**Oriental Institute of Science & Technology, Bhopal**

**ATM SIMULATION**

A

#### Project Work

Submitted as Minor Project in Partial fulfillment for the award of Graduate Degree in Bachelor of Engineering in Computer Science & Engineering.

#### Submitted to

## RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA

**BHOPAL (M.P)**

****

Submitted By--

## Students name (Enrollment No)

Under the Guidance of

## Prof. ANUBHA PRAJAPATI

## **(Department of Computer Science & Engineering)**



Oriental Institute of Science & Technology, Bhopal

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**JAN-JUN 2021**

## 

***CERTIFICATE***

This is to certify that the project entitled **“ATM SIMULATION ”** being submitted by SRAJAL SHARMA student of VI SEMESTER, B.Tech in Computer Science& Engineering have done their work as MINOR PROJECT for Partial fulfillment of the B Tech degree from RGPV, Bhopal (M.P.) is a record of bonafide work carried

out by her under our supervision.

**ACKNOWLEDGEMENT**

I take this opportunity to express my gratitude to my Project Guide, Mr. H.S. Bhatia for his unwavering encouragement and support throughout this endeavor. His insight and expertise in this field motivated and supported me during the duration of this project. It is my privilege and honor to have worked under his supervision. His invaluable guidance and helpful discussion in every stage of this project really helped me in materialized this project. Without his constructive direction and invaluable advice, this work would not have been completed.

I would also like to take this opportunity to present my sincere regards to Mr. H.S Bhatia, Head of the department (Computer Engineering) Ambedkar Polytechnic Delhi: -110092, for the support provided by him during the entire duration of diploma course and especially for this thesis. My gratitude is also extended to all teaching and non-teaching staff for their un- wavering encouragement and support in my pursuit for academics.

**ABSTRACT**

Now-a-days ATM (Automatic Teller Machines) is commonly used. It not only provides basic transaction services (e.g.…withdraw, balance enquiry, mini statement, money transfer) but also provide additional services like bill payments, mobile recharges, examination fees payment. But most of these facilities are available only to the urban people and not to the rural people. The reason why these facilities are not available to the rural people is illiteracy and the difficulty they face in remembering the entire mechanism of handling an ATM machine. Our project aims at making the ATM machines more user friendly (especially for rural people). And we are doing this by constructing the simulator of real world ATM machine.

As we know that simulator is a computerized model of real world system that demonstrate the working of the real world system (on the basis of certain assumptions) and can also be used to perform new experiments before it is deployed on the real system. The technique by which we are going to make the simulator friendlier to rural people is by enabling the simulator to be controlled by speech (i.e. any regional language).

So, in our project we are constructing a simulator that depicts the real world ATM machines and have certain features which are as follows:

 Works in networked environment, to demonstrate the real world ATM working environment.

 Demonstrates the basic banking services provided by ATM machines, i.e.….withdraw, deposit, balance enquiry, money transfer and others.

 Also, demonstrate the additional services provided by today‟s ATM machines, i.e. electricity bill payment, telephone bill payment.

 Enabling the simulator to be controlled by speech is the experiment we are going to perform on the simulator.

**LIST OF FIGURES**

Fig.1.User Interface Diagram Of ATM Machine

Fig.2.Use Case Diagram Of ATM Machine

Fig.3. Activity Diagram For ATM

Fig.4. Sequence Diagram For ATM And Bank Interaction

Fig.5.Class Diagram For ATM

Fig.6.Er Diagram For ATM Simulator

Fig.7. Data Flow Diagram For ATM Simulator

Fig.8.Flowchart Of ATM Machine

Fig.9.Test Case Diagram For ATM Simulator

**LIST OF CONTENTS**

CHAPTER 1.Introduction..........................................................................................................................

CHAPTER 2.Literature Survey..................................................................................................................

CHAPTER 3 Process Model...........................................................................................................

CHAPTER 4. Design...........................................................................................................

CHAPTER 5.Technical Details..................................................................................................................

CHAPTER 6.Coding..................................................................................................................................

CHAPTER 7.Testing..................................................................................................................................

CHAPTER 8. Screen Layout………………………………………………………………………………………………………………..

CHAPTER 9.Future Work.........................................................................................................................

CHAPTER 10.Conclusion.........................................................................................................................

*BIBLIOGRAPHY*

**CHAPTER -1**

**INTRODUCTION**

The real world ATM is continuously developing day by day. More and more new and exciting features are incorporated in the machine as per customer’s increasing needs. In the earlier times ATM was only recognized for cash withdrawal or balance enquiry. But now a day various new features like cash deposits, various Bill payments, Mobile recharge etc are added which are attracting more and more customers. In spite of these advancements in the ATM world, the ATM is not successful in rural areas. Most of the people are illiterate and even can’t perform a simple transaction like cash withdrawal due to language boundations. The same problem exists with the blind people. Providing more and more language options is useless because they may not be able to read a text written on the screen. Thus the existing system although growing day by day and fulfilling more and more needs of customers fails in the case when simplicity and user friendliness is measured.

**AIM:**

The aim of this project is to simulate the real world ATM machine.

**STUDY OF CURRENT SYSTEM:**

A real world automated teller machine (ATM) is a computerized telecommunications device that provides the customers of a bank with access to financial transactions in a public space without the need for a human clerk or bank teller

On most modern ATMs, the customer is identified by inserting a plastic ATM card with a magnetic stripe or a plastic smartcard with a chip that contains a unique card number, Bank account number and some security information, such as an expiration date or CVC (CVV). Security is provided by the customer entering a personal identification number (PIN).

The real world ATM provides a keypad and a touch screen. There is cash dispensing machine from which cash out can be accomplished. The cash dispensing machine optionally owns a cash In mechanism (if cash deposit feature is enabled in ATM machine), from which user can deposit money into his account. For customer‟s simplicity various language options are supported (like English, Hindi etc).

Various features provided by a real world ATM system are as follows:-

 Cash Withdrawal

 Cash Deposit

 Money Transfer

 Balance Enquiry

 Bill Payments

**PROPOSED SYSTEM:**

The proposed system “Speech controlled ATM Simulator” provides all the basic as well as advanced features of a standard real world ATM system. Along with it to add more simplicity and to make the machine more user friendly two new concepts are added:-

-Speech Synthesis-The ATM system will now help the user by guiding him throughout the transaction by giving speech in his or her mother language.

-Speech Recognition- The ATM system will now be able to recognize spoken commands. The customer can now give voice commands in his mother language with the help of a microphone and the ATM machine recognizes those spoken commands and takes the appropriate action.

**FEASIBILITY STUDY:**

Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called a feasibility study. It determines whether a project should be taken or not. Feasibility study was undertaken within tight constraints of time and culminated in the written and amoral feasibility reports. The constants of the reports were used as a sound basis for the decision of proceeding, postponing or the cancellation of the project

.

Various types of feasibility are studies they are:-

1) Technical feasibility***: -*** This is concerned with the specifying equipment and the software that successfully satisfies the user requirement. The technical need of the system very considerably. In examining the technical feasibility the configuration of the system is given importance than to the actual hardware. As the present system devoid of computer assistant a complete revamping of the whole set-up will be required.

 Does the necessary technology exist to do with what is suggested & can it required?

 Does the propose requirement have the technical capacity to hold the data required to use the new system?

 Can the system we updated it developed?

 What new skill will be required to new technology do the existing staff members have these skills if not can they we trained in due course of time?

*2)* Operational Feasibility: **-** It is mainly related to the human organizational and political aspect. It involves the study of change to be brought about in the system, new skills required etc. In the present system there is no use of computer. The operational staff is having no idea about operation of the computer system. An introductory training of the staff is needed.

3) Economic Feasibility: **-** It is the most frequently used technique for the evaluation of the effectiveness of the proposed system. It involves the cost/benefit analysis. Economic feasibility is the most frequently used method for evaluating the effectiveness system. More commonly known is cost beneficial the procedure is to determine the benefits and savings that are expected from a system and compare them with costs. If benefit is more

than the costs then the decision are made to design is made to design and implement the system.

4)Social Feasibility**: -** It is the determination of whether a system will be accepted by the user or not. This examines the probability of the project being accepted directed by the user. While discussing with users it was observed that they are afraid of using a computer system. This might be due to the fact they belong to diverse background. A computer literacy program is very necessary before the installation and implementation of the system failing which it may lead to rejection of the system.

5.Management Feasibility***: -*** It determines whether the proposed system will be acceptable to the management or.

After studying and testing the above feasibilities we have found that our project stands firmly on them.

**OVERVIEW OF DOCUMENT:**

**PRIMARY GOAL OF DOCUMENT:**

The primary goal of the document is to provide the entire information of the project. This includes the need of the project, objectives of the project, and step-by-step implementation of SDLC (Software Development Life Cycle). In case of our project the documentation describes the following details:

 Current System.

 Problem with current system

 Proposed system

 Modeling performed (Use cases analysis, sequence diagram, state chart diagram and DFD)

 System Design(Class Diagrams)

 Testing Performed

 Future Enhancements

**DOCUMENT CONVENTIONS:**

Font : Times New Roman Style

Heading : Bold

Sub-heading : Bold

Text : Normal

Font Size

Heading : 16

Sub-heading : 14

Text : 12

Alignment : Justified

Line Spacing : 1.5 lines

**CHAPTER -2**

**LITERATURE SURVEY**

**WHAT IS ATM?**

Automated Teller Machine (ATM), device used by bank customers to process account transactions. Typically, a user inserts into the ATM a special plastic card that is encoded with information on a magnetic strip. The strip contains an identification code that is transmitted to the bank's central computer by modem. To prevent unauthorized transactions, a personal identification number (PIN) must also be entered by the user using a keypad. The computer then permits the ATM to complete the transaction; most machines can dispense cash, accept deposits, transfer funds, and provide information on account balances. Banks have formed cooperative, nationwide networks so that a customer of one bank can use an ATM of another for cash access.

**WHAT IS SPEECH RECOGNITION?-**

Speech recognition (also known as automatic speech recognition or computer speech recognition) converts spoken words to machine-readable input (for example, to key presses, using the binary code for a string of character codes). The term "voice recognition" is sometimes incorrectly used to refer to speech recognition, when actually referring to speaker recognition, which attempts to identify the person speaking, as opposed to what is being said. Confusingly, journalists and manufacturers of devices that use speech recognition for control commonly use the term Voice Recognition when they mean Speech Recognition.

Speech recognition is the process of converting an acoustic signal, captured by a microphone or a telephone, to a set of words. The recognized words can be the final results, as for applications such as commands & control, data entry, and document preparation.

Speech recognition applications include voice dialing (e.g., "Call home"), call routing (e.g., "I would like to make a collect call"), domotic appliance control and content-based spoken audio search (e.g., find a podcast where particular words were spoken), simple data entry (e.g., entering a credit card number), preparation of structured documents (e.g., a radiology report), speech-to-text processing (e.g., word processors or emails), and in aircraft cockpits (usually termed Direct Voice Input).

**WHAT IS SIMULATION?**

Simulation is the imitation of some real thing, state of affairs, or process. The act of simulating something generally entails representing certain key characteristics or behaviours of a selected physical or abstract system. A simulation is a representation of a situation with a similar but simpler model that can easily be manipulated to determine experimental results. It is basically an experiment run as a model of reality. Using a simulator or experimenting with a fictitious system can show the eventual real effects of a given situation.

Examples:

Physical or manual simulations: These simulations rely on the use of physical methods to create data that mimic the types of information required. Physical simulations are used in many contexts, including the modeling of natural systems, and manual systems. Physical methods, such as using a dice to simulate data have the disadvantage of being too time consuming.

Computer simulations: Computer programs based on mathematical models are used to manipulate elements of the experiment. This type of simulation can be applied in many ways. In telegraphic engineering, computer simulations generate system topology and traffic models that can be used for network performance management. Some computer exercises that use a computer program to generate data or information are TELPACK, CSIM19, and Minitab.

**WHAT IS ATM SIMULATOR?**

ATM-Simulator (ATM-SIM) is a computer-based trainer used for teaching individuals with intellectual disabilities how to use an automated teller machine (ATM) to access their personal bank accounts. In the pilot evaluation, a prototype system was developed and used to train nine adults with intellectual disabilities how to use a local ATM. Participants were pre-tested on their ability to use an ATM and then were trained using the ATM-SIM prototype. After a brief training period, participants were again tested on their capacity to operate the actual ATM. Computer simulation of ATM can be used effectively to teach a specific independent living skill to adults with mental retardation.

Simulation has been used effectively in many settings to teach skills that require both complex problem solving abilities (e.g., nuclear power plant operators) as well as specific rote tasks (e.g., switchboard operators). Recently the increased availability, decreased cost, and emerging technological advances associated with computers have made computer simulation a viable alternative teaching method in a variety of settings. As a result, computer simulation has been used increasingly as a cost-effective alternative to actual hands-on training. With technology, such as ATMs, in which simulations may need to be used in training, it is likely that computers can provide reasonably priced and realistic simulations that might provide for transfer of skills to community settings.

ATM-SIM is designed as a multimedia training tool that provides individuals with intellectual disabilities step-by-step instructions for learning how to use an ATM. Each step in the process, from entering the ATM card, keying in an access code, selecting the desired transaction and completing the transaction was simulated on the computer and presented to the user with step-by-step visual and audio cues.

**WHAT IS SPEECH CONTROLLED ATM SIMULATOR?:**

Despite the success and widespread use of Automatic Teller Machines (ATMs), a significant proportion of bank customers can not or will not use them, or experience difficulties in their interactions. Speech technology has been suggested as a means by which non-users might be encouraged to use ATMs, while simultaneously improving usability for all. The potential advantages of speech interfaces include hands-free and eyes-free use for physically- and visually impaired users, and improved ease and speed of use through increased 'naturalness' of the interaction. This study investigated user attitudes to the concepts of a speech-based ATM, via large-scale survey and a series of focus groups. Objective performance was also considered in user trials with a prototype speech-driven ATM. The idea of using speech for ATM transactions led to a number of concerns.

**WHAT IS SPEECH CONTROLLED ATM SIMULATOR:**

Despite the success and widespread use of Automatic Teller Machines (ATMs), a significant proportion of bank customers can not or will not use them, or experience difficulties in their interactions. Speech technology has been suggested as a means by which non-users might be encouraged to use ATMs, while simultaneously improving usability for all. The potential advantages of speech interfaces include hands-free and eyes-free use for physically- and visually impaired users, and improved ease and speed of use through increased 'naturalness' of the interaction. This study investigated user attitudes to the concepts of a speech-based ATM, via large-scale survey and a series of focus groups. Objective performance was also considered in user trials with a prototype speech-driven ATM. The idea of using speech for ATM transactions led to a number of concerns. Privacy (the concern over one's personal financial details being overheard) and security (the fear of potential attackers hearing the user withdraw cash) were the major reasons given. The user trials confirmed that possible solutions, such as the adoption of a hood over the ATM or the use of a telephone handset as the speech input/output device, were ineffective.

As per literature survey we have learned what an ATM is, what a simulation is and why we simulate any real world system. So before we move further we must consider the following aspects regarding ATM Simulator.

The aspects are as follows-

 Security: The security is the main concern in case of ATM machines. So this point must also be considered in ATM Simulator. So secured handling of PIN (Personal Identification Number) and secured manipulation of other information across the network must also take place.

 Functionality: The ATM Simulator must include all the possible basic and advanced features that can be provided by a modern ATM machine.

 Usability: Unlike the modern ATM machines the simulator should be useful to a wide range of users and the users must use it without any assistance. The simulator must be user friendly and must support multiple languages.

 Efficiency: The simulator must be fast and easy in operation.

 Service Extension: The simulator must be helpful to extend the functionalities of the real world ATM machines. It means that the banks can experiment the enhancements on the simulator before they implement it in the real world system.

**CHAPTER-3**

**PRODUCT SPECIFICATION**

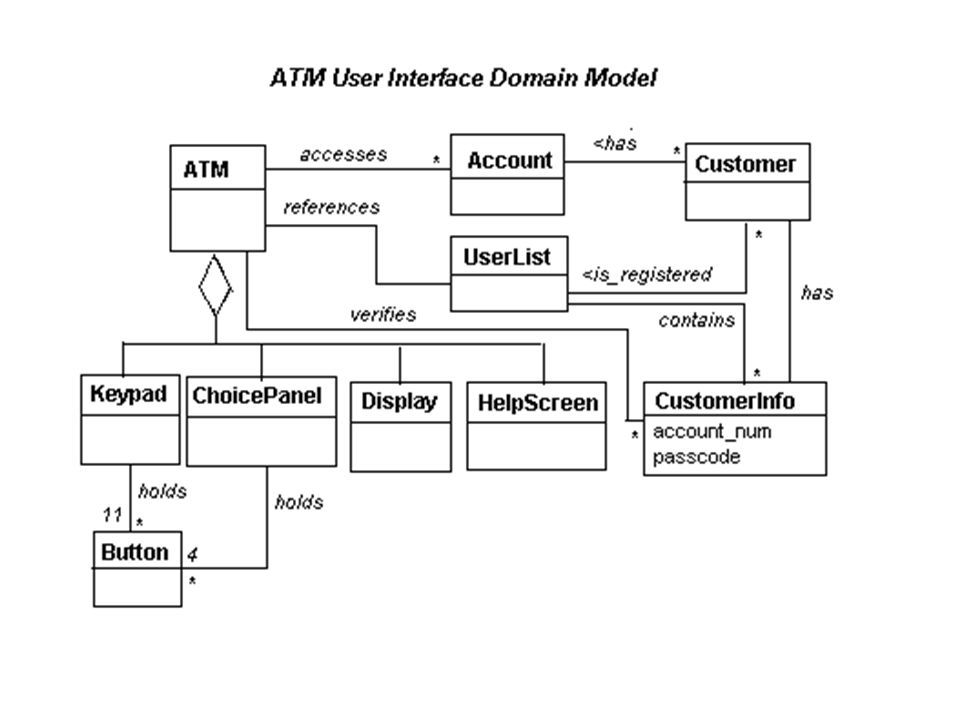
**PRODUCT PERSPECTIVE:**

|  |
| --- |
| The real world ATM is continuously developing day by day. More and more new and exciting features are incorporated in the machine as per customer‟s increasing needs. In the earlier times ATM was only recognized for cash withdrawal or balance enquiry. But now a day various new features like cash deposits, various Bill payments, Mobile recharge etc are added which are attracting more and more customers. In spite of these advancements in the ATM world, the ATM is not successful in rural areas. Most of the people are illiterate and even can‟t perform a simple transaction like cash withdrawal due to language bondation. The same problem exists with the blind people. Providing more and more language options is useless because they may not be able to read a text written on the screen. Thus the existing system although growing day by day and fulfilling more and more needs of customers fails in the case when simplicity and user friendliness is measured. |

|  |
| --- |
| **PRODUCT SCOPE:**  The scope of the project are as follows:  As the project is a simulator so the new experiments can be performed on it before we implement it on the real world machine. So here we are also performing an experiment of speech synthesis and speech recognition. The performance of the system can be studied on the simulator and then it can be implemented on real world machines. Similarly, we can also perform other advancements on the simulator before deploying it on the real system. |

|  |
| --- |
| **PRODUCT FUNCTIONS:**    The software to be designed will control a simulated Automated Teller Machine (ATM) having a capablility to read a card, a customer console (keyboard and display) for interaction with the customer and a printer for printing customer receipts. The simulator must work in a networked environment (i.e. LAN in our case).It must also communicate with the requisite servers available in LAN. The simulator software includes also the basic features of an ATM viz Withdrawal, deposit, transaction details, money transfer and balance queries. |

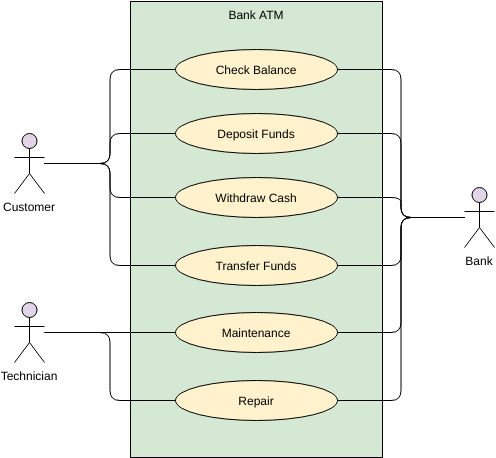
|  |  |
| --- | --- |
| **OPERATING ENVIRONMENT:**  As we know that the real world ATM machines work in a WAN (Wireless Area network). So it becomes necessary for our project to work at least in a networked environment. So for this we have implemented our project in a distributed environment which consists of a server end and multiple client ends. The software runs on the client end and the request made by the user at the client end goes for validation at the server end and if it has been validated then the requested transaction is performed at the server and the appropriate message is displayed at the server.  **COST & BENEFIT ANALYSIS:**  The cost & benefit analysis is a part that was done both by us and the organization which is  assumed to be bank here that wants us to develop his application.  In this part, we analyzed whether the costs spent on this application did actually benefit the bank in terms of benefit they are receiving after using it. The answer was undoubtedly “YES”.  Initially the bank was handling all these processes manually which resulted in serving a rush of customers with same kind of requests i.e. Account enquiry, deposit & withdrawal. This resulted in bank employees devoting a chunk of their time & resources for those customers who had common requests.  Soon the bank authorities realized that they are loosing on a lot of funds because the response times of bank has gone down due to which they are not able to serve all of their customers. Even those with these kinds of requests were left unattended sometimes thereby creating unrest among the customers and directly affecting the bank’s goodwill.  Moreover, the bank found itself unable to handle new big potential customers like NRIs & people who had to open a bank account. Soon the bank decided to hire additional people & spend on their training. There were also considerations about opening a new branch altogether to share the workload.  **CHAPTER -4**  **DESIGN**   |  | | --- | | **DESIGN**  **Architectural Design**  Under architectural design, after defining the whole system into a set of objectives & further subdividing them into functions, we defined the basic dependency & communication between them.  This means that all the prime functions, their required inputs, expected output/behavior & interdependency between other functions were clearly defined. The corresponding interfaces for the user for each function were designed to ensure user-friendliness.  We actually addressed the system-level problems here and made a conscious effort to build a robust design which can result in an effective communication within itself and with the system in terms of raw data or processed information.  All the primary database design for data storage was also done in this phase.    **Detailed Design**  In this phase, we further subdivided every function into a set of modules & defined required inputs & expected behavior for each of them. All the minute correlations, interdependencies, communication between the modules were clearly defined. The source, usage & processing of data for every module was carefully done. The database design was also normalized at this stage to ensure that the data is efficiently stored & retrieved.  Detailed design helped us to exactly concretize every problem into inputs & outputs and visualize them in terms of their communication with each other. We focused on interdependency & interoperability between the broken modules here.  It was this design phase where the factors like user-friendliness ease of use, scalability and self-explanation of interfaces & outputs were actually realized. For all the modules, the placement of controls, passing of information, communication of different interfaces, user messages, data transfer to databases was defined.  **USER INTERFACES:**  The user interfaces can be well defined by the following diagrams: | |



**FIG.1.USER INTERFACE DIAGRAM OF ATM MACHINE**

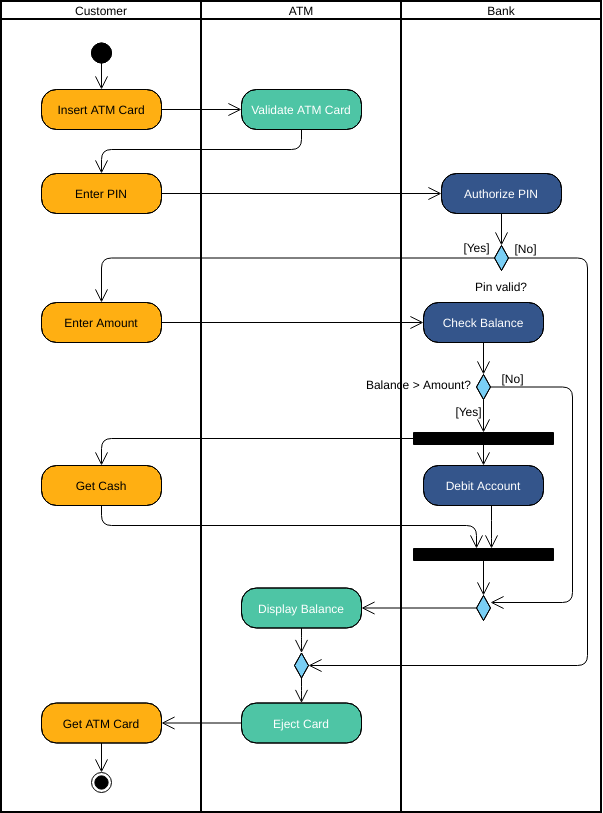
**USE CASE DIAGRAM FOR ATM:**

The use case diagram for atm is given below:

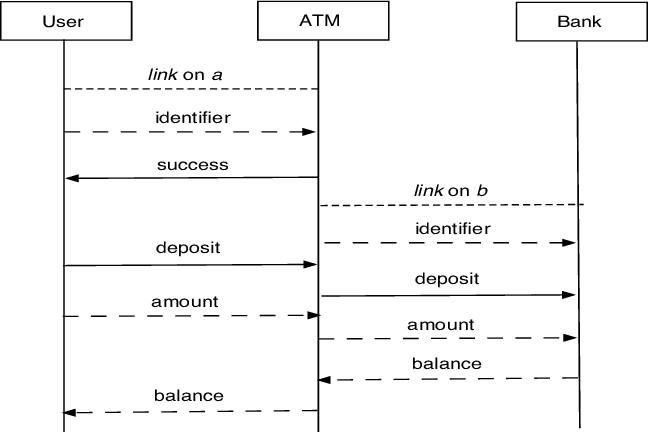


**FIG.2.USE CASE DIAGRAM OF ATM MACHINE**

|  |
| --- |
| **ACTIVITY DIAGRAMS:**  An activity diagram comes under “Scenario based modeling” of analysis modeling. The UML activity diagram supplements the use-case by providing a graphical representation of the flow of interaction within a specific scenario. |

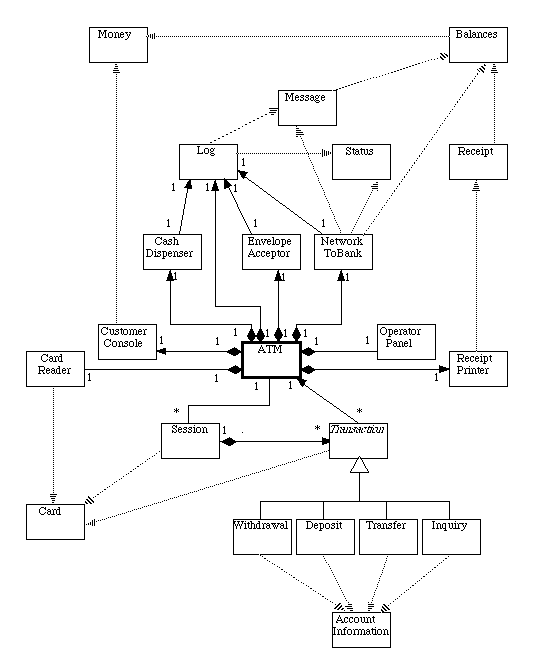


**SEQUENCE DIAGRAM:**



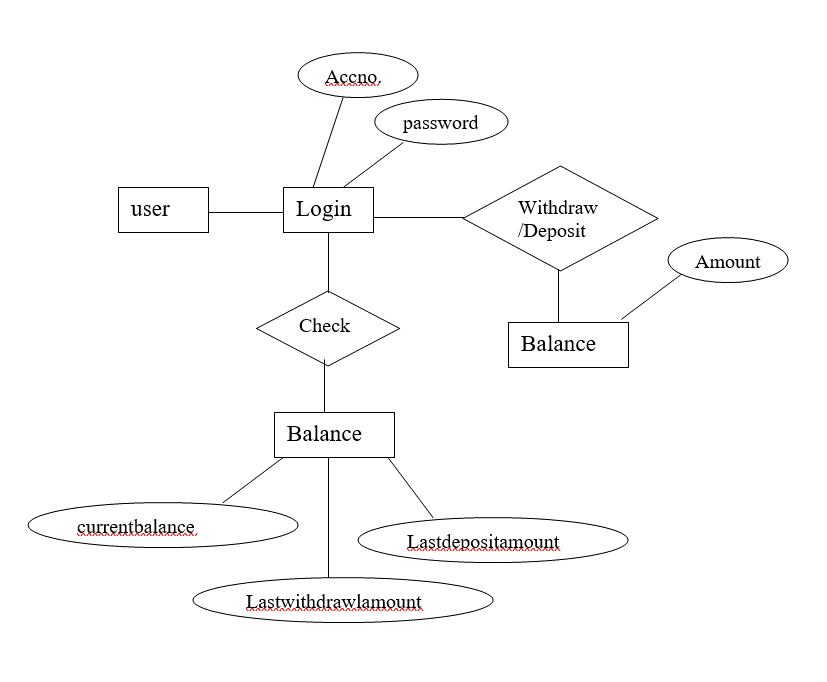
**FIG.4. SEQUENCE DIAGRAM FOR ATM AND BANK INTERACTION**

**CLASS DIAGRAM FOR ATM :**



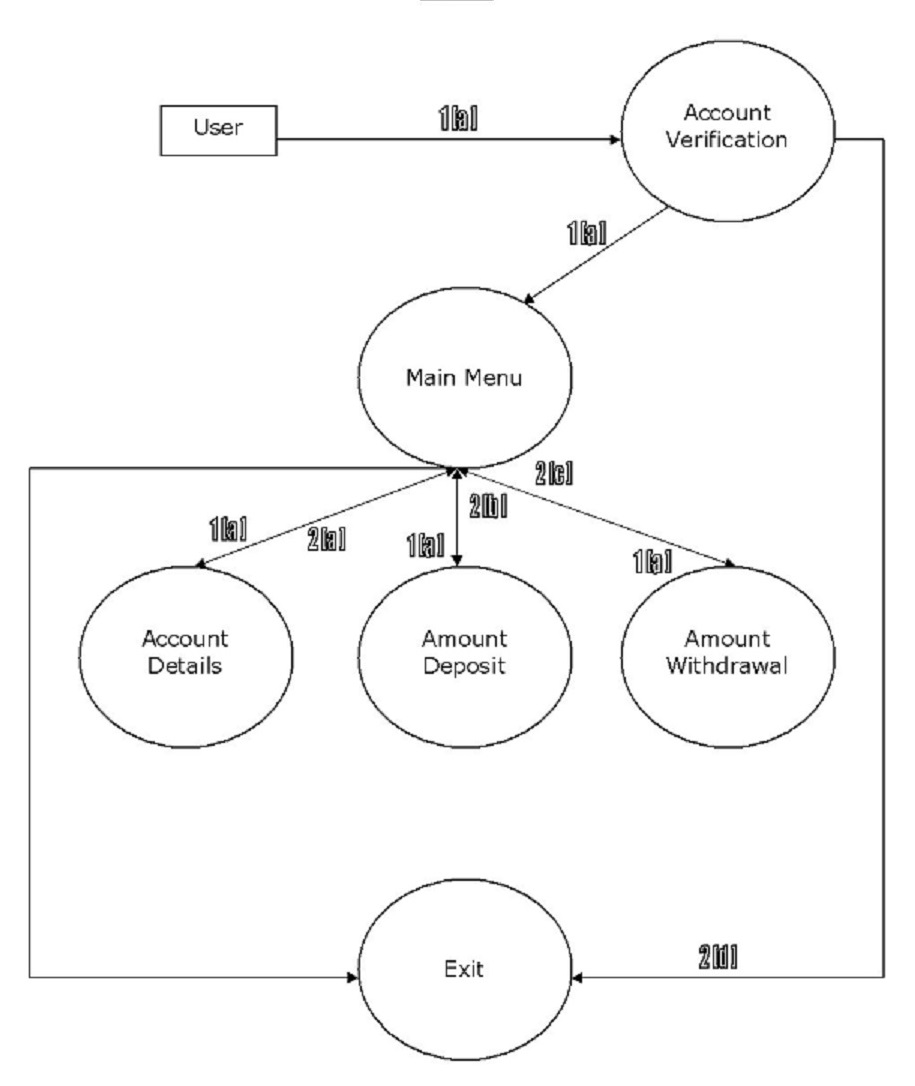
**FIG.5.CLASS DIAGRAM FOR ATM**

**ER DIAGRAM:**



**FIG.6.ER DIAGRAM FOR ATM SIMULATOTR**

**DATA FLOW DIAGRAM:**

****

**FIG.7. DATA FLOW DIAGRAM FOR ATM SIMULATER**

**Description of the data objects:**

**1[a]:** Account & Pin number entered by the user

**2[d]:** If the Account number does not match with Pin number or vice versa, the application cleanly exits

to the system with a proper user message like “Invalid Account / Pin no. entered”.

If the inputted details are correct, they are passed to the “Main Menu” process and further to different

modules which serve the customer’s requests individually using those details.

**2[a]:** Account information & details like Present Balance, Last Deposit & Last Withdrawal.

**2[b]:** Deposit details & account balance after deposit.

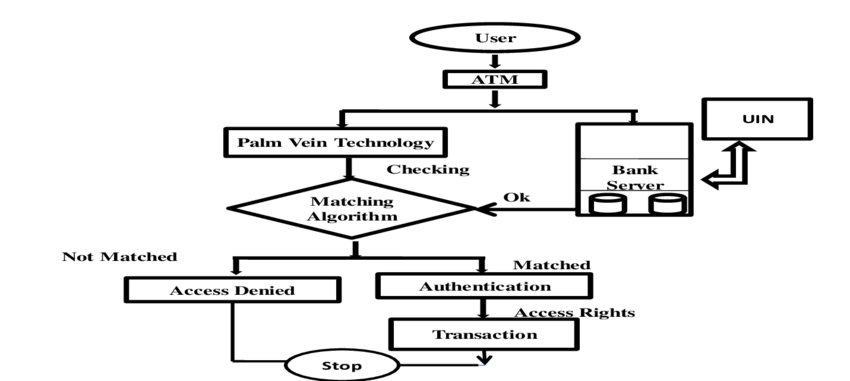
**2[c]:** This is withdrawal details & account balance after the withdrawal in case the withdrawal amount

entered is correct.

If the withdrawal amount entered is incorrect or invalid like greater than the balance, a message is

displayed to the user like “Invalid amount entered”.

**FLOW CHART:**



**FIG.8.FLOWCHART OF ATM MACHINE**

**CHAPTER -5**

**TECHNICAL DETAILS**

**HARDWARE SPECIFICATION:**

The minimum recommended hardware for installing and running Geomatica effectively is as follows:

**Processor :**

-Intel i5 or better.

- Multiple core processors strongly recommended.

**Memory**:

-8 GB or higher recommended1.

**Graphics Cards**:

- 24-bit Windows graphics card or accelerator running at 1280x1024.

- 256 MB RAM or higher recommended.

**Disk space:**

- 2.5 GB free hard disk space for software installation.

- 1 GB free hard disk space for installing demonstration datasets.

- Customer data storage requirements vary by project.

**OPERATING SYSTEM:**

The following operating systems are supported:

**Windows:**

-Version = Windows 10 Server 2016 / 2019

-Note = 64-bit only

**SOFTAWARE SPECIFICATION:**

**-**During installation, you have will be given the option of selecting Python 2.7 if you already have this version installed on your machine. Otherwise Python 3.5 will be installed.

-Many of the existing Python 2.7 scripts can be easily transformed to work within the new Python 3.5 environment, and PCI’s Support team can assist you.

**CHAPTER -6**

**CODING**

|  |
| --- |
|  |
|  | import getpass |
|  | import string |
|  | import os |
|  |  |
|  | # creatinga lists of users, their PINs and bank statements |
|  | users = ['user', 'user2', 'user3'] |
|  | pins = ['1234', '2222', '3333'] |
|  | amounts = [1000, 2000, 3000] |
|  | count = 0 |
|  | # while loop checks existance of the enterd username |
|  | while True: |
|  | user = input('\nENTER USER NAME: ') |
|  | user = user.lower() |
|  | if user in users: |
|  | if user == users[0]: |
|  | n = 0 |
|  | elif user == users[1]: |
|  | n = 1 |
|  | else: |
|  | n = 2 |
|  | Break |
|  | else: |
|  | print('----------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('INVALID USERNAME') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('----------------') |
|  |  |
|  | # comparing pin |
|  | while count < 3: |
|  | print('------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | pin = str(getpass.getpass('PLEASE ENTER PIN: ')) |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('------------------') |
|  | if pin.isdigit(): |
|  | if user == 'user1': |
|  | if pin == pins[0]: |
|  | Break |
|  | else: |
|  | count += 1 |
|  | print('-----------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('INVALID PIN') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('-----------') |
|  | print() |
|  |  |
|  | if user == 'user2': |
|  | if pin == pins[1]: |
|  | Break |
|  | else: |
|  | count += 1 |
|  | print('-----------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('INVALID PIN') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('-----------') |
|  | print() |
|  |  |
|  | if user == 'user3': |
|  | if pin == pins[2]: |
|  | Break |
|  | else: |
|  | count += 1 |
|  | print('-----------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('INVALID PIN') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('-----------') |
|  | print() |
|  | else: |
|  | print('------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('PIN CONSISTS OF 4 DIGITS') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('------------------------') |
|  | count += 1 |
|  |  |
|  | # in case of a valid pin- continuing, or exiting |
|  | if count == 3: |
|  | print('-----------------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('3 UNSUCCESFUL PIN ATTEMPTS, EXITING') |
|  | print('!!!!!YOUR CARD HAS BEEN LOCKED!!!!!') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('-----------------------------------') |
|  | exit() |
|  |  |
|  | print('-------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('LOGIN SUCCESFUL, CONTINUE') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('-------------------------') |
|  | print() |
|  | print('--------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print(str.capitalize(users[n]), 'welcome to ATM') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('----------ATM SYSTEM-----------') |
|  | # Main menu |
|  | while True: |
|  | #os.system('clear') |
|  | print('-------------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | response = input('SELECT FROM FOLLOWING OPTIONS: \nStatement\_\_(S) \nWithdraw\_\_\_(W) \nLodgement\_\_(L) \nChange PIN\_(P) \nQuit\_\_\_\_\_\_\_(Q) \n: ').lower() |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('-------------------------------') |
|  | valid\_responses = ['s', 'w', 'l', 'p', 'q'] |
|  | response = response.lower() |
|  | if response == 's': |
|  | print('---------------------------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print(str.capitalize(users[n]), 'YOU HAVE ', amounts[n],'EURO ON YOUR ACCOUNT.') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('---------------------------------------------') |
|  |  |
|  | elif response == 'w': |
|  | print('---------------------------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | cash\_out = int(input('ENTER AMOUNT YOU WOULD LIKE TO WITHDRAW: ')) |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('---------------------------------------------') |
|  | if cash\_out%10 != 0: |
|  | print('------------------------------------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('AMOUNT YOU WANT TO WITHDRAW MUST TO MATCH 10 EURO NOTES') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('------------------------------------------------------') |
|  | elif cash\_out > amounts[n]: |
|  | print('-----------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('YOU HAVE INSUFFICIENT BALANCE') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('-----------------------------') |
|  | else: |
|  | amounts[n] = amounts[n] - cash\_out |
|  | print('-----------------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('YOUR NEW BALANCE IS: ', amounts[n], 'EURO') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('-----------------------------------') |
|  |  |
|  | elif response == 'l': |
|  | print() |
|  | print('---------------------------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | cash\_in = int(input('ENTER AMOUNT YOU WANT TO LODGE: ')) |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('---------------------------------------------') |
|  | print() |
|  | if cash\_in%10 != 0: |
|  | print('----------------------------------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('AMOUNT YOU WANT TO LODGE MUST TO MATCH 10 EURO NOTES') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('----------------------------------------------------') |
|  | else: |
|  | amounts[n] = amounts[n] + cash\_in |
|  | print('----------------------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('YOUR NEW BALANCE IS: ', amounts[n], 'EURO') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('----------------------------------------') |
|  | elif response == 'p': |
|  | print('-----------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | new\_pin = str(getpass.getpass('ENTER A NEW PIN: ')) |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('-----------------------------') |
|  | if new\_pin.isdigit() and new\_pin != pins[n] and len(new\_pin) == 4: |
|  | print('------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | new\_ppin = str(getpass.getpass('CONFIRM NEW PIN: ')) |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('-------------------') |
|  | if new\_ppin != new\_pin: |
|  | print('------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('PIN MISMATCH') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('------------') |
|  | else: |
|  | pins[n] = new\_pin |
|  | print('NEW PIN SAVED') |
|  | else: |
|  | print('-------------------------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print(' NEW PIN MUST CONSIST OF 4 DIGITS \nAND MUST BE DIFFERENT TO PREVIOUS PIN') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('-------------------------------------') |
|  | elif response == 'q': |
|  | exit() |
|  | else: |
|  | print('------------------') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('RESPONSE NOT VALID') |
|  | print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |
|  | print('------------------') |

**CHAPTER -7**

**TESTING**

**Testing Methodology**

Software testing is the process of executing a program with the intention of finding errors in the code.

It is the process of exercising or evaluating a system or system component by manual or by automatic

means to verify that it satisfies specified requirements or to identify differences between expected

and actual results.

The objective of testing is to show incorrectness and testing is considered to succeed when an error is

detected. An error is a conceptual mistake made by either the programmer or the designer or a

discrepancy between a computed value and a theoretically correct value. A fault is a specific

manifestation of an error. An error may be cause of several faults. A failure is the inability of a

system or component to perform its required function within the specified limits. A failure may be

produced when a fault is executed or exercised.

Other activities that are often associated with software are static analysis and dynamic analysis. Static

analysis investigates the source code of software, looking for problems and gathering metrics without

actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to

provide information such as execution traces, timing profiles and test coverage information.

**Unit Testing or Module Testing**

The starting point of testing is Unit testing. In this, a module is tested separately at each step. This

helps to detect syntax and logical errors in the program and is performed by the coder himself /herself

during coding.

**Integration Testing**

The modules, which are tested in the Unit Testing, are integrated to build the overall system. It is

observed that many errors crop up when the modules are joined together. Integration testing uncovers these errors while integrating the modules. It helps in establishing confidence (correctness) in the complete, assembled system. It tests the System Design. It focus on control, communication, interfaces, performance (other system qualities). It make use of stubs, test-beds, data generators. It is the phase of software testing in which individual software modules are combined and tested as a group

It follows unit testing and precedes system testing.

Integration testing takes as its input [modules](http://en.wikipedia.org/wiki/Module_%28programming%29) that have been [unit tested](http://en.wikipedia.org/wiki/Unit_testing), groups them in larger aggregates, applies tests defined in an integration [test plan](http://en.wikipedia.org/wiki/Test_plan) to those aggregates, and delivers as its output the integrated system ready for [system testing](http://en.wikipedia.org/wiki/System_testing).

Integration testing concentrates entirely on module interactions, assuming that the details within each module are accurate. Module and Integration testing can be combined, verifying the details of each module's implementation in an integration context. Many projects compromise, combining module testing with the lowest level of subsystem integration testing, and then performing pure integration testing at higher levels. Each of these views of integration testing may be appropriate for any given project, so an integration testing method should be flexible enough to accommodate them all.

**System Testing**

The System testing is bringing together of all programs that a system comprises for testing purposes. System testing is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. Programs are typically integrated in a top-down, incremental fashion. It is a series of different tests whose primary purpose is to fully exercise the computer-based system. It includes the following tests: -

* **Recovery Testing: -** It is a system test that forces the software to fail in a variety of ways and verifies that recovery is properly performed.
* **Stress Testing:-** These are designed to confront program functions with abnormal situations. It executes a system in a manner that demands resources in abnormal quantity, frequency or volume.
* **Security Testing:-** This testing attempts to verify that protection mechanism built into a system will protect it from unauthorized penetration.

The system testing is an investigatory testing phase, where the focus is to have almost a destructive attitude and test not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

**Black Box Testing**

It is also known as Functional Testing. It tests the overall functional requirements of product. Inputs are supplied to product and outputs are verified. If the outputs obtained are the same as the expected ones then the product meets the functional requirements. In this, the internal procedures are not considered. In this the tester would only know the "legal" inputs and what the expected outputs should be, but not how the program actually arrives at those outputs. This Testing is more effective on larger units of code. In this test’s are done from user point of view.

**White Box Testing**

It is also known as Structure Testing. It focuses on the internal functioning of the product. It tests the loops of the Procedure, Decision points, Execution paths etc.

White box testing uses specific knowledge of programming [code](http://www.webopedia.com/TERM/W/code.html) to examine outputs. The test is accurate only if the tester knows what the program is supposed to do. He or she can then see if the program diverges from its intended goal. White box testing does not account for errors caused by omission, and all visible code must also be readable. As the knowledge of internal coding structure is prerequisite, it becomes very easy to find out which type of input/data can help in testing the application effectively. The other advantage of white box testing is that it helps in optimizing the code. It helps in removing the extra lines of code, which can bring in hidden defects.

**Acceptance Testing**

This Testing is done when the software is developed for the specific customer. A series of tests are conducted to enable the customer to validate all requirements. The end user/ customer conducts these tests and may range from adhoc test to well-planned systematic series of tests. Acceptance testing may be conducted for few weeks or months. The discovered errors will be fixed and better quality software will be delivered to the customer.

Acceptance testing is performed by the [customer](http://en.wikipedia.org/wiki/Customer) on a [system](http://en.wikipedia.org/wiki/System) prior to the customer accepting delivery or accepting transfer of ownership of that system.

The customer specifies scenarios to test when a user story has been correctly implemented. A story can have one or many acceptance tests, what ever it takes to ensure the functionality works. Acceptance tests are black box system tests. Each acceptance test represents some expected result from the system. Customers are responsible for verifying the correctness of the acceptance tests and reviewing test scores to decide which failed tests are of highest priority. Acceptance tests are also used as regression tests prior to a production release. A user story is not considered complete until it has passed its acceptance tests. This means that new acceptance tests must be created each iteration or the development team will report zero progress.

While testing the ATM machine, all levels of testing must be performed on it at the required

stages.

Given below are the various test cases for ATM:

1.Verify if the card reader is working correctly. A screen should ask you to insert the pin after inserting

the valid card.

2. Verify if the cash dispenser is working as expected.

3.Verify if the receipt printer is working correctly. Which means it can print the data on the paper and

the paper comes out properly.

4.Verify if the Screen buttons are working correctly. For touch screen: Verify if it is operational and working as per the expectations.

5.Verify if the text on the screen button is visible clearly.

6.Verify the font of the text on the screen buttons.

7. Verify each number button on the Keypad.

8.Verify the functionality of the Cancel button on the Keypad.

9.Verify the text color of the keypad buttons. The numbers should be visible clearly.

10.Verify the text color and font of the data on the screen. The user should be able to read it clearly.

**Verifying the Message:**

11**.** Insert the card and an incorrect PIN to verify the message.

12. Verify the message when there is no cash in the ATM.

13. Verify the messages after a transaction.

14.Verify if a user will get a correct message if a card is inserted incorrectly.

**Cash Withdrawal:**

15. Verify the cash withdrawal functionality by inserting some valid amount.

16.Verify if a user can perform only one cash withdrawal transaction per PIN insert.

17.Verify the different combinations of operation and check if there will be a power loss in the middle

of the operation.

**Negative Test cases:**

18.Verify the functionality by entering a wrong pin number for 3 or more times.

19.Verify the card reader functionality by inserting an expired card.

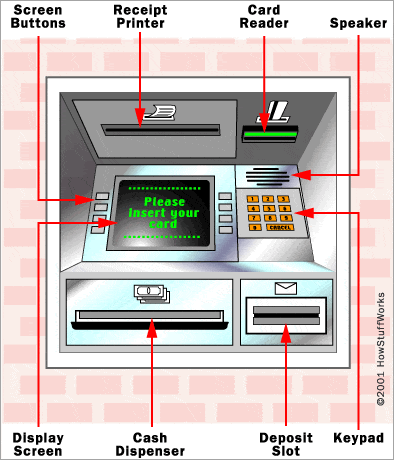
20.Verify the deposit slot functionality by inserting an invalid cheque.

21.Verify the cash withdrawal functionality by inserting invalid numbers like 10, 20, 50 etc.

22. Verify the cash withdrawal functionality by entering an amount greater than the per day limit,

23. Verify the cash withdrawal functionality by entering an amount greater than per transaction limit.

24. Verify the cash withdrawal functionality by entering an amount greater than the available balance in the account.



**FIG.9.TEST CASE DIAGRAM FOR ATM SIMULATOR**

**Test Case**

It describes an input description and an expected output description. These are derived during all phases of the development cycle. It helps in determining expected results before

running a test case.

Some of the test cases for this project are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Action** | **Input** | **Excepted Output** | **Pass (Y/N)** |
| 1. | To check account no. and passowrd | Valid Record | procedded | Y |
| 2. | To check account no and password | Invalid Record | Error message. | Y |
| 3. | To deposit amount | Invalid Record | No error  Message | N |
| 4. | To check deposited amount | Valid record | procedded | Y |
| 5. | To check functionality of interface button | Invalid Record | Error message | Y |
| 6. | To check functionality of interface button | Invalid Record | Blank interface | N |

**Test Reports and Debugging**

**Test Reports**

**AccountVerification**

In Account Verification form, First user enter own Account number and password. If your entered

Account number and password matched from backened Then project will process next step. If user put

wrong account number then project will be show error message. On Go, validation for each text box is

carried out and attention is focused on to respective text box that does not meet validation requirement

, thus chances of error are eliminated.

**Account Detail**

In this, module display the account detail with user’s account number.

Account detail module display the Last Deposit , Last withdrawl, Present Balance etc. regarding

account number and user name.

**Amount Deposit**

In Amount Deposit module, User can deposit the amount but user can deposit maximum three

notes. When user choice the Amount Deposit Then this module will be show three statement:

user present balance, Enter the amount to be deposited and Enter new balance after deposit.

If user click add without deposit amount then this project will display the error message..

**Amount Withdrawl**

By use of module, User can withdraw amount but if entered amount is available in bank.

If entered amount is not available then this project will be display error message. So user can

withdraw minimum amount than available in bank.

**EXIT**

This module will be work of EXIT. If user’s processing work have finish . Then user will be exit from

own Account .

**Debugging**

It is the activity of locating and correcting errors. It starts once a failure has been detected.

In this the following debugging induction approach is used:-

1. **Locate the pertinent data**

In this step, we take into account what the program did correctly and what it did incorrectly.

1. **Organize the data**

The next step is the structuring of data to allow one to observe patterns. For e.g. in our program the error can occur if we are assigning more than two project to the particular person.

1. **Devise a hypothesis**

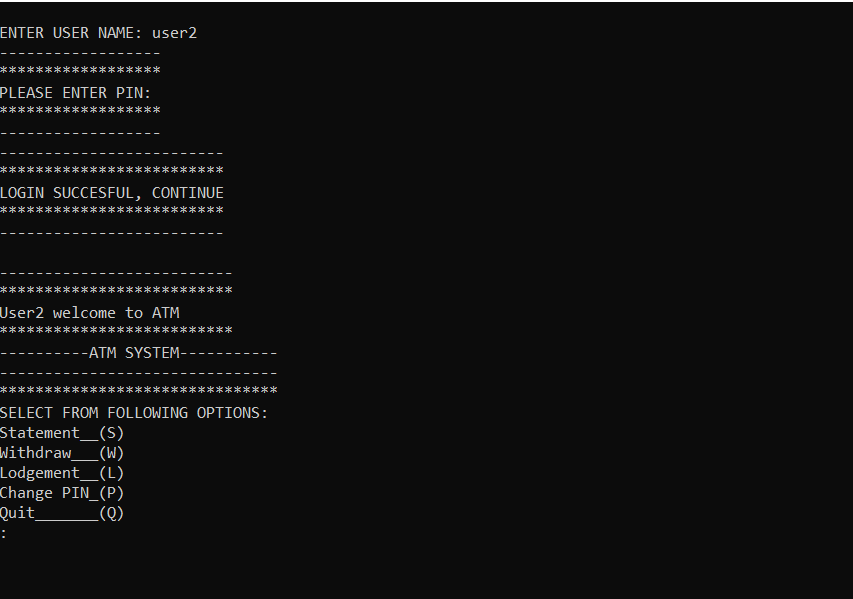
A hypothesis is devised so that according to it we can find out the most probable one, which is causing the error more frequently.

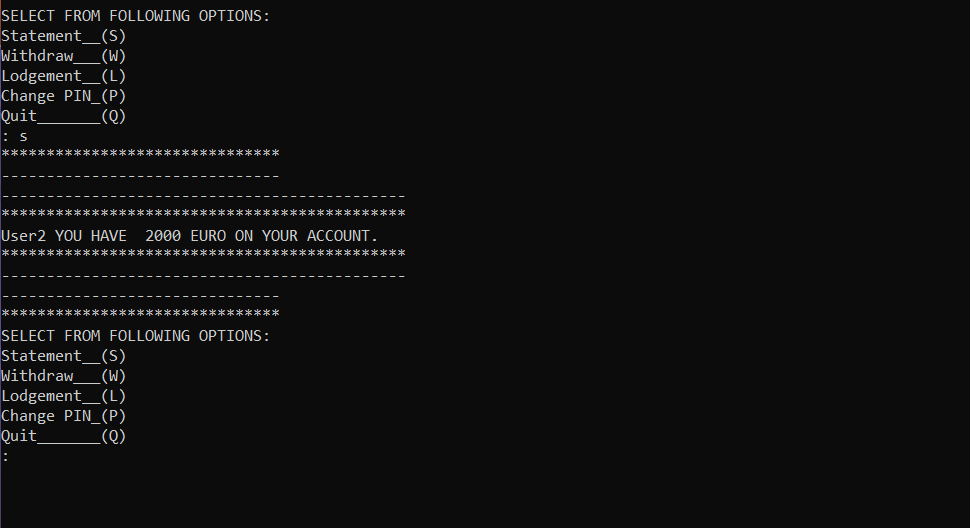
1. **Prove the hypothesis**

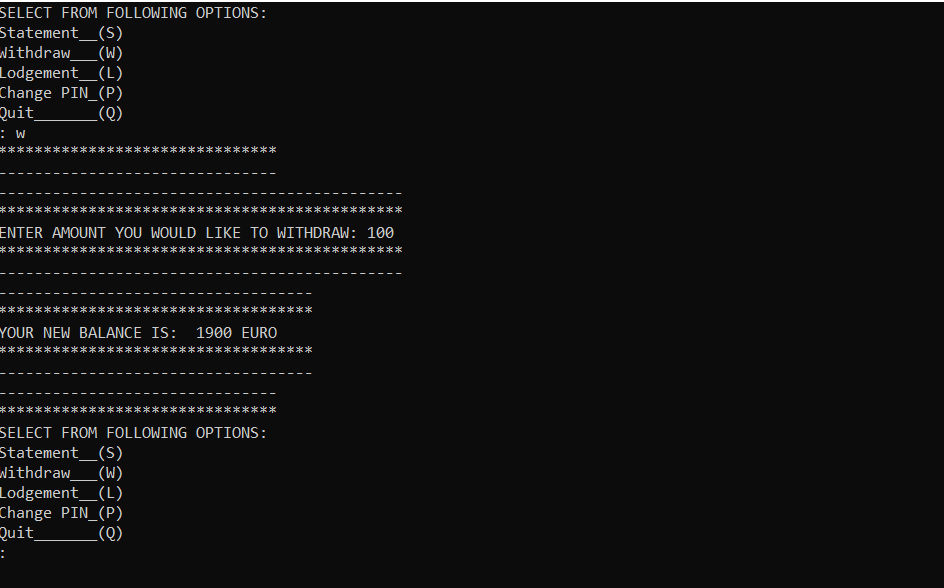
Comparing it to the original clues proves the hypothesis or data, making sure that this hypothesis completely explains the existence of the clues. If it does not either the hypothesis is invalid, or the hypothesis is incomplete, or the multiple errors are present.

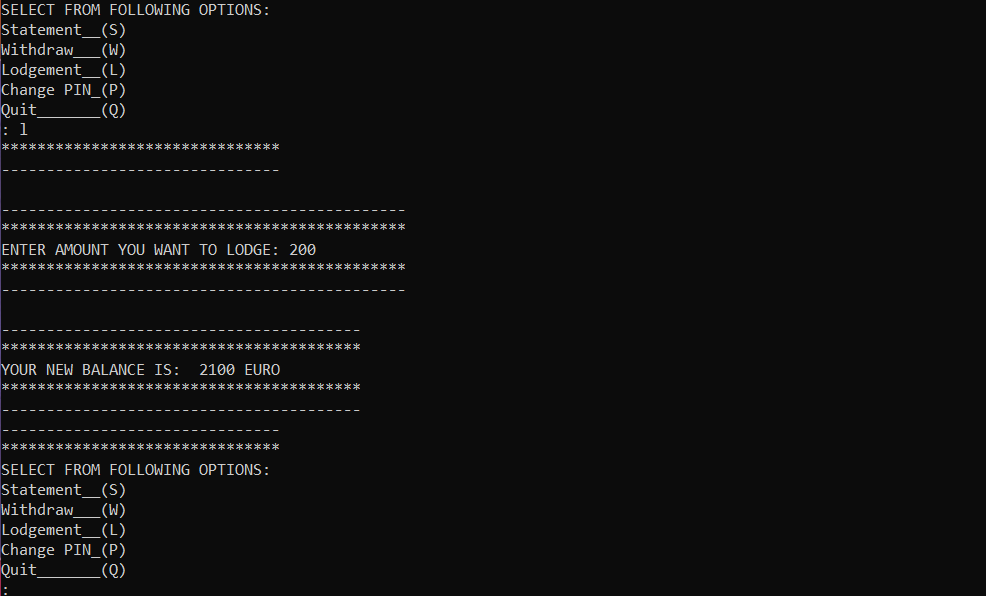
**CHAPTER -7**

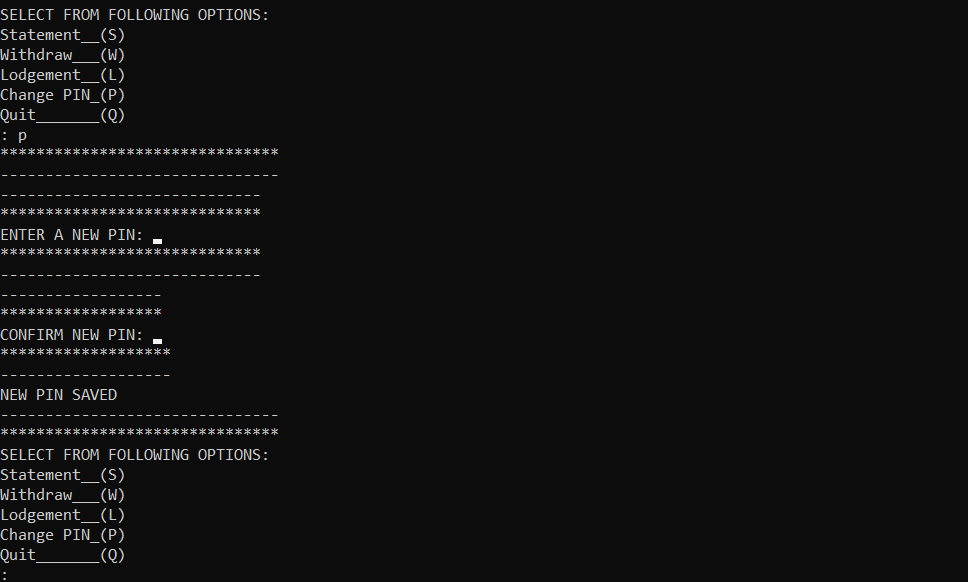
**SCREEN LAYOUTS**













**CHAPTER-9**

**FUTURE ENHANCEMENTS**

-More than just a cash machine***-***ATMs have changed our relationship with money, machines and bank branches. We look at how this will continue in the future.

-“The ATM has been the only useful innovation in banking for the past 20 years,” said the former chairman of the US Federal Reserve, Paul Volcker in 2009.

-The seemingly simple idea of allowing customers 24-hour, self-service access to cash has been a global success story. There are now around three million ATMs worldwide, and a further one

million are expected by 2020.

-ATMs have always been at the forefront of innovation, ever since the first one was installed in a Barclays Bank branch in Enfield, north London in 1967. Here are several ways in which innovation is keeping ATMs relevant.

**New technology:**

-Technology helps drive increased ATM functionality, ease of use and security for customers and

ATM deployers alike.

**For example:**

**-**Card-less atms allow pre-staged withdrawals via contactless, mobile, wearables, or ‘cash by code’ as described below. With no card entry slot, the machines look different, but are potentially more secure as skimming is impossible.

-Drive-up ATMs mean customers can withdraw or deposit cash without getting out of their car.

This improves convenience and security, especially for businesses wanting to bank takings at night.

-Likewise, biometric authentication, such as palm and finger vein scanning, enhances both security and ease of use.

**New services:**

-Smart ATMs are more than just a cash machine. They provide multiple services, such as cash and cheque deposits, multi-currency and Bitcoin disbursements, bill payment and the purchase of everything from mobile airtime to cinema tickets.

-Pre-staged withdrawals or money transfers at ATMs are now also possible. Users simply log on to their mobile banking app and select the amount they wish to withdraw. They receive a code, which they enter at participating ATMs, or can pass to a friend or family member.

**New ways to engage customers**

-ATMs are the ultimate self-service channel, allowing financial institutions to make direct contact

with customers as they are engaged with their finances. Screens are improving all the time with

colour and interactive pinch, swipe and zoom functionality to promote engagement.

-Video banking and interactive teller ATMs in bank lobbies extend service and opening hours, boost customer satisfaction and sales conversion at lower cost.

**New business models:**

-Cash recyclers present a new business model, especially for the convenience and supermarket

sectors. Stores can top up their machines with takings from the till, meaning they save on trips to

the bank, plus the costs of transporting, insuring and banking cash.

-Mobile ATMs and bank-in-a-box concepts could transform the role and service model of bank branches. They enable banks to extend or maintain a presence in unstaffed or remote areas at a

viable cost.

**How Cashflows can help:**

Cashflows is a member of all the major international card schemes and offers BIN Sponsorship to independent ATM deployers (IADs) and others wanting to access ATM acquiring opportunities.

We’ve been an active ATM BIN Sponsor since 2010, so are an experienced and reliable partner.

We currently work with all the largest IADs across Europe and are the principal ATM sponsor for

the UK’s leading transaction processor. We’re proud to have settled £2.1 billion from 38,500 ATMs in 2017.

**LIMITATIONS**

We could not think of many limitations of the system but as we know,

nothing is perfect; our system also comes with few limitations.

1. The system would work more realistically if the equipment that reads embossed characters from a credit card works in tandem with it.
2. If a database of a large number of customers could be accessed from a server
3. in a real time mode, the efficiency of the system could have been further explored.
4. The system does not give us an exhaustive balance with all the calculations like tax but this functionality can be introduced in same.

**CONCLUSION**

-From this presentation, one can observe that an ATM system is associated with the bank transactions of the consumers.

-Majorly, the ATM system is utilized for the money associated transactions from the consumers.

Consumers make major use of ATM to withdraw money from their bank account.

-It is a fast way to get money out of your account, especially when on the go or during a trip.

-No person can dispute the massive amount of money that this spectacular machine makes. From our

research we have discovered that the people of America use this machine more than any other universal machine for convenient banking.  it is very clear that this machine in particular helped our economy greatly when it was made and even now, even to this day is is just as popular as it ever has been.

**BIBLIOGRAPHY**

This is a small selected annotated bibliography of programming-related books. Most of the books listed are not Python-specific, but all of them are interesting, useful—and accessible.

Clean Code

Robert C. Martin (Prentice Hall, 2009, ISBN 0132350882). This book addresses many “tactical” issues in programming: good naming, function design, refactoring, and similar. The book has many interesting and useful ideas that should help any programmer improve their coding style and make their programs more maintainable.