**System Design Document**

for

**Campus Placer**

**Abhishek**

2217063

MCA

Aloysius Institute of Management and IT

Mangalore

21-06-2024

Under the guidance of

**Mr. Shine**

MCA Department

Aloysius Institute of Management and IT

Mangalore

Submitted to

****

**ST ALOYSIUS INSTITUTE OF MANAGEMENT AND INFORMATION TECHNOLOGY (AIMIT)**

**ST ALOYSIUS COLLEGE (AUTONOMOUS)**

**MANGALORE, KARNATAKA**

**1. Introduction**

**1.1 Purpose**

The purpose of the SDD is to provide a comprehensive design overview of the software system being developed. It serves as a blueprint that outlines the software structure, components, interfaces, and necessary data for the implementation phase. The intended audience includes developers, architects, testers, project managers, and stakeholders involved in the software development lifecycle.

**1.2 Scope**

This section outlines the relationship between the Software Requirements Specification (SRS) and the SDD. It specifies how the design document translates the requirements specified in the SRS into a detailed software architecture and design. The scope also defines the boundaries of the software system to be developed, including its functionalities, features, and constraints.

**1.3 Definitions, Acronyms, and Abbreviations**

SDD: Software Design Document

SRS: Software Requirements Specification

API: Application Programming Interface

GUI: Graphical User Interface

SQL: Structured Query Language

UML: Unified Modeling Language

HTTP: Hypertext Transfer Protocol

JSON: JavaScript Object Notation

UI: User Interface

UX: User Experience

**2. References**

List any documents that are referenced in the SDD. Include titles, report numbers, dates, publishing organizations, and sources where these references can be obtained. Examples may include:

* [Geekforgeeks](https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/?ref=lbp)
* [Tutorials point](https://www.tutorialspoint.com/uml/index.htm)
* [Draw.io](https://www.drawio.com/doc/faq/new-diagram)

**3. Attributes of Design Entities**

**3.1 Identification**

In the Campus Placer system, each entity is uniquely identified by its name to avoid any ambiguity. The entities are categorized based on the module they belong to. Below is a list of entities along with their unique names:

**Admin Module:**

* BranchManagement: Handles all operations related to managing branches.
* HODManagement: Manages Head of Departments (HODs) including view, edit, and delete operations.
* PlacementOfficerManagement: Manages placement officers with functionalities to view, edit, and delete.
* StudentManagement: Allows the admin to view student details.
* CompanyManagement: Enables the admin to view company details.
* JobCategoryManagement: Facilitates viewing of job categories.

**HOD Module:**

* StudentManagement: Manages students under a particular HOD including view, edit, and delete operations.

**Placement Officer Module:**

* JobCategoryManagement: Manages job categories including view, edit, and delete operations.
* CompanyManagement: Manages company details with functionalities to view, edit, and delete.
* StudentDetailsManagement: Allows the placement officer to view student details, check eligibility, and send email notifications.

**Student Module:**

* ResumeManagement: Allows students to add and update their resumes.
* JobManagement: Enables students to view and apply for jobs.

**3.2 Type**

The type attribute describes the nature of each entity in the Campus Placer system. Entities are categorized based on their functionality and role within the system.

**Admin Module**

* **BranchManagement**: Module
* **HODManagement**: Module
* **PlacementOfficerManagement**: Module
* **StudentManagement**: Module
* **CompanyManagement:** Module
* **JobCategoryManagement:** Module

#### HOD Module

* **StudentManagement:** Module

#### Placement Officer Module

* **JobCategoryManagement:** Module
* **CompanyManagement:** Module
* **StudentDetailsManagement:** Module

#### Student Module

* **ResumeManagement:** Module
* **JobManagement:** Module

**3.3 Purpose**

Each entity in the Campus Placer system serves a distinct purpose to fulfill specific functional requirements:

* BranchManagement: Manages the addition, viewing, editing, and deletion of branches within the campus.
* HODManagement: Facilitates the management of Head of Departments (HODs), including their addition, editing, viewing, and deletion.
* PlacementOfficerManagement: Manages the details of placement officers, including their addition, editing, viewing, and deletion.
* StudentManagement: Allows the admin to view student details across the campus.
* CompanyManagement: Enables the admin to view details of companies participating in campus placements.
* JobCategoryManagement: Allows the admin to view different job categories available for placements.

**HOD Module:**

* StudentManagement: Manages students under a particular HOD, including adding, editing, viewing, and deleting student details.

**Placement Officer Module:**

* JobCategoryManagement: Manages job categories, including adding, editing, viewing, and deleting categories.
* CompanyManagement: Manages company details, including adding, editing, viewing, and deleting companies.
* StudentDetailsManagement: Allows the placement officer to view student details, check eligibility for placements, and send email notifications.

**Student Module:**

* ResumeManagement: Allows students to add and update their resumes.
* JobManagement: Enables students to view and apply for jobs.

**3.4 Function**

The function of each entity involves specific operations to achieve desired outcomes:

* BranchManagement: Handles CRUD (Create, Read, Update, Delete) operations for branch information.
* HODManagement: Manages CRUD operations for HOD details.
* PlacementOfficerManagement: Manages CRUD operations for placement officer details.
* StudentManagement: Provides functionalities to view student information.
* CompanyManagement: Enables viewing of company information.
* JobCategoryManagement: Facilitates viewing of job categories.

**HOD Module:**

* StudentManagement: Provides CRUD operations for managing student details.

**Placement Officer Module:**

* JobCategoryManagement: Manages job categories through CRUD operations.
* CompanyManagement: Handles CRUD operations for company details.
* StudentDetailsManagement: Allows viewing student details, checking eligibility, and sending notifications.

**Student Module:**

* ResumeManagement: Manages resume addition and updates.
* JobManagement: Enables viewing and applying for job postings.

**3.5 Subordinates**

The subordinates attribute identifies the components that make up a higher-level entity, aiding in system decomposition:

* BranchManagement: Composed of branch details and related database handlers.
* HODManagement: Composed of HOD details and related database handlers.
* PlacementOfficerManagement: Composed of placement officer details and related database handlers.
* StudentManagement: Composed of student details and related database handlers.
* CompanyManagement: Composed of company details and related database handlers.
* JobCategoryManagement: Composed of job category details and related database handlers.

**3.6 Dependencies**

Dependencies describe how each entity interacts with other entities:

* BranchManagement: Depends on the DatabaseModule for data storage and retrieval.
* HODManagement: Depends on the AuthenticationModule for user verification and DatabaseModule for data management.
* PlacementOfficerManagement: Depends on the EmailNotificationModule for sending notifications and DatabaseModule for data management.
* StudentManagement: Depends on the DatabaseModule for accessing student information.
* CompanyManagement: Depends on the DatabaseModule for accessing company information.
* JobCategoryManagement: Depends on the DatabaseModule for accessing job category information.

**3.7 Interface**

The interface describes the methods and rules for interacting with an entity:

* BranchManagement: Methods include addBranch, editBranch, deleteBranch, and viewBranch.
* HODManagement: Methods include addHOD, editHOD, deleteHOD, and viewHOD.
* PlacementOfficerManagement: Mrthods include add edit and delete placement officer
* StudentManagement: Methods include viewStudent.
* CompanyManagement: Methods include viewCompany.
* JobCategoryManagement: Methods include viewJobCategory.

**3.8 Resources**

Resources identify all external needs for the entity to function properly:

* BranchManagement: Requires database servers, web servers, and adequate CPU/memory for processing.
* HODManagement: Requires access to a database, mail servers for notifications, and authentication services.
* PlacementOfficerManagement: Requires email services for notifications, database servers for data management.

**3.9 Processing**

Processing describes the rules and algorithms used by the entity:

* BranchManagement: Validates branch details, checks for duplicates, performs CRUD operations.
* HODManagement: Validates HOD details, ensures authentication, performs CRUD operations.
* PlacementOfficerManagement: Validates placement officer details, manages CRUD operations, sends notifications.

**3.10 Data**

Data describes how information is represented and managed within the entity:

* BranchManagement: Represents branch information in JSON format, initializes with empty records, stores branch details.
* HODManagement: Represents HOD information in JSON format, initializes with empty records, stores HOD details.
* PlacementOfficerManagement: Represents placement officer information in JSON format, initializes with empty records, stores placement officer details.

**4. Decomposition Description**

**4.1 General Structure**

The Campus Placer is structured into distinct modules, each comprising various entities that perform specific functions. This section outlines the main modules and their respective entities, detailing their purpose and functionality within the system.

**Admin Module**

* BranchManagement: Manages the operations related to branches within the campus, including adding, viewing, editing, and deleting branch details.
* HODManagement: Handles the management of Head of Departments (HODs), enabling the addition, viewing, editing, and deletion of HOD details.
* PlacementOfficerManagement: Manages placement officers' details, allowing the admin to add, view, edit, and delete placement officers.
* StudentManagement: Provides the admin with the ability to view student information across the campus.
* CompanyManagement: Enables the admin to view details of companies participating in the campus placement program.
* JobCategoryManagement: Allows the admin to view different job categories available for placements.

**HOD Module**

* StudentManagement: Empowers HODs to manage student details within their department, including adding, viewing, editing, and deleting student records.

**Placement Officer Module**

* JobCategoryManagement: Manages job categories, including adding, viewing, editing, and deleting job categories.
* CompanyManagement: Handles the addition, viewing, editing, and deletion of company details related to campus placements.
* StudentDetailsManagement: Allows placement officers to view detailed student profiles, assess eligibility for job applications, and send email notifications to students.

**Student Module**

* ResumeManagement: Allows students to add, update, and manage their resumes within the system.
* JobManagement: Enables students to view job listings, apply for jobs, and track their application status.

**4.2 Procedural Approach**

**Admin Module**

The Admin Module in the Campus Placer system comprises the Branch, Placement officer, student, job category, and HOD management. Each of these modules handles specific administrative tasks such as adding, editing, deleting, and viewing branches, HODs, placement officers, students, companies, and job categories. These modules interact primarily with the DatabaseModule for data storage and retrieval and may also communicate with the AuthenticationModule for user verification and the EmailNotificationModule for sending notifications.

**HOD Module**

The HOD Module focuses exclusively on StudentManagement. This module allows HODs to manage student details within their respective departments. Operations include adding, editing, viewing, and deleting student information. This module interacts with the DatabaseModule to access and modify student records, ensuring that HODs have the tools needed to maintain accurate and up-to-date student information.

**Placement Officer Module**

The Placement Officer Module includes JobCategoryManagement, CompanyManagement, and StudentDetailsManagement. These modules enable placement officers to manage job categories, company details, and student eligibility for placements. Placement officers can add, edit, view, and delete job categories and company details and view student information to check placement eligibility and send email notifications. This module interacts with the DatabaseModule for data management and the EmailNotificationModule for communication with students.

**Student Module**

The Student Module consists of ResumeManagement and JobManagement. These modules allow students to manage their resumes and apply for jobs. Students can add and update their resumes and browse and apply for job postings. The Student Module interacts with the DatabaseModule to store and retrieve resume data and job application details, ensuring students can efficiently manage their placement-related activities.

**4.2.1 Module Decomposition**

The Campus Placer project follows a modular design, with each module handling specific responsibilities within the system. This section outlines the basic modules, their interactions, and the procedural flow of operations.

**Admin Module**

The Admin Module is the core control center of the Campus Placer system. It handles administrative tasks and manages other modules, ensuring seamless coordination and data integrity.

1. BranchManagement

Description: Manages branch-related operations, including adding, viewing, editing, and deleting branches.

Interactions: Interfaces with the database to store and retrieve branch information. Calls methods in the DatabaseModule for CRUD (Create, Read, Update, Delete) operations.

1. HODManagement

Description: Manages the details of Head of Departments (HODs). Allows for adding, viewing, editing, and deleting HOD records.

Interactions: Communicates with the AuthenticationModule for user verification and the DatabaseModule for data management.

1. PlacementOfficerManagement

Description: Manages placement officer information. Facilitates adding, viewing, editing, and deleting placement officer records.

Interactions: Utilizes the EmailNotificationModule to send notifications and the DatabaseModule for CRUD operations.

1. StudentManagement

Description: Provides functionalities for viewing student details.

Interactions: Retrieves student data from the DatabaseModule.

1. CompanyManagement

Description: Manages company details related to campus placements. Allows for viewing company profiles.

Interactions: Accesses the DatabaseModule to fetch company information.

1. JobCategoryManagement

Description: Manages job categories available for placements. Supports viewing of job categories.

Interactions: Interfaces with the DatabaseModule to handle job category data.

**HOD Module**

The HOD Module is designed for department heads to manage students within their respective departments.

1. StudentManagement

Description: Manages student details under a specific HOD. Includes functionalities for adding, viewing, editing, and deleting student records.

Interactions: Uses the DatabaseModule to perform CRUD operations on student data.

**Placement Officer Module**

The Placement Officer Module facilitates the management of job categories, companies, and student eligibility for job applications.

1. JobCategoryManagement

Description: Handles job category operations, including adding, viewing, editing, and deleting categories.

Interactions: Communicates with the DatabaseModule for CRUD operations.

1. CompanyManagement

Description: Manages company details for placements. Facilitates adding, viewing, editing, and deleting company profiles.

Interactions: Interfaces with the DatabaseModule for company data management.

1. StudentDetailsManagement

Description: Allows placement officers to view student details, check eligibility for job applications, and send notifications.

Interactions: Utilizes the DatabaseModule to access student data and the EmailNotificationModule to send emails.

**Student Module**

The Student Module provides functionalities for students to manage their resumes and apply for jobs.

1. ResumeManagement

Description: Enables students to add, update, and manage their resumes.

Interactions: Stores and retrieves resume data through the DatabaseModule.

1. JobManagement

Description: Allows students to view job listings and apply for positions. Tracks application status.

Interactions: Interfaces with the JobCategoryManagement and CompanyManagement entities to display job opportunities and the DatabaseModule for application data management.

**4.2.2 Data Decomposition**

The Campus Placer project involves various data elements that are essential for the functioning of different modules. This section describes the decomposition of these data elements, detailing their structure, relationships, and how they are managed within the system.

**Admin Data Elements**

Admin data elements include information related to branches, HODs, placement officers, students, companies, and job categories. Each of these elements has attributes that define their properties. For example, a branch data element includes attributes such as branch ID, branch name, and branch description. These data elements are stored in the database and are accessed and modified by the Admin Module through CRUD operations.

**HOD Data Elements**

HOD data elements primarily involve student information within a specific department. Each student data element includes attributes such as student ID, name, department, year of study, and academic records. These elements are managed by the HOD Module, which allows for adding, viewing, editing, and deleting student records. The data is stored in the database and is accessed by the HOD Module to maintain accurate and up-to-date student information.

**Placement Officer Data Elements**

Placement officer data elements include job categories, company profiles, and student eligibility information. Job category data elements have attributes such as category ID, category name, and description. Company data elements include company ID, company name, industry type, and contact details. Student eligibility data elements involve attributes like student ID, eligible job categories, and application status. These data elements are managed by the Placement Officer Module, which ensures that job listings and company profiles are kept current and that students' eligibility status is accurately tracked.

**Student Data Elements**

Student data elements consist of resume information and job application records. Resume data elements include student ID, resume content, and last updated timestamp. Job application data elements involve attributes such as application ID, job ID, student ID, application date, and application status. These elements are managed by the Student Module, allowing students to maintain their resumes and track their job applications. The data is stored in the database and accessed by the Student Module to facilitate job searches and application processes.

**Database Management**

All data elements are stored in a centralized database managed by the DatabaseModule. This module handles the creation, retrieval, updating, and deletion of data elements, ensuring data integrity and consistency. It interacts with other modules to perform these operations and maintains relationships between different data elements. For instance, a student's eligibility for a job category is linked through their student ID, which is a key attribute in both the student and eligibility data elements.

**Authentication Data Elements**

Authentication data elements include user credentials and roles, such as user ID, username, password, and role (admin, HOD, placement officer, student). These elements are managed by the AuthenticationModule, which verifies user identities and controls access to various functionalities within the system.

**Email Notification Data Elements**

Email notification data elements consist of email templates, recipient details, and send status. Attributes include template ID, subject, body content, recipient email address, and status (sent, pending, failed). These elements are managed by the EmailNotificationModule, which sends notifications to students regarding job applications and other relevant information.

**4.3 Object-Oriented Approach**

**4.3.1 Use Case Diagrams**

**4.3.2 Class Diagrams**

**4.3.3 Sequence Diagrams**

**4.3.4 State Chart Diagrams**

**4.3.5 Activity Diagrams**

**5. Dependency Description**

In the Campus Placer project, various modules and data elements depend on each other to function cohesively.

The admin functions depend on the database for all CRUD operations involving branches, HODs, placement officers, students, companies, and job categories, and on the authentication system to verify user credentials.

HOD functions rely on the database to manage student records, ensuring data is accurate and up-to-date.

Placement officer functions depend on the database for managing job categories, company profiles, and student eligibility data, and on the email notification system for sending updates to students.

Student functions rely on the database for managing resumes and job applications, allowing students to maintain and track their job search activities.

The database module underpins all other modules, ensuring reliable data storage and retrieval.

The authentication system controls access to different functionalities based on user roles, interacting with various modules to verify credentials.

The email notification system relies on external email services and interacts with the placement officer functions to send necessary notifications.

These dependencies ensure that each part of the system operates correctly and interacts smoothly with others, maintaining overall system integrity and performance.

**6. Interface Description**

The interfaces in the Campus Placer project provide necessary details for interaction between modules.

Admin functions interface with the database to manage branches, HODs, placement officers, students, companies, and job categories, and with the authentication system to verify credentials.

HOD functions interface with the database to add, edit, view, and delete student records, and with the authentication system for user verification.

Placement officer functions interface with the database to manage job categories, companies, and student eligibility, with the email notification system to send updates, and with the authentication system for user verification.

Student functions interface with the database to manage resumes and job applications, and with the authentication system for user verification.

The database module interfaces with the underlying database system for data operations.

The authentication system interfaces with secure storage for user credentials and roles.

The email notification system interfaces with external email services to send notifications.

These interfaces ensure that each module can perform its functions and interact seamlessly with others, with detailed specifications for method names, parameters, data formats, and error codes.

**7. Detailed Design**

The detailed design section outlines the internal workings of each entity.

* Identification: Each entity, such as admin functions, HOD functions, placement officer functions, student functions, database module, authentication system, and email notification system, is uniquely identified.
* Processing: Algorithms and rules for managing data, performing operations, and interacting with other modules are specified for each entity.
* Data: Structures for storing and handling data, including attributes, formats, and acceptable values, are detailed for each entity.