**CONSTRUCTION MANAGEMENT SYSTEM**

*Project Report Submitted by*

**ABIN SUNIL**

**Reg. No.: AJC21MCA-2004**

*In Partial fulfillment for the Award of the Degree Of*

**MASTER OF COMPUTER APPLICATIONS**

**(MCA TWO YEAR)**

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING**

**KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with ‘A’ grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

# 2022-2023

## DEPARTMENT OF COMPUTER APPLICATIONS

### AMAL JYOTHI COLLEGE OF ENGINEERING

**KANJIRAPPALLY**



**CERTIFICATE**

This is to certify that the Project report, “**CONSTRUCTION MANAGEMENT SYSTEM”** is the bona fide work of **ABIN SUNIL(Reg.No: AJC21MCA-2004)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2022-23.

**Mr. Ajith G S Ms. Meera Rose Mathew**

**Internal Guide Coordinator**

**Rev. Fr. Dr. Rubin Thottupurathu Jose**

**Head of the Department**

**DECLARATION**

I hereby declare that the project report **“CONSTRUCTION MANAGEMENT SYSTEM”** is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2022-2023.

**Date: ABIN SUNIL**

**KANJIRAPPALLY Reg: AJC21MCA-2004**

# ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Mr. Ajith G S** for his inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

ABIN SUNIL

# ABSTRACT

Now a day, technology is on a boost. People wish to live a luxurious life with minimum physical work. Here we provide a website for find out the best construction companies. This website help to the customers for find out the companies and find out the affordable designs and check whether the total cost is affordable or not. The customer can login to the system and set the requirements in detail and uploads the requirements into the website. Also the qualified companies can create their own accounts and view the requirements. The companies have the option for accept or reject the customer‘s requirements.

**CONTENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL. NO** | | **TOPIC** | **PAGE NO** | |
| **1** | | **INTRODUCTION** | **1** | |
| **1.1** | | **PROJECT OVERVIEW** | **2** | |
| **1.2** | | **PROJECT SPECIFICATION** | **2** | |
| **2** | | **SYSTEM STUDY** | **5** | |
| **2.1** | | **INTRODUCTION** | **6** | |
| **2.2** | | **NATURAL SYSTEM** | **6** | |
| **2.3** | | **DRAWBACKS OF EXISTING SYSTEM** | **7** | |
| **2.4** | | **DESIGNED SYSTEM** | **7** | |
| **2.5** | | **ADVANTAGES OF PROPOSED SYSTEM** | **7** | |
| **3** | | **REQUIREMENT ANALYSIS** | **8** | |
| **3.1** | | **FEASIBILITY STUDY** | **9** | |
| **3.1.1** | | **ECONOMICAL FEASIBILITY** | **9** | |
| **3.1.2** | | **TECHNICAL FEASIBILITY** | **9** | |
| **3.1.3** | | **BEHAVIORAL FEASIBILITY** | **10** | |
| **3.1.4** | | **FEASIBILITY STUDY QUESTIONNAIRE** | **10** | |
| **3.2** | | **SYSTEM SPECIFICATION** | **11** | |
| **3.2.1** | | **HARDWARE SPECIFICATION** | **11** | |
| **3.2.2** | | **SOFTWARE SPECIFICATION** | **11** | |
| **3.3** | | **SOFTWARE DESCRIPTION** | **11** | |
| **3.3.1** | | **PHP** | **11** | |
| **3.3.2** | | **MYSQL** | **12** | |
| **4** | | **SYSTEM DESIGN** | **14** | |
| **4.1** | | **INTRODUCTION** | **15** | |
| **4.2** | | **UML DIAGRAM** | **15** | |
| **4.2.1** | | **USE CASE DIAGRAM** | **16** | |
| **4.2.2** | | **SEQUENCE DIAGRAM** | **22** | |
| **4.2.3** | | **STATE CHART DIAGRAM** | **25** | |
| **4.2.4** | | **ACTIVITY DIAGRAM** | **26** | |
| **4.2.5** | | **CLASS DIAGRAM** | **28** | |
| **4.2.6** | | **OBJECT DIAGRAM** | **29** | |
| **4.2.7** | | **COMPONENT DIAGRAM** | **29** | |
| **4.2.8** | | **DEPLOYMENT DIAGRAM** | **30** | |
| **4.3** | | **USER INTERFACE DESIGN USING FIGMA** | **32** | |
| **4.4** | | **DATA BASE DESIGN** | **37** | |
| **4.4.1** | | **RELATIONAL DATABASE MANAGEMENT SYSTEM** | **37** | |
| **4.4.2** | | **NORMALIZATION** | **38** | |
| **4.4.3** | | **SANITIZATION** | **40** | |
| **4.4.4** | | **INDEXING** | **40** | |
| **4.4.5** | | **TABLE DESIGN** | **41** | |
| **5** | | **SYSTEM TESTING** | **45** | |
| **5.1** | | **INTRODUCTION** | **46** | |
| **5.2** | | **TEST PLAN** | **47** | |
| **5.2.1** | **UNIT TESTING** | | **47** |
| **5.2.2** | **INTEGRATION TESTING** | | **48** |
| **5.2.3** | **VALIDATION TESTING** | | **48** |
| **5.2.4** | **USER ACCEPTANCE TESTING** | | **49** |
| **5.2.5** | **AUTOMATION TESTING** | | **49** |
| **5.2.6** | **SELENIUM TESTING** | | **50** |
| **6** | **IMPLEMENTATION** | | **51** |
| **6.1** | **INTRODUCTION** | | **52** |
| **6.2** | **IMPLEMENTATION PROCEDURE** | | **53** |
| **6.2.1** | **USER TRAINING** | | **53** |
| **6.2.2** | **TRAINING ON APPLICATION SOFTWARE** | | **53** |
| **6.2.3** | **SYSTEM MAINTENANCE** | | **53** |
| **7** | **CONCLUSION & FUTURE SCOPE** | | **55** |
| **7.1** | **CONCLUSION** | | **56** |
| **7.2** | **FUTURE SCOPE** | | **56** |
| **8** | **BIBLIOGRAPHY** | | **57** |
| **9** | **APPENDIX** | | **59** |
| **9.1** | **SAMPLE CODE** | | **60** |
| **9.1.1** | **LOGIN PAGE** | | **60** |
| **9.1.2** | **LABOUR REGISTRATION** | | **70** |
| **10.1** | **SCREEN SHOTS** | | **82** |

## List of Abbreviation

IDE - Integrated Development Environment

HTML - Hyper Text Markup Language.

CSS - Cascading Style Sheet

JS- Java Script

SQL - Structured Query Language

PHP – PHP: PHP Hypertext Preprocessor

UML - Unified Modeling Language

# CHAPTER 1

# INTRODUCTION

### PROJECT OVERVIEW

The project ‘Construction Management System’ is defined as a website that aims to all the levels of management providing information within a construction field. This system can be used as find out the experienced construction companies and their services. In this website different construction company can get their own accounts and explore the customer requirements. Also customers can create their own accounts and uploads their requirements.

It will also provide some features like customers can make the necessary changes within their account information and their uploaded requirement. Customers can access the website to use the login features. The companies can view the latest customer requirements.

### PROJECT SPECIFICATION

At present, there is no centralized system for managing labors properly in the order they arrive in Kerela as well as allocating them proper work to earn money. A platform consisting of user, contractor, labor and police together is not established yet. So that’s the main motive behind the development of the proposed system ‘Labour Management System’.

Once a labour is registered, his personal details will be strictly evaluated by admin and then by police. A labor will allow to continue only after the successful verification by admin and police. On successful enrollment to system, contractors can choose verified labors to take part In their work. Works are added by admin along with the corresponding minimum wage per hour . Also contractor fee, sub contractor fee will be appended to work details while registering a work. There is no restriction that a contractor be able to contract a work of a particular area. Contractor has freedom to choose any work of his interest . But sub-contractor is under the supervision of contractor and sub contractor might be expert in any kind of job. Based on the requirements, contractor can hire a sub-contractor to his works on permanent basis. Once a sub-contractor is added , he is only the part of the works assigned by his authorized contractor. On committing a work, either contractor or sub contractor can allocate labors to the work. Hours of work, time to start work, estimated date and time will be added by contractor.

In this proposed Construction management website. It will be a platform for customer can find out the best construction company. They can find out the company based on the their personal requirements. The current requirement status of the uploaded requirements are viewed by the customer profile. The admin can anytime view the total company that are active and how many customers are in there.

For the purpose of getting insightful knowledge about the performance of companies a sentimental analysis using NLTK (Natural Language Tool-Kit) library has implemented in Django framework. The feedback data consists of several interesting feedbacks collected through the customer feedback section.

The major modules included in the system are:

* Admin
* Company
* User

# CHAPTER 2

# SYSTEM STUDY

### INTRODUCTION

Data collection and analysis, problem-solving, and system alteration ideas are all steps in the system analysis process. During this problem-solving process, there must be substantial contact between the system developers and users. A system analysis or research should be the first step in any system development process. The system analyst closely examines the operation of the current system in the role of an interrogator. The system's input is acknowledged, and the system is seen as a whole. The organization's outputs and various processes might be connected. System analysis involves comprehending the problem, identifying the crucial elements, evaluating and synthesizing the many components, and choosing the best or, at the very least, most acceptable course of action .

Preliminary research is the process of gathering and analyzing data in order to use it for future system inquiries. Initial research requires problem solving , therefore system users and developers must work closely together. It carries out several feasibility studies. These studies give a thorough picture of how the system functions, which can be utilised to choose the best methods for study and analysis.

### NATURAL SYSTEM

In present scenario, In the natural system people have to physically visit the construction site or contractors for construction works of their needs. They searched for the construction works with the help of the company . In this method time as well as physical work is required. There is a chance that the customer demanded construction work is not available on the company, They may request to agreement with a particular company. Without gathering any background information of these companies.

### DRAWBACKS OF EXISTING SYSTEM

* Not easy to approach the company directly
* Credibility issues
* Providing quality service is difficult
* No proper medium to verify their personal details well
* No security
* lacking reliability

### DESIGNED SYSTEM

Proposed system is more flexible when compared with the manual gear shifting method of finding companies step by step with more time. This system helps find companies within a few clicks with security features approval of admin. so trustworthiness and reliability will be high when we glance at this. In designed system, a lot of actions performed with huge man power can be minimized , instead manpower can be used for more productive tasks that requires building fame of the organization by understanding business process in a quick-eyed inspection.

### ADVANTAGES OF PROPOSED SYSTEM

• Less manual effort with advanced analytics

• Time efficient system which handles everything in a timely fashion

• Interaction can be leveraged through the system.

• Security is improved.

• Productivity can be leveraged

• Review based rating for estimating best performing company can be found

• Effective decision making and reward granting for the contractors can be done easily

# CHAPTER 3

# REQUIREMENT ANALYSIS

## FEASIBILITY STUDY

Planning, organizing, and managing resources to ensure the achievement of particular project goals and objectives is the process of project management. A feasibility study is a preliminary examination of a prospective project or end to determine its merits and viability. A feasibility study aims to provide an objective assessment of the technical, economic, financial, legal, and environmental elements of a proposed project. The information can then be used by decision- makers to decide whether to proceed with the project or not. The findings of the feasibility study can also be used to develop a practical project plan and budget. It cannot be simple to determine whether or not a proposed project is worthwhile pursuing without a feasibility study. The document provides the feasibility of the project that is being designed and lists. Various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibility. The following are its features:

### Economical Feasibility

Cost and benefit analyses are required to support the developing system. criteria to make sure that focus is on the project that will yield the best results and return the earliest. The price that would be involved in developing a new system is one of the variables. Some significant financial queries raised during the initial investigation include the following:

The costs conduct a full system investigation?

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system.

* The cost of the hardware and software?
* Also all the resources are already available.

As per the primary studies, I have found that economically the system is feasible enough to proceed with this idea especially when considering it’s future benefits when making publicly accessible through internet. It can also support future economy by reducing manual and time-consuming processes that need a specific cost to do it.

### Technical Feasibility

The system needs to be assessed first from a technical standpoint. The outline design of the system requirement in terms of input, output, programs, and procedures must serve as the foundation for the assessment of this feasibility. After determining an outline investigation must continue to identify the necessary equipment kind. Once the system has been designed, there are several ways to run it.

When considering it’s technical aspects, We can see that all the online equivalent platforms we use to automate our daily tasks will be technically feasible solutions . In case of labour management, a lot of labours and contractors have sudden access to their profile, work requests, work management etc. Beyond this, police module added in this application will be responsible to keep the security of people ( customers ) and labours as well.

Today , in this busy world, everyone can access the services provided by this platform can be enjoyed easily with a few amount of clicks. No one has to search and wander for contractor labour for their works pending at home. Robustness and user-friendliness of this application helps to attract more customers and users to the system. Definitely this will be a great initiative to the world of labour management. Some of the questions and things to discuss before proceeding with a digital solution to measure it’s technical feasibility is as follows:

Is the project feasible within the limits of current technology.

* YES
* Technical issues raised during the investigation are:
* NOTHING
* Can the technology be easily applied to current problems?
* YES
* Does the technology have the capacity to handle the solution?
* YES

### Behavioral Feasibility

The proposed system includes the following questions:

* Is there sufficient support for the users
* YES
* Will the proposed system cause harm?
* NO

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.

### Feasibility Study Questionnaire

This simplified version of Construction management which allow the users to find out the best construction company. Those who have been registered can be visible to companies who had been in the system. Whenever a customer, in this context, common people like us (users) sends a request regarding the need of work.

1. How actually the company is being selected for the work ?

Users are normally selected the company based on the feedback of previous customers.

1. How will be the users approach a company for their works ?

They normally approach companies over the phones.

1. How many time take a company to done a work?

Normally between 8 to 12 months.

1. What is the most significant criteria for choosing a company ?

A company is always be chosen by public on the basis of their previous works they have done.

1. How the company rated?

Based on the customer feedback.

1. How can we ensure companies trustfulness?

There is no a proper mechanism for that.

1. Normally, how can we agreement with the company ?

By signing the hand written agreement

1. Can user design the plan directly rather than approaching companies?

No. Normally , no one will do so.

1. How is their technical knowledge ?

Now a days they have improved much better. They know basic knowledge.

1. Suppose you have been contracted a work, immediately the company who has agreed to work is unable to reach for a day. How would you manage such schedules ?

If such a situation occurs, primarily we will request users to postpone the work with a little delay. If the work is very urgent, definitely we will contact another reputed company .

## SYSTEM SPECIFICATION

### Hardware Specification

Processor - Intel core i5

RAM - 8 GB

Hard disk - 1 TB HDD

### Software Specification

Front End - HTML, CSS

Backend - MYSQL

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, J Query, Python, CSS

## SOFTWARE DESCRIPTION

### Python

Python is a well-known programming language created by Guido van Rossum and released in 1991. It is used for server-side web development, software development, mathematics, and system programming. Python is an object-oriented, high-level programming language with dynamic semantics that is interpreted. It is particularly suitable for use as a scripting or glue language to tie existing components together because to its high-level built-in data structures, dynamic typing, and dynamic binding. Python's concise, easy-to-learn syntax emphasises readability, lowering programme maintenance costs. Python provides support for modules and packages, which promotes programme modularity and code reuse. For all major platforms, the Python interpreter and substantial standard library are freely available in source or binary form, and can be freely distributed.

### MySQL

MySQL, the most popular open source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

**• MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

**• MySQL databases are relational.**

Instead of placing all the data in one huge warehouse, a relational database keeps the data in individual tables. Physical files that are optimized for speed contain the database structures. The logical model provides a flexible programming environment with objects like databases, tables, views, rows, and columns. One-to-one, one-to-many, unique, compulsory or optional, and "pointers" between distinct tables are a few examples of the rules you might build up to regulate the relationships between various data fields. With a well-designed database, your application won't ever encounter inconsistent, duplicate, orphan, out-of-date, or missing data since the database enforces these rules. "Structured Query Language" is what the SQL portion of "MySQL" stands for. The most popular standard language for accessing databases is SQL. Depending on your programming environment, you might explicitly enter SQL (for example, to generate reports), incorporate SQL statements into other languages' code, or use a language-specific API that obscures the SQL syntax. By way of the ANSI/ISO SQL Standard, SQL is defined. Since its inception in 1986, the SQL standard has undergone multiple revisions. In this document, "SQL92" refers to the 1992 standard, "SQL: 1999" to the 1999 standard, and "SQL: 2003" to the most recent version of the standard. The SQL Standard as it exists at any one time is referred to as "the SQL standard".

**• MySQL software is Open Source**.

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

**• The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If scalability, reliability, speed etc. are what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

**• MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that includes a multithreaded SQL server that supports numerous client programs and libraries, administrative tools, and a large variety of application programming interfaces (APIs). Additionally, we provide MySQL Server as an integrated multi-threaded library that you can link into your program to create a standalone offering that is smaller, faster, and simpler to operate.

# CHAPTER 4

# SYSTEM DESIGN

* 1. **INTRODUCTION**

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

## UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997. UML stands for Unified Modeling Language. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

• Class diagram

• Object diagram

• Use case diagram

• Sequence diagram

• Activity diagram

• State chart diagram

• Deployment diagram

• Component diagram

## USE CASE DIAGRAM

A use case diagram is a graphic depiction 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 requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems. System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

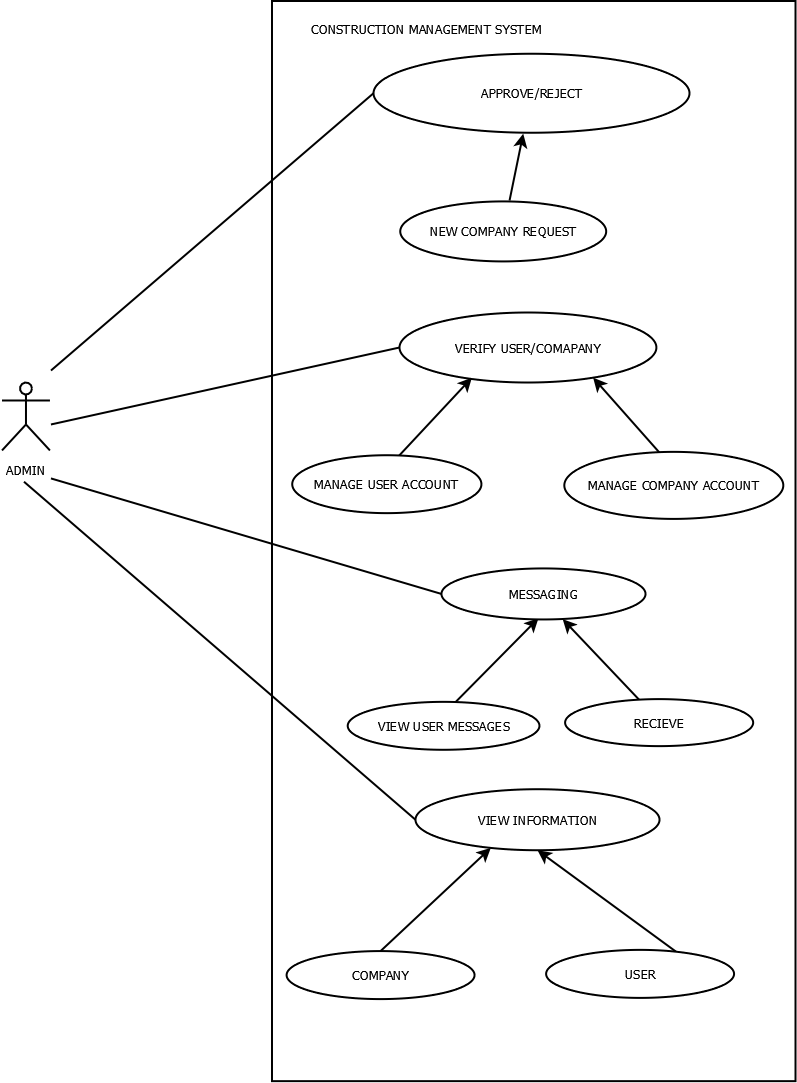
* The boundary, which defines the system of interest in relation to the world around it.
* The actors, usually individuals involved with the system defined according to their roles.
* The use cases, which are the specific roles are played by the actors within and around the system.
* The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

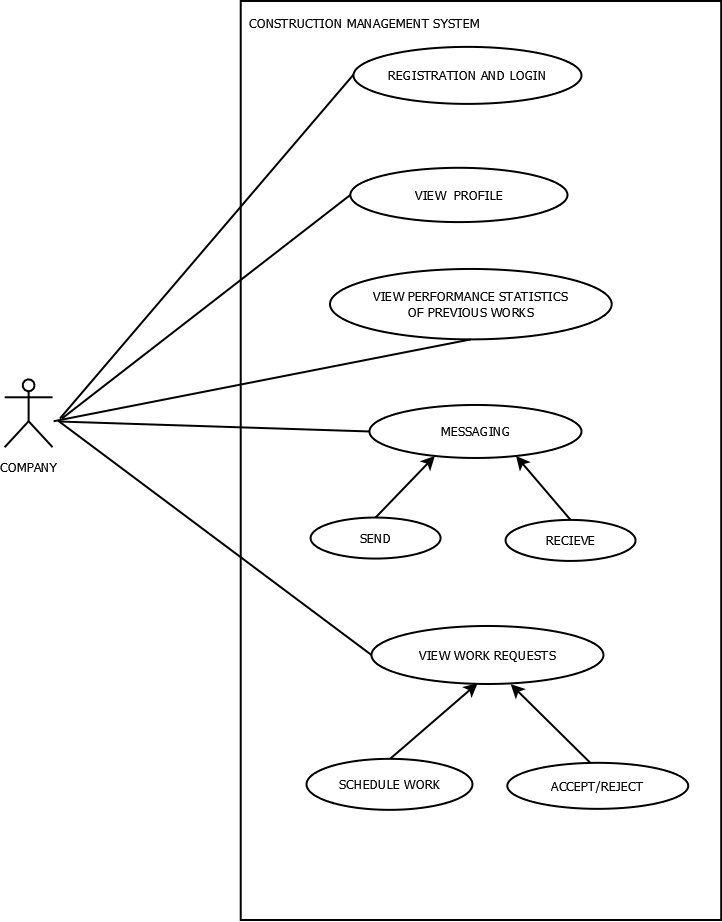
* The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
* Give a suitable name for actors.
* Show relationships and dependencies clearly in the diagram.
* Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
* Use notes whenever required to clarify some important points.

DIAGRAM

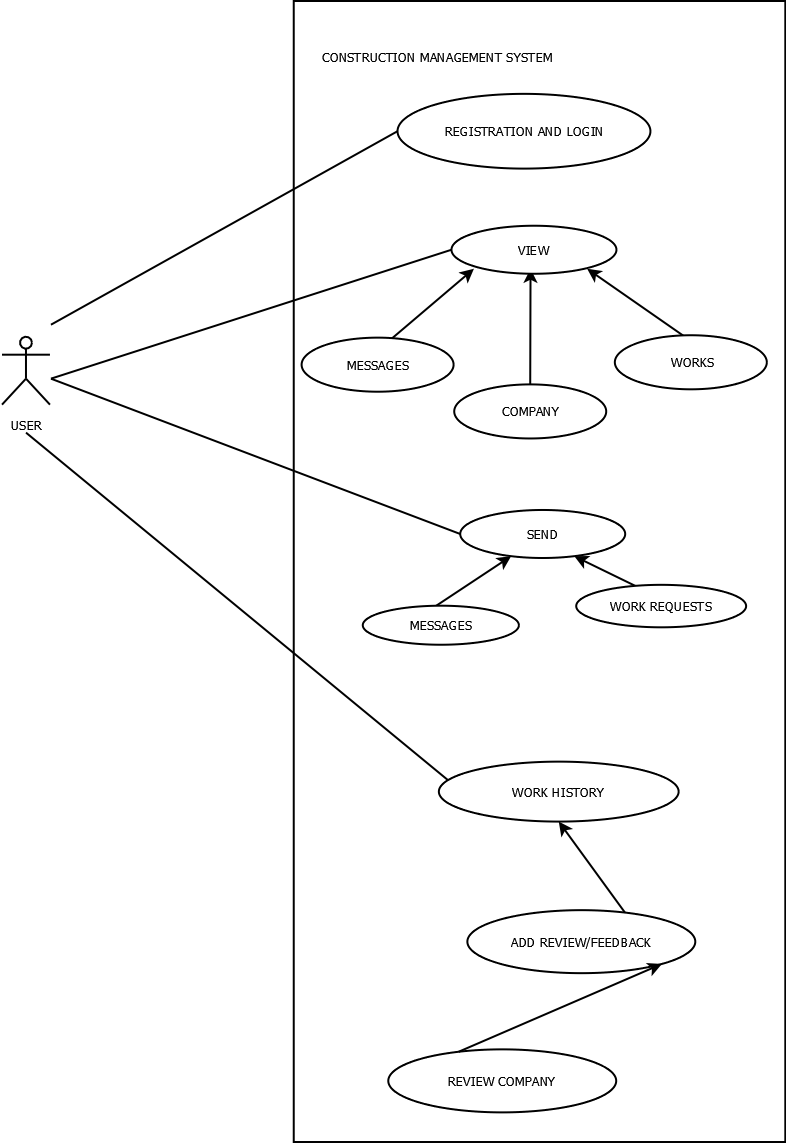
USE CASE DIAGRAM FOR ADMIN



USE CASE DIAGRAM FOR COMPANY



USE CASE DIAGRAM FOR USER



## SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

**Sequence Diagram Notations –**

1. **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.
2. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.
3. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

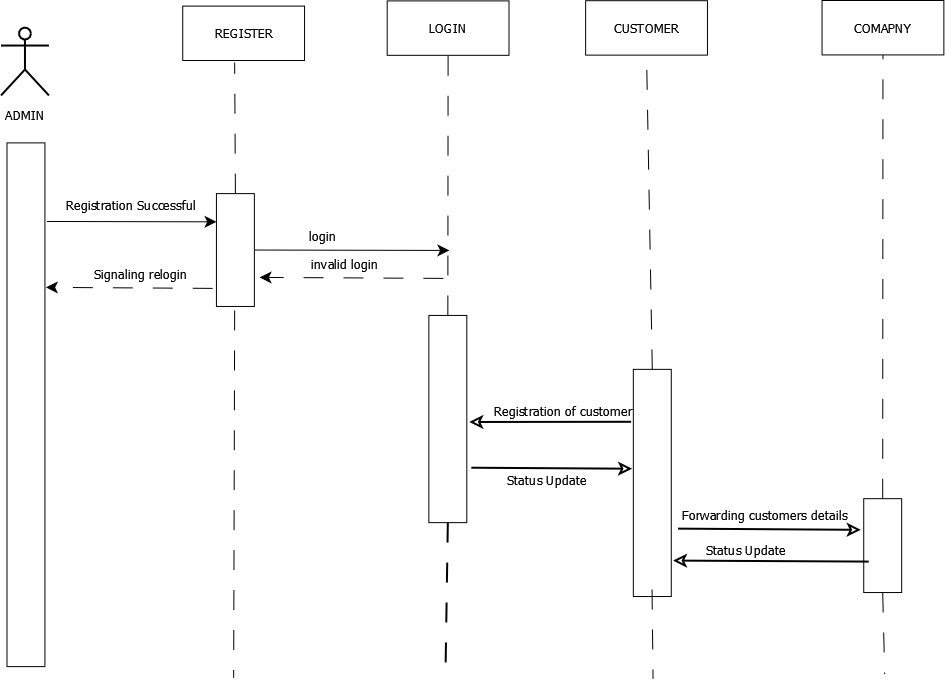
Messages can be broadly classified into the following categories:

* Synchronous messages
* Asynchronous Messages
* Create message
* Delete Message
* Self-Message
* Reply Message
* Found Message
* Lost Message

1. **Guards** – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

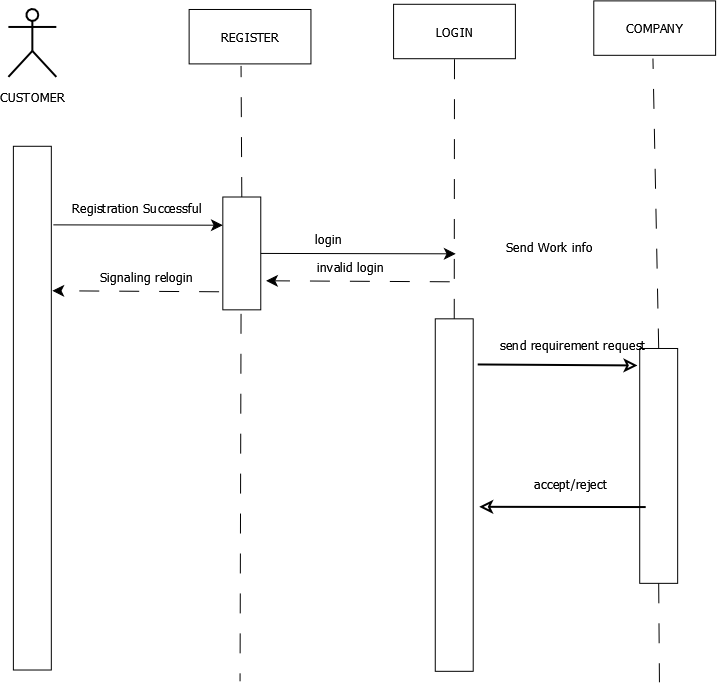
**Uses of sequence diagrams –**

* Used to model and visualize the logic behind a sophisticated function, operation or procedure.
* They are also used to show details of UML use case diagrams.
* Used to understand the detailed functionality of current or future systems.
* Visualize how messages and tasks move between objects or components in a system

DIAGRAM

SEQUENCE DIAGRAM ADMIN

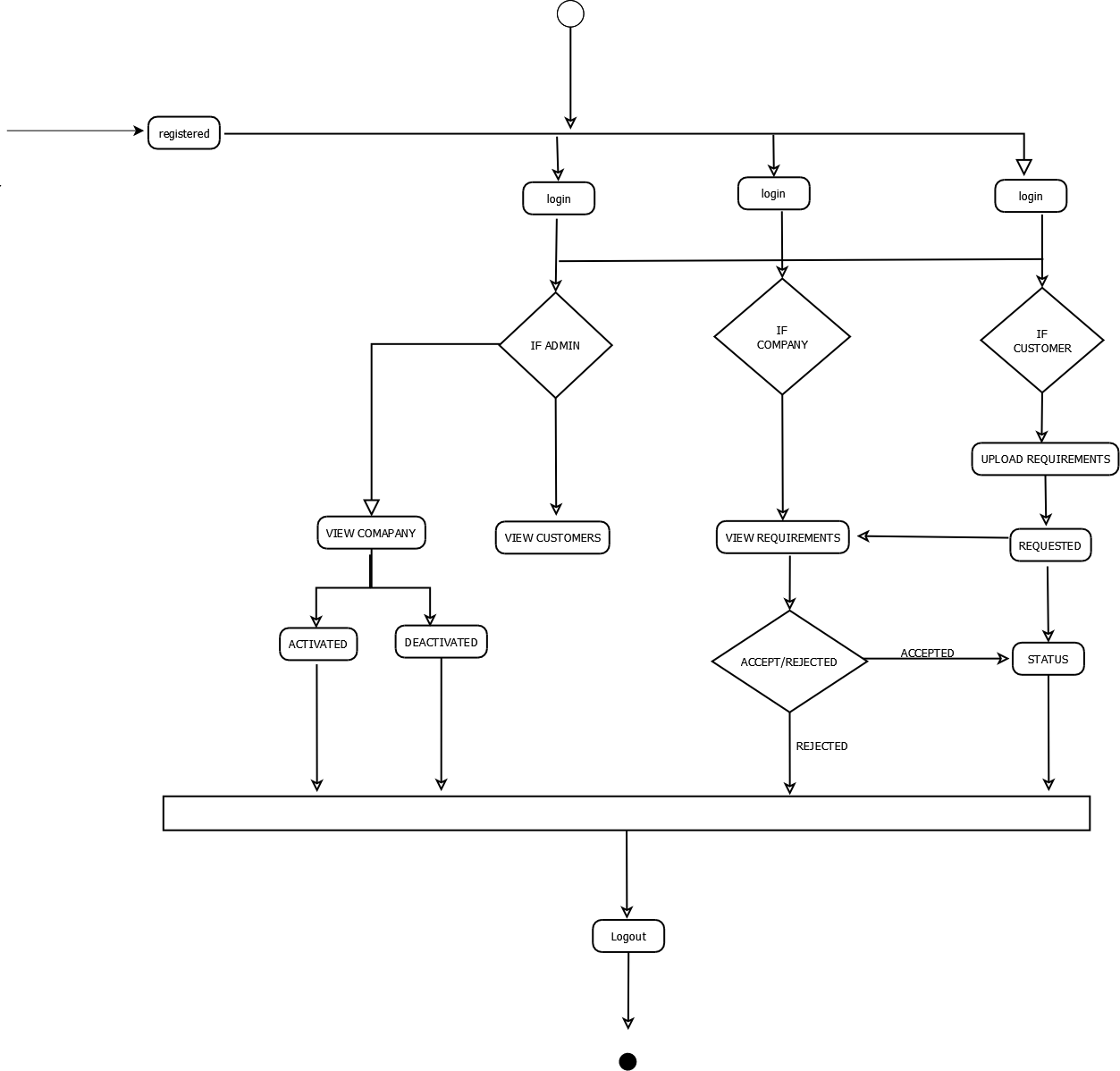
SEQUENCE DIAGRAM OF CUSTOMER



## 4.2.2 State Chart Diagram

A particular form of diagram used in computer science and related subjects to explain how systems behave is called a state diagram. State diagrams call for the system being represented to consist of a finite number of states; occasionally, this is the case, and other times, it's only an acceptable abstraction. State diagrams come in a variety of shapes and sizes, each with a distinct meaning. State diagrams are there to provide a system's behaviour an abstract explanation. In order to evaluate and demonstrate this behaviour, a sequence of events that could occur in one or more hypothetical states is employed. "Each diagram typically depicts objects of a single class and monitor the different states of its objects across the system," according to this.

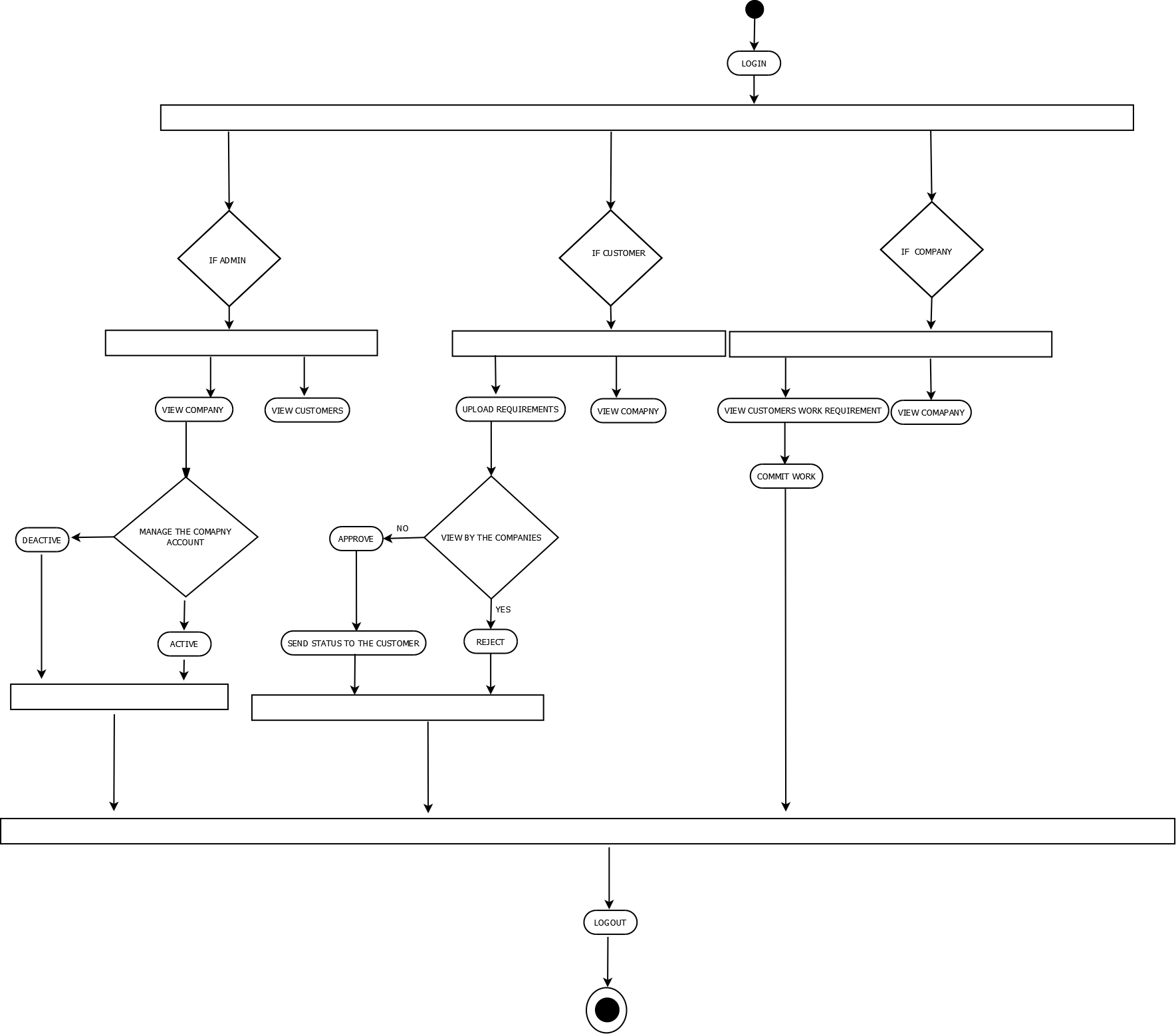
DIAGRAM



## Activity Diagram

Activity diagrams depict how different levels of abstraction of activities are linked to provide a service. Typically, an event should be completed by some activities, particularly when the activity is intended to do multiple separate goals that need coordination. Another typical requirement is how the events in a single use case interact with one another, particularly in use cases where operations may overlap and require coordination. It may also be used to show how a collection of interrelated use cases interacts to reflect business operations.

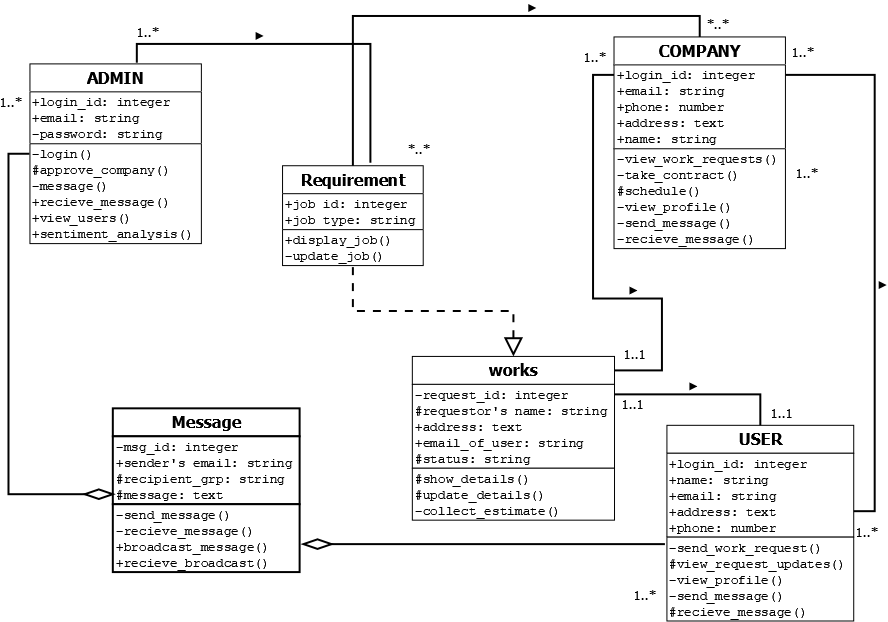
DIAGRAM



## Class Diagram

Class diagram is a static diagram. It represents the static view of the application. Class diagrams are useful for vizualising, describing, and documenting various system components as well as for writing executable code for software applications. A class diagram describes the constraints imposed on the system together with the properties and operations of a class. The only UML diagrams that can be directly converted into objectoriented languages are class diagrams, which are extensively utilised in the designing of object-oriented systems. An assortment of classes, interfaces, affiliations, partnerships, and limitations are displayed in a class diagram. It also goes by the name "structural diagram."

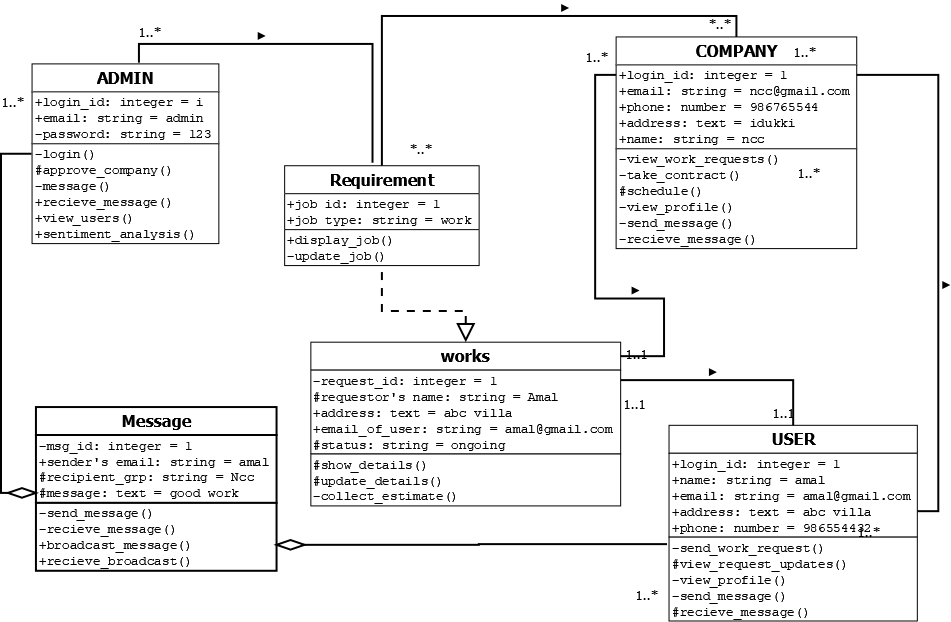
DIAGRAM



## Object Diagram

Class diagrams are necessary before object diagrams can be created since they are the ancestor of object diagrams. An object diagram represents a particular instance of a class diagram. The underlying concepts used in class and object diagrams are the same. Object diagrams may also describe the static view of a system, although this static view only depicts a current state of the system. Object diagrams are used to show links between a set of things.

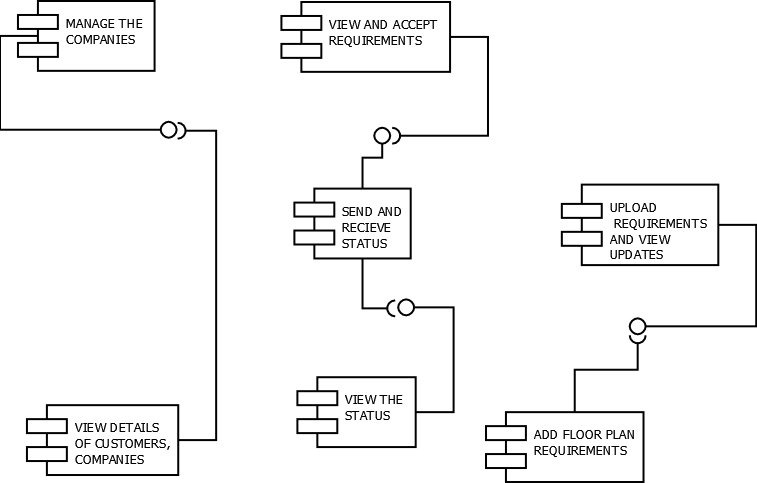
DIAGRAM



## Component Diagram

Component diagrams have different behaviors and personalities. The physical parts of the system are represented using component diagrams. Executables, libraries, files, documents, and other items that are physically present in a node are just a few examples. Component diagrams are used to show how the components of a system are connected and arranged. These diagrams may also be used to construct systems that can be run.

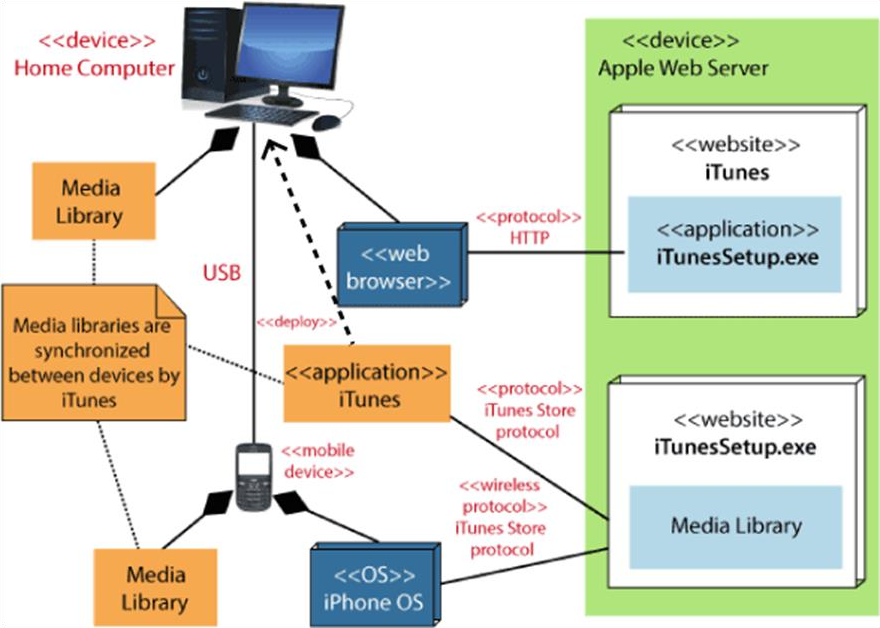
DIAGRAM



**4.2.6 Deployment Diagram**

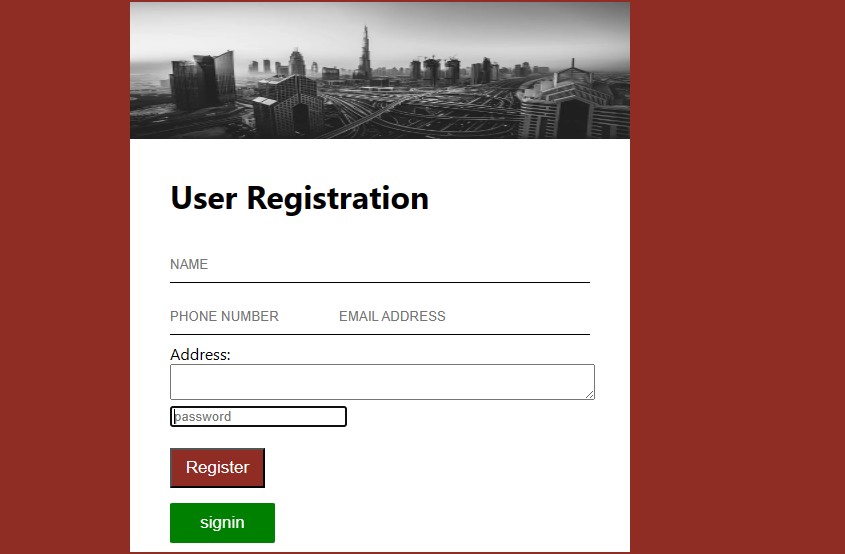
An execution architecture of a system, containing nodes like hardware or software execution environments, and the middleware linking them, is shown in a deployment diagram, a form of UML diagram. Typically, deployment diagrams are used to represent the actual hardware and software of a system. By using it, you can comprehend how the hardware will physically deliver the system. In contrast to other UML diagram types, which primarily depict the logical components of a system, deployment diagrams assist describe the hardware structure of a system.

DIAGRAM

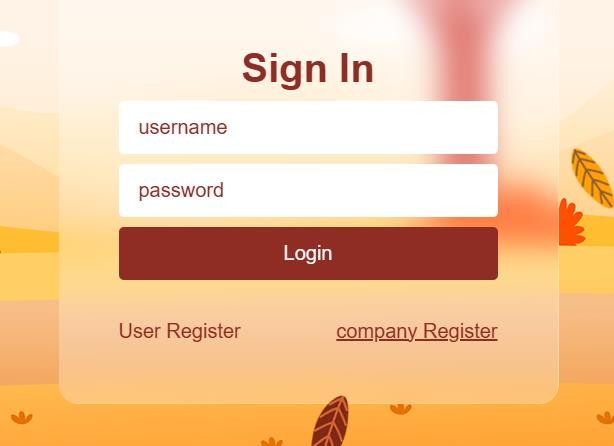


## 4.3 USER INTERFACE DESIGN USING FIGMA

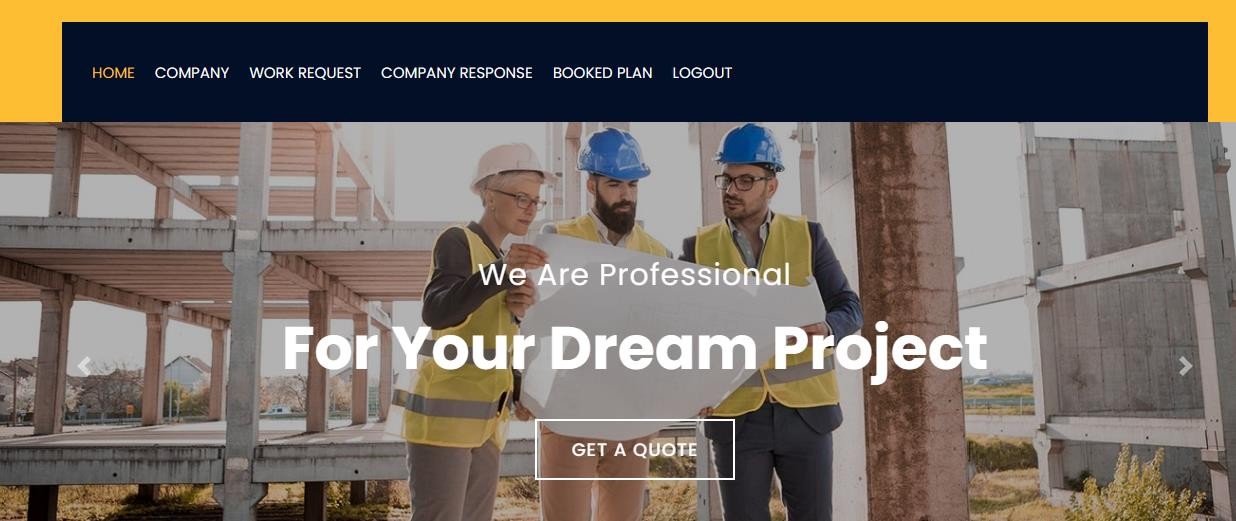
### Form Name: Signup page



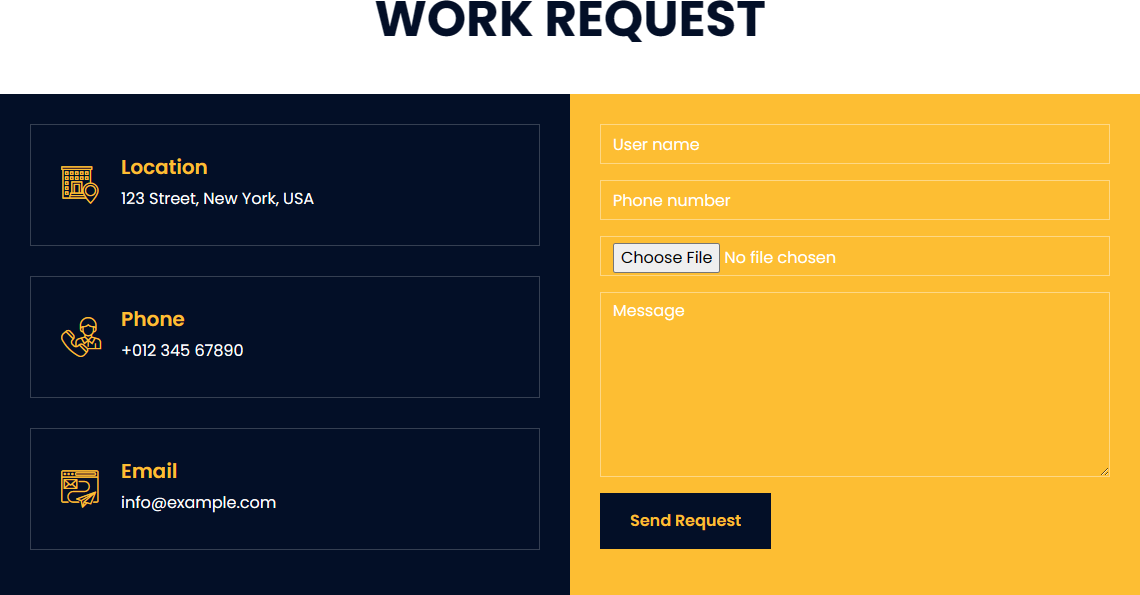
**Form Name: Login page**



**Form Name: Home page**



**Form Name: Work request page**



**4.4 DATABASE DESIGN**

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

* Data Integrity
* Data independence

### 4.4.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

**Relations, Domains & Attributes**

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

**Relationships**

* Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
* Entity Integrity enforces that no Primary Key can have null values.
* Referential Integrity enforces that no Primary Key can have null values.
* Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

### 4.4.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table.There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

### Normalize the data.

### Choose proper names for the tables and columns.

### Choose the proper name for the data.

### First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

### Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

### Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

### 4.4.3 Sanitization

The first lesson anyone learns when setting up a web-to-database—or anything-to-database gateway where untrusted user input is concerned—is to always, always sanitize every input. Failure to sanitize inputs can lead to attackers including SQL code in form inputs so they can do any number of interesting things, ranging from deleting information from a database to injecting information.

Injecting information into a database can not only cause records in the database to be incorrect, they can lead to further compromise. A classic example of this is when a web server allows dangerous and powerful functions to be executed, such as the php [system()](https://www.php.net/manual/en/function.system.php) call. Here, a data sanitization failure could lead to remote code execution on the server itself, an altogether much more serious problem.

Many database attacks have been carried out by exploiting poorly coded—or too relaxed—input sanitization rules to obtain remote command execution against the SQL server. However, this is by no means the only way to accomplish code execution [attacks using SQL injection](https://www.dnsstuff.com/sql-injection). Data sanitization is also important where system commands are executed with user-specified parameters, as in a router’s web interface, and this opens the door to appending partial commands into a database that will only be completed when predictable user input is received. These partial commands are often harder to detect.

There are many approaches to data sanitization, but most people start by blocking the most common characters, like “;” and “).” Developers writing their own input sanitization routines will usually [rely on regular expressions](https://www.w3schools.in/php-script/email-validation-php-regular-expression/) to help filter out unwanted inputs, and indeed this is one of the first skills a web developer is expected to learn.

### 4.4.4 Indexing

Indexing is a data structure technique which allows you to quickly retrieve records from a database file. An Index is a small table having only two columns. The first column comprises a copy of the primary or candidate key of a table. Its second column contains a set of [pointers](https://www.guru99.com/c-pointers.html) for holding the address of the disk block where that specific key value stored.

An index –

* Takes a search key as input
* Efficiently returns a collection of matching records.

### 4.5 TABLE DESIGN

User\_register

Primary key: user\_id

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| User\_Id | Int(11) | Primary key, auto increment | Unique id for each user |
| name | Varchar(40) | Not null | name of user |
| address | Varchar(40) | Not null | Address |
| email | Varchar(40) | Not null | Email id for login |
| password | Varchar(40) | Not Null | Password for login |
| Phone | Varchar(40) | Not Null | Contact number |

login\_tb: stores login credentials of contractors

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| login\_id | int(11) | primary key, auto increment | login id |
| email | varchar(40) | unique, not null | email id |
| password | varchar(8) | unique,not null | password |

table: company

Primary key: company\_id

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| company\_id | int(11) | foreign key | Unique for company |
| name | varchar(40) | not null | company name |
| email | varcha(40) | Not null | email |
| phone | varchar(40) | not null | phone number |
| address | varchar(40) | Not null | Address |
| image | Varchar(30) | Not null | Company logo |

Table : work\_agreement

Primary key: idwork\_agreement

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| idwork\_agreement | Int(11) | Primary key, auto increment | Id of work agreement |
| Company\_plan\_images | Varchar(25) | Not null | Plan image |
| Work\_start\_date | date | Not null | Start date |
| Work\_end\_date | date | Not null | End date |
| Total\_expense | int | not null | Total expense |

Table: company\_response

Primary key: id\_company\_response

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| id\_company\_response | int(11) | primary key | id of company response |
| Id\_work\_request | varchar(25) | Foreing key | References work request |
| Company\_id | varchar(25) | Foreing key | References company |
| Response\_date | date | not null | Response date |
| description | varchar(40) | Not null | description |
| amount | varchar(30) | not null | amount |
| status | varchar(30) | not null | status |

# CHAPTER 5

# SYSTEM TESTING

* 1. **INTRODUCTION**

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic

analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of

software while it is executing, to provide information such as execution traces, timing profiles,

and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

* Testing is a process of executing a program with the intent of finding an error.
* A good test case is one that has high possibility of finding an undiscovered error.
* A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

* For correctness
* For implementation efficiency
* For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

## TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

* Unit testing
* Integration Testing
* Data validation Testing
* Output Testing

### Unit Testing

Unit testing focuses verification effort on the smallest unit ofsoftware design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm’s execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code where removed and ensured that all modules are working, and gives the expected result.

### Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

### Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions thatwill fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

### Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm’s need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

* Input Screen Designs
* Output Screen Designs

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

**5.2.5 AUTOMATION TESTING**

A test case suite is executed using specialised automated testing software tools as part of the software testing technique known as automation testing. The test stages are meticulously carried out by a human performing manual testing while seated in front of a computer.Additionally, the automation testing software may generate thorough test reports,compare expected and actual findings, and enter test data into the System Under Test. Software test automation necessitates significant financial and material inputs. Repeated execution of the same test suite will be necessary during subsequent development cycles. This test suite can be recorded and replayed as needed using a test automation tool. No further human involvement is needed once the test suite has been automated.

**5.2.6 SELENIUM TESTING**

Selenium is a free and open-source tool for testing web applications across multiple browsers and operating systems. Selenium Test Scripts can be written in different programming languages, including Java, C#, JavaScript, Python, etc. Automation is performed using the Selenium framework is referred to as Selenium automation testing.

**Example:**

**Test Case 1**

**Code**

import selenium

from selenium import webdriver

from selenium.webdriver.chrome.service import Service chrome\_driver\_path = "path/to/chromedriver"

service = Service(executable\_path=chrome\_driver\_path) driver = webdriver.Chrome(service=service)

from selenium.webdriver.common.by import By from selenium.webdriver.common.keys import Keys

PATH= 'C:\\Users\\User\\Desktop\\Test\\chromedriver.exe' driver = webdriver.Chrome (PATH) driver.get("http://127.0.0.1:8000/login") driver.maximize\_window()

driver.find\_element(By.ID, "userid").send\_keys("Abin") driver.find\_element(By.ID, "password").send\_keys("Abin@123") driver.find\_element(By.ID, "btn").click() expectedurl="http://127.0.0.1:8000/userhome" currentur1=driver.current\_url

if expectedurl== currentur1: print("Test Case Passed")

else:

print("Failed")

### Screenshot

**Test Report**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case 1** | | | | | |
| **Project Name:** CONSTRUCTION MANAGEMENT SYSTEM | | | | | |
| **Login Test Case** | | | | | |
| **Test Case ID: Test\_1** | | | **Test Designed By:** Abin Sunil | | |
| **Test Priority(Low/Medium/High):** High | | | **Test Designed Date:** 10/5/2023 | | |
| **Module Name**:login | | | **Test Executed By :** Ajith G S | | |
| **Test Title:** User Login | | | **Test Execution Date:** 10/5/2023 | | |
| **Description:** User Login | | |  | | |
| **Pre-Condition:** User has valid username and password | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status(Pass/ Fail)** |
| 1 | Navigation  to Login Page |  | Dashboard shouldbe displayed | Login  page displayed | Pass |
| 2 | Provide  Valid Username | User Name: Abin | User should be able to Login | User Logged in and navigated to Dashboard | Pass |
| 3 | Provide  Valid  Password | Password: Abin@123 |
| 4 | Click on  Login |  |  |  |  |
|  | Button |  |  |  |
|  |  |  |  |  |  |
|  |  |  |
|  |  |  |
| **Post-Condition:** User successfully login into account | | | | | |

### Test Case 2: User registration

**Code**

import selenium

from selenium.webdriver.common.keys import Keys from selenium.webdriver.chrome.service import Service from selenium import webdriver

from selenium.webdriver.common.by import By from selenium.webdriver.support.ui import Select

PATH= 'C:\\Users\\User\\Desktop\\Test\\chromedriver.exe' driver = webdriver.Chrome (PATH)

username\_field = driver.find\_element(By.NAME, "email") username\_field.send\_keys("[jesnamol1112@gmail.com"](mailto:jesnamol1112@gmail.com))

password\_field = driver.find\_element(By.NAME, "password") password\_field.send\_keys("Jesna@123")

password\_field.send\_keys(Keys.RETURN) driver.get("http://127.0.0.1:8000/user\_registration/")

book\_cover = driver.find\_element(By.NAME, "username")

submit\_button = driver.find\_element(By.XPATH,'/html/body/div[3]/div/form/button') submit\_button.click()

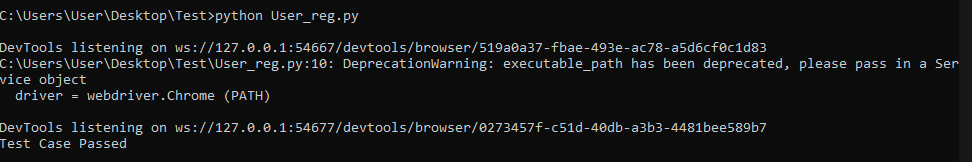
driver.get("http://127.0.0.1:8000/user\_registration/")

dashboard\_element = driver.find\_element(By.XPATH, "//h1[contains(text(), ')]") if dashboard\_element:

print("Test Case Passed") else:

print("Test failed.")

### Screenshot



**Test Report**

**Test Case 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Name:** CONSTRUCTION MANAGEMENT SYSTEM | | | | | |
| **User registration Test Case** | | | | | |
| **Test Case ID: Test\_2** | | | **Test Designed By:** Abin Sunil | | |
| **Test Priority(Low/Medium/High):** High | | | **Test Designed Date:** 10/5/2023 | | |
| **Module Name**: User registration | | | **Test Executed By:** Ajith G S | | |
| **Test Title:** User registration | | | **Test Execution Date:** 10/5/2023 | | |
| **Description:** Register new User | | |  | | |
| **Pre-Condition:** User has valid username and password | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status(Pass/ Fail)** |
| 1 | Navigation to Login Page |  | Dashboard shouldbe displayed | Login page displayed | Pass |
| 2 | Provide Valid  Username | User Name: Abin | User should be able to Login | Admin Logged in andnavigated to Dashboard | Pass |
| 3 | Provide  Valid  Password | Password: Abin@123 |
| 4 | Click on  user |  |  |  |  |
|  | registration |  |  |  |
|  | Button |  |  |  |
| 5 | Provide the registration page |  |  |  | Pass |
| 7 | Click submit button |  |
| **Post-Condition:** User successfully registered. | | | | | |

### Test Case 3: Upload requirements

**Code**

import selenium

from selenium.webdriver.common.keys import Keys from selenium.webdriver.chrome.service import Service from selenium import webdriver

from selenium.webdriver.common.by import By

PATH= 'C:\\Users\\User\\Desktop\\Test\\chromedriver.exe'

driver = webdriver.Chrome(service=service) driver.maximize\_window() driver.get("http://127.0.0.1:8000/login/login/")

username\_field = driver.find\_element(By.NAME, "username") username\_field.send\_keys("Abin")

password\_field = driver.find\_element(By.NAME, "password") password\_field.send\_keys("Abin@123")

password\_field.send\_keys(Keys.RETURN) dashboard\_element = driver.find\_element(By.XPATH, "//h")

driver.get("http://127.0.0.1:8000/1")

product\_addcart = driver.find\_element(By.XPATH, "//h")

if upload requirement: print("Test Case Passed")

else:

print("Test Failed") driver.quit()

### Screenshot

**Test Report**

|  |  |
| --- | --- |
| **Test Case 3** | |
| **Project Name:** CONSTRUCTION MANAGEMENT SYSTEM | |
| **View requirements** | |
| **Test Case ID: Test\_3** | **Test Designed By:** Abin Sunil |
| **Test Priority(Low/Medium/High):** High | **Test Designed Date:** 10/5/2023 |
| **Module Name**: View requirements | **Test Executed By:** Ajith G S |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Title:** View requirements | | | **Test Execution Date:** 10/5/2023 | | |
| **Description:** view the user requirements | | |  | | |
| **Pre-Condition:** User has valid username and password | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status(Pass/ Fail)** |
| 1 | Navigation to Login Page |  | Dashboard shouldbe displayed | Login page displayed | Pass |
| 2 | Provide  Valid Username | User Name: Abin | User should be able to Login | Admin Logged in andnavigated to Dashboard | Pass |
| 3 | Provide  Valid  Password | Password: Abin@123 |
| 4 | Click on  Login |  |  |  |  |
|  | Button |  |  |  |
| 5 | Click view requirements |  | View the requirement | View the requirements | Pass |
| **Post-Condition:** User successfully login into account and view the requirement | | | | | |

# CHAPTER 6

# IMPLEMENTATION

## INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

➢ Careful planning.

➢ Investigation of system and constraints.

➢ Design of methods to achieve the changeover.

## IMPLEMENTATION PROCEDURES

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, an evaluation of change over methods. The implementation process begins with preparing a plan for the implementation of the system. According to this, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

### User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

### Training on the Application Software

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of theirrole in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

For the easiness, the system is designed to the maximum level of simplicity and have necessary font size and understandable icons. This well designed menus forms and other GUI elements itself act as a nice way of interacting with the web site. Completeness of information and proper alerts and pop-up messages inform the user about the result of their actions .

### System Maintenance

Maintenance is the final and one of the non-skippable phase of system development. The maintenance phase of the software development lifecycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes". Normally maintenance engineers are responsible to maintain a software with their collaborative operations . Proper updates, debugging, code optimization, database back up and schema redefinition, SSL renewal , security updates etc. are some of the responsibilities of maintenance engineers. Now a days, more generally DevOps engineering has been termed instead of maintenance engineers. In the era of cloud and version control , most of the vendors and technology companies started their enterprises works on the basis of continuous development and integration . This is a part of AGILE methodology, a framework in software engineering principle. Continuous development and integration can be achieved through DevOps operations executed by devops engineers. They may use intermediate tools and servers to form deployment pipelines to propagate proper updates to the production environment from the development environment through systematic and automated procedures using Jenkin like servers and platforms. Version control servers like github, bitbucket etc. may be linked in this area so that maintenance time, test time , update time can also be reduced to maxium.

# CHAPTER 7

# CONCLUSION AND FUTURE SCOPE

## CONCLUSION

The analysis shows that our system “CONSTRUCTION MANAGEMENT SYSTEM” is better product for doing commercial work in a community. The existing manual system is very poor.

The new system that be introduced is effective, efficient and simple in managing construction fields. It is capable of keeping all the details and viewing them easily.

This system tested with sample data and output obtained in according the requirements .the performance of the system is evaluated , and is found to be much efficient than the existing system. Though it could not be claimed that is an ideal project, it will meet the primary requirements of the concern .so the project has to be proved by having modifications as and when necessity arises in due course. The system was implemented and tested with real and were.

* 1. **FUTURE SCOPE**
* A mobile application can be developed and integrated with this system for site inspection and capturing photographs of implemented proposals.
* Real time map for marking specific relief centers for convenient use.
* The user can directly designed their own house plans with the help of system. It may increase the customer satisfaction

.

# CHAPTER 8

# BIBLIOGRAPHY

### REFERENCES:

* + - Gary B. Shelly, Harry J. Rosenblatt, “System Analysis and Design”, 2009.
    - Roger S Pressman, “Software Engineering”, 1994.
    - Pankaj Jalote, “Software engineering: a precise approach”, 2006.
    - James lee and Brent ware Addison, “Open source web development with LAMP”, 2003
    - The Complete reference MySQL by Vikram Vaswani
    - CSS Cookbook by Christopher Schmitt

### WEBSITES:

* [www.w3schools.com](http://www.w3schools.com)
* [www.jquery.com](http://www.jquery.com)
* https://stackoverflow.com/
* [www.geeksforgeeks.com](http://www.geeksforgeeks.com)
* [www.tutorialspoint.org](http://www.tutorialspoint.org/)

# CHAPTER 9

# APPENDIX

## Sample Code

#### **9.1.1. LOGIN PAGE**

def login(request):

if request.method == "POST":

userid = request.POST['userid']

password = request.POST['password']

print("post username and password")

cursor = connection.cursor()

cursor.execute("select \* from company where name= '" + userid + "' AND password = '" + password + "' AND status = 'approved'")

company = cursor.fetchone()

print('level 2')

if company == None:

messages.error(request, 'Invalid Username Or Password!!')

cursor.execute("select \* from login where admin\_id = '" + userid + "' AND password = '" + password + "'")

admin = cursor.fetchone()

print('level 3')

if admin == None:

cursor.execute("select \* from user\_register where email= '" + userid + "' AND password = '" + password + "'")

user = cursor.fetchone()

u= list(user)

userid1 = u[1]

print('level 4-----------------------------------------------------',userid1)

if user == None:

print('level 5')

return redirect('login')

else:

request.session["userId"] = userid1

return redirect('userhome')

else:

request.session["adminId"] = userid

return redirect('adminhome')

else:

request.session["companyId"] = userid

print("company login")

return redirect("companyhome")

print('failed login')

return render(request, 'login.html')

#### **9.1.2. USER REGISTRATION**

def user\_signup(request):

if request.method == "POST":

name = request.POST['name']

address = request.POST['address']

email = request.POST['email']

phone = request.POST['phone']

password = request.POST['password']

cursor = connection.cursor()

cursor.execute("select \* from user\_register where name = '"+str(name)+"' AND email = '"+str(email)+"' AND password= '" + str(password) + "' ")

company = cursor.fetchone()

if company == None:

cursor.execute("insert into user\_register values(null,'" + str(name) + "','" + str(address) + "','" + str(email) + "','" + str(phone) + "','" + str(password) + "')")

request.session["userId"] = name

return redirect("userhome")

else:

return HttpResponse("<script>alert('Please enter a strong password');window.location='../usersignup';</script>")

else:

return render(request,"user/user\_register.html")

## Screen Shots