**INTERNSHIP REPORT**

**A Report submitted**

By

**SANDEEP KUMAR MOHANTY**

**1501021118**

**CSE -B**

**Under the Guidance of**

**Mr. Ajay Kumar**

**Professor, Department of CSE**



In Partial fulfilment of the requirements for the Degree of

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE & ENGINEERING**

**DIT UNIVERSITY, DEHRADUN**

(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

**Mussoorie Divsersion Road, Dehradun, Uttarakhand248009, India**

# CERTIFICATE

This is to certify that the project report entitled **“e-Trainer Pool Management”** being submitted by **“Sandeep Kumar Mohanty”** in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the DIT University is a record of bona fide work carried out by them under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Mr. Ajay Kumar Mr. A.K Gupta

## Project Guide Designation Head of Department

**Computer Science Computer Science**

## CANDIDATES DECLARATION

I hereby certify that the work, which is being presented in the project report,

Entitled “e-Trainer Pool Management”, in partial fulfillment of the requirement for the award of the Degree of Bachelor of Technology and submitted to the university is an authentic record of out own work carried out during the period 19-Jan-2019 to 15-May-2019 under the supervision of Mr. Manoj Kulkarni (Trainer).

Date: Signature of the Candidate

This is to certify that the above statement made by the candidate is correct to the best of my knowlwdge.

Date: Signature of the Supervisor

viii

## TABLE OF CONTENTS

Title Page No.

CERTIFICATE……………………………………………………..……i

ABSTRACT………………………………………………………..……xi

ACKNOWLEDGEMENT………………………………………………xii

LIST OF TABLES………………………………………………………xiii

LIST OF FIGURES……………………………………...……………...xiv

ABBREVIATIONS………………………………………………………xv

NOTATIONS………………………………………………..…………….xvi

### CHAPTER 1 INTRODUCTION

1.1 Purpose………………………………………………………………….1

1.2 Objective…………………………………………………………………2

1.3 Motivation……………………………………………………………….3

1.4 Definition and Overview…………………………………………………4

### CHAPTER 2 OVERALL DESCRIPTION

2.1 Project Perspective………………………………………………………7

2.2 Project Functions…………………………………………………………8

2.3 E-R, DFD Diagrams, Class Diagram, Use Case…………………………10

2.4 Constraints and Assumptions…………………………………………….12

### CHAPTER 3 EXTERNAL INTERFACE REQUIREMENTS

3.1 User Interfaces…………………………………………………………14

3.2 Hardware Interfaces……………………………………………………15

3.3 Software Interfaces………………………………………………………17

3.4 Communication Interfaces………………………………………………20

**CHAPTER 4 SYSTEM FEATURES**

4.1 System Feature 1………………………………………………………24

### CHAPTER 5 OTHER NONFUNCTIONAL REQUIREMENTS

5.1 Performance Requirements………………………………………….27

5.2 Safety Requirements…………………………………………………28

5.3 Security Requirements………………………………………………..29

5.4 Software Quality Attributes…………………………………………...29

### CHAPTER 6 CONCLUSION AND FUTURE WORK

6.1 Conclusions……………………………………………………………34

6.2 Scope for Future Work…………………………………………………35

**REFERENCES**……………………………………………………………37

APPENDIX………………………………………………………………….38

**ACKNOWLEDGEMENT**

For the successful completion of this project, I'd extend a sincere thanks to our project guide as well as trainer Mr. Manoj Kulkarni Sir, who has been there with us while building the complete code from scratch. Without his guidance and teaching, it'd have been impossible of us to create the project.

I'm also grateful to our batch owners and people who've helped us out in every way possible, Shilpa Mahajani Ma’am, Mohit Kariya Sir, Gerard Thomas Sir and Komal Sanjay Pawar Ma’am. Without them, internship at such a huge company would have not been completed successfully. They ensured our smooth functioning and handled all the quirks and doubts while the sessions which helped us a lot in order to calmly finish the project.

I'd also like to thank my college for giving us such an amazing opportunity to work through the college semester in order to get a corporate experience and education. It is only because of his thought that we were able to gain such a training with a parallel balance of college academics.

**ABSTRACT**

This report is aimed at documenting the “Trainer Pool Management System” that has been developed as a solution of a common interface between the allocators and the nominees. While starting with the project we went through some similar existing platforms that inspired the project to be developed in the first place, however it provided the motivation to do it even better. The system we have created provides a common platform for logging in / signing up as the predefined user types that are as mentioned, the allocators and the nominees and thereafter surfing their corresponding dashboards in order to complete their tasks. Their dashboards comprise of windows for their respective functions that is easy to navigate and control since the main objective of the management system has been kept as the same. In this project, the function of allocator is performed by the ‘Admin’ and nominees are the ‘Trainers’ and the ‘SMEs’. The trainers based on their skill sets and dates of availability, request for training a particular course and the SMEs nominate themselves to be a trainer with the same. Both of these applicants can update their profiles, skill sets or even dates once they have sent the request/ nomination. Also, the trainer can be allocated to multiple courses that fits in his available timeline and course schedule. The admin performs the task of managing a calendar which includes details of the program / course and the trainer allocated for the respective dates. His another function includes managing the database of trainers and SMEs along with upgrading the eligible and available SMEs to trainers as well as allocating the trainers to program / course based on his request and current requirement.

# LIST OF TABLES

**Table No. Title Page No.**

2.3.1 Computer Vision Metrics 13

# LIST OF FIGURES

**Figure No. Title Page No.**

1.3.1 Computer Vision 3

1.4.1.1 Definition 4

1.4.2.1 Overview 6

2.2.1 PieChart 9

2.3.1 Image Processing 10

2.3.2 Industrial Computer Vision 11

2.3.3 Processing 11

2.3.4 Image Processing 12

2.4.2.1 Color Image 13

3.1.1 User Interface Mac 14

3.2.1 Processor 15

3.2.2 RAM 16

3.2.3 HardDisk 17

3.3.1 Anaconda 18

3.3.2 Python 19

3.3.3 Image Example 20

# ABBREVIATIONS

CBC Cipher Block Chaining CFB Cipher Feedback Mode

DDES Double Data Encryption Standard

DES Data Encryption Standard

ECD Electronic Codebook Mode

IEEE Institute of Electrical and Electronics Engineers

RMSE Root Mean Square Deviation

RFE Recursive Feature Elimination

GBM Gradient Boosting Machine

SVM Support Vector Machines

RF Random Forest

## 1. INTRODUCTION

**1.1 Purpose of this document**

This document is aimed at:

* Providing the necessary inputs to the detailed requirements gathering phase and further on for the SDLC processes.

The purpose of this document is to systematically capture requirements for the project and the system to be developed. Functional requirements are captured in this document.

**1.2 Project Overview**

1.2.1 Objectives

**Administrator**

Below are the objectives of Administrator Module:

* User registration & credential authentication
* Allocate trainer with particular program
* Generate confirmation of trainer for allocated program
* Generate Calendar list of allocated Trainers

**Trainer**

Below are the objectives of Trainer Module:

* User registration & credential authentication
* Procurement of trainer information with skill group and availability
* Allocated program list for particular Trainer

**SME**

Below are the objectives of SME Module:

* User registration & credential authentication
* Procurement of trainer skill information and availability
* Allocated program list for particular SME

**CHAPTER 2**

**LITERARY SURVEY**

1. **Literary Survey**

**2.1Methodology**

* SDLC:

In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), a software development process is the process of dividing [software development](https://en.wikipedia.org/wiki/Software_development) work into distinct phases to improve [design](https://en.wikipedia.org/wiki/Software_design), [product management](https://en.wikipedia.org/wiki/Software_product_management), and [project management](https://en.wikipedia.org/wiki/Software_project_management). It is also known as a **software development life cycle**. The methodology may include the pre-definition of specific [deliverables](https://en.wikipedia.org/wiki/Deliverable) and artifacts that are created and completed by a project team to develop or maintain an application.

Most modern development processes can be vaguely described as [**agile**](https://en.wikipedia.org/wiki/Agile_software_development). Other methodologies include [*waterfall*](https://en.wikipedia.org/wiki/Waterfall_model)*,*[*prototyping*](https://en.wikipedia.org/wiki/Software_prototyping)*,*[*iterative and incremental development*](https://en.wikipedia.org/wiki/Iterative_and_incremental_development)*,*[*spiral development*](https://en.wikipedia.org/wiki/Spiral_development)*,*[*rapid application development*](https://en.wikipedia.org/wiki/Rapid_application_development)*, and*[*extreme programming*](https://en.wikipedia.org/wiki/Extreme_programming)*.*

Some people consider a life-cycle "model" a more general term for a category of methodologies and a software development "process" a more specific term to refer to a specific process chosen by a specific organization.

* Agile:

"Agile software development" refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve via collaboration between self-organizing cross-functional teams

Agile software development uses iterative development as a basis but advocates a lighter and more people-centric viewpoint than traditional approaches. Agile processes fundamentally incorporate iteration and the continuous feedback that it provides to successively refine and deliver a software system.

There are many agile methodologies, including:

* [Dynamic systems development method](https://en.wikipedia.org/wiki/Dynamic_systems_development_method) (DSDM)
* [Kanban](https://en.wikipedia.org/wiki/Kanban_(development))
* [Scrum](https://en.wikipedia.org/wiki/Scrum_(development))
* Client–server model:

**Client–server model** is a [distributed application](https://en.wikipedia.org/wiki/Distributed_application) structure that partitions tasks or workloads between the providers of a resource or service, called [servers](https://en.wikipedia.org/wiki/Server_(computing)), and service requesters, called [clients](https://en.wikipedia.org/wiki/Client_(computing)).[[1]](https://en.wikipedia.org/wiki/Client%E2%80%93server_model#cite_note-1) Often clients and servers communicate over a [computer network](https://en.wikipedia.org/wiki/Computer_network) on separate hardware, but both client and server may reside in the same system. A server [host](https://en.wikipedia.org/wiki/Host_(network)) runs one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

**2.2Technology and Tools**

* + **Front End:**
    - **Java (HTML, CSS, JavaScript)** 
      * **HTML:**
* **Hypertext Markup Language** (**HTML**) is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) for creating [web pages](https://en.wikipedia.org/wiki/Web_page) and [web applications](https://en.wikipedia.org/wiki/Web_application). With [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) and [JavaScript](https://en.wikipedia.org/wiki/JavaScript), it forms a triad of [cornerstone](https://en.wikipedia.org/wiki/Cornerstone) technologies for the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web).
* [Web browsers](https://en.wikipedia.org/wiki/Web_browser) receive HTML documents from a [web server](https://en.wikipedia.org/wiki/Web_server) or from local storage and [render](https://en.wikipedia.org/wiki/Browser_engine) the documents into multimedia web pages. HTML describes the structure of a web page [semantically](https://en.wikipedia.org/wiki/Semantic_Web) and originally included cues for the appearance of the document.
  + - * **CSS:**
* **Cascading Style Sheets** (**CSS**) is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language) like [HTML](https://en.wikipedia.org/wiki/HTML). CSS is a cornerstone technology of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), alongside HTML and [JavaScript](https://en.wikipedia.org/wiki/JavaScript).
* CSS is designed to enable the separation of presentation and content, including [layout](https://en.wikipedia.org/wiki/Page_layout), [colors](https://en.wikipedia.org/wiki/Color), and [fonts](https://en.wikipedia.org/wiki/Typeface).[[3]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-3) This separation can improve content [accessibility](https://en.wikipedia.org/wiki/Accessibility), provide more flexibility and control in the specification of presentation characteristics, enable multiple [web pages](https://en.wikipedia.org/wiki/Web_page) to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.
  + - * **JavaScript:**
* **JavaScript** often abbreviated as **JS**, is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [programming language](https://en.wikipedia.org/wiki/Programming_language) that conforms to the [ECMAScript](https://en.wikipedia.org/wiki/ECMAScript) specification. It is a programming language that is characterized as [dynamic](https://en.wikipedia.org/wiki/Dynamic_programming_language), [weakly typed](https://en.wikipedia.org/wiki/Weak_typing), [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming) and [multi-paradigm](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language).
* Alongside [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS](https://en.wikipedia.org/wiki/CSS), JavaScript is one of the core technologies of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web). JavaScript enables interactive [web pages](https://en.wikipedia.org/wiki/Web_page) and is an essential part of [web applications](https://en.wikipedia.org/wiki/Web_application). The vast majority of [websites](https://en.wikipedia.org/wiki/Website) use it, and major [web browsers](https://en.wikipedia.org/wiki/Web_browser) have a dedicated [JavaScript engine](https://en.wikipedia.org/wiki/JavaScript_engine) to execute it.
  + **Middleware:**
    - **Java (Java Servlet, JDBC)** 
      * + **Java Servlet:**
* A Java servlet processes or stores a [Java class](https://en.wikipedia.org/wiki/Java_class) in [Java EE](https://en.wikipedia.org/wiki/Java_EE) that conforms to the Java Servlet API,,a standard for implementing Java classes that respond to requests. Servlets could in principle communicate over any [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server_model) protocol, but they are most often used with the [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol).
* Thus "servlet" is often used as shorthand for "HTTP servlet". Thus, a [software developer](https://en.wikipedia.org/wiki/Software_developer) may use a servlet to add [dynamic content](https://en.wikipedia.org/wiki/Dynamic_web_page) to a [web server](https://en.wikipedia.org/wiki/Web_server) using the [Java platform](https://en.wikipedia.org/wiki/Java_platform). The generated content is commonly [HTML](https://en.wikipedia.org/wiki/HTML), but may be other data such as [XML](https://en.wikipedia.org/wiki/XML) and more commonly, JSON. Servlets can maintain [state](https://en.wikipedia.org/wiki/State_(computer_science)) in [session](https://en.wikipedia.org/wiki/Session_(computer_science)) variables across many server transactions by using [HTTP cookies](https://en.wikipedia.org/wiki/HTTP_cookie), or [URL mapping](https://en.wikipedia.org/wiki/URL_mapping).
  + - * + **JDBC:**

**Java Database Connectivity** (**JDBC**) is an [application programming interface](https://en.wikipedia.org/wiki/Application_programming_interface) (API) for the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), which defines how a client may access a [database](https://en.wikipedia.org/wiki/Database). It is a Java-based data access technology used for Java database connectivity. It is part of the [Java Standard Edition](https://en.wikipedia.org/wiki/Java_Standard_Edition) platform, from [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation). It provides methods to query and update data in a database, and is oriented towards [relational databases](https://en.wikipedia.org/wiki/Relational_database). A JDBC-to-[ODBC](https://en.wikipedia.org/wiki/ODBC) bridge enables connections to any ODBC-accessible data source in the [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) host environment.

* + **Backend:** *{can run on any database}*
    - **Oracle/SQL Server**
* **MySQL:**
* **MySQL**  is an [open source](https://en.wikipedia.org/wiki/Open-source_software) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). "[SQL](https://en.wikipedia.org/wiki/SQL)", is abbreviation for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language).
* MySQL is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), and is also available under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) licenses. MySQL was owned and sponsored by the [Swedish](https://en.wikipedia.org/wiki/Sweden) company [MySQL AB](https://en.wikipedia.org/wiki/MySQL_AB), which was bought by Sun Microsystems (now [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation)).

**ANALYSIS**

1. **Analysis**
   1. **Software Requirements**

* Operating System: Linux OS, Windows 7/8/10
* IDE: Eclipse IDE for Java EE Developers (Oxygen)
* Server: MySQL WorkBench Server 6.2, Tomcat 8.5
* RDBMS: MySQL
* Environment: JDK 1.6, 1.7, 1.8 for Java 6, 7, 8 configured on the workstation

**3.2 Hardware Requirements**

* Processor: 1.7GHz Intel Core2Duo or above
* RAM: 4 GB
* Hard Disk: 100 GB-1 TB
* Network Adaptor

**CHAPTER 4**

**DESIGN**

1. **Design**
   1. **Diagrams**

ADMIN FLOW CHART

Login

Trainer List

Show Trainer List available for particular program

Allocate Trainers

Admin will be able to allocate

Trainer based on availability

Calendar List

1-month Calendar details with which trainer is allocated, with which program and for what duration

Confirmation Request

Admin will be able to send the confirmation request to trainer regarding skill program allocation.

Logout

TRAINER FLOW CHART

Signup/Register Page

Registration Page

Procure Trainers information with skill set

Trainer Availability

On which date Trainer is available for skill program

Confirmation Notification

Trainer will get a confirmation notification from admin

Logout

SME FLOW CHART

Signup/Register Page

Registration Page

Procure SME information with skill set

SME Request

SME will be able to request for Trainer approval from support team

Confirmation Notification

SME will get a confirmation notification from admin

Logout

**4.2 Tables**

ADMIN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Field Type | Data Type | Mandatory | Possible Values |
| UserName | Text (20) | Varchar | Yes |  |
| Password | Text (20) | Varchar | Yes |  |

*Table 1*

TRAINER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Field Type | Data Type | Mandatory | Possible Values |
| First Name | Text (50) | Varchar | Yes |  |
| Last Name | Text (50) | Varchar | Yes |  |
| Age | Numeric (2) | Integer | Yes |  |
| Gender | Drop Down | NA | Yes | Male/Female |
| Contact Number | Numeric (13) | Integer | Yes |  |
| UserId | Text (20) | Varchar | Yes |  |
| Password | Text (20) | Varchar | Yes |  |

*Table 2*

SME

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Field Type | Data Type | Mandatory | Possible Values |
| First Name | Text (50) | Varchar | Yes |  |
| Last Name | Text (50) | Varchar | Yes |  |
| Age | Numeric (2) | Integer | Yes |  |
| Gender | Drop Down | NA | Yes | Male/Female |
| Contact Number | Numeric (13) | Integer | Yes |  |
| UserId | Text (20) | Varchar | Yes |  |
| Password | Text (20) | Varchar | Yes |  |

*Table 3*

**CHAPTER 5**

**CONCLUSION**

1. **Conclusion**

The trainer pool management system guided us through the proper architecture of a management system, making us learn the structured levels of development. Being new to the software, a little problem was faced while writing the complete code from scratch however learning the new technologies in order to build the project has leveled up the making process of management system. As a limitation of project, there could be a few more functionalities added to enhance the system and similarly a better architecture can be used to make it smoother. But whatever is made, it has surely cleared our basics and taught us a lot. Concluding to it, we look forward to enhance the features of the system as well as expanding the possible opportunities that come its way.

**REFERENCES**

* w3schools.com
* docs.oracle.com
* stackoverflow.com
* css-tricks.com
* [www.techonthenet.com](http://www.techonthenet.com)
* [www.tutorialspoint.com](http://www.tutorialspoint.com)
* beginnersbook.com