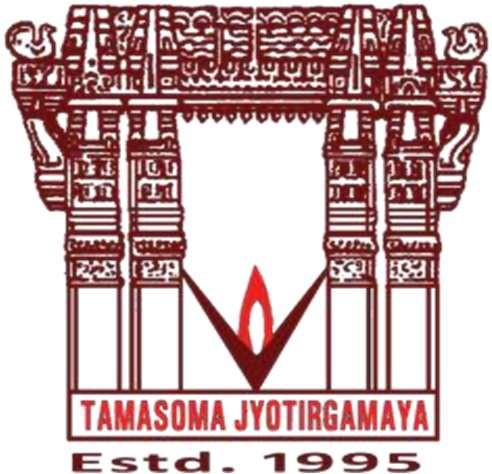


**High Level Design Document**

**On**

**Online Forum**



VNR Vignana Jyothi Institute of Engineering & Technology Bachupally, Nizampet (S.O), Hyderabad–90

***Submitted By***

***Group Details:***

**A.ROHITH - 19071A05C4**

**A.SURYA THEJA – 19071A05C5**

**MD.SHOIEB IQBAL – 19071A05F2**

**M.AKHIL REDDY – 19071A05F3**

**T.C.KEVIN SUCHETAN – 19071A05H8**

Contents

[1. Revision History 3](#_Toc74225336)

[1.1 Purpose 4](#_Toc74225337)

[1.2 Audience 4](#_Toc74225338)

[1.3 Design Process 4](#_Toc74225339)

[2. Requirements 5](#_Toc74225340)

[2.1 Proposed Solution 6](#_Toc74225341)

[2.2 Capacity Planning 6](#_Toc74225342)

[3. Architecture 7](#_Toc74225343)

[3.1 Design 7](#_Toc74225344)

[Use Case Diagram: 7](#_Toc74225345)

[Class Diagram : 8](#_Toc74225346)

[Server: 9](#_Toc74225347)

[Apache Server 9](#_Toc74225348)

[MongoDb Database 9](#_Toc74225349)

[3.1.1 Version 9](#_Toc74225350)

[3.1.2 Server Roles 10](#_Toc74225351)

[Hardware Requirements: 10](#_Toc74225352)

[Operating System: 10](#_Toc74225353)

[Packages Installed: 10](#_Toc74225354)

[3.4 System Connectivity 10](#_Toc74225355)

[4 Standards 11](#_Toc74225356)

[4.1 Security Standards 11](#_Toc74225357)

[4.1.1 Authorization and Logon 11](#_Toc74225358)

[4.2 Disaster Recovery 11](#_Toc74225359)

[5. Support 11](#_Toc74225360)

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Date | Author | Section | Comments/Changes |
| 1.0.0 | 01 May | All | All | Initial Revision |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* 1. **Project Overview**

## Purpose

The purpose of this document is to specify the high-level design for the ONLINE FORUM. This document will act as an outline for implementation and discuss the design considerations.

## Audience

This high-level design is intended to be used by members of the development team who are going to implement the functionality of the ONLINE FORUM. This document will also be used to communicate the high-level design and design considerations to the ONLINE FORUM staff members.

* + - * A. Rohith , Development Team
      * A. Surya Theja, Development Team
      * Md. Shoeib Iqbal, Development Team
      * M. Akhil Reddy, Development Team
      * T.C. Kevin Suchetan, Development Team

## Design Process

The high-level design was selected by deciding what aspects of the system were most important for the software. Several architectures were proposed initially but after going through a lot of discussion we decided to go with the single page architecture.The pros and cons of each architecture and technology were discussed in meetings. Communication between client and server would utilize XML and AJAX. . This technology would also provide the ability for any interface to utilize the web services of the web server on the network.

AJAX, or Asynchronous JavaScript and XML. AJAX’s unobtrusive data requests, combined with the power of JavaScript to dynamically update the Document Object Model (DOM), and the use of CSS to change the page’s style on the fly, brought AJAX to the forefront of modern web development.

Communication between the client and server would be invocations through a proxy passing JSON objects as parameters. After research and proof of concepts, the team reached consensus to implement a single page architecture that uses AJAX technology.

# Requirements

The overall requirements of this system remain unchanged from the “Analyze” phase gate deliverable of the Software Requirements Specification. In summary from that document, the main requirements are:

* The user or employee can login through any computer on the network using his ID and password
* The system must be able to authenticate and give respective privileges to add a query, post and answer the questions.
* The server should be up and running during working hours without a single.
* point of failure. It should be able to support all kinds of loads.
* The Server and application must be robust.
* A registered user should have access to all the features of a Web application.
* The software should be capable of finding the user's interest in different domains.
* Checking the website on a regular interval basis will be done to detect eventual anomalies

## Proposed Solution

The proposed solution of our project involves providing services to the geeks by making it easy for them to post their queries and get a solution for it. The system provides a simple interface to both users and administrators. It takes login ID and password as input from the user for login and will be developed with features such as joining groups and posting questions.

There will be an express server running in the backend which fetches the data from MongoDB and an API will be developed via which the client interface will interact with the server.

## Capacity Planning

Online forums allow the users to interact in the discussion on a specified topic.

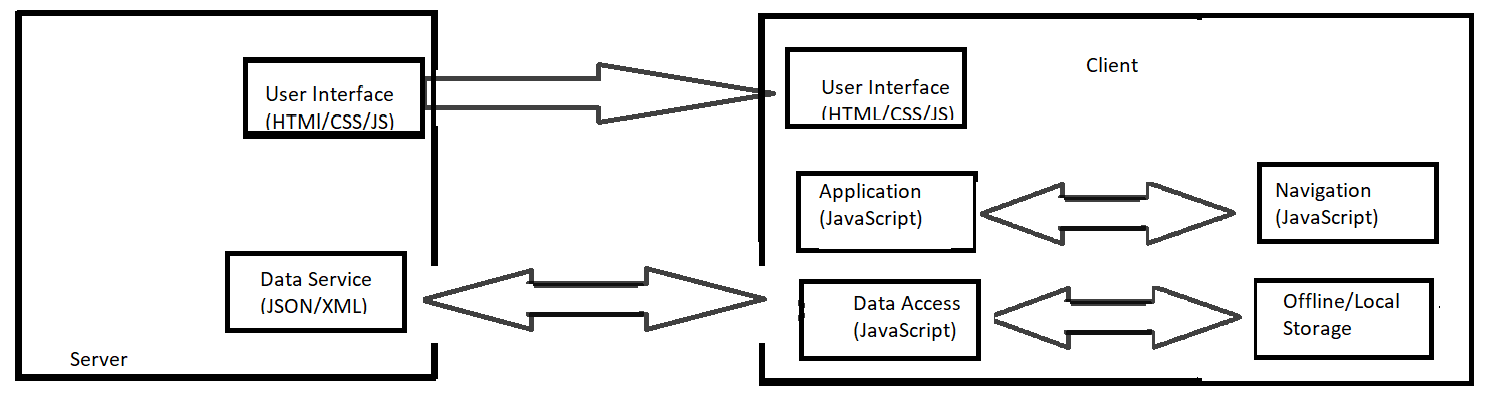
Using Online forums the users information can be managed by the database.

The system should be connected to the internet and having minimum software requirement.Modern Online forum is based on web-development -- the user can able to post their comment and queries.The time of posting is minimized based on the various factors and response time is also dependent.And it is ensured that no loss of data of the us

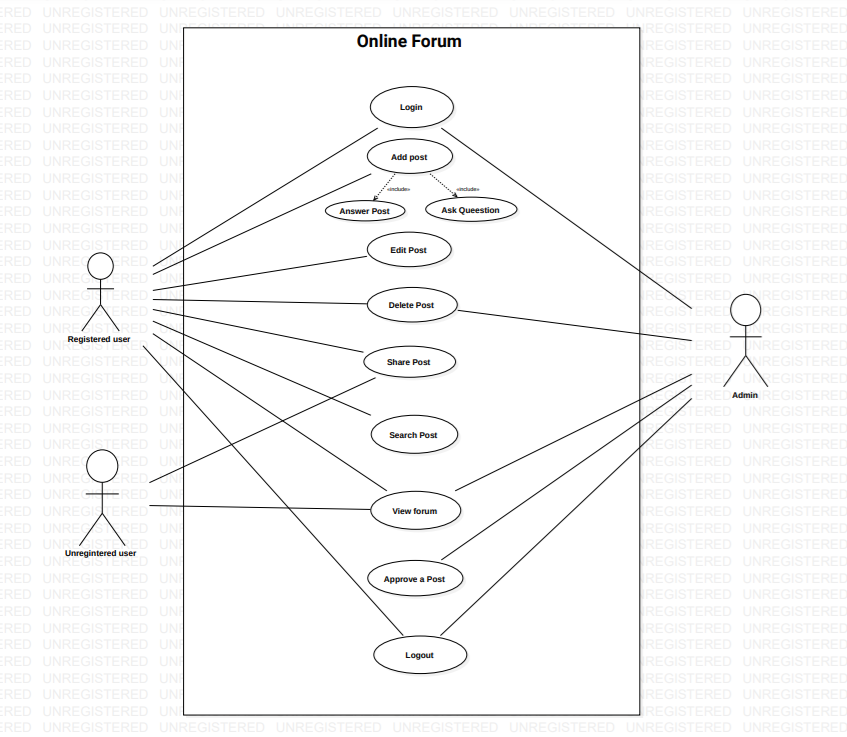
.

# Architecture

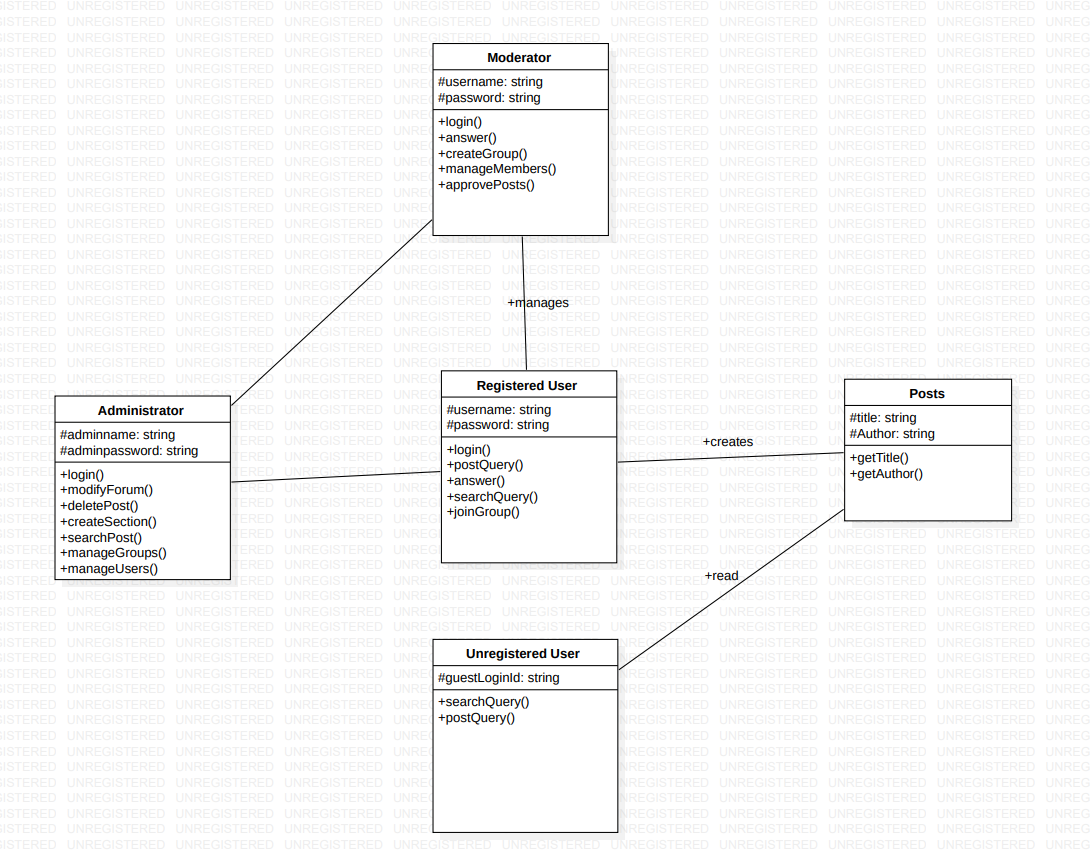
## Design

****

### Use Case Diagram:

****

### Class Diagram :

****

**Class :**

**Administrator:**

The administrator often makes decisions regarding web hosting, which is a service

that makes the forums accessible to visitors through the web.

**Moderator:**

A forum moderator oversees the communication activity of an Internet forum.

He monitors the interchange of contributors and makes decisions regarding

Content and the direction of threads.

**Posts:**

Posts are the queries posted by the registered user so that others registered users

can also view them.

**Unregistered User:**

The users who are not registered are unregistered users .They can only read the

posts without modifying them.

**Registered User:**

The users who are registered to the website are registered users .They can create

posts and interact with other users.

### 

### Server:

**Web Server**

The web server for the Online Forum System provides access for all authorized users to the Online Forum client Web application. To access the client application, users have to login to the web application and the system will spawn the main client application onto the client machine. Apache has been chosen as the web server that will handle authentication and client application delivery.

### Apache Server

The server communicates with the client application. The communication between the client and the server will be implemented using AJAX technology. An RMI registry will be running on the server in order to provide naming services to the remote objects.

### 

### MongoDb Database

The database provides all the necessary data that the Apache server requires in order to compute the correct report data. The schema and entries of the database are provided by the admin or by its officials . New data can be added to the database at any time. Data from the database can be retrieved by the api request.

**Client**:

The web client of the Online Forum handles authentication for the OF system, and initiation of the main thin-client application. The web client will communicate with the Apache web server through HTTP and is a combination of HTML and a simple JSON file. It is accessible by any machine that has a Java class library and common web browser installed.

### 

### 3.1.1 Version

Version 1.0 – Initial version of the system that has the basicarchitecture.

### 3.1.2 Server Roles

A server that is needed to connect with the application. The functionality of this server will not only handle the hosting of client software, it will also provide the only means of authentication. It runs on all major server operating systems like Windows 10 and Linux. The client makes a request to the server, the server will connect with the Database, and it will give the correct data according to their credentials to the client-side Machine.

**3.2 Access**

The system will be accessible through a Java thin client.A thin client is a **lightweight user-side application** that doesn’t need to process too many functions The client system will download and execute the Java thin client from the server side. The Java thin client will run on the client system and access the server side for functionality. The server also accesses another system for the mongoDb database.

**3.3 Hardware and Platform Requirements**

### Hardware Requirements:

### Operating System:

* + Windows version 8.0 or greater , 4 GB ram,Internet connectivity.

### Packages Installed:

* + Node.js-14.17.0 LTS -- <https://nodejs.org>
  + Angular-cli

## 

## 3.4 System Connectivity

The client will connect with the server using API. The API is the main interface through which all the transactions will be done and the necessary ports are available to both client and server.

# Standards

## Security Standards

### 4.1.1 Authorization and Logon

The system shall verify the username and password using Authentication, Authorization and Access Control of Apache.

## Disaster Recovery

Restart the server program as directed in the operational manual and the system will return to its last safe state.

# 5. Support

The following support documentation will be provided: Code, Design Document, Operations Manual, and Deployment Plan. The system code shall be documented according to the "Code Conventions for the Java Programming Language" available at https://angular.io/guide/styleguide.The system shall be described by a "Design Document." The system shall be accompanied by an "Operations Manual" describing proper use of the system. The system shall be deployed using operations described in the"DeploymentPlan.