



**ATM
system**

Acknowledgement

It is a great pleasure for us to acknowledge all those who have contributed towards the conception, origin and nurturing of this project that is on System analysis the ***“ATM System”***.

The way can't walk itself. We have to walk on it. For that we must have a guide. Many guides have contributed to the successful completion of the project. We would like to place on record my grateful thanks to each one of them who help us in this project.

Before we get into thick of the thing, we would like to add a few heartfelt words for the people who gave us unending time support whichever and whenever necessary.

Our grateful thanks go to our Dept., which provides us an opportunity as a project subject in **6th Semester** to develop a report work skill in this System analyzing.

We would like to thank our parents & friends for giving us full feedback when we are in trouble. Our special thanks go to **Ms. Kinjal Jadav** to give their expert guidance to us whenever necessary.

Last but not the least; I heartily thank our H.O.D. **Mr. Mahesh Panchal**.

PREFACE

Case study (Software Engineering) is a long establishment method of organizing, learning, so as to encourage coordination of Subject areas, it aims at closer integration of theory and practical give fundamental basic for student learning and produce, more practical techniques. Project method in common with other students centered technique help to develop market survey, planning decision making on for example and identifying the critical integrated industrial process for producing equipment, a component or a model of process.

ABSTRACT

This report attempts to understand the design of an Automated Teller Machine (ATM) system, a 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. Some ATMs will also accept credit cards for cash advances. The first ATM was installed in 1969 by Chemical Bank at its branch in Rockville Centre, New York. A customer using a coded card was dispensed a package containing a set sum of money.

Project Profile

Project Title	: ATM Sytem
Front End Tool	: Microsoft Visual Studio 2008
Back End Tool	: Microsoft SQL Server 2005
Project Platform	: ASP.NET
Project Guide	:Ms. Kinjal K. Jadav
Developed By	:Mr. Kamlesh Korat(08-CE-13) Mr. Chirag Korat(08-CE-12)
Submitted To	: Department of Computer Engineering, Kalol Institute Of Technology

DETAIL OF CHAPTERS

1.0 Introduction

1.1 Project Summary

1.2 Purpose

1.3 Scope

2.0 Project Management

2.1 Project Planning and scheduling

2.1.1 Project Development Approach

2.1.2 Project Plan

2.1.3 Schedule Representation

2.2 Risk Management

2.2.1 Risk Identification

2.2.2 Risk Analysis

2.2.3 Risk Planning

2.3 Estimation

2.3.1 Effort Estimation

3.0 System Requirements Study

3.1 User Characteristics

3.2 Hardware and Software Requirements

3.3 Constraints

4.0 System Analysis

4.1 Study of Current System

4.2 Problem and Weaknesses of Current System

4.3 Requirements of New System

4.4 Feasibility Study

4.5 Requirements Validation

4.6 Functions Of System

4.6.1 Use Cases, event trace or scenario

4.7 Data Modeling

4.7.1 Flow chart or activity design

4.7.2 Class Diagram/ E-R diagrams

4.7.3 System Activity or Object interaction Diagram

4.7.4 Data Dictionary

4.8 Functional and Behavioral Modeling

4.8.1 Context Diagram

4.8.2 Data Flow Diagram (0 and 1 level)

4.8.3 Process Specification and Decision Table

4.8.4 Control flow diagram

5.0 Testing

5.1 Testing Plan

5.2 Testing Strategy

5.3 Testing Methods

5.4 Test Cases

6.0 Limitation and Future Enhancement

7.0 Conclusion and Discussion

1.0 Introduction

1.1 Project Summary

An automated teller machine (ATM) or automatic banking machine (ABM) is a computerised telecommunications device that provides the clients of a financial institution with access to financial transactions in a public space without the need for a cashier, 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 smart card with a chip, that contains a unique card number and some security information such as an expiration date or CVVC (CVV). Authentication is provided by the customer entering a personal identification number (PIN).

1.2 Purpose

Using an ATM, customers can access their bank accounts in order to make cash withdrawals (or credit card cash advances) and check their account balances as well as purchase cellphone prepaid credit. If the currency being withdrawn from the ATM is different from that which the bank account is denominated in (eg: Withdrawing Japanese Yen from a bank account containing US Dollars), the money will be converted at a wholesale exchange rate. Thus, ATMs often provide the best possible exchange rate for foreign travelers and are heavily used for this purpose as well.

ATMs are known by various other names including Automated Transaction Machine, automated banking machine, cashpoint (in Britain), money machine, bank machine, cash machine, hole-in-the-wall, Bancomat (in various countries in Europe and

Russia), Multibanco (after a registered trade mark, in Portugal), and Any Time Money (in India)

1.3 Scope

The main purpose of the ATM division and information service is to provide the customers financial flexibility, worldwide acceptance and round-the clock convenience. Bank issues only VISA Credit Cards, the renowned Credit Card brand. Cardholders can purchase goods/services up to the credit limit and can reuse the credit facility upon repayment. Credit Card is a safer substitute to cash and is the major mode of payment worldwide. Standard Chartered Bank is the first to introduce the TAKA CREDIT CARD. The card is issued basically to a person's name and the specific person can use the card in anywhere in Bangladesh. The business activity of Premier Bank Credit Card section is to keep the records of all sales and customers' requests, the information of cardholders and reports them to necessary documents.

2.0 Project management

In this chapter we will discuss about project planning and scheduling. Our goal is to establish a pragmatic strategy for controlling, tracking, and monitoring a complex technical project.

In project management following things must be done.

- Project Planning and Scheduling
- Risk Management
- Estimation

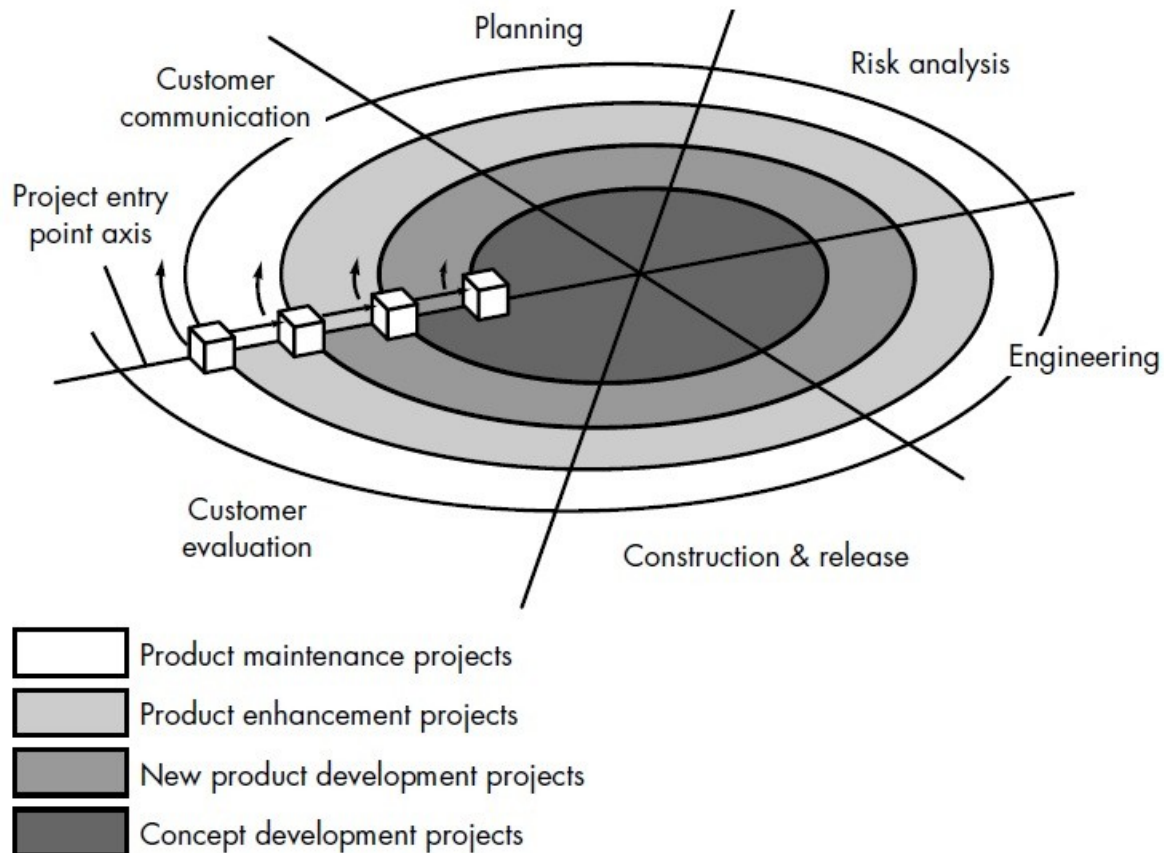
In Project planning and scheduling, Planning of the project is done. In scheduling different task are schedule according to the deadline of the project.

2.1Project Planning and scheduling

Project planning must deals with the following things.

- **Project Complexity:** - Project complexity has a strong effect but is heavily influenced by past practitioner experience.
- **Project Size:** - As size increases the interdependency of elements also grow. Watch out for scope creep.
- **The degree of structural uncertainty:** - the degree to which requirements are solidified and the ease of functional decomposition. The purpose of project planning is to ensure that the end result is completed on time, within budget, and exhibits quality!

2.1.1 Project development approach



The Spiral model is an evolutionary software process model that couples the iterative nature of prototyping with the controlled and systematic aspects of the linear sequential model. It provides the potential for rapid development of incremental versions of the software. Using the spiral model, software is developed in series of incremental release.

A spiral model is divided into a number of framework activities, also called *task regions*. There are between three and six task regions. Figure depicts a spiral model that contains six task regions:

- Customer communication – tasks required to establish effective communication between developer and customer.
- Planning – tasks required to define resources, timelines, and other project related information.
- Risk analysis – tasks required to assess both technical and management risks.
- Engineering – tasks required to build one or more representations of the application.
- Construction and release – tasks required to construct, test, install, and provide user support.
- Customer evolution – tasks required to obtain customer feedback based on evolution of the software representations created during the engineering stage and implemented during the installation stage.

Each of the regions is populated by a set of work tasks, called a task set, that are adapted to the characteristics of the project to be undertaken. For small projects, the number of work tasks and their formality is low. For larger, more critical projects, each task region contains more work tasks that are defined to achieve a higher level of formality.

In our case, we have to provide medium level of formality for making a good project report. We will take decision about cost, schedule and number of iterations required to complete the software.

2.1.2Project Plan

Stages of Software Lifecycle

- **Software Requirement Analysis**

This is the first stage of the project, which involves interaction with the customer to understand his/her needs, requirements, information, required

functions, performance and interfacing in MLM software. For this purpose requirement analyst will arrange a meeting for gathering information and additional details for software development. After completing requirement gathering tasks developer team will take a look for understand how requirements can be computerized. The requirement is documented in the form of a Software Requirement Specification (SRS) which is then presented to the customer for review.

- **Design**

Beginning once software requirements have been analyzed and specified, software design is the first of three technical activities – design, code generation, and test – that are required to build and verify the software.

Design is multi level process which defines following details:

- Data Design
- Architecture Design
- Interface Design
- Component level Design

- **Development**

The design must be translated into a machine-readable form. The coding step performs this task. In this stage, the developers will actually code the programs. The specifications arrived at the design stage for each and every function will be converted to code using tools that are finalized for the implementation of the Software. At this stage the testing methodology to be adopted will be finalized. For each program test cases will be prepared and for each of these test cases, test data will also be prepared. The actual

developers will do a first cur checking at this stage to see that the programs written by them are error free.

- **Testing**

In this stages the test group of the development team, using the cases and the test data already prepared will test the programs. Only after all the functions are tested singularly, an integrated testing will be performed to see that inter-function dependability is satisfied. Separate test cases and test data will be worked out for the integrated testing.

- **Acceptance Test**

This round of testing will be performed by the test group formed by the users of MLM software. This test group has to insure that the developed software is working as per their requirements. If some problems are found then it should be immediately communicated Development group so that the problem can be looked into and hence rectified.

- **Data Creation**

For software, data is most important part. Data is information which is handled by software. So before coding software, all master table data will have to be created.

- **Implementation**

Now the implementation of software is to be done by programmers. All the requirements and information gathered by the analyst is now take actual image in form of software. After making software it is uploaded in to the system so users, for whom software is developed, can use the software.

Once we examine that the project is feasible, we undertake project planning. The table below describes how we planned our project.

2.1.3Schedule Representation

#	Phases	Time Period
1	Project Summary	01/02/2011
2	Project Plan	08/02/2011
3	Risk Analysis	15/02/2011
4	Effort Estimation	15/02/2011
5	System Requirement Study	01/03/2011
6	Feasibility Study	08/03/2011
7	Data and function Modeling	29/03/2011
8	Testing	29/03/2011

2.2 Risk Management

2.2.1 Risk Identification

During the project plan we have consider all the proactive which we have think we will face during the project period. Here I have listed the risks which we have considered during the project plan:

- ❖ Possibility that the components are not available during the project period.
- ❖ Possibility that products purchased was not compatible.
- ❖ Possibility that the hardware resources are not available during the project period.
- ❖ Possibility that Ethernet connection between robot controller and computer may not occur, to which we have to make the connection might not available.
- ❖ Possibility that software inter-compatibility may not be there.

2.2.2 Risk Analysis

Risk analysis is the important aspect of the project planning, whenever planning the software, programmer always has to consider the risks of the projects which he might face in the future during designing the software.

Risks are of two types

- Proactive Risk.
- Reactive Risk.

Risk Impact

Risk	Effect
Possibility of getting illness.	Serious.
Possibility of component not available.	Catastrophic
Possibility of component not effective.	Serious.
Possibilities of hardware resources are not available.	Catastrophic.
Possibility of robot controller not available.	Serious.
Possibility of communication between systems might not happen on time.	Serious

Proactive Risk Assessment Table

This are the proactive risks which we can consider during the project plan period so we can cop up with them easily and we can find the solution easily but we can find other proactive risks which we haven't considered in the project plan:

Risk	Effect
Possibility that because of some problem we have lost the backup of some days.	Serious
Possibility that because of finance problem in the company we won't get the full resources.	Catastrophic
Possibilities that project Manager leave the project.	Serious

Reactive Risk Assessment Table

Proactive risks are most dangerous risks which we haven't considered during the project period and to cop up with them is not easy. These kinds of risks are risks, so programmer must have been active to cop up with them smartly.

Risk Estimation

Risk estimation attempts to rate each risk in two ways:

- The likelihood or probability that risk is real and.
- The consequences of the problems associated with the risk should it occurs.
- The project planner, along with other managers and technical staff, performs four risk projection steps:
 - ❖ Establish a scale rate that reflects the perceived likelihood of a risk.
 - ❖ Delineate the consequences of the risk.
 - ❖ Estimate the impact of the risk on the project and the product.
 - ❖ Note the overall accuracy of the risk projection so that there will be no misunderstanding.

The intent if these steps are to consider risks in a manner that leads to prioritization. No software team has the resources to address every possible risk with the same degree of rigor. By prioritizing risks, the team can allocate resources where they will have the most impact.

2.2.3 Risk Planning

After estimating all the risks and risk effects, we will look how to manage this kind of risks:

With the kind of proactive risks we will prepare the plan for how to manage these risks, we will think about their options, if their kinds of risks are facing and if they are reactive risks we have to manage them on the spot to avoid their dangerous effects:

Risk	Risk Management
Possibility of getting illness.	We will provide some more time (approx. one week) to prepare the software.
Possibilities of the components are not available.	We will check for other components, which have the same effect as those components.
Possibility of component not effective.	We will manage for the new components of that kind.
Possibilities of hardware resources are not available	We will manage for optional hardware, while starting the project.
Possibility of robot controller not available.	We try to arrange a new controller.

2.3 Estimation

2.3.1 Effort Estimation

Effort estimation methods are one of the important tools for project managers in controlling human resources of ongoing or future software projects. The estimations require historical project data including process and product metrics that characterize past projects.

Software cost and effort estimation will never be an exact science. Too many variables human, technical, environment, political can affect the ultimate cost of software and effort applied to develop it. However, software project estimation can be transformed from a black art to a series of systematic steps that provide estimate with acceptable risk. To achieve reliable cost and effort estimates, a number of options arise:

❖ Software Sizing

- Function point sizing
- Standard component sizing

❖ Problem-Based Estimation

LOC and FP data are used in two ways during software project Estimation:-

- As an estimation variable to size each element of the software and
- As baseline matrices collected from past projects and used in conjunction with estimation variables to develop cost and effort projections.

❖ Schedules

Obtain an early view of staffing requirements and constraints, and demonstrate the impact of changing deadlines, understaffing, and staff loading.

❖ Quality

Quantify the impact on defect rates of building to deadlines or reducing staff.

❖ Risk

Fine tune risk levels for all the major types of risk: size, requirements, technology, maintenance, systems integration, and defects.

3.0 System Requirements Study

3.1 User Characteristics

There are 4 types of user dealing with the system.

User A? Administrator

Administrator: Admin is having all the rights on the application.

User B? Employee

Employee: Employee of the company is one of the 4 users of this project

User C? Anonymous User

Anonymous User: Anyone who visits website. And any person applying for the posted job on the website.

User D? Client

Client: This is the registered user. Who come to know about his/her project's progress

3.2 Hardware and Software Requirement:

Hardware Specification:

Processor	: Intel Dual based system
Processor Speed	: 1GHz to 2 GHz
RAM	: 256MB to 512 MB
Hard Disk	: 4 GB to 30 GB
Keyboard	: 104 keys

Software Specification:

Language : JDK 1.6
Database : Oracle 9i
Operating System : Windows NT/XP/Vista
RAM : 512 MB

3.3 Constraints:

General Constraints

- 1) This system will not take care of any virus problem that might occur on the computer with which it is installed. Avoiding the use of pirated/illegal software and ensuring that floppies and other removable media are scanned for viruses before use could minimize the possibility of viral infection.
- 2) Recovery of data after a system crash will be possible only if backups are taken at regular intervals.
- 3) Manual interfaces cannot be fully avoided. Documented proofs like dates etc. will have to be verified by the concerned staff before entering it into the computerized system

Hardware Constraints

The performance of the system will be dependent on the machine conditions. The primary memory (RAM) and the secondary memory (Hard Disk Space) requirement of the system will be the same as that required by the normal application and the operating system. And the space required storing the data. The space required to store the data would increase as more and more records are added to the system.

Assumptions and Dependencies

- a. It is assumed that the user is familiar with the basic computer fundamentals.
- b. Timely backup of data should be taken to avoid data loss in case of system crash.
- c. Floppies and other removable media should be scanned for viruses before use.
- d. It is assumed that the maintenance of the database will be assigned to the authorized person only.
- e. Only authorized persons will be allowed inside the system

4.0 System Analysis

4.1 Study of Current System

The OBS Administration falls short of controlling the employee's activities in analyzing his/her strengths and weakness. The decision for appraisal of assigning next project to the employee or to train him/her to enhance the skills – where lies with proper projection. He is not provided with the detailed project information done or to be assigned based on Application / Verticals.

4.2 Problem and Weaknesses of Current System

- Need of extra manual effort.
- It used to take much time to find any employee
- Not very much accurate.
- Danger of losing the files in some cases.

4.3 Requirements of New System

Decision in assigning proper skillful hands for the project is an important issue in OBS Module. The OBS Administrator should report with the personal holding the necessary skills required for the project assignment. The decision in making analysis about the employee's skills is a prime important before booting in. The proposed system of OBS Module is the right software to be incorporated into the Automation of OBS Software for helping the organization needs with respect to skilful Human Resource.

The proposed system provides detail general information about the employee along with Educational, Certification, Skill and Project details. It enhances the OBS Management in adding, viewing and updating employees' details and generates various

reports regarding employee's skill and experience. Suggestions and Grievances posted by the employees are upheld for taking care of the necessary steps in forwarding company's obligation.

ADVANTAGES OF PROPOSED SYSTEM:

- Very fast and accurate.
- No need of any extra manual effort.
- No fever of data loss.
- Just need a little knowledge to operate the system.
- Doesn't require any extra hardware device.
- At last very easy to find the employees.

4.4 FEASIBILITY STUDY:

Once the problem is clearly understood, the next step is to conduct feasibility study, which is high-level capsule version of the entered systems and design process. The objective is to determine whether or not the proposed system is feasible. The tOBSee tests of feasibility have been carried out.

- Technical Feasibility
- Economical Feasibility
- Operational Feasibility

❖ TECHNICAL FEASIBILITY

In Technical Feasibility study, one has to test whether the proposed system can be developed using existing technology or not. It is planned to implement the proposed system using java technology. It is evident that the necessary hardware and software are available for development and implementation of the proposed system. Hence, the solution is technically feasible.

❖ ECONOMICAL FEASIBILITY

As part of this, the costs and benefits associated with the proposed system compared and the project is economically feasible only if tangible or intangible benefits outweigh costs. The system development costs will be significant. So the proposed system is economically feasible.

❖ OPERATIONAL FEASIBILITY

It is a standard that ensures interoperability without stifling competition and innovation among users, to the benefit of the public both in terms of cost and service quality. The proposed system is acceptable to users. So the proposed system is operationally feasible.

4.5 Requirements Validation:

Requirement Validation examines the specification to ensure that all system requirements have been stated unambiguously; those inconsistencies, errors have been detected and corrected and the work products conform to the standard.

There are many requirements from user perspective and taken care while designing a system, are as follows:

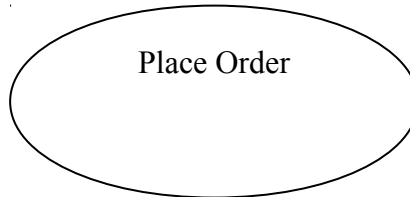
- Dynamic nature of system. i.e. System change its working depending on situation.
- Component based definition of system. i.e. System is divided into smaller components which will work independently also there combined effort is result into output of system.
- Flexibility of system. i.e. System should work with great ease with different types of documents.
- Flexible Database design should be done to accommodate information about new input.
- Flexible front-end design so that it can support functionality of all types of input.
- Back-end should not affect front-end or vice versa.
- All Database changes should be done by front-end only.
- Simplicity should be there in system design.
- User friendliness should be achieved.
- System should be easily Maintainable and Adaptive.
- Design for such system is created in such a way that related information is kept in same tables.
- Different information related to different component is stored in different tables.
- To make the task of data entry easy various combo boxes and list The Database design should support the accommodation of new component information in a way, such that it should allow to continue the existing relationship with the other components and other parts of system. boxes are designed so that user can have to just select the values from the given options.

4.6 Functions of System:

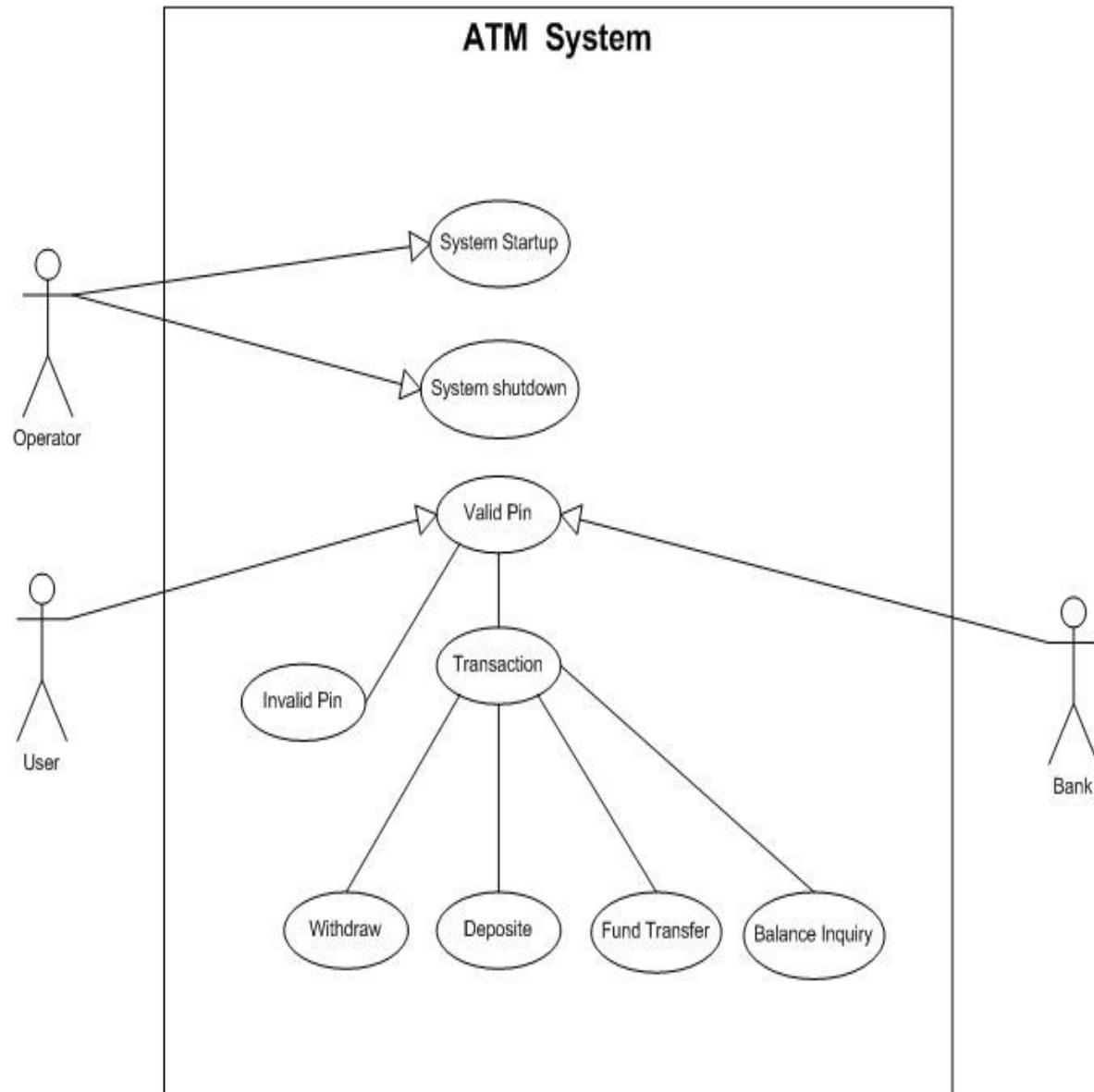
Use Case:

Use case is a description of a set of sequence of actions that a system performs that yields an observable result of value to a particular thing in a model.

Graphically, Use Case is rendered as an ellipse with dashed lines, usually including only its name as shown below.



USE CASE DIAGRAM:-



4.7 Data Modeling:

4.7.1 Flow chart or activity design:

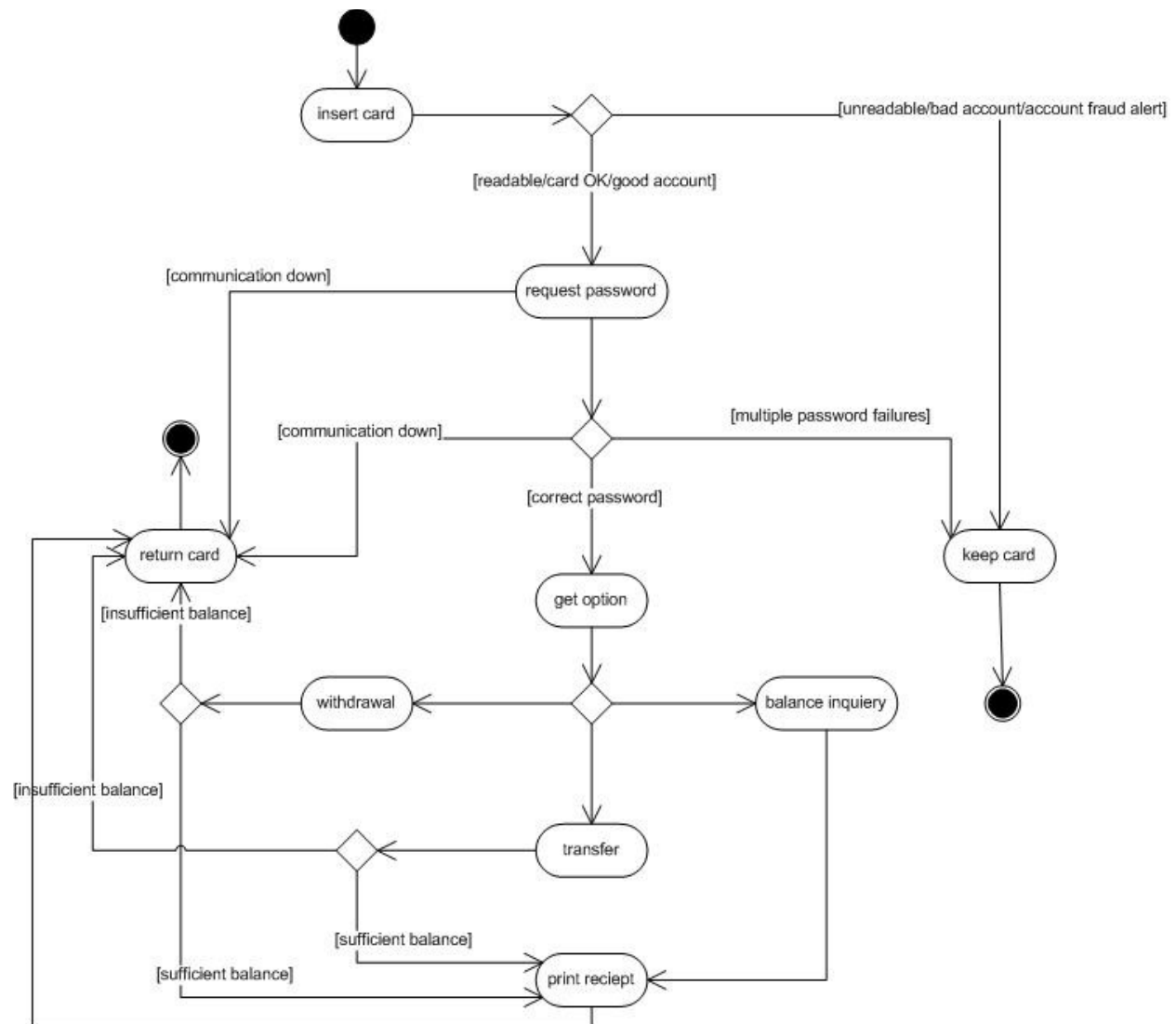
ACTIVITY DIAGRAM:

An Activity Diagram is essentially a flow chart showing flow of control from activity to activity. They are used to model the dynamic aspects of a system. They can also be used to model the flow of an object as it moves from state to state at different points in the flow of control.

Content:

Activity diagrams commonly contain: Fork, Start & End Symbol

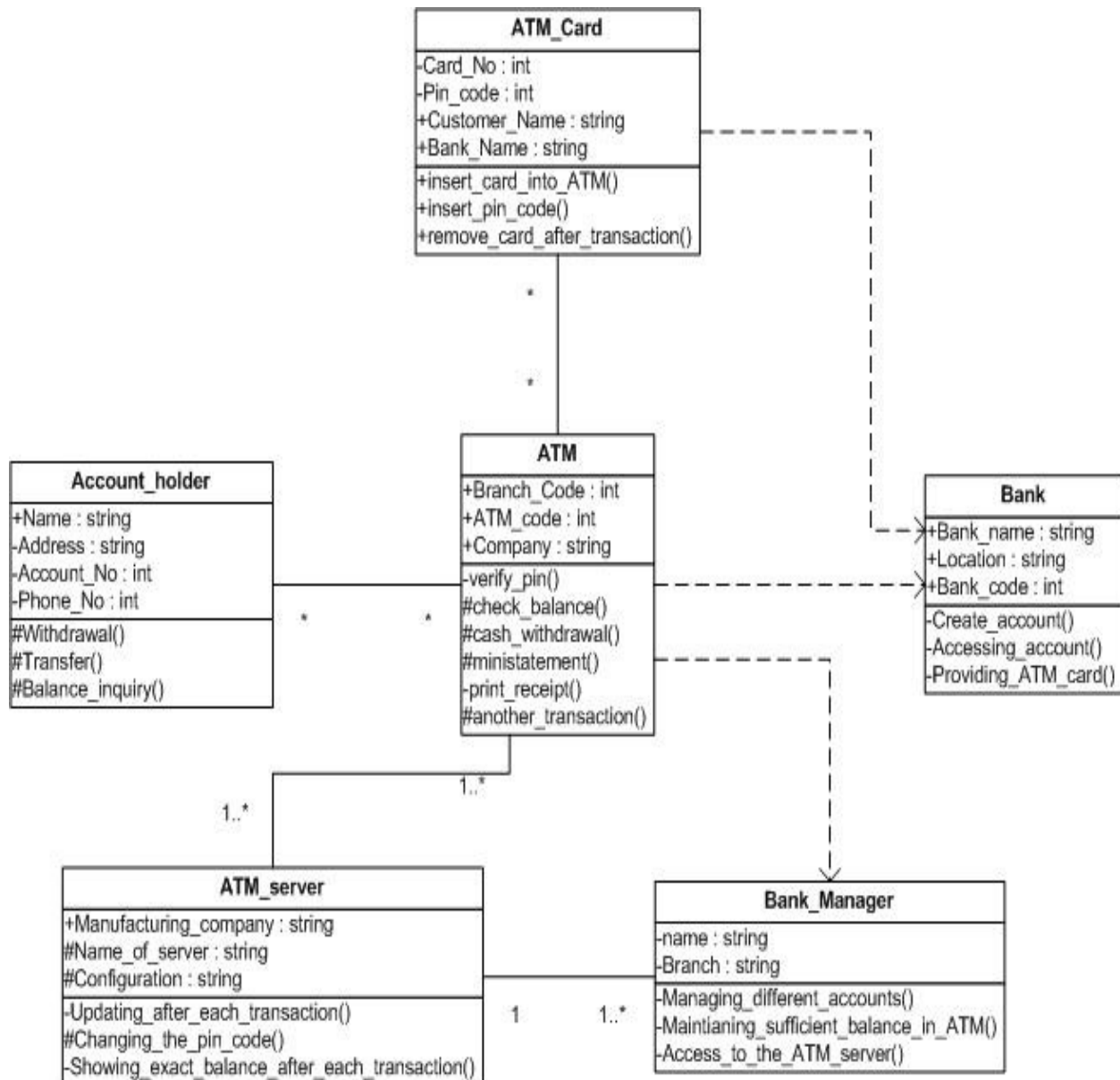
ACTIVITY DIAGRAM:-



4.7.2 Class Diagram/ E-R diagrams:

Class diagrams are the most common diagrams found in modeling object-oriented systems. A class diagram shows a set of classes, interfaces, and collaborations and their relationships. Graphically, a class diagram is a collection of vertices and arcs.

CLASS DIAGRAM:-



Class Diagram is a graph that represents the relationship between the classes and represents their semantics.

❖ Here ATM works as main class. All other classes are related with this class.

ATM does following operations:

- Verify_pin()
- Check_balance()
- Cash_withdrawal()
- Ministatement()
- Print_receipt()
- Another_transaction()

❖ ATM card related with ATM through many to many relationship.

It does following operations:

- Insert_card_into_ATM()
- Insert_Pin_code()
- Remove_card_after_transaction()

❖ Account holder related with ATM through many to many relationship.

It performs following operations:

- Withdrawal()
- Transfer()
- Balance_inquiry()

❖ ATM server related with ATM by one or many to one or many relationship. It also performs some task shown as below:

- Updating_after_each_transaction()
- Changing_the_pin_code()
- Showing_exact_balance_after_each_transaction()

❖ Bank Manager associated with ATM through one or many to one relationship. ATM is not related with this class, but dependent on this class. So there is a dependent relationship assigned to them. Bank Manager does following tasks:

- Managing_different_accounts()
- Maintaining_sufficient_balance_in_ATM()
- Access_to_the_ATM_server()

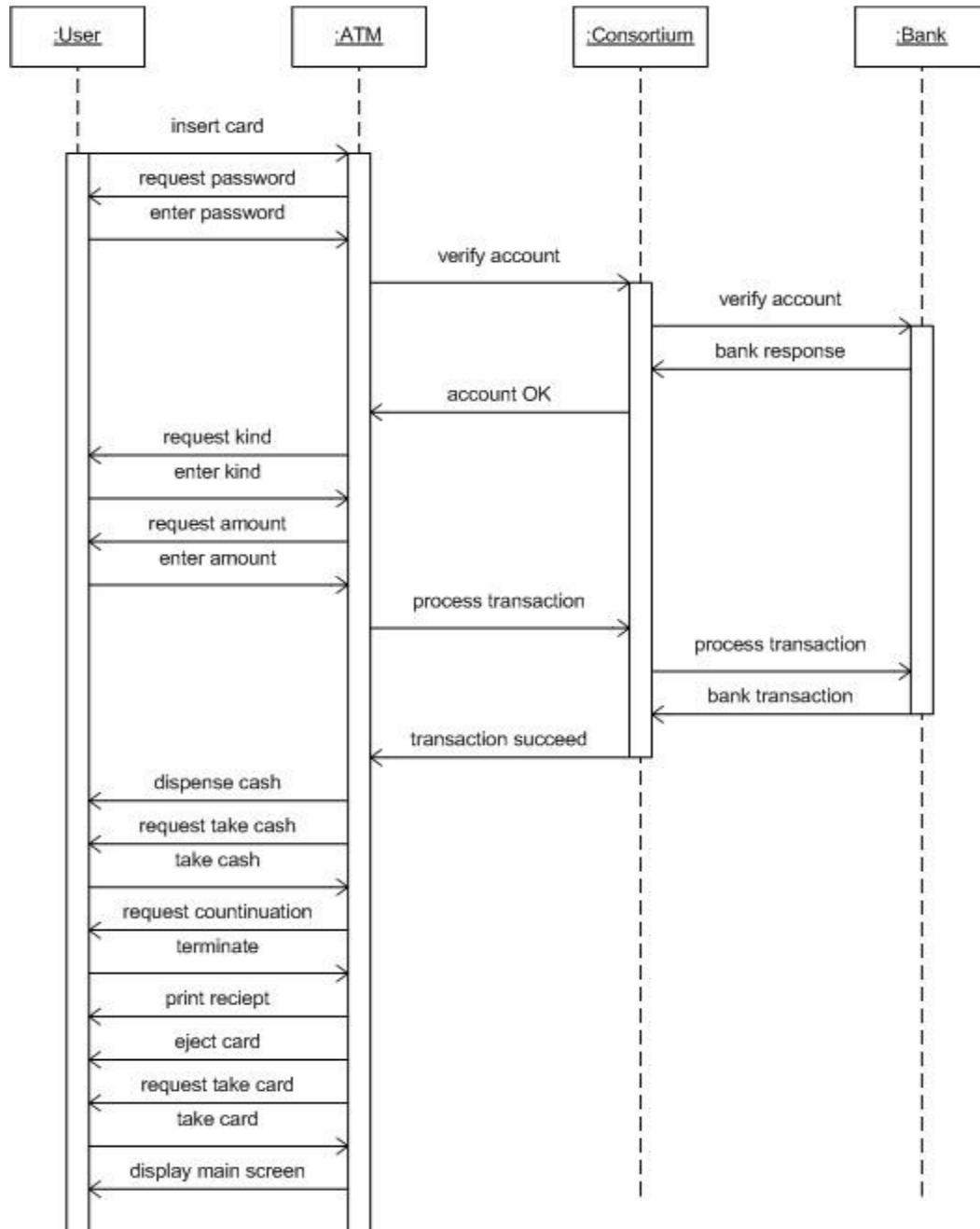
❖ Here, a class Bank not related to any of the class, but some classes are dependent on these classes which are shown as dependent relationship with it. Normally this class performs following operations:

- Create_Account()
- Accessing_Account()
- Providing_ATM_card()

4.7.3 System Activity or Object interaction Diagram:

An Interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them. Interaction diagrams are used for modeling the dynamic aspects of the system.

INTERACTION DIAGRAM:-



4.7.4 Data Dictionary:

A Data Dictionary is a Catalogue – a repository of element in a system. As the name suggest, these elements center around data and the way these are structured to meet the user requirement and system needs. In Data Dictionary we find list of all the elements are data flows, data stores, and process. The Data Dictionary stores details and description of these elements.

If anybody wants to know how many characters are in a data item, by what other names it is referenced in the system, or where it is used in the system, they should be able to find the answer in a properly developed data dictionary.

The Data Dictionary is developed during data flow analysis and assists the system development in determining the user requirement.

✓ **Importance of Data Dictionary:-**

Analysis use data dictionary for five important reasons:

1. To manage the detail in large system.
2. To communicate a common meaning for all system elements.
3. To Document the feature of the system.
4. To Facilitates analysis of the details in order to evaluate characteristics and determine where system changes should be made.
5. To Locate error and omissions in the system.

The Following Tables are used within the system:

1. ATM ::

It provides information about ATM's Bank, Branch and Company.

ATM

Branch_Code	Integer	Code of particular Branch
ATM_Code	Integer	Code of particular ATM
Company	String	Name of Manufacturer of ATM

2. ATM_Card ::

It provides information about ATM card, its holder and Bank.

ATM_Card		
Card_No	Integer	No. of particular Card
PIN_Code	Integer	Secrete code of card
Customer_Name	String	Name of Card Holder
Bank_Name	String	Name of Bank

3. Account_holder ::

It has all the details about ATM Card holder.

Account_holder		
Name	String	Name of Account owner

Address	String	Address of Account owner
Account_No	Integer	Account no. of card holder
Phone_No	Integer	Phone no. of Account owner

4. ATM_server ::

It has detail of ATM. As well as it provides interface between Bank and Account holder.

ATM_server		
Manufacturer_Company	String	Name of Manufacturer Company
Name_of_Server	String	Server name to identify it
Configuration	String	Configuration set by Bank Manager

5. Bank_Manager ::

It contains data of managers who set and manage the server of ATM.

Bank_Manager		
Name	String	Name of Manager
Branch	String	Branch name where manager do work

6. Bank ::

It provides detail of Bank to get information about Account of a user.

Bank		
Bank_Name	String	Name of Bank
Location	String	Location of Bank
Bank_Code	Integer	Code of Bank to identify it

4.8.1 Context Diagram

The top-level diagram is often called a “*context diagram*”. It contains a single process, but it plays a very important role in studying the current system. The context

diagram defines the system that will be studied in the sense that it determines the boundaries. Anything that is not inside the process identified in the context diagram will not be part of the system study.

4.8.2 Data Flow Diagram (0 and 1 level)

A graphical tool used to describe and analyze the movement of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also known as a data flow graph or a bubble chart.

A graphical tool used to describe and analyze the movement of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also known as a data flow graph or a bubble chart.

TYPES OF DATA FLOW DIAGRAMS:

Data Flow Diagrams are of two types as follows:

- (a) Physical DFD
- (b) Logical DFD

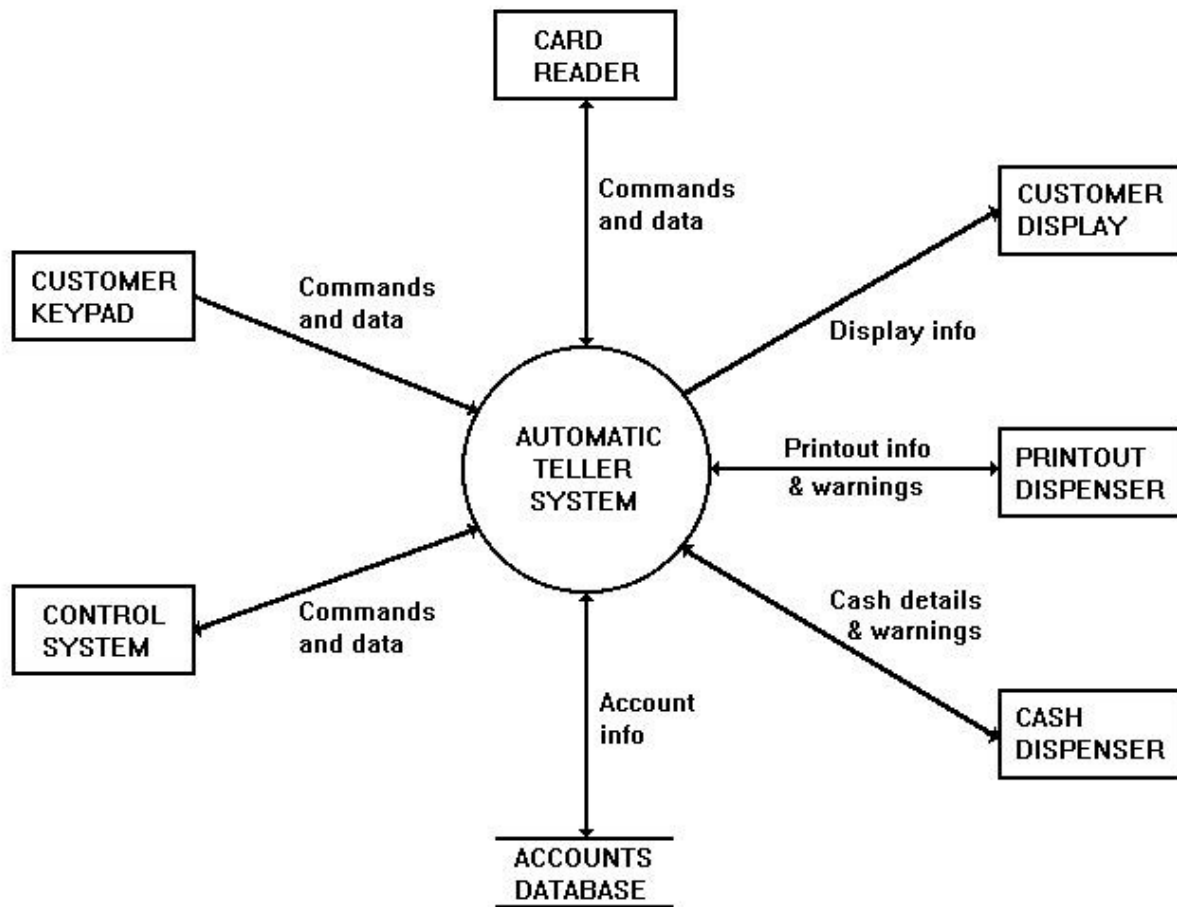
1. PHYSICAL DFD:

Structured analysis states that the current system should be first understood correctly. The physical DFD is the model of the current system and is used to ensure that the current system has been clearly understood. Physical DFDs show actual devices, departments, and people etc., involved in the current system.

2. LOGICAL DFD:

Logical DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts.

Data Flow Diagram(Level-0):-



This diagram shows the Automatic Teller System Software and the hardware that it interacts with. The arrows show the direction and type of data flowing between the software and each hardware element.

External Entities:-

CONTROL SYSTEM

This system enables and disables the customer interface and receives customer requests and system reports. A suitable Control System would be a personal computer linked to a central computer system with access to the Accounts Database. The customer interface (keypad, display, etc) is controlled by enabling and disabling the Card Reader, which is the customer's entry-point to the system. Requests for statements and chequebooks are posted to the Control System. It also receives status reports for low printer-paper and cash levels.

ACCOUNTS DATABASE

This is a database containing account numbers, balances and other account information. Data is retrieved from the database when a customer requests a balance report or a cash withdrawal. The database is updated after a withdrawal.

CARD READER

The Card Reader receives the customer's card and retrieves the PIN and account number stored on it. This information is transmitted to the software system which enables the Customer Keypad and initiates the PIN verification procedure. When business is completed the Card Reader is instructed to return the card. If the customer enters an incorrect PIN, a fixed number of retries is permitted, after which the Card Reader is instructed to confiscate the card.

CUSTOMER KEYPAD

The Customer Keypad allows a customer to enter a PIN number, select options and enter cash values. The keypad is only enabled when a card is detected in the Card Reader.

CUSTOMER DISPLAY

The Customer Display presents messages, options and reports to the customer. The display is active at all times.

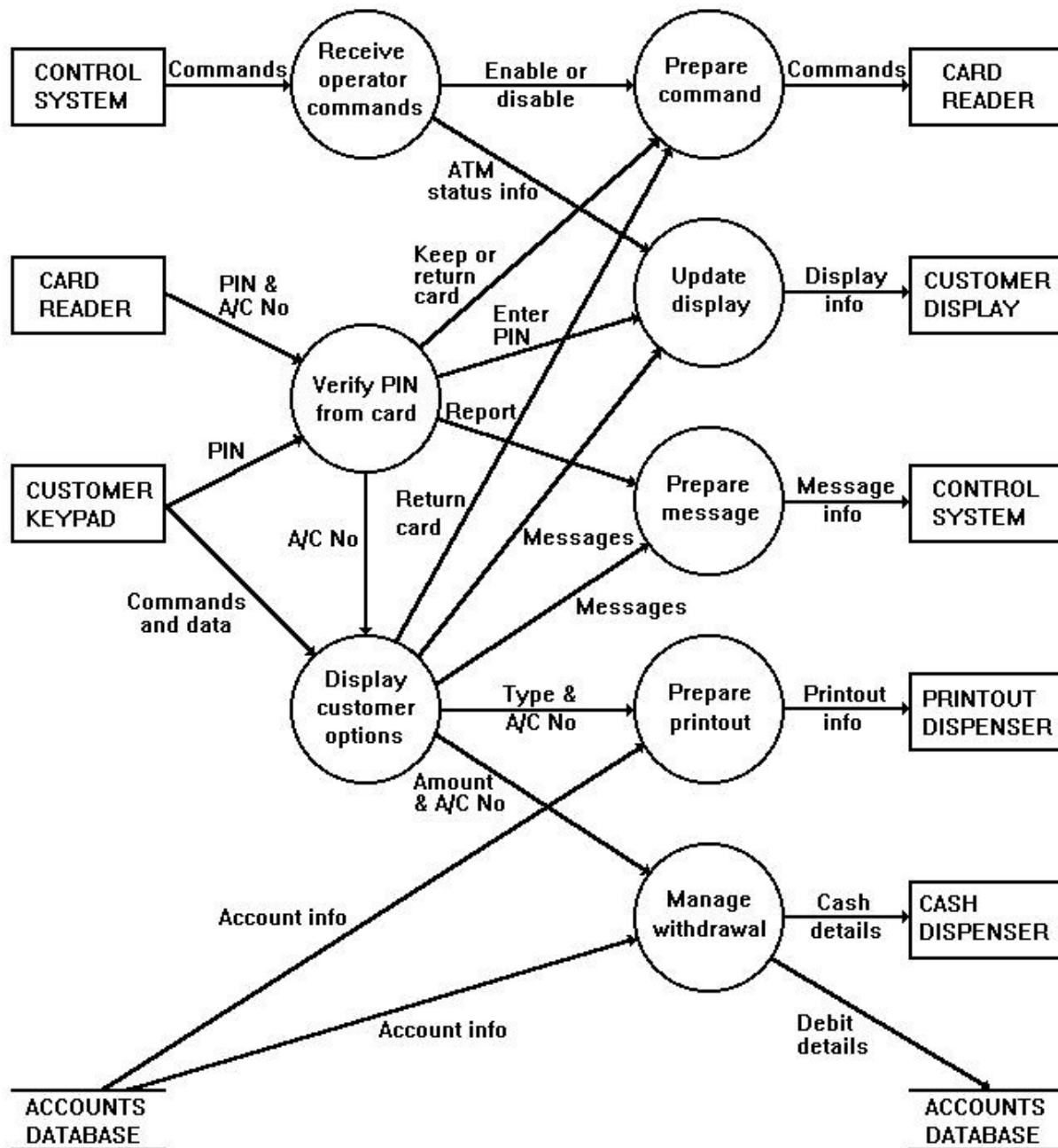
PRINTOUT DISPENSER

This provides the customer with a printed balance or receipt. The Printout Dispenser reports to the system if the paper level is low.

CASH DISPENSER

This assembles and delivers cash to the customer. The dispenser receives information about the values and quantities of notes to dispense (1 x \$20, 2 x \$5, etc). The Cash Dispenser reports to the system if the cash levels are low.

Data Flow Diagram(Level-1):-



This diagram shows data entering and leaving the system. Input data is received from the hardware elements on the left. Various types of data are processed by different parts of the software system. Output data is sent to the elements of hardware on the right.

DFD Level 1 Processes:-

Interact With Operator

This process deals with commands from the system operator. These are the commands which enable or disable the customer interface by controlling the Card Reader. The operator may issue these commands from another computer system or by using a switch on a control panel.

Interact With Customer

This process handles all interactions with the customer and operates only when a card is detected in the Card Reader. Input is received initially from the Card Reader and then directly from the customer via the Customer Keypad. The customer receives output from the Customer Display, the Printout Dispenser and the Cash Dispenser. Customer interactions may also involve sending reports to the Control System. The initial step of all customer interactions is to verify the customer's PIN number. After this a menu of options is presented on the display which the customer selects by pressing appropriate keys on the keypad. These options lead to other displays and requests for further input. Some options require account details which are retrieved from the Accounts Database and may also involve updating the database. During the final stage of all customer interactions the Card Reader is instructed to either return or confiscate the card.

Prepare Command

This process handles communication with the Card Reader hardware. The system requires that the Card Reader is able to receive the following commands:

ENABLE	Makes the Card Reader ready to receive a card
DISABLE	Prevents the Card Reader from accepting a card
RETURN	Ejects a card from the Card Reader
RETAIN	Confiscates an unauthorized card

The Card Reader is enabled and disabled by commands from the system operator. A card is returned or retained in response to interactions with the customer.

Update Display

This process deals with the Customer Display screen. When no card is in the Card Reader, the Customer Display shows general information (such as 'Insert Card'). When a card is detected the display is updated in response to customer interactions. If the system is disabled by the system operator, the display is updated to indicate the system status.

The following is a list of screens which are shown on the Customer Display.

General Information	-Insert Card and other messages
---------------------	---------------------------------

PIN Verification	- Enter PIN message
Main Options	- Some or all of the following; Display Balance, Print Balance, Cash With Receipt, Cash Without receipt, Order Statement, Order Chequebook, Return Card
Current Balance	-Customer's account balance and cleared funds
Balance Printed	-Take Your Balance message
Withdrawal Options	-Pre-defined cash amounts and Other Amount
Cash Amount	-Enter Amount for cash withdrawal
Cash Dispensed	-Take Your Cash message
Receipt Printed	-Take Your Receipt message
Statement Ordered	-Statement Ordered message
Chequebook Ordered	- Chequebook Ordered message
Card Returned	-Take Your Card message
Card Retained	-Card Retained message (for failed PIN Verification)

Prepare Message

This process prepares and transmits messages to the Control System. These messages can be requests from customers for statements and chequebooks or reports concerning the levels of printer-paper and cash.

Prepare Printout

This process prepares and controls the use of the Printout Dispenser to produce balance reports and receipts. The customer's balance is retrieved from the Accounts Database (if required). If the printer-paper level becomes low options which involve printouts are disabled and a warning message is sent to the Control System.

Manage Withdrawal

This process receives requests for withdrawals of specific amounts from a certain account and operates the Cash Dispenser. Before proceeding, the customer's details in the Accounts Database are checked. If the request exceeds the customer's balance (or agreed overdraft) the withdrawal is denied. The system uses a denomination selection algorithm based on the notes available and the amount required. The Accounts Database is updated after each withdrawal. If the cash level becomes low, options providing cash withdrawals are disabled and a warning message is sent to the Control System.

4.8.3 Process Specification and Decision Table

1. Admin Login

If admin_login=true
Then give rights to access the application as well as admin functionalities
Else
Show error message
End if

2. User login

If user_login=true
Then give rights to access the application
Else
Show error message
End if

3. Entry of details of respective forms to save in database

If information entered = valid
Then save
Else
Show error message
End if

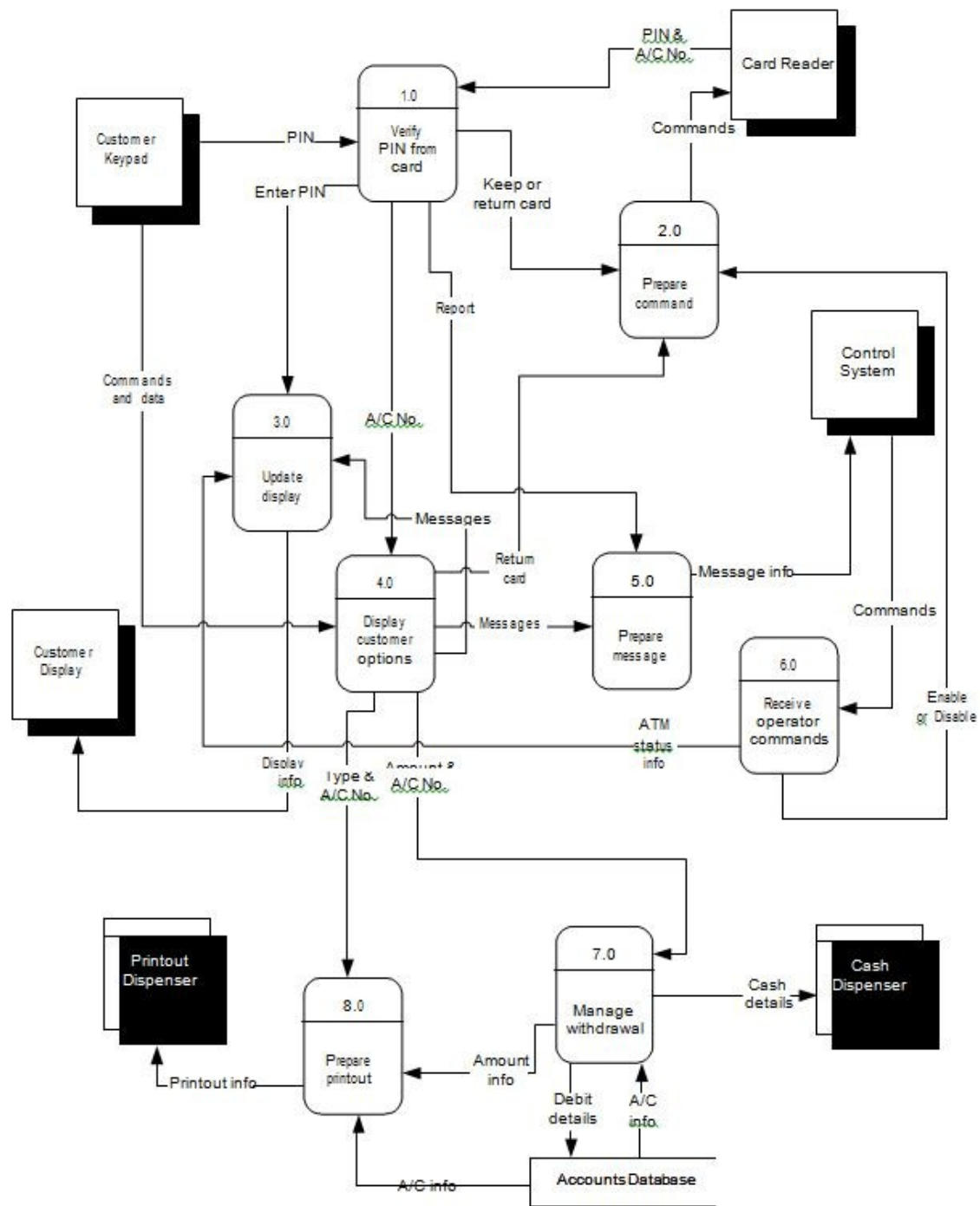
4. Entry of details to update the records in database

If information updated = valid
Then do update in database
Else
Show error message
End if

5. Generate reports

If parameter for report = valid
Then view the records
Else
Show error message
End if

4.8.4 Control Flow Diagram:-



5.0 Testing

5.1 Testing Plan:

Any system, to be successful, must be thoroughly tested, and well managed test plan should be prepared before actual testing is being performed. “Modules” have been developed and need to be tested in a manner that can reduce occurring of defects as low as possible. Following are the activities we planned to test the system.

1. This system is indeed an evolutionary system so every unit of the system is continuously under testing phase.
2. One test activity “Basis Path Testing” that will try to cover all paths in the system. This activity identifies all paths that provide different functionality of the system, and also other paths to reach at that functionality.
3. Other testing activity is ”Control Structure Testing”, which will test each and every condition with positive and negative data combination.
4. This testing activity will also perform “Data Floe Testing” in which it will be tested how the data re following the system. And will also check whether the data entered from one procedure, is reflected whenever it requires or not.
5. All conditions will be tested with “Boundary Value Analysis” where different input will be given to test whether the system is functioning with boundary values or not.
6. Along with the boundary value analysis, the system is also tested with “Range Value Tested” where editable values will be tested with ranges of values.
7. The system is being tested in “Unit Testing” manner where at the completion of one unit that is tested thoroughly with above mentioned testing activities.
8. The integration testing will also be performed to ensure that the integrated unit is working properly with other units or not.

5.2 Testing Strategy:

CONTENT TESTING:

Errors in Project content can be as trivial as minor typographical error as incorrect information, improper organization or validation of intellectual property laws. Content Testing attempt to uncover this and many other problems before the user encounter them.

Content Testing Objectives:

There are three types of objectives.

- 1) To uncover syntactic errors in text-based documents, graphical representation and other media.
- 2) To uncover semantic errors in any content object represented as navigation error.
- 3) To find errors in organization or structure of content that is presented to the end-user

INTERFACE TESTING

Interface design model is reviewed to ensure that generic quality criteria established for all user interfaces have been achieved and that application specific interface design issue has been properly addressed.

Interface testing strategy:

The overall strategy for interface testing is to (1) Uncover error related to specific Interface mechanisms (2) uncover errors in the way the interface implements the semantics of navigation, WebApp functionality, or content display. To accomplish this strategy, a number of objectives must be achieved:

- Interface features are tested to ensure that design rules, aesthetics and related visual content are available for the user without error.
- Individual interface mechanisms are tested in a manner that is analogous to unit testing For example; tests are designed to exercise all forms, client-side scripting, dynamic XML.

Testing Interface Mechanisms:-

When a user interacts with a system, the interaction occurs through one or more interface mechanisms.

Forms: -

At a microscopic level, tests are performed to ensure that

1. Labels correctly identified fields within the form and that mandatory fields are identified visually for the user.
2. The server receives all information content within the form and their no data are lost in the transmission between client and server.
3. Appropriate defaults are used when the user does not select from a pull down menu or set of buttons.
4. Browsing functions don't corrupt data entered in the form.
5. Scripts that perform error checking on data entered work.
6. Properly and provide meaningful error message.

Client side scripting:-

Black box tests are conducted to uncover any error in processing as the script is executed. These tests are coupled with forms testing because script input is often derived from data provided as part of forms processing

Dynamic XML:-

Here the data transfer between the systems occurs in the form of xml file. The formatting of the XML file should remain unaltered. Changes in the internal attribute structure can also lead to an error. Hence for this XML parsing functionalities are provided on the client side.

Application specific interface mechanisms:-

Test conforms to a checklist of functionality and features that are defined by the interface mechanism.

- Boundary test minimum and maximum number of item that can be placed in to shopping chart.

- Test to determine persistence of image capture contents.
- Test to determine whether the system can be record co-ordinate content at some future date.

USABILITY TESTING:-

Usability test may be designed by Project engineering team.

1. Define a set of usability testing categories and identify goal for each.
2. Design test that will enable each goal to be evaluated.
3. Select participants who will conduct test.
4. Instrument participant's interaction with system while testing is conducted.
5. Develop a mechanism for assessing the usability of the system.

The following test categories and objective illustrate establish testing:

Interactivity- Are interaction mechanism easy to understand and use?

Layout- Are navigation mechanism, content and function place in a manner that allows the user to find them quickly?

Readability- Is the text well written and clear?

Aesthetics- Do layout color, typeface, and related characteristics lead to ease of use?

Display Characteristics- Does the system make optimal use of screen size and resolution?

Time Sensitivity- Can important features, functions and content be used in a timely manner?

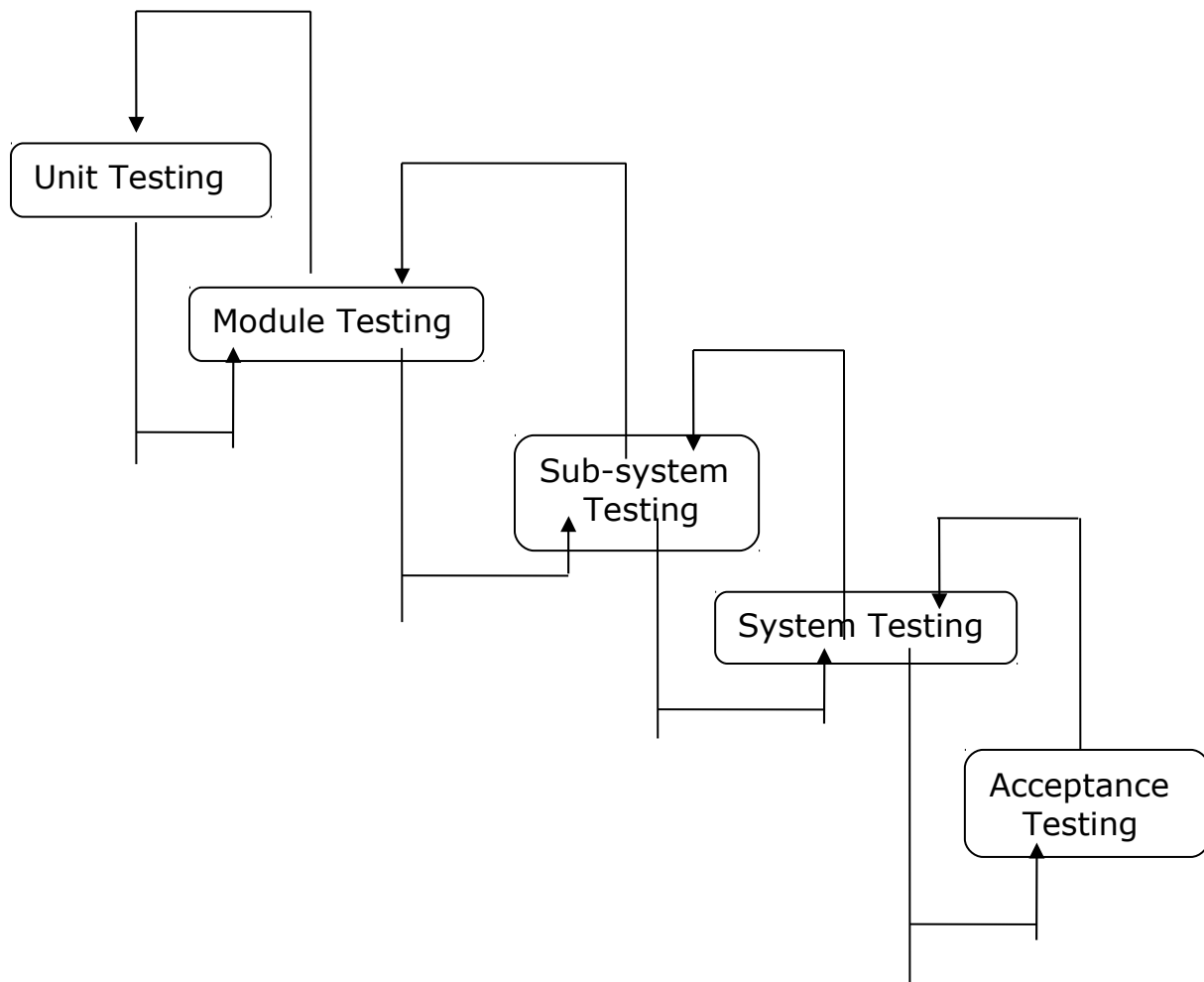
Accessibility- Is the system accessible to people who have Disabilities?

COMPATIBILITY TESTING:-

Project must operate within environment that differs from one another. Different computer, display device, OS, browser and network connection speed can have significant on system operation. The Project team derives a series of compatibility, validation tests, derived from existing interface tests, navigation tests, performance tests and security tests.

5.3 Testing Methods:

Analyze and check system representation such as the requirement document, design diagrams and the program source code. They may be applied at all stages of the process.



There are different Models of testing. On the basis of testing methods there are two types of testing:

1. White-box testing.

2. Black-box testing

1). WHITE-BOX TESTING

White-box testing sometimes called **glass-box testing**, is a test case design method that users the control structure of the procedural design to drive the test case.

- Logical errors and incorrect assumption are inversely proportional to the probability that a program will be executed. Errors tend to creep into our work we design and implement function, condition or control that is out of the mainstream tends to be well understood.
- We often believe that a logical path is not likely to be executed when in fact it may be executed on a regular basis. The logical flow of a program times counter intuitive.

2). BLACK-BOX TESTING:

For our project periodically we have tested our software using black-box testing. Thinking as a client we have evaluated the software for its easy going and convenience.

➤ **Unit Testing:**

During the programming stages each and every form, modules and class treated unit has been put into the test data. Every module is tested independently. The steps are follows:

1. Manually code is tested like spelling checks, logic and errors.
2. Once the manual checking is over the complication has been done. Syntactical errors if any have to be corrected.

3. After the clean compilation of the program, some dummy data, as specification, has been used for testing of the module to see if it works as specified.

Integration Testing

After our individual's modules were tested out we go the integrated to create a complete system. This integration process involves building the system and testing the resultant system for problems that arise from component interaction.

Performance Testing

Performance testing is designed to test the runtime performance of the system within the context of the system. These tests were performed as module level as well as system level. Individual modules were tested for required performance.

Condition Testing

Performance testing is a test case design method that exercises the logical conditions.

Interface Testing

Interface testing is integral part of integration. We examined the code to be tested and explicitly list each call to an external component. In the system standards tests for GUIs have been performed, which are as follows:

- The position and related labels for all controls were checked.
- Validations for all inputs were done.
- Pull down controls was verified for proper functionality.
- Whether the non-editable text controls disabling and it was also verified that it doesn't exceed the maximum allowed length.

5.4 Test Cases:

(1)Invalid Parameter Setting.

Test Case:-

If the cameras are calibrated in a wrong manner, then the values derived from the images would be incorrect.

Solution

Solution of this case is that we have to check the values from the .CSV file and see to that they are within the permissible range or not.

(2)Light source moves out of the camera plane.

Test Case:-

If the any user by mistake moves the light source out of the camera plane, the captured image would then not contain any co-ordinate information.

Solution

Solution of this case is that there is a condition kept when no co-ordinates are extracted pass on the value of the origin.

(3)If any of the cameras are not connected.

Test Case:-

If the cameras are disconnected from the computer the initialization of the process will not occur.

Solution:-

Solution of this case is that user has to be aware of all the hardware has been correctly connected to the system before initialization. User has to keep a checklist of all the system before initializing the system.

(4)If a proper base is not taught.

Test Case:-

If the user teaches an invalid base then there remain chances of the robot to collide with the surrounding.

Solution

Solution of this case is that user is not aware about the base, and then the user should execute the robot motion in T1 testing mode.

(5)Communication error between controller and computer.

Test Case:-

If user is unable to connect the computer to the controller, then user would not be able to pass on the co-ordinate values.

Solution

Solution of this case is that user has to first establish the connectivity by configuring the ApiConfig file, and check the connecting Ethernet cable. Successful connection can be known by using the PING command.

6.0 Limitation and Future Enhancement

Limitation:

- Although I have tried to add all the related features to this online Bus Reservation System but there are also some limitation.
- This system is stand alone system so data saved during different processes are stored in the machine in which that process was executed.
- So there is the problem of distributed database.

Future Enhancement:

- As discussed the limitation of this system, we can implement this as client/server system. So all the data will be stored in the single machine, and for any purpose all the data will be retrieved from this central database.
- So there will be no human work require for the employee. There will be only one person required who will maintain this central database.

Conclusion

Back in 1969, Chemical Bank announced that a new form of banking was being launched. With that, customers were provided with plastic cards designed with a magnetic strip that could be used with a machine built into a wall. Gone were the days of having to stand in line for a teller or not having money on hand after normal banking hours. Almost everyone has heard of and used an ATM machine. Interestingly, some of people feel that ATM machines are the best thing to happen in the banking world while other people consider them a curse. The main complaint heard about ATM machines is that while they are convenient, they are expensive to use. However, if we look at it from a banking perspective, business is business. Regardless of what we think of ATM machines, there is no doubt that they have changed the world and the way in which we do things. For example, think how many times we have been out somewhere only to discover we have no cash and we are out of checks, ah, but in the corner, there is an ATM machine. In the blink of an eye, we swipe the card and now have cash on hand. In addition to pulling money out, the ATM machine also makes it convenient to deposit money, transfer money, and check balances. Best of all, to use an ATM machine, we do not have to go to the bank. We will find ATM machines at other banks, grocery stores, shopping malls, along the roadside, Buckingham Palace, airports, in casinos, and even on the South Rim of the Grand Canyon. For this reason, ATM machines are extremely helpful!

Bibliography

Book Reference

- ➔ Fundamentals of Software Engineering
By Rajib Mall
- ➔ Software Engineering
By Ian Somerville
- ➔ Analysis and Design of Information Systems
By James Senn

Website Reference

- www.google.com
- www.ask.com
- www.microsoft.com
- www.iit.edu.com