

BANASTHALI VIDYAPITH

University for Women: University with a difference



VidyaPath:Placement Portal

[Group ID: CSD111]

Software Design Specifications

DEPARTMENT OF COMPUTER SCIENCE

GUIDED BY:

Dr. Vaibhav Vyas
Associate Professor

SUBMITTED BY:

Chhavi Sharma (2216257)
Pranjal Dubey (2216354)
Kesar (2216301)
Shreya Uttam (2216394)
Shruti Sharma (2216399)

B.Tech. CSE (V Semester)

Chapter 1 SDS: TABLE OF CONTENTS

1. Introduction

1.1 Purpose	3
1.2 Scope	3
1.3 Definitions, Acronyms, Abbreviations	4
1.4 References	5
1.5 Overview of Document	6

2. System Architecture Document

2.1 Overview of modules/components	7
2.2 Structures and Relationships	8
2.3 User Interface Issues	15

3. Detailed Description of Components

3.1 Component Template Description	16
3.2 X Component	17
3.3 Y Component	18
3.4 Z Component	20

4. Reuse and Relationships to other Products

5. Design decision and trade-offs

6. Pseudocode for components

1. INTRODUCTION

The “VidyaPath “placement portal is a comprehensive placement platform designed to streamline the recruitment process for students, and the college's placement cell. The platform is a centralized hub where students can create profiles, upload resumes, and apply for internship opportunities posted by the placement cell. The placement cell can efficiently manage the placement process, track recruitment trends, and maintain company records. VidyaPath bridges the gap, ensuring a smooth and organized placement experience for all students. It also provides students with data of companies that visited the campus earlier and the number of students placed in it.

1.1 Purpose

- Simplify the recruitment process for students, companies, and the placement cell through a centralized platform.
- Allow students to create profiles, upload resumes, and apply for internships.
- Provide students with data on companies that previously visited the campus and the number of students placed.

1.2 Scope

- Software Product to be produced:

A centralized web-based placement portal, VidyaPath, for students, and the college’s placement cell.

- What the Software Product Will Do:

Allow students to create profiles, upload resumes, and apply for opportunities. Enable the placement cell to post internships on the notice section and send the SMS or mail for the same as notification, track applications, and manage company data. Provide data on past recruitment trends and company visits and ensure that important updates reach students in a timely manner. It facilitates timely communication between the placement cell and students regarding campus recruitment activities.

- Application and Relevant Benefits:

The key benefits, objectives, and goals include:

- Collaboration and Transparency
- Seamless Profile Creation and Application
- Centralized Data Access
- Enhanced Placement Cell Efficiency
- Improved Communication and Transparency

1.3 Definitions, Acronyms and Abbreviations

- **ADMIN**

The administrator is responsible for controlling and managing the entire VidyaPath system, overseeing all user activities, and maintaining system records.

- **REGISTERED USER**

A user who has created an account logs into the system, and can access various features such as applying for the internship posted by placement cell and viewing the previously visited company's data, uploading resumes.

- **IEEE**

The Institute of Electrical and Electronics Engineers, a global organization promoting technical advancements, including standards for computer systems and communications.

- **SERVER**

The main computer on the network that stores data shared by multiple computers, including the VidyaPath platform.

- **BROWSER**

A software application used to locate and display web pages.

- **SDS**

Software Document Specification

- **HTML**

Hypertext Markup Language is a Markup language used to design static web pages of the VidyaPath portal.

- **TCP/IP**

Transmission Control Protocol/Internet Protocol, the suite of communication protocols used to connect hosts on the Internet. TCP/IP uses several protocols, the two main ones being TCP and IP.

- **IE**

Internet Explorer is a deprecated (or discontinued for most modern Windows editions) series of graphical web browsers developed by Microsoft that were used in the Windows line of operating systems.

- **RAM**

Random-access memory is a form of electronic computer memory that can be read and changed in any order, typically used to store working data and machine code.

- **IIS**

Internet Information Services is an extensible web server created by Microsoft for use with the Windows NT family.

- **DBA**

Database administrators use specialized software to store and organize data. The role may include capacity planning, installation, configuration, database design, migration, performance monitoring, security, troubleshooting, as well as backup and data recovery.

- **HDD**

A hard disk drive, hard disk, hard drive, or fixed disk, is an electro-mechanical data storage device that stores and retrieves digital data using magnetic storage with one or more rigid rapidly rotating platters coated with magnetic material.

1.4 References

- 0132350882, Ian Sommerville, Software Engineering, 9th edition, Pearson Education, April 2010.
- 0321213351, Martin Fowler, Patterns of Enterprise Application Architecture, 1st edition, Addison-Wesley, November 2002.
- 0134685997, Robert C. Martin, Clean Architecture: A Craftsman's Guide to Software Structure and Design, Pearson, September 2017.
- 0132350884, Michael L. Scott, Programming Language Pragmatics, 4th edition, Morgan Kaufmann, February 2015.
- 0201633612, Erich Gamma, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, October 1994.

1.5 Overview of Document

This SDS will be composed of several major sections to clearly and comprehensively understand the system design. Each section is dedicated to a specific area of the system, so all pertinent information related to the design can be clearly understood and traced back:

Introduction: This section sets out the goals, scope, and purpose of the SDS. It will offer an essential context for establishing what the system intends to achieve and what its rationale might be in terms of design.

System Architecture: In this section, we explain the high-level structure of the system. It represents components, modules, and relationships. It provides for a modular, maintainable, and scalable design.

Detailed Component Description: Every component of the system will be explained in detail, including its purpose, functionality, dependencies, interfaces, and resources. This part will guarantee that detailed details are realized to understand each part in contributing towards the overall system.

User Interface Considerations: A concentrated presentation of design principles of the user interface based on the user roles defined above. This encompasses descriptions of how users will interact with the system.

Data Design: A presentation of structures and database design, aimed at ensuring that all system data are handled efficiently and safely.

Constraints and Dependencies All constraints and dependencies, such as hardware and software as well as regulatory requirements together with inter-system components, are defined in this section.

2. System Architecture Description

2.1 Overview of modules / components

2.1.1 User Authentication Module

- Purpose: Handles the login, registration, and user verification processes for students and the placement cell.
- Components:
 - Login: Accepts username and password for authentication.
 - Registration: Captures student details such as name, college ID, personal email, branch, year, course, phone number, smartcard ID.
 - Password Recovery: Manages resetting forgotten passwords.

2.1.2 Resume Management Module

- Purpose: Allows students to upload and manage resumes.
- Components:
 - Resume Upload: Facilitates the uploading of new resumes.
 - Previous Resumes Storage: Stores all previous versions of uploaded resumes for future reference.
 - Resume Templates: Provides the templates for students to use when creating their resumes.

2.1.3 CGPA & Achievements Verification Module

- Purpose: Ensures the validity of CGPA and achievements listed in resumes through document uploads.
- Components:
 - Mark sheets Upload: Allows students to upload mark sheets for CGPA verification.
 - Certificates Upload: Allows students to upload certifications for achievements listed in the resume.

2.1.4 Placement Notices Module

- Purpose: Provides updates and notifications about upcoming companies and placement opportunities.
- Components:
 - Notice Creation: Placement cell can post notices about upcoming companies.
 - Student Notifications: Sends notifications to students regarding placement opportunities.
 - Notice Management: Allows the placement cell to edit or deactivate notices after deadlines have passed.

2.1.5 Student Profile Management Module

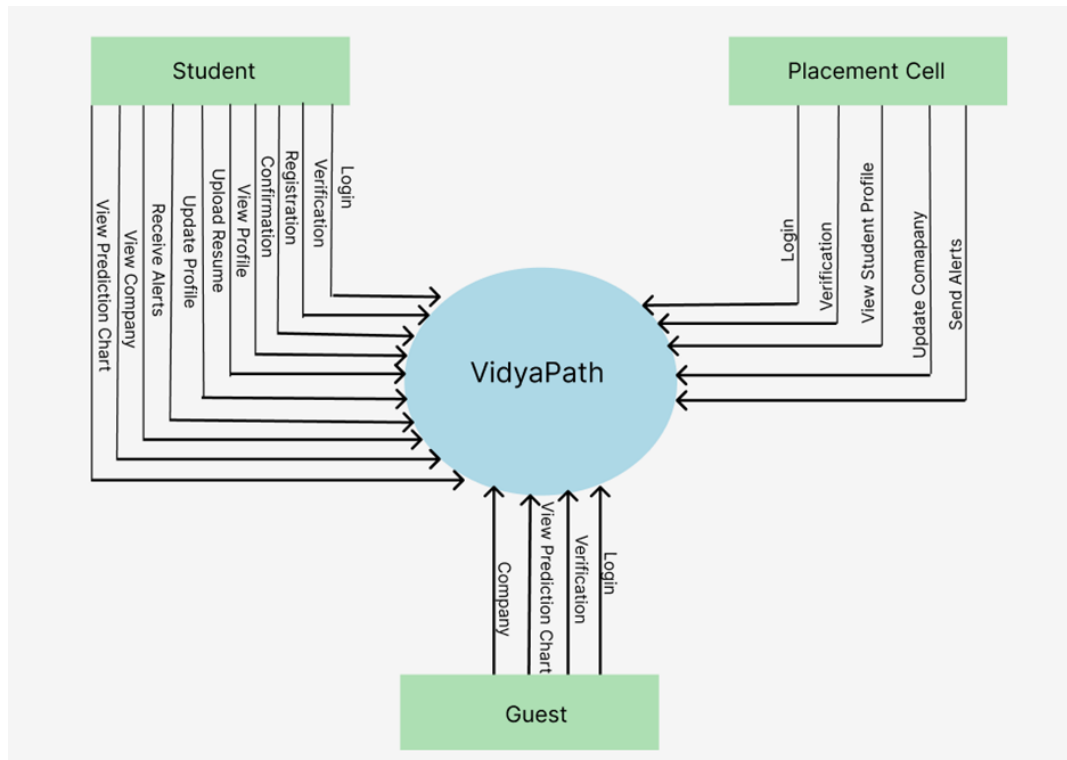
- Purpose: Manages student profiles, including personal details, resumes, and uploaded documents.
- Components:
 - Profile Editing: Students can update personal information and upload new documents.
 - View History: Students can view their previously uploaded resumes.

2.1.6 Admin and Placement Cell Dashboard Module

- Purpose: Provides an interface for the placement cell to manage the placement process.
- Components:
 - Student List: View and manage the list of students, including their profiles and documents.
 - Notice Management: Post, edit, and delete notices regarding placements.
 - Verification Management: Review uploaded documents (e.g., mark sheets, certificates) for verification.

2.2 Structure and Relationships

The system consists of multiple modules that interact with each other to provide a seamless experience for both students and the placement cell. Below is a high-level overview of the structure and relationships between the modules.



2.2.1 User Authentication Module

- Relationships:
 - Interacts with the Student Profile Management Module to authenticate users based on their credentials.
 - Communicates with the Resume Management Module to grant access to upload and view resumes after successful login.

2.2.2 Student Profile Management Module

- Structure:
 - Manages student details such as name, college ID, branch, and year.
 - Stores links to uploaded mark sheets, certificates, and resumes.
- Relationships:
 - Works with the Resume Management Module to store and display resumes.
 - Connects with the CGPA & Achievements Verification Module for document validation.

2.2.3 Resume Management Module

- Structure:
 - Contains fields for storing multiple resumes per student.
 - Stores and retrieves resumes based on the user's profile.
- Relationships:
 - Links with the Student Profile Management Module to access student details.
 - Interacts with the CGPA & Achievements Verification Module to ensure resumes accurately reflect the verified data.

2.2.4 CGPA & Achievements Verification Module

- Structure:
 - Stores mark sheets and certificates uploaded by students.
 - Flags unverified data for review by the placement cell.
- Relationships:
 - Connects to the Resume Management Module to verify that the CGPA and achievements listed in resumes are valid.
 - Collaborates with the Admin and Placement Cell Dashboard Module for document review and approval.

2.2.5 Placement Notices Module

- **Structure:**
 - Stores information on upcoming placement events and sends notifications to students.
- **Relationships:**
 - Links to the Student Profile Management Module to notify registered students about upcoming placements.
 - Controlled by the Admin and Placement Cell Dashboard Module to create and update notices.

2.2.6 Admin and Placement Cell Dashboard Module

- **Structure:**
 - Provides a centralized interface for managing the placement process.
 - Includes functionalities for managing student lists, verifying documents, and creating notices.
- **Relationships:**
 - Works closely with all other modules:
 - Manages notices via the **Placement Notices Module**.
 - Reviews and verifies student documents through the **CGPA & Achievements Verification Module**. Tracks student data in the **Student Profile Management Module**.

User Accessibility:

The website should be intuitive, allowing students to register, log in, upload resumes, and view placement notices. The placement cell (admin) will manage student profiles, resumes, and post notifications about company visits.

Responsiveness:

The website must be fully responsive, ensuring smooth access on both desktop and mobile devices for all users, including students and admin.

Role-Based Access:

Separate interfaces for students and the placement cell (admin). Admin can post company details and manage student data, while students can upload resumes, track their progress, and view placement notifications.

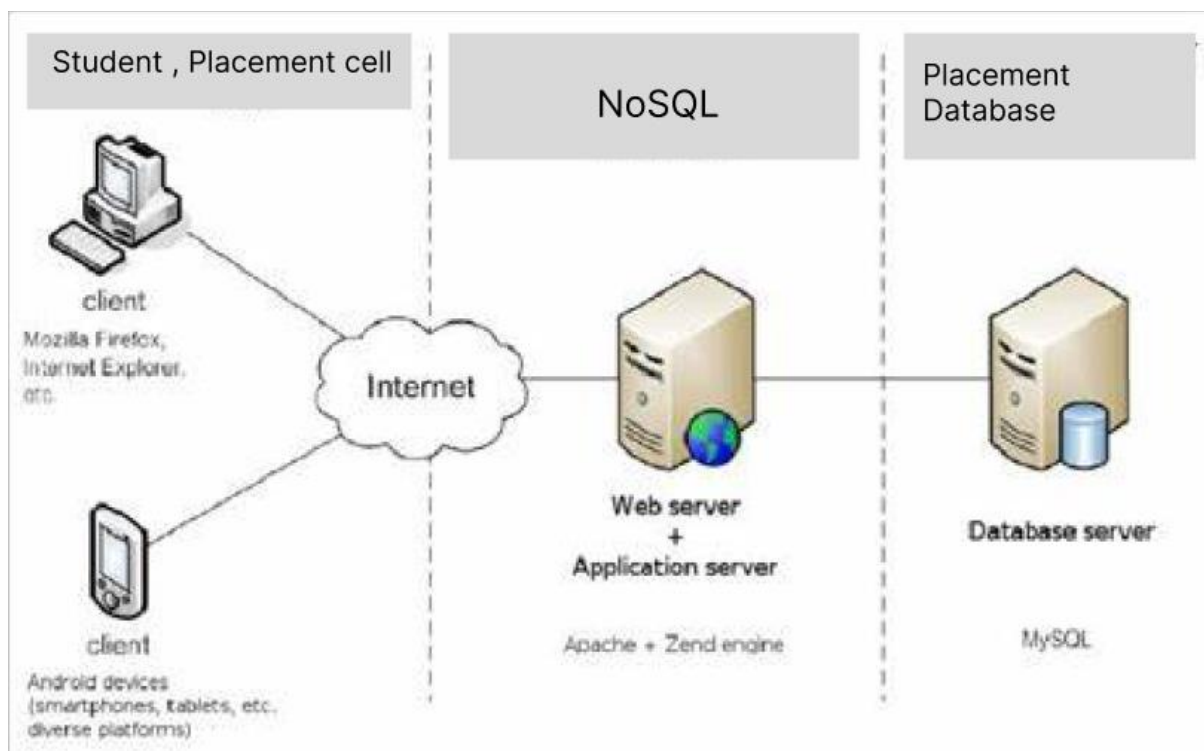
Notifications:

Detailed notifications will be sent via email and SMS to students who log in with their college ID. Only authenticated students will receive these alerts through personal channels.

Profile and Resume Management:

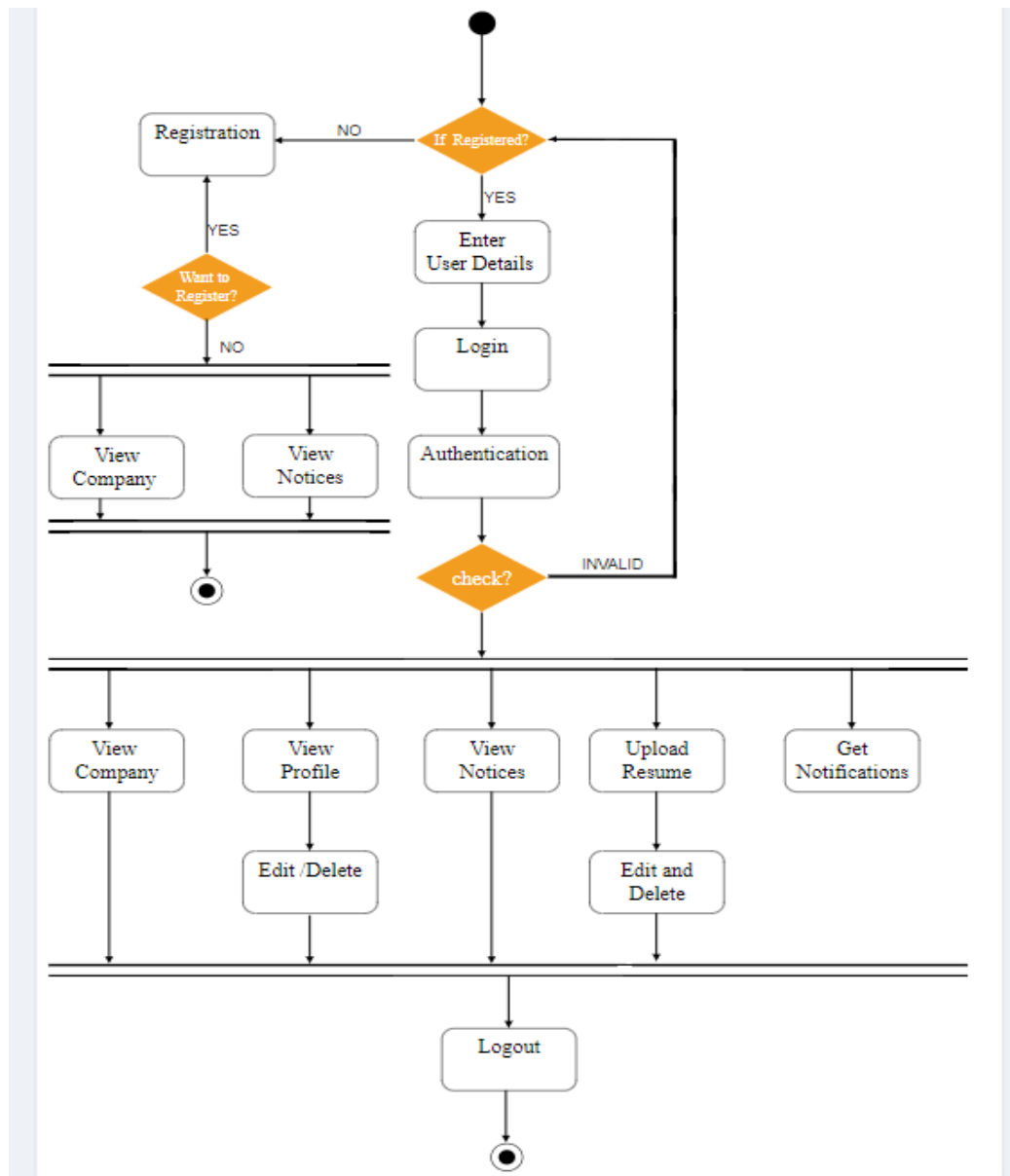
Students should be able to upload and update their resumes easily, with a history of previous versions available to track their growth.

HIGH- LEVEL DESIGN OVERVIEW

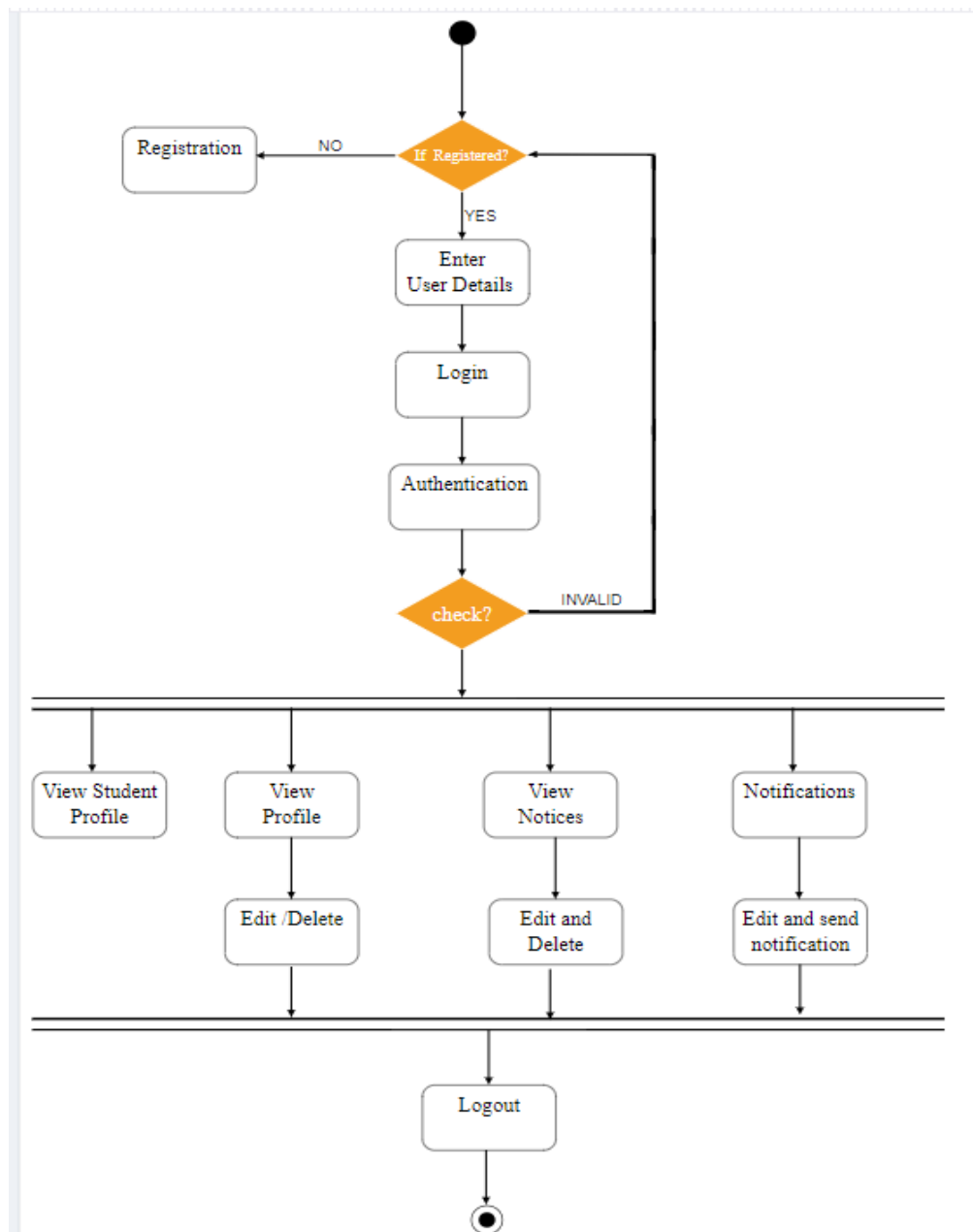


ACTIVITY DIAGRAMS

1. Student

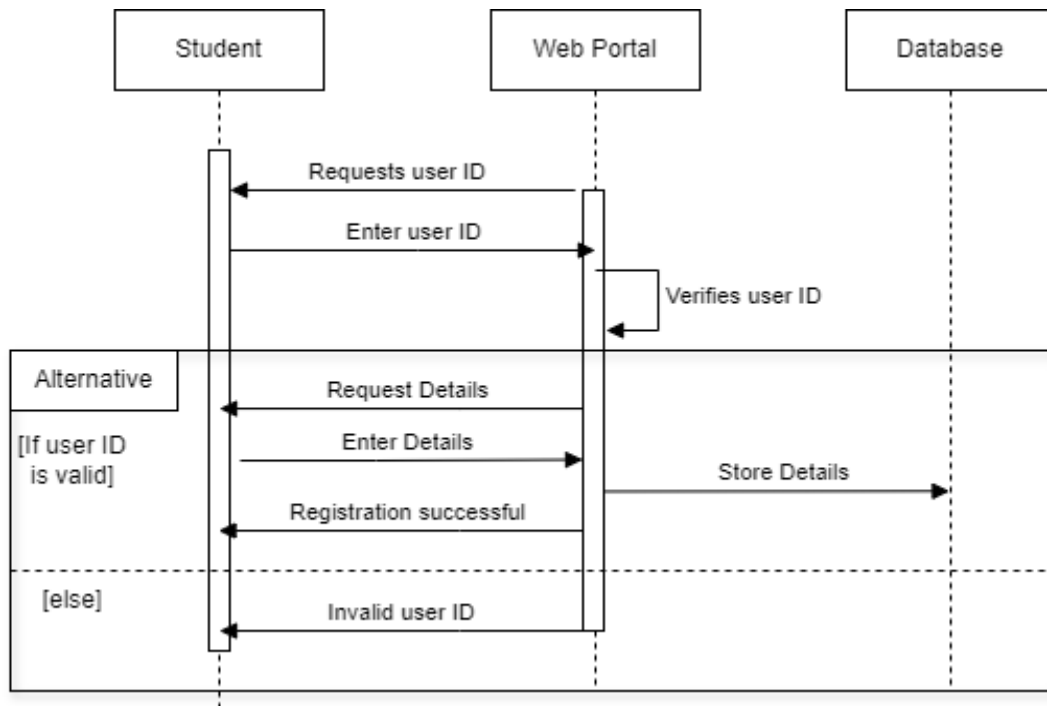


2. Admin

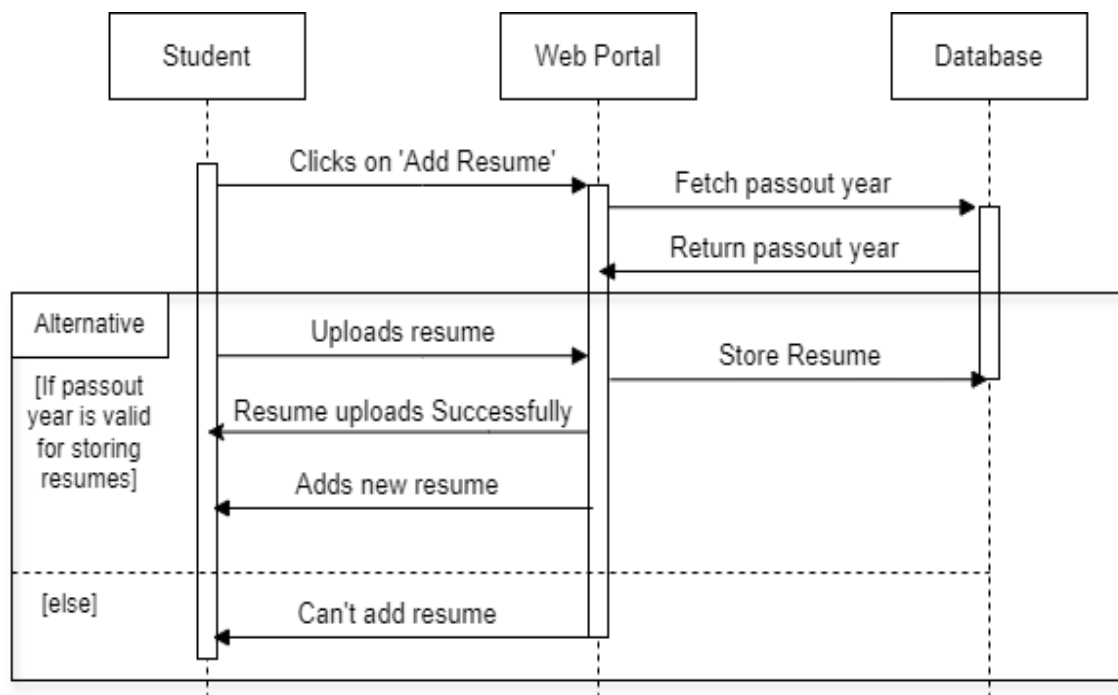


SEQUENCE DIAGRAMS

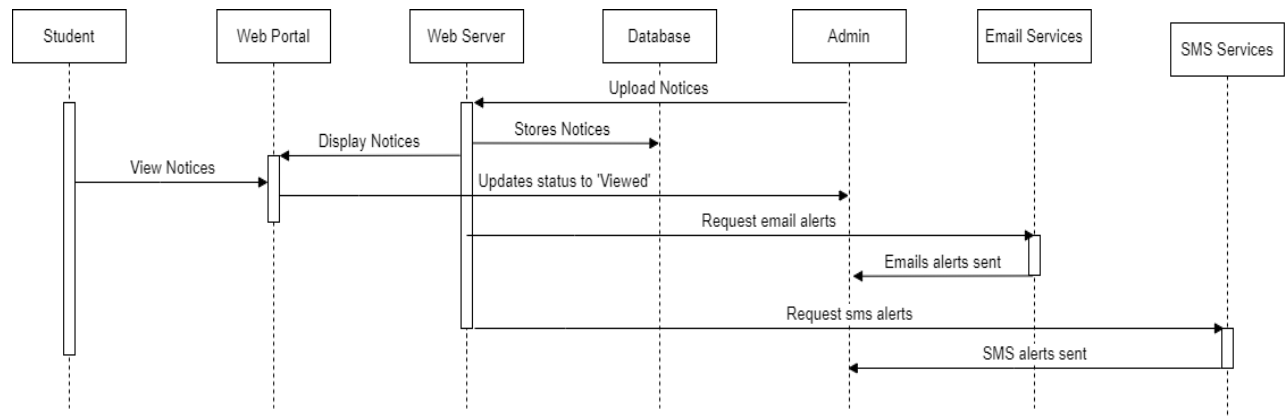
1. Registration



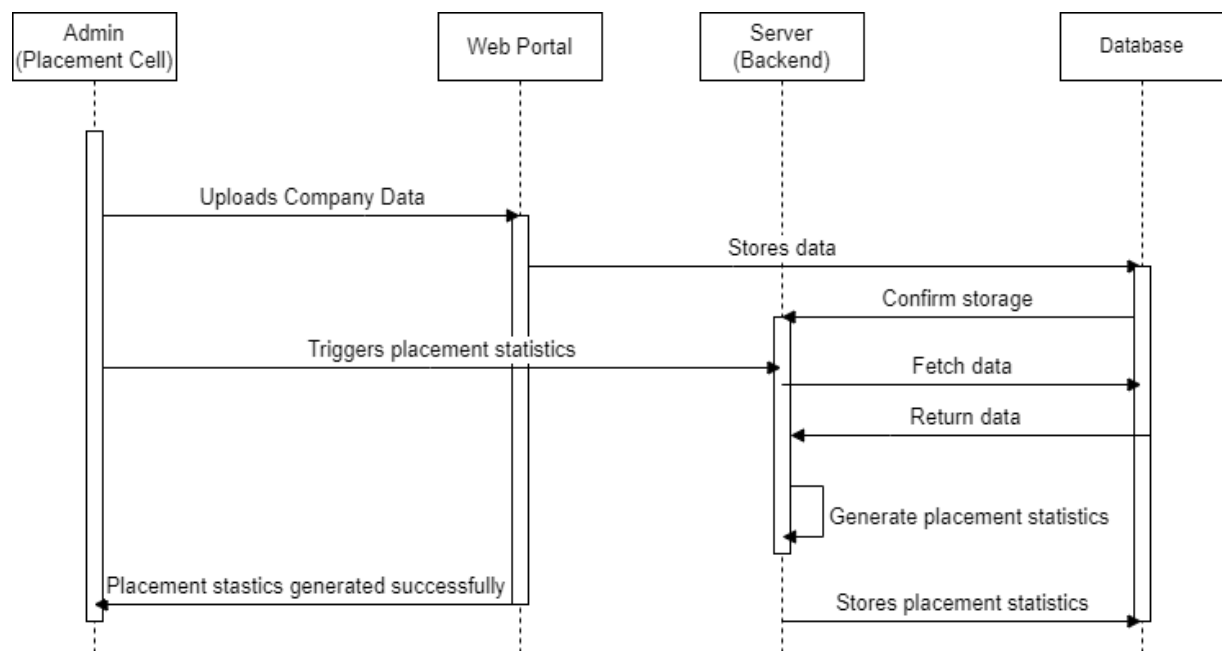
2. Resume Management



3. Notice Management



4. Placement Statistics



2.3 User interface issues

The user interface for the Placement Portal is designed with simplicity and ease of use in mind, catering to both **students** and **placement cell (admin)** users.

- **Clarity and Simplicity:**
 - The interface is clean and simple, ensuring users can easily navigate the system. The **students** should see key functions like "Upload Resume," "View Previous Resumes," and "Upcoming Companies" prominently on their dashboard.

- **Admins** (Placement Cell) should have clear access to functions like "Post Notification," "Verify Student Documents," and "Manage Company's information."
- **Personalization:**
 - When a **student** logs in, they are greeted with their name and a progress bar showing the number of resumes uploaded and verified.
 - **Placement Cell (Admin)** views will show an overview of students who have submitted resumes and need verification.
- **Responsive Design:**
 - The UI is mobile-friendly, ensuring students and admins can access the portal from different devices without compromising functionality or readability.
- **Notifications and Alerts:**
 - Important notifications, such as **new company placements**, will appear as pop-ups or banners at the top of the screen for **students**, ensuring they are immediately aware of upcoming opportunities.
- **Consistency:**
 - The design, colours, and fonts will remain consistent across the portal to reduce confusion.

3. Detailed description of components

3.1 Component template description

Component Attribute	Description
Identification	The unique name of the component and its location within the system. This allows it to be distinctly recognized and referenced.
Type	Specifies whether the component is a module, a subprogram, a data file, a control procedure, a class, etc.
Purpose	Describes the function and performance requirements the component must fulfil. This includes derived requirements, which are not explicitly stated in the SRS but implied by the design.

Function	Outlines what the component does, including the input it processes, algorithms used, outputs it generates, and where the data is stored or modified.
Subordinates	Explains the internal structure of the component, listing any subcomponents and their respective functional roles.
Dependencies	Describes how the component interacts with other components, its relationship to other modules, and any timing or data-sharing requirements.
Interfaces	Provides details on both external and internal interfaces. This includes communication mechanisms, error messages, error codes, screen formats, and user interface elements related to the component.
Resources	Lists the resources needed by the component, such as hardware, memory, I/O channels, or external libraries, to function properly.
Processing	Elaborates on the function description by giving a step-by-step account of the logic or algorithms used. Pseudocode or equations can be included for clarification.
Data	Describes the internal data used by the component, including its format, initial values, and usage. This information is typically part of the data dictionary.

3.2 X Component

Component Attribute	Description
Identification	Student Profile Management Module, located in the central data layer of the placement system.
Type	Module
Purpose	Manages all student-related data, including personal information, resumes, academic details, and profile updates. It ensures the integrity and security of student profiles, handling unlimited previous resumes.

Function	Handles the creation, modification, and retrieval of student profiles. Stores multiple versions of resumes, ensuring older resumes are kept for future comparison. It accepts user inputs such as name, email, course, and resume uploads, processes and stores them in the database.
Subordinates	N/A
Dependencies	Depends on the Resume Management Component for storing and retrieving resume files. Also interacts with the Placement Cell Dashboard for notifications.
Interfaces	External Interface: Student uploads resume via the web interface. Placement Cell accesses student profiles for evaluation.
Resources	Requires database storage for resumes and profile data. Uses server memory for storing previous resume versions.
Processing	When a student uploads a new resume, the old version is archived, and the new version is marked as active. The module fetches the most recent resume on-demand and allows access to previous ones.
Data	Stores student details such as name, email, course, college and resume history. Resume data is stored with metadata indicating the upload date and time.

3.3 Y Component

Component Attribute	Description
Identification	The Y Component is the "Resume Management" module located within the Student Profile Management System of the placement portal.
Type	Module
Purpose	The purpose of the Resume Management module is to allow students to upload new resumes, while the system stores all previous resumes in a secure manner. It enables students to

	track their resume progression over time and provides storage for an unlimited number of resumes.
Function	This component handles the following functions: <ul style="list-style-type: none"> Accepts new resume uploads from students. Stores previous resumes for each student without overwriting the new one. Enables students and placement cell administrators to view and download both current and historical resumes.
Subordinates	The Resume Upload Subsystem, Previous Resume Archive, Resume Retrieval System.
Dependencies	This component relies on the Student Profile Management System for student authentication and profile retrieval. It interacts with the File Storage System to store and manage the uploaded resume files. Additionally, the Placement Cell Dashboard uses this component to access student resumes for company placement purposes.
Interfaces	<ul style="list-style-type: none"> External Interfaces: File Upload Interface for students to upload resumes. Internal Interfaces: Communicates with the student profile system to retrieve student IDs and ensure proper resume storage. "Resume Archive" list for students to view their historical resumes.
Resources	This component requires file storage (HDD/SSD space), database access for storing metadata about each resume (e.g., upload date, user id), and processing resources for file management and retrieval functions.
Processing	N/A
Data	The stored data includes user ID, file paths for each uploaded resume, upload date, and the version number of each resume. This is stored in the Resume Management Database and updated with each new upload. The resumes themselves are stored as PDF files in a separate file storage system.

3.4 Z Component

Component Attribute	Description
Identification	Z Component – A unique identifier that places this component within the Placement Management System. It is part of the Resume Management Module.
Type	Class – Represents a Python class that manages student resumes, responsible for storing, retrieving, and displaying past and current resumes.
Purpose	The component allows students to upload new resumes while storing previous versions for future reference. It provides version control for resumes.
Function	The component accepts new resumes and checks if any previous resumes exist for the student. If yes, it moves the old resume to an archive table. It also allows users (students and placement cell) to view both current and archived resumes. The system modifies the resume data when new resumes are uploaded and stores them appropriately.
Subordinates	<ul style="list-style-type: none"> - Upload Class: Handles the actual uploading of resumes. - Archive Class: Moves old resumes to an archive. - View Class: Displays current and previous resumes for both students and placement cell.
Dependencies	Depends on the Student Profile Management Module for student data. It shares data with the Placement cell Module for reviewing student resumes. Timing dependencies exist with the Resume Upload Process.
Interfaces	<ul style="list-style-type: none"> - User Interface: The "Resume Upload" button and archive view are part of the user interface. - API Interface: Interacts with the backend through REST APIs for fetching and storing resume data. - Error Handling: Provides specific error codes like "File format not supported" and "Upload failed".
Resources	- CPU Execution Time: Moderate, due to file handling and potential archival processes.

	<ul style="list-style-type: none"> - Memory: Requires sufficient memory for storing multiple versions of resumes. - Storage: Needs adequate space for storing both current and archived resumes, potentially across different academic years.
Processing	N/A
Data	<ul style="list-style-type: none"> - resume_id: Unique identifier for each resume. - user_id: Foreign key linking to the Student table. - resume: Stores the actual resume file. - upload_date: Timestamp for when the resume was uploaded. - archive_status: Indicates whether the resume is archived.

DATA DESIGN

DATABASE DESCRIPTION

USER Table (General user, Student, Placement Cell)

Attribute	Type	Constraint
user_id	Varchar(20)	Primary Key
username	Varchar(50)	Unique, Not Null
password	Varchar(100)	Not Null
role	Varchar(10)	Not Null
email	Varchar(50)	Unique, Not Null
phone_number	Varchar(15)	Unique, Not Null

STUDENT Table

Attribute	Type	Constraints
smartcard_id	Varchar(20)	Primary Key
name	Varchar(50)	Not Null
user_id	Varchar(20)	Foreign Key(references USER.user_id)
personal_id	Varchar(20)	Unique, Not Null
passout_year	Varchar(4)	Not Null
branch	Varchar(40)	Not Null
phone_number	Varchar(15)	Not Null

RESUME Table

Attribute	Type	Constraints
resume_id	Varchar(20)	Primary Key
smartcard_id	Varchar(20)	Foreign Key(references STUDENT.smartcard_id)
current_resume	Varchar(20)	Not Null
upload_date	Date	Not Null

RESUME_HISTORY Table (for storing all past resumes)

Attribute	Type	Constraints
resume_id	Varchar(20)	Foreign Key(references RESUME.resume_id)

smartcard_id	Varchar(20)	Foreign Key(references STUDENT.smartcard_id)
old_resume	Varchar(40)	Not Null
upload_date	Date	Not Null

ATTACHMENT Table

Attribute	Type	Constraints
attachment_id	Varchar(20)	Primary Key
type	Varchar(20)	Not Null
attachment	Varchar(40)	Not Null
upload_date	Date	Not Null

PLACEMENT_CELL Table

Attribute	Type	Constraints
pid	Varchar(20)	Primary Key
email	Varchar(50)	Unique, Not Null
name	Varchar(40)	Not Null
phone_number	Varchar(15)	Unique, Not Null

COMPANY Table

Attribute	Type	Constraints
cid	Varchar(20)	Primary Key

company_name	Varchar(50)	Unique, Not Null
company_detail	Varchar(200)	Not Null

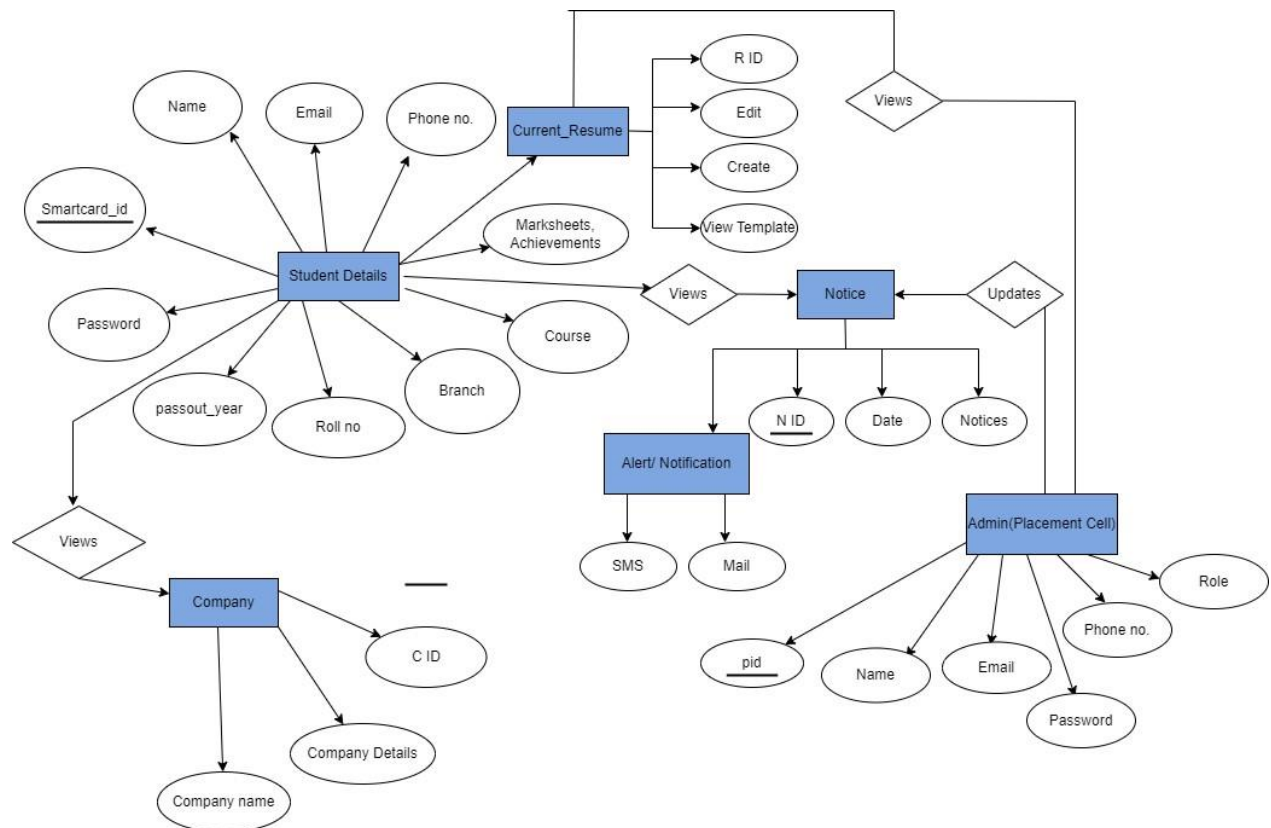
NOTICES Table

Attribute	Type	Constraints
nid	Varchar(20)	Primary Key
upload_date	Date	Not Null
notices	Varchar(50)	Not Null
time	Time	Not Null

NOTIFICATION Table

Attribute	Type	Constraints
notification_id	Varchar(20)	Primary Key
nid	Varchar(20)	Foreign Key(references NOTICES.nid)
notification	Varchar(100)	Not Null
upload_date	Date	Not Null
time	Time	Not Null

ER DIAGRAM



4. Reuse and Relationships to Other Products

This section outlines the potential reuse that will be considered during the design and implementation phases of the placement portal project.

4.1 Reuse in Product Design

The project considers reusing several existing tools and frameworks to ensure efficient product design:

- **Visualization Tools:** Existing solutions for visualizing placement statistics will be used to display placement statistics. These tools will save time and effort compared to building custom visualizations from scratch.
- **Authentication Modules:** Pre-built modules for handling secure login and user management will be reused to ensure robust access control for students and administrators.

- **Responsive Design Frameworks:** A front-end framework such as Bootstrap will be reused to create visually appealing websites more quickly and efficiently.

4.2 Reuse in Product Implementation

During the implementation stage, the following reuse strategies could be anticipated:

- **Resume Upload Handling:** Libraries or frameworks for secure file uploads will be reused to manage students' resume uploads, ensuring that resumes uploaded are efficiently and stored securely.
- **Database Management:** Standard database schemas and management solutions will be reused and adapted for storing student data, resumes, and placement records. This will help ensure scalability and reliability.
- **Notification Tools:** Open-source or third-party tools for sending notifications (email and SMS alerts) will be reused to automate informing students about placement opportunities and events.

4.3 Motivation for Not Reusing Certain Materials

For aspects of the project where reuse is not viable, the following reasons apply:

- **Security Concerns:** Certain third-party libraries or tools may not meet the security standards needed to handle sensitive data like student resumes, and thus will be avoided.
- **Customization Requirements:** Custom workflows, such as the eligibility check for resume uploads and placement predictions, may not be addressed by existing solutions, necessitating custom development.

5. Design Decisions and Tradeoffs for the "Banasthali Placement Portal"

The "Banasthali Placement Portal" is designed to efficiently manage the placement process for students while providing valuable tools for both the placement cell and students. The decisions made during the design process were influenced by the need for usability, scalability, and integration with existing

academic workflows. Below is an explanation of the key design decisions and the trade-offs associated with each function:

1. Student Registration & Profile Management

- **Design Decision:** The portal allows students to register using their unique college ID, ensuring that only authorized users can access the system. Students can create a profile with their academic information (year, branch) and personal details.
- **Trade-off:**
 - **User Experience vs. Security:** While integrating the college ID ensures security, it may slightly complicate the registration process if students face issues retrieving their ID or if the ID validation fails. However, this was prioritized over a simpler email-based registration to prevent unauthorized access.
 - **Scalability:** The registration process can be easily scaled as more students join each year, and profiles are stored and managed in a structured manner.

2. Resume Management

- **Design Decision:** The portal provides customizable resume templates, allowing students to create, upload, and edit resumes. It also archives older versions of resumes to help track growth and provide a historical record.
- **Trade-off:**
 - **Customization vs. Complexity:** Offering customizable resume templates gives students flexibility, but it adds complexity to the system, requiring the maintenance of multiple formats. Simpler designs might have been easier to implement but would limit students' ability to personalize their resumes for specific roles or companies.
 - **Storage Concerns:** Archiving old resumes means increased data storage. The trade-off here is between storage costs and the benefit of allowing students to track changes in their resumes over time. We decided to prioritize tracking student progress.

3. Document Verification

- **Design Decision:** The portal allows students to upload academic and extra-curricular attachments. These documents are accessed by the placement cell.
- **Trade-off:**
 - **Manual Verification vs. Automation:** Verifying documents manually by the placement cell ensures accuracy, but it requires more effort. Automation (e.g., using OCR to extract data from documents) could reduce the workload but might introduce errors. Although to reduce time automation was chosen.
 - **Document Formats:** Limiting document uploads to specific formats (PDF, JPG) simplifies system processing but may inconvenience students who have documents in other formats. This was deemed necessary to ensure smooth handling and secure storage of documents.

4. Past Placement Trends

- **Design Decision:** The system provides visualizations of past placements and internships to help students analyze company trends and make informed decisions.
- **Trade-off:**
 - **Data Visualization vs. Data Storage:** While providing rich visual insights into past placement trends is helpful, maintaining detailed records of past placements requires additional data storage and computational resources. The benefit of guiding students in career choices outweighed the additional complexity and resource requirements
 - **Data Privacy:** There's a balance between displaying enough information to be useful and protecting the privacy of past students. The design ensures aggregated data is displayed without compromising individual privacy.

5. Company Information

- **Design Decision:** Detailed information on companies, job roles, eligibility, and recruitment schedules is provided to students.
- **Trade-off:**
 - **Dynamic Updates vs. Static Data:** Dynamic, real-time updates of company information would provide students with the most current

data, but it could introduce complexity in syncing data with external sources. The trade-off here was to prioritize consistency over real-time updates, as companies often provide schedules well in advance.

- **Data Integration:** Integration with company databases or APIs could automate some of the processes but adds the complexity of dealing with multiple external systems. Manual updates are easier to control and implement in the short term but require more effort from the placement cell.

6. Notifications & Alerts

- **Design Decision:** Notifications are sent to students about upcoming placement drives, deadlines, and visits. The placement cell can manage these alerts.
- **Trade-off:**
 - **Real-Time Notifications vs. Batch Processing:** While real-time notifications would ensure immediate updates, they are more complex and costly to implement at scale. A scheduled, batch-based notification system is simpler and reliable for most use cases. Immediate notifications are reserved for critical updates.
 - **Platform Compatibility:** Notifications are sent via email and SMS. Extending this to mobile push notifications or SMS would enhance accessibility but introduces higher maintenance and costs. We prioritize both SMS and email alerts due to wider accessibility.

7. Placement Cell Management

- **Design Decision:** The placement cell has full control over managing companies, placement drives, and document verification. It can track students' progress and handle all backend operations.
- **Trade-off:**
 - **Centralized Control vs. Decentralization:** A centralized control system for the placement cell ensures efficiency and consistency, but it may limit the ability of individual departments or student representatives to manage their own processes. We chose centralization to maintain uniformity and reduce the risk of data inconsistencies.
 - **Automation vs. Manual Management:** Automating certain tasks (e.g., document verification, company outreach) would reduce the placement cell's workload but could introduce errors. Manual

oversight ensures accuracy, and the trade-off was made in favour of precision over convenience.

Abandoned Ideas:

1. Manual Verification of Document Verification:

- **Reason for Abandonment:** While automation could save time, but manual will increase the manpower and time complexity.

2. Real-Time Data Sync with Company Databases:

- **Reason for Abandonment:** Although real-time updates from companies would be ideal, the complexity of integrating with multiple external systems and the variability in company processes made this option impractical. Static updates by the placement cell provide sufficient information without added complexity.

5. Pseudocode for components

N/A