Final Year Project Title

Submitted By

**Student Name Roll #**

**Student Name Roll #**

**Student Name Roll #**

Supervised By

**Supervisor’s Name**

BS (Computer Science)

Session 2020 – 2024

Logo

Description automatically generated

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF GUJRAT

Table of Contents

[Second Deliverable For Structured Approach 2](#_Toc499027506)

[Chapter 2 System Requirement Specification 2](#_Toc499027507)

[5.1. Introduction 4](#_Toc499027508)

[5.2. Entity Relationship Diagram 4](#_Toc499027509)

[5.3. Data flow diagram (Functional Model) 5](#_Toc499027510)

[5.4. State Transition Diagram 10](#_Toc499027511)

[5.5. Architectural design 11](#_Toc499027512)

[5.6. Component Level Design 14](#_Toc499027513)

# Second Deliverable for Structured Approach

# Chapter 2 Design Document

## 2.1. Introduction

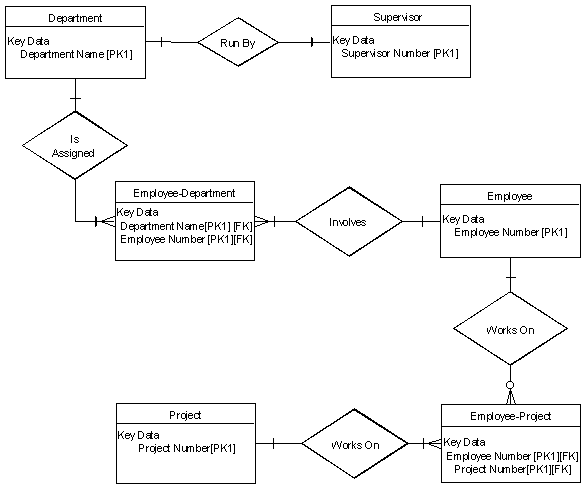
Analysis & Design Model for structured approach must contain following artifacts

1. Entity Relationship Diagram
2. Data Flow Diagram (Functional Model)
3. State Transition Diagram (Behavioral Model)
4. Architecture Design
5. Component Level Design

## 2.2. Entity Relationship Diagram

In the analysis model, Entity Relationship Diagram is used to understand the system under consideration with respect to entities involved and their relationships. Each entity is documented by extracted its attributes, cardinality, and modality.

**Example**



#### Cardinality

From the description of the problem, we see that

Each department has exactly one supervisor.

A supervisor is in charge of one and only one department.

Each department is assigned at least one employee.

Each employee works for at least one department.

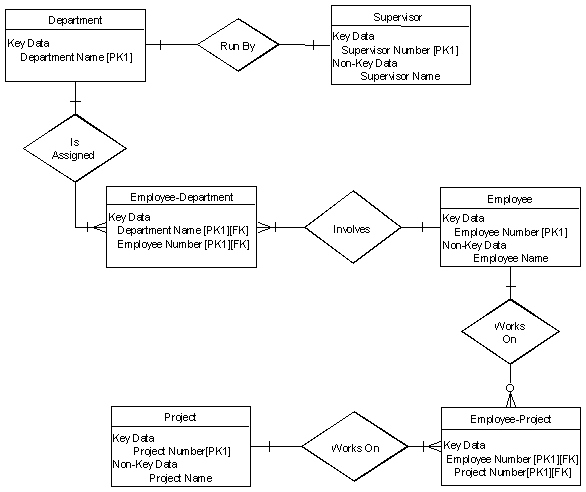
Each project has at least one employee working on it.

An employee is assigned to 0 or more projects.

#### Identify Attributes

The only attributes indicated are the names of the departments, projects, supervisors and employees, as well as the supervisor and employee NUMBER and a unique project number.

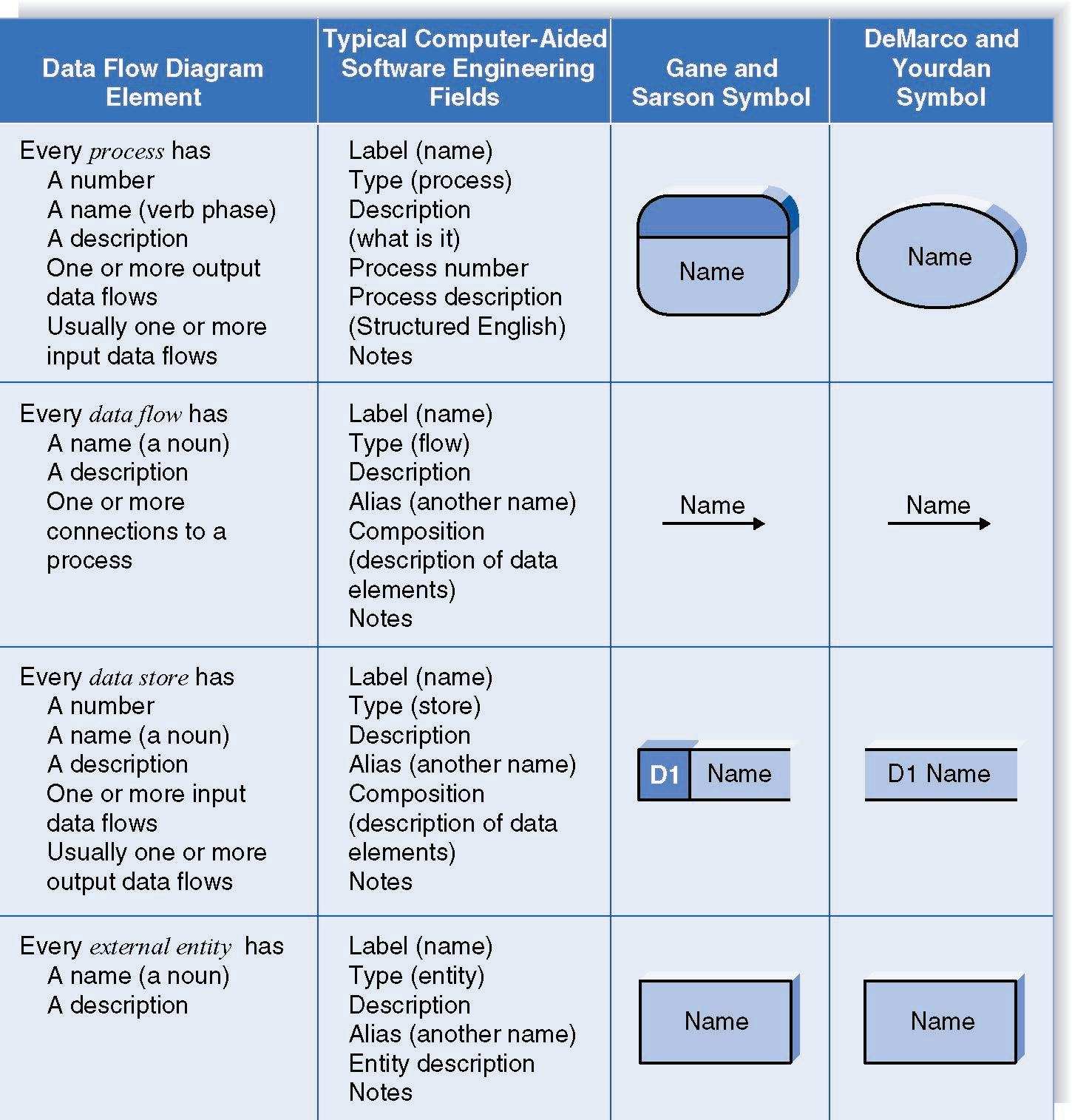
#### Fully Dressed ERD



## 2.3. Data flow diagram (Functional Model)

DFD is all about to identify the major processes in your system and develop Data Flow Diagram up to required level.

**DFD Constructs**



Context Level DFD

A context diagram shows the context into which the business process fits. It also shows the overall business process as just one process and shows all the outside entities that receive information from or contribute information to the system.

Level 1 Diagram

This diagram shows all the processes that comprise the overall system and how information moves from and to each process. Data stores are added to it.

Level 2 Diagram

This diagram shows all the processes that comprise a single process on the level 1 diagram and how information moves from and to each of these processes. It also shows in more detail the content of higher-level process. Level 2 diagrams may not be needed for all level 1 processes.

Level 3 Diagram

This diagram shows all processes that comprise a single process on the level 2 diagram and how information moves from and to each of these processes. Level 3 diagrams may not be needed for all level 2 processes. Correctly numbering each process helps the user understand where the process fits into the overall system.

Integrating Scenario Descriptions

DFDs generally integrate scenario descriptions

Names of use cases become processes

Names of inputs and outputs become data flows

Combining “small” data inputs and outputs into a single flow

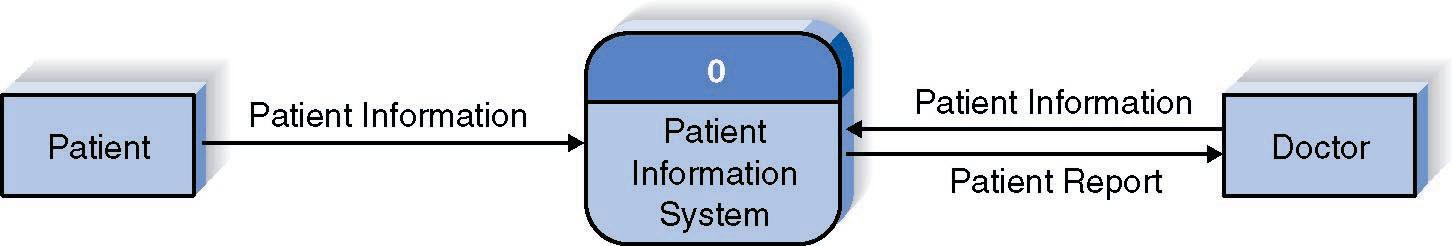
Steps in Building DFDs

* Build the context level DFD
* Create DFD fragments for each scenario
* Organize DFD fragments into level 1
* Decompose level 1 DFDs as needed
* Validate DFDs with user

DFD Fragment Tips

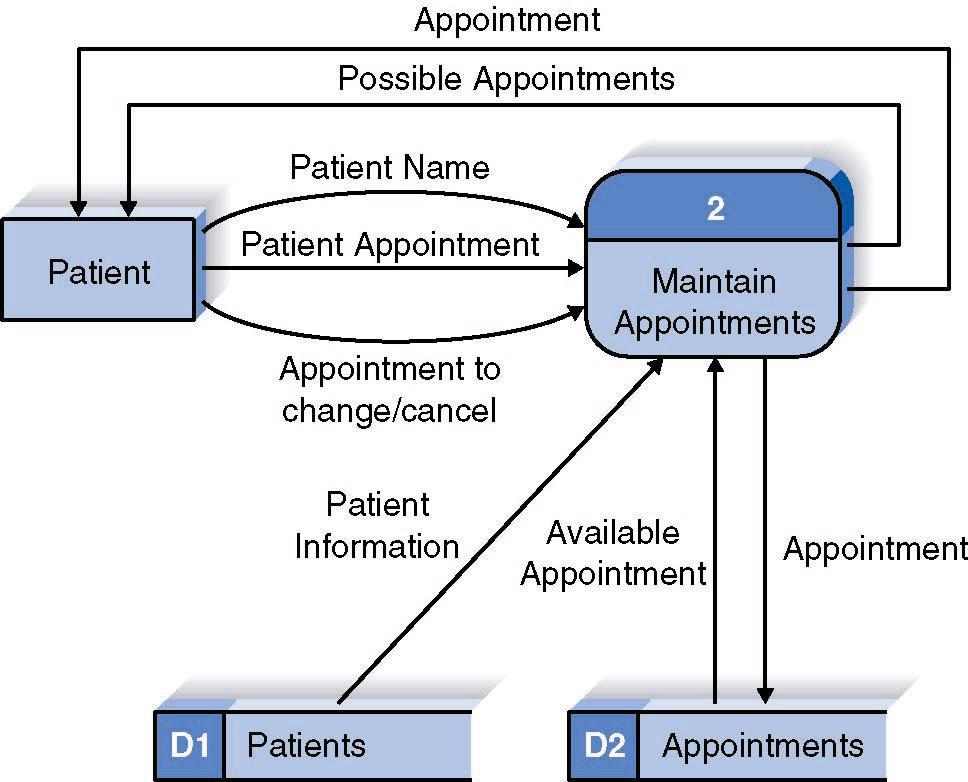
* All process names must be verb phrases
* Maintain organization’s viewpoint in naming processes
* Layouts often place
* Processes in the center
* Inputs from the left
* Outputs to the right
* Stores beneath the processes

A DFD Fragment Example



Context Level DFD

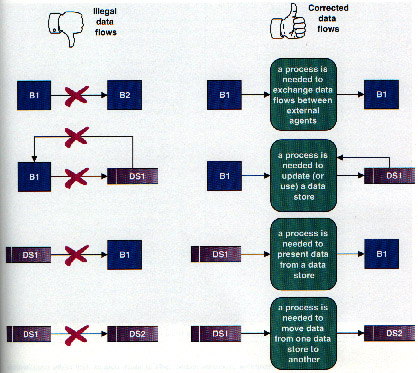
Context Level DFD



Context Level DFD

Level 1 DFD

Illegal Data Flows



## 2.4. State Transition Diagram

State Transition Diagram is developed to represent the behavior of the system under consideration. The constructs of STD are as follows

State

A set of observable circum-stances that characterizes the behavior of a system at a given time

State transition

The movement from one state to another

Event

An occurrence that causes the system to exhibit some predictable form of behavior

Action

Process that occurs as a consequence of making a transition

Guidelines for developing a state transition diagram

* Make a list of the different states of a system (How does the system behave?)
* Indicate how the system makes a transition from one state to another (How does the system change state?)
* Indicate event
* Indicate action
* Draw a state transition diagram

Example



State Transition Diagram for Microwave

## 2.5. Architectural design

The Focus of architecture design is the Mapping of Requirements into Software Architecture. DFD prepared in analysis model is analyzed to do it.

Two major structural patterns or two major alternatives are Transform (Flow) Analysis and Transaction (Flow) Analysis.

Beginning the Design Process

* Review the fundamental system model i.e. DFD and Software Requirement Specification document.
* Review and refine data flow diagrams for the software
* Determine whether DFD exhibits transform or transaction characteristics

Example

* There is a string conversion system.
* It has the ability to reverse strings, count number of characters, and append new strings with an old string.
* A user would be using this system and would be providing the string to it. The string would be validated. If approved the system would be displaying different choices including reversal of string, character counting and appending of strings.
* The user would select a choice and enter the appropriate selection. According to the selected choice the system would perform the required action.
* If “Reverse String” is selected the valid string is attained and reversed.
* If “Count Characters” is selected the valid string is attained and the number of characters are counted.
* If “Append String” is selected the valid string is attained and the user also enters a new string. Both the strings are appended together and the result produced.
* Whatever the choice selected the result is displayed.

See the diagrams on next page.

Context Level DFD



Level 1 DFD

Level 2 DFD



Program Structure/Design Architecture

Validate the Input

**String Conversion Executive**

Get User Selection

Reverse String

Controller

Reverse String

Processing Controller

Display Choices

Read Char.

Increment Count

Append String

Controller

Input String

Controller

Append Str

Get new Str

Display Output

Count String

Controller

Get Str

## 2.6. Component Level Design

Every component, which is appearing in program structure/design architecture, is logically analyzed and pseudocode or flow chart is prepared for each module. This flow chart is then given to programmer who translates it into a machine-readable code. The options available for component level design are

* Flow chart
* Box Diagram
* Decision Table
* Psuedocode

# 