Chapter 4

System Design

System design is the process of defining the architecture, components, modules and interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

Actual Design of the problem is described in this chapter as under-

4.1 DFD - Data Flow Diagrams

It is a graphical representation of the "flow" of data through an information system, modelling its process aspects.

A Data Flow Diagram (DFD) is often used as preliminary step to create an overview of the system without going into great details, which can later be elaborated.

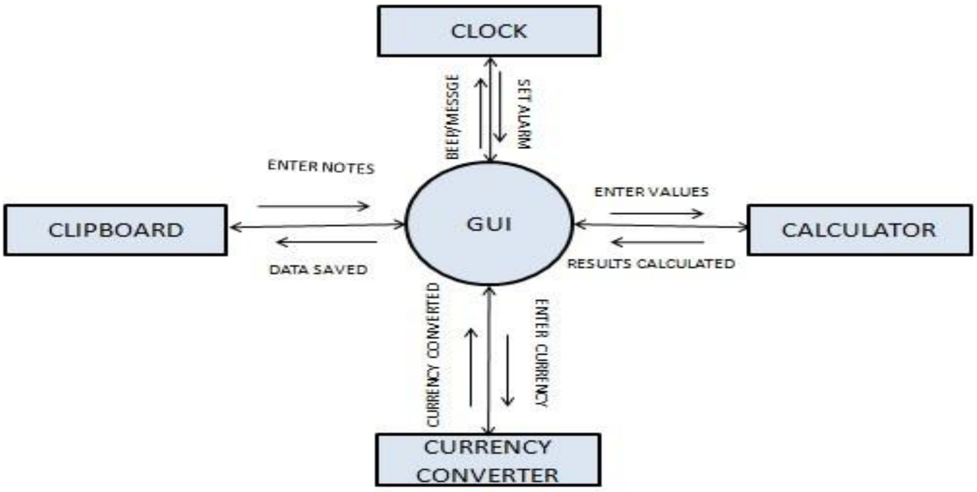

Fig. 4.1: Data Flow Diagram

The data flow diagram represented in figure 4.1 clearly shows the flow of data. When some input is given to the clock (i.e. alarm is set), the data is entered into the GUI and correspondingly an output is generated for the given input by the clock (alarm beeps or pops up a message). The same mechanism happens in a similar fashion for clipboard, calculator and currency convertor. In case of the clipboard, as and when the user copies, pastes, cuts or types characters on the clipboard, the same is reflected on the GUI. In case of calculator, the user gives a mathematical expression for the calculator via GUI and correspondingly, an output is generated in form of the result to the mathematical expression. In case of currency convertor, the user gives a denomination in one currency to be converted in some other foreign currency equivalent via GUI and the output for the same is provided.

4.2 Unified Modeling Language (UML) – Diagrams

UML is a standardized modelling language enabling developers to specify, visualize, construct and document artefacts of a software system. Thus, UML make these artefacts sealable, secure and robust in execution.

4.2.1 - Class diagram

It is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods) and the relationships among objects.

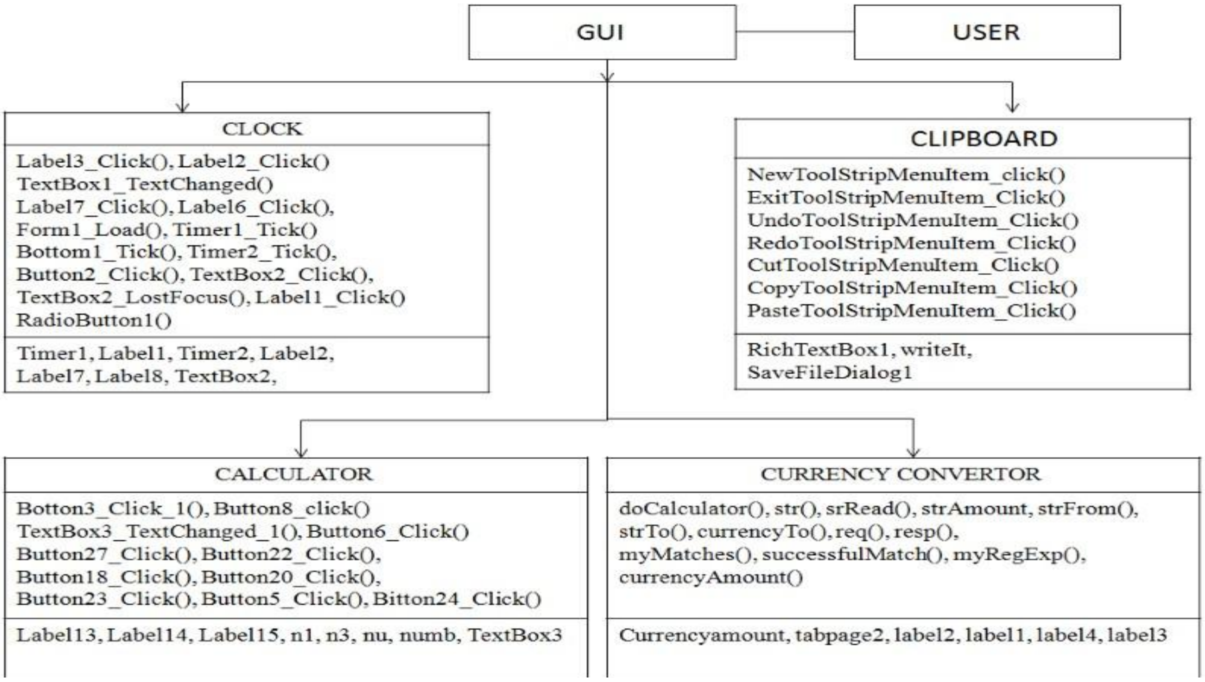
GUI
USER
CLOCK
Label3_Click(), Label2_Click()
TextBox1_TextChanged()
Label7_Click(), Label6_Click(),
Form1_Load(), Timer1_Tick()
Bottom1_Tick(), Timer2_Tick(),
Button2_Click(), TextBox2_Click(),
TextBox2_LostFocus(), Label1_Click()
RadioButton1()
Timer1, Label1, Timer2, Label2,
Label7, Label8, TextBox2,
CLIPBOARD
NewToolStripMenuItem_click()
ExitToolStripMenuItem_Click()
UndoToolStripMenuItem_Click()
RedoToolStripMenuItem_Click()
CutToolStripMenuItem_Click()
CopyToolStripMenuItem_Click()
PasteToolStripMenuItem_Click()
RichTextBox1, writeIt,
SaveFileDialog1
CALCULATOR
Botton3_Click_1(), Button8_click()
TextBox3_TextChanged_1(), Button6_Click()
Button27_Click(), Button22_Click(),
Button18_Click(), Button20_Click(),
Button23_Click(), Button5_Click(), Bitton24_Click()
Label13, Label14, Label15, n1, n3, nu, numb, TextBox3
CURRENCY CONVERTOR
doCalculator(), str(), srRead(), strAmount, strFrom(),
strTo(), currencyTo(), req(), resp(),
myMatches(), successfulMatch(), myRegExp(),
currencyAmount()
Currencyamount, tabpage2, label2, label1, label4, label3

Fig. 4.2.1: Class diagram

As stated above a class diagram shows classes, data members and member functions used in the system. The GUI developed; "Personalized GUI" has mainly four classes: Clock, Clipboard, Calculator and Currency Convertor. These four classes are related to GUI which holds the scope and acts as one unit. The classes listed above have further its own member functions and data members as shown inn figure 4.2.1.

4.2.2 - Object diagram

Object Diagram is a graph of instance, including objects and data value. A static object diagrams an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time.

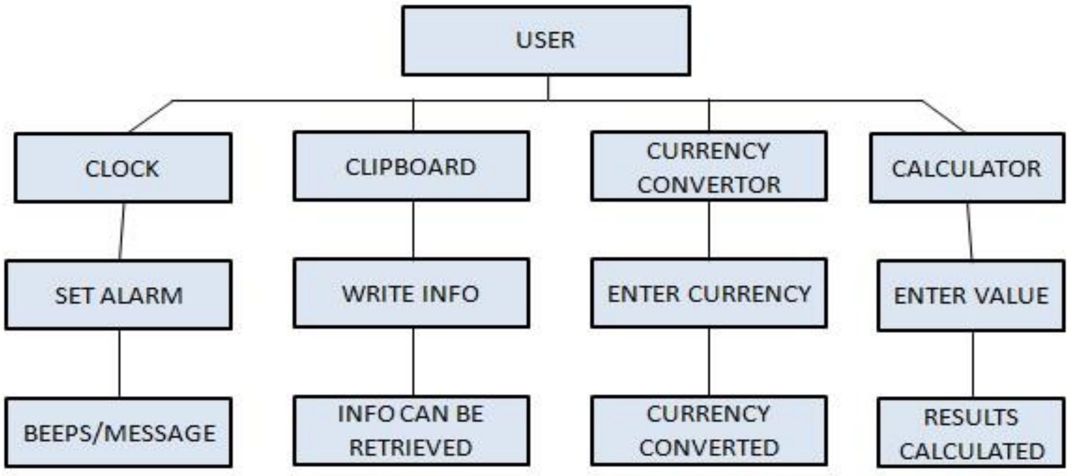

Fig. 4.2.2: Object Diagram

A UML object diagram represents a specific instance of a class diagram at certain moment in time. When represented graphically, we see many similarities to the class diagram. An object diagram focuses on the attributes of a set of objects and how these objects relate to each other. The class Clock, Clipboard, Calculator and Currency Convertor have their own objects defined for specific purposes. The figure 4.2.2 illustrates the object diagram.

4.2.3 - State diagram

State Diagram describes the behaviour of the systems. It requires that the system described is composed of a finite number of states.

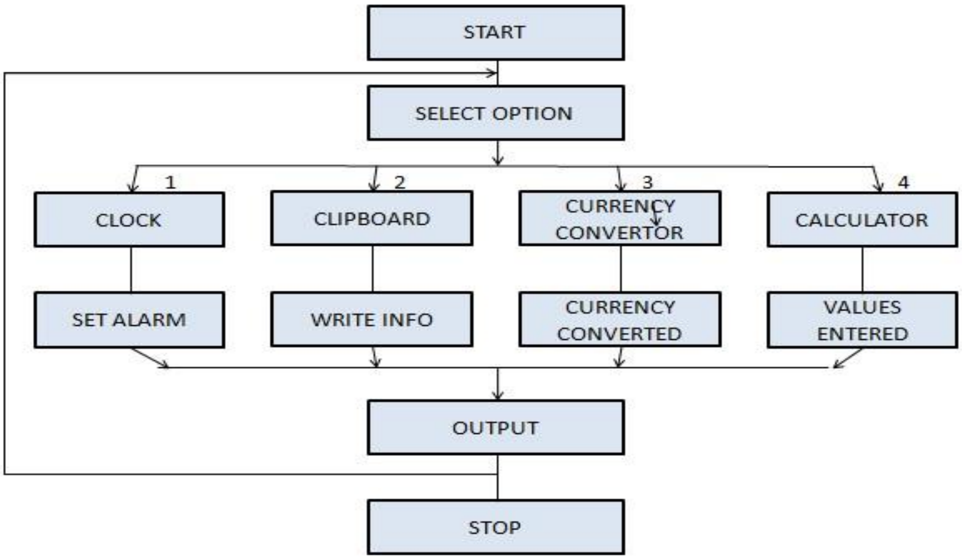

Fig. 4.2.3: State Diagram

A state diagram shows the behaviour of classes in response to external stimuli. Specifically, a state diagram describes the behaviour of a single object in response to a series of events in a system.

The figure 4.2.3 shows a state diagram when the system is started and personalized GUI is selected, it further gives four options. On selection on either of the option, it further takes to option and on giving specific input, an output is generated. The flow goes to GUI for next input selection, unless existed.

4.2.4 - Activity diagram

Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes.

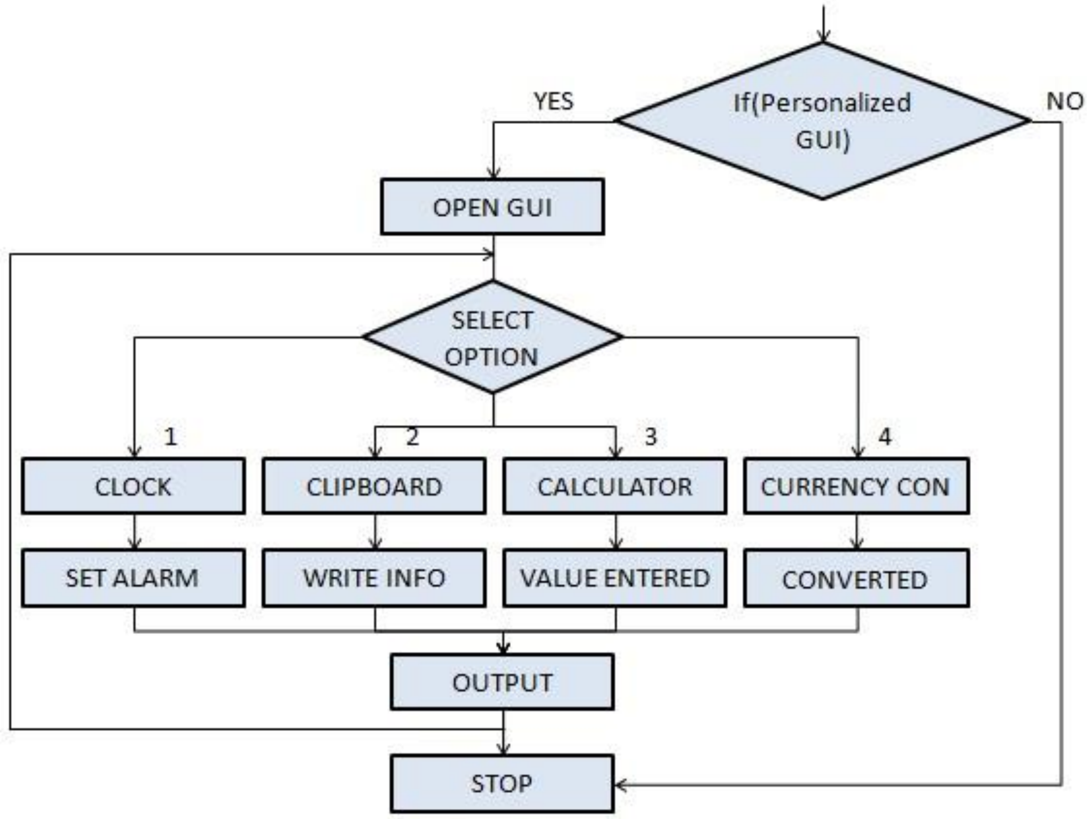

Fig. 4.2.4: Activity Diagram

Activity diagram is UML behaviour diagram which shows flow of control with emphasis on sequence and condition of the flow. Fig. 4.2.4 shows the activity diagram.

On selection of personalized GUI, GUI opens up and gives further four options to select. Input is sought by the four options and a particular output is generated. Once the output is generated, it is ready for next input entry and option selection. If the user does not want to carry on any more, stop is to be selected. If the personalized GUI is not selected, scope comes to exit.

4.2.5 - Sequence diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows how objects interactions arranged in time sequence. Sequence diagrams are sometimes referred as event diagrams or event scenarios.

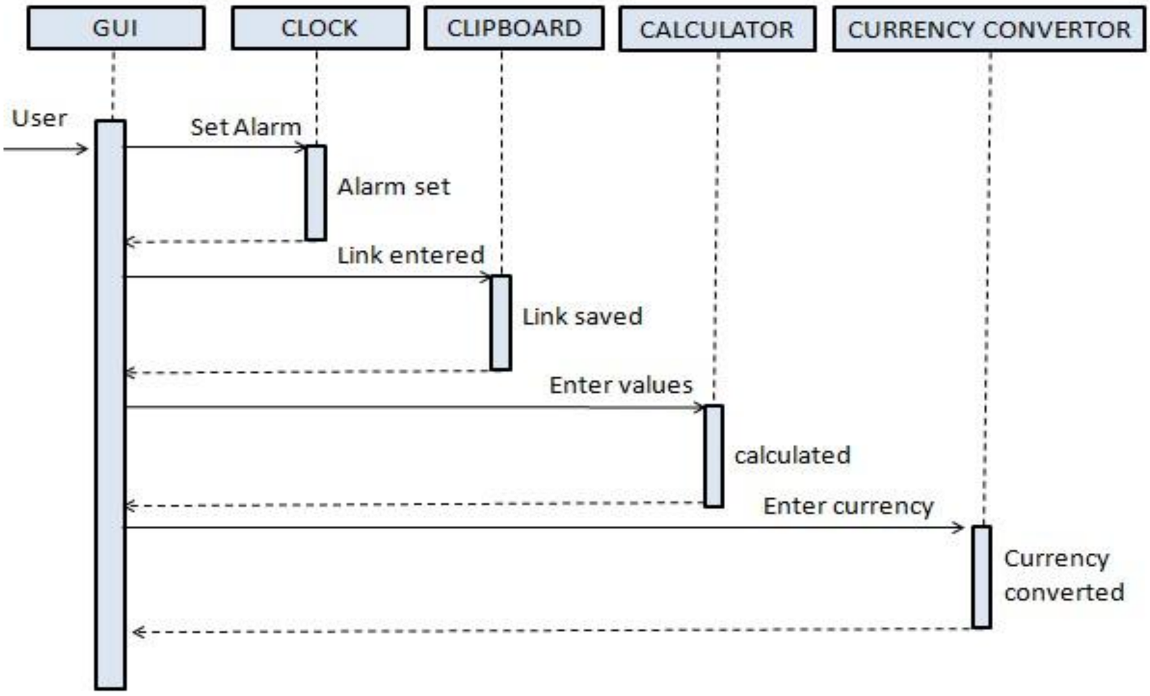

Fig. 4.2.5: Sequence Diagram

UML Sequence diagrams are used to show how objects interact in a given situation. An important characteristic in sequence diagram is that time passes from top to bottom.

The figure 4.2.5 shows the sequence of the personalized GUI. When a user opens GUI, and then selects any one among four options. Its work will be performed by taking input and output generated will be given to the user.

4.2.6 - Use case

A use case diagram is a graphical description of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify and organize system’s requirements.

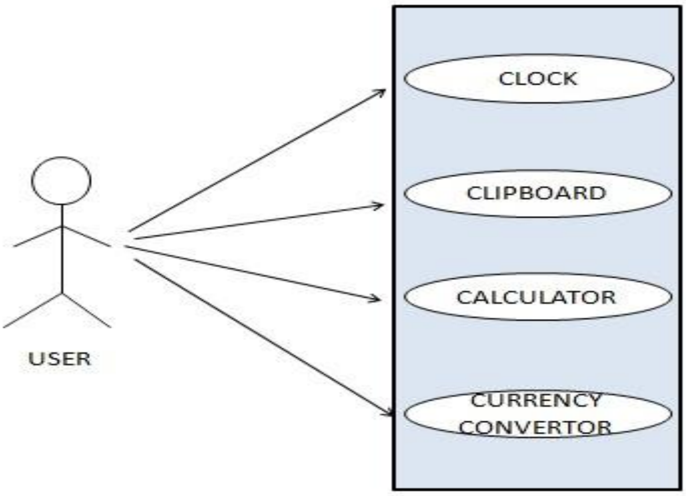

Fig. 4.2.6: Use case Diagram

Use case diagram is a behavioural UML diagram type and frequently used to analyse various systems. It enables us to visualize the different types of roles in a system and how those roles interact with the system. The use case diagram in the figure 4.2.6 shows personalized GUI's use case. The use case identified is clock, clipboard, calculator and currency convertor.

4.2.7 - Collaboration diagram

A collaboration diagram, also called a communication or interaction diagram, is an illustration of relationships and interactions among software interactions in the Unified Modelling Modeling (UML).

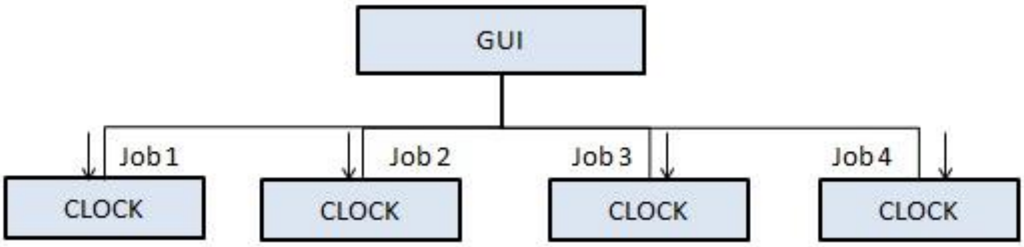

Fig. 4.2.7: Collaboration Diagram

UML collaboration diagram illustrates the relationship and interaction between software objects. The collaboration diagram Fig. 4.2.7, illustrates messages being sent between classes and objects.

4.2.8 - Component diagram

In a Unified Modeling Language, a component diagram depicts how components are wired together to form large components or software systems. They are used to illustrate structure of arbitrarily complex systems.

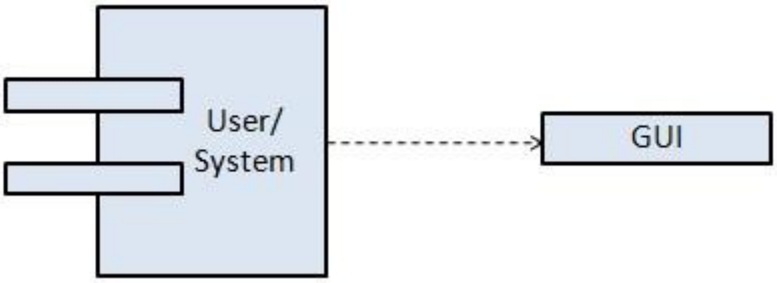

Fig. 4.2.8: Component Diagram

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Figure 4.2.8 shows the component diagram of the personalized GUI system. The components listed are user, system (i.e. O.S) and GUI.

4.2.9 - Deployment diagram

It is a structure diagram which shows architecture of the system as deployment of software artefacts to deployment targets.

The name deployment itself describes the purpose of the diagram. Deployment diagram are used for describing the hardware components where software components are deployed. Fig. 4.2.9 shows deployment diagram of the personalized GUI system. The components are user, O.S and GUI.

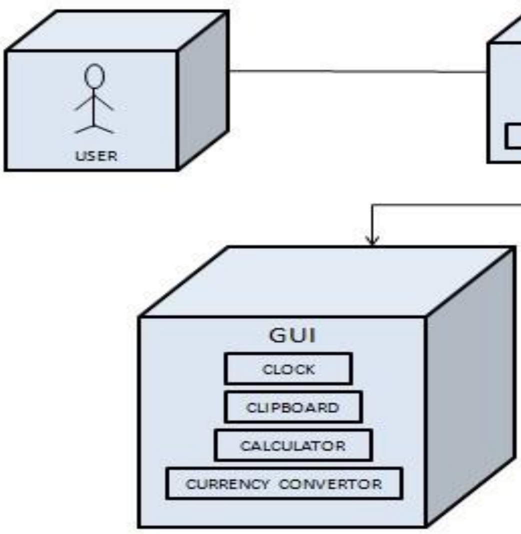

Fig. 4.2.9: Deployment Diagram