



Nexus – Application For Searching Contractor And Worker In Cities

POTNURU UMA MAHESWARA RAO

Under the Esteemed Guidance of

K. RATNAREDDY

ASSISTANT PROFESSOR

ABSTRACT

Objective of the project is to provide a PC Grid solution focused on the workers of unorganized sector includes mason, carpenters, plumber, painter etc. PC Grid is a web solution through which workers register themselves for a specific skill.

Using this information, users, contractors or organized sectors can select the workers as per their need. At the time of worker selection, company or user can view the worker skill, references given by those who have taken their service in the past, work location and availability of particular workers.

System sends SMS to selected workers regarding work and customer details. A worker confirms either through phone or this system and either fixed up meeting or work start date. Organized sector user or general public can rate worker skill, charges, particularity about time, dedication, behavior, habits etc through this system. And also User can select Contractors in this site. So Contractor also Registered with in this site, After that Contractor also search or select the user with specific skills. Users can put their demands regarding particular skill workers along with project location, and project details.

CHAPTER 1 –INTRODUCTION

Social exclusion is a concept that captures the root causes of social and economic disadvantage in contemporary societies. The concept of social exclusion was introduced in France in 1974 to explain the deprivation experienced by those who were unable to access social protection which, at the time, was available only to employed persons and their families (Peace 2001). Estimates indicated at the time, that approximately 10% of the population in France was excluded. These groups included homeless people, drug addicts, handicapped people, single parents, abused children, and households experiencing multiple problems (Peace 2001; Sen 2000). These aforementioned groups often faced barriers to employment and also experienced isolation from family and, as a result, they also lacked access to state support. The outcome was extreme deprivation, poverty, and social isolation. The fact that so many were afflicted, presented problems of social cohesion and economic prosperity for society as a whole.

1.1 Objectives

System sends SMS to selected workers regarding work and customer details. A worker confirms either through phone or this system and either fixed up meeting or work start date. Organized sector user or general public can rate worker skill, charges, particularity about time, dedication, behavior, habits etc through this system.

1.2 System Specifications

Hardware Requirements:-

Processor : Intel 3

Installed memory (RAM) : 4 GB

Hard Disk : 500 GB

Operating System : Windows 7,8,10 - 64 bit

Software Requirements: -

Operating System : Windows OS

Front-End : HTML5, CSS3, Bootstrap

Back-End : PHP 7.4, MYSQL

Control End : Angular Java Script

PHP TOOLS :

xampp-win64-8.1

CHAPTER 2 – LITERATURE REVIEW

The literature in this context refers to published scholarly work. For instance, scientific scholarly literature includes journal articles, conference proceedings, technical reports and books. A “literature review” reviews the scholarly literature on a specific topic by summarizing and analyzing published work on that topic. A literature review has several purposes: to evaluate the state of research on a topic to familiarize readers and students with what has already been done in the field to suggest future research directions or gaps in knowledge Green, Johnson and Adams describe a literature review as a type of research article published in a professional peer-reviewed journal. The purpose of a literature review is to objectively report the current knowledge on a topic and base this summary on previously published research. A literature review provides the reader with a comprehensive overview and helps place that information into perspective. 1 Most research articles include a section that reviews previous work, but this is not as comprehensive as an article-length literature review. A literature review does not include new, unpublished research results.

In an article database, search for the keywords “literature review” or “review” (in title or topic) in addition to your topic keywords Some article databases have “reviews” coded as a separate article format. For example, Web of Science□ lists “reviews” under document type. Do a topic search, then click this box to refine to only review articles. In some databases (such as BIOSIS) this is called literature type. Pubmed includes Review as a filter under “type of article”. In PyscINFO, include “literature review” as a methodology. You will still need to look at the papers you find to determine if they are truly literature reviews. Access□ the full-text of the paper (using the orange UC eLinks button) and read the first page or two. If it reports original research results, it’s not a review article. Most reviews explicitly state that they are reviewing the literature in the abstract or first paragraphs, or use the phrase “We review recent studies” or similar. Review articles also typically cite dozens of

other papers in their quest to be comprehensive, so their bibliography sections are longer than usual. Don't forget to search review journals – try Annual Reviews (<http://www.annualreviews.org>). □ Remember not all specific topics have review papers already published on them – you may need to □ broaden your keywords (by using fewer or less specific terms) to find a review of your subject.

Existing Solution:

The users or companies have to trace the information about the workers and contractors manually by using news papers or advertisements or through Google search given by the corresponding people. This process has to be done separately for specific skilled worker. A particular skilled worker who is efficient and dedicated is difficult to find. To know worker skills, semi skills and performance is difficult to trace. Contractor also has difficulty to join a worker in their organization without knowing anything about the worker.

Proposed Solution:

The proposed system provides an application where all types of workers, organizational sectors, companies can use the system to satisfy their requirements.

CHAPTER 3 OVERALL DESCRIPTION OF THE PROPOSED SYSTEM

3.1 Module Description

By using contact information of worker, general public can communicate with workers from home only. Communication is possible among the workers, general public, contractors, and administrator.

3.2 System Features

In the life of the software development, problem analysis provides a base for design and development phase. The problem is analyzed so that sufficient matter is provided to design a new system. Large problems are sub-divided into smaller once to make them understandable and easy for finding solutions. Same in this project all the task are sub-divided and categorized.

System Modules:

- User
- Admin
- Worker/Contractor

Module Description:

Users or Companies

Users can find worker various skills and semi skills within the system. Users can easily search for contractors also. Users and contractors can give the feedback of the worker performance. Feedback about the workers is useful for the remaining users and contractor who are searching for a worker. By using contact information of worker, general public can communicate with workers from home only. Communication is possible among the workers, general public, contractors, and administrator.

Workers and Contractors

Contractors can find a worth full workers based on the feedback given by public users about their work. contractor are also give the feedback about worker. Workers can present their multi skills information within site. From any where workers and users can access the system. No more bargaining with the customers, because charges are fixed. They can easily register in the site with their skills and semi skills with a free of cost. No advertisement charges for the workers. Communication is possible among the workers, general public, contractors, and administrator.

MODULES

The modules designed for the system are Registration, Administrator, General Public or Users, Workers, Contractor and feedback(rating bar)

MODULE DESCRIPTION

Registration

The Workers, public users, company sectors or contractors will undergo the registration procedure by specifying the identity proof, skills, education, experience, and address-proof and obtain a username and password, which is used for further transactions.

Worker

The worker is registered in this site itself. Worker can able to update the details, experience, location, skills which is been added during the registration procedure. The worker can communicate with other users like Admin, contractors, general public.

Feedback

User and Contractor can give the feedback of the work performance about worker. Feedback about the workers is usefull for the remaining users and contractors Who are searching for the workers and also it help for worker to increase its job performace.

General Public or User

The User can look into the site to get the Workers and Contractor for the house-hold work like mason, painter, Carpentry, Plumbing,etc. The General Public should able to see the Workers details like Skill set, Semi-Skill set, Contact details, etc.At the worker selection time user must be registered.Id proof mandatory. They should also have facility to see the feedback given by Customers and also give the feedback about worker who take service in past etc. to get knowledge about the Workers working style.User can see the Contractor details in this site and select also.At the time of selection,user select the worker that time System send the SMS to related worker automatically regards user details.

Administrator

this module provides administrator related functionality. Administrator manages all information and has access rights to verify,update, delete and view the data related to workers and contractors etc.. The admin should able to communicate with others (the consumers of the site) through, email, and also should able to verify userdetails .

Contractor

Contractors is registered in this site itself. contractor can find a worthfull workers based on feedback given by public users Or contractors about Their work. worker can present their multi skills information within site .Contractor sector or companies will undergo the registration procedure by specifying identity proof, skills ,experience and address proof and obtain a username and password which is use for further transaction. Purpose of registration is give the priority to contractor

by administrator and it is helpful for contractors. User can also select best contractor in this site so must contractor registered with details. contractor select the worker that time System send the SMS to related worker automatically regards contractor details.

CHAPTER 4 – DESIGN

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

4.1 UML Diagrams:

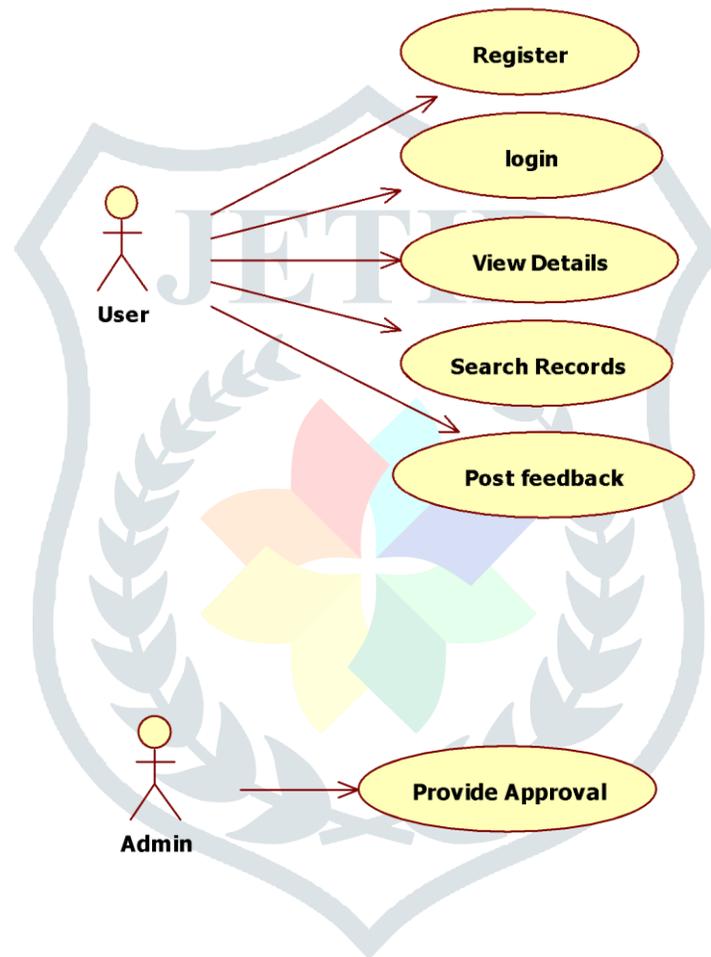
UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

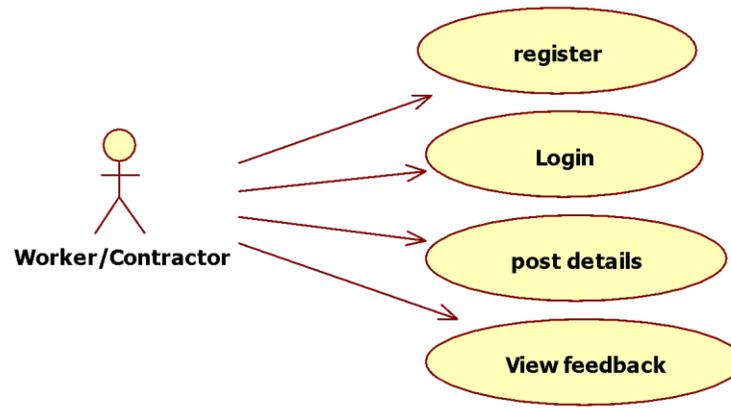
There are various kinds of methods in software design:

- Use case Diagram
- Sequence Diagram
- Collaboration Diagram

4.1.1 Usecase Diagrams:

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor. Use case diagram can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can't do.





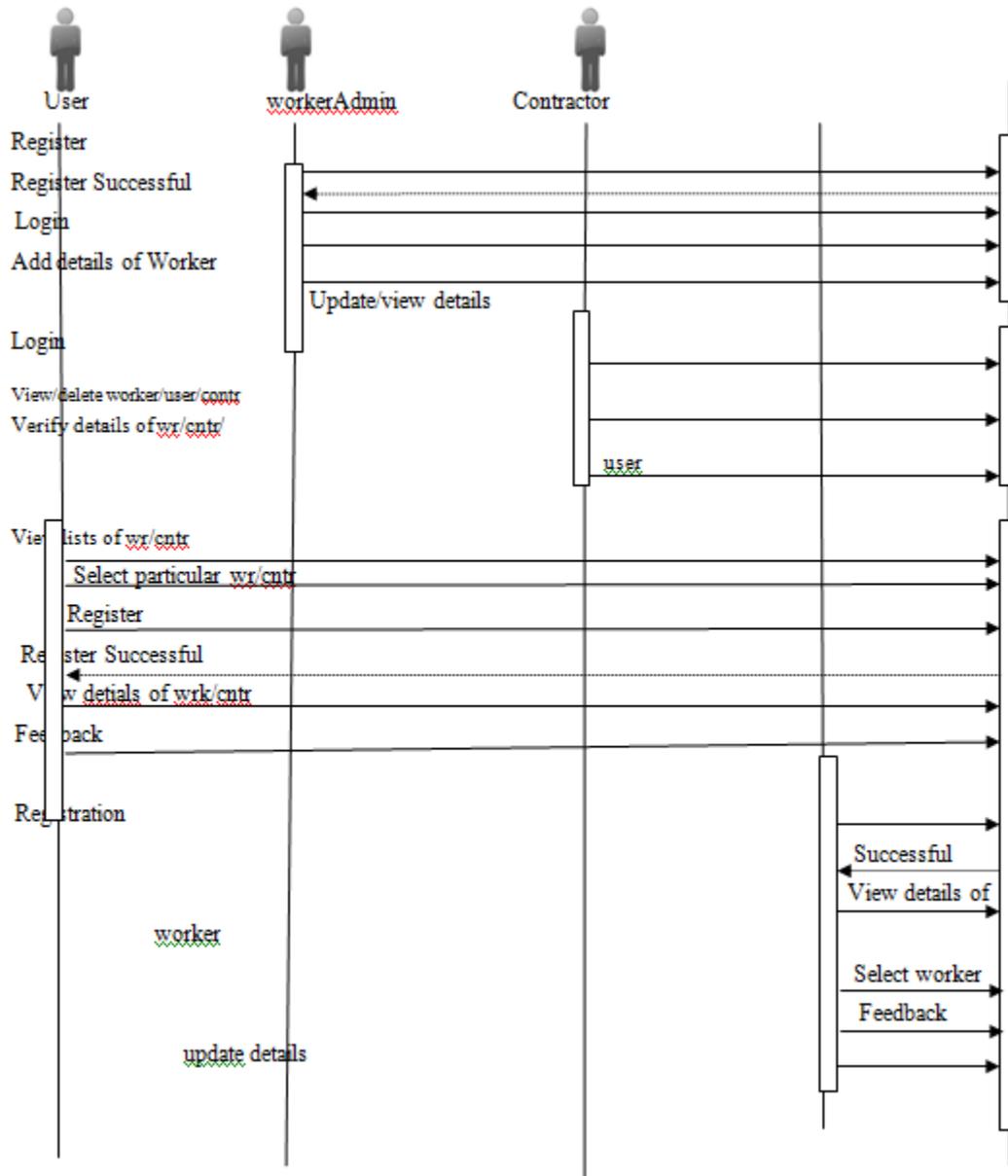
Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

- The purpose is to show the interactions between the use case and actor.
- To represent the system requirements from user's perspective.
- An actor could be the end-user of the system or an external system

4.1.2 Sequence Diagram:

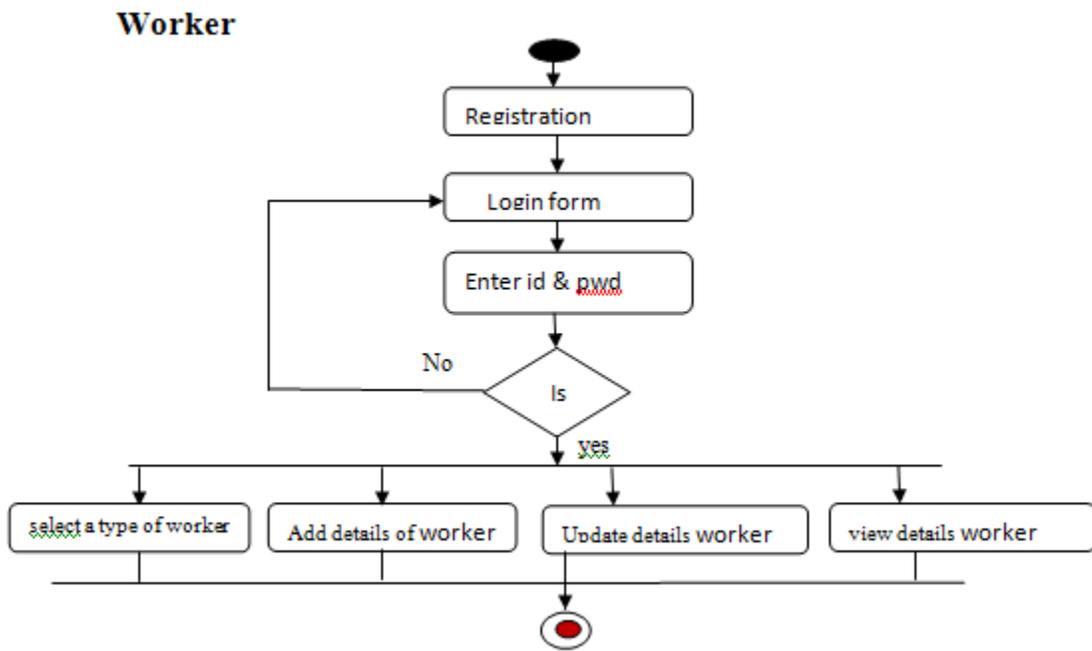
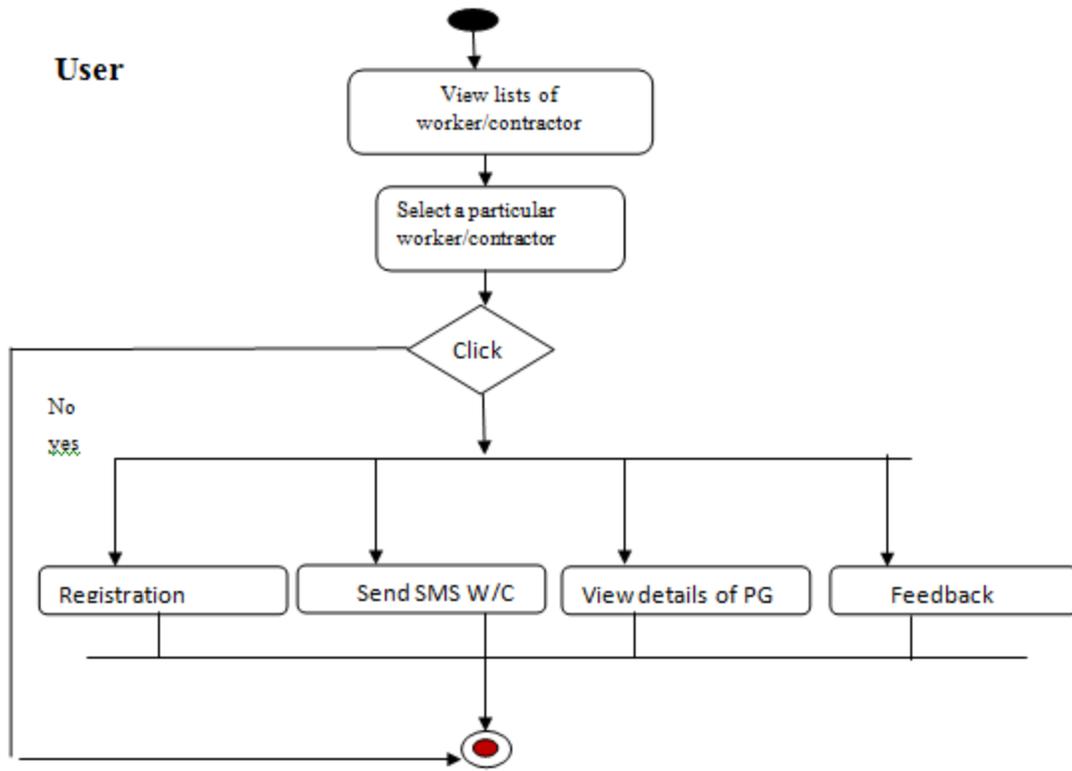
Sequence diagram and collaboration diagram are called INTERACTION DIAGRAMS. An interaction diagram shows an interaction, consisting of set of objects and their relationship including the messages that may be dispatched among them.

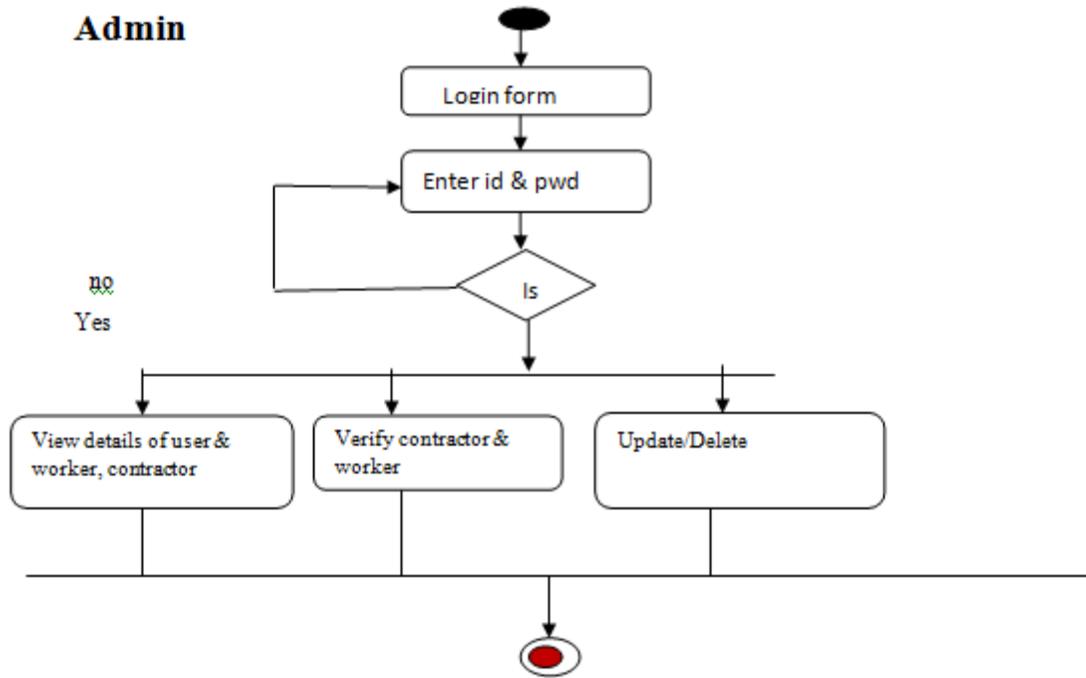
A sequence diagram is an introduction that empathizes the time ordering of messages. Graphically a sequence diagram is a table that shows objects arranged along the X-axis and messages ordered in increasing time along the Y-axis.



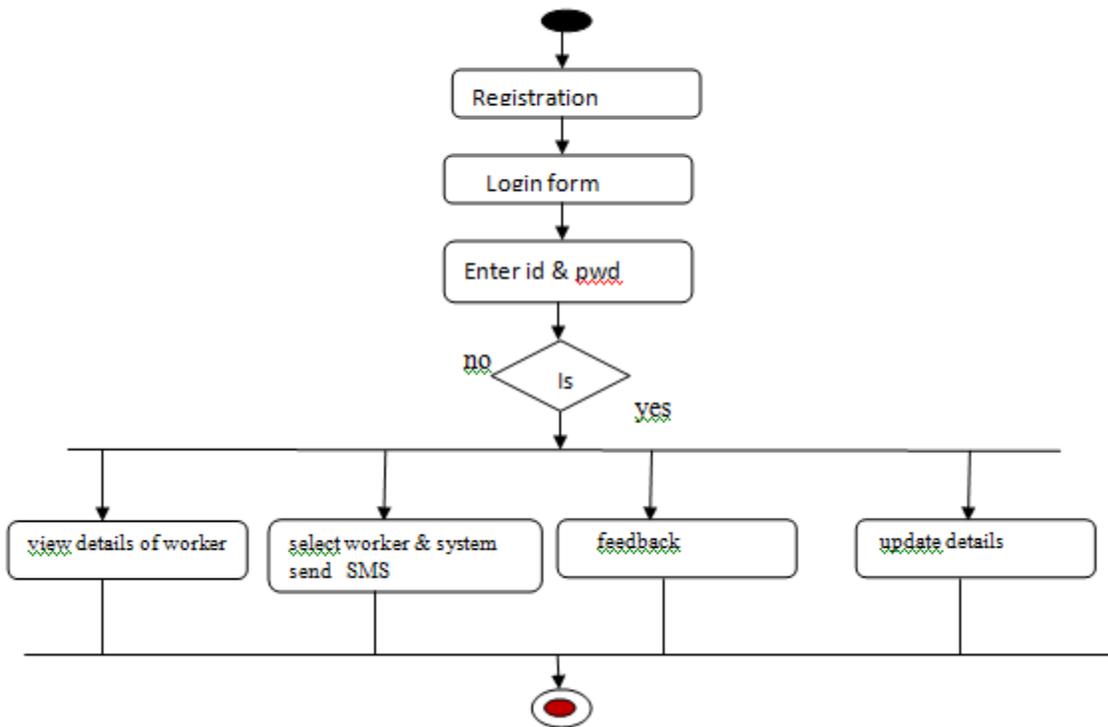
4.1.3 Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another.



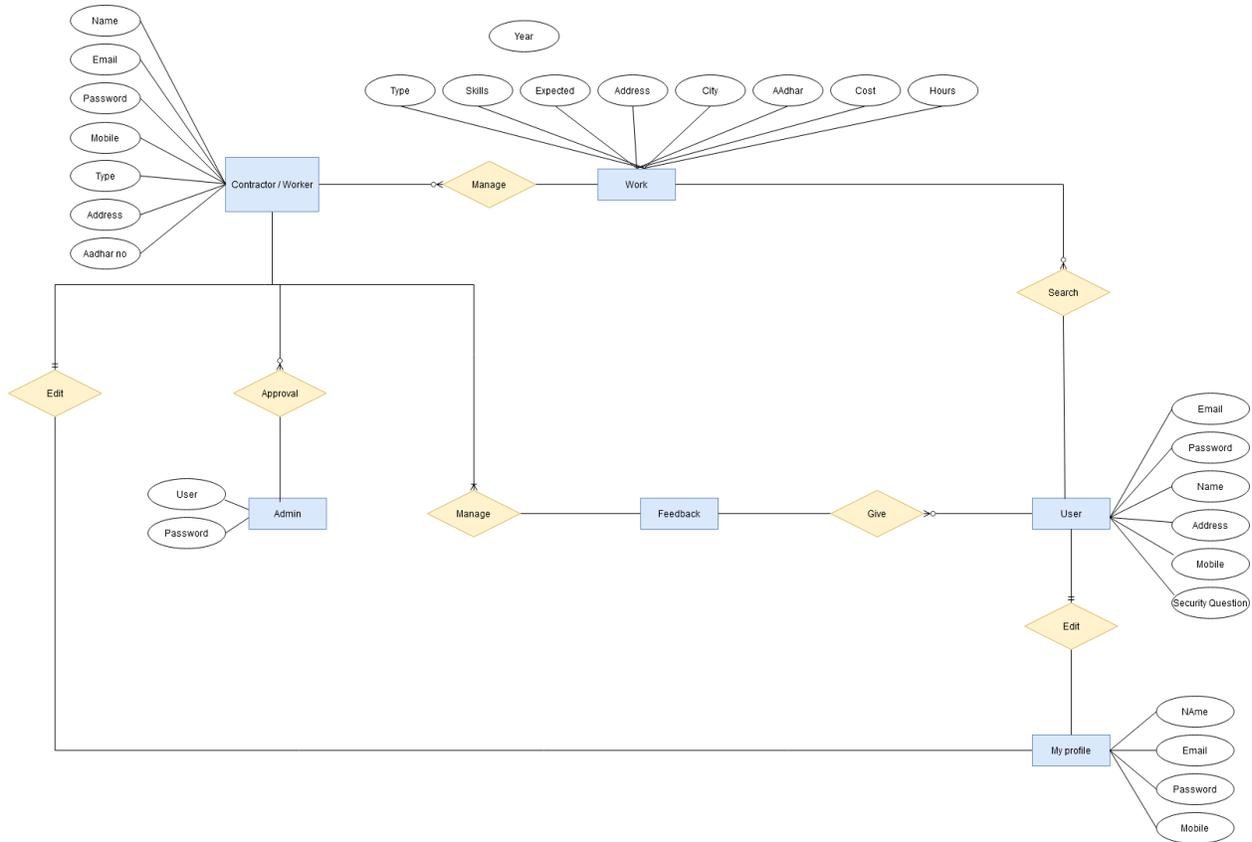


Contractor

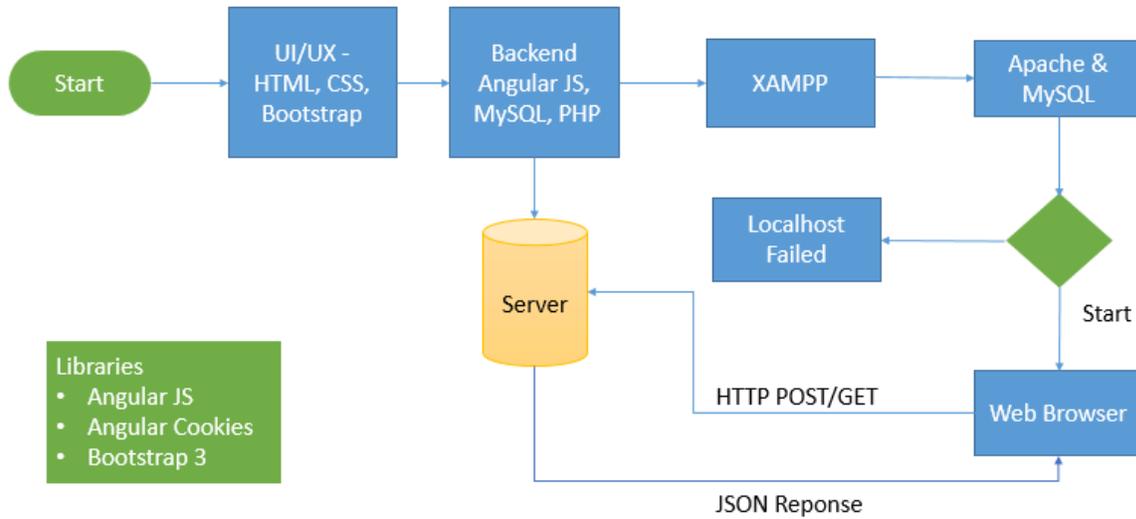


Entity Diagram

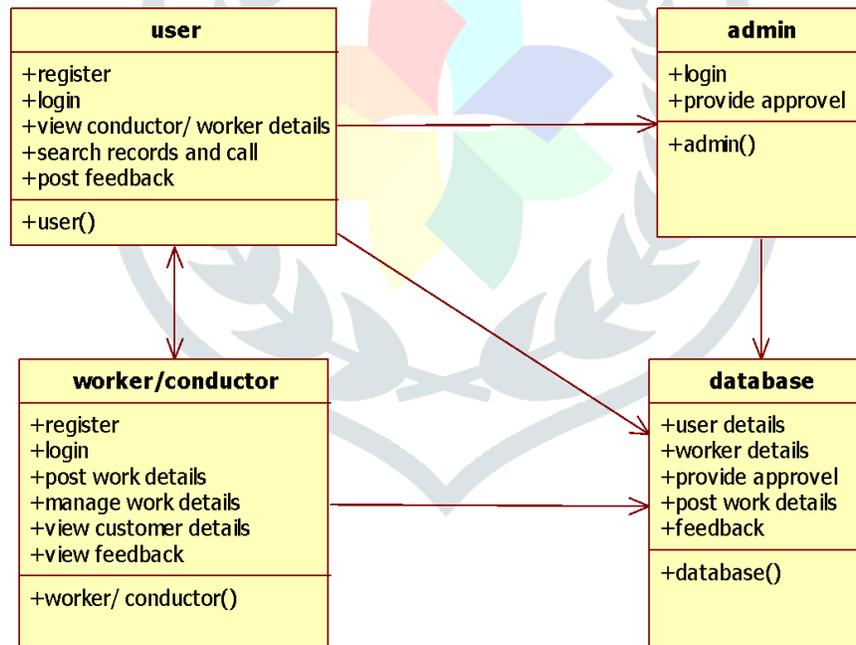
An entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.



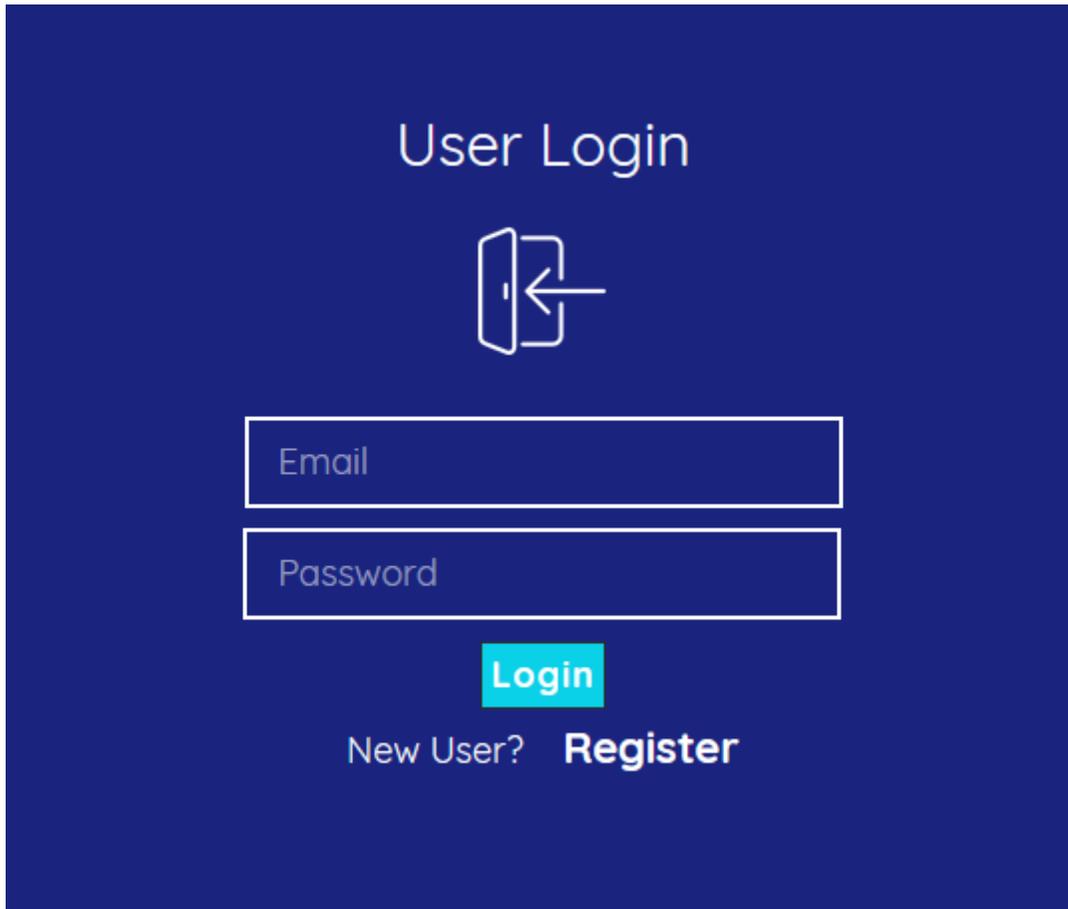
WORKFLOW DIAGRAM



Class DIAGRAM



CHAPTER 5 - OUTPUT SCREENSHOTS



☰ WeFix



Worker Details



Contractor Details



My Register



My Profile



View Work / Contract Details

Search

Worker ID: **30**

Work: **Carpenter**

Work Type: **Worker**

Skills: Furniture making, mathematical calculation

[Register](#)



The logo for WeFix, featuring a white hamburger menu icon (three horizontal lines) to the left of the text "WeFix" in a white sans-serif font, all on a dark blue background.

View Work / Contract Details

Search

Worker ID: **29**

Work: **Painter**

Work Type: **Con**

Skills: Calculating Quantities of Wallpaper

[Register](#)



CHAPTER 6 – IMPLEMENTATION DETAILS

6.1 Introduction to Html Framework

Hyper Text Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology used to create web pages, as well as to create user interfaces for mobile and web applications. Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

HTML elements form the building blocks of HTML pages. HTML allows images and other objects to be embedded and it can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `` and `<input />` introduce content into the page directly. Others such as `<p>...</p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages. HTML markup can also refer the browser to Cascading Style Sheets (CSS) to define the look and layout of text and other material.

HyperText Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web.[1] Web browsers receive HTML documents from a webserver or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `` and `<input />` introduce content into the page directly. Others such as `<p>...</p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

HTML documents imply a structure of nested HTML elements. These are indicated in the document by HTML tags, enclosed in angle brackets thus: `<p>`[61]

In the simple, general case, the extent of an element is indicated by a pair of tags: a "start tag" `<p>` and "end tag" `</p>`. The text content of the element, if any, is placed between these tags.

Tags may also enclose further tag markup between the start and end, including a mixture of tags and text. This indicates further (nested) elements, as children of the parent element.

The start tag may also include attributes within the tag. These indicate other information, such as identifiers for sections within the document, identifiers used to bind style information to the presentation of the document, and for some tags such as the `` used to embed images, the reference to the image resource.

Some elements, such as the line break `
`, do not permit any embedded content, either text or further tags. These require only a single empty tag (akin to a start tag) and do not use an end tag.

Many tags, particularly the closing end tag for the very commonly used paragraph element `<p>`, are optional. An HTML browser or other agent can infer the closure for the end of an element from the context and the structural rules defined by the HTML standard. These rules are complex and not widely understood by most HTML coders.

The general form of an HTML element is therefore: `<tag attribute1="value1" attribute2="value2">"content"</tag>`. Some HTML elements are defined as empty elements and take the form `<tag attribute1="value1" attribute2="value2">`. Empty elements may enclose no content, for instance, the `
` tag or the inline `` tag. The name of an HTML element is the name used in the tags. Note that the end tag's name is preceded by a slash character, `/`, and that in empty elements the end tag is neither required nor allowed. If attributes are not mentioned, default values are used in each case.

6.2 Cascading Style Sheets (CSS)

CSS is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers. CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file. For each matching HTML element, it provides a list of formatting instructions. For example, a CSS rule might specify that "all heading 1 elements should be bold", leaving pure semantic HTML markup that asserts "this text is a level 1 heading" without formatting code such as a `<bold>` tag indicating how such text should be displayed.

This separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to display the web page differently depending on the screen size or device on which it is being viewed. Although the author of a web page typically links to a CSS file within the markup file, readers can specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author has specified. If the author or the reader did not link the document to a style sheet, the default style of the browser will be applied. Another advantage of CSS is that aesthetic changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in one file, rather than by a laborious (and thus expensive) process of crawling over every document line by line, changing markup.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable.

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language.[1] Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech,

or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.[2]

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts.[3] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified.

Changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in the CSS file they use, rather than by changing markup in the documents.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

6.3 MYSQL Server

MySQL is an open-source relational database management system (RDBMS);[6] in July 2013, it was the world's second most widely used RDBMS, and the most widely used open-source client–server model RDBMS. It is named after co-founder Michael Widenius's daughter, My. The SQL acronym stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-

profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

MySQL (officially pronounced as /maɪ ˌɛskjuːˈɛl/ "My S-Q-L",[6]) is an open-source relational database management system (RDBMS).[7] Its name is a combination of "My", the name of co-founder Michael Widenius' daughter,[8] and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.[9] For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google[10][11] (though not for searches), Facebook,[12][13][14] Twitter,[15] Flickr,[16] and YouTube.

6.4 PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive backronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management system and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification.

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994,[4] the PHP reference implementation is now produced by The PHP Development Team.[5] PHP originally stood for Personal Home Page,[4] but it now stands for the recursive acronym PHP: Hypertext Preprocessor.[6]

PHP code may be embedded into HTML or HTML5 code, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.[7]

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.[8]

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification.

6.5 ANGULAR JAVA SCRIPT

AngularJS (commonly referred to as "Angular" or "Angular.js") is an open-source web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. It aims to

simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel(MVVM) architectures, along with components commonly used in rich Internet applications.

The AngularJS framework works by first reading the HTML page, which has embedded into it additional custom tag attributes. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

According to JavaScript analytics service Libscore, AngularJS is used on the websites of Wolfram Alpha, NBC, Walgreens, Intel, Sprint, ABC News, and approximately 8,400 other sites out of 1 million tested in July 2015.

AngularJS is the frontend part of the MEAN stack, consisting of MongoDB database, Express.js web application server framework, Angular.js itself, and Node.js runtime environment. AngularJS (commonly referred to as "Angular.js" or "AngularJS 1.X") is a JavaScript-based open-source front-end web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. The JavaScript components complement Apache Cordova, the framework used for developing cross-platform mobile apps. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications. In 2014, the original AngularJS team began working on Angular (Application Platform).

The AngularJS framework works by first reading the HTML page, which has embedded into it additional custom tag attributes. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

According to JavaScript analytics service Libscore, AngularJS is used on the websites of Wolfram Alpha, NBC, Walgreens, Intel, Sprint, ABC News, and approximately 12,000 other sites out of 1 million tested in October 2016.[3] AngularJS is the 6th most starred project of all time on GitHub.[4] AngularJS is the frontend part of the MEAN stack, consisting of MongoDB database, Express.js web application server framework, Angular.js itself, and Node.js server runtime environment.

AngularJS is built on the belief that declarative programming should be used to create user interfaces and connect software components, while imperative programming is better suited to defining an application's business logic.[5] The framework adapts and extends traditional HTML to present dynamic content through two-way data-binding that allows for the automatic synchronization of models and views. As a result, AngularJS de-emphasizes explicit DOM manipulation with the goal of improving testability and performance.

AngularJS's design goals include: to decouple DOM manipulation from application logic. The difficulty of this is dramatically affected by the way the code is structured. To decouple the client side of an application from the server side. This allows development work to progress in parallel, and allows for reuse of both sides. To provide structure for the journey of building an application: from designing the UI, through writing the business logic, to testing.

Angular implements the MVC pattern to separate presentation, data, and logic components.[6] Using dependency injection, Angular brings traditionally server-side services, such as view-dependent controllers, to client-side web applications. Consequently, much of the burden on the server can be reduced.

CHAPTER 7- SYSTEM STUDY

7.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden

to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY
- ◆

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

CHAPTER 8-TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with

it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

8.1 Non Functional Requirements

Non-functional requirements are the quality requirements that stipulate how well software does what it has to do. These are Quality attributes of any system; these can be seen at the execution of the system and they can also be the part of the system architecture.

8.2 Accuracy:

The system will be accurate and reliable based on the design architecture. If there is any problem in the accuracy then the system will provide alternative ways to solve the problem.

8.3 Usability:

The proposed system will be simple and easy to use by the users. The users will comfort in order to communicate with the system. The user will be provided with an easy interface of the system.

8.4 Accessibility:

The system will be accessible through internet and there should be no any known problem.

8.5 Performance:

The system performance will be at its best when performing the functionality of the system.

8.6 Reliability:

The proposed system will be reliable in all circumstances and if there is any problem that will be affectively handle in the design.

8.7 Security:

The proposed system will be highly secured; every user will be required registration and username/password to use the system. The system will do the proper authorization and authentication of the users based on their types and their requirements. The proposed system will be designed persistently to avoid any misuse of the application.

CHAPTER 9-SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TESTCASE

S.NO	SCENARIO	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1	User register	Email and Password	All the user details register successfully	Register successfully Or Register unsuccessfully
2	User Login	Email and Password	If correct directed to home page otherwise show "Invalid Login"	Login successfully or Login unsuccessfully
3	View Contractor and Worker Details	User check and view all Contractor and Worker Details	User view Contractor and Worker Details	View division Contractor and Worker Details
4	Search Records and Call	User can check and view all Records and Call details	view Records and Call details	View all Records and Call details

5	Post Feedback	User will enter all post Feedback	If all the post Feedback “Created Successfully”	Created successfully or created unsuccessfully
6	Admin login Details	Admin enter login details	Login successfully or if incorrect login details “Login successfully”	Login successfully or Login unsuccessfully
7	Approve Contractor and Worker	Admin approve the Contractor and Worker	If all the Contractor and Worker correct admin approve the “approved successfully”	Approve successfully or Not approve
8	Worker register	Username and Password	Worker details register successfully	Register successfully Or Register unsuccessfully
9	Worker Login	Email and Password	If correct directed to home page otherwise show “Login Successfully”	Login successfully or Login unsuccessfully
10	Post Work Details	Worker will enter all post Work Details	If all the post Work Details “Created Successfully”	Created successfully or created unsuccessfully
11	Manage Work Details	Worker can edit and delete the Work Details	If any changes edit updated successfully or delete successfully	Updated successfully or unsuccessfully Delete successfully or unsuccessfully
12	Manage Customer	Worker can edit and delete the Customer	If any changes edit updated successfully or delete successfully	Updated successfully or unsuccessfully Delete successfully or unsuccessfully

13	View Feedback	Worker check and view all Feedback	Worker Feedback	view	View division Feedback
----	---------------	------------------------------------	-----------------	------	------------------------

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

9.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page

9.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

9.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

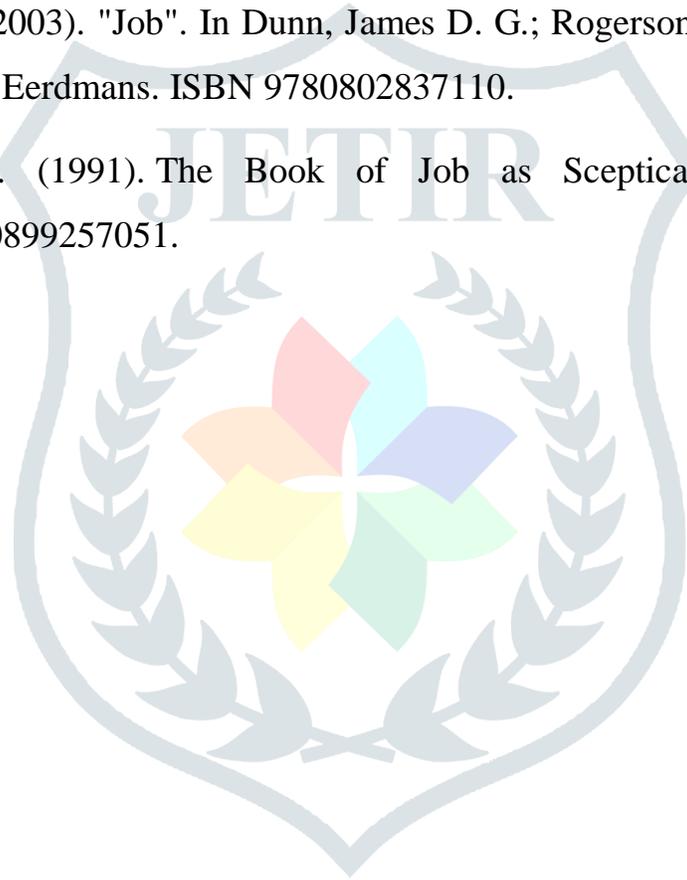
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 10- CONCLUSIONS

Using this information, users, contractors or organized sectors can select the workers as per their need. At the time of worker selection, company or user can view the worker skill, references given by those who have taken their service in the past, work location and availability of particular workers. Here we enhanced sms system to server. Once we registered to this system you can get message from server.

CHAPTER 11- REFERENCES

- Allen, J. (2008). "Job III: History of Interpretation". In Longman, Tremper; Enns, Peter. Dictionary of the Old Testament: Wisdom, Poetry & Writings. InterVarsity Press. ISBN 9780830817832.
- Brueggemann, Walter (2002). Reverberations of faith: a theological handbook of Old Testament themes. Westminster John Knox. ISBN 9780664222314.
- Bullock, C. Hassell (2007). An Introduction to the Old Testament Poetic Books. Moody Publishers. ISBN 9781575674506.
- Dell, Katharine J. (2003). "Job". In Dunn, James D. G.; Rogerson, John William. Eerdmans Bible Commentary. Eerdmans. ISBN 9780802837110.
- Dell, Katherine J. (1991). The Book of Job as Sceptical Literature. Walter de Gruyter. ISBN 9780899257051.



CHAPTER 12- CODING

12.1 INDEX.HTML

```
<!DOCTYPE html>
```

```
<html ng-app="myapp">
```

```
<head>
```

```
<!-- Basic -->
```

```
<meta charset="utf-8">

<meta name="keywords" content="HTML5 Template" />

<meta name="description" >

<meta name="author" content="pixelgeeklab.com">

<!-- Mobile Metas -->

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- Web Fonts -->

<link href='css/css.css' rel='stylesheet' type='text/css'>

<link rel="stylesheet" href="fonts/font.css">

<!-- Vendor CSS -->

<link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.css">

<link rel="stylesheet" href="vendor/owlcarousel/owl.carousel.css" media="screen">

<link rel="stylesheet" href="vendor/owlcarousel/owl.theme.css" media="screen">

<link href="vendor/owl-carousel/owl.transitions.html" rel="stylesheet" media="screen">

<link rel="stylesheet" href="vendor/flexslider/flexslider.css" media="screen">

<link rel="stylesheet" href="vendor/chosen/chosen.css" media="screen">

<link rel="stylesheet" href="vendor/magnific-popup/magnific-popup.css" media="screen">

<!-- Theme CSS -->

<link rel="stylesheet" href="css/theme.css">

<link rel="stylesheet" href="css/theme-animate.css">

<!-- Style Switcher-->

<!-- Head libs -->
```

```
<script src="vendor/modernizr/modernizr.js"></script>
```

```
<!--[if IE]>
```

```
  <link rel="stylesheet" href="css/ie.css">
```

```
<![endif]-->
```

```
<!--[if lte IE 8]>
```

```
  <script src="vendor/respond/respond.js"></script>
```

```
  <script src="vendor/excanvas/excanvas.js"></script>
```

```
<![endif]-->
```

```
<style>
```

```
.container {
```

```
  background-color: #1A247E;
```

```
  position:relative;
```

```
  margin-top:100px;
```

```
}
```

```
.row{
```

```
  margin-left:50px;
```

```
  margin-right:50px;
```

```
  text-align:center;
```

```
}
```

```
.login p{
```

```
  position:relative;
```

```
  text-align:center;
```

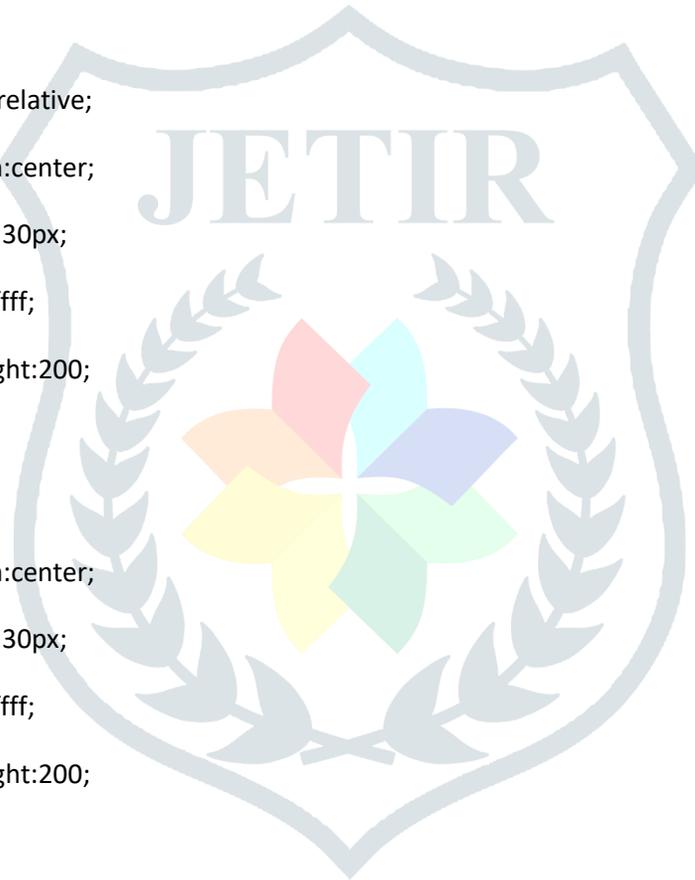
```
  font-size:30px;
```

```
  color:#ffffff;
```

```
}
```



```
p{
position:relative;
text-align:center;
font-family:'Quicksand';
font-size:35px;
color:#ffffff;
font-weight:200;
}
h2{
position:relative;
text-align:center;
font-size:30px;
color:#ffffff;
font-weight:200;
}
h3{
text-align:center;
font-size:30px;
color:#ffffff;
font-weight:200;
}
img{
align:middle;
}
```



```
</style>
```

```
<!-- script back button -->
```

```
<script src="cordova.js"></script>
```

```
<script>
function onLoad()
{
    document.addEventListener("deviceready", deviceReady, false);
}
function deviceReady()
{
    document.addEventListener("backbutton", backButtonCallback, false);
}
function backButtonCallback()
{
    navigator.app.exitApp();
}
</script>
</head>

<!-- angular js -->
<script src="js/angular-1.3.js"></script>
<script src="js/angular_cookies.js"></script>

<body >

    <div class="container">

        <h2>WeFix</h2>

        <div class="row">

            <div class="col-md-12" >

            </div>

            <div class="login" style="margin-left:0px;">
```

```
<div class="col-md-4" >
```

```
    <a href="admin_login.html"> <button style="background-color:white;font-weight:bold;color:#000;letter-spacing:1px">Admin Login</button> </a> <br><br>
```

```
    <a href="user_login.html"> <button style="background-color:white;font-weight:bold;color:#000;letter-spacing:1px">User Login</button> </a> <br><br>
```

```
    <a href="worker_login.html"> <button style="background-color:white;font-weight:bold;color:#000;letter-spacing:1px">Worker/Co Login</button> </a> <br><br>
```

```
    <a href="worker_register.html"><button href="user_register.html" style="background-color:white;font-weight:bold;color:#000;letter-spacing:1px">Worker Register</button> </a><br><br>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<script src="vendor/jquery/jquery.js"></script>
```

```
<script src="vendor/bootstrap/bootstrap.js"></script>
```

```
<script src="vendor/jquery.validation/jquery.validation.js"></script>
```

```
<script src="vendor/owlcarousel/owl.carousel.js"></script>
```

```
<script src="vendor/flexslider/jquery.flexslider-min.js"></script>
```

```
<script src="vendor/countdown/countdown.min.js"></script>
```

```
<script src="vendor/chosen/chosen.jquery.min.js"></script>
```

```
<script src="vendor/pricefilter/jquery.pricefilter.js"></script>
```

```
<script src="vendor/masonry/imagesloaded.pkgd.min.js"></script>
```

```
<script src="vendor/masonry/masonry.pkgd.min.js"></script>
```

```
<script src="vendor/uikit/uikit.js"></script>
```

```
<script src="vendor/magnific-popup/jquery.magnific-popup.js"></script>
```

```
<!-- Theme Base, Components and Settings -->
```

```
<script src="js/theme.js"></script>
```

```
<!-- Style Switcher -->
```

```
<script type="text/javascript" src="style-switcher/js/switcher.js"></script>
```

```
<!-- angular js -->
```

```
<script src="js/angular_product.js"></script>
```

```
<body onload="onLoad()">
```

```
</body>
```

```
</html>
```

```
12.2 user_login.php
```

```
<?php
```

```
/* Following code will match admin login credentials */
```

```
//user temp array
```

```
$response = array();
```

```
// include db connect class
```

```
require_once __DIR__ . '/db_connect.php';
```

```
// connecting to db
```

```
// check for post data
```

```
$data = json_decode(file_get_contents("php://input"));
```

```
$get_empid = ($data->email);
```

```
$get_password = ($data->password);
```

```
if(empty($get_empid) || empty($get_password))
```

```
{
```

```
    $response["success"] = 2;
```



```
    echo json_encode($response);
}
else
{
    $result = mysqli_query($conn,"SELECT * FROM login WHERE email = '$get_empid' AND password = '$get_password'
");

    if (mysqli_num_rows($result))
    {
        $Allresponse = mysqli_fetch_array($result);
        // temp user array
        $response = array();
        $response = $Allresponse;
        $response["success"] = 1;
        echo json_encode($response);
    }
    else
    {
        // success
        $response["success"] = 0;
        // echoing JSON response
        echo json_encode($response);
    }
}
?>
```