SOFTWARE DESIGN SPECIFICATION

INTERNSHIP PROGRAM 2023

<WEB DEVELOPMENT>

<COLLEGE LISTER>

Created By: <Ruchi Mankar>   Approved By: <Domain Lead Name>
Created On: 29- 09-2023   Approved On:DD-MMM-YYYY

Page left blank intentionally

INDEX

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
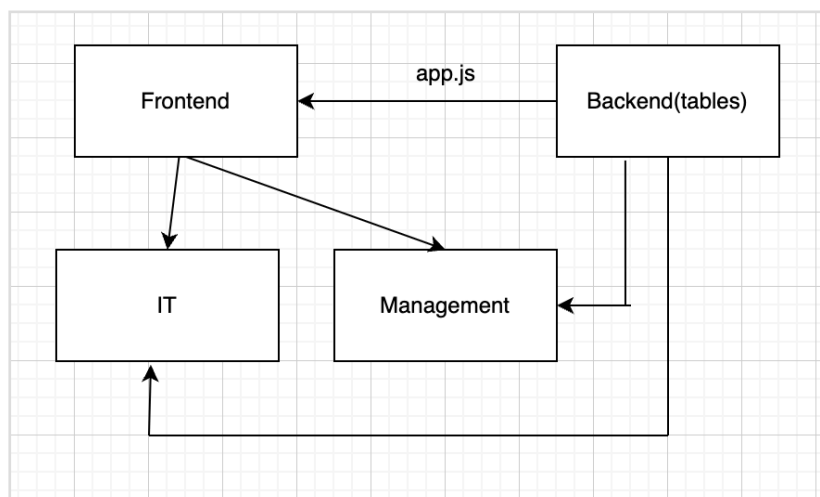
1. PURPOSE

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 scope of the College Lister includes its distinct features, its benefits, and its limitations. The system's distinct features allow it to search for colleges without any distractions.

3. SYSTEM OVERVIEW

This section will provide an outline of the various components and subsystems of College Lister

4. DESIGN CONSIDERATIONS

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

4.1. Requirements
Internet connection

4.2. Assumptions
Knows the course of interest and the eligibility

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

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	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.

7. INTEGRATIONS

e.g. The Campus Police Automated Ticketing System will be used interface with a Brother MW-140BT Mobile Printer in order to print tickets. The printer runs on a rechargeable lithium battery, allowing it to be carried anywhere. The printer has Bluetooth communication, enabling wireless communication with the PDA. The PDA can communicate with the printer once the printer's software driver has been installed on the PDA. Using the PDA the Brother MW-140BT can be set as the default printer. The Campus Police Automated Ticketing System will have an option to print a ticket. The ticket will then be transmitted wirelessly to the printer. The ticket will then be printed.

- The PrinterCE.NetCF Software Developer Kit (SDK) will be used to add printing functionality to mobile applications developed with the .Net Compact Framework.

- ADOCE In The Hand SDK will be used to enable read and write functionality to Windows CE Property Databases, usually called Pocket Access.

- The Campus Police Automated Ticketing System will be used to interface with a Microsoft Access Database currently being used by the Campus Police.

- Microsoft ActiveSync 3.8 is the software will be used to synchronize the database on the PDA with the primary computer-based database..

8. APPENDICES

8.1. Appendix A – Detailed Description of Components

For e.g.

Identification	*LoginScreen*
Type	*Class/Form*
Purpose	*The login screen assures that only officers and parking attendants can access the system.*
Subordinates	*This screen contains links to the following screens:* • *Main Menu Screen* • *New User Account Screen*
Dependencies	*The following screen links to this screen:* • *Main Menu Screen*
Interfaces	*The links are contained in the bottom half of the screen. The screen is designed to be easy to view using the resolution standard on the PDA.*
Resources	*Database Access Requirements: access to the violator information found in the appropriate database tables. Please see Appendix A for a description of the information is associated with a violator.*
Processing	*The only type of processing required is inputting information into the text boxes and navigating to other forms using links in the bottom half of the screen. Each link directs the user to a different screen that corresponds to the link that the user selects.*
Data	*The data for the Login Screen is the username and password entered by the user. It is validated with a query against the database.*

Identification	**<Component 1>**
Type	Class/Form/
Purpose	
Subordinates	
Dependencies	
Interfaces	
Resources	

Processing	
Data	

Identification	<mark>\<Component 2\></mark>
Type	Class/Form/
Purpose	
Subordinates	
Dependencies	
Interfaces	
Resources	
Processing	
Data	

<mark>\<Add component table as required\></mark>