# Chapter-2

Requirement Engineering

## 2.1 Functional Requirement

**2.1.1 USER LOGIN**

Description of feature

This feature used by the user to login into system. They are required to enter user id and password before they are allowed to enter the system. The user id and password will be verified and if invalid id is there user is allowed to not enter the system.

Functional requirements

* User id is provided when they register.
* The system must only allow user with valid id and password to enter the system.
* The system performs authorization process which decides what user level can acess to.
* The user must be able to logout after they finished using system.
* The users with user type ‘user’ can generate request, view request status and update profile.
* The users with user type ‘admin’ can change request status, add new admin or engineer and update profile.
* The users with user type ‘engineer’ can view the list of requests to which he is allotted. He can also view the request on the map and change his profile.
* The users with user type ‘contractor’ can view the requests to which his tender is selected and also the list of requests to which the status is selected as ‘engineerAlloted’ and update his profile.

**2.1.2 REGISTER NEW USER**

This feature can be performed by all users to register new user to create account.

Functional requirements

* System must be able to verify information.
* System must be able to check if the user already exist.

**2.1.3 REQUEST FOR ROAD REPAIR**

Description of feature

This feature allows user to request for road repair.

Functional requirements

* System must be able to locate the location of road.
* System must be able to update the request schema.

## 2.2 Non-Functional Requirements

## 2.2.1 PRODUCT REQUIREMENT

## 2.2.1.1 EFFICIENCY REQUIREMENT

When a request is generated the user allots an engineer to the request for verification and after verification a contractor is allotted for further processing.

**2.2.1.2** **RELIABILITY REQUIREMENT**

The system should accurately locate the address where road repairs is requested. The user information is fetched automatically to the request.

**2.2.1.3 USABILITY REQUIREMENT**

The system is designed for a user-friendly environment so that a user can freely request for road repairs.

**2.2.2 ORGANIZATIONAL REQUIREMENT**

**2.2.2.1 IMPLEMENTATION REQUIREMENTS**

In implementing whole system, it uses pug in front end with NodeJS as server-side scripting language which will be used for database connectivity and the backend i.e. the database part is developed using mongoDB.

## 2.3 Technology Used

## 2.3.1 HTML

**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. Web browsers receive HTML documents from a web server or from local storage and render the documents 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. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

**2.3.2 CSS**

Css Stands for "Cascading Style Sheet." Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML.

CSS helps Web developers create a uniform look across several pages of a Web site. Instead of defining the style of each table and each block of text within a page's HTML, commonly used styles need to be defined only once in a CSS document. Once the style is defined in cascading style sheet, it can be used by any page that references the CSS file. Plus, CSS makes it easy to change styles across several pages at once. For example, a Web developer may want to increase the default text size from 10pt to 12pt for fifty pages of a Web site. If the pages all reference the same style sheet, the text size only needs to be changed on the style sheet and all the pages will show the larger text.

**2.3.3 JAVASCRIPT**

JavaScript (JS) is a programming language mostly used to dynamically script webpages on the client side, but it is also often utilized on the server-side, using packages such as Node.js.

JavaScript should not be confused with the Java programming language. Both "Java" and "JavaScript" are trademarks or registered trademarks of Oracle in the U.S. and other countries. However, the two programming languages are significantly different in their syntax, semantics, and uses.

Conceived as a server-side language by Brendan Eich (then employed by the Netscape Corporation), JavaScript soon came to Netscape Navigator 2.0 in September 1995. JavaScript enjoyed immediate success and Internet Explorer 3.0 introduced JavaScript support under the name JScript in August 1996.

In November 1996, Netscape began working with ECMA International to make JavaScript an industry standard. Since then, the standardized JavaScript is called ECMAScript and specified under ECMA-262, whose latest (eight) edition is available in June 2017.

JavaScript is mostly used in the browser, enabling developers to manipulate webpage content through the DOM, manipulate data with AJAX and IndexedDB, draw graphics with canvas, interact with the device running the browser through various APIs, and so forth. JavaScript is one of the world's most commonly-used languages, owing to the recent growth and performance improvement of APIs available in browsers.

Recently, JavaScript's popularity has expanded even further through the successful Node.js platform-the most popular cross-platform JavaScript runtime environment outside the browser. Node.js allows developers to use JavaScript as a scripting language to automate things on a PC and build fully functional HTTP and Web Sockets servers.

**2.3.4 MONGODB**

MongoDB is an open source database that uses a document-oriented data model.

MongoDB is one of several database types to arise in the mid-2000s under the NoSQL banner. Instead of using tables and rows as in relational databases, MongoDB is built on architecture of collections and documents. Documents comprise sets of key-value pairs and are the basic unit of data in MongoDB. Collections contain sets of documents and function as the equivalent of relational database tables. Like other NoSQL databases, MongoDB supports dynamic schema design, allowing the documents in a collection to have different fields and structures. The database uses a document storage and data interchange format called BSON, which provides a binary representation of JSON-like documents. Automatic sharding enables data in a collection to be distributed across multiple systems for horizontal scalability as data volumes increase.

MongoDB was created by Dwight Merriman and Eliot Horowitz, who had encountered development and scalability issues with traditional relational database approaches while building Web applications at DoubleClick, an Internet advertising company that is now owned by Google Inc.

**2.3.5 NODEJS**

Node.js is a very powerful JavaScript-based framework/platform built on Google Chrome's JavaScript V8 Engine. It is used to develop I/O intensive web applications like video streaming sites, single-page applications, and other web applications. Node.js is open source, completely free, and used by thousands of developers around the world.