



INTERNSHIP PROGRAM 2023

SOFTWARE DESIGN SPECIFICATION

Web Development

Campus Search

Created By:	Atharv Musale	Approved By:	
Created On:	20-08-2024	Approved On:	23-07-2024

INDEX

1	PURPOSE	2
2	PROJECT SCOPE.....	2
3	SYSTEM OVERVIEW.....	2
4	DESIGN CONSIDERATIONS	2
4.1	Requirements	2
4.2	Assumptions.....	3
4.3	Dependencies.....	3
5	SYSTEM ARCHITECTURE.....	3
5.1	Architectural Strategies	4
5.2	Structure & Relationships	4
6	DETAILED DESCRIPTION OF COMPONENTS	5
7	INTEGRATION.....	5
8	APPENDICES	1
8.1	Appendix A – Detailed Description of Components.....	1

General Instructions for using the Live Project POC Document

- This template and the subsequent document created using this template is a confidential document and is the intellectual property of Cloud Counselage Pvt. Ltd. Circulating it outside of the organisation without the consent of Cloud Counselage Pvt. Ltd. is the breach of company policies and will lead to legal actions
- The Design Specification of a software forms the basis of development of software
- The **text between inequality (< >)** is to be replaced by relevant text
- Please **remove the yellow highlight on the Text** between the inequality (< >). This is done to help you notice the text to be changed/replaced
- The text in *italics* highlighted in grey is just for reference and should be removed after adding the relevant text

1 PURPOSE

This document is created based on the requirement specification document. The purpose of this Software Design Specification (SDS) Document is to break down the project into components to describe in detail what the purpose of each component is and how it will be implemented. The SDS will also serve as a tool for verification and validation of the final product.

2 PROJECT SCOPE

The Campus Search Application provides students with an intuitive platform to search for colleges in India, focusing specifically on IT and management programs. The system allows users to search and view detailed information about colleges, including course offerings, campus facilities, and more. The system aims to solve the problem of scattered information by aggregating reliable college data into a single, user-friendly interface.

3 SYSTEM OVERVIEW

This section will provide an outline of the various components and subsystems of Campus Search:

- Home Page (index.html): Provides a general overview and access to the search feature.
- Results Page (result.html): Displays search results, showing colleges matching user-entered criteria.
- Contact Page (contact.html): Allows users to contact support or ask for more information.

Additional features include filtering and sorting options, a responsive UI, and images of campuses stored in the database.

4 DESIGN CONSIDERATIONS

This section describes requirements, assumptions and dependencies to be addressed to devise a complete design solution.

4.1 Requirements

The system requires:

- College data (including names, courses, and contact information).
- Responsive design to ensure compatibility across different devices.
- Integration of a database to store college details and images.

4.2 Assumptions

- College data will be regularly updated.
- The application will handle large datasets efficiently.
- The user interface should be intuitive and easy to navigate.

4.3 Dependencies

The project depends on reliable data sources and an efficient hosting solution for the web application.

5 SYSTEM ARCHITECTURE

The software system architecture refers to the logical organization of a distributed system into software components. It defines how components of a software system are assembled, their relationship and communication between them. It serves as a blueprint for software application and development basis for developer team. An effective architecture serves as the conceptual glue that holds every phase of the project together for all of its stakeholders, enabling agility, time and cost savings, and early identification of design risks.

The Software architecture:

- Defines structure of a system
- Defines behaviour of a system
- Defines component relationship
- Defines communication structure
- Balances stakeholder's needs
- Influences team structure
- Focuses on significant elements
- Captures early design decisions

Below some important characteristics which are commonly considered are explained.

Operational Architecture Characteristics:

- Availability
- Performance
- Reliability
- Low fault tolerance
- Scalability

Structural Architecture Characteristics:

- Configurability
- Extensibility
- Supportability
- Portability

- Maintainability

Cross-Cutting Architecture Characteristics:

- Accessibility
- Security
- Usability
- Privacy
- Feasibility

5.1 Architectural Strategies

The Campus Search Application follows a client-server architecture, using web technologies such as HTML, CSS, JavaScript, and a SQL-based backend for data storage. The architectural strategies applied to this project ensure efficient data handling, a responsive user interface, and scalability. The system is divided into three layers: Presentation Layer, Application Logic Layer, and Data Layer.

- **Presentation Layer:** This is the front-end of the system, developed using HTML, CSS, and JavaScript. The index.html file acts as the entry point, allowing users to input their search queries. The results are dynamically displayed on the results.html page based on the input. A responsive design ensures that the application works across different devices and screen sizes.
- **Application Logic Layer:** JavaScript is used in this layer to handle user interactions, search functionality, and display logic. It controls the flow of data between the front-end and back-end, ensuring that users receive results that are both accurate and presented in a user-friendly manner. This layer also manages filtering, sorting of search results, and validation on the contact.html page.
- **Data Layer:** A SQL-based database stores all information related to colleges, including their names, locations, courses offered, and contact information. This layer is responsible for providing search results to the application logic layer based on the queries received from users.
- **Key Architectural Features:**
 1. **Separation of Concerns:** The system follows the principle of separating presentation, logic, and data management concerns, ensuring that each layer can be modified without affecting others.
 2. **Scalability:** The system is designed to accommodate large datasets, with the ability to easily extend to include additional colleges or more detailed information such as fees, rankings, etc.

5.2 Structure & Relationships

- **Home Page:** Contains a search bar and brief introduction to the platform.

- **Results Page:** Displays search results dynamically based on the entered query.
- **Contact Page:** Handles form submissions for user inquiries.

6 DETAILED DESCRIPTION OF COMPONENTS

For detailed description of the components, please refer **Appendix A – Detailed Description of Components**

The below template will be used to specify the details of all the components

Table 1: Detailed Design Specification Template

Identification	Campus Search Module
Type	Module (JavaScript functions and HTML form).
Purpose	The Campus Search Module implements the search functionality that allows users to find colleges based on the name or courses offered. It retrieves information from the database and displays the results in a user-friendly format on the result.html page.
Subordinates	Search Input Form: HTML form on index.html for user queries. Search Handler: JavaScript function in script.js to process input and query the database. Result Display: Dynamic rendering of results on result.html.
Dependencies	This module has several dependencies that it relies on to function correctly. It depends on the backend database for storing and retrieving college information. The module interacts with user interface elements such as the input form and the results display section. Additionally, it utilizes other JavaScript functions for data processing and dynamically rendering the results on the webpage
Interfaces	User Interface: Collects input and displays results or error messages. Database Interface: Queries the database to fetch relevant information. Screen Formats: Input form and results page with dynamic content.
Resources	Requires client devices (PCs, tablets, smartphones) and a web hosting environment. Uses front-end technologies (HTML, CSS, JavaScript)
Processing	The user inputs a search query on index.html. The JavaScript function processes this query, retrieves data from the database, and displays the results on result.html. If no match is found, an error message is shown.
Data	The database holds details such as the college name, address (city), and contact number. This data is queried using the search input provided by the user. The initial dataset is pre-populated to facilitate search operations, ensuring quick and reliable access to college information. The search query strings are matched against the relevant database fields to retrieve and display results in an intuitive format.

7 INTEGRATIONS

The application requires integration with a database that contains comprehensive information about colleges. Additionally, image files of the colleges should be stored in a separate folder. Possible future integrations may include third-party services like Google Maps for campus locations or APIs for real-time college rankings.

8 APPENDICES

8.1 Appendix A – Detailed Description of Components

Identification	Home Page
Type	HTML Page
Purpose	The Home Page allows users to search for colleges by entering search terms such as college name, course, and location.
Subordinates	- Results Page (result.html) - Contact Page (contact.html)
Dependencies	Requires backend services to handle search queries and fetch results.
Interfaces	The page contains a search bar and submit button. Navigates to Results Page upon submission.
Resources	Requires a connection to the database for fetching college information and a frontend JavaScript handler for form submission.
Processing	On submission, JavaScript captures the search input, validates the field, and submits the request to the backend to retrieve the results.
Data	The input consists of search keywords entered by the user, which is then passed to the backend for query execution.

Identification	Results Page
Type	HTML Page
Purpose	Displays search results based on user queries entered on the Home Page. Users can browse through a list of colleges relevant to their search terms.
Subordinates	Contact Page (contact.html)
Dependencies	Requires backend service for fetching search results.
Interfaces	Search result cards, pagination, and filters for additional search refinement.
Resources	Backend service for retrieving filtered data, JavaScript for dynamically rendering the results.
Processing	The backend processes the search query and returns a list of colleges that match the user's criteria, which is displayed dynamically via JavaScript.
Data	College data such as name, location, and available courses are retrieved from the backend database based on the user's search.

Identification	Contact Page
Type	HTML Page
Purpose	Allows users to submit their inquiries or feedback through a contact form.
Subordinates	Home Page
Dependencies	Requires backend service to handle form submissions and send the data to appropriate storage (e.g., database or email).
Interfaces	Contact form with fields like name, email, and message. A submit button triggers form submission.
Resources	JavaScript for form validation and backend service for handling submitted data.
Processing	JavaScript validates form fields (name, email, message), then sends the data to the backend for processing. A confirmation message is shown upon successful submission.
Data	User input such as name, email, subject, and message are validated and sent to the backend.