**CHAPTER THREE**

**METHODOLOGY**

**3.1 METHODOLOGY**

This part of the research proposal involves the use of system analysis and design as a tool. it is  
dedicated to the requirement collection (done with the collaboration of the customer, in this case  
the bidder), the design of the system, done using UML use cases and class diagrams, and of the  
relational database, done with EER (extended entity relationship diagram) as modeling  
technique.

We are implementing web-based auction system for selling and buying items. Users can domestically sign up and join to the auction through a web by using a web  
browser. Two different types of actors are using this system: user actor, which can be registered to the system and selling an item after uploading the item to the system or buying items by placing a bid. The bid must be higher than the start or the current bid. Administrative actor, which can login to the system with a particular username and password. Administrative performs many actions in the system that contain managing category, removing users, editing categories, etc. The functionality and more features are explained in more detail in this chapter.

The main characteristics of the system are:

* Each item on auction is distinguished by an id, name, a description, a picture,  
  and a category (i.e. music, beauty, accessories, etc.)
* Every item on the auction has a commencement a time and date, and an end  
  time.
* The user capable of adding an item on auction for sale, adjusting it and remove  
  it by login to the auction site and accessing his/her personal page.
* A visitor can access the web site through a web browser and look at the items on auctioning but he/she unable to make a bid without register to the website

**3.2 SYSTEMS DEVELOPMENT METHODOLOGY**

A system development methodology refers to the framework that is used to structure, plan, and  
control the process of developing an information system. A wide variety of such frameworks  
have evolved over the years, each with its own recognized strengths and weaknesses. One system  
development methodology is not necessarily suitable for use by all projects. Each of the  
available methodologies is best suited to specific kinds of projects, based on various technical,   
organizational, project and team considerations.

**3.2.1 SYSTEM DESIGN**

**Rapid application development**

This is a development methodology in which a system designer produces prototypes for an end  
user. The end user reviews the prototype and offers feedback on its suitability. This process is  
repeated until the end user is satisfied with the final system. It puts less emphasis on planning  
tasks and more emphasis on development. It is driven by user interface requirements.

**RAPID APPLICATION DEVELOPMENT**

**REQUIREMENT**

**PLANNING**

**DESIGN PHASE**

**CONSTRUCTION**

**CUTOVER**

**Figure 3.1 stages of RAD()**

**Requirement planning**

This phase is similar in concept to the traditional analysis phase. However, the nature of planning  
when constantly reviewing prototypes means that it is often revisited within what would  
traditionally be regarded as design and implementation. Rather than the client working with the  
systems analyst to devise a software specification, the client is continually involved in reviewing  
the design and related prototypes that are generated throughout the process. This can result in  
alteration to the client’s requirements. It is often the case that clients have little or no technical  
awareness. Only when they see prototypes of different elements of the system during the  
subsequent design are better prepared to firm up and finalize the requirements that they may  
have had during a traditional analysis phase. Often requirement planning focuses on planning for  
a range of prototypes.

**Design phase**

Once the clients initial needs have been determined, the software development team will invest  
as much as they can in generating prototypes that can be taken back to the client for refinement.  
This avoids situations where the client agrees with a proposal on paper but then disagrees with  
the software that has been generated over a long period of time which can happen when using the  
waterfall method. It is usual for some crossover to exist between design and implementation. It is only possible to construct a prototype of elements of the system if there has been some degree of  
implementation.

**Construction phase**

Within the construction phase, the programmers take the preferred prototypes and begin to  
construct a formal solution. This is naturally involves implementation and will focus on aspects  
of the software that were not part of the prototypes shared with the clients such as those that have  
been time boxed. The advantage if having already designed prototypes is that some of the  
implementation may have been complete already. As with all aspects of RAD, the client can be  
involved if necessary. This would be likely during testing.

**Cutover phase**

One the construction and in house testing has been completed, the project moves into the cutover  
phase. Within this phase the client and the client’s staff will be using the software as normal. The  
development team will remain available for a predefined timescale to carry out corrective  
maintenance, incorporating feedback from the client in relation to the client’s evaluation of hoe  
the software operates in real context. This phase is regarded as streamlined and concise  
equivalent of traditional evaluation and maintenance.

**Justification of RAD Faster delivery time**

This is achieved by rapid prototyping and by using automated tools like Computer Aided Software Engineering of CASE tools that enable the developers to re-use previously generated codes thus saving time needed for manual coding.

**Better quality**

Although RAD may imply some compromise in terms of scalability and the range of features in  
the product been delivered. It enhances the product quality by providing a considerable reduction  
in the errors due to the use of automation tools and prototyping. Errors and omissions are  
detected in the early stages of development therefore preventing ay extra effort and cost.

**Reduced risk**

Due to the iterative approach and prototyping, testing and integration of end user feedback  
happens at each stage of development. Hence, the end products have less number of defects and  
changes, thereby minimizing the risks involved in the project. There is little or no testing effort  
involved.

**3.3 SYSTEM REQUIREMENT ANALYSIS**

**Functional requirements**

It deals with what the system should do or provide to users. They include description of required  
functions, outline of associated reports or inline queries and details of data to be held in the  
system. A user can register to the system and able to login once registered, browse the available items, post his/her bid and post a new auction. The administrator, on the other hand, can insert and  
modify available data about items, users and categories of items.

**Non-Functional requirements**

They detail constraints, targets or control mechanisms for the new system. They describe how,  
how well or to what standard a function should be provided. The system should be user friendly, flexible enough to be done some changes if need be, it should be able to be used by any auction company, it should also be easy to be used by anyone interested.

**3.4 DATABASE STRUCTURE**

****

**Figure 3.2 Database Diagram**

|  |  |  |
| --- | --- | --- |
| S/N | Name | Description |
| 1 | AspNetUsers | Use for the user details storage |
| 2 | AspNetRoles | Role of the users definition |
| 3 | AspNetUserRoles | Stores the users role |
| 4 | Categories | Houses the categories of product in the system |
| 5 | Products | Products for the system |
| 6 | AuctionItems | Product Auction details for real stimulation of auction |
| 7 | Bids | Stimulate real life bidding actions relating to a particular auction |

Table 3.1 Database Table Description

**3.5 HARDWARE AND SOFTWARE REQUIREMENTS  
3.5.1 HARDWARE REQUIREMENTS (MINIMUM)  
Server side**

|  |  |
| --- | --- |
| Processor | 2.20 GHz |
| Ram | 2 GB |
| Hard disk | 10 GB Free Space |

**Client side**

|  |  |
| --- | --- |
| Processor | 2 GHz |
| Ram | 1 GB |
| Hard disk | 4 GB Free Space |

**3.5.2 SOFTWARE REQUIREMENTS (MINIMUM)  
SERVER SIDE**

|  |  |
| --- | --- |
| Operating system :- | Windows server 2003 or any compatible server OS |
| Framework | Net framework 4.6 |
| Web Server | IIS 8.0 |
| Front End | ASP. NET MVC with C# (.NET) using Visual Studio 2017 |
| Back End | SQL server 2014 with Microsoft SQL Management Studio |

**Client side**

|  |  |
| --- | --- |
| Operating system | Windows XP or any compatible OS |
| Browser | Google Chrome, IE, Opera and FireFox |