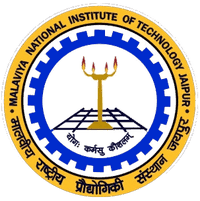
MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY, JAIPUR

****

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

WEB APPLICATION ON ONLINE TRANING & PLACEMENT MANAGEMENT SYSTEM

Submitted by-

Ashish Goyal

(2016ucp1100)

**TABLE OF CONTENTS:**

1. **Introduction**
   1. Purpose
   2. Scope
   3. Definitions, Acronyms and Abbreviations
   4. References
   5. Overview
2. **General description**
   1. Product perspective
   2. Product function summary
   3. User characteristics
   4. General constraints
   5. Any Assumptions and dependencies
3. **Specific Requirements** 
   1. Functional requirements
   2. External interface requirements
   3. Performance requirements
   4. Design constraints
   5. Attributes
4. **Appendices**
   1. Data Dictionary
   2. Data Flow Diagrams (DFD)
   3. Entity Relationship Diagram
   4. Use Case Diagram
   5. Class Diagram
   6. Sequence Diagram
   7. Algorithms
   8. Testing
5. **Introduction:**

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose of this document is described and a list of abbreviations and definitions is provided.

* 1. **Purpose:**

The purpose of this document is to give a detailed description of the requirements for the “Web Application for Online Training & Placement Management System”. It will illustrate the purpose and complete declaration for the development of system.

The intended audience of this document includes head of T.P.O (admin), the recruiters and the students looking for On-Campus Training and Placement

* 1. **Scope:**

This web application can be used for smooth conduction of training and placement process in the college. It helps the TPO of the college to manage the whole intern/placement data of student and company in an effective online manner.

It has been developed to override the problems prevailing in the current manual system.

It provides many flexible and convenient features that allow admin, students and company to minimize the time and maximize the efficiency. This system allows students to update their information. Also the admin manages both students and companies in an easy and effective way and the complete process is managed on the site online without accessing and changing database manually. This online system can be accessed from anywhere on PC (having internet connection).

The web application is optimized to avoid errors while entering data by checking various constraints. It also display error messages if entered data is incorrect. No formal knowledge is needed for the user to use this system. Hence, it is user-friendly.

* 1. **Definitions, Acronyms and Abbreviations:**

|  |  |
| --- | --- |
| **Term** | **Definition/Acronyms/Abbreviations** |
| SRS | Software Requirements Specification |
| Admin | System administrator of college who has permission for managing and controlling the system |
| T & P Cell | Training & Placement Cell |
| TPO | Training & Placement officer |
| RAM | Random Access Memory |
| DFD | Data Flow Diagram |
| ERD | Entity Relationship Diagram |
| PC | Personal Computer |
| SQL | Structured Query Language. |
| PHP | Hypertext Preprocessor |
| HTML | Hypertext Markup Language |
| CSS | Cascading Style Sheets |
| CGPA | Cumulative Grade Points Average |

* 1. **References:**
* Pressman, R. S.: Software Engineering, A Practitioner's Approach (European adaptation, fifth edition). McGraw Hill, 2000.
* .<http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs_example_2010_group2.pdf>
* IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
* <https://arxiv.org/ftp/arxiv/papers/1005/1005.0169.pdf>
  1. **Overview:**

The remainder of this document includes two sections and appendices. The second section provides an overview of the system functionality and system interaction with other systems. This section also introduces different types of stakeholders and their interaction with the system. Further, the section also mentions the system constraints and assumptions about the product.

The third section provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

The Appendices in the end of the document include some UML diagrams, DFDs, ERD, Algorithms, and Testing.

1. **General description:**

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what are their functionalities. At last, the constraints and assumptions for the system will be presented.

* 1. **Product perspective:**

In the current existing system, TPO (admin) have to manage all the data of students and companies manually in an inefficient manner. Also the TPO have to explicitly provide information to the students when a company is visiting for training and placement using email (this work is too time consuming). Also once the student got selected for any company no confirmation mail is sent to the student (this is not efficient and authentic). If the student wants to update any of his details that he fills incorrectly for any company then the student has to go to T & P office and the updation process is manual.

The Online Training & Placement Management System will provide many advantages to resolve the existing problem. It will maintain a systematic database to store details of the students and companies. The complete system will become automatic and manual work will be reduced significantly. The authenticity and security will also enhanced using the mail feature in the system. It will save both time and effort in the complete training and placement process.

* 1. **Product function summary:**

The Online Training & Placement Management System would provide the facility of viewing both the academic and personal information of the student and the company.

The System will check whether student (register for any company) is fulfilling all the criteria of the company, it will also provide the list of the students that registered successfully for any company.

For authenticity, a mail is sent to the selected student along with the company name in which he got selected. If student forgot password for login then a new password will be send to his mail id.

The web application will be secure i.e. If any user is logged in then no other user can log in at the same time on the same browser. No one can access the Web Pages (that opens after user login) directly by address bar of the browser.

* 1. **User characteristics:**

The major User in the System would be:

* **Student:** 
  + Student can register at training and placement web application.
  + Student can view his profile after login.
  + Student can update his information in case if he entered incorrect information. Student can also change his password in the update section.
  + Student can have new password in case if he forgot password by receiving mail with new password.
  + Student can also view in how many companies he had registered.
  + Student can apply for the visiting company for internship or placement if he satisfies the company criteria.
  + Student can view general notifications from T & P cell and can also see the details of companies that are visiting.
* **Company:** 
  + A company can register at training and placement cell.
  + Company can add the criteria for students, e.g. the branch for which the company is coming, minimum CGPA required, place of posting, stipend(in case of internship), visiting date etc.
* **Admin:** 
  + Only Admin can register new admin at training and placement cell.
  + Add general notifications for the students.
  + Have access to all the information of the students and companies.
  + Only the admin gives approval to the company after verifying that it is an authentic company or not.
  + Admin can also update the absents of students.
  + Admin can view the list of students that registered for any company.
  + Admin changes the status of the students from not selected to got intern/placement after receiving final result from the company and a verification mail is sent to the selected student.
  + Admin can view the list of students who got placement or internship along with the company name.
  1. **General constraints:**
* This web application can be deployed on windows machine with Apache Server and MySQL database.
* Minimum RAM 3GB.
* 30GB or more Storage Space.
* Internet Connectivity with Ports configured for Xampp.
* This application can be accessed by user on a windows PC having any web browser that support Html, Css, and Javascript.
  1. **Assumptions and dependencies:**
* It is assumed that the user should have some basic knowledge of computer.
* The admin should have provided the correct information and notification to the students.
* All the non updatable details must be filled correctly.

1. **Specific Requirements:**

This section contains all the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

* 1. **Functional requirements:**

**F1: Student Registration.**

**Input:** Personal and Academic details.

**Output:** If successfully registered, then entry of student details in the database.

**F2: Company Registration.**

**Input:** General details from company.

**Output:** If successfully registered, then entry of company details in the database.

**F3: Admin Registration.**

**Input:** Official information of the admin.

**Output:** If successfully registered, then entry of admin details in the database.

**F4: Admin Login.**

**Input:** name, id and password.

**Output:** admin’s index page will open and his session becomes active.

**F5: Student Login (separate for internship and placement).**

**Input:** name, id, password.

**Output:** student’s index page will open and his session becomes active.

**F6: Company Login.**

**Input:** name, id, password

**Output:** company’s index page will open and its session becomes active.

**F7: View student profile.**

**Input:** Click on ‘view profile’ button on the student’s index page.

**Output:** student’s profile will display along with his photograph and internship/placement status.

**F8: Update student profile.**

**Input:** Click on ‘update profile’ button on the student’s index page.

**Output:** an update form will open through which student can update his information.

**F9: Apply for company.**

**Input:** Name of the company for which the student wants to apply.

**Output:** If student satisfies the company requirement then the student’s name along with the company name will be inserted in the registered student table in the database.

**F10: View notification (Students).**

**Input:** Click on ‘Notifications’ button on the student’s index page.

**Output:** A notification panel will open where student can see various notifications like visiting companies, registered companies and general notifications from T & P cell.

**F11: Add Criteria for students by company.**

**Input:** branch, cgpa, salary, stipend, place of posting etc.

**Output:** all these details will be inserted into the database and will be used to check student’s eligibility for that company.

**F12: Company Approval by admin.**

**Input:** company name

**Output:** Company’s status will changed to ‘approved’ from ‘not approved’ and student can see that company in their notification panel.

**F13: Add Absentees**

**Input:** Id of the student

**Output:** Increment the absent of the student by 1 in the database of the student.

**F14: Change Student Status**

**Input:** Student Id and the company name in which he got selected.

**Output:** Student status will change to ‘selected’ from ‘not selected’ and database will be modified and a confirmation mail will be sent to the selected student.

**F15: Add notification.**

**Input:** notification or information for the student by admin.

**Output:** notification will be inserted in the database and student can view these notifications in their notification panel.

**F16: Add new admin.**

**Input:** information of the new admin.

**Output:** Entry of new admin details in the database.

**F17: View Notification (Statistics).**

**Input** Click on ‘Notifications’ button on the admin’s index page.

**Output:** A new page will open in which admin can see various statistics like registered students, selected student etc.

* 1. **External interface requirements:**

This section provides a detailed description of all inputs into system and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

* + 1. **User interfaces:**
* There is separate login for admin, student and company. Also there are separate login for students for placement and internship.
* The design or layout of every form will be very clear and very interactive to the user.
* If a student forgot his password then he can get new password through email by simply clicking on forgot password button.
* In the screen layout the background color will be very light and the graphics and font style will be in proper manner and well organized.
* If any user is logged in then no other user can log in at the same time on the same browser. Also no one can access the WebPages (that open after login) directly by address bar of the browser, this will provide better security.
  + 1. **Hardware interfaces:**
* The program will communicate with hard drive (the file system and database) via the appropriate PHP code.
* The user can communicate through browser using keyboard and a display through graphical interface displayed on user’s screen.
  + 1. **Software interfaces:**
* The software interface uses a XAMPP set-up.
* Operating system: Windows
* Web Server: Apache
* Database: MySQL
* Scripting Language: PHP, HTML, CSS and JavaScript

The product will host a local Apache web server where the user interface will be displayed via the web browser. The scripting language PHP will define size and overall set-up. It helps to start connection with database. It also helps to provide password encryption, constraints checking and provide security. CSS will be used to create background colors, border colors, text display and all the styling. HTML is to design basic outline of the application and also used to make forms. JavaScript will be used to control the slide show of images on main page of the web application. The MySQL database will store the data of various users and data can be retrieved back with the help of MySQL queries.

* + 1. **Communications interfaces:**

Communicational interfaces includes email, notification through web browser, forms etc. Communication security and encryption issues will handle by PHP i.e. the passwords are saved in the database as encrypted form and session variables are used to provide security, after and before login.

* 1. **Performance requirements:**
* The users must get the response within seconds. i.e. the response time of a particular function should be minimum.
* User should provide correct information so that no ambiguity will occur.
* Internet connection should be good so that email feature can work efficiently.
  1. **Design constraints:**
* Software Language Used: The languages that shall be used for coding the Training & Placement Management System are HTML, PHP.
* Development Tools: Will use the Xampp server with MySQL database.
  1. **Attributes:**

The Online Training & Placement Management System has many qualities attribute:-

* + **Usability**-The user interface of the software is very good with proper description and unambiguous display, so it is very easy to use.
  + **Correctness**-The results of the function are correct and accurate.
  + **Maintainability**-After the deployment of the project if any error occurs then it can be easily maintained by the developer.
  + **Portability**-The software can be deployed at any PC having windows.
  + **Reliability**-The software is reliable as it gives correct result always in all conditions.
  + **Reusability**-Modules in this software are designed in such a way that they can be reused easily.
  + **Robustness**-If there is any error in any module then it does not affect the remaining part of the software.
  + **Testability**-The software will be tested at every stage, by Alpha Testing, Beta Testing, Acceptance Testing.
  + **Productivity**-This software will produce desired result with accuracy.
  + **Timelines**- It will save much time and provide fast accessing of data.
  + **Security**-This system is highly secure as password for each user is saved in the database as encrypted form. Also session variables provide security

1. **Appendices:**

* **Data Dictionary:**

**admin**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Description** | **Null** | **Type** | **Default** |
| ADMIN\_ID *(Primary)* | Admin’s Id | No | varchar(50) | - |
| ADMIN\_NAME | Name of Admin | No | varchar(100) | - |
| A\_PASSWORD | Encrypted Password of admin | No | varchar(100) | - |
| POST | Post of admin in the college | No | varchar(100) | - |
| EMAIL | Email id of admin | No | varchar(100) | - |
| CONTACT\_NO | Contact number of admin | No | varchar(100) | - |
| DOB | Date of birth of admin | No | date | - |
| QUALIFICATION | Educational qualification of admin | No | varchar(100) | - |

**company**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Description** | **Null** | **Type** | **Default** |
| COMPANY\_ID *(Primary)* | Company id | No | varchar(50) | - |
| COMPANY\_NAME | Company name | No | varchar(100) | - |
| C\_PASSWORD | Encrypted password of company | No | varchar(100) | - |
| WEBSITE | Website of the company | No | varchar(100) | - |
| ADDRESS | Address of the company | No | varchar(100) | - |
| STATUS | Status of the company (visited or visiting the college) | Yes | varchar(50) | visiting |
| COMING\_DATE | Date on which company will visit college | No | date | - |
| APPROVAL | Company approved or rejected by admin | Yes | varchar(50) | not approved |

**student**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Description** | **Null** | **Type** | **Default** |
| STUDENT\_ID *(Primary)* | College Id of student | No | varchar(50) | - |
| S\_PASSWORD | Encrypted Password of student | No | varchar(100) | - |
| STUDENT\_NAME | Student Name | No | varchar(100) | - |
| GENDER | Gender of student | No | varchar(100) | - |
| DOB | Date of birth of student | No | date | - |
| EMAIL | Mail id of student | No | varchar(100) | - |
| ADDRESS | Residential Address of student | No | varchar(100) | - |
| CONTACT\_NO | Contact number of student | No | varchar(100) | - |
| BRANCH | Branch of student in the college | No | varchar(100) | - |
| TENTH\_PER | Percentage in 10th class | No | varchar(100) | - |
| TENTH\_PASS\_YEAR | 10th class passing year | No | int(11) | - |
| TWELTH\_PER | Percentage in 12th class | No | varchar(100) | - |
| TWELTH\_PASS\_YEAR | 12th class passing year | No | int(11) | - |
| CGPA | Current cgpa | No | double | - |
| PASSING\_YEAR | College passing year | No | int(11) | - |
| BACKLOGS | Current number of backlogs | No | int(11) | - |
| APPLY\_FOR | Apply for internship or placement | No | varchar(100) | - |
| STATUS | Selected in a company or not | Yes | varchar(50) | NS |
| APPLY\_COUNT | Total count of applied companies | Yes | int(11) | 0 |
| ABSENT | Total absents in the interview process after registered for a company. | Yes | int(11) | 0 |
| IMAGE | Image of student | No | longblob | - |

**companybranch**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Description** | **Null** | **Type** | **Default** |
| COMPANY\_NAME *(Primary)* | Name of company | No | varchar(100) | - |
| C\_TYPE *(Primary)* | Company is coming for internship or placement | No | varchar(50) | - |
| BRANCH *(Primary)* | Branch for which company is coming | No | varchar(50) | - |
| MIN\_CGPA | Minimum cgpa requirement for the branch | No | double | - |
| MAX\_BACKLOGS | Maximum backlogs allowed to the student | Yes | int(11) | 0 |
| MAX\_SALARY | Salary offered to the student | No | double | - |
| MAX\_STIPEND | Stipend offered to the student | No | double | - |
| JOB\_PROFILE | Job profile in the company | No | varchar(100) | - |
| PLACE\_OF\_POSTING | Place of Posting | No | varchar(100) | - |

**registered\_student**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Description** | **Null** | **Type** | **Default** |
| STUDENT\_ID *(Primary)* | College Id of student | No | varchar(50) | - |
| STUDENT\_NAME | Name of student | No | varchar(100) | - |
| COMPANY\_NAME *(Primary)* | Name of company | No | varchar(100) | - |

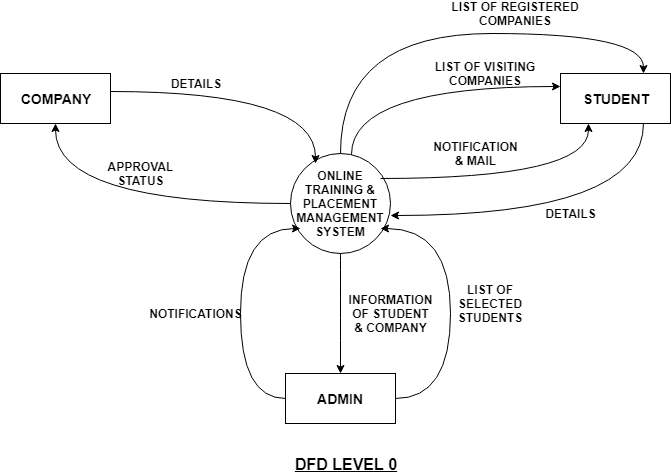
**selected\_student**

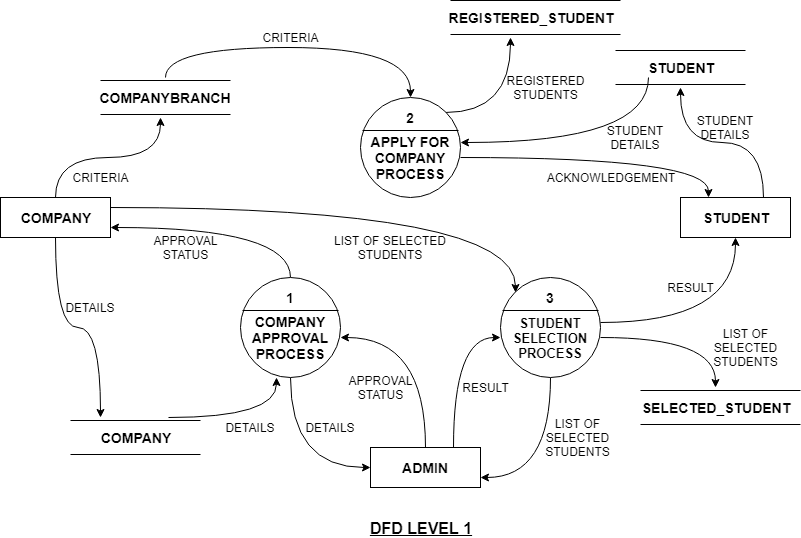
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Description** | **Null** | **Type** | **Default** |
| STUDENT\_ID *(Primary)* | College Id of student | No | varchar(50) | - |
| COMPANY\_ID | Company Id | No | varchar(100) | - |
| STUDENT\_NAME | Name of student | No | varchar(100) | - |
| COMPANY\_NAME | Name of company | No | varchar(100) | - |
| PACKAGE | Package of the student | No | double | - |

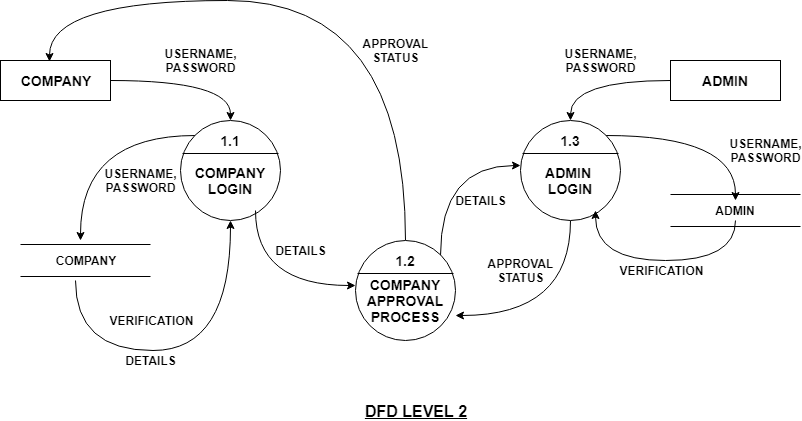
**notification**

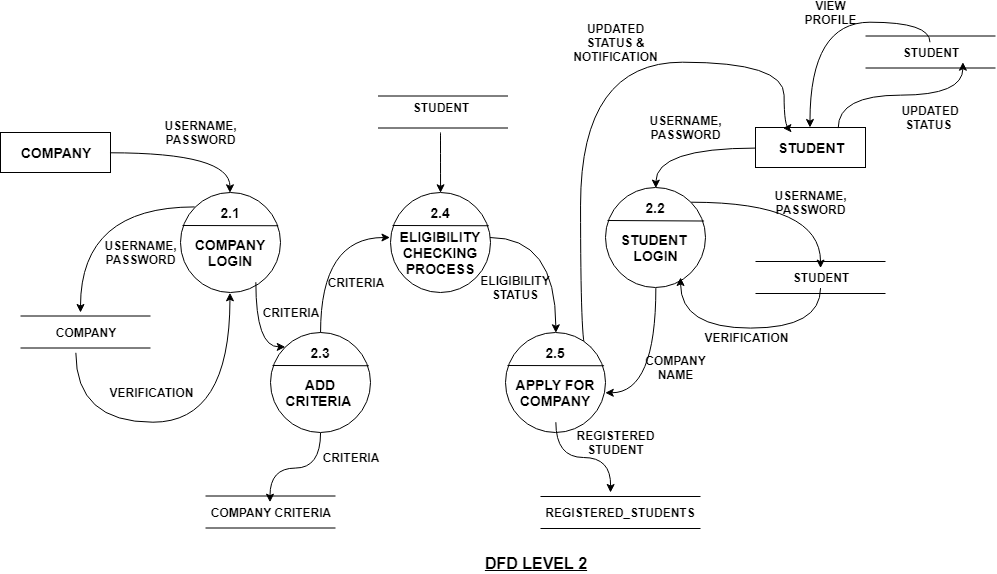
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Description** | **Null** | **Type** | **Default** |
| noti *(Primary)* | Notifications for students | No | varchar(200) | - |

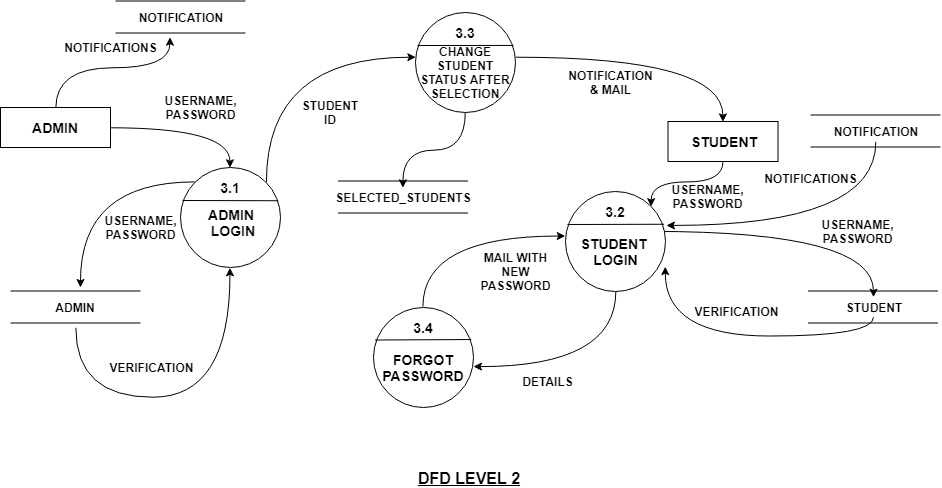
* **Data Flow Diagram:**

****

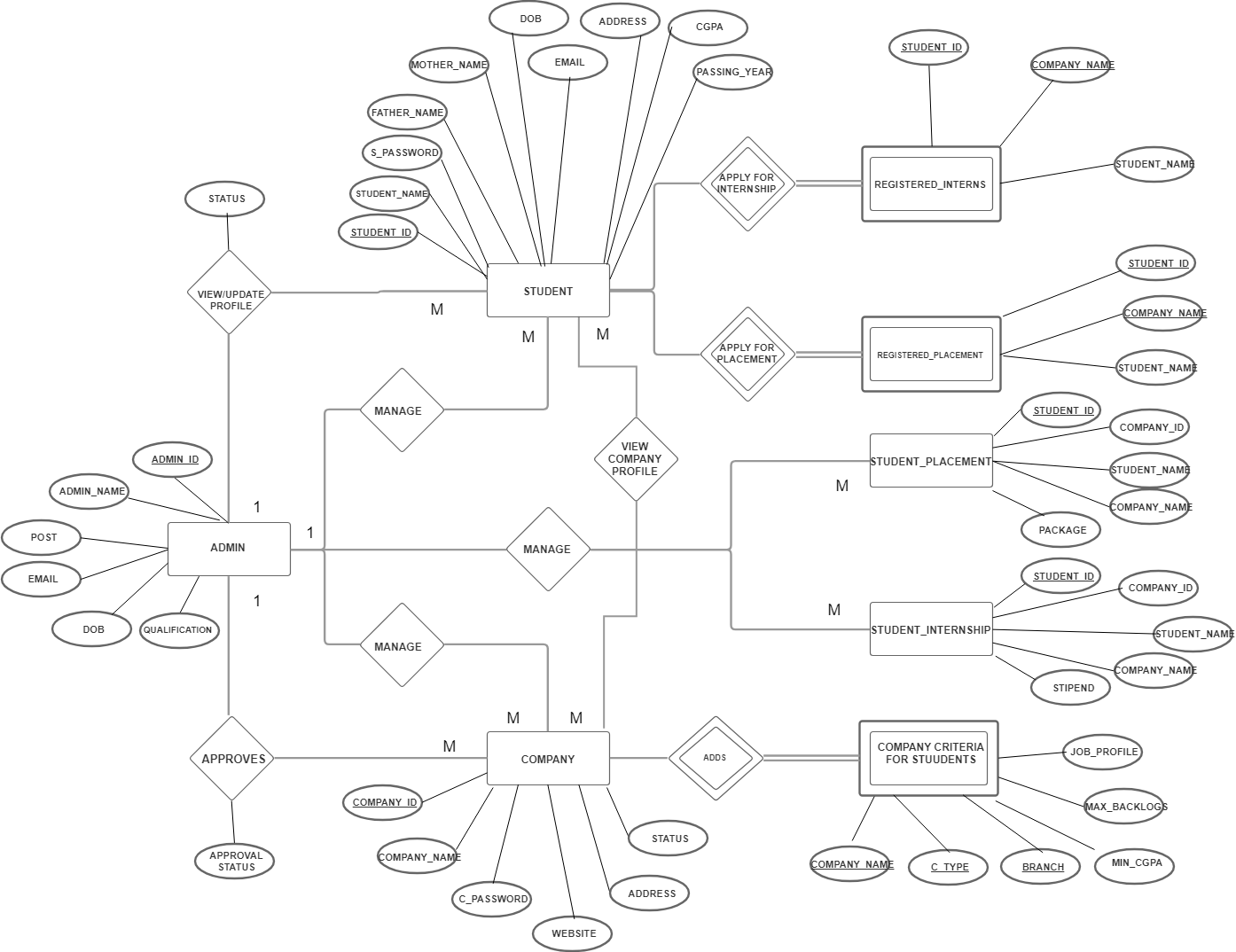
****

****

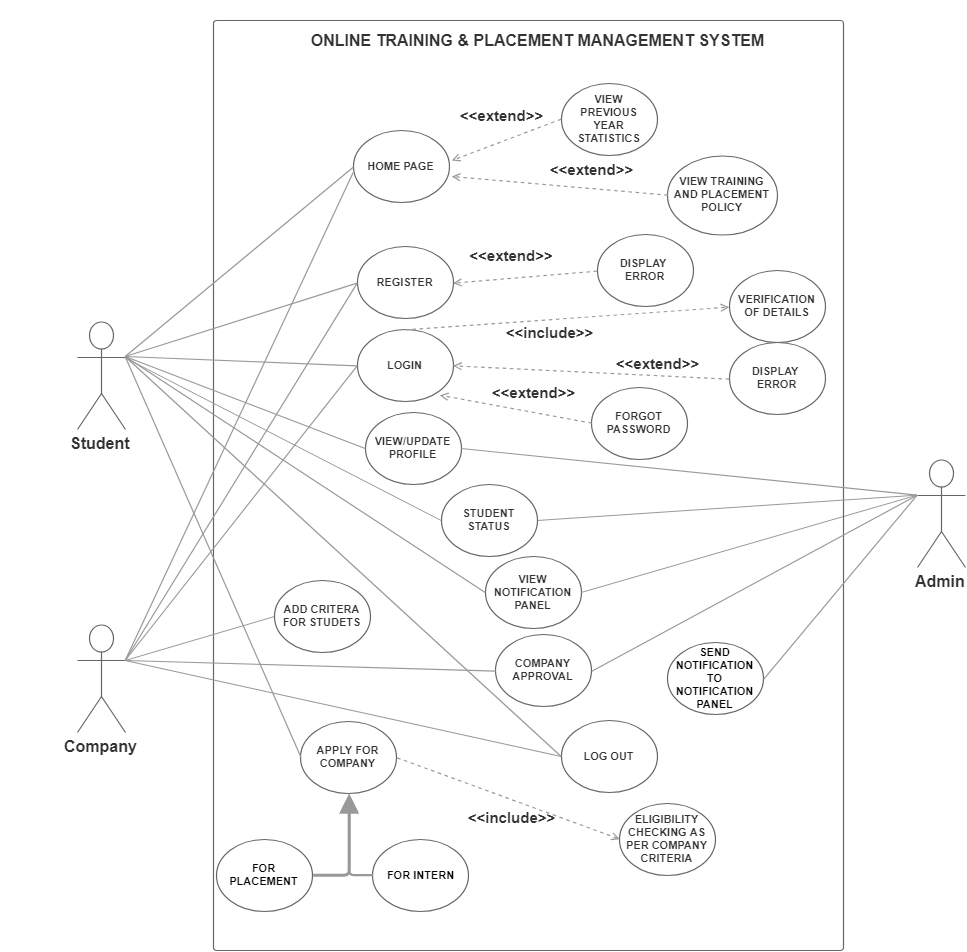
****

****

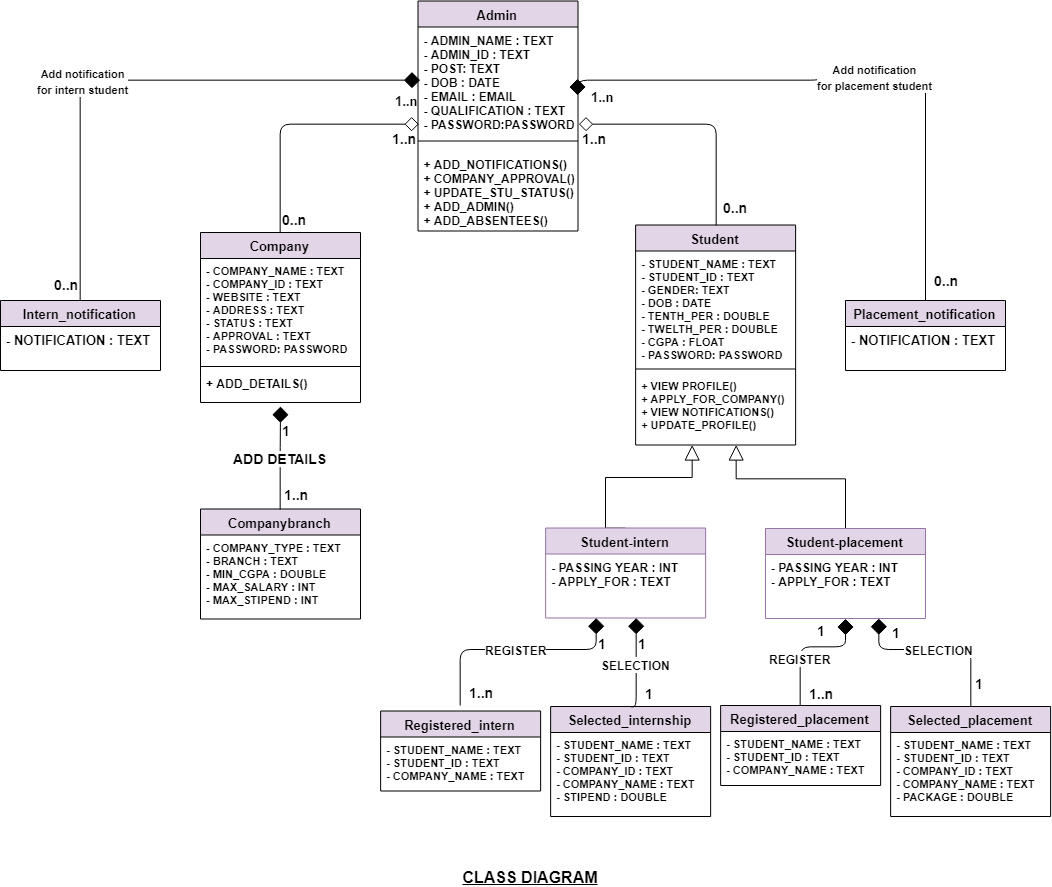
* **Entity-Relationship Diagram:**

****

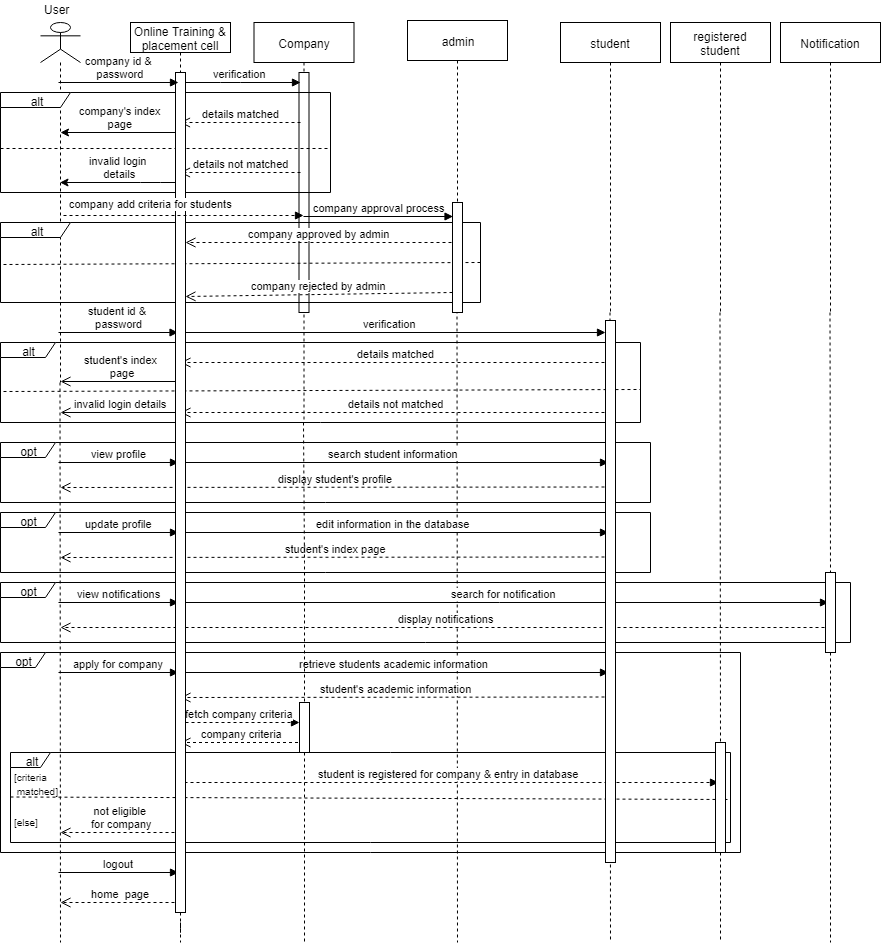
* **Use Case Diagram:**

****

* **Class Diagram:**

****

* **Sequence Diagram:**

****

* **Algorithm:**

**A. Web Page Algorithm (home page):**

1. Display\_choices (“1. Student 2.Admin 3.Company 4.View Training and

Placement Policy 5.Placement and Training Statistics”)

2. Take input (click) from user from above choices

3. Switch (choices)

4. Case1: A page opens that contains 3 buttons i.e. student registration,

student login for intern and student login for placement.

5. Case2: A page opens that contains 2 buttons i.e. admin registration and

admin login.

6. Case3: A page opens that contains 2 buttons i.e. company registration

and company login.

7. Case4: MNIT placement and training policy will open.

8. Case5: MNIT previous year training and placement statistics will open.

**B. Login Page Algorithm (student):**

1. IF (forgot password)

2. Click on forgot password link.

3. END IF

4. IF (Not registered)

5. Click on Signup link.

6. END IF

7. Take inputs from user i.e. student\_name, student\_id, student password.

8. IF (any input is not entered by student)

9. Display error “input is required”.

10. GOTO step 1

11. END IF

12. status = check\_validity ( student\_name, student\_id, student password)

13. IF (status=1)

14. Student is authentic and student’s index page will open

15. ELSE

16. display error “wrong information is entered”

17. END IF

**FUNCTION**: check\_validity

1. check\_validity ( student\_name, student\_id, student password)

2. IF (student\_name=database\_student[student\_name] AND

student\_id=database\_student[student\_id] AND

student\_ password =database\_student[student\_ password] )

3. return 1

4. ELSE

5. return 0

6. END IF

7. END check\_validity function.

**C. Registration Page Algorithm (company):**

1. IF (Already registered)

2. Click on Sign in link.

3. END IF

4. Take inputs from user i.e. company\_name, company\_ id, address

company\_password, confirm\_password, website, visiting date.

5. IF (any input is not entered by company)

6. display error “input is required”.

7. GOTO step 1

8. END IF

9. IF (website is not valid)

10. display error “Invalid Url”

11. END IF

12. IF (company\_password != confirm\_password)

13. display error “Password not match”

14. GOTO step 1

15. END IF

16. status = check\_validity (company\_name, company\_ id, address

company\_password, confirm\_password, website, Visiting date)

17. IF (status=1)

18. Company successfully registered and company’s index page will open.

19. ELSE

20. display error “company already registered”

21. END IF

**FUNCTION**: check\_validity

1. check\_validity (company\_name, company\_ id, address

company\_password, confirm\_password, website, Visiting date)

2. IF (company\_id=database\_company[company\_id] )

3. return 0

4. ELSE

5. insert all function arguments in the company database

6. return 1

7. END IF

8. END check\_validity function.

**D. Apply for company Algorithm (student):**

1. Student’s index page contains the “apply for company” button.

2. Student chooses the visiting company for which he wants to apply from drop down list.

3. IF (APPLY (company\_name))

4. display message “Successfully registered for the company ”

5. ELSE

6. display error “not eligible for the company”

7. END IF

**FUNCTION**: APPLY

1. APPLY (company\_name)

2. IF (student\_cgpa < company\_cgpa)

3. return 0

4. END IF

5. IF (student\_backlogs > company\_backlogs)

6. return 0

7. END IF

8. IF (student\_absents > 2)

9. return 0

10. END IF

11. IF (student\_id=registered\_student[student\_id])

12. display error “already registered”

13. return 0

14.END IF

15. insert student information in the database “registered\_student”

16. return 1

17. END APPLY function

* **Testing:**

**Black Box Testing:**

**1. Equivalence class Partitioning**

For CGPA:

|  |  |
| --- | --- |
| **Class** | **Expected Outcome** |
| <1 | Invalid |
| >=1 and <=10 | Valid |
| >10 | invalid |

For 10th class percentage:

|  |  |
| --- | --- |
| **Class** | **Expected Outcome** |
| <1 | Invalid |
| >=1 and <=100 | Valid |
| >100 | invalid |

1. **Boundary Value Analysis**

For CGPA:

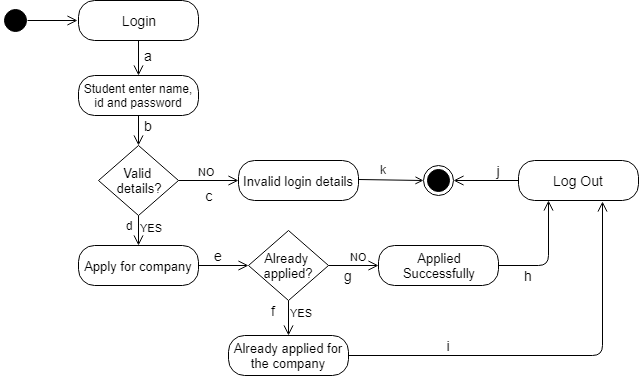
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Min-** | **Min** | **Min+** | **Normal** | **Max-** | **Max** | **Max+** |
| 0.9 | 1 | 1.01 | 7.11 | 9.99 | 10 | 10.1 |

For 10th class percentage:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Min-** | **Min** | **Min+** | **Normal** | **Max-** | **Max** | **Max+** |
| 0.9 | 1 | 1.01 | 80 | 99.9 | 100 | 100.01 |

**White Box Testing:**

White box testing for “apply for company” process:

****

**1. Statement Coverage:**

**Input:** Student details: company name

**Statement coverage test cases**:

**T1)** Valid student details: valid company name (i.e. not already applied).

**T2)** Invalid student details:

**T3)** Valid student details: invalid company name.

**2. Branch Coverage:**

**Input:** Student details: company name

**Branch coverage test cases**: **Paths**

**T1)** Valid student details: valid company name. (abdeghj)

**T2)** Invalid student details: (abck)

**T3)** Valid student details: invalid company name. (abdefij)