**Software Design Document**

**Smart Search for Education**

#### **Team SSE**

# K Manoj Kumar U101114FCS187

# Vinit Koshti U101114FCS217

# Jagrit Kaushik U101114FCS205

# Jeel Shah U101114FCS172

# Mansi Singh U101114FCS090

#### **NIIT University - Software Engineering**

# Table of contents

1. **Introduction**

**1.1 Purpose of this document**

**1.2 Scope of the development project**

**1.3 Definitions, acronyms, and abbreviations**

**1.4 References**

**1.5 Overview of document**

1. **System architecture description**

**2.1 Overview of modules / components**

**2.2 Structure and relationships**

**2.3 User interface issues**

1. **Detailed description of components**

**3.1 Component template description**

**3.2 X Component (or Class or Function ...)**

**3.3 Y Component (or Class or Function ...)**

**3.n Z Component (or Class or Function ...)**

**4.0 Reuse and relationships to other products**

**5.0 Design decisions and tradeoffs**

**6.0 Pseudo code for components**

**7.0 Appendices (if any)**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to describe the implementation of the “Smart Search for Education” whose requirements have been described in detail in the SRS document submitted before.

This document will provide guidelines related to the structure and the design of the project and will contain the following too.

Component Diagram : information about the external and internal components of the system.

Class Diagram : specific information about the expected input, output, classes, and functions. The interaction between the classes to meet the desired requirements.

Sequence Diagram : specific information about how objects operate with one another and in what order.

**1.2 Scope of the Development Project**

This webapp is intended to perform unstructured dynamic search on learning course materials useful for learners to easily locate the topics in loads of learning material uploaded. This software will be designed to maximise the productivity of learners by providing them most recommended search according to their request. The webapp will be very simple to understand and use, keeping the efficiency consistent.

The webapp will be using OpenSource Optical Character Recognition Software ***Tesseract*** and Search algorithms from ***Elasticsearch*** to display the most recommended suggestion according to the search keywords given by the user. The interface will be very simple and easy to understand like the Google Search.

**1.3 Definitions, acronyms, and abbreviations**

|  |  |
| --- | --- |
| Term | Description |
| User | Any person who is interacting with the application is a *user*. |
| System | The package of all the components which takes input and gives output to demonstrate the features of the application is called System. |
| Database | Collection of information on different topics related to each other. |
| Store | This is the persistence layer of whole system. |

**1.4 References**

* IEEE. IEEE Standard 1016 IEEE Recommended Practice for Software Design Specifications. IEEE Computer Society, 1998.
* [Tesseract-OCR](https://github.com/tesseract-ocr) - to extract text from images.
* [ElasticSearch](http://www.elastic.co/) - Developed on Apache Lucene (Search).

**1.5 Overview**

This document is divided into sections 2, 3, 4, 5, 6 and 7 with intended readers being, the developers and software managers but sections have been written in a manner that it can be understood by anyone having little knowledge about software.

**2. System architecture description**

**2.1 Overview of modules / components**

This subsection will introduce the various components and subsystems.

This project uses open source APIs and are being efficiently integrated in our web application. Elastic Search is the main API used in this project

The entire application is java/javascript based application and there are various modules and component within this Elastic Search API.

Modules: CSS module - user interface styles,

Javascripts module - main functionality of the app, files like main.jsp, search.jsp, upload.jsp etc.

Elastic Search is being installed on the host machine/server

This javascript file interact with the user interface to take input and to generate results.

**2.2 Structure and relationships**

A simple finite state machine is useful in demonstrating the operation of the product. state.png

User search for query, results will be displayed, to search another query replace it with the another query in the search box, this process can be repeated.

As the search results are obtained, if found useful user can download from the result link or can add to library as an direct access to view document on web.

**2.3 User interface issues**

User interface will be simple keeping in mind hassle free user experience

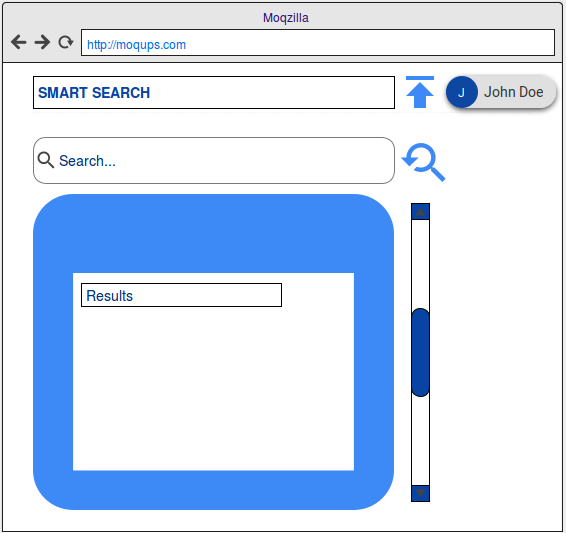
It will appear like normal search page in top right corner your logged in info and on its side there will be an upload option, as it is assume that we are free to upload documents.

A simple search bar will be present at the centre of the page, when the user enters the search query, the result will be the download link of the related document found in the database, user will have an option whether to download or add it to library that is user can open document in another web page and read.

This result will be displayed below the search bar and will have scroll bar, to view more than three results

In the top right corner on pressing the upload button, add file dialog box will appear, user has to select file to upload and done. Uploaded file will be indexed by the application and will appear on next searches.

Top right corner also contain your login information.



**3. Detailed description of components**

**4. Reuse and relationships to other products**

* How reuse is playing a role in your product design

Reuse plays a major role in our product design. The smart search is based on two main components: Tesseract and Elastic search algorithm. They are open source and thus we are reusing a code which has already been generated. Tesseract is used for optical character recognition for extracting text from images. This is necessary because search can only be applied on text and the books can be in an image format too. Elastic search algorithms are used to display the most relevant suggestions available in the database.

* How reuse is playing a role in your product implementation (and the motivation for changes)
* We have reused LAMP server for our project implementation which is the base of our project and thus plays a major part. Apache Lucene is also used for development of ElasticSearch algorithm.

**5. Design decisions and tradeoffs**

Our team is good at Front-end languages HTML, CSS and JS for the UI. At backend we are writing in PHP using MySql database.

We are using open-source libraries Tesseract for Optical Character Recognition and Elasticsearch for unstructured search.

Elasticsearch is a distributed RESTful search engine built for the cloud.

This package contains an OCR engine - libtesseract and a command line program - tesseract.

Developers use libtesseract [C](https://github.com/tesseract-ocr/tesseract/blob/master/api/capi.h) or [C++](https://github.com/tesseract-ocr/tesseract/blob/master/api/baseapi.h) API to build their own application.

**6. Pseudo code for components**

**7. Appendices (if any)**