# Software Requirements Specification

For

# PlaceMentor

## Version 0

**Prepared by ‘CodeNatives’**

**Date: 20/03/2021**

## Contents

## Revision History

| Name | Date | Reason for Changes | Version |
|---|---|---|---|
| | | | |
| | | | |

# 1 Introduction

## 1.1 Purpose

The purpose of this application is to ease the process of student placements by integrating various stakeholders of the process on a single platform. It allows students to worry less about the placement process and focus more on their skill preparation. Given the large number of students and companies involved in the process, this application allows an efficient way for recruiters to have a look at student profiles and shortlist candidates according to their resume'. It also provides students with an opportunity to interact with the alumni of their institute working/who have worked in these companies for advice/feedback on their resume'. To summarize, ***PlaceMentor*** bridges the gap between the students and the recruiters in an effective manner avoiding the innumerable hassles in an offline setup.

## 1.2 Document Conventions

### Start Page:

Font – Calisto MT

### Remaining Pages:

Heading: Font – Calibri, Font size – 18

Sub-Heading: Font – Calibri, Font size – 16

Text Content: Font – Calibri, Font size – 14

Words in italics in most cases indicate important details with respect to the software implementation and are defined in the Glossary (see Appendix A: Glossary).

## 1.3 Intended Audience and Reading Suggestions

This SRS document can be referred to by all students of the institute, recruiters, developers, college administrations, placement cells and in general anyone who wishes to understand how the software application functions. The forthcoming sections describe all internal, external, functional and non-functional information with respect to ***PlaceMentor***. Readers are advised to follow this document in the order as given, starting with the operating system and dependencies required for this project, up to specific details of various Use-Cases and Class descriptions involved in its implementation. Use-Case and Class diagrams provided within the SRS will make the implementation details lucid.

## 1.4 Project Scope and Need

***PlaceMentor*** provides college administrations and placement cells an effective tool to manage the student placement process effectively. Using this application, students can provide their job preferences along with their resume' quite effectively as compared to that in an offline process. The *College Administration* ensures that students do not miss out on any updates with respect to the placement process. Recruiters can view student profiles along with their resume' for short listing viable candidates. In addition to these features, students also have an opportunity to contact registered alumni and review their resume' or ask for advice, thus helping them to build a strong network. Such software efficiently manages the preferences of each individual student and encourages hiring companies to participate in a smooth process.

## 1.5 References

- **IEEE Recommended Practice for Software Requirements Specifications, 1998, ISBN 0-7381-0332-2**

# 2. Overall Description

## 2.1 Product Perspective

PlaceMentor is a new product developed to help the placement department of college, students and the recruiters to ease the placement process, and the process of getting updates for students; and notifying the students, for the institute. The product is designed as a replacement of offline based placement process in the institute.  It is an open source project and expected very active developer team to support it and provide feedback to users. It was developed to run on Windows, Mac OS X and Linux.

## 2.2 Product Functions

- Users can register for an account by providing the necessary details.
- Users can login to an existing account
- Users specifically students, can upload a resume’
- To read the notifications received by a user
- To view the details of a company registered in the system.
- To view the feedback given by an alumnus to a particular student.
- Establish communication between various Users by following some specified rules.
- To apply for a job position mentioned by a particular company.
- To provide feedback on student profiles or share experiences about particular job profiles
- To provide details about the job role and the associated requirements.
- To access student profiles and view their details.
- To shortlist students according to profiles/resume’ and notify them about the status of selections.
- To post important updates about the placement process visible to all the Users in the system.
- To check the profiles of registered alumni and the organisations he/she has worked with in the past.
- To load the messages received by a particular user.

## 2.3 User Classes and Characteristics

- Students: Students of the Institute which is using the PlaceMentor to ease its placement process and keep updated to the placement process of the institute
- Company Administrators: Admin of recruitment companies who wants to release job profiles and recruit students and to have a glance at the talent in the institute
- Institute Administrators: Admin of institute which wants to ease the placement process and have hassle free process of notifying the students
- Alumni: Institute's alumni, who have worked for various companies, can help by writing in their feedbacks and the interested students may contact them for further information

## 2.4 Operating Environment

- Windows 2000
- Windows XP
- Windows Vista
- Windows 7,8,10
- Mac OS X
- Linux

## 2.5 Design and Implementation Constraints

TBD

## 2.6 User Documentation

TBD

## 2.7 Assumptions and Dependencies

PlaceMentor is developed as a web-based application and therefore requires a browser to be installed on the user's system. However the latest stable versions of any popular browsers are expected to have the latest things to work. This applies to Windows and Linux users. On Mac OS X, Java is bundles with the application.

# 3. External Interface Requirements

## 3.1 User Interfaces

- Front-end software: Any Browser that runs HTML & JavaScript

## 3.2 Hardware Interfaces

The minimum hardware requirements of PlaceMentor are that these specifications can handle running a browser which supports HTML & JavaScript.

## 3.3 Software Interfaces

For running the software there must be a proper browser. And a SQL based database could be used for relational databases.

## 3.4 Communication Interfaces

PlaceMentor requires an internet connection to interact with the backend. This project supports all types of web browsers. The client device must preferably have browsers like IE9 or above, Mozilla Firefox (version 3.5 or above), or Opera 10 or Chrome (version 29 or above)

# 4 System Features

## 4.1 Register an account

### 4.1.1 Functional Requirements

**Use Case Name:** Register

**Purpose: Users** can register for an account by providing the necessary details.

**Pre-condition: User** has valid details required for registration

**Post-condition: User** details will be registered in the system and an account will be created.

**Actors: Student**, *InstituteAdmin*, *Alumni*, *CompanyAdmin*

**Failure Conditions: User** does not have valid details to register

**Basic Path:** *User* will be asked about the type of account to be created. Depending on this choice, the user will be required to fill in the required details and submit it to complete the process of account creation.

### 4.1.2 Stimulus/Response

User must click on the register button which will allow the system to verify the details and store them.

## 4.2 Login to an account

### 4.2.1 Functional Requirements

**Use Case Name:** Login

**Purpose:** *Users* can login to an existing account

**Pre-condition:** *User* has the correct credentials for logging in

**Post-condition:** *User* will be logged into his/her account

**Actors:** *Student*, *InstituteAdmin*, *Alumni*, *CompanyAdmin*

**Basic Path:** *User* will be asked for his name and password, in order to login to his/her account. After the correct details are entered, the user will be logged into his/her account.

**Failure Conditions:** *User* does not have valid credentials to login

### 4.2.2 Stimulus/Response

*User* must click on the login button and the system initiates the user profile.

## 4.3 Upload Resume'

### 4.3.1 Functional Requirements

**Use Case Name:** Upload Resume'

**Purpose:** *Users* specifically students, can upload a resume'

**Pre-condition:** *User* is logged into his/her profile

**Post-condition:** Resume' will be uploaded in the system

**Actors: Student**

**Basic Path:** *Students* will have an option to upload their resume' in a valid format. Once entered, the pdf is stored in the system.

**Failure Conditions: User** does not upload the resume' in a proper format.

### 4.3.2 Stimulus/Response

*User* must submit the pdf which allows the system to store the resume in the database.

## 4.4 Load Notifications

### 4.4.1 Functional Requirements

**Use Case Name:** Load Notifications

**Purpose: To** read the notifications received by a user

**Pre-condition:** User is logged into his/her profile

**Post-condition:** Notifications will be displayed on the screen

**Actors: Student**, Institute Admin, Alumni, Company Admin

**Basic Path:** Once the user logs into his/her account, he/she can select the notifications box to display the received messages.

**Failure Conditions: No** such failure condition

### 4.4.2 Stimulus/Response

*User* must click on the inbox button to view the notifications. As a result, the notifications will be fetched and displayed on the screen.

## 4.5 View Company Profiles

### 4.5.1 Functional Requirements

**Use Case Name:** View Company Profile

**Purpose: To** view the details of a company registered in the system.

**Pre-condition:** *User* is logged into his/her profile and selects the required company.

**Post-condition:** Details will be displayed on the screen

**Actors: Student**, *Alumni*, *InstituteAdmin*

**Failure Conditions:** No such failure condition

**Basic Path:** Once the user logs into his/her account, he/she can search a particular company by its name. As a result, the user can select the matching company and request its details.

### 4.5.2 Stimulus/Response

User must select the required company name, upon which the system will display the details associated with the company on the screen.

## 4.6 View Alumni Feedback

### 4.6.1 Functional Requirements

**Use Case Name:** View Alumni Feedback

**Purpose:** To view the feedback given by an alumnus to a particular student.

**Pre-condition:** User is logged into his/her profile

**Post-condition:** Obtained feedback will be displayed on the screen

**Actors:** Student

**Failure Conditions:** No such failure condition

**Basic Path:** A student will be prompted to click on the notification box to display any feedbacks received from alumni of the institute.

### 4.6.2 Stimulus/Response

User must click on the notification inbox to display the feedbacks received by a particular alumnus.

## 4.7 Communication between users

### 4.7.1 Functional Requirements

**Use Case Name:** Message

**Purpose:** Establish communication between various *Users* by following some specified rules.

**Pre-condition:** User is logged into his/her profile and selects some other valid user with whom chatting is allowed.

**Post-condition:** The message sent by a particular *User* is delivered to the target *User*.

**Actors: Pair** of *Users* specified in the basic path

**Basic Path:** In the system, communication through sharing of text messages takes places among three pairs of accounts:

- *Institute Admin* to *Student*
- *Student* to *Alumnus* and vice versa
- *Company Admin* to *Institute Admin* and vice versa

On selecting a valid *User*, a message will be delivered to the inbox of the receiving *User*.

**Failure Conditions:** No such failure condition

### 4.7.2 Stimulus/Response

*User* must select a valid *User* (as indicated in the basic path) to send the message. As a result, the message will be delivered to the correct target account.

## 4.8 Apply to a job position

### 4.8.1 Functional Requirements

**Use Case Name:** Apply to Job Position

**Pre-condition: User** is logged into his/her profile

**Purpose:** To apply for a job position mentioned by a particular company.

**Post-condition: Applications** interested in the given job role will be sent to the *Company Admin*

**Actors:** *Student*

**Basic Path:** Students interested in a particular company or a job profile can always search for these by name. The job positions are available on the company's profile and can be applied to considering the student satisfies the minimum criteria. On submitting the application, the *Student* would be notified appropriately.

**Failure Conditions:** Students must satisfy the minimum criteria associated with a job position/role to submit their applications.

### 4.8.2 Stimulus/Response

*Students* must apply to a job profile by selecting the *Apply* button, upon which the application will be verified by the system and finally delivered to the *Company Admin.*

## 4.9 Give feedback

### 4.9.1 Functional Requirements

**Use Case Name:** Give feedback

**Purpose:** To provide feedback on student profiles or share experiences about particular job profiles

**Pre-condition:** *User* is logged into his/her profile

**Post-condition:** Feedback submitted will be delivered to the target users (*Student*/*CompanyAdmin*)

**Actors: Alumni**

**Basic Path: The** alumnus can search for student profiles and share genuine feedback on student resume'. He can also respond to any queries received by particular students. In addition to this, he can also search through various companies and share his/her experiences through a messaging feature, visible to all Users, which will be displayed on the company profile page.

**Failure Conditions:** No such failure condition

### 4.9.2 Stimulus/Response

Alumnus must select the required target user and submit the message using the *Send* option. As a result, the system will deliver the feedback to the target User.

## 4.10 Provide job description

### 4.10.1 Functional Requirements

**Use Case Name:** Provide job description

**Purpose:** To provide details about the job role and the associated requirements.

**Pre-condition:** User is logged into his/her profile

**Post-condition:** Provided Job description is sent to the Institute Admin for approval and is then made available for students.

**Actors:** Company Admin

**Basic Path: A** company can create a new job opening by selecting an appropriate option on its profile page. As a result, it must provide all details associated with the job role. The job role shall be visible to all *Users* except among other companies.

**Failure Conditions:** No such failure condition

### 4.10.2 Stimulus/Response

*Company Admin* must fill in the necessary details and click on the *Submit* button. On submitting the job profile, system will deliver it to the Institute Admin for approval.

## 4.11 View student profile

### 4.11.1 Functional Requirements

**Use Case Name:** View student details

**Purpose: To** access student profiles and view their details.

**Pre-condition:** User is logged into his/her profile and selects a student

**Post-condition:** Details of a student are displayed along with the resume'

**Actors:** Al*umni, Company Admin, Institute Admin*

**Basic Path:** Search for a *Student* by his/her name and select the appropriate user to get the details.

**Failure Conditions:** No such failure condition

### 4.11.3 Stimulus/Response

User must select a student profile after which the system displays his/her details along with the resume'.

## 4.12 Selection of students for a job profile

### 4.12.1 Functional Requirements

**Use Case Name:** Selecting applications

**Purpose: To** shortlist students according to profiles/resume’ and notify them about the status of selections.

**Pre-condition:** User is logged into his/her profile and selects a student

**Post-condition:** A student may be accepted or rejected, thus notifying him/her appropriately

**Actors:** Company Admin

**Basic Path: Select** students among those who have applied for the job role and notify all the participants with respect to their status.

**Failure Conditions:** No such failure condition

### 4.12.2 Stimulus/Response

A student may be accepted/rejected by the Company Admin by selecting the corresponding buttons on his profile. As a result, each student will be notified about the status of his/her selection.

## 4.13 Post placement updates

### 4.13.1 Functional Requirements

**Use Case Name:** Post placement updates

**Purpose:** **T**o post important updates about the placement process visible to all the *Users* in the system.

**Pre-condition:** Institute Admin must be logged into the account and manage the message/updates to be posted

**Post-condition:** Once the updates are posted, these will be visible to all users.

**Actors:** *Institute Admin*

**Basic Path:** The *Institute Admin* can post important updates about the placement process not limited to the new job openings by different companies. These messages are notified to every user by the system.

**Failure Conditions:** No such failure condition

### 4.13.2 Stimulus/Response

The Institute Admin must select the associated button to post the updates, upon which all the users will be notified and the message will be published to all the Users.

## 4.14 View Alumni profile

### 4.14.1 Functional Requirements

**Use Case Name:** View Alumni Profile

**Purpose:** To check the profiles of registered alumni and the organizations he/she has worked with in the past.

**Pre-condition:** The User must be logged into the system and select an *Alumnus* profile.

**Post-condition:** Once an alumnus account is selected, the details of the associated account will be displayed.

**Actors: Users** except *Alumni*

**Failure Conditions:** No such failure condition

### 4.14.2 Stimulus/Response

The user must select an appropriate alumnus account by clicking on his profile. As a result, the information will be displayed on the screen.

## 4.15 Load Messages

### 4.15.1 Functional Requirements

**Use Case Name:** Load Messages

**Purpose: To** load the messages received by a particular user.

**Pre-condition: Users** must be logged in.

**Post-condition: Load** the messages received by a user account.

**Actors:** Al*umni, Company Admin, Institute Admin*

**Basic Path:** Given the user has logged into the system, check for any received messages by selecting the appropriate option and display them on the screen.

**Failure Conditions:** No such failure condition

### 4.15.2 Stimulus/Response

For each user, select the appropriate button/ option to display the list of received messages.

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

A relational database is being used and it is necessary that the database has to be normalized. If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme.

## 5.2 Safety Requirements

To ensure that no one of PlaceMentor's users loses any data while using PlaceMentor (due to a crash or a bug of some kind) the application will be well tested before the release. There will be a bug tracker in the developers page where users can report any bugs they have encountered so that the developers can fix it in the next release.

## 5.3 Security Requirements

Security systems need database storage just like many other applications. However, for the special requirements of the security, the database partners have been chosen properly.

Proper methods like hashing will be incorporated with the passwords, so that in the extreme case even there is data breach, data cannot be accessed.

## 5.4 Software Quality Attributes

PlaceMentor provides the users with both simple and advanced features. It is well designed and easy to use interface. However, users must already have a basic knowledge of using an interactive basic webpage before using it.

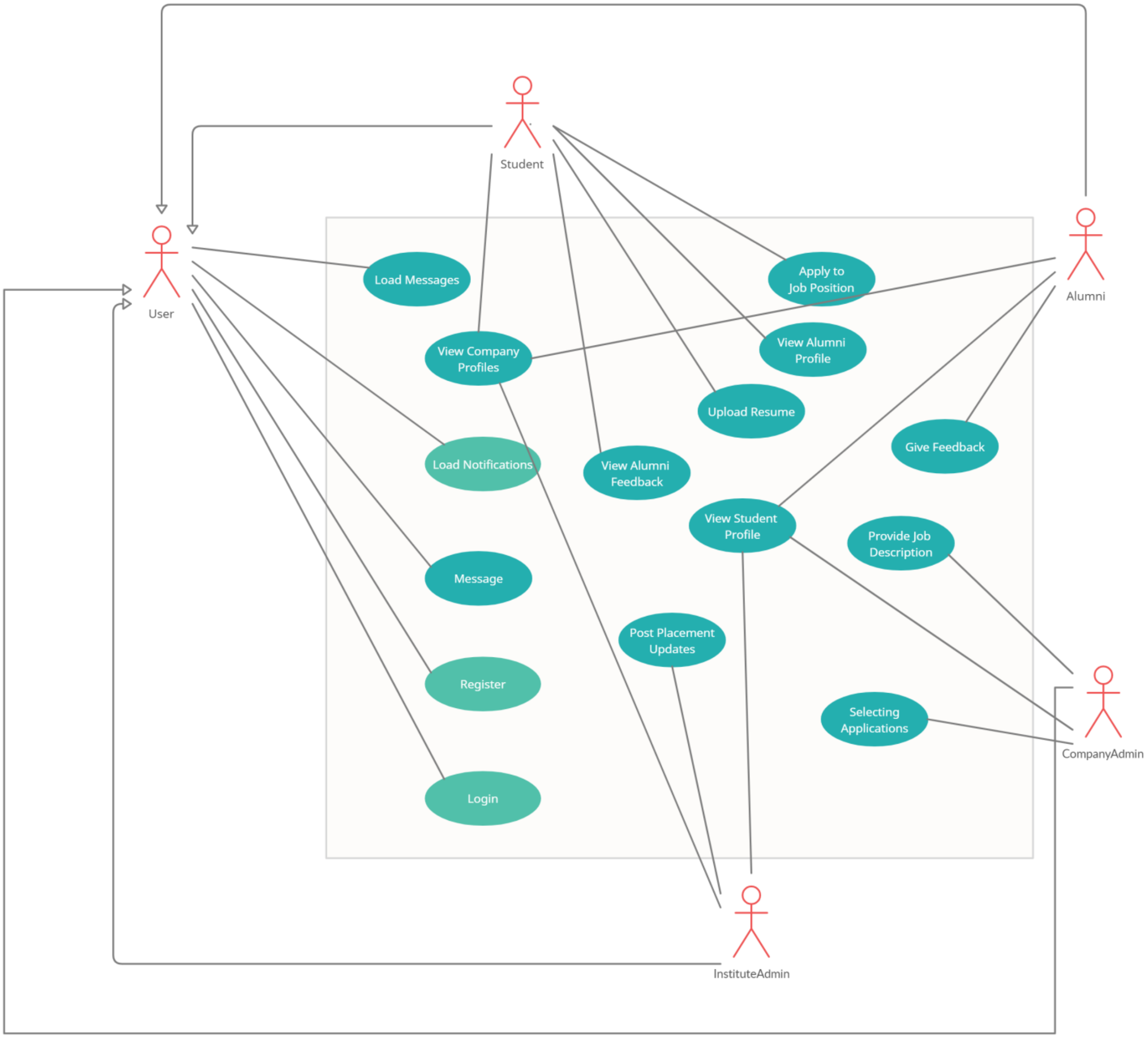
# 6. Other Requirements

## APPENDIX A: GLOSSARY

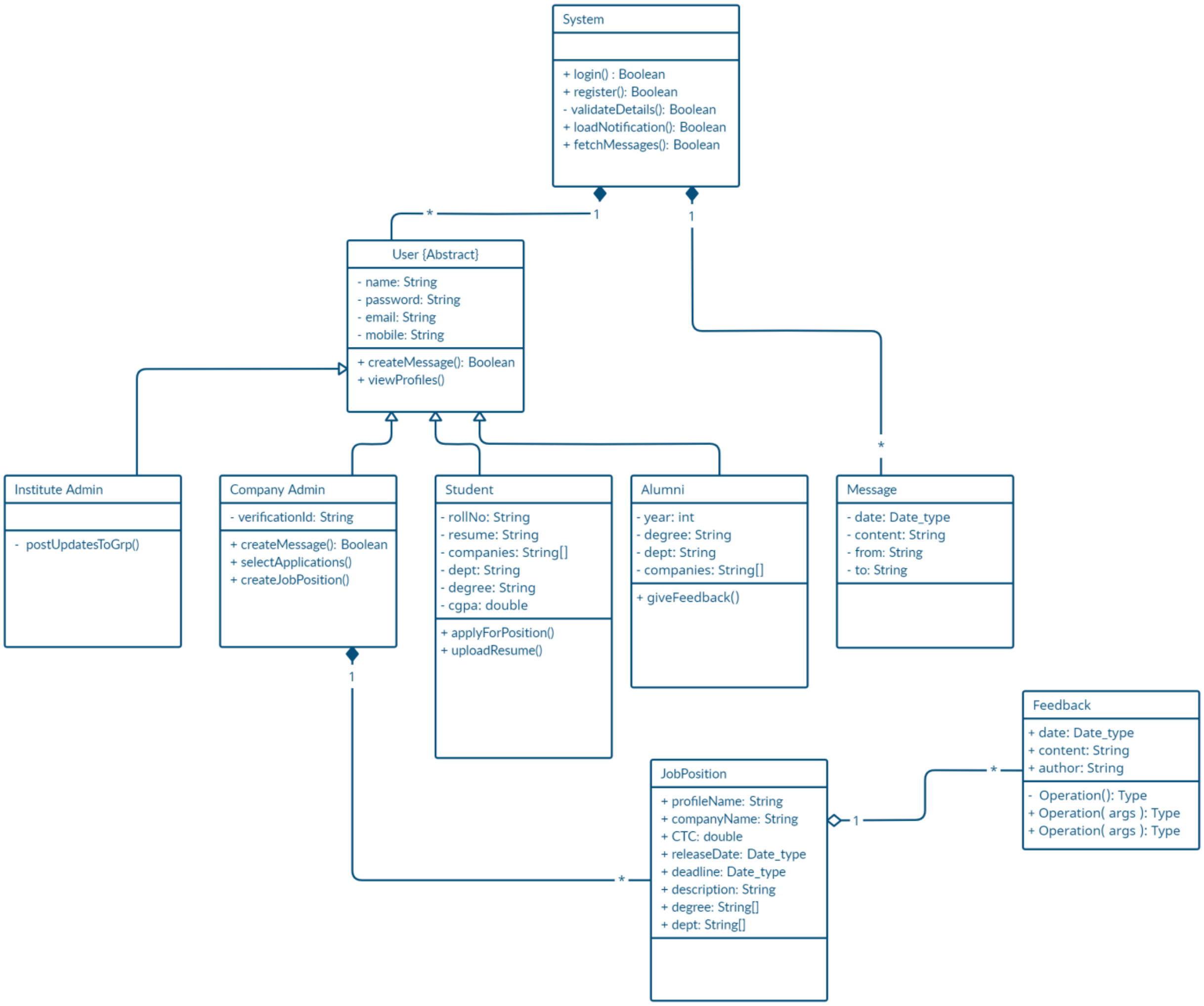
- Users : Concerned with all the users in the system which can interact with the system
- Users can be divided into multiple derived classes of the form Alumni, Company Admin, Institute Admin, Students.
- Each of these class specialisations have attributes and methods which define the operations individual objects related to these classes will perform.
- All their use-cases are specifically mentioned in the Use-Case Diagram in Appendix B.

# APPENDIX B:

## USE CASE DIAGRAM OF THE PRODUCT:



# CLASS DIAGRAM


System
+ login() : Boolean
+ register(): Boolean
- validateDetails(): Boolean
+ loadNotification(): Boolean
+ fetchMessages(): Boolean
*
1
1
User {Abstract}
- name: String
- password: String
- email: String
- mobile: String
+ createMessage(): Boolean
+ viewProfiles()
*
Institute Admin
- postUpdatesToGrp()
Company Admin
- verificationId: String
+ createMessage(): Boolean
+ selectApplications()
+ createJobPosition()
Student
- rollNo: String
- resume: String
- companies: String[]
- dept: String
- degree: String
- cgpa: double
+ applyForPosition()
+ uploadResume()
Alumni
- year: int
- degree: String
- dept: String
- companies: String[]
+ giveFeedback()
Message
- date: Date_type
- content: String
- from: String
- to: String
1
Feedback
+ date: Date_type
+ content: String
+ author: String
- Operation(): Type
+ Operation( args ): Type
+ Operation( args ): Type
*
1
JobPosition
+ profileName: String
+ companyName: String
+ CTC: double
+ releaseDate: Date_type
+ deadline: Date_type
+ description: String
+ degree: String[]
+ dept: String[]
*