot	1-to-1	Fail	Group deposit/check slot into an I/O device table
23	ATM machine	Connects	Cash dispenser	1-to-1	Fail	Group cash dispenser into an I/O device table
24	ATM machine	Connects	Camera	1-to-1	Fail	Group camera into a security device table
25	ATM machine	Connects	IoT context-aware	1-to-1	Fail	Group IoT context-aware sensor into a security device table

			sensor			
26	ATM machine	Connects	IoT temperature sensor	1-to-1	Fail	Group IoT temperature sensor into a security device table
27	ATM machine	Translates	Languages	1-to-M	Pass	None
28	ATM machine	Contains	Fraud claims	1-to-M	Pass	None
29	Bank	Operates	Bank accounts	1-to-M	Pass	None
30	Bank	Partner	Crypto Exchanges	M-to-N	Pass	None, but use an associative relationship here.
31	Bank Account	Operates	Bank	M-to-1	Pass	None
32	Checking account	Linked	Bank account	1-to-1	Fail	A checking account is a bank account, therefore we don't need to link the two.
33	Savings account	Linked	Bank account	1-to-1	Fail	A savings account is a bank account, therefore we don't need to link the two.
34	Checking account	Has	Account balance	1-to-1	Fail	Account balance can be made an attribute
35	ATM account	Has	Portfolio	1-to-1	Pass	None
36	Crypto account	Belongs to	Portfolio	M-to-1	Pass	None
37	Crypto account	Has	Crypto loans	1-to-M	Fail	The crypto account entity is not what creates a loan here, the open lending protocol the crypto account connects with shall create the crypto loan.
38	Portfolio	Has	ATM account	1-to-1	Pass	None
39	Portfolio	Has	Crypto accounts	1-to-M	Pass	None
40	Portfolio	Connects	Open lending protocol	1-to-1	Fail	A portfolio is not what connects to open lending protocols, a crypto account connects to open lending protocols. The 1-to-1 relation is also incorrect, 1-to-M should be used.
41	Portfolio	Has	Crypto	1-to-M	Fail	A crypto account shall have zero or many crypto

			loans			loans, not the portfolio.
42	Portfolio	Access	Crypto Exchange	1-to-M	Fail	A portfolio does not need to access a crypto exchange, a crypto account needs to access a crypto exchange to buy/sell.
43	Crypto exchange	Partners	Bank	M-to-N	Pass	None
44	Exchange rate	Belongs to	Currency	1-to-1	Pass	None
45	ATM transaction	Involves	ATM account	1-to-M	Fail	It does not make sense for an ATM transaction to involve many accounts, an ATM transaction shall be performed by one ATM account.
46	Withdrawal transaction	Involves	ATM account	1-to-M	Fail	The type of transaction, withdrawal, should be made the “type” attribute for ATM transactions. Don’t overcomplicate transactions with too many tables.
47	Withdrawal transaction	ISA	ATM transaction	1-to-1	Fail	Make withdrawal the “type” attribute for ATM transactions.
48	Transfer transaction	ISA	ATM transaction	1-to-1	Fail	Make transfer the “type” attribute for ATM transactions.
49	Open lending protocol	Create	Crypto loans	1-to-M	Pass	None
50	Open lending protocols	Collect	Crypto loan payments	1-to-M	Fail	An open lending protocol should monitor that crypto loan payments are being made, but there should be an associative relationship between the crypto account and open lending protocol connection so that the loan payment can be tied back to its respective account
51	Account notifications	Belongs to	ATM account	M-to-1	Pass	None
52	Friends list	Belongs to	ATM account	1-to-1	Fail	Remove Friends list table entirely, low priority in the broader scope of things.
53	Friends	Belongs to	Friends list	M-to-N	Fail	Too complicated to implement, since a Friend entity would be categorized as an ATM account? An authenticated user, only a general user? At this point in the development process we don’t know this information.
54	Devices	Has	ATM	M-to-1	Pass	None

			machine			
55	Devices	ISA	I/O device	1-to-1	Pass	None
56	Devices	ISA	Authentication device	1-to-1	Pass	None
57	Devices	ISA	Transaction device	1-to-1	Pass	None
58	Devices	ISA	Security device	1-to-1	Pass	None
59	I/O device	ISA	Devices	1-ot-1	Pass	None
60	I/O device	ISA	ATM display	1-to-1	Pass	None
61	I/O device	ISA	Keypad	1-to-1	Pass	None
62	I/O device	ISA	Speaker	1-to-1	Pass	None
63	Authentication device	ISA	Devices	1-to-1	Pass	None
64	Authentication device	ISA	Card reader device	1-to-1	Pass	None
65	Authentication device	ISA	Fingerprint scanner	1-to-1	Pass	None
66	Transaction device	ISA	Devices	1-to-1	Pass	None
67	Transaction device	ISA	Deposit/ Check slot	1-to-1	Pass	None
68	Transaction device	ISA	Cash dispenser	1-to-1	Pass	None
69	Transaction device	ISA	Receipt printer	1-to-1	Pass	None
70	Security	ISA	Devices	1-to-1	Pass	None

	device					
71	Security device	ISA	Camera	1-to-1	Pass	None
72	Security device	ISA	IoT context-aware sensor	1-to-1	Pass	None
73	Security device	ISA	IoT temperature sensor	1-to-1	Pass	None
74	ATM machine	Interacts	General user	1-to-1	Fail	An ATM machine shall only interact with an authenticated user. A general user needs to verify their bank account and then authenticate themselves with an auth token prior to using an ATM machine.
75	General user	Authenticate	Bank account	1-to-M	Fail	The general user will verify their bank account after scanning their mobile wallet or ATM/Debit card. Therefore, the relation should be changed to verify, and the entities involved should be the card reader device which verifies the bank account.
76	Authenticated user	Verify	ATM account	1-to-1	Fail	A general user becomes authenticated only after they verify their bank account, then authenticate themselves using the fingerprint scanner. Once an authenticated user has an auth token, all they have to do is login, there is no verifying necessary.
77	Smart phone	Connects	Mobile wallet	1-to-1	Fail	Including a smartphone entity is redundant, why would we need a table for this? We'd need to create a table to record the type of authentication methods the general user has, which would be either a mobile wallet (contained inside of a smartphone) or a ATM/Debit card.
78	General user	Has	User authentication method	1-to-M	Pass	None, now we have a better way of storing a user's authentication method.
79	General user	Has	Debit/ATM card	1-to-1	Fail	This requirement is not necessary since a user can have many authentication methods, which are mobile wallets or Debit/ATM cards.
80	General user	Has	Mobile wallet	1-to-1	Fail	This requirement is not necessary since we have many authentication methods. The 1-to-1 is also incorrect because the general user may have many mobile wallets, we can't assume they only have one since this would limit the user's

						experience.
81	User authentication method	Has	Authentication type	1-to-M	Fail	This requirement is unnecessary since the authentication type can be made an attribute. No table is needed here.
82	Bank	Has	General users	1-to-M	Fail	A bank shall have many bank accounts, which are owned by general users. This relationship does not make sense between general users and a bank.
83	General user	Has	Bank	1-to-M	Fail	A general user shall have many bank accounts associated with many banks, is a more realistic approach to creating this relation.
84	General user	Has	Bank account	1-to-M	Pass	None
85	Authenticated user	Has	Bank account	1-to-M	Fail	We already have the relationship between a general user and their respective bank accounts, therefore we do not need a relationship between an authenticated user and a bank account.
86	General user	Interacts	Fingerprint scanner	1-to-1	Pass	None
87	Authenticated user	Scans	Card reader device	1-to-1	Fail	An authenticated user is not the entity scanning the card reader device. An authentication method, which is a mobile wallet or Debit/ATM card, is what scans the card reader device.
88	Card reader device	Scans	User authentication method	1-to-1	Pass	None
89	Card reader device	Verifies	Bank account	1-to-1	Pass	This is the correct relation we want. The card reader device is what will verify the existence of at most one bank account at a time.
90	Fingerprint scanner	Authenticates	Authenticated user	1-to-1	Pass	This is the correct relation we want. The fingerprint scanner is what authenticates an authenticated user, who is then provided with their auth token.
91	ATM account	Creates	ATM transactions	1-to-M	Fail	Yes, an ATM account should initially create an ATM transaction, but the ATM transaction should be finalized by a transaction action first prior to being processed.
92	ATM admin	Monitors	Devices	M-to-N	Pass	None, but create an associative relationship here.

93	Region	Contains	Banks	1-to-M	Fail	We don't care about what region banks are here. We care that ATM machines are in different regions because we will use a CDN to distribute security content to the ATM machine's ATM admins more effectively.
94	Bank	Contains	Region	M-to-1	Fail	We don't care about the relationship between banks and regions.
95	Transaction action	Finalizes	ATM transaction	1-to-M	Pass	None
96	Cash Dispenser	Processes	Transaction action	1-to-1	Fail	The cardinality here should be 1-to-M
97	Deposit/check slot	Processes	Transaction action	1-to-1	Fail	The cardinality here should be 1-to-M
98	Receipt printer	Prints	Transaction action	1-to-1	Fail	The cardinality here should be 1-to-M
99	Transaction action	Permissions	Checking account	1-to-1	Fail	A transaction action can be processed by a bank account, since a checking account is a bank account.
100	Transaction action	Permissions	Savings account	1-to-1	Fail	A transaction action can be processed by a bank account, since a savings account is a bank account.
101	Region	Contains	ATM machines	1-to-M	Pass	None
102	ATM account	Create	ATM transactions	1-to-M	Fail	An ATM account should not be what creates an ATM transaction. This should be changed to the ATM's accounts portfolio. A user will use their portfolio to create either ATM actions or crypto actions
103	Portfolio	ATM transaction info	ATM transaction	1-to-M	Fail	Change cardinality to M-to-N. There should be more than one portfolio which can create ATM transactions. I think the mixup here is that an ATM account has one portfolio, so I thought one portfolio should make many ATM transactions. But, it makes sense that many portfolios can create many ATM transactions.
104	Portfolio	ATM transaction info	ATM transaction	M-to-N	Pass	None

105	Portfolio	Access	Exchange rates	1-to-M	Fail	A portfolio should access exchange rates, but a portfolio should have the relation “convert” instead, which will be tied to an entity set of exchange rates which belong to currency.
106	Currency	Has	Exchange rate	1-to-1	Fail	Shouldn’t need a separate table for currencies here. Create an exchange rates table with a source currency/target currency instead of using a Currency table.
107	Exchange rate	Belongs to	Currency	1-to-1	Fail	An exchange rate does not belong to a currency. An exchange rate shall involve two currencies.
108	Portfolio	Converts	Currency	1-to-M	Fail	The original idea was to have an aggregation of the currencies and exchange rates, which the portfolio would then “convert,” but this complexity is unnecessary. All we need here is a relation between portfolio and exchange rates.
109	Portfolio	Finds	Exchange rate	M-to-N	Pass	None
110	Currency	Includes	Exchange rate	M-to-N	Pass	None
111	Exchange rate	Includes	Currency	M-to-N	Pass	None
112	ATM Machine	Contains	Region	M-to-1	Pass	None
113	Region	Host	kubeCDN	1-to-M	Pass	None
114	kubeCDN	Hosted by	Region	1-to-1	Pass	None
115	kubeCDN	Distribute	Security media content	1-to-M	Pass	None
116	ATM admin	Downloads	Security media content	M-to-N	Pass	None

8. Section VII: Database model / EER

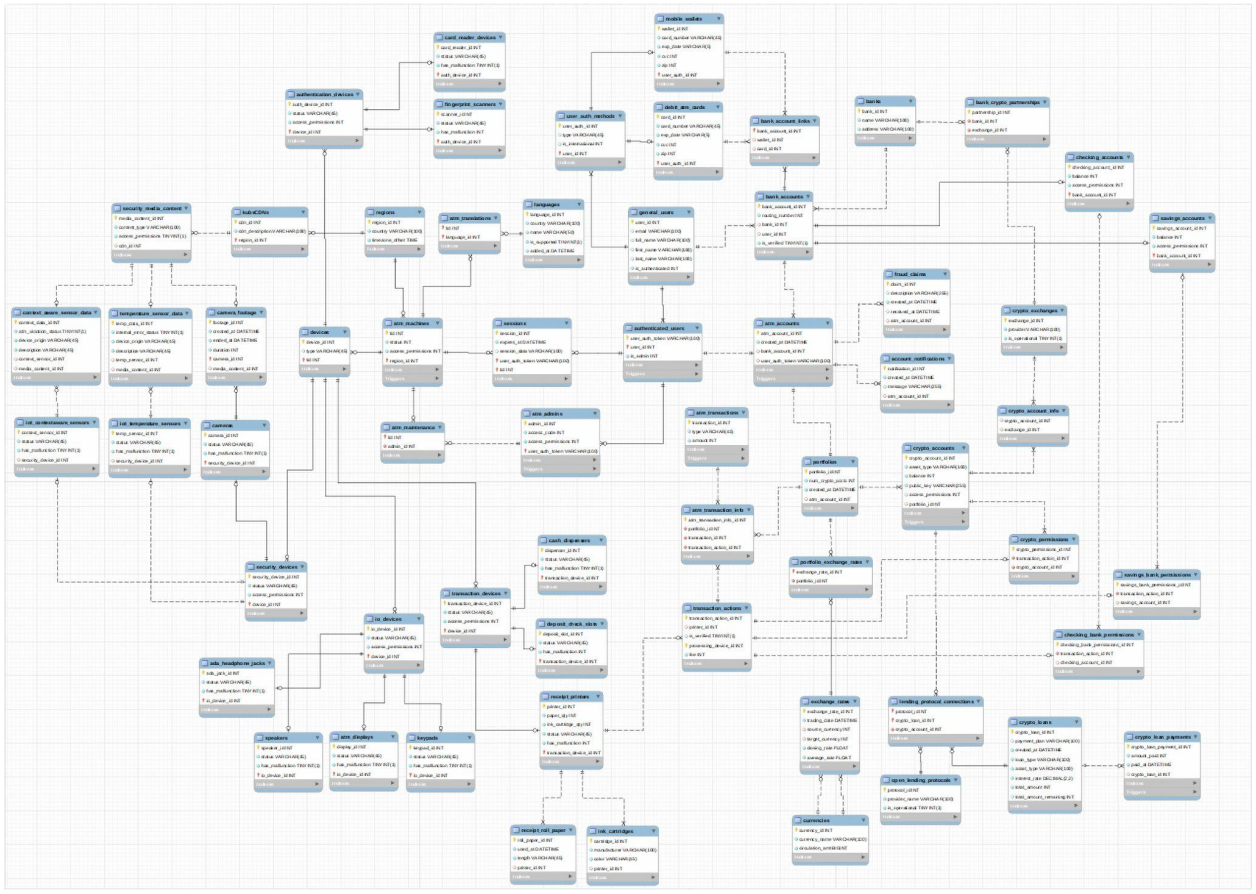


Table	FK	On Delete	On Update	Comment
mobile_wallets	user_auth_method	CASCADE	CASCADE	If a user authentication method is deleted, then the mobile wallet associated with the method must be deleted as well.
debit_atm_cards	user_auh_method	CASCADE	CASCADE	If a user authentication method is deleted, then the debit/ATM card associated with the method must be deleted as well.
authenticated_users	general_users	CASCADE	CASCADE	If a general user is deleted, then the user's authentication data should also be deleted.
sessions	authenticated_users	CASCADE	CASCADE	If an authenticated user is deleted, then the user's session should also be destroyed.
sessions	atm_machine	CASCADE	CASCADE	If an ATM machine is deleted, then any sessions associated with the machine shall be deleted.
atm_admins	authenticated_user	CASCADE	CASCADE	If an authenticated user is deleted, then any atm admin row related to the authenticated user must also be deleted.
devices	atm_machine	CASCADE	CASCADE	If an ATM machine is deleted, any associated devices should be set to null until they are assigned a new ATM machine.
io_devices	devices	CASCADE	CASCADE	If a device is deleted, then the io device associated with this device must also be deleted.
authentication_devices	devices	CASCADE	CASCADE	If a device is deleted, then the authentication device associated with this device must also be deleted.
transaction_devices	devices	CASCADE	CASCADE	If a device is deleted, then the transaction device associated with this device must also be deleted.
atm_displays	io_device	CASCADE	CASCADE	If an IO device is deleted, then the atm display associated with this device must be deleted.
keypads	io_device	CASCADE	CASCADE	If an IO device is deleted, then the keypad associated with this device must

				also be deleted.
speakers	io_device	CASCADE	CASCADE	If an IO device is deleted, then the speaker associated with this device must also be deleted.
ada_headphone_jacks	io_device	CASCADE	CASCADE	If an IO device is deleted, then the ADA headphone jack associated with this device must also be deleted.
card_reader_devices	authentication_devices	CASCADE	CASCADE	If an authentication device is deleted, then the card reader device associated with it must also be deleted.
fingerprint_scanners	authentication_devices	CASCADE	CASCADE	If an authentication device is deleted, then the fingerprint scanner device associated with it must also be deleted.
receipt_printers	transaction_devices	CASCADE	CASCADE	If a transaction device is deleted, then the receipt printer associated with it must also be deleted.
receipt_roll_paper	receipt_printers	SET NULL	CASCADE	If a receipt printer is deleted, then the receipt roll papers associated with it should be set to null until they are assigned to a new receipt printer.
ink_cartridges	receipt_printers	SET NULL	CASCADE	If a receipt printer is deleted, then the ink cartridges associated with it should be set to null until they are assigned to a new receipt printer.
deposit_check_slots	transaction_devices	CASCADE	CASCADE	If a transaction device is deleted, then the deposit/check slot it is paired with should also be deleted.
cash_dispensers	transaction_devices	CASCADE	CASCADE	If a transaction device is deleted and is a cash dispenser, then delete the corresponding row.
cameras	security_devices	CASCADE	CASCADE	If a security device is deleted and is associated with a camera device, delete the camera device as well.
iot_context_aware_sensor	security_devices	SET NULL	CASCADE	If a security device is deleted, then the IoT context aware sensor associated with this security device shall be set to null until it is reassigned.
context_awware_sensor_data	iot_contextaware_sensor	SET NULL	CASCADE	If a IoT context aware sensor is deleted and it is associated with context aware

				sensor data, set the column value to null.
iot_temperature_sensor	security_device	SET NULL	CASCADE	If a security device is deleted and it is paired with an iot temperature sensor, set the affected row to null until it is reassigned.
temperature_sensor_data	iot_temperature_sensor	SET NULL	CASCADE	If a iot temperature sensor is deleted and its associated with temperature sensor data, set the affected fk row to null.
bank_accounts	banks	SET NULL	CASCADE	If a bank row is deleted and associated with a bank account, set the affected column in bank account to null.
bank_accounts	general_users	SET NULL	CASCADE	If a row in the general user's table is deleted, set the affected column in the bank accounts table to null.
atm_accounts	bank_accounts	SET NULL	CASCADE	If a row in bank accounts is deleted, set the affected column in ATM accounts to null.
bank_account_links	bank_accounts	CASCADE	CASCADE	If a row in bank accounts is deleted, then delete the bank account link connected to it.
bank_account_links	mobile_wallets	SET NULL	CASCADE	If a row in mobile wallets is deleted, set the affected column in bank account links to null.
bank_account_links	debit_atm_cards	SET NULL	CASCADE	If a row in debit/ATM cards is deleted, set the affected column in bank account links to null.
checking_accounts	bank_accounts	CASCADE	CASCADE	If a row in bank accounts is deleted, delete the checking account tied to it as well.
savings_accounts	bank_accounts	CASCADE	CASCADE	If a row in bank accounts is deleted, then delete the savings account tied to it as well.
portfolios	atm_accounts	SET NULL	CASCADE	If a row in ATM accounts is deleted, then set the corresponding column in a portfolio to null.
crypto_accounts	portfolios	SET NULL	CASCADE	If a row in portfolios is deleted, set the affected column in the corresponding

				crypto account row to null.
crypto_account_info	crypto_accounts	SET NULL	CASCADE	If a row in crypto accounts is deleted, set the affected column in the corresponding crypto account info row to null.
crypto_account_info	crypto_exchanges	SET NULL	CASCADE	If a row in crypto exchanges is deleted, set the affected column in the corresponding crypto account info row to null.
atm_machines	regions	CASCADE	CASCADE	If a row in regions is deleted, delete the corresponding row in ATM machines as well.
kubeCDNs	regions	CASCADE	CASCADE	If a row in regions is deleted, delete the corresponding row in kubeCDNs as well.
security_media_content	kubeCDN	SET NULL	CASCADE	If a row in kubeCDNs is deleted, set the corresponding column in security media content to null.
context_aware_sensor_data	security_media_content	SET NULL	CASCADE	If a row in security media content is deleted, set the corresponding column in context aware sensor data to null.
temperature_sensor_data	security_media_content	SET NULL	CASCADE	If a row in security media content is deleted, set the corresponding column in temperature sensor data to null.
camera_footage	security_media_content	SET NULL	CASCADE	If a row in security media content is deleted, set the corresponding column in camera footage to null.
exchange_rates	currencies	SET NULL	CASCADE	If a row in currencies is deleted, then set the corresponding column in exchange_rates to null so it can be set to a new currency in the future.
transaction_actions	receipt_printers	SET NULL	CASCADE	If a row in receipt printers is deleted, then set the corresponding column in transaction actions to null so we can continue to see these rows for future reference.
lending_protocol_connections	open_lending_protocols	CASCADE	CASCADE	If a row in open lending protocols is deleted, then delete the corresponding row inside lending protocol connections.
lending_protocol_co	crypto_loans	CASCADE	CASCADE	If a row in crypto loans is deleted, then

nnections				delete the corresponding row inside lending protocol connections.
lending_protocol_connections	crypto_accounts	CASCADE	CASCADE	If a row in crypto accounts is deleted, then delete the corresponding row inside lending protocol connections.
crypto_loan_payments	crypto_loans	SET NULL	CASCADE	If a row in crypto loans is deleted, set the corresponding row in crypto loan payments to null.
atm_transaction_info	portfolios	CASCADE	CASCADE	If a row in portfolios is deleted, then delete the corresponding row inside atm transaction info.
atm_transaction_info	atm_transactions	CASCADE	CASCADE	If a row in atm transactions is deleted, then delete the corresponding row inside atm transaction info.
atm_transaction_info	transaction_actions	CASCADE	CASCADE	If a row in transaction actions is deleted, then delete the corresponding row inside atm transaction info.
account_notifications	atm_accounts	SET NULL	CASCADE	If a row in atm accounts is deleted, then set the corresponding row in account notifications to null.
fraud_claims	atm_accounts	SET NULL	CASCADE	If a row in atm accounts is deleted, then set the corresponding row in fraud claims to null.
atm_translations	atm_machines	CASCADE	CASCADE	If a row in atm machines is deleted, then delete the corresponding row in atm translations.
atm_translations	languages	CASCADE	CASCADE	If a row in languages is deleted, then delete the corresponding row in atm translations.
crypto_permissions	transaction_actions	CASCADE	CASCADE	If a row in transaction actions is deleted, then delete the corresponding row in crypto permissions.
crypto_permissions	crypto_accounts	CASCADE	CASCADE	If a row in crypto accounts is deleted, then delete the corresponding row in crypto permissions.
checking_bank_permissions	transaction_action_id	CASCADE	CASCADE	If a row in transaction action is deleted, then delete the corresponding row in checking bank permissions.

checking_bank_permissions	checking_accounts	CASCADE	CASCADE	If a row in checking accounts is deleted, then delete the corresponding row in checking bank permissions.
savings_bank_permissions	transaction_actions	CASCADE	CASCADE	If a row in transaction actions is deleted, then delete the corresponding row in savings bank permissions.
savings_bank_permissions	savings_accounts	CASCADE	CASCADE	If a row in savings accounts is deleted, then delete the corresponding row in savings bank permissions.
atm_maintenance	atm_machines	CASCADE	CASCADE	If a row in atm machines is deleted, then delete the corresponding row in atm maintenance.
atm_maintenance	devices	CASCADE	CASCADE	If a row in devices is deleted, then delete the corresponding row in atm maintenance.
atm_maintenance	atm_admins	CASCADE	CASCADE	If a row in atm admins is deleted, then delete the corresponding row in atm maintenance.
user_auth_methods	general_users	CASCADE	CASCADE	If a row in general users is deleted, then delete the corresponding row in user authentication methods since we no longer need to store this information for the general user.
bank_crypto_partnerships	banks	NO ACTION	CASCADE	If a row in banks is deleted, don't perform an action. Just because a bank is deleted does not mean its partnership with a crypto exchange is void.
bank_crypto_partnerships	crypto_exchanges	NO ACTION	CASCADE	If a row in crypto exchanges is deleted, don't perform an action. Just because a crypto exchange is deleted does not mean its partnership with a bank is void.

9. Section VIII: Forward Engineering

a. Please refer to eer.mwb file in https://github.com/sfsu-joseo/databases-system-fall21-mitchthebaker/tree/master/milestones/Milestone2/files

10. Section IX: Inserting Data

a. Please refer to inserts.sql in https://github.com/sfsu-joseo/databases-system-fall21-mitchthebaker/tree/master/milestones/Milestone2/files

11. Section X: Testing

a. Please refer to tests.sql in https://github.com/sfsu-joseo/databases-system-fall21-mitchthebaker/tree/master/milestones/Milestone2/files

12. Section XI: Testing Table

Entity	SQL Query	Pass/Fail	Error Description	Solution
general_users	Delete	Pass	None	None
general_users	Update	Pass	None	None
banks	Delete	Fail	A foreign key constraint fails	Update bank_crypto_partnerships ON DELETE option to CASCADE
banks	Update	Pass	None	None
user_auth_methods	Delete	Pass	None	None
user_auth_methods	Update	Pass	None	None
mobile_wallets	Delete	Fail	Tried to update a table without a WHERE that uses a KEY column	Reference wallet_id instead of card_number
mobile_wallets	Update	Pass	None	None
debit_atm_cards	Delete	Pass	None	None
debit_atm_cards	Update	Pass	None	None
bank_account_links	Delete	Pass	None	None
bank_account_links	Update	Pass	None	None
authenticated_users	Delete	Pass	None	None
authenticated_users	Update	Pass	None	None
regions	Delete	Pass	None	None
regions	Update	Pass	None	None
atm_machines	Delete	Pass	None	None
atm_machines	Update	Pass	None	None
languages	Delete	Pass	None	None
languages	Update	Pass	None	None

atm_accounts	Delete	Pass	None	None
atm_accounts	Update	Pass	None	None
portfolios	Delete	Pass	None	None
portfolios	Update	Pass	None	None
crypto_accounts	Delete	Pass	None	None
crypto_accounts	Update	Pass	None	None
crypto_account_info	Delete	Pass	None	None
crypto_account_info	Update	Pass	None	None
bank_crypto_partnerships	Delete	Pass	None	None
bank_crypto_partnerships	Update	Pass	None	None
open_lending_protocols	Delete	Pass	None	None
open_lending_protocols	Update	Pass	None	None
crypto_loans	Delete	Pass	None	None
crypto_loans	Update	Pass	None	None
crypto_loan_payments	Delete	Pass	None	None
crypto_loan_payments	Update	Pass	None	None
lending_protocol_connections	Delete	Pass	None	None
lending_protocol_connections	Update	Fail	A foreign key constraint fails	Set NN attribute on crypto_account column to off so we can set a crypto account id to null
currencies	Delete	Pass	None	None
currencies	Update	Pass	None	None
exchange_rates	Delete	Pass	None	None
exchange_rates	Update	Pass	None	None
portfolio_exchange_rates	Delete	Pass	None	None
portfolio_exchange_rates	Update	Pass	None	None

atm_transactions	Delete	Pass	None	None
atm_transactions	Update	Pass	None	None
devices	Delete	Pass	None	None
devices	Update	Pass	None	None
security_devices	Delete	Pass	None	None
security_devices	Update	Pass	None	None
authentication_devices	Delete	Pass	None	None
authentication_devices	Update	Pass	None	None
io_devices	Delete	Pass	None	None
io_devices	Update	Pass	None	None
transaction_devices	Delete	Pass	None	None
transaction_devices	Update	Pass	None	None
iot_contextaware_sensors	Delete	Pass	None	None
iot_contextaware_sensors	Update	Pass	None	None
iot_temperature_sensors	Delete	Pass	None	None
iot_temperature_sensors	Update	Pass	None	None
cameras	Delete	Pass	None	None
cameras	Update	Pass	None	None
card_reader_devices	Delete	Pass	None	None
card_reader_devices	Update	Pass	None	None
fingerprint_scanners	Delete	Pass	None	None
fingerprint_scanners	Update	Pass	None	None
ada_headphone_jacks	Delete	Pass	None	None
ada_headphone_jacks	Update	Pass	None	None
speakers	Delete	Pass	None	None
speakers	Update	Pass	None	None
atm_displays	Delete	Pass	None	None

atm_displays	Update	Pass	None	None
keypads	Delete	Pass	None	None
keypads	Update	Pass	None	None
cash_dispensers	Delete	Pass	None	None
cash_dispensers	Update	Pass	None	None
deposit_check_slots	Delete	Pass	None	None
deposit_check_slots	Update	Pass	None	None
receipt_printers	Delete	Pass	None	None
receipt_printers	Update	Pass	None	None
receipt_roll_paper	Delete	Pass	None	None
receipt_roll_paper	Update	Pass	None	None
ink_cartridges	Delete	Pass	None	None
ink_cartridges	Update	Pass	None	None
kubeCDNs	Delete	Pass	None	None
kubeCDNs	Update	Pass	None	None
security_media_content	Delete	Pass	None	None
security_media_content	Update	Pass	None	None
context_aware_sensor_data	Delete	Pass	None	None
context_aware_sensor_data	Update	Pass	None	None
temperature_sensor_data	Delete	Pass	None	None
temperature_sensor_data	Update	Pass	None	None
camera_footage	Delete	Pass	None	None
camera_footage	Update	Pass	None	None
sessions	Delete	Pass	None	None
sessions	Update	Pass	None	None

atm_admins	Delete	Pass	None	None
atm_admins	Update	Pass	None	None
atm_maintenance	Delete	Pass	None	None
atm_maintenance	Update	Fail	Column ‘admin_id’ cannot be null	Turn off the NN attribute for ‘admin_id’ column
transaction_actions	Delete	Pass	None	None
transaction_actions	Update	Pass	None	None
atm_transaction_info	Delete	Pass	None	None
atm_transaction_info	Update	Pass	None	None
crypto_permissions	Delete	Pass	None	None
crypto_permissions	Update	Pass	None	None
checking_accounts	Delete	Pass	None	None
checking_acconts	Update	Pass	None	None
savings_accounts	Delete	Pass	None	None
savings_accounts	Update	Pass	None	None
checking_bank_permissions	Delete	Fail	A foreign key constraint fails	checking_account_id must already be referenced, so use another checking account id
checking_bank_permissions	Update	Pass	None	None
savings_bank_permissions	Delete	Pass	None	None
savings_bank_permissions	Update	Pass	None	None
account_notifications	Delete	Pass	None	None
account_notifications	Update	Pass	None	None
fraud_claims	Delete	Pass	None	None
fraud_claims	Update	Pass	None	None