# Use Case Details

Final versions of the requirements documents used for walk-throughs with the main stakeholders.

IDs can be anything from consecutive numbers to meaningful acronyms about what kind of use case this is and what system it is a part of.

## UC1 Withdraw Cash

Repeat this template for each use case to be documented.

### General info

#### Description:

Informal use case or short summary that can be extracted and used in the use case summary page.

Actor goes to machine, inserts card, enters PIN, selects Withdraw Cash, enters amount.   
Bank verifies amount and actor gets cash and receipt.

#### Actors: User, Maintenance

The roles that can initiate this use case. It does not include participants.

#### Supporting roles/systems (other stakeholders or participants): Bank

Sometimes called (supporting actors) and are other “actors” that are involved during the course of events. These can also be called interfaces.

#### Type: System

Options: system | business | blended (business tasks interspersed with system tasks). Include the system name if several are used.)

#### Pre-conditions:

Rules for beginning this use case: state of system prevents usage, must be testable. Or in a business use case, this must be the current state that has met a goal through another use case that this use case can now follow.

* Internet connection good.
* The ATM must have **Account\* Maximum business withdrawal amount** in the cash drawer.

### Scope info

#### Level: Goal

Options: goal | partial goal | group of goals | group of partial goals. Goal level will comprise 90% of the use cases.

#### Includes:

The use cases that are extracted out of this use case and given a special name, so they can be reused. They are required to be a part of this use case. This use case can be considered a grouped use case if it includes one of a group of partial goal use cases.

#### Included in:

The use case(s) that uses this one as a necessary part of it.

#### Use cases grouped by this ID:

If it doesn’t have an included group above, then it will be a category for several use cases.

#### Grouped by: GUC1 Do Transaction

The group that has others like this one.

### Tracking info

#### Author: ATM class of 3/15/2021

#### Date created: 3/19/2021

#### Person and Date revised: ATM class of 3/15/2021 on 3/19/2021

### Project info

#### Design constraints: ATM chassis Bazinga 278385495

Pure business term descriptions are hard to write. Constraints describe Any kind of policy, infrastructure, time, location, budget, hardware, or software that must be accommodated by this process e.g. web site is required, SQL Server is required, location must be…, hardware must be…

#### Priority: 9 - High

Priority will be by goal level or higher. Partial goal use cases will take their priority from the highest level that it is included in.

#### Value to actor: Provides convenience by having 24/7 availability.

#### Value to sponsor: Provides service outside of bank business hours saves overhead and labor costs.

Value must be specified by the requirement that it is supporting for the business.

#### Sponsor: Mr. Big

Who is accountable for this use case being delivered successfully?

### Course of Events

The sequence of tasks in conversation format between actor and system. For best linking to other steps, start each number with a system task except for the trigger. Combine actor responses to system events when well. Rules are placed under the task unless they can be reused and then they are referenced and placed in a separate file.

The number of tasks per number is usually small and starts with the system or the role. Tasks are individually stated so the system/role can do multiple things but in separate sentences. The last task will prepare the state of the system so that this use case can be performed again. There will be no condition statements to branch into two separate use cases. There may be a section that is removed to a named partial use case and called an <<include>> to shorten the detailed use case.

References that can be used here to document anything other than a functional requirement are:

* **T#** - Text file item number – used for error messages and small prompts mostly on forms.
* **D#** - Design file item number – used for web pages, full screen menus, etc.
* **R#** - Report file item number – used for printed or on-screen report formats
* **\* -**  a Data Dictionary item – used to refer to data description and validation so that the detail doesn’t have to be specified here. Also bolded and colored is good.
* **Rule#** - Rule file item number – used to refer to process rules. Generally, this will follow one path only and another use case will pick up any other options. Some data validation rules find their way here but should be collected under the Data Dictionary. Unnumbered rules are not reusable and will just be defined below their functional requirement.

1. The use case starts when the actor puts card in the slot.
2. The system reads the **Card**\* magnetic strip info.
3. The system prompts for a PIN number (D#1 - PIN entry). The actor enters their PIN.
   1. **RULE** - four digits only
4. The system sends **Card\*** info and PIN and requests that the bank validates the PIN. The bank returns **Account**\* information. The system creates a **Log Entry\*** of the communication. The system prompts for a selection from the main menu (D#2 - Main menu).
5. The actor selects Withdraw Cash.
6. The system prompts the actor to select account (D#3 - **Account**\* selection). The actor selects **Account**\*.
7. The system prompts the actor for **Transaction**\* amount to withdraw (D#4 - Withdrawal amount). The actor inputs withdrawal amount and confirms.
8. The system validates the amount.
   1. **RULE** - Available funds - Amount requested is equal to or less than **Account**\* available funds.
   2. **RULE** - $20 increment - Amount requested is in increments of $20.
   3. **RULE** - Maximum daily withdrawal amount - Amount request is equal to or less than **Account**\* Maximum withdrawal amount.
9. The system requests the bank to record the transaction. The bank confirms. The system creates a **Log Entry\*** of the communication.
10. The system dispenses cash. The system prompts the actor to take cash. (D#5 - Take cash) The actor retrieves the cash. The system creates a **Log Entry\*** of the cash dispensing.
11. The system prompts actor for another transaction (D#6 - Another transaction). The actor declines.
12. The system prompts actor for a receipt (D#7 - Do you want a receipt). The actor confirms getting a receipt.
13. The system prints the receipt (R#1 Receipt). The system prompts actor to take receipt (D#8 - Take receipt) The actor retrieves the receipt.
14. The system ejects the card. The system prompts actor to take card (D#9 - Take card) The actor retrieves the card.
15. The system subtracts withdrawal amount from **ATM**\* Cash on Hand.
16. The system returns to welcome screen. (D#0 Welcome screen)

### Alternate flows (errors, exceptions)

The error flows are where a rule is broken, or something interrupts the normal “happy path” of the course of events. This often is during communication or other type of I/O.

**Invalid PIN** (#4) - The use case continues at #3 for two more times. If PIN is still invalid, the ATM locks the account and keeps the card. Then the use case ends with a prompt of instructions for returning card.

**Available funds** (#8a) - The system prompts the user that they don't have enough funds. The use case continues at #7.

**$20 increments** (#8b) - almost same as above.

**Cancel button** (2,4-6…) - prompt and cancel transaction. The use case continues at #14.

### Alternate flows (extension points, optional)

An exception to branching is when there is an optional <<extends>> of a partial use case. But the use case returns to where the option was taken.

**Look up available balance** (#7) - The system prompts for getting balance. The actor confirms and the system shows the balance.

### Post-conditions

What are your tests that tell you that this is a successful completion of a use case? It may be a repetition of one of the tasks or a file or document that has been completed. But there are minimal ways to complete the goal and there are very excellent ways to complete it. Put both down. Some people use MoSCow – must have, should have, could have instead of min and max conditions.

The actor has retrieved cash, receipt, and card.

The bank has processed the transaction.

### Notes/ Special Requirements

Any kind of quality, capacity, security, availability, disaster recovery information that is because of this use case. Maybe you also have ideas about design, or people who need to be checked with, etc.

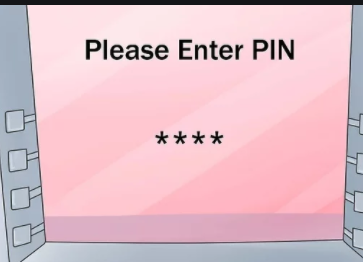
Braille keyboard.

ATM can speak the prompts.

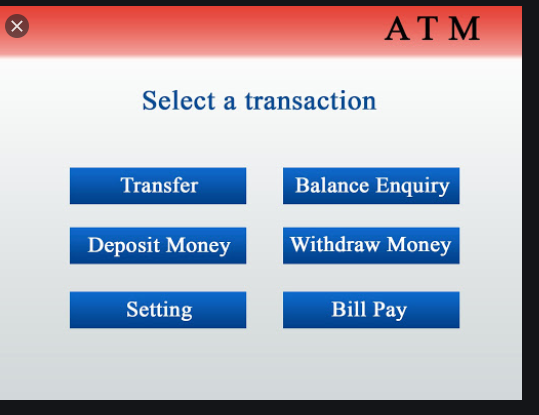
CCTV.

# Screen and report designs

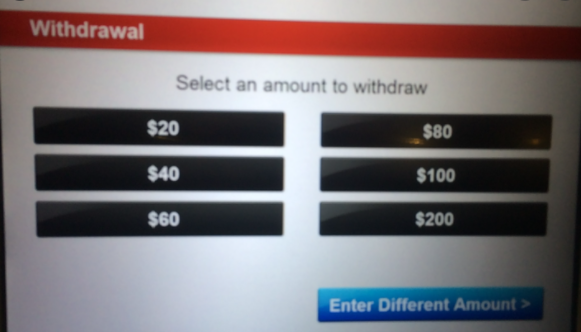
D#1 - PIN entry



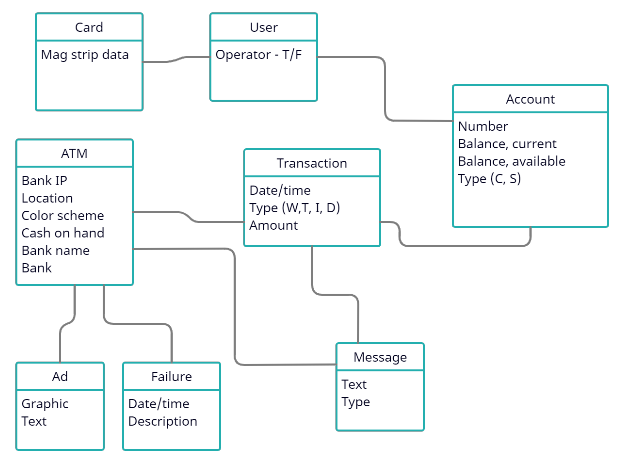
D#2 - Main menu



D#4 - Withdrawal amount



# Data dictionary

ATM

* + Bank IP
  + Location
  + Color scheme
  + \*Failure
  + \*Ad
* Ad
  + Graphic
  + Text
* Failure
  + Date/time
  + Description
* User
  + Operator (T/F?)
  + \*Account
* Account
  + Number
  + Balance
  + Type (checking | savings)
  + Maximum daily withdrawal amount -
    - Day is midnight to midnight.
    - Maximum business account is $5000.
    - Maximum personal account is $300.
* Transaction
  + Date
  + Time
  + Type (Withdrawal | Transfer | Inquiry | Deposit)
  + Amount
* Card
  + \*User
  + Magnetic strip data
* Log entry