

## FUNCTION-ORIENTED SOFTWARE DESIGN:

- OVERVIEW OF SA/SD METHODOLOGY
- STRUCTURED ANALYSIS
- DEVELOPING THE DFD MODEL OF A SYSTEM
- STRUCTURE DESIGN
- DETAILED DESIGN
- DESIGN REVIEW
- OVERVIEW OF OBJECT ORIENTED DESIGN

## USER INTERFACE DESIGN:

- CHARACTERISTICS OF GOOD USER INTERFACE
- BASIC CONCEPTS
- TYPES OF USER INTERFACES
- FUNDAMENTALS OF COMPONENT-BASE GUI DEVELOPMENT
- A USER INTERFACE DESIGN METHODOLOGY.

## INTRODUCTION :

- Software design begins after the requirement engineering is over.
- The SRS document is used to understand program organization.
- Function-oriented design is the most useful top-down design.
- Function is a subprogram of a software and all the functions are organized using function oriented design methodology.

## • OVERVIEW OF SA/SD METHODOLOGY:

→ Function oriented design methodology is a process oriented approach for developing software solutions. It has following two activities.

1. Structure Analysis (SA)
2. Structured design (SD)

→ The working of these two activities in function oriented design is schematically shown as follows.

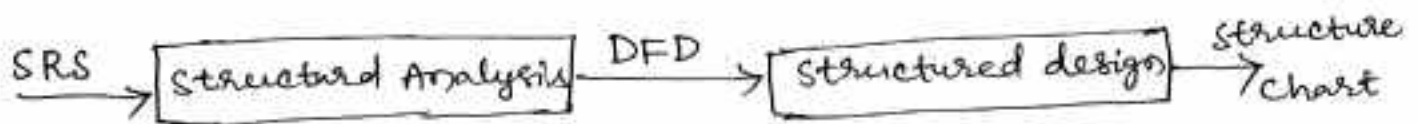


Fig: SA/SD methodology.

## • STRUCTURED ANALYSIS

→ structured analysis is also referred to as "process modelling" or "Data flow modelling".

→ Data flow modelling is useful to understand the working process of the system.

→ structured analysis uses a graphical tool called "Data flow diagram" (DFD).

### Data flow diagram (DFD):

→ A DFD is a graphical tool that describes the flow of data through a system and functions performed by the system.

→ A DFD has four different symbols processes, data flows, data store and actors.

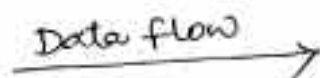
## Process

→ A process is represented by a circle and it denotes transformation of the input data to produce the output data.



## Data flow

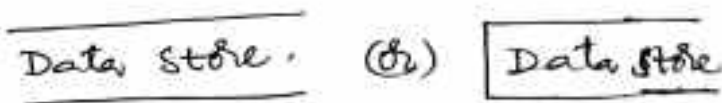
→ Data flows are ~~data~~ represented by arrows, connecting one data transformation to another.



## Data store

→ Indicate the updating of data store while outgoing arrows indicate the retrieval of data from the data store.

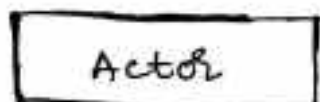
→ It is represented by an open-ended horizontal rectangle



## Actor

→ It is the external entity that represents the source or destination.

→ It is represented by a rectangle.



For example of DFD for cash withdrawal from an ATM machine is shown in figure.

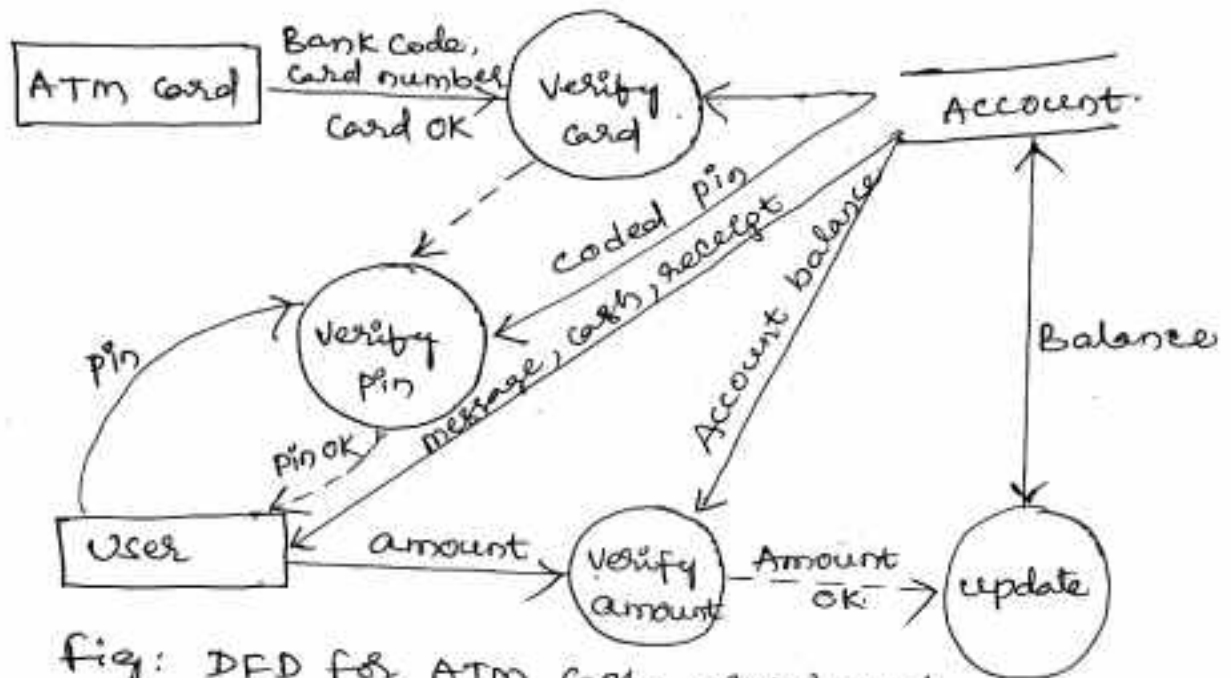


Fig: DFD for ATM cash withdrawal.

- A DFD consists of two actors; User and ATM Card. And four processes: Verify card, Verify pin, Verify amount, update balance. And a data store, i.e Account.
- The system verifies the card details, pin, and the amount in the User account. The card pin verification will be performed through matching with the coded details. Amount verification checks if there is sufficient amount in the account and that the amount to be withdrawn. The User will receive a message, cash, and a receipt after updating the account balance.

Levels in DFD diagram

Level 0 : It is also called as "context diagrams".  
It represents overview of whole system.

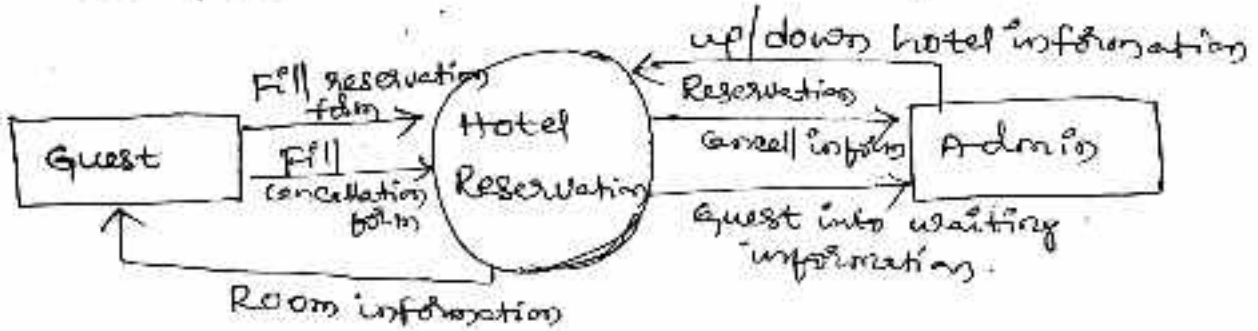


Fig: Level 0 DFD diagram for Hotel reservation system.

Level 1

provides more detailed breakout of pieces of the context level diagram (a) level 0 diagram.

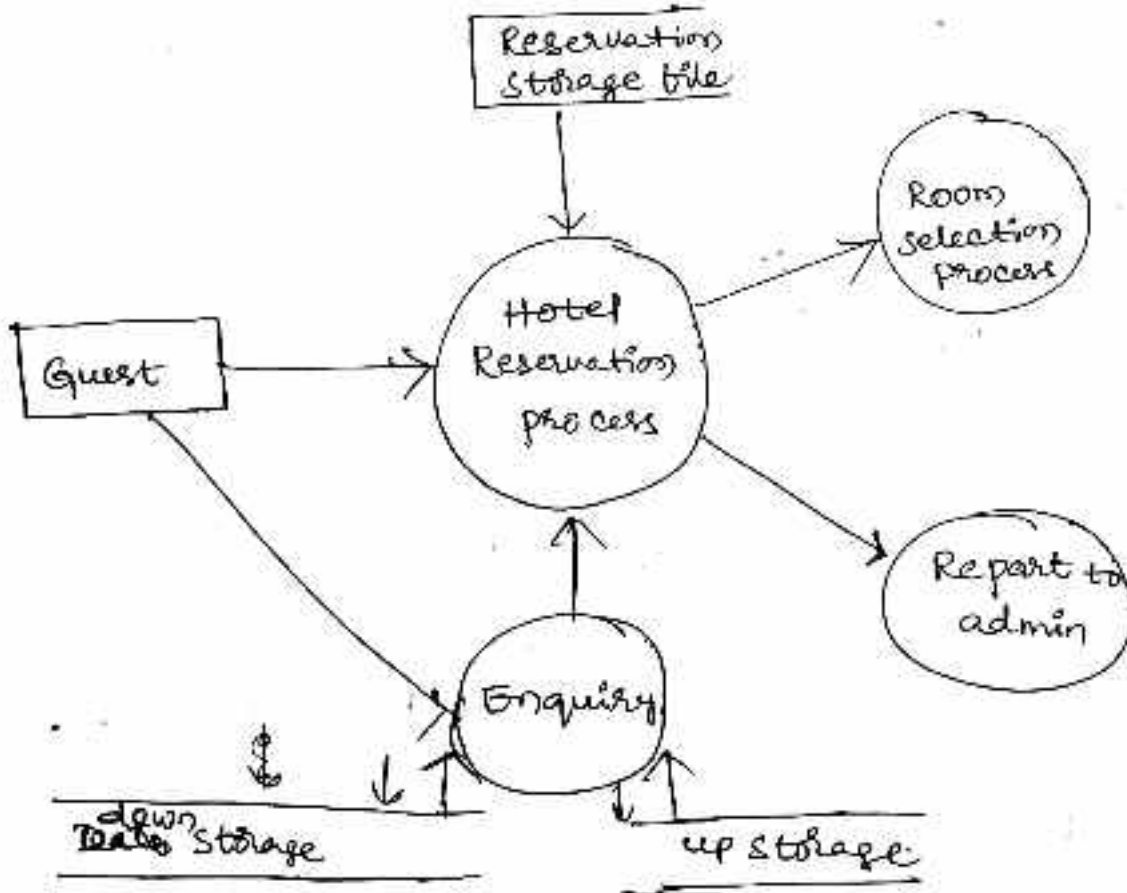


Fig: Level 1 DFD diagram.

## Level 2 DFD

It goes one step deeper into parts of Level 1.

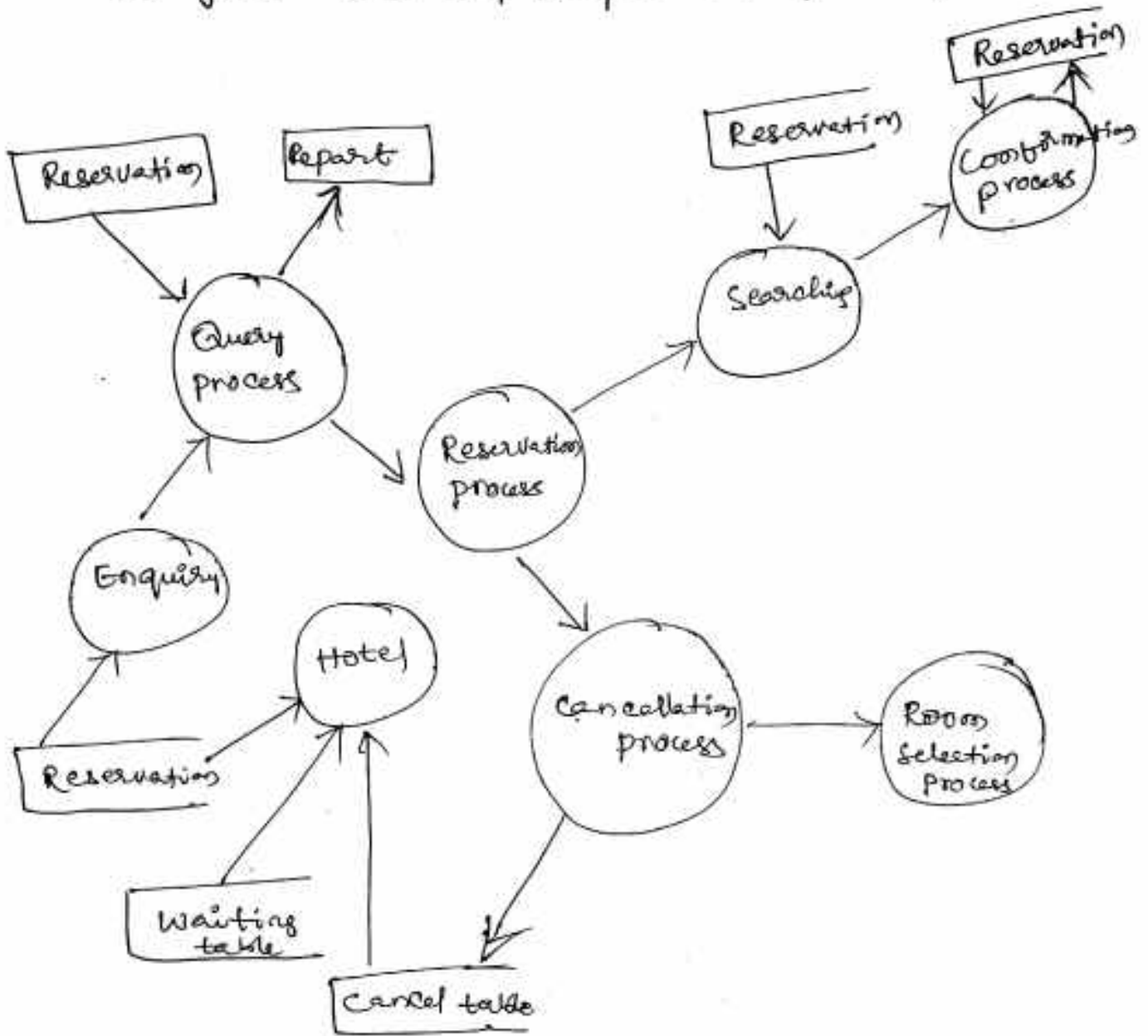


Fig: Level 2 DFD for Hotel reservation system.

## Data dictionary

The data about data is called "Data dictionary".

Ex: Bill pay



STRUCTURED DESIGN:

- Structural design is one of the most widely used function oriented design methodology which follows mainly top down design strategy.
- The basic approach of structured design is to transform the data flow diagrams of structural analysis into structure charts.

Structure chart:

- The structure chart is a graphical representation of procedural programs in the structured design methodology. It represents the modules of a system in a hierarchical fashion.
- The modules or programs in a structure chart are specified by a rectangle with their names. The module can either be a superordinate module or subordinate module.
- The calling module is known as "superordinate module".
- The called module is known as "subordinate module".

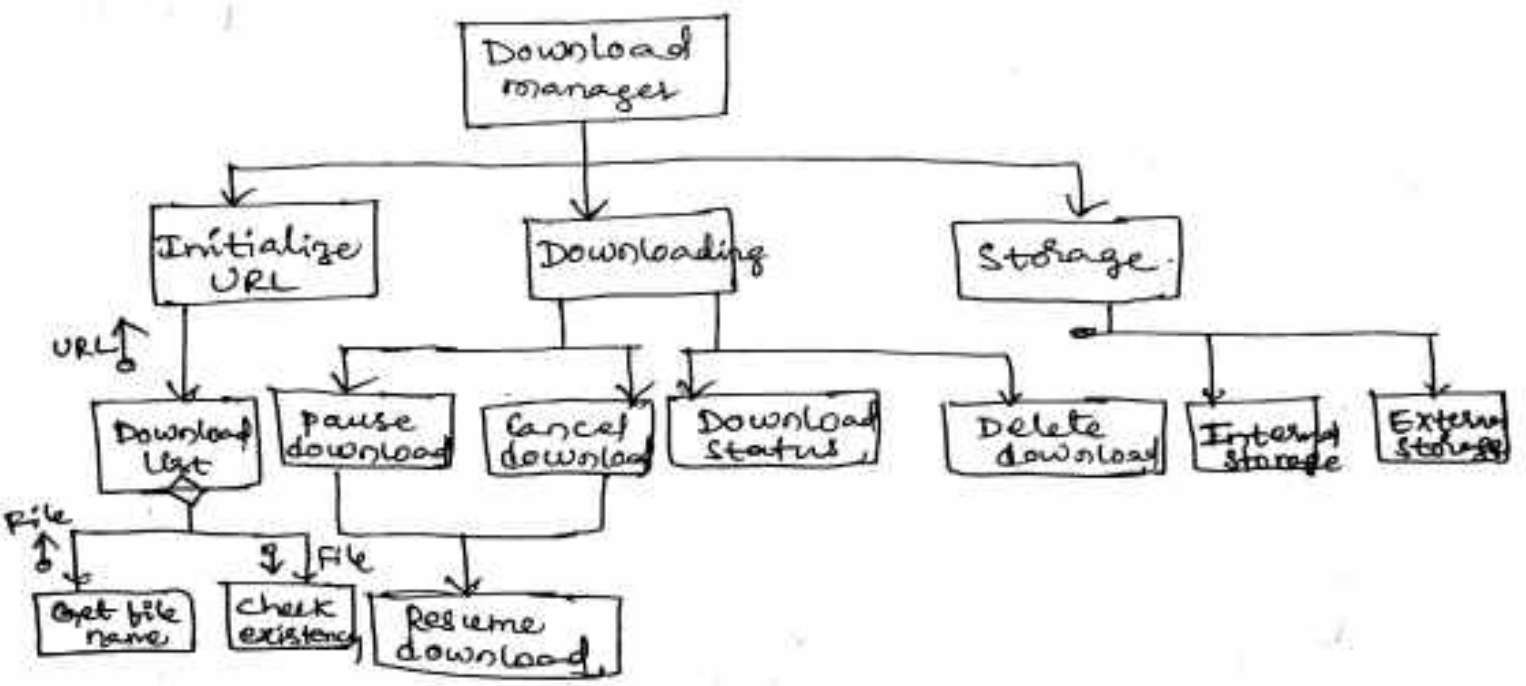


fig: structure chart for download manager.

## DETAILED DESIGN:

- It describes the internal details of the modules and interfaces.
- The basis of detailed design is the blueprint, for which algorithmic details and data structures are provided.
- There are various detailed design tools such as algorithm, pseudo code, data structure diagrams, structural flowcharts.

### → Algorithmic design

- An algorithm is a step by step process of problem solving.
- An algorithm has two parts, problem definition and design of algorithm.
- Problem definition states the problem to be solved by the algorithm. It describes the major data structures, input and output of the algorithm.
- Design of algorithm is the sequence of steps performed to solve the problem.

## DESIGN REVIEW:

- The outcome of design should be verified because it is in the subsequent stages of software development.
- A good design can easily be modified and maintained in the future.
- There are two most common approaches used for design verification and these are design reviews and design inspections.
- Design reviews are conducted at the end of software design.
- Design inspections are performed by expert team members.



• OVERVIEW OF OBJECT ORIENTED DESIGN:

→ Object-oriented analysis and design (OOAD) is a combined approach to problem analysis and designing the solution.

→ It is also called as "modelling".

OOA (Object Oriented Analysis)

The Object Oriented Analysis focus on modelling of the application domain

OOD (Object Oriented Design)

The Object Oriented design focus on modelling of the solution domain.

Object Oriented concepts

Object is an important element of an OOD.

The Object is an real world entity, for example files, employees, trains, television, cash, etc.

class is an a set of objects that share a common structure. class is an real world entity, for example person, place, thing. A class has common attributes and operations. Attributes are the properties of a class.

Ex:

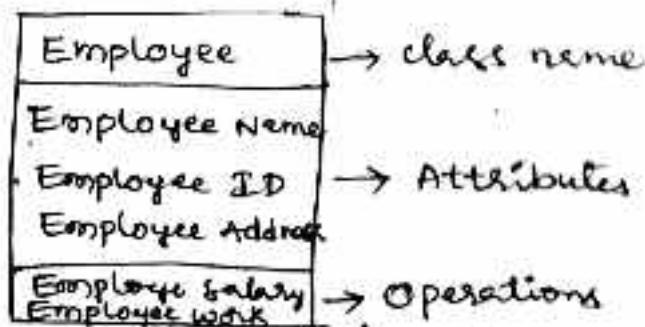


fig: class

# UML (Unified modelling language)

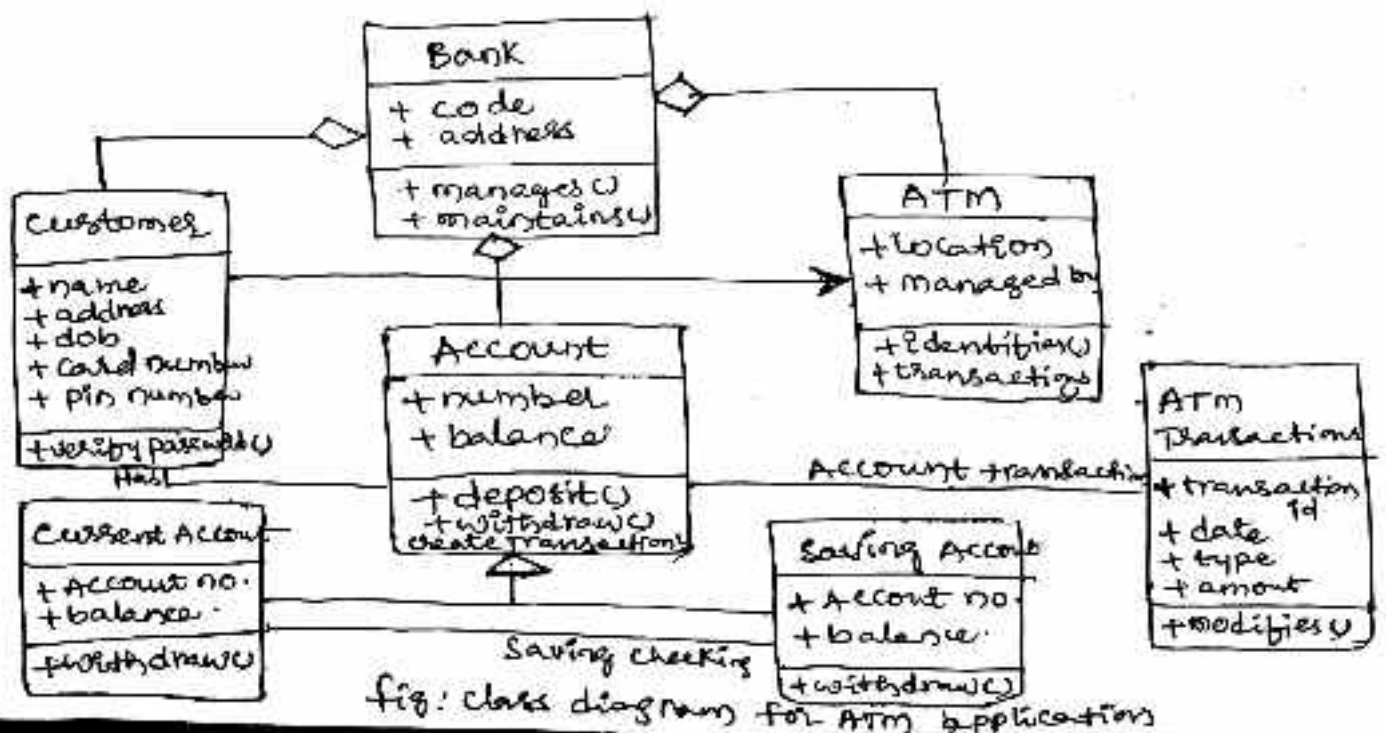
- Unified modelling language is a graphical language.
- It is proposed by "Grady Booch".

## UML diagrams



1. class diagrams
2. Usecase diagram
3. Sequence diagram
4. statechart diagram.

## class diagram

- The class diagram describes the structural description of the system.
- The class diagram consists of various classes, their attributes and operations are identified.
- class diagram is prepared by establishing relationships among classes.
- class relationships are using aggregation, generalizations, associations and dependency relationship.



## 2. Use case diagram

- The use case diagram represents the overall scenario of the system.
- It consists of Actors and Usecases and Associations.
- An Actor is an entity which interact with the system.
- It is denoted by the symbol & notation is  Actor.
- The Usecases represents the behavioural of the system activities & functions.
- It is indicated by the symbol is  usecase.
- Association represents the link between Actor and Usecases. It is indicated by '—'.

For example Usecase diagram for ATM application.

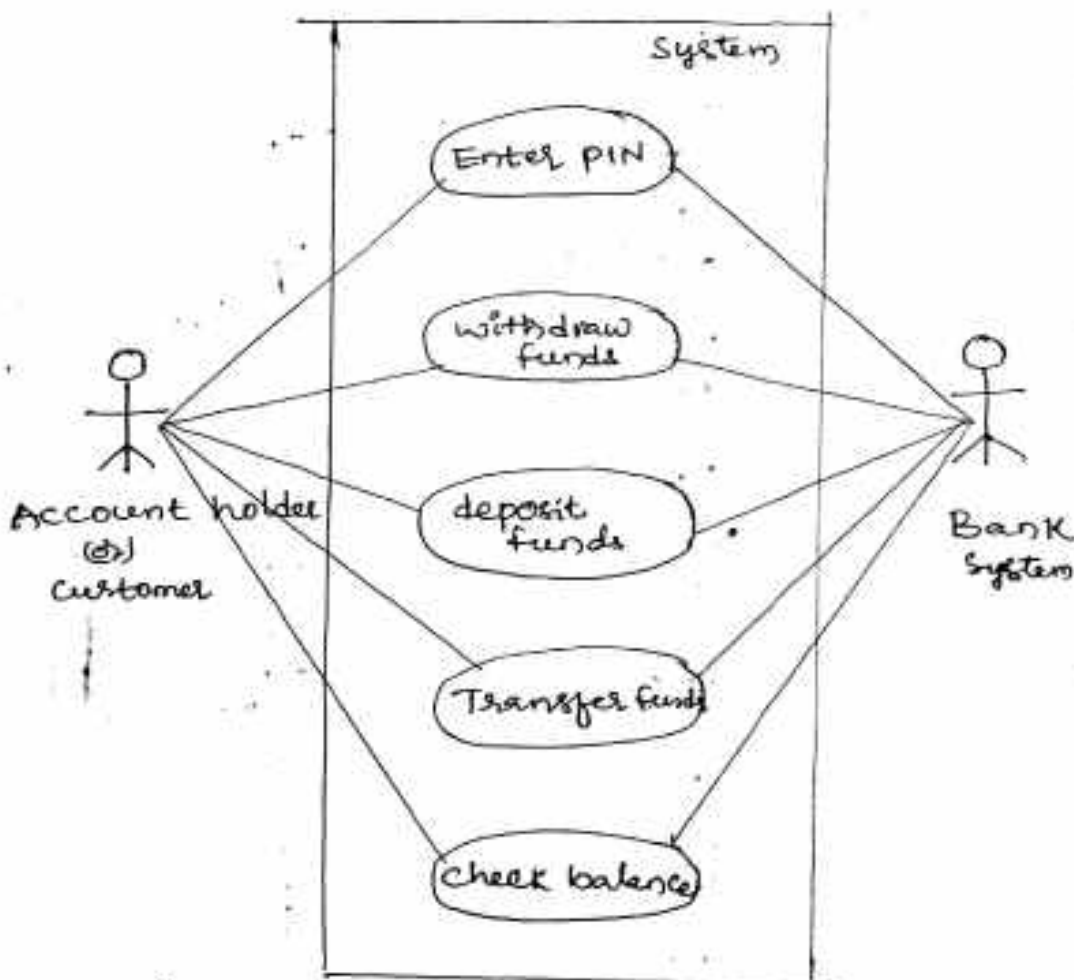
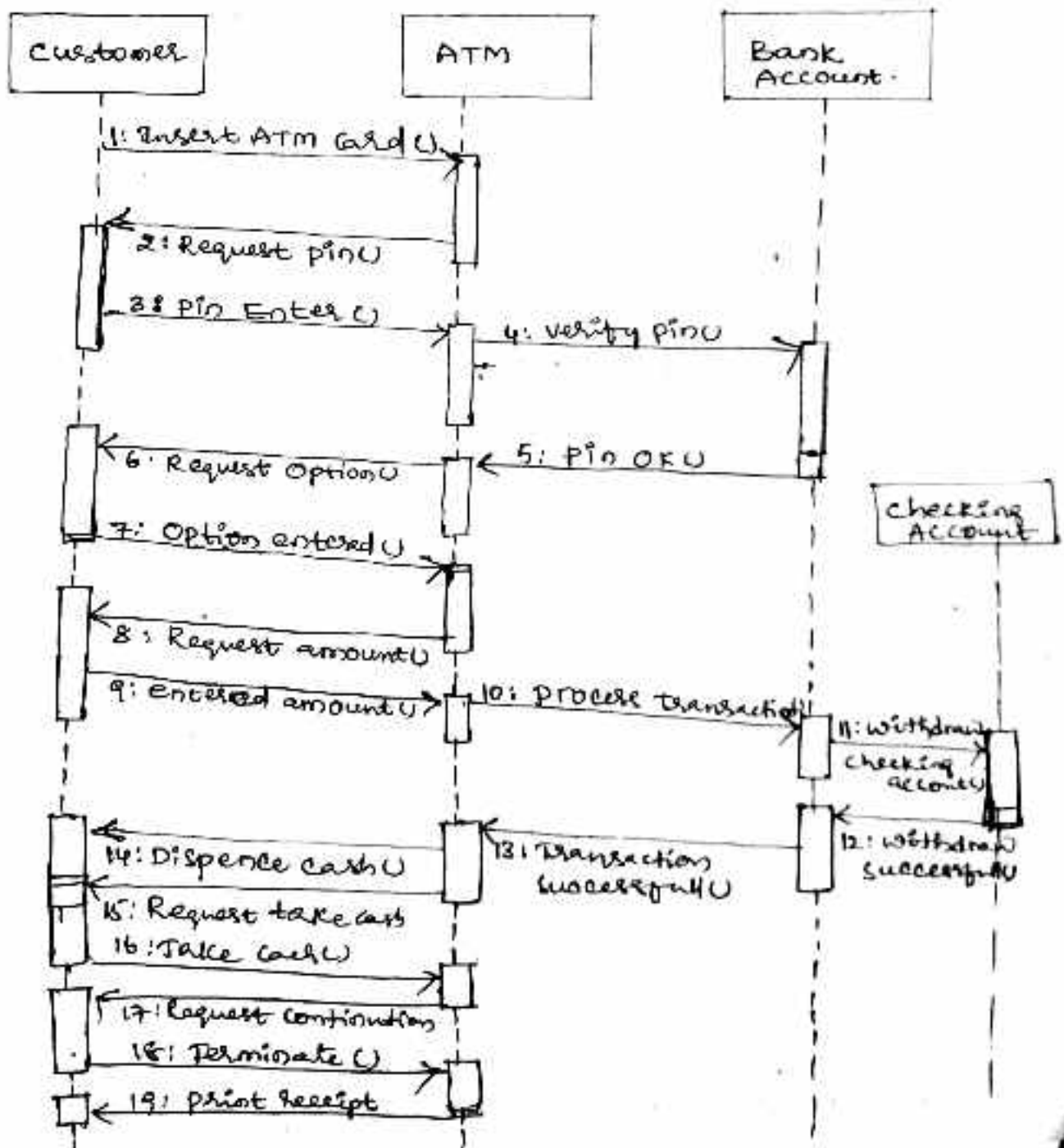


fig: Use case diagram for ATM application.

### ③ Sequence diagram :

- A sequence diagram shows the interaction among objects by passing messages in a time based sequence.
- It describes what task is to be carried out next.
- It consists of entities and vertical dimension has vertical dashed life line.

Example ~~for~~ Sequence diagram for ATM application.



④ State chart diagram

→ The state chart diagrams are used to ~~graphically~~ graphically represent the states of the system.

→ The state chart diagram consists of initial states, final states and simple states.

→ Initial state: It indicates the starting state of the system. It is denoted by the symbol is filled circle, ●

Final state: It indicates the end of the activity. It is denoted by the symbol is surrounding the small filled circle, ⊙

Example: state chart diagram for ATM application.

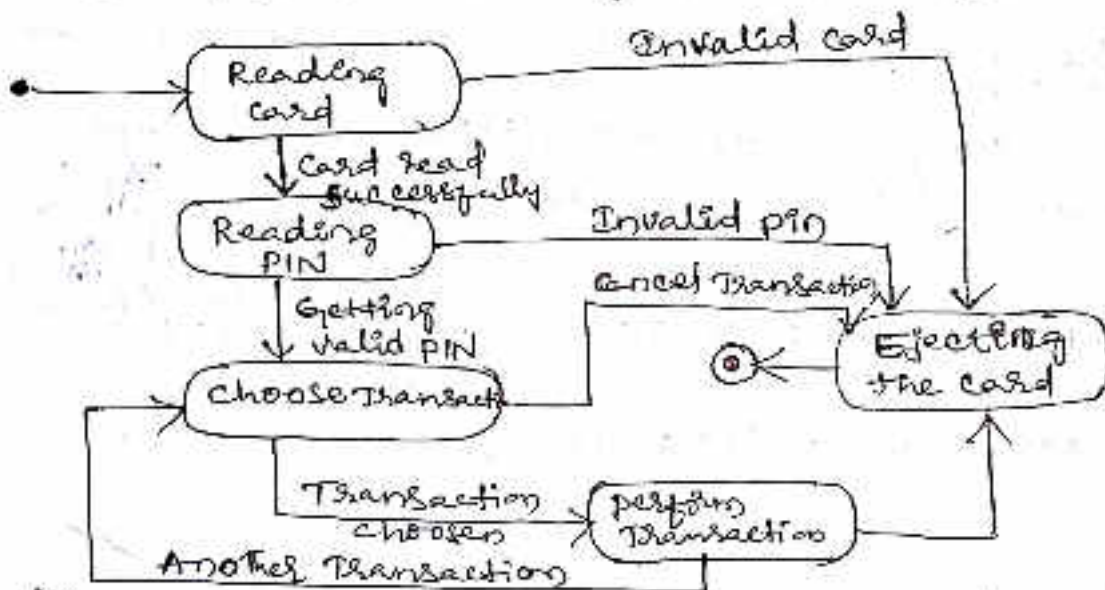


Fig: state chart diagram for ATM withdraw application.

Working process ATM

The customer first insert the ATM card into the system. The system starts the ATM session for reading the card. First of all system authenticates customer by reading the card and pin number entered by the customer. If the authentic customer is present then system displays the different service options. The customer selects withdrawal cash option. He then enters required amount. The system performs the transaction. It effects the system dispenses the requested amount to the customer. The system records transactions log entry for withdrawal. The customer take the amount, and close the session.

## USER INTERFACE DESIGN:

→ The User interface is an interactive system that helps User to operate and control the hardware and software. shown in figure.

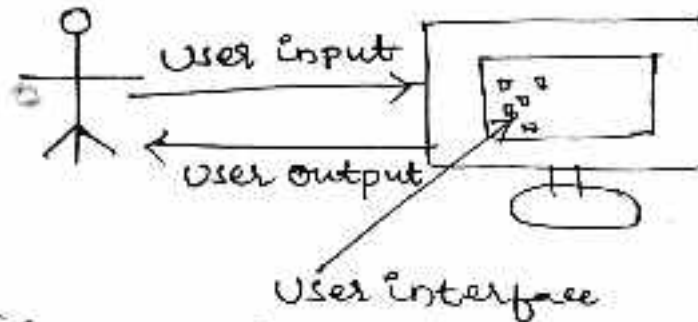


Fig: User interaction with computer.

→ The user interact <sup>with</sup> different devices, these are mobile phones, computers, watches, musical players, etc. lift elevators etc.

→ The Human computer interaction (HCI), which explains how humans interacts with computer systems.



Fig: Lift system.

## CHARACTERISTICS OF A GOOD USER INTERFACE

User interaction with computer system is easier natural way.

1. Usability
2. Feedback
3. Responsiveness
4. User guidance
5. Control
6. Robustness

① Usability.

Usability means ease of use the system or product.

A Usable product can be characterized by the following quality attributes.

- ① Effectiveness → The user successfully achieves their goals.
- ② Efficiency → The user can complete the task quickly (fastly).
- ③ User Satisfaction → User feel pleasant in using software.
- ④ Easy to learn → Easy to operate the system ex: lift elevator.
- ⑤ Error tolerant → The user interact with the system to recover ~~some~~ errors.
- ⑥ flexibility → A good user interface provides multiple ways of doing tasks.
- ⑦ Simplicity → A simple interface is easy to learn.
- ⑧ portability → A system is used in different environments.

② Feedback

Feedback is very important in user interface design. feedback explains what is happening currently, what is going to happen next, what happened after submission.

For example software installation, online student registration forms.

User can interact with buttons, links, navigation, forms and programs.

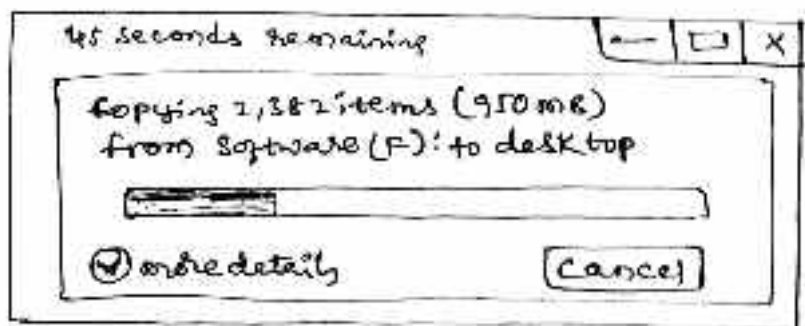


fig: User feedback through progress bar.

### ③ Responsiveness

A good User interface responds quickly to the user. Fast response improve the user experience. If it responds slowly or waiting, then user may feel frustrated. If it responds quickly.

### ④ User Guidance

While working with the system interface, sometimes user doesn't understand the process of execution, commands, that facilities are available with the systems.

Ex: Use guidance such as help systems, on-line manuals, comments, notes etc.

Example, start mail merge is described along with the help option as shown in fig.

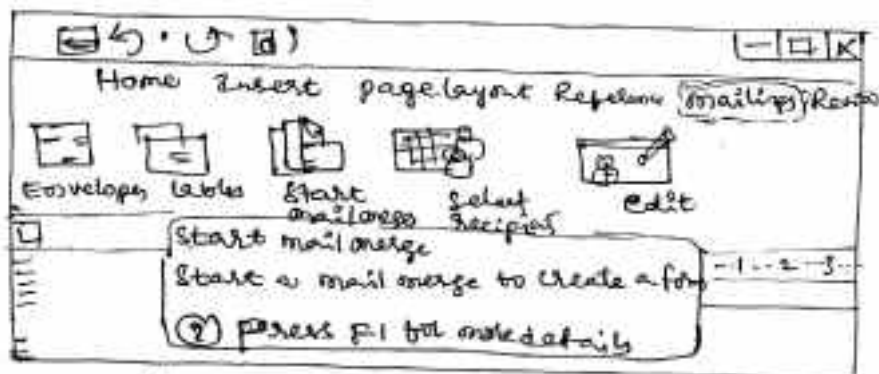


fig: Help menu to start mail merge.

### ⑤ Control

A good user interface covers mostly the constraints and limitations desired by the user.

### ⑥ Robustness

A good user interface supports different categories of known and unknown users. There should be recovery tips at any state of the system.



## BASIC CONCEPTS :

User interface design is a key aspect to the successful use of a software product. For example, the web browsers, interactive games, interactive applications <sup>are</sup> widely used by the users.

The basic concepts are

1. Interface design elements
2. Visual design
3. Help System

### 1. Interface design elements

Interface design is a way of user interaction.

User ~~can~~ interacts different elements these are

- (a) Input design controls - control buttons, text fields, check boxes.
- (b) Navigation - menu items, search fields, page numbers
- (c) Feedback - progress bar, notifications, message boxes
- (d) Output designs - screenshots

### 2. Visual design :

Visual design is the use of symbols, color, graphics to represent information to the user.

Ex: Animation, 2D, and 3D animations (graphics).

### 3. Help System :

User need to know how to work with the application. The user guidance affects the usability of the application. Sometimes user type wrong input that creates problem later.

Ex: password must an upper case letter, a lowercase letter, a special character, a number, a combination of all these but not less than 8 symbols. Then should proper feedback or message. Ex: you are registered successfully. Retype password.

## TYPES OF USER INTERFACES :

User interaction with computer has now been trending in various ways.

Command prompt based interfaces are not used nowadays as there exists many interfacing modes and languages that provides more comfortable user interactions. Some other modes are graphical, voice, touch based, and so on.

The following categories of user interactions.

1. Command language interface.
2. Graphical User interface (GUI)
3. Menu driven interface.
4. Form fill in interface.
- 5.

### 1. Command language interface :

A Command language interface allows the user to control the application with a series of keystrokes, commands, & phrases. Most commonly, command line interface is used in DOS computer. User interaction is possible through running commands correctly.

User types commands on command prompt, indicating the current position on command line.

Ex: DOS Operating system shown in figure.

```
c:\mark Administrator Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c)
C:\Users\admin> cd..
C:\Users> chdir
C:\Users
C:\Users> date
The current date is: 12-09-2013
C:\Users>
```

Fig: Command prompt.

## ② Graphical User Interface (GUI)

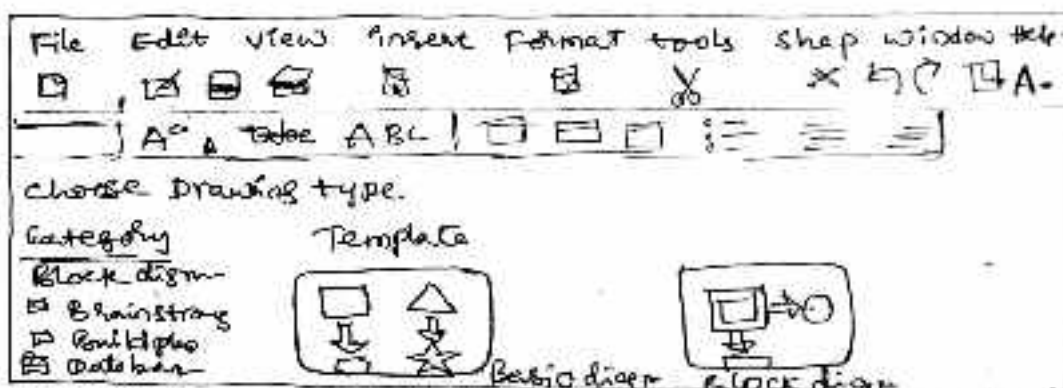
A Graphical User interface (GUI) is, nowadays, the most commonly used user interface.

The interface is very user friendly for people to interact with the applications. It makes pictures, graphics, and icons.

Windows operating system is an example of GUI based software. User interacts with the computer through the mouse and the keyboard.

\* GUI contains elements such as windows, icons, menus, pointing devices, and graphics.

Ex: GUI windows screen is shown in figure.



GUI based windows screen.

## ③ Menu driven interface:

A menu-driven interface provides the user with a list of available selections through screens & menus.

Ex: mouse, touchscreen, using keyboard. The user selects options.

ATM machine shows menu options.

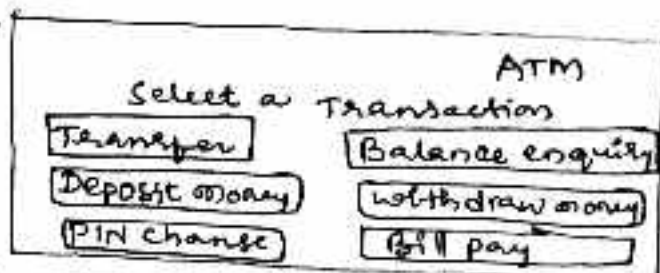


Fig: menu driven interface for ATM machine.

#### 4. Form fill in Interface:

Form fill in interface consists of onscreen forms displaying fields containing data items that need to be communicated to the user.

The screen is designed as a form in which user can enter data in the pre-defined form fields. The control key tab or arrow key can be used to switch between the fields.

For example: Online Student Registration.

"A student will be typical user of this system. The user has to fill up an online registration form. In this form he has to submit the student information and then he select the courses which he/she wants to adopt."

online student registration form

Student Information

Name

Academic year

Student Id

status

Address

Telephone Number

Course Selection

Department  course

fig. online student registration form.

FUNDAMENTALS OF COMPONENT-BASED GUI DEVELOPMENT

Before going to discussing component based GUI development, let us discuss some basic elements in GUI.

GUI environment which includes.

1. Menus and Navigation
2. Windows
3. Controls
4. GUI programming.

1. Menus and Navigation

The hierarchical listing of commands or choices available to user is known as menu.

A menu bar contains group of commands in the application window. Each of these commands is the component available with GUI programming languages.

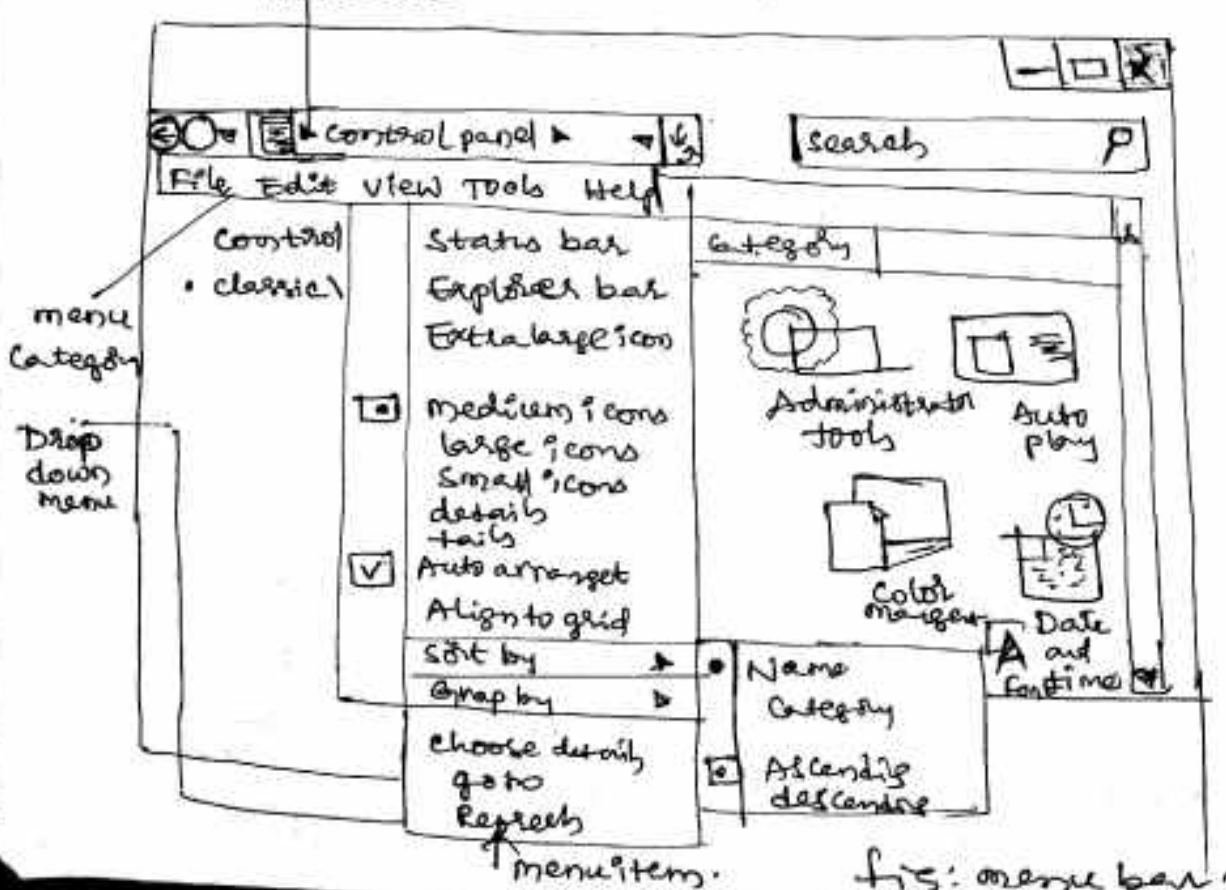
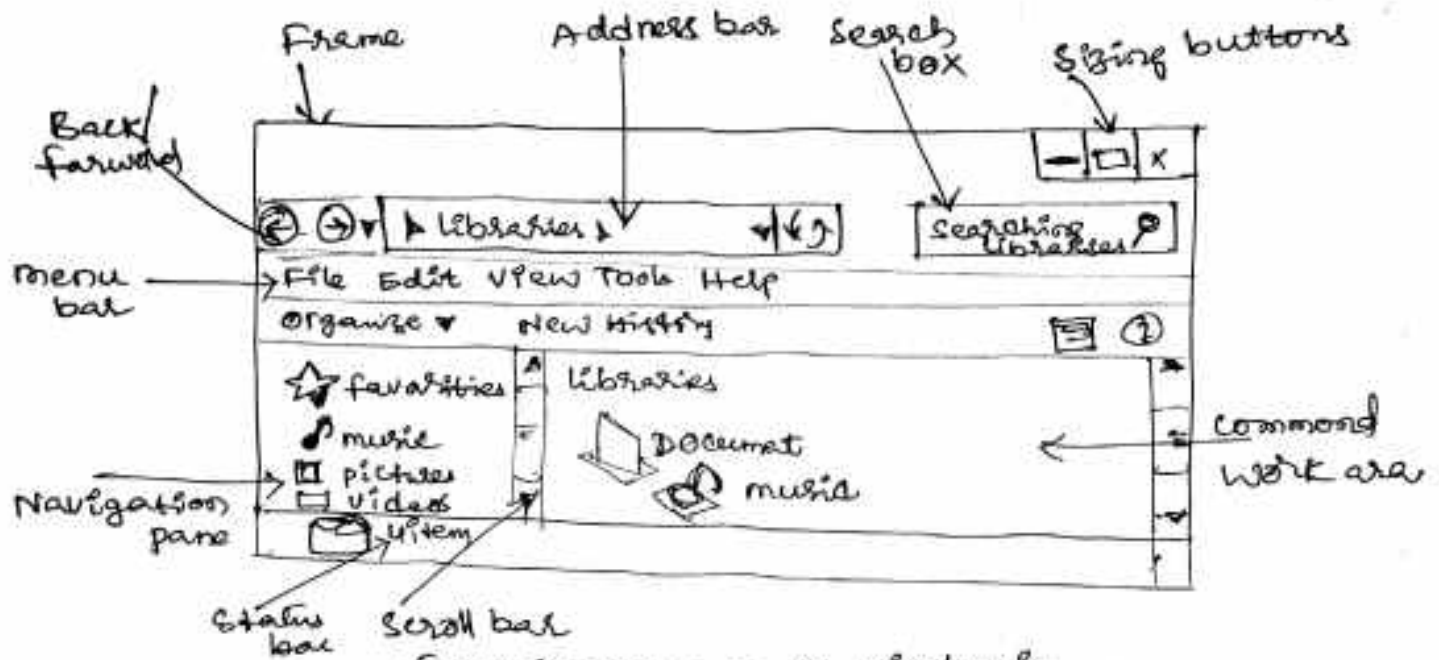


fig: menu bar

## 2. Windows

A window is an area in rectangular shape. It is viewing area that displays contents of applications. Windows can be minimized, resized or maximized to size of ~~part~~ of screen.

For example, one window can be used for MS Word and other window can be used for control panel setting.



## 3. controls

It is also called widgets are user interface design. Different controls are

- (a) label control
- (b) Dialogue box
- (c) Text Box
- (d) Buttons,
- (e) Radio button
- (f) check box
- (g) list box,
- (h) slider
- (i) Combo box
- (j) Dropdown list
- (k) list view
- (l) Data grid
- (m) scroll bar
- (n) picture box.

## 4. GUI programming

→ GUI is an event-driven programming, which provides GUI components. GUI components such as scroll bars, text, input boxes, menus etc.

→ A GUI program creates icons and widgets that are displayed to a user and organize them inside the a screen windows. → GUI programming languages like JAVA is using classes that are developed. GUI components

## A USER INTERFACE DESIGN METHODOLOGY:

The process of GUI design and implementation is a like SDLC. The pictorial representation of the GUI design methodology is shown in figure.

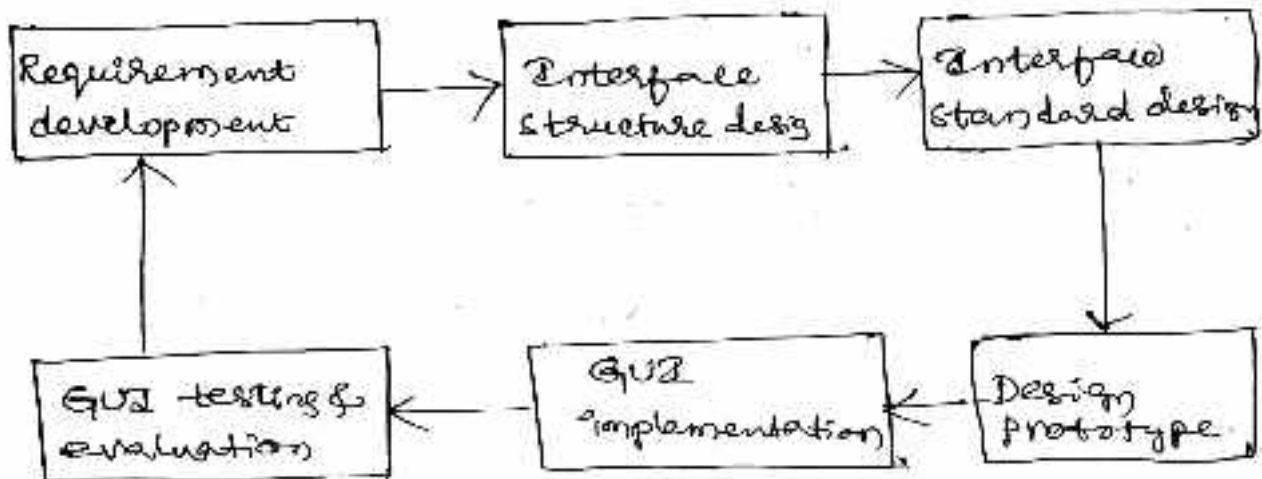


fig: GUI design methodology.

The process starts with requirement identification to GUI evaluation through user interaction phases of methodology.

GUI design methodology consists of the following phases.

1. Requirement development → user needs, challenges, problems identified.
2. Interface structure design → structure charts
3. Interface standard design → forms, reports, menus, screens
4. Design prototype → Rough layout
5. GUI implementation → Rough layout to final layout
6. GUI testing & evaluation → Usability, Acceptance, consistency evaluated.

## Important Questions

- ① what is meant by DFD? ~~And draw flow on~~  
~~example~~
- ✓ ② Explain briefly DFD (Data flow diagram) with neat sketch?
- ✓ ③ Explain briefly OOD (Object oriented design)?
- ④ Explain briefly structured Analysis and structured design (SA&SD)?
- ⑤ What are the good characteristics of User interface?
- ✓ ⑥ What are the types of User interfaces?
- ⑦ Explain briefly fundamentals of component based GUI development process?

