NAME	MUHAMMAD ZEESHAN
ID#	14882
SECTION#	B
DEPARTEMENT	BS(SE)
SEMESTER	4TH
PAPER	SOFTWARE ENGINEERING
CLASS TIMING	MON: 11 TO 2

INSTRUCTOR:

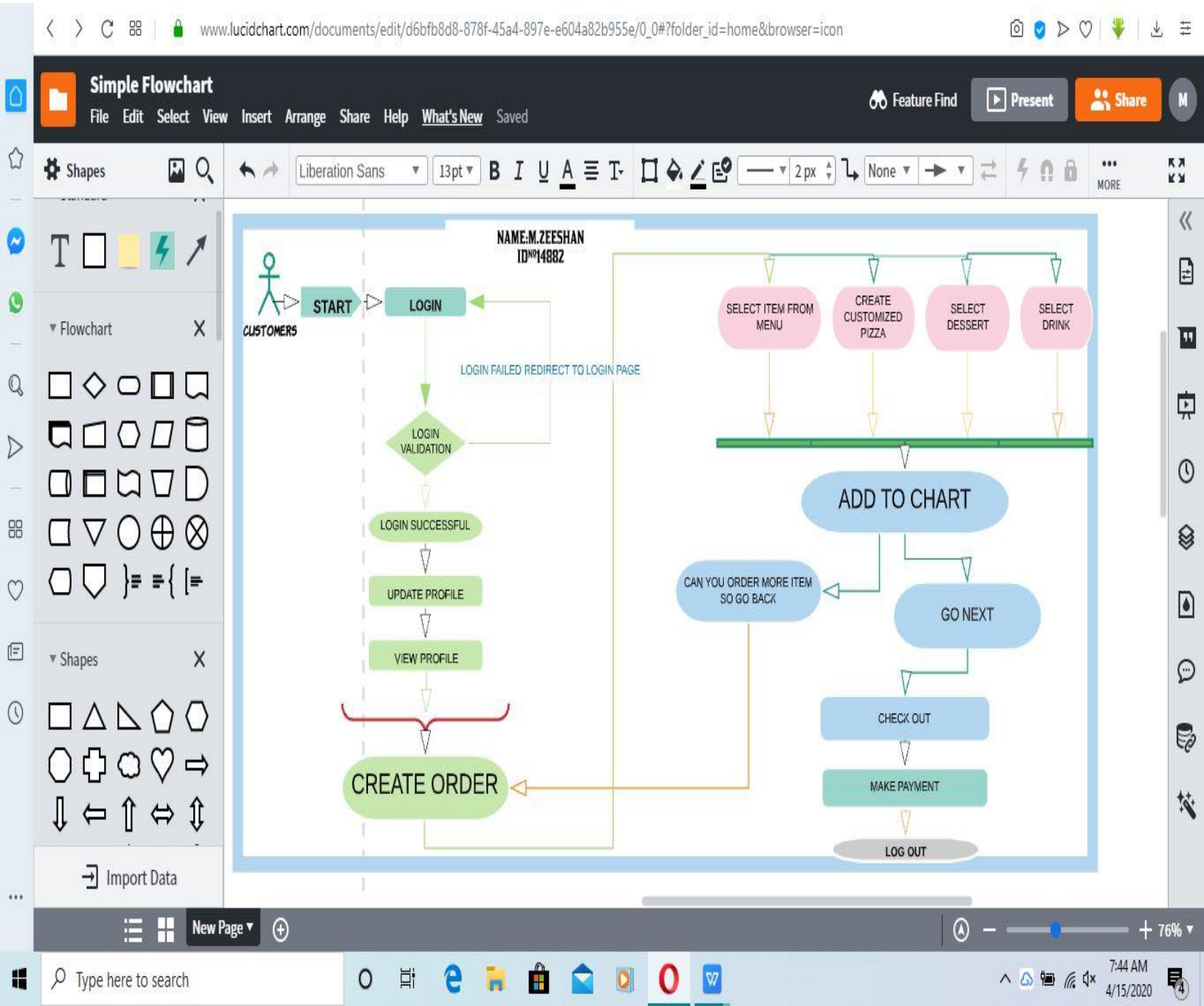
GHASSAN HUSNAIN

QUESTION#1

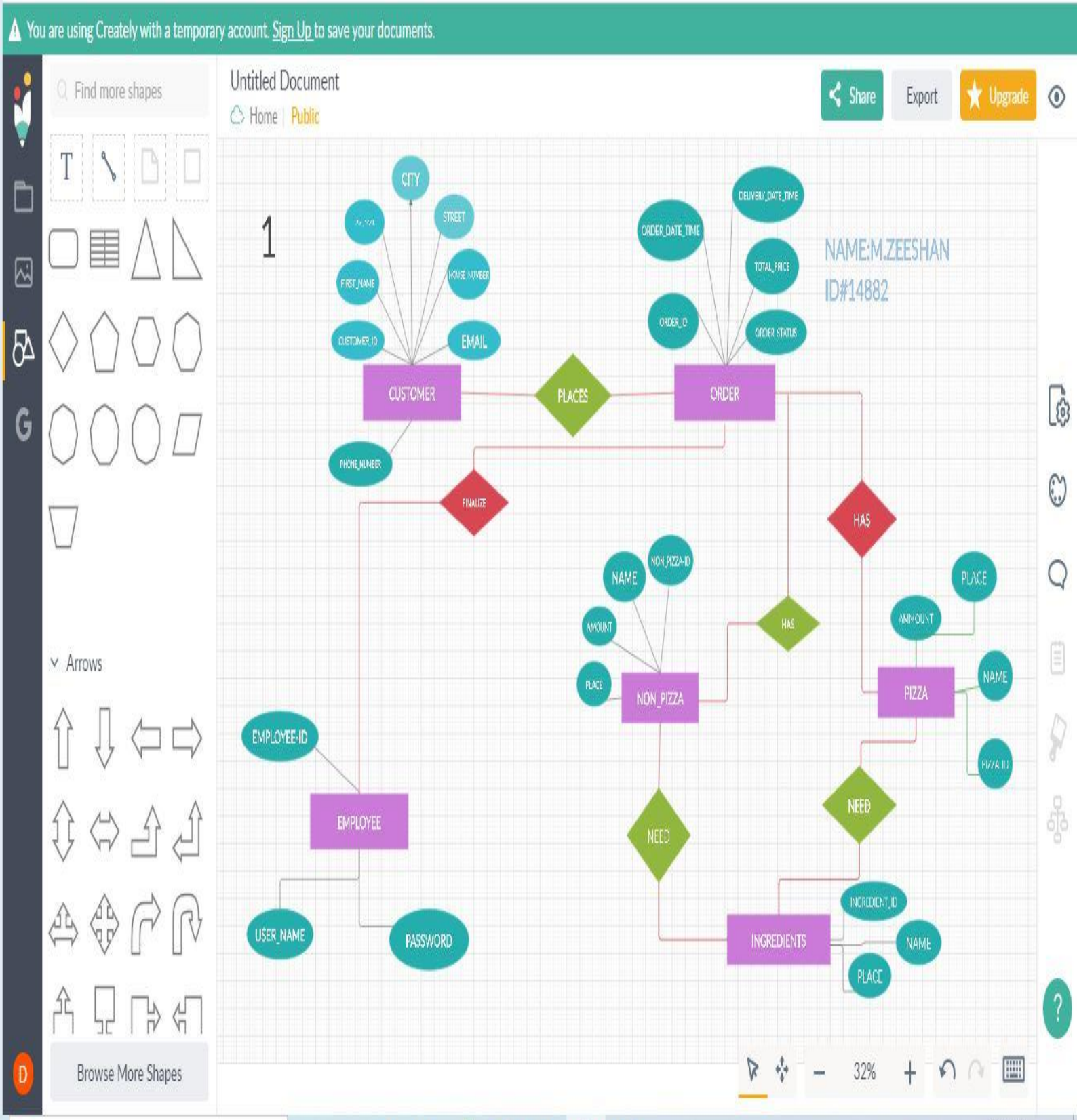
THE PIZZA ORDERING SYSTEM

the pizza ordering system the pizza ordering system allows the user of a web browser to order pizza for home delivery to place an order a shopper searches to find items to purchase ads, items one at a time to a shopping cart and possibly searches again for more items when all items have been chosen, the shopper provides a delivery address. if not paying with cash, the shopper also provides a credit card information?

THE PIZZA ORDERING SYSTEM:



ANOTHER DIAGRAM OF ERD



➔This **Use Case Diagram** is a graphic depiction of the interactions among the elements of **Pizza Ordering System**. It represents the methodology **used** in **system** analysis to identify, clarify, and organize **system** requirements of **Pizza Ordering System**.

➔This is the UML sequence **diagram** of **Pizza Ordering System** which shows the interaction between the objects of Payments, Customer, **Order**, **Pizza.**

➔A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

QUESTION#2

Suggest how an engineer responsible for drawing up a system requirements specification might keep track of the relationship between functional and non-functional requirements?

FUNCTIONAL REQUIREMENTS:

A functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior and outputs. Functional requirements may be calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish…..

EXAMPLES FUNCTIONAL REQUIREMENTS:

- ***Business Rules.***
- ***Transaction corrections, adjustments, and cancellations.***
- ***Administrative functions.***
- ***Authentication.***
- ***Authorization levels***
- ***Audit Tracking.***
- ***External Interfaces.***
- ***Certification requirements***.

NON-FUNCTIONAL REQUIREMENTS:

Non-Functional requirement specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. In general, Non-Functional requirements define how a system is supposed to be. Non-Functional requirements are often called qualities of a system.

EXAMPLES OF NON-FUNCTIONAL REQUIREMENTS

- ***Volumetric.***
- ***Scalability.***
- ***Capacity.***
- ***Availability.***
- ***Reliability.***
- ***Recoverability.***

(OR)

the classic tool is a requirements traceability (sic) matrix. There are even tools dedicated to just tracking requirements, Doors is a great example of that. Fundamentally you could still use an RTM, because in reality it's just a spreadsheet and numbering scheme you maintain to track relationships..

In Agile development, the index cards representing user stories must be tied to requirements you isolate from them, creating additional work that doesn't provide as much value in terms of testing - these stories are intended to mostly need only a few tests and be relatively atomic. What I mean by that is this: you might as well perform the testing before trying to trace, defining "requirements" as late as possible while still developing tests as early as possible.

- The user needs to search for the candidate list for the interview.
- It is a functional requirement.
- That the search should return all the list of candidates who are attending the interview.
- It is a non-functional requirement.
- Therefore, it helps the engineer to avoid overlap and that relates to each other.
- And it keeps track the relationships between functional and non-functional requirements. It is

QUESTION#3

To reduce costs and the environment impact of computing. Your company decide to close the number of offices…….?

It is difficult to introduce agile methods into large companies for a number of reasons:

- Project managers who don't have experience of agile methods may be reluctant to accept the risk of a new approach, as they do not know how this will affect their particular projects

- Large organizations often have quality procedures and standards that all projects are expected to follow because of their bureaucratic nature, these are likely to be incompatible with agile methods. Sometimes, these are supported by software tools (example requirement management tools) and the use of these tools is mandated for all projects.

- Agile methods seem to work best when team members have a relatively high skill level. However within large organizations there are likely to be a wide range of skills and abilities. And people with lower skill levels may not be effective team members in agile processes.

- There may be cultural resistance to agile methods, especially in those organization that have a long history of using conventional systems engineering processes.

➔ Change management and testing procedures are example of company procedures that may not be compactable with agile methods. Change management is the process of controlling changes to a system so that impact of changes is predictable and costs are controlled. All changes have to be approved in advance before they are made and this conflicts with the notion of refactoring. In XP, any developer can improve any code without getting external approval for large systems there are also testing standards where a system build is handed over to an external testing team. This may conflict with the test first and test often approaches used in XP.

QUESTION#4

Discover/difficulties ambiguities or omission of the following statement of requirements for part of a ticket-issuing system.

Discover ambiguities or omissions in the following statement of requirements for part of ticket-issuing systems: An automated ticket-issuing system sells rail tickets. Users select their destination and input a credit card and a personal identification number. The rail ticket is issued and their credit card account charged. When the user presses the start button, a menu display of potential destinations is activated, along with a message to the user to select destination. Once a destination has been selected, users are requested to input their credit card. Its validity is checked and the user is then requested to input a personal identifier. When the credit transaction has been validated, the ticket is issued?

- Can a customer buy several tickets for the same destination together or must they be bought one at a time?
- Can customers cancel a request if a mistake has been made?
- How should the system respond if an invalid card is input?
- What happens if customers try to put their card in before selecting destination (as they would in ATM machines)?
- Must the user press the start button again if they wish to buy another ticket to a different destination?
- Should the system only sell tickets between the station where the machine is situated and direct connections or should it include all possible destinations?

- Give customer a rail ticket, and charge credit account accordingly.

- Determine customer's destination, calculate the charge for the Trip, and charge the customer the appropriate amount. If charge is complete, print the ticket, otherwise, print no ticket, and report error to customer

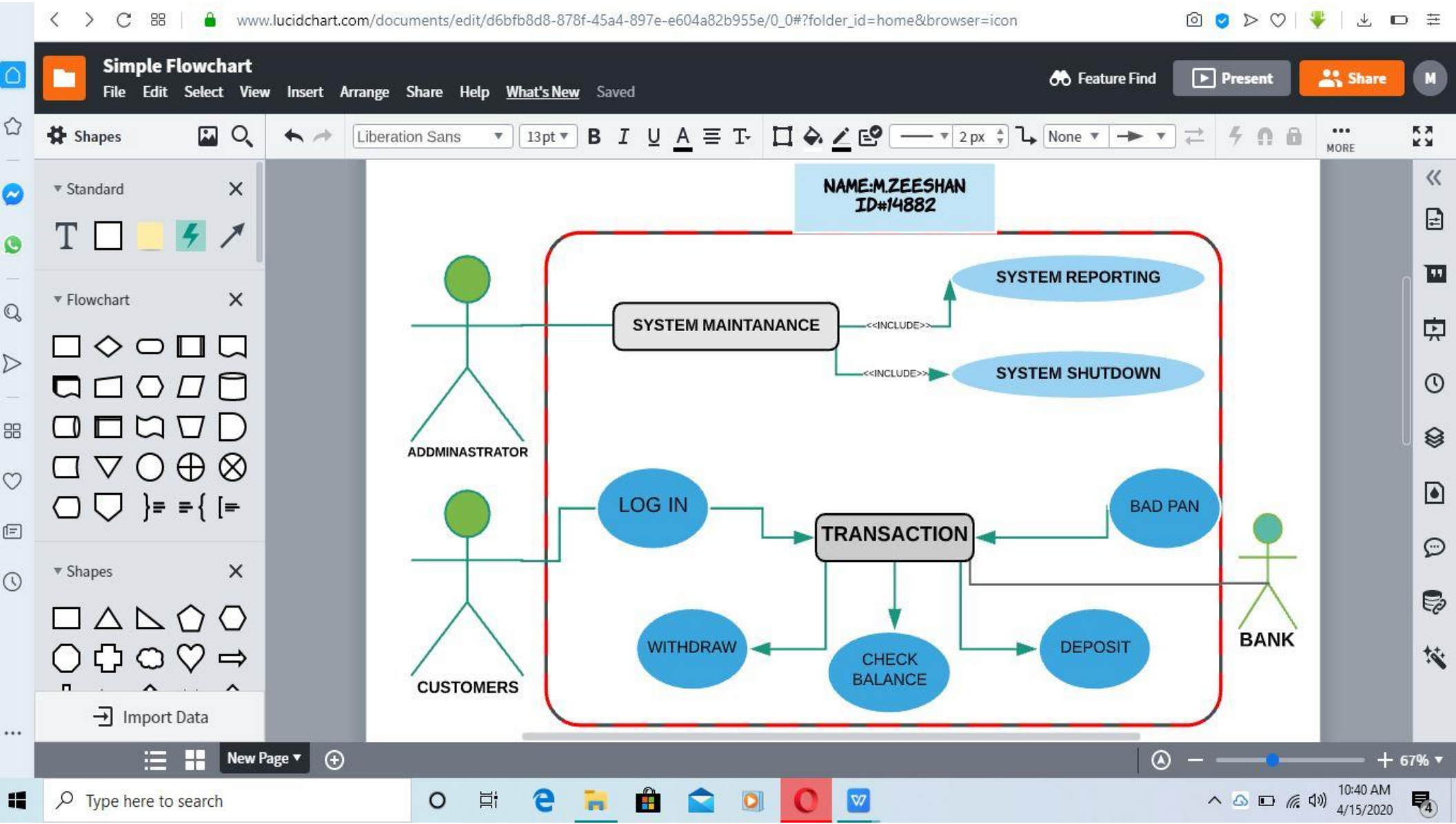
(OR)

- If a user want to buy multiple tickets, user doesn't have option.
- If user want to cancel tickets, user doesn't have option.
- Users cannot buy tickets with cash.
- There was ambiguity in how potential destinations is activated.

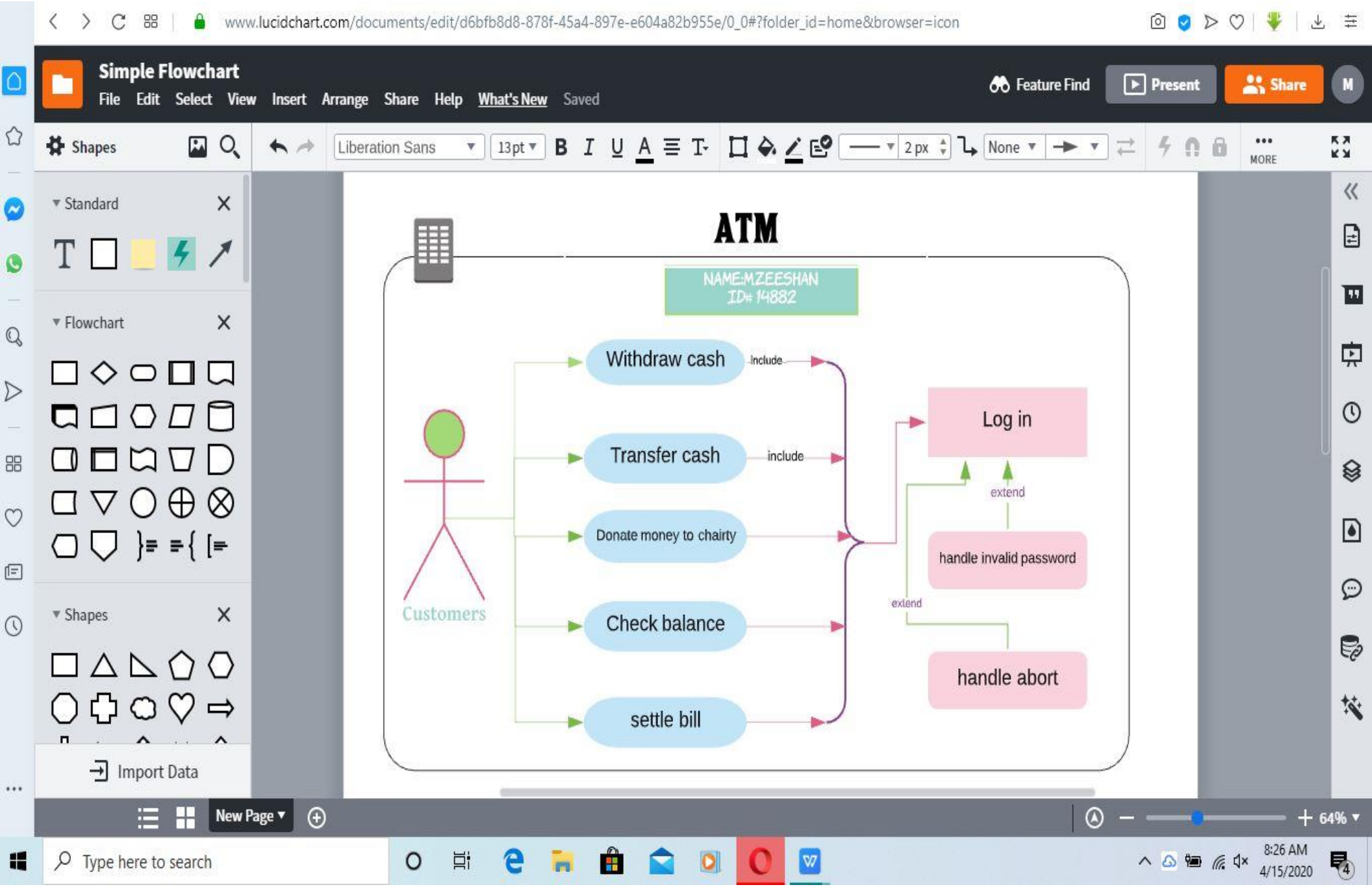
QUESTION#5

Using your knowledge of how an ATM is used, develop a set of use cases that could serve as a basis for understanding the requirements for an ATM system?

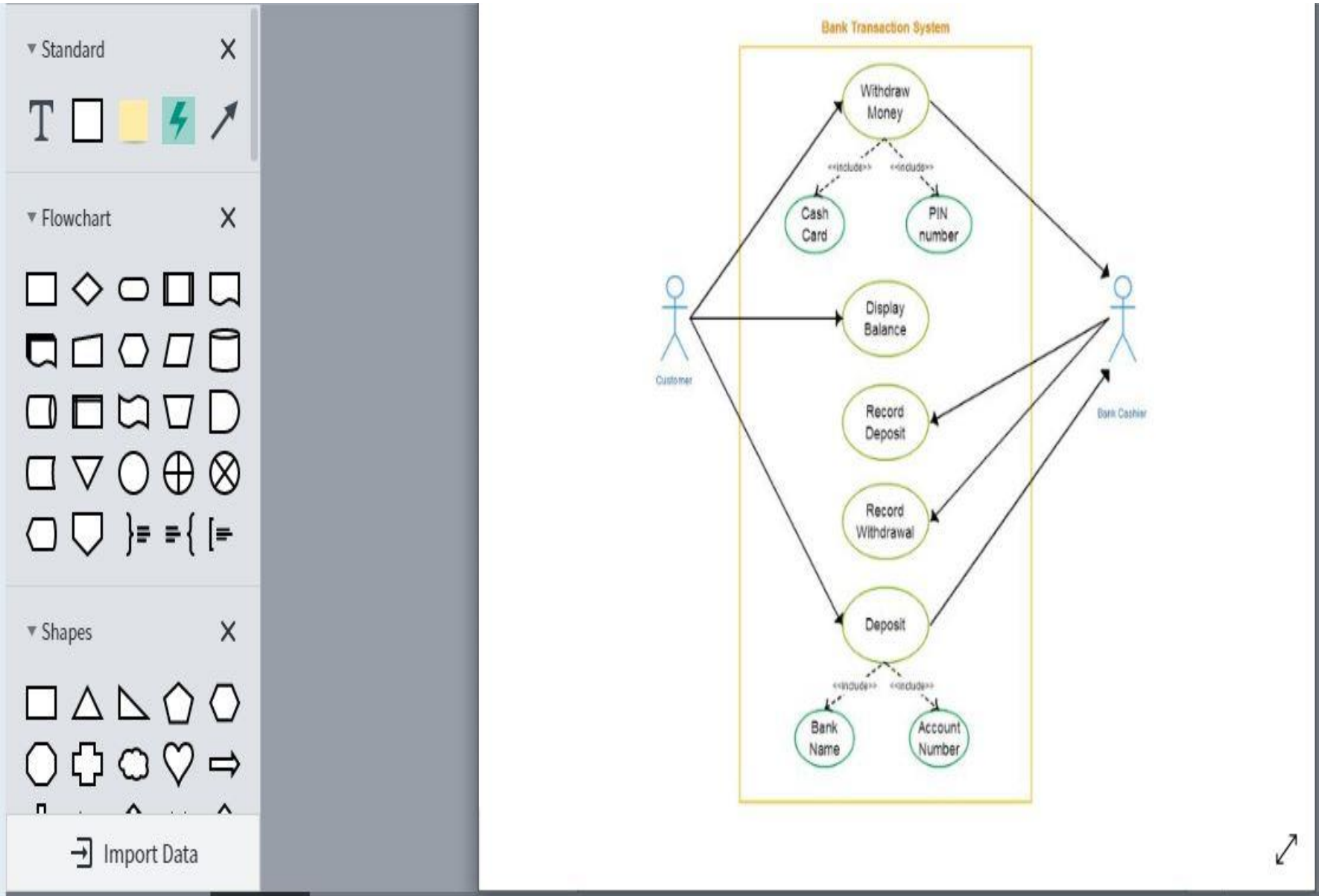
ATM SYSTEM



➔ANOTHER DIAGRAM OF ATM:



BALANCE TRANSACTION:



➔USES CASES FOR ATM DIAGRAM:

➔Transaction Use Case:

A transaction use case is started within a session when the customer chooses a transaction type from a menu of options. The customer will be asked to furnish appropriate details (e.g. account(s) involved, amount). The transaction will then be sent to the bank, along with information from the customer's card and the PIN the customer entered.

➔Withdrawal Transaction Use Case:

A withdrawal transaction asks the customer to choose a type of account to withdraw from (e.g. checking) from a menu of possible accounts, and to choose a dollar amount from a menu of possible amounts. The system verifies that it has sufficient money on hand to satisfy the request before sending the transaction to the bank. (If not, the customer is informed and asked to enter a different amount.) If the transaction is approved by the bank, the appropriate amount of cash is dispensed by the machine before it issues a receipt. (The dispensing of cash is also recorded in the ATM's log.)A withdrawal transaction can be cancelled by the customer pressing the Cancel key any time prior to choosing the dollar amount.

➔Deposit Transaction Use Case:

A deposit transaction asks the customer to choose a type of account to deposit to (e.g. checking) from a menu of possible accounts, and to type in a dollar amount on the keyboard. The transaction is initially sent to the bank to verify that the ATM can accept a deposit from this customer to this account. If the transaction is approved, the machine accepts an envelope from the customer containing cash and/or checks before it issues a receipt. Once the envelope has been received, a second message is sent to the bank, to confirm that the bank can credit the customer's account - contingent on manual verification of the deposit envelope contents by an operator later. (The receipt of an envelope is also recorded in the ATM's log.)

➔Transfer Transaction Use Case:

A transfer transaction asks the customer to choose a type of account to transfer from (e.g. checking) from a menu of possible accounts, to choose a different account to transfer to, and to type in a dollar amount on the keyboard. No further action is required once the transaction is approved by the bank before printing the receiptA transfer transaction can be cancelled by the customer pressing the Cancel key any time prior to entering a dollar amount.

Inquiry Transaction Use Case:

An inquiry transaction asks the customer to choose a type of account to inquire about from a menu of possible accounts. No further action is required once the transaction is approved by the bank before printing the receipt.

An inquiry transaction can be cancelled by the customer pressing the Cancel key any time prior to choosing the account to inquire about.

➔Invalid PIN Extension

An invalid PIN extension is started from within a transaction when the bank reports that the customer's transaction is disapproved due to an invalid PIN. The customer is required to re-enter the PIN and the original request is sent to the bank again. If the bank now approves the transaction, or disapproves it for some other reason, the original use case is continued; otherwise the process of re-entering the PIN is repeated. Once the PIN is successfully re-entered, it is used for both the current transaction and all subsequent transactions in the session. If the customer fails three times to enter the correct PIN, the card is permanently retained, a screen is displayed informing the customer of this and suggesting he/she contact the bank, and the entire customer session is aborted.

If the customer presses cancel instead of re-entering a PIN, the original transaction is cancelled.

THANK YOU SIR

THE END