**WORDICT**

**A Minor PROJECT REPORT**

**SUBMITTED BY**

AYUSHI SINGH

0827CS141066

GUNIKA JAIN

0827CS141089

****

**DEPARTMENT of COmputer Science & Engineering**

**ACROPOLIS INSTITUTE OF TECHNOLOGY & RESEARCH, INDORE**

**JUNE, 2017**

**RAJIV GANDHI TECHNICAL UNIVERSITY, BHOPAL**

**WORDICT**

**A Minor PROJECT REPORT**

**SUBMITTED IN VI SEMESTER**

**Bachelor of Engineering**

**BY**

AYUSHI SINGH

0827CS141066

GUNIKA JAIN

0827CS141089

**UNDER THE ESTEEMED GUIDANCE OF**

**MR. YOGESH KAKDE**

****

**Department of Computer Science & Engineering**

**ACROPOLIS INSTITUTE OF TECHNOLOGY & RESEARCH, INDORE**

**JUNE, 2017**

**RAJIV GANDHI TECHNICAL UNIVERSITY, BHOPAL**

**Certificate OF ORIGINALITY**

We, hereby declare that the project entitled “WORDICT” submitted to **department of Computer Science & Engineering, Acropolis Institute of Technology & Research, Indore(M.P.)** in BE VI semester in session June2017 is an authentic record of our own work carried out under the guidance of **Mr. Yogesh Kakde** & that the project has been developed by us and not previously formed the basis for the award of any other degree.

Signature of Student

AYUSHI SINGH

0827CS141066

GUNIKA JAIN

0827CS141089

This is to certify that the above statement made by the student is correct to the best of my knowledge.

Signature of HOD Signature of Guide

Prof. Sanjay Bansal Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of External Examiner Signature of Internal Examiner

Name of External Examiner Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date:

Place: Indore

**ACKNOWLEDGEMENT**

We take this opportunity to express our deepest and sincere gratitude to our guide **Mrs. Preeti Jain, Computer Science & Engineering Department, Acropolis Institute of Technology & Research, Indore** for her insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project and also for her constant encouragement and advice.

We express our deep gratitude to **Ms. Kavita Namdev, SeniorAssistant Professor, Computer Science & Engineering Department, Acropolis Institute of Technology & Research, Indore** for her regular support, co-operation, and co-ordination. We also express our hearty gratitude to **Prof. Sanjay Bansal, Professor & Head, Computer Science & Engineering Department, Acropolis Institute of Technology & Research, Indore** for his support, and help provided during the project tenure.

We also express our deep gratitude to **Dr. Kamal Bharani,Principal, Acropolis Institute of Technology & Research, Indore** for all the help provided for the completion of project.

The in-time facilities provided by the department throughout the Bachelors program are also equally acknowledgeable.

Finally, yet more importantly, I would like to express my deep appreciation to my parents, classmates and well-wishers for their perpetual support and encouragement.

**Ayushi Singh (0827CS141066)**

**Gunika Jain (0827CS141089)**

**ABSTRACT**

The idea of this project is to create a blogging web application and provide certain services to users who register. This online application elementally focuses on to make people interact and get informed about all the different interests of the people in the world with authentic, pertinent and timely knowledge to lead edifying live.

While making decisions about health, it is important to gather the most reliable information. This online platform shall bring the latest facts and figures about health and medicine sector.

From searching about the health problems to discussing about them to health experts, this website will also provide the latest news, hospitals information, serves details of medicines with category name and also, a forum to solve the queries of customers. Moreover, this is a single platform which will act as a source of particulars of health, medication and clinics knowledge, FAQs for immediate answers.

As a matter of fact, this would also bring together patients, doctors, health organizations from everywhere. Moreover, it is possible to create awareness about services and products pertaining to healthcare and medicine. It shall provide opportunities to explore, learn and it serves as a solution to some of the major health problems and risky diseases.

In this application, Admin can supervise all the activities.

**,**

**CHAPTER 1 - Software Project Management Plan (SPMP)**

|  |
| --- |
| Table of Contents Page No. |
| 1. INTRODUCTION  1.1 Project Overview  1.2 Project Deliverables  2. PROJECT ORGANIZATION  2.1 Software Process Model  2.2 Roles and Responsibilities  2.3 Tools and Techniques  3. PROJECT MANAGEMENT PLAN  3.1 Tasks  3.2 Information Gathering  3.3 Resources Needed  3.4 Dependencies and Constraints  3.5 Risks and Contingencies  4. PROJECT SCHEDULING  4.1 Gantt Chart |

**1. INTRODUCTION**

**1.1 Project Overview**

The purpose of this document is to present a detailed description of the Blogging Website. It will explain the purpose and features of the website, its interfaces, its working, and the constraints under which it must operate and how the web application will react to external stimuli. This software application will be a Web application for a local blogger. This web application will be designed to maximize the blogger’s productivity by providing tools to assist in automating the blog review and publishing process, which would otherwise have to be performed manually. By maximizing the blogger’s work efficiency and production the web application will meet the blogger’s needs while remaining easy to understand and use.More specifically, this web application is designed to allow a blogger to manage and communicate with a group of reviewers and authors to publish blogs to a public website. The software will facilitate communication between authors, reviewers, and the bloggers via E-Mail. Pre-formatted reply forms are used in every stage of the blogs’ progress through the web application to provide a uniform review process. The web application also contains a relational database containing a list of Authors, Reviewers, and Blogs.

**1.2 Project Deliverables**

Our web application focuses on simplifying the way of blogging for users. This website is easily available, charge-less, simple to operate, well-structured and interactive. Everyone can get all the information related to what they like, how to prepare a blog and how to post it. It also lets everyone to interact and give their experience, feedbacks and some valuable tips from known bloggers.The main feature of the Web Application is that it keeps record of all essentials details of blogs that come for people. It provides all essential information related to the different interests and also what is happening worldwide. Also the people already blogging, can share their blogs, opinions and feedback.

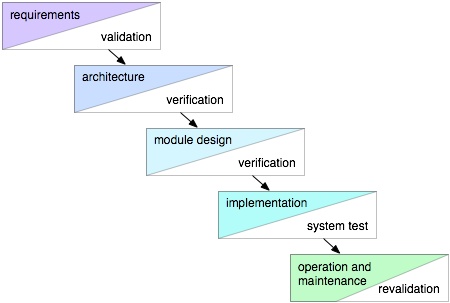
**2. PROJECT ORGANIZATION**

**2.1 Software Process Model**

A system too large for one person to build is usually also too large to build without an overall plan that coordinates the people working on it, the tasks that need to be done, and the artifacts that are produced. Researchers and practitioners have identified a number of software development process models for this coordination.

These process models are alternatives, but not exclusive ones: most describe different aspects of a process, and it is common for a development group to be following two or more simultaneously

The software process model used;The waterfall model was in use as early as the late 1950's. It was first described explicitly (by [Royce](http://www.thomasalspaugh.org/pub/fnd/softwareProcess.html#Royce1970-mdls) in 1970) as a way software should not be produced.



The simple waterfall model describes a sequence of activities and corresponding artifacts, from the most general (requirements) through successively more detailed steps to implementation, after which the software is put into operation and maintained. Each step has a corresponding verification and validation activity: requirements are validated, architectures are verified against requirements, module designs are verified against architectures and requirements, implementation is tested, and the operational system is revalidated through its use. In this idealized process, each stage is a (relatively) complete and correct description of the final system, produced using the results of the preceding stage. If all goes well, the final system will meet the initial requirements and the stakeholders will rejoice.

As [Royce](http://www.thomasalspaugh.org/pub/fnd/softwareProcess.html#Royce1970-mdls) pointed out, though, this can't actually be expected to happen. In working out the requirements, we can't possibly take advantage of what we will learn later when doing the architecture, design, implementation, and operation of the system: and it is not humanly possible that the requirements will be completely appropriate without that later knowledge. The same thing is true for each later phase: to make an appropriate architecture, we have to know the things we won't learn until we do the design, implementation, and operation; and so forth. Each phase has to be partially re-done based on the knowledge gained later. Shown are the feedback paths that occur in practice: every later phase uncovers knowledge that is needed to do each early phase well. In order to work well, this feedback and rework has to be organized somehow.

The simple waterfall model is perhaps most useful for showing the minimum dependencies that occur among the software development phases and artifacts: everything else depends on the requirements; design and implementation also depend on the architecture; and implementation also depends on design.

**2.2 Roles and Responsibilities**

**Blogger**:

A blogger is someone that writes a blog. A blog is an online journal that a writer can use to share his ideas, thoughts, expertise, or interests. Therefore, a blogger is someone that writes his ideas and thoughts online. Blogging is about self-expression. Bloggers write about their lives, their experiences, their expertise, their passions, and their opinions.

Because many do not understand blogging, they expect journalism from bloggers. Bloggers express how they feel about issues. On the other hand, a journalist is someone who writes for a magazine or newspaper, usually writing news items and is required to tell all sides of a story. Journalists are required to be unbiased and to refrain from expressing their opinions in news stories. Journalists are required to have a number of different sources before publishing an article. Objectivity is key in journalism. Blogger can be biased.

Blogging is not journalism. Blogging involves self-expression. Journalism seeks to disseminate facts as unbiased and be as impersonal as possible.

In their pure forms, journalism and blogging are two different professions, though they may have similarities and may overlap.

**Responsibilities**

* Write fiction or nonfiction through scripts, novels, and biographies
* Conduct research to obtain factual information and authentic detail
* Choose subject matter that interests readers
* Work with editors and clients to shape the material so it can be published
* Manage the blog community by moderating and responding to comments.
* SEO implementation
* Handle social media marketing activities such as posting to Twitter and Facebook.
* Answer emails on behalf of the blog.
* Perform blog maintenance activities such as updating plugins.
* Write content for the blog.
* Analyze web analytics data to gauge content performance.
* Manage split testing and other content performance tests.
* Manage subscriptions, syndication, and email newsletters.
* Manage guest posting requests.

**Administrator:**

An Administrator provides both clerical and administrative support to professionals, either as part of a team or individually. The role plays a vital part in the administration and smooth-running of businesses throughout industry.

**Responsibilities:**

* Using a variety of software packages, such as Microsoft Word, Outlook, Powerpoint, Excel, Access, etc., to produce correspondence and documents and maintain presentations, records, spreadsheets and databases;
* using content management systems to maintain and update websites and internal databases;
* attending meetings, taking minutes and keeping notes;
* managing and maintaining budgets, as well as invoicing;
* devising and maintaining office systems;
* sorting and distributing incoming post and organizing and sending outgoing post;
* liaising with colleagues and external contacts to book travel and accommodation;
* organizing and storing paperwork, documents and computer-based information;
* Photocopying and printing various documents, sometimes on behalf of other colleagues.
* recruiting, training and supervising junior staff and delegating work as required;
* manipulating statistical data;
* Arranging both in-house and external events.
* booking rooms and conference facilities;
* liaising with staff in other departments and with external contacts;
* ordering and maintaining stationery and equipment;

# WEB DESIGNER:

Web designers plan, create and code internet sites and web pages, many of which combine text with sounds, pictures, graphics and video-clips using both non-technical and technical skills to produce websites that fit the customer's requirements.

They are involved in the technical and graphical aspects of pages, producing not just the look of the website but determining how it works as well. Web designers might also be responsible for the maintenance of an existing site.

The term web developer is sometimes used interchangeably with web designer, but this can be confusing. Web developing is a more specialist role, focusing on the back-end development of a website and will incorporate, among other things, the creation of highly complex search functions.

There can be some pressure to meet deadlines when extra hours may be required. Promotional prospects are excellent for employees willing to change jobs regularly. Those with relevant experience can command good salaries.

**Responsibilities**

Web designer duties can vary depending upon the type of organization worked for and the technical level of the website, but can include:

* meeting clients to identify their needs and liaising regularly with them;
* drawing up detailed website specifications;
* designing sample page layouts including text size and colours;
* designing graphics, animations and manipulating digital photographs;
* registering web domain names and organising the hosting of the website;
* presenting initial design ideas to client;
* coding using a variety of software;
* working with different content management systems;
* search engine optimisation;
* meeting relevant legal requirements such as accessibility standards, freedom of information and privacy;
* designing the website's visual imagery and ensuring it is in line with company branding policy or the requirements of the client;
* proofreading content and grammar and making changes where necessary;
* editing content, debugging code and re-designing web pages;
* working with other web specialists including web developers and graphic designers;
* liaising with outside agencies;
* testing the website to ensure it is working;
* handing the completed website over to the client;
* post-sales technical support;
* training client's staff;
* researching current design trends;
* continual professional development to keep up to date with new software developments.

# WRITER:

Writers produce literary compositions, articles, reports, books and other texts.Writers are involved in the creation and development of works of fiction and non-fiction.

This covers a number of wide and varied forms including poetry, prose, life writing, and material for the theatre, screen and radio (such as comedy/soap opera scripts, drama productions and documentaries). Writers may also create the content for websites or write articles for magazines or newspapers. New media is also opening doors for writers in areas such as mobile phone content and computer game scripts.

Most writers work freelance and are self-employed. They often have to support themselves through additonal types of work, such as teaching, lecturing, editing or other roles in publishing, as well as entirely unrelated jobs outside the writing industry.

#### **Responsibilities**

* selecting subject matter based on personal or public interest, or commissioned by a publisher or agent;
* developing the technical skills of writing and maintaining originality;
* using literary skills to develop themes and storylines, while making characters and plots believable;
* working to tight deadlines, especially for theatre, screen and radio;
* undertaking research, including plot-lines, places, themes and characters;
* verifying the factual content of written work;
* conducting interviews with people either face-to-face, over the telephone or by email;
* submitting material for publication in the required and expected format;
* rewriting and adapting material (and sometimes the work of others) for alternative formats, e.g. adapting novels for stage or producing an e-book;
* ghost writing - writing for others under the other person's name;
* maintaining an active interest in the specific genre, such as novels, film, TV, radio;
* exercising self-discipline and time management to organise writing in conjunction with developing financial management/self-employment skills;
* encouraging and acting upon critical feedback in the most appropriate manner;
* being prepared to rewrite and revise work (often several times) following feedback;
* liaising with publishers, agents, script editors, producers and directors;
* finding, pursuing and maintaining knowledge of publication opportunities;
* attending courses and participating in workshops to improve and build upon writing skills;
* appearing at public readings and book signings, schools, libraries, colleges and literary festivals;
* teaching in higher and further education and privately;
* private online creative writing tuition;
* private critiquing service to aspiring writers.

**2.3 Tools and Techniques**

**Tools:**

Blogging as an E-learning Tool Instructional blogging operates as a knowledge-centered instructional tool. In this model the instructor involves students in research activities, engages them in discussions with practitioners, and leads them through developmental concepts of the discipline's knowledge domain. There's a course content included modules such as technological evolution, information architecture, strategic planning, learning objects, content management, and usability and prior technical experience with blogging was not a course requirement. One can use blogging as a learner-centered instructional tool by giving positive feedback to students on their comments in blog entries and by adding comments to discussion

Blogging expresses the importance of social and peer interaction as foci of the learning community. Instructors of courses rooted in a knowledge discipline can use blogs to lead students through the foundations of that discipline in order to contextualize real-world experiences. Because they are able to advance their own perspectives and experiences, students make an investment in what they post to their. Since blogs are presented on Web pages, embedded links are appropriate and convenient, and they make it easy for other students to access new resources quickly. Although instructors should be cognizant of the delicate balance between the synchronicity of time and place on the one hand, and the need to keep discussions focused on the topic on the other hand, blogging provides opportunities for students to interact in meaningful ways that extend instruction in the virtual classroom.

**Techniques:**

Receptive Learning, Directive Learning, and Guided Discovery Instructional blogging also utilizes the three instructional techniques). Supporting student knowledge acquisition through: (a) receptive techniques, which involve building instructional modules that open avenues to a great deal of information while limiting application and experimentation; (b) directive techniques, which emphasize frequent responses from learners with immediate feedback from the instructor; and (c) guided discovery techniques, which place the instructor in the role of the expert leading students toward identifying appropriate conceptual processes and solving real-life challenges is explained. As a receptive learning tool, blogging can be used to frame assignments within a theoretical context that encourages students to acquire information and report what they have learned specific issues, and to direct students to explore additional material. Finally, the opportunities for each student to post substantive comments to other students' blog entries add an additional tier of interactivity and social interaction. In online courses where communication remains largely text-based, such opportunities to enhance community can make significant contributions to student learning.

**3. PROJECT MANAGEMENT PLAN**

**3.1 Tasks**

1. Implementation of an website will successfully completed.

2. Manually it is possible to update, insert, delete a record of registered user.

3. Excel sheet can be easily uploaded and related operations like insertion of row, column is possible.

4. Report generation is possible and it can be weekly, monthly.

5. We will also provide push notification for each update.

6. Each and everyone will be easy connected and well aware of their status and record.

7. The system can be further enhanced and several other functionalities can be added.

8. The rreports could be generated anywhere.

**3.2 Information Gathering**

The information exchange system on devices is a very effective tool which can be used to a great extent. The system is portable and can be easily opened and used on any mobile phones supporting any OS.The use of this website can result in an efficient way of information exchange system between users. It also provides an interface which is easy to understand by the users and greatly helps in adapting to the use of this website. In present trend internet and web applications are playing important role in providing information for users on go. Because of vast features of web applications are designed to get closer to users and provide better service for any corner of the world.  
This website will help users to ease their day to day life. After adding basic data, admin user can able to maintain the profiles. Website can be upgraded according to user’s and administrator’s requirements with little changes. We can take back up for database and restore to memory card.

**3.3 Resources Needed**

The only way in which systems will meet their performance targets is for them to be specified clearly and unambiguously. It is a simple fact that if performance is not a stated criterion of the system requirements then the system designers will generally not consider performance issues. While loose or incorrectly defined performance specifications can lead to disputes between clients and suppliers. In many cases performance requirements are never ridged as system that does not fully meet its defined performance requirements may still be released as other consideration such as time to market. In order to assess the performance of a system the following must be clearly specified:

1. Response Time

2. Workload

3. Scalability

4. Platform

In other words, a functional requirement will describe a particular behavior of function of the system when certain conditions are met, for example: Send email when a new user

sign up or Open a new account.

A functional requirement for an everyday object like a cup would be: ability to contain tea or coffee without leaking.

**RESPONSE TIME**

In some cases the system response times are clearly identified as part of a business case for example a criminals fingerprint need to be identified while the criminal is still in custody (less than an hour). In some cases the response time will be dictated by legal requirements although this is rare. The general advice on response time from Jakon Nielsen book on Usability is:

0.1 second is about the limit for having the user feel that the system is reacting instantaneously, meaning that no special feedback is necessary except to display the result.  
1.0 second is about the limit for the user’s flow of thought to stay uninterrupted, even though the user will notice the delay. Normally, no special feedback is necessary during delays of more than 0.1 but less than 1.0 second, but the user does lose the feeling of operating directly on the data. 10 seconds is about the limit for keeping the user's attention focused on   
 the dialogue. For longer delays, users will want to perform other tasks while waiting for the system to finish, so they should be given feedback indicating when the system expects to be done. Feedback during the delay is especially important if the response time is likely to be highly variable, since users will then not know what to expect.

For systems that have to support significant numbers of users the cost of response times delays can actually be measured in monetary terms and therefore can form part of trade-off studies between different architectures providing different levels of performance.

The response time for website is 0.2 seconds.

**RELIABILTY**

The website provides a platform that is reliable in itself and in today's scenario the dependability on websites is very high. Our website will be very reliable in terms of bridging the gap between users. The proper implementation will lead to an efficient flow of data and will eventually lead to building the trust factor.

**SCALABILTY**

The website can be scaled to help the whole college and not just the Computer Science Department. Once we are providing good service to the CS dept. we will include all the other branches one by one. Also, more colleges can be brought on this website for better information exchange throughout the city.

**SECURITY**

Particularly the website does not really involve high end security. We are just trying to help exchange of information. The only place where security plays an important role is that we need to restrict users from registering as a fake person. Since users are only retrieving data, they don't pose a security threat as such.

**3.4 Dependencies and Constraints**

Due to lack of official platform for information sharing and retrieval, it has led to a lot of missed out opportunities and ambiguous situations. Currently used platforms like E-mail, WhatsApp, Social Media, one on one communication etc have proven to be very ambiguous at times. Also information like attendance, assignments, departmental news, events, competitions etc often goes unheard and require a lot of efforts to be fetched.

Web applications have now become the standard way of sharing information in the recent era, thus, converting the age old traditional form of information exchange into this new modern and convenient mode will impact the masses a tad higher. Hence to nullify the inconsistency inherited due to the previously used methods, we incline towards providing a proper mode of information to college pupils.

**3.5 Risks and Contingencies**

**Risks**

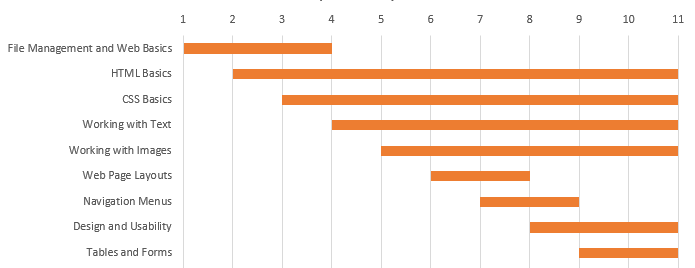
Problems arise in every organization. Such problems as what products/systems to develop, should capacity be expanded, or should a computer be purchased are just a few of an endless number of continuing problems about which management must concern itself if the firm is to survive. These problems and their alternative solutions establish some elements of change around which the organization must adapt. Projects are generally established to carry out these changes and someone is always responsible for each project's successful completion.

**Contingencies**

Every project is unique in terms of the problems that arise, the priorities and resources assigned it, the environment in which it operates, and the project manager's attitude and style used to guide and control project activities. Therefore, the organizational structure for the project must be designed to fit within that project's operating constraints. The organizational structure implemented may not be the same structure used throughout the life cycle of the project due to changes in priorities, available resource, project personnel, laws, and other contingencies. Regardless of the project management structure chosen, management must realize that a dynamic state of equilibrium between limited personnel and financial resources and the objectives of the project will be necessary if project management is to be successful in their particular organization. Before touching on the major tools and techniques of project management, let's get to the bottom of what project management truly is. Later, I will list the benefits that the tools and techniques of project management bring to the systems analysis process.

**4. PROJECT SCHEDULING**

**4.1 Gantt Chart**



Figure

**CHAPTER 2 - Software Requirements Specifications (SRS)**

|  |
| --- |
| Table of Contents Page No. |
| 1. INTRODUCTION  1.1 Product Overview  2. SPECIFIC REQUIREMENTS  2.1 External Interface Requirements  2.1.1 User Interfaces  2.1.2 Software Interfaces  2.1.3 Communications Protocols  2.2 Software Functional Requirements  2.3 Software Non-FunctionalRequirements  2.3.1 Reliability  2.3.2 Availability  2.3.3 Security  2.3.4 Maintainability  2.3.5 Portability  2.3.6 Performance |

# INTRODUCTION

## Purpose

The purpose of this document is to present a detailed description of the Blogging Website. It will explain the purpose and features of the website, its interfaces, its working, the constraints under which it must operate and how the web application will react to external stimuli.

## Scope

This software application will be a Web application for a local blogger. This web application will be designed to maximize the blogger’s productivity by providing tools to assist in automating the blog review and publishing process, which would otherwise have to be performed manually. By maximizing the blogger’s work efficiency and production the web application will meet the blogger’s needs while remaining easy to understand and use.

More specifically, this web application is designed to allow a blogger to manage and communicate with a group of reviewers and authors to publish blogs to a public website. The software will facilitate communication between authors, reviewers, and the bloggers via E-Mail. Pre-formatted reply forms are used in every stage of the blogs’ progress through the web application to provide a uniform review process. The web application also contains a relational database containing a list of Authors, Reviewers, and Blogs.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Active Blog | The document that is tracked by the web application; it is a narrative that is planned to be posted to the public website. |
| Database | Collection of all the information monitored by this web application. |
| Blogger | Person who receives blogs, sends blogs for review, likes, comments, etc |
| Field | A cell within a form. |
| Reader | Anyone visiting the site to read blogs. |
| Review | A written recommendation about the appropriateness of a blog for publication; may include suggestions for improvement. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed web application and the constraints under which it must operate. For example, this document. |
| Viewer | Reviewer or Author. |

## 1.4. References

## [facebook.github.io/react/](https://facebook.github.io/react/)

* [tutorialspoint.com/mongodb/index.htm](https://www.tutorialspoint.com/mongodb/index.htm)
* [kirupa.com](https://www.kirupa.com)
* meteor.com

## Overview

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the website. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

# OVERALL DESCRIPTION

This section of the SRS describes all general factors of the website and its requirements.

## Product Perspective

Our web application focuses on simplifying the way of blogging for users. This website is easily available, charge-less, simple to operate, well-structured and interactive. Everyone can get all the information related to what they like, how to prepare a blog and how to post it. It also lets everyone to interact and give their experience, feedbacks and some valuable tips from known bloggers

### System Interfaces

### User Interfaces

### Hardware Interfaces

### Software Interfaces

### Communication Interfaces

### Memory Constraints

### Operations

### Site Adaption Requirements

## Product Functions

The main feature of the Web Application is that it keeps record of all essentials details of blogs that come for people. It provides all essential information related to the different interests and also what is happening worldwide. Also the people already blogging, can share their blogs, opinions and feedback.

## User Characteristics

The system has two users: the registered users & the non-registered users.

* **Registered Users:** They are mainly linked to the website privileges. They will be able to have login id of their own, can easily access any blog of any interest. They can like, comment, vote in the polls, report any blog.
* **Non-Registered Users:** They can visit home page and view a section of trending blog. For rest of the privileges, they need to register.

## Constraints

The general constraints of this system are of two types:

1. Hardware Constraints

2. Software constraints

Any devices which can run on Android, iOS, etc operating systems and the following are the software constraints of the system.

* **Server side**
* **Client side**

## Assumptions and Dependencies

* The database mentioned within this SRS document is previously administered with the correct information needed by the Web Application.
* The system also assumes that details regarding each blog would be made correctly.
* Data being used for setup and user recognition is dependent on information in a database administered outside of the capabilities of the Web Application.

## Apportioning of Requirements

# SPECIFIC REQUIREMENTS

## External Interface Requirements

### User interfaces

* UI-1: The Web Application shall provide details of users present in the database.
* UI-2: These details can be clicked in order to view a particular blogger’s record.
* UI-3: All modifications to the database will be done through a keyboard.
* UI-4: The program will provide a page that produces current statistics.

### Hardware Interfaces

**1) Server Side:**

* Processor: Pentium 4.0 processor or higher
* RAM: 1 GB or more
* Hard Drive: 10 GB or more

**2) Client side:**

Any network enabled device which is able to connect to the server and running any Operating System.

* Any phone, tablet, computer, laptop, etc.

### Software Interfaces

* Web Application-This software will transmit the data from database to any phone via Internet.
* Database-The Web Application will communicate with the database to perform the following options.

1. To allow a user to login.
2. To check details of users.

### Communication Interfaces

* CI-1: The security of a user must be consistent through the use of passwords.
* CI-2: The Web Application will communicate to the database through Internet.

## Specific Requirements

### Sequence Diagram

A Sequence diagram is an [interaction diagram](https://en.wikipedia.org/wiki/Interaction_diagram) that shows how objects operate with one another and in what order. It is a construct of a [message sequence chart](https://en.wikipedia.org/wiki/Message_sequence_chart). A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged betwepen the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios..A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

### Classes of classification of Specific Requirements

### They are mainly linked to the website privileges. They will be able to have login id of their own, can easily access any blog of any interest. They can like, comment, vote in the polls, report any blog.

### User class – Non-registered user

### They can visit home page and view a section of trending blog. For rest of the privileges, they need to register.

* **User class – System Administrator**

Provides login to all the users. Upload all the details required by the user.

## Performance Requirements

PE-1: The blog must be able to get published by user as soon as it gets reviewed by the admin.

PE-2: Queries upon the database shall be performed in less than 5 seconds.

PE-3: Upon the registration of any user, all the information shall be displayed on the admin’s database within 10 seconds.

## Design Constraints

**Software Language Used**

The languages that shall be used for coding the Web Application are JavaScript, CSS, and HTML.

**Development Tools**

We have made use of the available React.js (JS library), MongoDB (database program) and Meteor (web frameworks) for the development.

Also we have made use of the online references available for developing our web application.

## Software System Attributes

### Reliability

As the website provides the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.

### Availability

The website is available 100% for the user and is used 24 hrs a day and 365 days a year. The website shall be operational 24 hours a day and 7 days a week.

### Security

* SR-1: In order to edit database information, the user shall be required to enter a password and id. This password and id shall be stored on the admin’s PC after initial entry in order that it must only be entered once.
* SR-2: Passwords shall be stored in an admin’s database and verified upon each session of roll or database modification.

### Maintainability

Updating and modification of database is required.

### Portability

Easily accessible on any Operating System.

## Other Requirements

Mongo DB is used as database.

# SUPPORTING INFORMATION

## Table of contents and Index

**1. Introduction**

   1.1. Purpose

   1.2. Scope

   1.3. Definitions, acronyms & abbreviations

   1.4. References

   1.5. Overview

**2. Overall description**

   2.1. Product perspective

        2.1.1. System interfaces

        2.1.2. User interfaces

        2.1.3. Hardware interfaces

        2.1.4. Software interfaces

        2.1.5. Communications interfaces

2.1.6. Memory constraints

        2.1.7. Operations

        2.1.8. Site adaptation requirements

   2.2. Product functions

   2.3. User characteristics

   2.4. Constraints

   2.5. Assumptions and dependencies

   2.6. Apportioning of requirements

**3. Specific Requirements**

3.1 External interface requirements

3.1.1 User interfaces

3.1.2 Hardware interfaces

3.1.3 Software interfaces

3.1.4 Communication interfaces

3.2 Specific requirements

3.2.1 Sequence diagrams

3.2.2 Classes for classification of specific requirements

3.3 Performance requirements

3.4 Design constraints

3.5 Software system attributes

3.5.1 Reliability

3.5.2 Availability

3.5.3 Security

3.5.4 Maintainability

3.6 Other requirements

**4. Supporting information**

4.1 Table of contents and index

4.2 Appendix

## Appendix

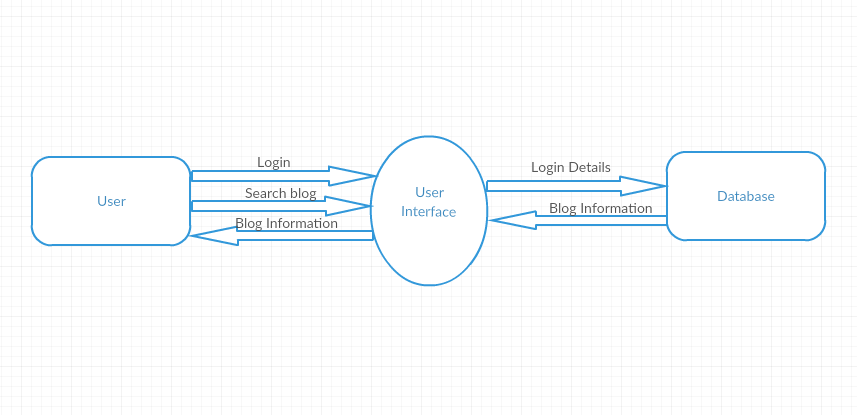
|  |  |
| --- | --- |
| **Term** | **Definition** |
| Active Blog | The document that is tracked by the web application; it is a narrative that is planned to be posted to the public website. |
| Database | Collection of all the information monitored by this web application. |
| Blogger | Person who receives blogs, sends blogs for review, likes, comments, etc |
| Field | A cell within a form. |
| Reader | Anyone visiting the site to read blogs. |
| Review | A written recommendation about the appropriateness of an blog for publication; may include suggestions for improvement. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed web application and the constraints under which it must operate. For example, this document. |
| Viewer | Reviewer or Author. |

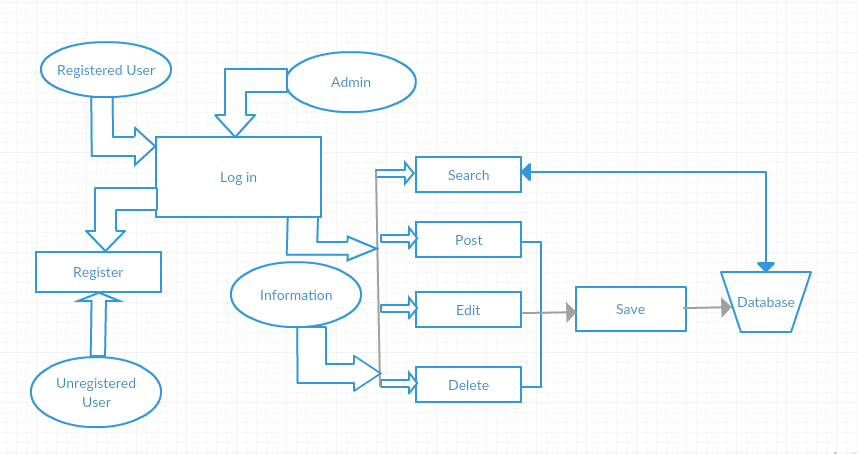
**CHAPTER 3 - Software Design Description (SDD)**

|  |
| --- |
| Table of Contents Page No. |
| 1 INTRODUCTION  1.1 Design Overview  1.2 Diagrams  1.2.1 Data Flow Diagram  1.2.2 Use Case Diagram  1.2.3 Class Diagram  1.2.4 Activity diagram  1.2.5 Sequence Diagram  1.3 Database Design  1.3.1 Entity Relationship Diagram  1.3.2 Tables Structure  1.3.3 Normalization Details  1.3.4 Data Dictionary  2. SYSTEM ARCHITECTURAL DESIGN  2.1 Chosen System Architecture  3. DETAILED DESCRIPTION OF COMPONENTS  3.1 Component 1  -  -  -  3.n Component n  4. USER INTERFACE DESIGN  4.1 Description of the User Interface  4.1.1 Screen Images |

1. **INTRODUCTION** 
   1. **Design Overview**
   2. **Diagrams**
      1. **Data Flow Diagram**

A Data Flow Diagram (DFD) is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describes the flow of data and the processes that change or transform the data throughout a system.





**1.2.2. Use Case Diagram**

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.



**1.2.3. Class Diagram**



**1.2.4. 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 form one activity to another activity. Activity diagrams are very similar to a flowchart because of modeling a workflow from activity to activity. An activity diagram is basically a special case of a state machine in which most of the states are activities and most of the transitions are implicitly triggered by completion of the actions in the source activities.



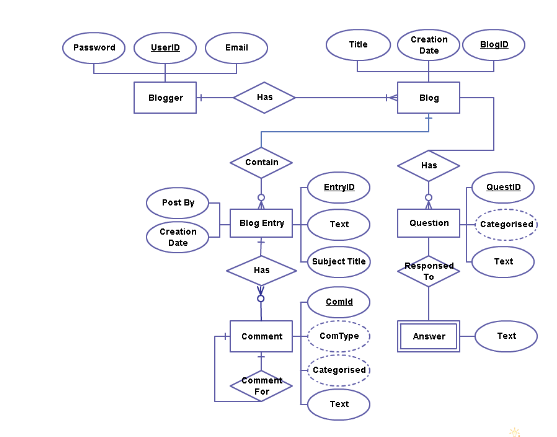
**1.2.5. Sequence Diagram**

A Sequence diagram is an [interaction diagram](http://en.wikipedia.org/wiki/Interaction_diagram) that shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

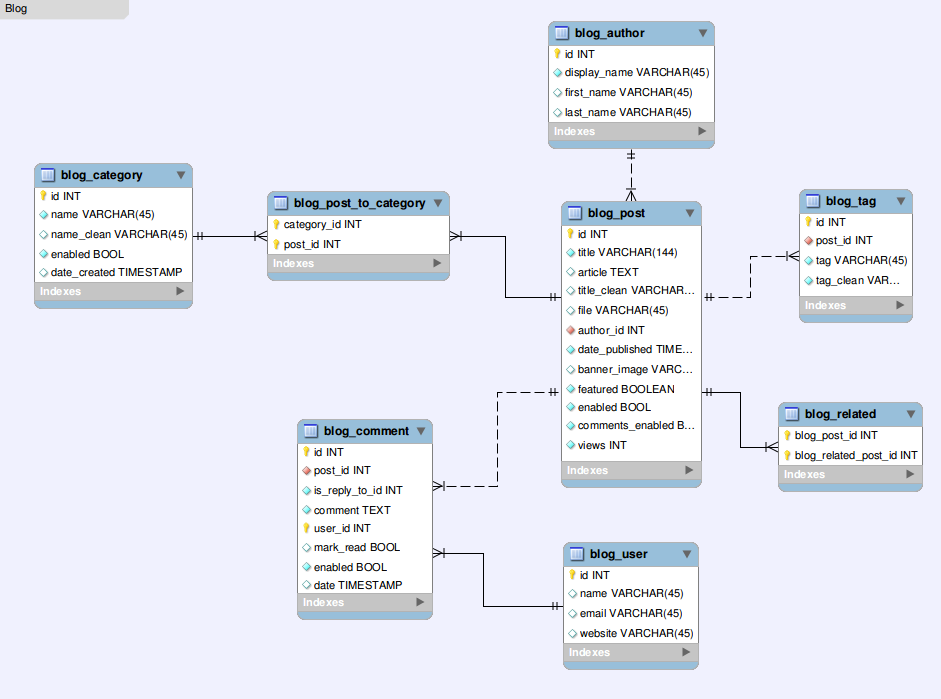


**1.3 Database Design**

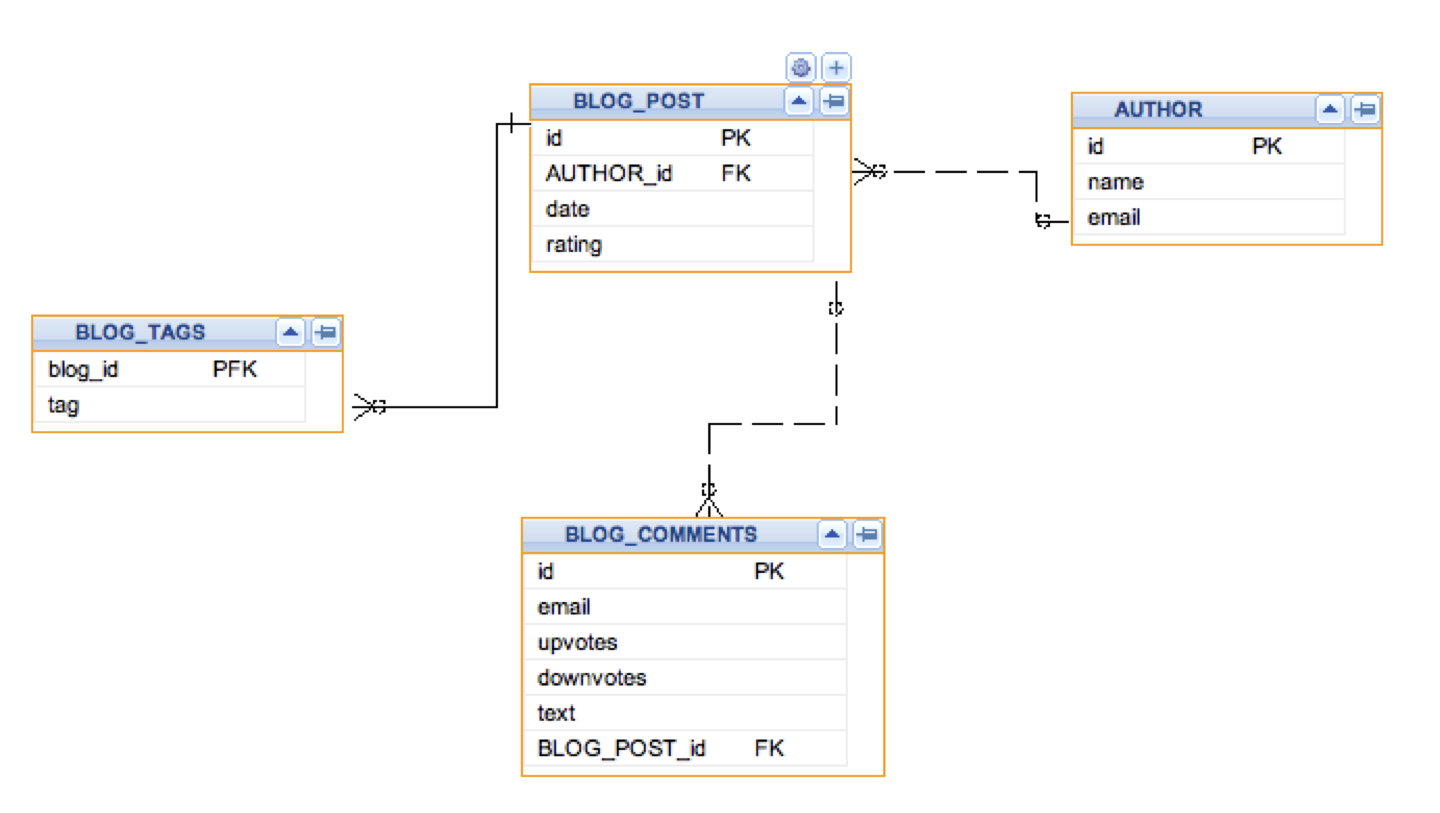
**1.3.1 Entity Relationship Diagram**



**1.3.2 Tables Structure**



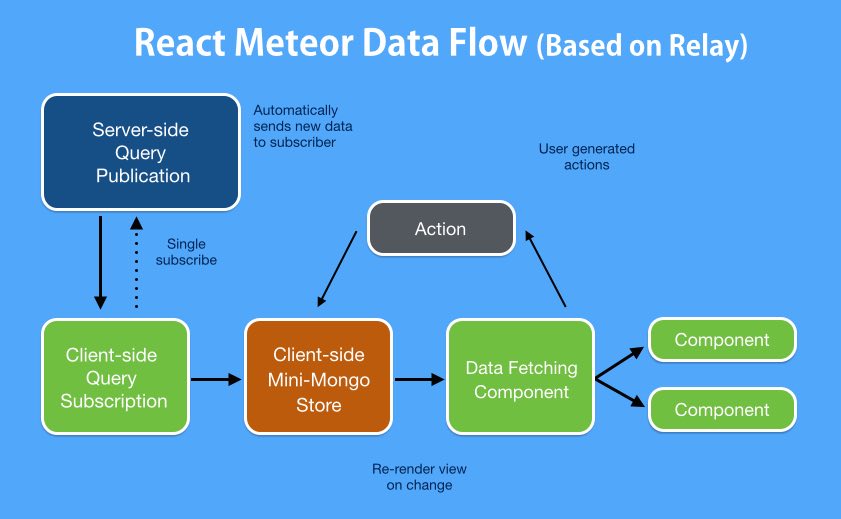
* + 1. **Normalization Details**



**2..SYSTEM ARCHITECTURAL DESIGN**

**2.1 Chosen System Architecture**

React.js, a JavaScript library, introduced a concept called Component-Based-Architecture, a method for encapsulating individual pieces of a larger user interface (aka components) into self-sustaining, independent micro-systems  
CBA also requires that all methods and APIs pertaining to a single component exist within that component’s structure. To understanding it better: a component is comprised of a JavaScript class (a new feature in ES2015). Whereas MVC separates structure, helper methods, and routing into different levels of the application, components contain all of those features within a single class. This means that developers don’t have to spend much time trying to find which functions pertain to which parts of an application’s UI.  
In the case of CBA, responsibility is split on a component-by-component basis. This means that the design, logic, and helper methods exist all within the same level of the architecture (generally the view). As aforementioned, everything that pertains to a particular component is defined within that component’s class.



**3. DETAILED DESCRIPTION OF COMPONENTS**

**3.1 Component**

Components let you split the UI into independent, reusable pieces, and think about each piece in isolation .Conceptually, components are like JavaScript functions. They accept arbitrary inputs (called "props") and return React elements describing what should appear on the screen.

The simplest way to define a component is to write a JavaScript function:  
  
function Welcome(props) {  
return <h1>Hello, {[props.name](http://www.google.com/url?q=http%3A%2F%2Fprops.name&sa=D&sntz=1&usg=AFQjCNG2eSSi_-vlBCPhykn79lRMN7ZvKw)}</h1>;  
}

This function is a valid React component because it accepts a single "props" object argument with data and returns a React element. We call such components "functional" because they are literally JavaScript functions.

**4. USER INTERFACE DESIGN**

**4.1 Description of the User Interface**

**React's Material UI**

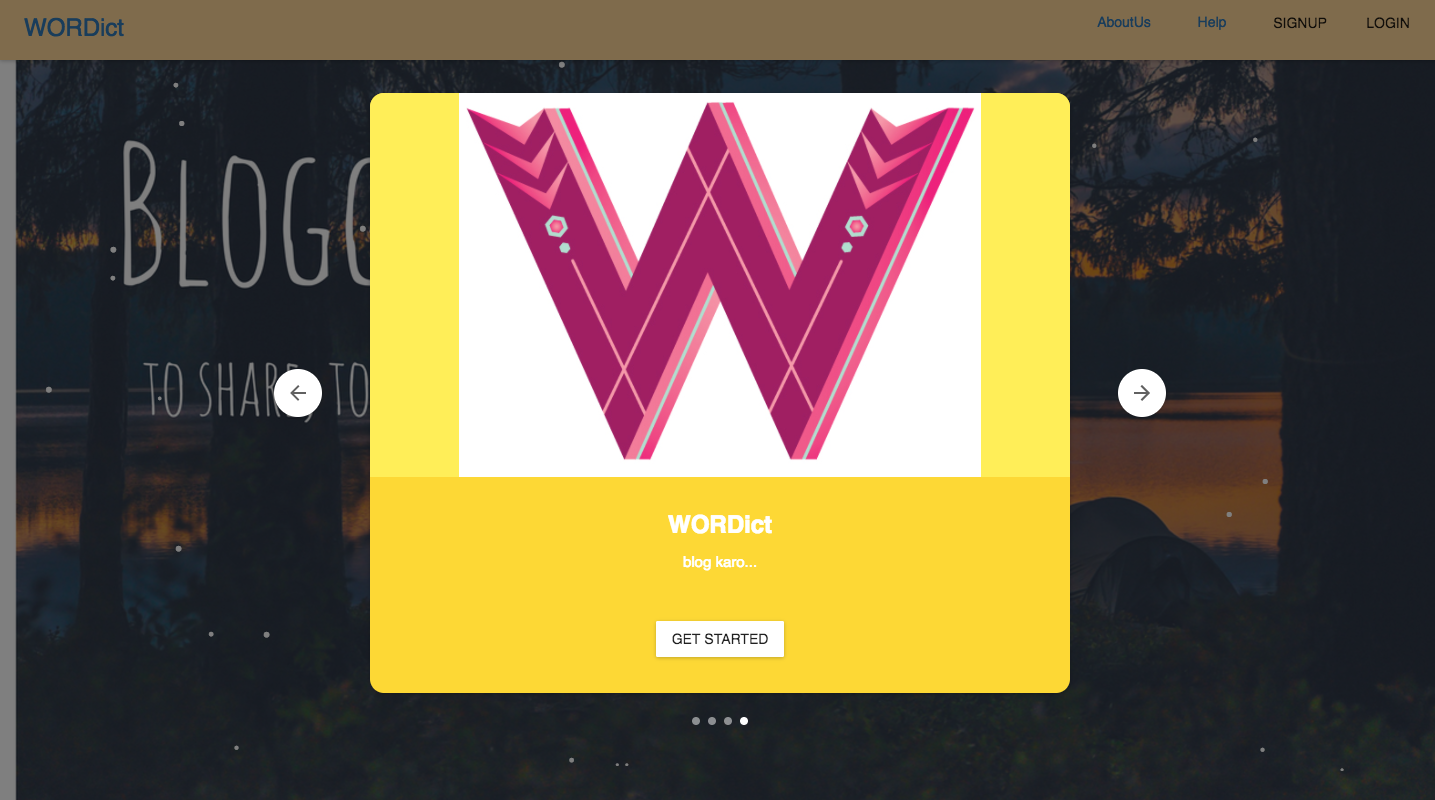
We have used React's Material UI for designing the user interface in our website.

React is a JavaScript library for building user interfaces. React asks you to break up user interfaces into different state-driven components. We then leverage the library to help us transition between these component states. It’s a noticeably different way of thinking when it comes to building the frontend of your application.

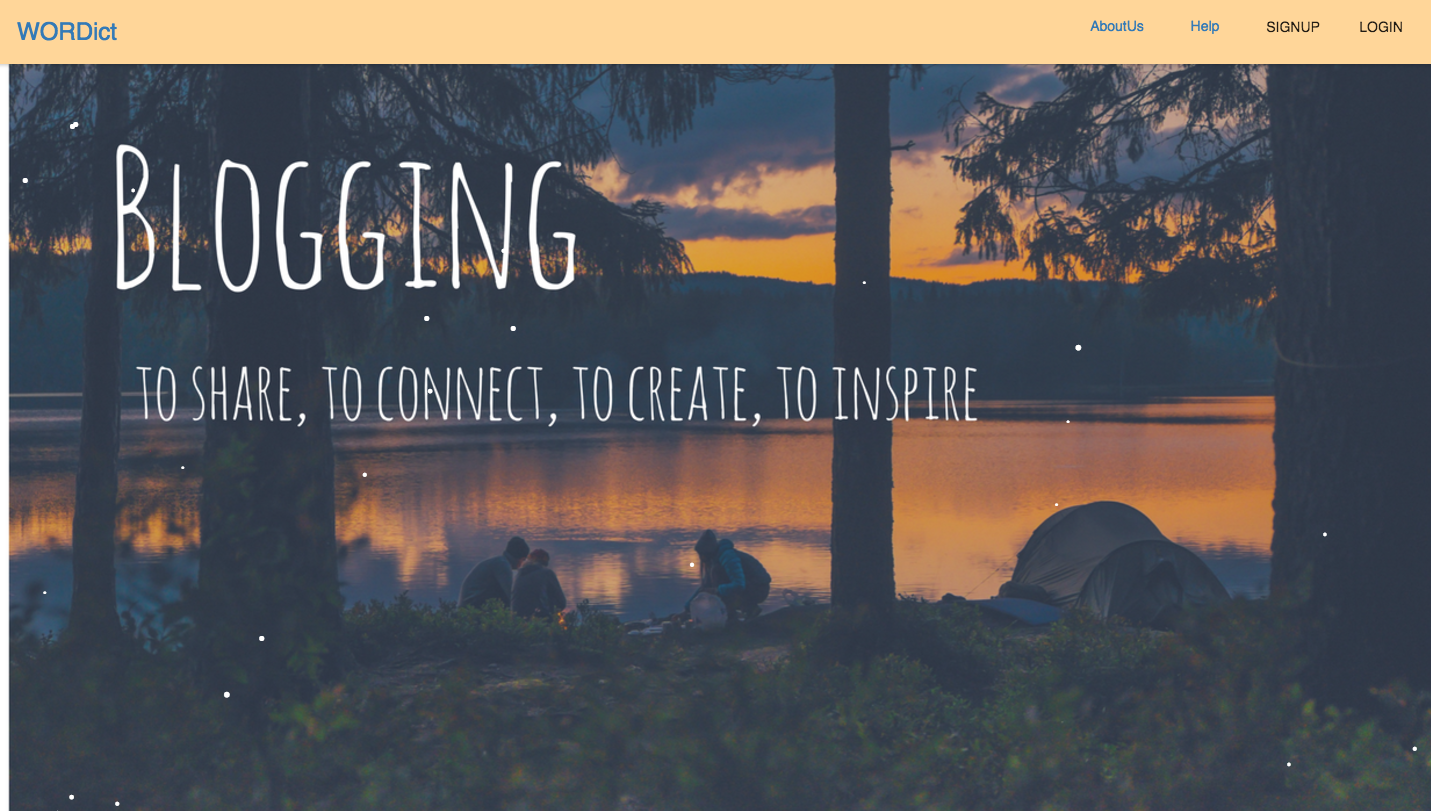
When we break up an application into states, there’s a need to also separate each piece of the view into components. Now, components aren’t really a brand new idea in web development. Developers have been breaking their views into reusable pieces for years. In this context, React is simply executing the process of interacting with these reusable views really well.

## 4.1.1. .Screen Images

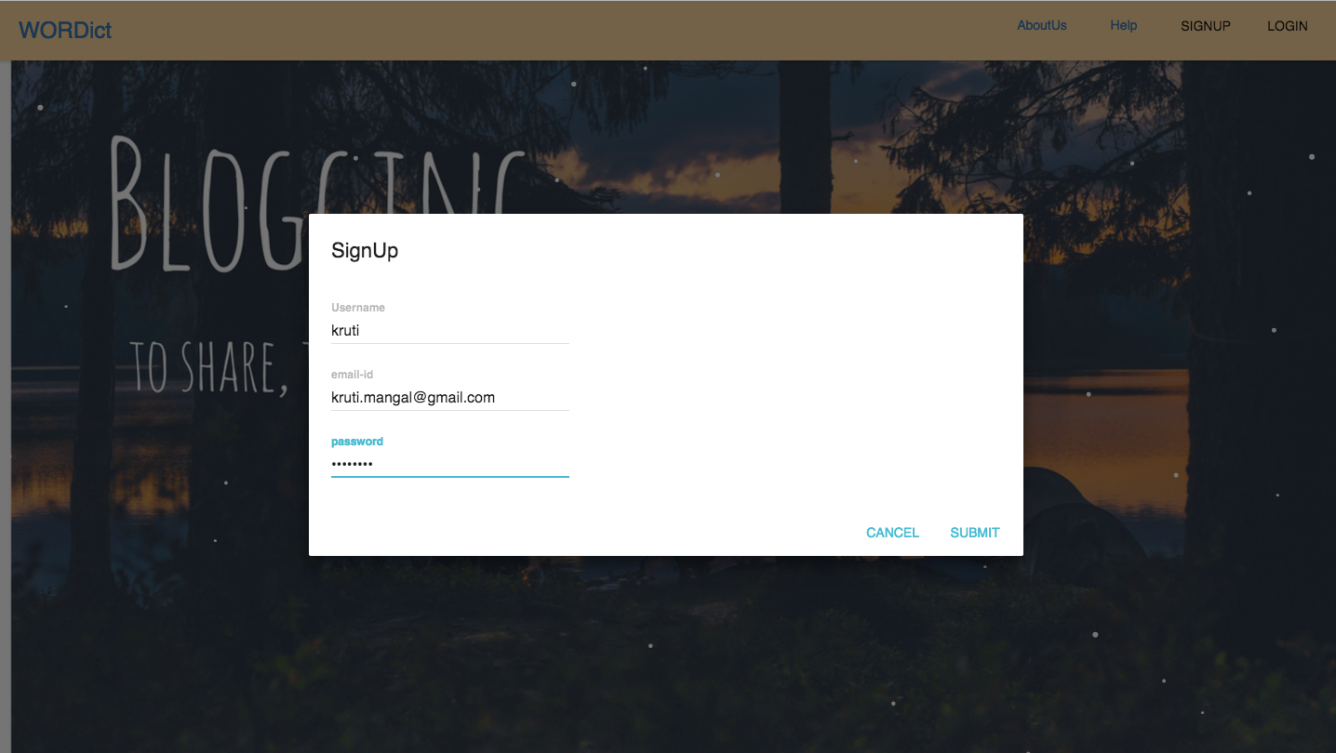
1. The first page that appears when you open the website:



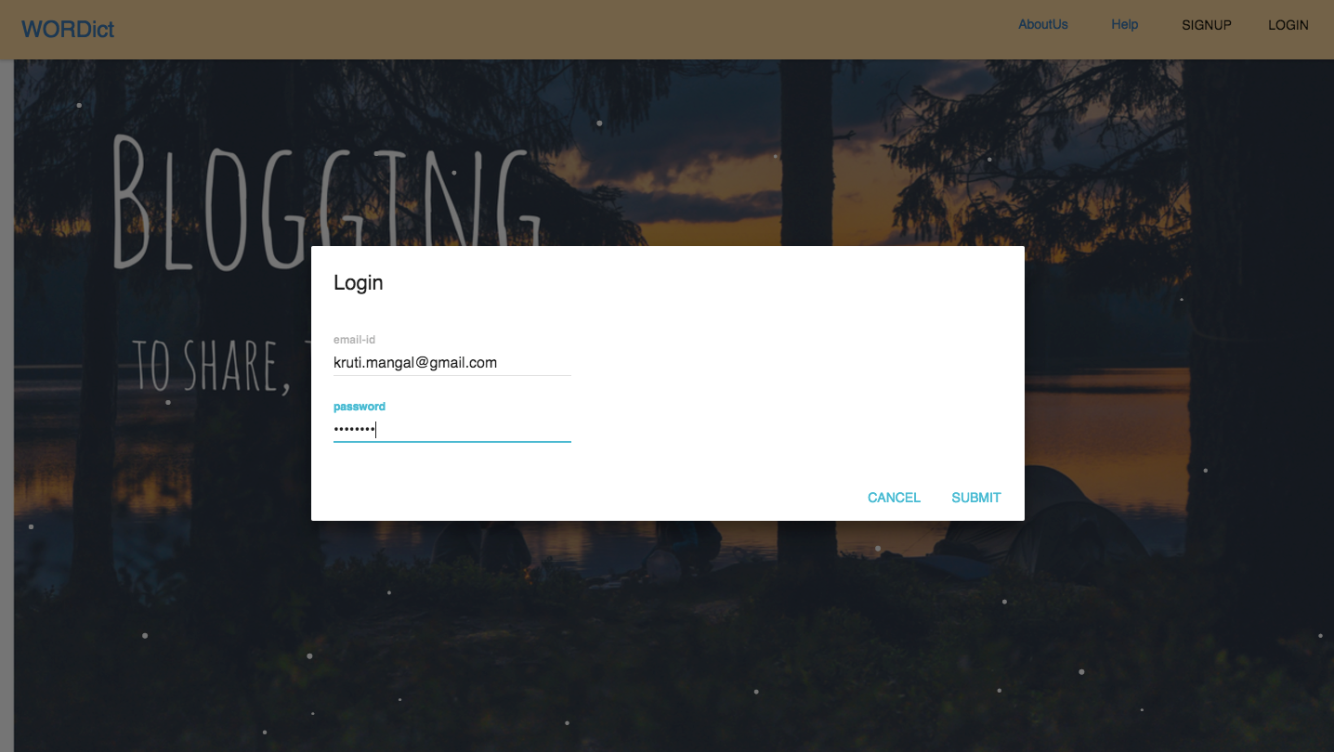
B.The homepage of the web application:



1. Sign up box appears:



1. Sign in page:



1. When user logins with the user id and password:



**CHAPTER 4 - Software Test Documentation (STD)**

|  |
| --- |
| Table of Content Page No. |
| 1 INTRODUCTION  1.1 System Overview  1.2 Test Approach  1.3 Testing Objectives  2. TEST PLAN  2.1 Features to be tested  2.2 Features not to be tested  3 TEST CASES  3.1 Unit Testing  3.2 Functional Testing  3.3 System Testing  3.4 Integration Testing  3.5 Validation Testing  3.6 Top down integration  3.7 Bottom up integration  4 Conclusion &Future Scope  APPENDIX  Bibliography  References |
|  |

1. **Introduction**

Testing documentation involves the documentation of artifacts that should be developed before or during the testing of software documentation for software testing helps in estimating the testing effort required, test coverage, requirement tracking/tracing, etc. This section describes some of the commonly used documented artifacts related to software testing. Testing allows you to ensure your application works the way you think it does, especially as your codebase changes over time. If you have good tests, you can refractor and rewrite code with confidence. Tests are also the most concrete form of documentation of expected behavior, since other developers can figure out how to use your code by reading the tests.

* 1. **System Overview**

Wordict is a blogging web application. This application has been built upon Meteor framework, using React language.

ReactTestUtils makes it easy to test React components in the testing framework of your choice. We can also use Jest for painless JavaScript testing. Automated testing is critical because it allows you to run a far greater set of tests much more often than you could manually, allowing you to catch regression errors immediately.

* 1. **Test Approach**

Dynamic user interfaces began to be an important feature for today web applications. To maintain quality, they must be tested properly. There are variety of approaches - each has its pros and cons. As a React developer we may need to choose a proper approach to test our components in an effective way. When we test React components, there are at least two concerns that must be tested:

* Given properties and state, what structure our rendered tree will have?
* Given an output of render, is there a possibility to transition from state A to state B?

Those two concerns have their own approaches to test them. We usually name the first concern as testing structure, and the second one as testing behavior. These concerns are separate, but some of testing structure approaches can affect available ways of testing behavior approaches. Since React components render a real DOM based on the virtual one, one approach is to get an actual DOM and test against it. Testing with this approach works well when you want to test things in a more ‘acceptance’ way. Components rendered can affect the outcome of such test.

* 1. **Testing Objective**

Testing should systematically uncover different classes of errors in a minimum amount of time and with a minimum amount of effort. A secondary benefit of testing is that it demonstrates that the software appears to be working as stated in the specifications. The data collected through testing can also provide an indication of the software's reliability and quality. But, testing cannot show the absence of defect -- it can only show that software defects are present.

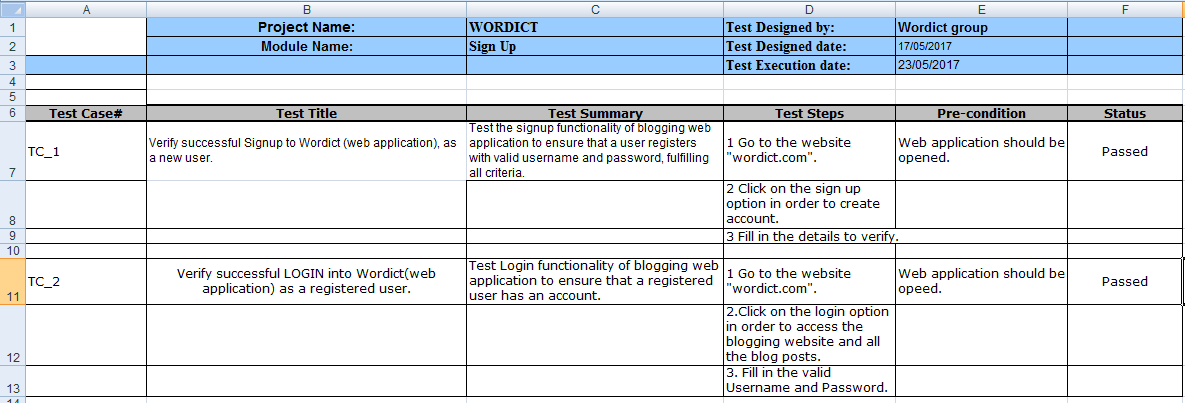
* Testing is a process of executing a program with the intent of finding an error.
* A good test case is one that has a high probability of finding an as yet undiscovered error.
* A successful test is one that uncovers an as yet undiscovered error.

1. **TEST PLAN**

A Software Test Plan is a document describing the testing scope and activities. It is the basis for formally testing any software/product in a project. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice and any risks requiring contingency planning.

* 1. **Features to be tested**
  2. **Features not to be tested**

1. **TEST CASES**

****

* 1. **Unit Testing**

Unit testing is the process of isolating a section of code and then testing that the internals of that section work as you expect. We can also apply the same structure to testing React components and recommend the Enzyme package, which simulates a React component’s environment and allows you to query it using CSS selectors. We use Enzyme’s shallow function to render the any app component, and the resulting object to query the document, and also to simulate user interactions.

To run the tests that our app defines, we run our app in test mode:

meteor test --driver-package practical meteor: mocha

* 1. **Functional Testing**

Functional Testing is a testing technique that is used to test the features/functionality of the system or Software, should cover all the scenarios including failure paths and boundary cases. React forces us to build everything as “components.” You can think of components as widgets, or as chunks of HTML with some logic. They follow many of the best principles of functional programming, except they’re objects. For instance, given the same set of parameters, a component will always render the same output. No matter how many times it’s rendered, no matter who renders it, no matter where we place the output. Always the same. As a result, we don’t have to perform complex scaffolding to test a React component. It only cares about its properties, no tracking of global variables and config objects required.

* 1. **System Testing**

System Testing tests an integrated system to verify that it meets its requirements. . This kind of test is simply executing programs to check logical changes made with intention of finding errors. A system is tested for online response, volume of transaction, recovery from failure etc. System testing is done to ensure that the system satisfies all the user requirements. We'll write a test that exercises one of our Methods (which form the "write" part of our app's API), and verifies it works correctly.

To do so, we'll add a test driver for the Mocha JavaScript test framework:

meteor add practical meteor: mocha

We can now run our app in "test mode" by calling out a special command and specifying to use the driver (we'll need to stop the regular app from running, or specify an alternate port with --port XYZ):

meteor test --driver-package practical meteor: mocha

* 1. **Integration Testing**

An integration test is a test that crosses module boundaries. In the simplest case, this simply means something very similar to a unit test, where you perform your isolation around multiple modules, creating a non-singular “system under test”.

Although conceptually different to unit tests, such tests typically do not need to be run any differently to unit tests and can use the same meteor test mode and isolation techniques as we use for unit tests.

However, an integration test that crosses the client-server boundary of a Meteor application (where the modules under test cross that boundary) requires a different testing infrastructure, namely Meteor’s “full app” testing mode.

* 1. **Validation Testing**

If you want to write a test that can be run against any running version of your app and verifies at the browser level that the right things happen when you push the right buttons, then you are writing a Validation test (sometimes called “end to end test”). Such tests typically try to hook into the application as little as possible, beyond perhaps setting up the right data to run a test against. Validation testing is the process of taking an unmodified version of our application and testing it from the “outside” to make sure it behaves in a way we expect. Typically if an app passes validation tests, we have done our job properly from a product perspective. As validation tests test the behavior of the application in a full browser context in a generic way, there are a range of tools that you can use to specify and run such tests. We’ll demonstrate using Chimp, a Validation testing tool with a few neat Meteor-specific features that makes it easy to use.

Chimp requires node version 4 or 5. You can check your node version by running:

node –v.

* 1. **Top-down Integration**

In this approach testing is conducted from main module to sub module. If the sub module is not developed a temporary program called STUB is used for simulate the sub module.

* 1. **Bottom-up Integration**

In this approach testing is conducted from sub module to main module, if the main module is not developed a temporary program called DRIVERS is used to simulate the main module.

1. **CONCLUSION AND FUTURE SCOPE**

**Conclusion**

This single page blogging web application is a very effective tool which can be used to a great extent. The system is portable and can be easily installed and used on any mobile phones/tablet/desktop supporting Android OS or iOS. The use of this system can result in an efficient way of information exchange system between bloggers all around the world. It also provides information which is easy to understand by the users and greatly helps in adapting to the use of this system. In present trend internet and web applications are playing important role in providing information for users on go. Because of vast features of web applications are designed to get closer to users and provide better service for any corner of the world.

This application will help users to ease their day to day life. Admin can post trending blogs on the basis of views or likes. Users can maintain their profile. The registered user can like and comment on blogs. Any blog, if found offensive, can be reported by the viewer to the Admin. Application can be upgraded according to user’s and administrator’s requirements with little changes. We can take back up for database and restore to memory card. Therefore, the product is useful for everyone to share information or journals of their interests.

**Future Enhancement**

As you know that the number of blogs are becoming larger day by day. So the blogs which have high quality content survive and the blogs with lower content quality will not succeed on the internet. The young generation are now moving from static content to dynamic content like flash and updating information like twitter. Twitter has a big value in social networking sites. Most bloggers share their website links on twitter. So you have to keep in touch with twitter to have latest news about blogs. Nowadays you have to not worry about how to get latest information, you just have to visit the specific site of your interest and view the main page. The main page of website gives you latest information about the website content. In future we will focus on four aspects mainly, for the enhancement of our web application.

* Using a Theme-Without writing any code, our blog application is already themeable. To use a theme, we mainly need to develop the theme by writing customized view files in the theme. For example, to use a theme named classic that uses a different page layout, we would create a layout view file. We also need to change the application configuration to indicate our choice of the classic theme.
* Internationalization -We may also internationalize our blog application so that its pages can be displayed in different languages. This mainly involves efforts in two aspects. First, we may create view files in different languages. For example, for the index page we can create a view file. When the application is configured to use simple Chinese (the language code is zh\_cn. Second, we may create message translations for those messages generated by code.
* Improving Performance with Cache.
* Adding New Features- Our blog application only has very basic functionalities. To become a complete blog system, more features are needed, for example, calendar portlet, email notifications, post categorization, archived post portlet, and so on. We will leave the implementation of these features to interested readers.

**APPENDIX**

|  |  |
| --- | --- |
| Back-end | A back-end application or program serves indirectly in support of the front-end services, by being closer to the required resource or having the capability to communicate with the required resource. |
| Front-end | Those elements of a website that the customer sees and interacts with directly. |
| GUI(Graphical User Interface) | A [computer program](http://www.britannica.com/EBchecked/topic/130654/computer-program) that enables a person to communicate with a [computer](http://www.britannica.com/EBchecked/topic/130429/computer) through the use of symbols, visual metaphors, and pointing devices. |
| HTML(Hypertext Markup Language) | The standard [markup language](http://en.wikipedia.org/wiki/Markup_language) used to create [web pages](http://en.wikipedia.org/wiki/Web_page). |
| JavaScript | JavaScript is the programming language of HTML and the Web. |
| Software Requirement Specification | A document that completely describes all of the functions and the behavior of proposed software and the constraints under which it must operate. |
| UML(Unified Modeling Language) | A standard language for writing software blueprints. |

**BIBLIOGRAPHY**

* https://www.google.co.in/
* React: Up and Running by StoyanStefanov
* https://en.wikipedia.org/

**REFERENCES**

* Available at https://www.meteor.com/
* Available at http://www.stackoverflow.com
* Available at http://www.w3schools.com
* Available at http://www.material-ui.com/#/
* Available at http://www.softwaretestingclass.com
* Available at http://www.tutorialspoint.com
* Available at https://www.kirupa.com/react/

**LIST OF FIGURES**

List of Symbols

List of Tables

LIST OF ABBREVATIONS