

## CS 255 Sample Business Requirements Document for an ATM

The following is a sample of the business requirements document for an ATM system. Use this as an example of the types of requirements you should be collecting for each of the categories. It is important to know that this is not an exhaustive list of all functionality for an ATM system, but just a sample.

### System Components and Design

#### Purpose

- This project is for one of our clients, a banking institution. The purpose is to make their customer services easier and more efficient as well as to provide customers with access to their accounts.

#### System Background

- The client asked us to build a system that enables customers to withdraw, deposit, and check account balances using an ATM card.
- The system, called an ATM or automated teller machine, will provide banking services to customers.
- The system will consist of a large machine that is connected to the central bank via internet.

#### Objectives and Goals

- The system will provide the user with the following functions:
  - Withdraw money
  - Deposit money
  - Print transactions
  - Check balance

### Requirements

#### Nonfunctional Requirements

- ATM must lock card after three failed attempts to enter PIN.

#### Performance Requirements

- The time it takes to connect to the central bank, retrieve information, and display it to the customer should not exceed 10 seconds.
- Updates to the ATM should be pushed from a central bank system with the presence of a technician for backup.

Rationale: Performance requirements capture the aspects of the system such as the speed of different functions, storage capacity, battery (if applicable), and updates. For an ATM, it would be important to have quick transactions to avoid user frustration and long lines. Since an ATM deals with sensitive financial information, updates should be made in a very controlled way to make sure that the updates are verified and that someone is on hand in case there are any issues.

### Platform Constraints

- The ATM will run a Windows operating system.
- No database will exist in the ATM, but it will connect to the central bank that has all the proper APIs necessary to perform the needed functions.

**NOTE:** API stands for application programming interface. It connects a client (in this case an ATM) with a larger server (in this case the central bank).

Rationale: Platform constraints include limitations and restrictions due to the operating system(s) the application will run on as well as any “back-end” tools such as a database. For example, in the ATM case, we are running Windows on a basic machine. This means that we are limited by memory size and the hardware being used. Additionally, an ATM is not a multitasking system; the application will serve one user at a time.

### Accuracy and Precision

- The ATM should distinguish Gold users from regular users. Gold users are not charged a fee if they use other bank ATM cards at our ATM machines. Furthermore, the upper limit (maximum withdrawable amount per day) for Gold users is twice what it is for non-Gold users.
- The ATM should identify a technician user when the technician enters the proper PIN and uses the proper (special) card.

Rationale: Accuracy and precision detail distinctions between different types of users, what type of input validation is necessary for different functions, and the types of situations where an admin would need to be notified. In this case, an ATM might have different “levels” of users, such as a “Gold” user. There is some type of verification for a technician user, who would have different permissions from a regular user.

### Adaptability

- The system should adapt to weekly updates received from Microsoft. The ATM should have no problems downloading and installing Windows updates as they are released from Microsoft.

Rationale: Adaptability describes how the system adapts to different changes. An ATM should not be very adaptable because that could open security holes in the system, allowing hackers to get access to the machine. That being said, the ATM should have basic adaptability features in that it should be able to run normally when Windows pushes new updates.

### Security

- The system should allow user to try the PIN three times. If the user fails for the third time, the ATM will HOLD the user’s card. The user must retrieve it from the closest bank branch.
- The ATM should inform the bank branch of existing ATM cards it’s holding so the bank can send someone to pick them up.
- The ATM should use PRESS BUTTONS with a cover on top of the buttons so others do **not** see PIN.
- The ATM should have a camera to record user’s activity and face as user is making transactions.

Rationale: Security is extremely important in any system. Consider logins, passwords, any data exchanges that might occur, how to deal with multiple login attempts, and so on. For a banking system,

a low number of login attempts is appropriate. Putting a hold on the card could prevent identity theft, since a person would have to show up at the bank to retrieve their card. Notice that some of the security concerns are physical (how to “cover” the buttons, the camera) as well as in the software.

### Functional Requirements

- The system shall determine customer ID/information from the card reader.
- The system shall show customer the following options: Deposit, Withdraw, Display Balance.
- The system shall present the customer with the option to print receipt, email receipt, or include no receipt after every transaction.
- The system shall allow only **one** transaction per PIN. If more than one transaction is required, the PIN must be entered again.
- The system shall produce a technical report when a special card and proper PIN are entered by a technician.

Rationale: Functional requirements discuss what the system does. Think about all the things different users need to do with the system. Make sure that you have accounted for the actions of different types of users.

### User Interface

- The device consists of a display screen (which is also a touch screen), a keypad, a card reader, and a printing device (to print receipts).
- The keypad (which can be touchscreen or digital) consists of the numbers 0 to 9, CLEAR, DELETE, CANCEL, and OK.
- The user interface consists of basic messages such as the following:
  - Enter PIN
  - Enter Amount
- The keyboard should have Braille on the keys to help blind users use the ATM.
- The ATM should communicate with user via screen and audio (special beeps).

Rationale: Think about what the interface will look like. In this case, there is a digital display as well as hardware (the actual machine buttons). Different types of users were considered to make the machine more accessible—for example, simple language (Enter PIN, Enter Amount) and Braille on buttons. Other considerations might include different language options and so on.

### Assumptions

- The user has a valid, functioning (not damaged) ATM card to use the system.
- Electricity is available at location.
- Internet is available 24/7 for ATM to communicate with central bank.

Rationale: Assumptions are things being assumed about the system design. For example, all of our functionality depends on a user having a working ATM card. If it doesn't, what would happen? The ATM depends on electricity and internet—without these things, the ATM will not work. These things are reasonable assumptions; we just want to make sure that we capture them.

### **Limitations**

- The system is not voice activated.
- The system will stop functioning once electricity is lost.
- The system will stop functioning once it runs out of money.
- The system will stop functioning once internet or connection to central bank is lost.

Rationale: These limitations describe things the system cannot do. The system is pretty reasonable. The feature of voice activation is nice and might make the ATM more easily accessible for some users who have limited dexterity or mobility. It may not be possible due to the project scope or budget in this particular case.