



SOFTWARE DESIGN SPECIFICATION

WEB DEVELOPMENT  
  
COURSECONNECT

|  |  |  |  |
| --- | --- | --- | --- |
| **Created By:** | Nidhi Krishna Pednekar | **Approved By:** | <Domain Lead Name> |
| **Created On:** | 19-06-2024 | **Approved On:** | DD-MMM-YYYY |

Page left blank intentionally

**INDEX**

[**1** **PURPOSE** 2](#_Toc142418236)

[**2** **PROJECT SCOPE** 2](#_Toc142418237)

[**3** **SYSTEM OVERVIEW** 2](#_Toc142418238)

[**4** **DESIGN CONSIDERATIONS** 2](#_Toc142418239)

[4.1 Requirements 3](#_Toc142418240)

[4.2 Assumptions 3](#_Toc142418241)

[4.3 Dependencies 3](#_Toc142418242)

[**5** **SYSTEM ARCHITECTURE** 3](#_Toc142418243)

[5.1 Architectural Strategies 4](#_Toc142418244)

[5.2 Structure & Relationships 4](#_Toc142418245)

[**6** **DETAILED DESCRIPTION OF COMPONENTS** 4](#_Toc142418246)

[**7** **INTEGRATION** 5](#_Toc142418247)

[**8** **APPENDICES** 1](#_Toc142418248)

[8.1 Appendix A – Detailed Description of Components 1](#_Toc142418249)

**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

# **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.

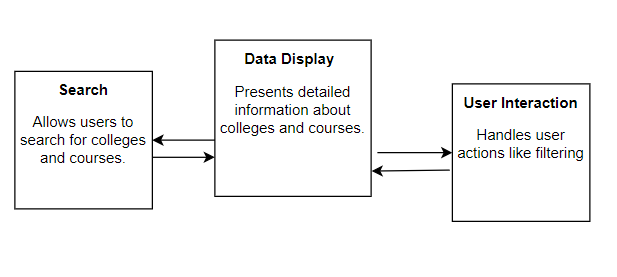
# **PROJECT SCOPE**

The scope of the CourseConnect includes its distinct features, its benefits, and its limitations. The system's distinct features allow it to facilitate college and course searches to students by using Express,MySQL.The system enables the user to Improved access to college information and enhanced search and filtering capabilities.

# **SYSTEM OVERVIEW**

This section will provide an outline of the various components and subsystems of CourseConnect.

CourseConnect simplifies college and course discovery for students with an intuitive platform that offers detailed insights into colleges and courses. It is designed to be inclusive and accessible, supporting students in making informed decisions and achieving academic success.



# **DESIGN CONSIDERATIONS**

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

## Requirements

Hardware Requirements:

Server Type:- Dedicated server or cloud-based virtual machine. Processor:-Minimum 4 CPU cores (e.g., Intel Xeon, AMD Ryzen). Memory (RAM):-Minimum 8 GB RAM, recommended 16 GB for improved performance. Storage:-Minimum 100 GB SSD storage for the database and related files, Network:- High-speed internet connection with a minimum bandwidth of 100 Mbps.

Software Requirements:

IDE: Visual Studio Code. Programming languages : HTML, CSS, JavaScript for front-end, and Express.js for back-end development. DBMS : MySQL. Browser :Google Chrome.

## Assumptions

IT or management colleges will be searched.

## Dependencies

(MySQL) for storing and retrieving data, a backend framework (Express.js) for server-side logic.

# **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:**

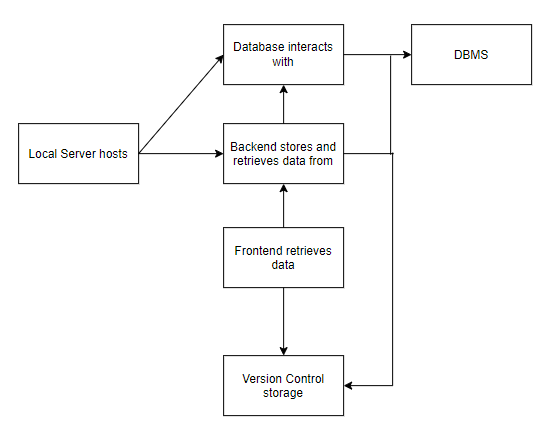
* Accessibility
* Security
* Usability
* Privacy
* Feasibility

## Architectural Strategies

CourseConnect can be broken down into the following major architectural components:

* **Frontend:** HTML, CSS, JavaScript for user interface design and interaction.
* **Backend:** Express.js for server-side application logic and routing.
* **Database:** MySQL for storing and managing data.
* **Local Server:** XAMPP for hosting the backend locally
* **Version Control:** GitHub for version control.

## Structure & Relationships



# **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** | The unique name for the component and the location of the component in the system. |
| **Type** | A module, a subprogram, a form, a data file, a control procedure, a class, etc. |
| **Purpose** | Function and performance requirements implemented by the design component, including derived requirements. Derived requirements are not explicitly stated in the SRS - but are implied or adjunct to formally stated SDS requirements. |
| **Subordinates** | The internal structure of the component, the constituents of the component, and the functional requirements satisfied by each part. |
| **Dependencies** | How the component’s function and performance relate to other components. How this component is used by other components. The other components that use this component. Interaction details such as timing, interaction conditions (such as order of execution and data sharing), and responsibility for creation, duplication, use, storage, and elimination of components. |
| **Interfaces** | Detailed description of all external or internal interfaces as well as of any mechanism for communicating through messages, parameters, or common data areas. All error messages and error codes should be identified. All screen formats, interactive messages, and other user interface components (originally defined in the SRS) should be given here. |
| **Resources** | A complete description of all resources (hardware or software) external to the component but required to carry out its functions. |
| **Processing** | A full description of the functions presented in the Function subsection. Pseudocode can be used to document algorithms, equations, and logic. |
| **Data** | For the data internal to the component, describes the representation method, initial values, use, semantics, and format. |

# **INTEGRATIONS**

CourseConnect will integrate with XAMPP for local hosting and managing of the database and server environment. XAMPP provides the necessary infrastructure for MySQL database management and PHP server-side scripting, ensuring seamless data storage and retrieval. Additionally, Git version control via GitHub will be utilized for collaborative development and code management*.*

# **APPENDICES**

## Appendix A – Detailed Description of Components

|  |  |
| --- | --- |
| **Identification** | Frontend |
| **Type** | Module |
| **Purpose** | Handles user interface and interaction. |
| **Subordinates** | HTML, CSS, JavaScript files. |
| **Dependencies** | Backend APIs for data retrieval. |
| **Interfaces** | User interactions via browser. |
| **Resources** | Web server, browser resources. |
| **Processing** | Render UI elements, handle user events. |
| **Data** | User input data, rendered UI elements. |

|  |  |
| --- | --- |
| **Identification** | Backend Server |
| **Type** | Subprogram |
| **Purpose** | Implements server-side logic and APIs. |
| **Subordinates** | Express.js routes, middleware. |
| **Dependencies** | XAMPP (Apache, MySQL) for server and database |
| **Interfaces** | HTTP requests, database queries. |
| **Resources** | Server resources, database connection. |
| **Processing** | Process incoming requests, execute business logic. |
| **Data** | HTTP request data, database query results. |

|  |  |
| --- | --- |
| **Identification** | Database |
| **Type** | Data file |
| **Purpose** | Stores application data |
| **Subordinates** | Tables (Courses, Colleges, college\_courses) |
| **Dependencies** | Backend server for data retrieval and storage. |
| **Interfaces** | MySQL client interface. |
| **Resources** | Storage space, database management system |
| **Processing** | Store and retrieve structured data. |
| **Data** | College and course details. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Component** | | | Content | | |
| 1. PURPOSE | | | 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 scope of the CourseConnect includes its distinct features, its benefits, and its limitations. The system's distinct features allow it to facilitate college and course searches to students by using Express, MySQL. The system enables the user to Improved access to college information and enhanced search and filtering capabilities. | | |
| 3. SYSTEM OVERVIEW | | | CourseConnect simplifies college and course discovery for students with an intuitive platform that offers detailed insights into colleges and courses. It is designed to be inclusive and accessible, supporting students in making informed decisions and achieving academic success. | | |
| 4. DESIGN CONSIDERATIONS | | | This section describes requirements, assumptions, and dependencies to be addressed to devise a complete design solution. | | |
| 5. SYSTEM ARCHITECTURE | | | The section defines the software system architecture of CourseConnect and its key characteristics. | | |
| 6. DETAILED DESCRIPTION OF COMPONENTS | | | This section refers to Appendix A for detailed descriptions of each component. | | |
| 7. INTEGRATION | | | CourseConnect integrates with XAMPP for local hosting and managing the database and server environment. GitHub is used for version control to manage collaborative development and code management. | | |
| 8. APPENDICES | | | Appendix A – Detailed Description of Components | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
|  | | |  | | |
|  |  |  | |  |  |
|  |  |  | |  |  |