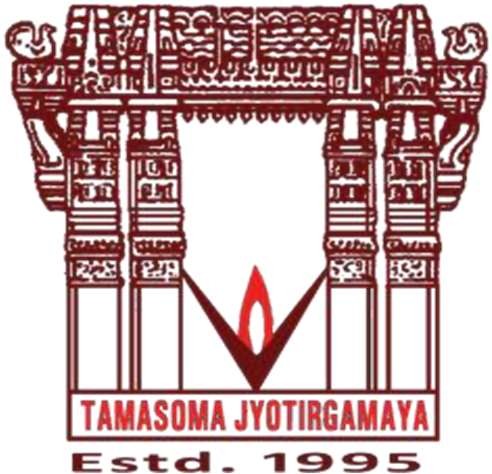


High Level Design Document

**On**

E-Book Management System



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

***Submitted By***

***Group Details:***

***Vinay Kumar – 19071A05H0***

***V.Tejaswini – 19071A05H9***

***K.H.S.Supriya – 19071A05E8***

***P.Akash – 19071A05G5***

***Rukmananda Reddy – 19071A05G6***

Contents

[Revision History 2](#_TOC_250027)

* 1. [Purpose 3](#_TOC_250026)
  2. [Audience 3](#_TOC_250025)
  3. [Design Process 3](#_TOC_250024)

1. [Requirements 4](#_TOC_250023)
   1. [Proposed Solution](#_TOC_250022) 4
   2. [Capacity Planning 5](#_TOC_250021)
2. [Architecture](#_TOC_250020) 5
   1. [Design](#_TOC_250019) 5

[3.1.1 Web Server](#_TOC_250018) 5

3.1.2 Web Server Software - Apache 5

3.1.3 Client 6

3.1.4 [Data Base - MongoDB](#_TOC_250016) 6

* 1. [Version](#_TOC_250019) 6
  2. [Server Roles](#_TOC_250019) 7
  3. [Access](#_TOC_250019) 7
  4. [Hardware Platform Requirements](#_TOC_250019) 7

3.5.1 [Hardware Requirements](#_TOC_250019) 7

3.5.2 [Packages Installed](#_TOC_250019) 7

* 1. [System Connectivity](#_TOC_250019) 8

1. [Standards](#_TOC_250004) 8
   1. [Security Standards](#_TOC_250003) 8

[4.3.1 Authorization and Login](#_TOC_250002) 8

* 1. [Disaster Recovery](#_TOC_250001) 8

1. [Support](#_TOC_250000) 8

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Date | Author | Section | Comments/Changes |
| 1.0.0 | 31 April | All | All | Initial Revision |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* 1. **Project Overview**

## Purpose

## The purpose of this document is to specify the high-level design for the E-BOOK MANAGEMENT SOFTWARE. This document will act as an outline for implementation and discuss the design considerations.

## Audience

## This High-level document is intended to be used by the members of the team who are going to implement the functionality of E-Book Management System. This will also help the people who try to develop the features from outside of the team.

## S.Vinay Kumar, Development Team

## K.H.S.Supriya, Development Team

## A.Rukmananda Reddy, Development Team

## P.Akash, Development Team

## V.Tejaswini , Development Team

## Design Process

## The High-level design was designed by us considering every aspect required for the software. We have gone through lot of discussion and several architectures were proposed but finally we decided to go with the architecture “Distributed System using Web Technology” by considering the advantages and disadvantages. Web services technology would allow the system to be less coupled and more cohesive.

## Nowadays it is the most used technology in the market as well. Communication between client and server will be done using sockets.

## After all this research the team decided to go with the architecture Distributed System using Web Technology.

# Requirements

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

# The system must be able to authenticate and give respective privileges to add books and buy them, navigate, etc.

# The server should be up and running at all times without a single point of failure. It should be able to support all kinds of loads.

# Checking the website on a regular interval basis will be done to detect eventual anomalies. The project should be well documented to help in maintenance.

## Proposed Solution

The proposed solution of our project involves providing services to the common people through an online book store. The system provides 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 searching a book, adding book to the user’s cart, buying a book etc. There will an express server running in the backend which fetched the data from MongoDB and an API will be developed via which the client interface will interact with the server.

## Capacity Planning

## The server capacity planning is also being done where the decisions have been such that the server should be up and running at all times and will support approx. 75 transactions at a given period with a sever load of approx. 60. We will currently be employing a 4 TB shared sever. But it is to be noted that as our requirements change, we will upgrade to a server with a better plan.

# Architecture:

## Design

## 

**ARCHITECTURE:**

DATABASE

MongoDB

**HTTP REQUEST**

WEB SERVER

APACHE

USER

**HTTP RESPONSE JSON FORMAT**

### 3.1.1 Web Server

### The web server provides an access for all authorized users to the application and fetches the data from the database. When users’ login to the application or search for a book or perform any action that involves data, the system will send appropriate authentication requests to the sever and based on the responses the user will be directed to the respective components.

### 3.1.2 Web Server Software – Apache

### Apache HTTP Server is the world’s most used web server software. Apache is an open-source project. Runs on all major server operating systems. Apache has been chosen as the web server that will host our website and handle client application delivery.

### 3.1.3 Client

### The web client of the E-book Management System handles authentication for the website, displays the books and shows the payment gateway interface. The web client will communicate with the server through an API. It is accessible by any machine that has a common web browser.

### 3.1.4 Data Base - MongoDB

### MongoDB is a document-oriented NoSQL database used for high volume data storage. We will be using MongoDB because, instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. We will be storing data as documents that consist of key-value pairs. Collections contain sets of documents and functions which is the equivalent of relational database tables. The schema and entries of the database are provided by the admin. New data can be added to the database at any time by the admin. And based on the API request the sever running on the database will send the data requested in JSON format.

## Version

## Version 1.0 – Initial version of the product / application

## Version 2.0 – Considering the changes requested by clients, users, improving UI more creatively and fixing bugs if found.

## MAJOR: Significant change to UI or code and/or structure

## MINOR: New features

## PATCH: Bug Fixes

## Server Roles

## Apache –

## A web server that is needed to host the application. The functionality of this server will not only handle the hosting of application, it will also provide the only means of authentication.

## It runs on all major server operating systems.

## The client made a request to the server, the server needs to connect with the Data Base,

## And it needs to give the correct data according to their credentials to the client-side Machine.

## Access

## The system will be accessible through website which is client built with an Angular interface. The client can be setup by visiting the website through a web browser and the client communicated with the server through an API. Access to the website is controlled by authentication and admin has more privilleges than a normal user. The client will authenticate and access data from the DB using the API.

## Hardware and Platform Requirements

### 3.5.1 Hardware Requirements:

### A min of 64MB RAM is must, although a 512 MB is recommended. The computer should have a disk space of at least 150 MB. Any modern browser like chrome, fire fox etc. can be used.

### 3.5.2 Packaged Installed:

### Nodejs

### Angular-cli

## System Connectivity

The client will connect with the sever 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

### 4.3.1 Authorization and Logon

The system shall verify the username and password using Google OAuth Authentication.

## Disaster Recovery

Restart server program as directed in the operational manual and the system will return to 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 "Deployment Plan."