# CHAPTER FOUR DESIGN

## 4.1: Design in Software Engineering

A software design is a meaningful engineering representation of some software product that is to be built. It can be traced to the customer’s requirements and can be accessed for quality against predefined criteria. In the software engineering context, design focuses on four major areas of concern: data, architecture, interfaces, and components. [1]

The design process is very important, because the emphasis in design is on quality; therefore it provides the representation of software that can be accessed for quality.

Furthermore, this is the only phase in which the customer’s requirements can be accurately translated into a finished software product or system. Thus, software design serves as the foundation for all software engineering steps that follow regardless of which process model is being employed. [1]

During the design process the software specifications are transformed into design models that describe the details of the data structures, system architecture, interface and components. Each design product is reviewed for quality before moving to the next phase of software development. At the end of the design process a design specification document is produced, which is composed of the design models that describe the data, architecture, interfaces and components. [1]



Figure 1: Generic Overview of Web Based Payroll System

## 4.2: Flow of the data



Figure 2: Level 0 Data Flow Diagram



Figure 3: Level 1 Data Flow Diagram

## 4.3: Class Diagram

A class diagram describes the types of objects in the system and the various kinds of static relationships that exist among them. A class is the description of a set of objects having similar attributes, operations, relationships and behavior.



Figure 4: Class Diagram of Domain Model



Figure 5: Class Diagram of Controller

## 4.4: Overall Sequence diagram

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. A sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur.



## 4.5: Website Architecture

Website architecture is an approach to the design and planning of websites which, like architecture itself, involves technical, aesthetic and functional criteria. As in traditional architecture, the focus is properly on the user and on user requirements. This requires particular attention to web content, a business plan, usability, interaction design, information architecture and web design.

"Website architecture" has the potential to be a term used for the intellectual discipline of organizing website content. "Web design", by way of contrast, describes the practical tasks, part-graphic and part-technical, of designing and publishing a website.

Before I got started with anything, I had to keep in mind few design guidelines to lead the project design. The design of the website should utilize the functionality of AJAX in order to make the web as user friendly as possible.

I ensure the following so AJAX can be utilized.

* Ensure no page reloads by dynamically update the content which needs to be updated.
* Ensure only JSON, html, and plain text format to be used in the data transmission between the client side and the server side.
* Ensure there is a progress status on every AJAX request is made so that the user is aware that the request is being sent to the server.

### 4.5.1: Architecture Model

There are quite a few website architecture models that can be adopted, depends on the type of website you are developing. Following are some of the architecture models; I will briefly explain three models and then the model that has been used to develop the Web Based Payroll System.

Website Architecture Models:

#### 4.5.1.1: All-in-one model



Figure 6: All-in-one Architecture Model

*http://www.webdesignfromscratch.com/snippets/ia\_diagram\_allinone.gif*

This is the simplest possible model. Everything goes on a single Home page.

#### 4.5.1.2: Flat model



Figure 7: Flat Architecture Model

*http://www.webdesignfromscratch.com/snippets/ia\_diagram\_monocline2.gif*

A flat pattern is where all pages are arranged as peers, and everyone is accessible from every other one. This is very common for simple sites, where there are a few standard topics, such as: Home, About Us, Contact Us, Products.

*I will be using flat model for the Web Based Payroll System so the accessing and navigation can be simpler and easier.*

#### 4.5.1.3: Hub-and-spoke / Daisy model



Figure 8: Hub-and-spoke (or Daisy) pattern

*http://www.webdesignfromscratch.com/snippets/ia\_diagram\_hub\_and\_spoke.gif*

This model is useful for multiple, distinct linear workflows. A good example may be an email application, where you will return to your inbox at several points, e.g. after reading a message, after sending a message, or after adding a new contact.

## 4.6: Navigation Design

Navigation design is the design of moving from one page, content, or area of the website to another. It organizes in such a way that the user will be navigating from one page to another more easily. As of Web Based Payroll System, the user will navigate using the mouse to interact with the web user interface.

### 4.6.1: Accordion

Accordion is a widget which displays collapsible content panels for presenting information in a limited amount of space.

*Accordion is used in the Web Based Payroll System to provide navigation in the web application.*



Figure 9: Web Based Payroll System Accordion

### 4.6.2: Paging

Paging is a term that will be familiar to all web users. This is where you get a piece of content that spans several pages. You are given standard tools that let you navigate previous, next, or jump directly to specific page. In Web Based Payroll System user can choose whether 20, 50, or 100 results is shown upon the request. The previous/next button will be disabled if there is no previous/next page.



Figure 10: Paging interface of Web Based Payroll System