



Human Resource Management System For St. Nicholas' Education Services Ltd

R.A.Y.S. RANASINGHE

BIT Registration No: R111734

Index No: 1117343

Client: Rev. Fr. B. Anselm Shiran

Supervisor: P.Y.C. Paranamanna


December 2017



**This dissertation is submitted in partial fulfillment of the requirement of
the Degree of Bachelor of Information Technology (external) of the
University Of Colombo School Of Computing**

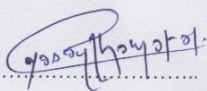
DECLARATION

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a Degree or Diploma in any University and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organization.

Signature of Candidate: .....

Date: 28.10.2017.....

Name of Candidate: R.A.Y.S. Ranasinghe

Signature of Supervisor: .....

Date: 28.10.2017.....

Name of the Supervisor: P.Y.C. Paranagama

ABSTRACT

St. Nicholas International College is an international college which prepares students only for Cambridge examinations. It was established in June 2003. Now it belongs to the Archdiocese of Colombo. It is located at Nelumpitiya in Negombo. Currently the school's human resource management processes are conducted manually which lacks information accuracy, security and decision management.

The proposed system could minimize the disadvantages of manual procedures and facilitate resource management in a very effective and efficient manner. Existing system lacks the functionalities such as Staff Detail Management, Subject Allocation, Leave Management, Performance Summary, Seminar and Training Management and etc. Proposed solution will provide access to various kinds of management reports and facilitate the search process.

The system is developed in accord with the Object Oriented techniques. Unified Modeling Language was used in the analysis and design phases. Java language was used to implement the proposed system. Development tools such as NetBeans IDE 8.2 and Gluon JavaFX Scene Builder were used to aid the development process. Hibernate ORM, JavaFX and JFoenix library frameworks were used to ease the development of system GUIs. MySQL Workbench, MYSQL Server 5.5 were used to develop the database for the system and other tools such as Adobe Photoshop CS6 and Visual Paradigm were used to aid the design process.

When studying problem domain and software engineering theories AUP (Agile Unified Process) was identified as the most suitable development methodology for the system.

The system will fulfill all the client requirements and provide effective and efficient answers to the existing problems. Proposed system will facilitate the school's regular activities and provide better service to their staff members while maintaining information security.

ACKNOWLEDGMENT

It is my privilege to express my gratitude to all the people who have helped me directly or indirectly to complete the project successfully.

First of all I would like to express my profound gratitude to BIT coordinator of the UCSC and to the Project Examination Board for giving me this valuable opportunity to follow a world recognized degree program providing us good guidance continuously.

I humbly give my great appreciation to Rev. Fr. Anselm Shiran the Principal, former Principal Rev. Fr. Ruwan Deshapriya and Mr. Lakmina Wijethunga the Vice Principal of St. Nicholas International College for being very kind and supportive from the beginning to the end of this work. Special thanks must go to my project supervisor Mr. P.Y.C. Paranamanna.

Furthermore, I would also like to acknowledge with much appreciation the crucial role of the staff of St. Nicholas International College, who gave all the necessary information and the support from the requirement analysis stage. And also extend my heartfelt thanks to my family members for their support without which my success in the project have been a distant reality. Finally, my thanks go to all my friends & everyone who gave their valuable support in numerous ways & encouraged me to successfully complete my project.

TABLE OF CONTENTS

ABSTRACT.....	ii
ACKNOWLEDGMENT	iii
TABLE OF CONTENTS.....	iv
LIST OF FIGURES	ix
LIST OF TABLES.....	xiii
LIST OF ACRONYMS	xiv
CHAPTER 01 – INTRODUCTION	1
1.1 Brief Introduction of St. Nicholas' International College.....	1
1.2 Motivation for Project	1
1.3 Objectives of the Project	2
1.4 Scope of the Project	3
1.5 Structure of the Dissertation.....	3
CHAPTER 2 – ANALYSIS	5
2.1 Requirement Gathering	5
2.2 Current System and the Process	6
2.2.1 Recoding teachers Information.....	6
2.2.2 Subject Allocation	6
2.2.3 Leave Management.....	6
2.2.4 Seminar and Training Arrangements.....	6
2.2.5 Teacher's performance Analysis	7
2.3 Use Case Diagram of Existing System	7

2.4 Functional Requirements of the System	7
2.5 Nonfunctional Requirements of the System	8
2.6 Study of a Similar System.....	8
2.6.1 Negombo International School Teacher Management System.....	8
2.6.2 Behance HR system.....	9
2.6.3 Comparison between above similar systems	11
2.7 Selected Methodology	12
CHAPTER 3 – DESIGN.....	14
3.1 Alternate solutions evaluation.....	14
3.2 Object Oriented Designing.....	14
3.3 Use Case Diagram for the proposed system	15
.....	16
3.4 Use Case Narratives for the proposed system.....	16
3.4.1 Use Case narratives for the Apply Leave	16
3.4.2 Use Case narratives for the seminar allocation	17
.....	17
3.5 Activity Diagram.....	17
.....	18
3.6 Class Diagram	18
3.7 Sequence Diagram	20
3.8 Database Design.....	22
3.9 Entity Relationship Diagram.....	22
3.10 User Interface Design	24
CHAPTER 4 – IMPLEMENTATION	25

4.1 Implemented Environment.....	25
4.1.1 Hardware Requirements	25
4.1.2 Software Requirements.....	25
4.2 Development Tools	26
4.2.1 NetBeans IDE 8.2	26
4.2.2 Java Language	26
4.2.3 MySQL Server.....	27
4.2.4 JavaFX Scene Builder	27
4.2.5 Visual Paradigm	28
4.2.6 Hibernate	28
4.2.7 Jasper Report	30
4.3 Implementation	30
4.4 Data Layer Implementation.....	31
4.4.1 Hibernate Configuration	32
<i>Code 4.1 hibernate.cfg.xml</i>	32
4.4.2 Create Java Entity Classes.....	32
4.4.3 Hibernate Session	35
4.5 Business Logic Layer Implementation.....	36
4.5.1 DAO (Data Access Object)	36
4.5.2 Event Handlers	39
4.6 Interface Layer Implementation	39
CHAPTER 5 – EVALUATION	45
5.1 Testing Methods.....	45
5.1.1 Unit Testing	45

5.1.2 Integration Testing.....	46
5.1.3 System Testing	46
5.2 Test Cases.....	46
5.2.1 Test cases and test results for Login Module	47
5.2.2 Test cases and test results for Subject Allocation Module	47
5.2.3 Test cases and test results for Development Module	48
5.2.4 Test cases and test results for Performance Module.....	48
5.3 User Evaluation	49
5.3.1 Results respect to the Objectives	49
5.3.2 Client Feedback	50
CHAPTER 6 - CONCLUSION	53
6.1 Lessons Learnt	53
6.2 Future Improvement	54
REFERENCE.....	55
APPENDIX.....	57
APPENDIX – A	57
System Documentation.....	57
A.1 SOFTWARE INSTALLATION AND CONFIGERATION.....	57
APPENDIX – B	70
B.1 Design Documentation	70
B.1.1Activity Diagram	70
B.1.2 Sequence diagram	71
B.1.3 Use Case diagram	73
B.2 Database Design	75

APPENDIX – C	80
USER DOCUMENTATION	80
APPENDIX – D.....	84
MANAGEMENT REPORT.....	84
APPENDIX-E.....	86
TEST RESULTS.....	86
APPENDIX F	89
CODE LISTING	89
APPENDIX G.....	91
CLIENT CERTIFICATE.....	91
GLOSSARY	92
INDEX.....	93

LIST OF FIGURES

Figure 2.3 : Use Case Diagram for existing system	7
Figure 2.6 : Behance -Add new employee.....	9
Figure 2.6 : Behance - Employee Details	10
Figure 2.7 : Rational Unified Process Model	12
Figure 3.3 : Use Case diagram of the proposed system.....	16
Figure 3.5 : Activity Diagram for Logging to the system	18
Figure 3.6 : Class diagram of the system.....	19
Figure 3.7 : Sequence diagram for logging.....	20
Figure 3.8 : Entity Relationship diagram for database design	23
Figure 3.10 : Subject Allocation User Interface	23
Figure 4.2 Architecture of Hibernate	29
Figure 4.3. MVC Architecture	31
Figure 4.3 Classes Organizing in the System	31
Figure 5. 1 Sample User Evaluation Form.	51
Figure A. 1: Installation progress of JVM (Step1)	57
Figure A. 2:Installation progress of JVM (Step 2)	58
Figure A. 3: Installation progress of JVM (Step 3)	58
Figure A. 4:Installation progress of JVM (Step4)	58
Figure A. 5: Installation progress of JVM (step 6)	59
Figure A. 6: Installation progress of JVM (step 5)	59
Figure A. 7:Installation Progress MYSQL server(Step 1).....	60
Figure A. 8:Installation Progress MYSQL server (Step 2)	60
Figure A. 9: Installation Progress MYSQL server (Step 3).....	60
Figure A. 10:Installation Progress MYSQL server (Step 4).....	61

Figure A. 11: Installation Progress MYSQL server (Step 5).....	61
Figure A. 12: Installation Progress MYSQL server (Step 6).....	62
Figure A. 13: Installation Progress MYSQL Server (Step 7)	62
Figure A. 14 Installation Progress MYSQL Server (Step 8)	62
Figure A. 15: Installation Progress MYSQL Server (Step 9).....	63
Figure A. 16: Installation Progress MYSQL server (Step 1).....	63
Figure A. 17: Installation Progress MYSQL server (Step -2).....	64
Figure A. 18: Installation Progress MYSQL server (Step 3).....	64
Figure A. 19: Installation Progress MUQL server (Step 4).....	64
Figure A. 20: Installation Progress MYSQL server (Step 5).....	65
Figure A. 21: Installation Progress MYSQL server (Step 6).....	65
Figure A. 22: Installation Progress MYSQL Server (Step 7)	66
Figure A. 23: Installation Progress MYSQL Server (Step 8).....	66
Figure A. 24: Installation Progress MYSQL Server (Step 9).....	66
Figure A. 25:Installation Progress MYSQL Server (Step 10).....	67
Figure A. 26: Installation Progress MYSQL Query Browser (Step 1)	67
Figure A. 27: Installation Progress MYSQL Query Browser (Step 2)	67
Figure A. 28: Installation Progress MYSQL Query Browser (Step 3)	68
Figure A. 29: Installation Progress MYSQL Query Browser (Step 4)	68
Figure A. 30: Open MYSQL Administrator & Insert server host, port & the user name as given above & enter the given password (Step 5).....	68
Figure A. 31: To restore the database create a new database the select restore database & last set database to system database (Step6).....	69
Figure B. 1: Activity Diagram for Insert New User	70
Figure B. 2 : Activity Diagram for Apply Leave.....	71
Figure B. 3: Sequence Diagram for Report generation of the system	72

Figure B. 4 Use case diagram for Principal Requirements	73
Figure B. 5 Use case diagram for Human Resource Manager Requirement	73
Figure B. 6 Teacher Table	75
Figure B. 7 Civil Status Table	76
Figure B. 8 Designation Table	76
Figure B. 9 Gender Table.....	76
Figure B. 10 Qualification Table	76
Figure B. 11 Qualification Type Table	76
Figure B. 12 Salary Table	77
Figure B. 13 Teacher Salary Table	77
Figure B. 14 Leaves Table	77
Figure B. 15 Leave Type Table	78
Figure B. 16 Leave Statues Table	78
Figure B. 17 Leave Category Table	78
Figure B. 18 Section Table	78
Figure B. 19 Section Grade Table	78
Figure B. 20 Subject Table	79
Figure B. 21 User Table.....	79
Figure B. 22 User Privilege Table	79
Figure B. 23 Module Table.....	79
Figure B. 24 Performance Table	79
Figure C. 1 Login Window	80
Figure C. 2 Create Employee Window	81
Figure C. 3 Performance Window	82
Figure C. 4 Development Window	83

Figure D. 1 Teachers performance according the Year	84
Figure D. 2 Semiar Details by Teacher.....	85
Figure G. 1 Client Certificate	91

LIST OF TABLES

Table 3.4 Use case narrative for leave request	16
Table 3.4 Use case narrative for seminar allocation	17
Table 5.2 Test cases and Test results for Login Module	47
Table 5.2 Test Cases and Test results for subject Allocation module	47
Table 5.2 Test Cases and Test Results for Development module	48
Table B. 1 Add User	74
Table B. 2 Approval of leave	75
Table E. 1 Test Results for Create Employee Module	86

LIST OF ACRONYMS

ER	-	Entity Relationship
HR	-	Human Resource
OOD	-	Object Oriented Designing
RUP	-	Rational Unified Process
SRS	-	Software Requirement Specification
UML	-	Unified Modeling Language

CHAPTER 01 – INTRODUCTION

This chapter provides an introduction to the Human Resources Management System for St. Nicholas' Education Services Limited with detailed description on what it's about, motivation to carry out the selected project, need for the project objectives and scope.

1.1 Brief Introduction of St. Nicholas' International College

St. Nicholas' International College is the first catholic international college in Sri Lanka. This school prepares students only for Cambridge examinations.

The St. Nicholas' International College has the main branch in Negombo and second branch in Colombo. There are 80 teachers in Negombo School. Teachers can be categorized in to a number of different types. They are permanent, contract, probation, visiting etc., Most of them are degree holders. The subject allocation is done according to the teacher's education qualification and experience.

There are 21 leaves for permanent teachers. These leaves include casual and annual leaves. If a teacher gets a baby, she will have 84 leaves. The leave management of teachers is done by Human Resource Manager.

Teachers are encouraged to attend seminars and special training programs which are organized by the Cambridge University, British Council and School for the benefit of teachers.

1.2 Motivation for Project

Currently, traditional methods such as file based systems are being used by school to handle their human resources activities. There is no computerized system to access any teacher's information. Because of that if they want to access any teacher's information such as the appointment date, academic qualifications, working experience they need to search information manually.

Updating qualifications of teachers who have completed further studies is not accurately done by the existing system. Subject allocation and responsibilities

entrusted are sometimes incompatible with the qualifications and experience of the teachers.

There are different types of employment in the staff. There is no proper way to analyze a teacher's performance. Because of that many complications arise when it comes to salary calculations, increments, bonus etc., of a teacher.

In applying leave, time wastage takes place due to the need for manually referring to leave registers and in the process delays occur in getting approval from authorized personnel concerned.

When sending teacher for seminars, some teachers complain that certain teachers take part in seminars regularly while some are constantly neglected. This is due to lack of updating of the records.

The proposed system will help to solve the problems involved with the manual system and enhance the performance of the school administration.

1.3 Objectives of the Project

The main objective of this project is managing human resources activities such as Leaves and Performance management of the teachers. Also system increases the decision making ability of management by providing various kinds of reports. Information security will be assured by this newly proposed system.

- Improve the staff information management.
The system will facilitate entering, updating and searching details of teachers efficiently.
- Teacher's confidential information will be secure.
Tasks and information are easily available for authorize users, but also the system will prevent digging of information by unauthorized personnel.
- Reduce the time and human effort in leave management.
Absence of leave can be applied by the teachers for consideration by the management.

- Sending teachers for seminars will be done in a fair manner without overlooking any teacher.
- Improve the efficiency of teacher's subject allocation.
The system will facilitate allocation for teacher to several subject streams according their section.
- Increasing the accuracy of data.
This system will validate the correctness of entering data and only privileged users can change the data.
- Provide supportive information to the management.
System will provide daily, weekly and monthly reports according to the management's needs.

1.4 Scope of the Project

The system will provide with

- Manage the employees personal information, academic qualification and working experience.
- System user levels and privilege management.
- Manage teacher's subject allocation.
- Manage leaves of employees.
- Manage teacher seminars.
- Manage teacher performance analysis.
- Report generation

1.5 Structure of the Dissertation

This dissertation presents overall knowledge about Human Resources Management system for St. Nicholas' Education Services Limited and how activities carried out during each phase of the project. There are seven main chapters with the Introduction Chapter are included in this report. They are briefly described as follows.

Chapter 2: Analysis

Gathering functional and non-functional requirements and analyzing them and how activities process in the current manual system are identified in this chapter. UML diagrams such as use case diagrams are drawn and provided to study the current system and clearly identify the requirements of the user.

Chapter 3: Design

This chapter will describe how the system is designed to fulfill the user's requirements and alternative solutions. It will discuss about requirements of user, architecture of the proposed system, how to design the database, and how to provide friendly interface to users. Additionally, it will provide use case diagrams, class diagrams, Entity Relationship diagrams and other related diagrams for use.

Chapter 4: Implementation

This chapter will explain how the project was implemented according to the system design. Development environment, software tools, and technologies used as well as major code segments were discussed.

Chapter 5: Evaluation

Testing the implemented system by different kind of testing methods will be discussed in this chapter. How the system was tested using sample data and the results of the tests together with testing procedure and test cases.

Chapter 6: Conclusion

The lessons that learnt during the project and how the system could be further improved will be discussed in this chapter.

CHAPTER 2 – ANALYSIS

Requirement analysis is the first and most critical stage in software development process. All the other phases depend on the information gathered in this phase. Requirement analysis is the process of identifying users and their duties, understand the problem domain and user requirement.

2.1 Requirement Gathering

Combination of several fact finding techniques were used to capture the requirement of the Human Resource Management System. Following are the methods used for requirement gathering.

- Reading existing school document

One must have a clear understanding of school process to build up the system. This is one of the best techniques to get a good and clear idea of what the process is. Below is the list of documents that I consider.

- Organization Chart
- School objective & strategic plan
- Sample of manual log books and files
- Existing leave register
- Interviews and Discussions

The employees of the school who are involved in the current human resources activities were interviewed. Thereby user requirements, difficulties in the current system and new ideas were discussed and identified. Extended discussions were carried out with some user groups for clarifications and correctness of gathered information.

- Prototyping

Built a simple working system and showed it to users and asked what are their requirements, changes of system. Prototyping is good when the user don't have

clear idea about the system. Prototype helps future users and other stakeholders to get an idea of what the system will look like. [1]

2.2 Current System and the Process

The activities of the St. Nicholas' International College are currently handled manually. They record their all details and information in books and files manually. There are five main processes that can be identified in the current system.

2.2.1 Recoding teachers Information

Currently they handle only some of the information of teachers. Academic qualification and working experience which teachers provide are collected in separate files. The contact numbers, email address and appointment dates are written down.

2.2.2 Subject Allocation

After recruitment process of the teachers, allocation of subjects and syllabuses is the main activity of the current system. There are two sections in the school middle school and upper school. According to the section teachers are allocated. This is done by the Head Teacher with the assistance of Principal or Vice Principal.

2.2.3 Leave Management

There are 21 days leave for permanent teachers. This leave includes casual and annual leave. If a teacher has a baby, she will have 84 maternity leaves. All the details regarding leave are written in the leave register. The leave management of teacher is done by Human Resource Administrator.

2.2.4 Seminar and Training Arrangements

The seminars/special training programs which are organized by the Cambridge University, British Council or by the school is arranged by the Human Resource Administrator for the benefit of teachers.

2.2.5 Teacher's performance Analysis

There are different types of employment in the staff. There are many ways to analyze teacher's performance such as monitoring their leave, correction of books, completion of syllabus etc., All these monitoring is done manually using separate file for each teacher by the Principal.

2.3 Use Case Diagram of Existing System

Figure 2.3 given below shows the use case diagram for the existing system

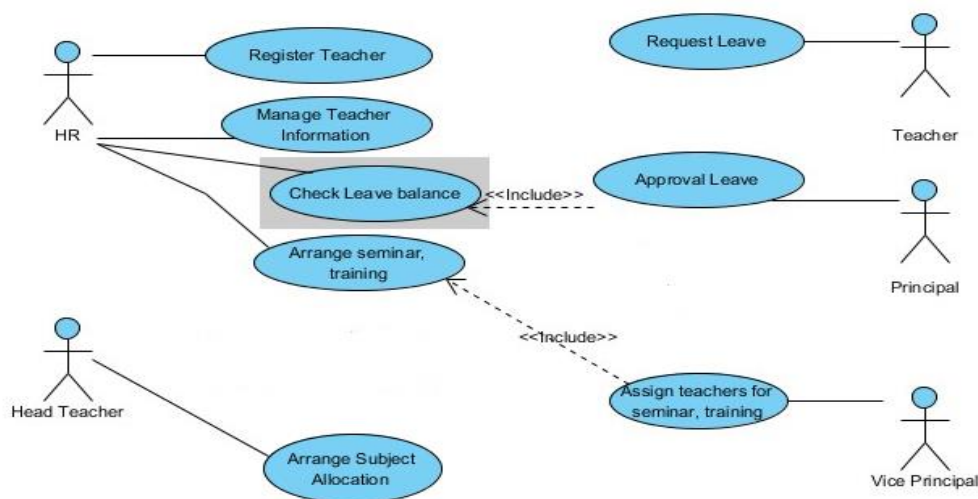


Figure 2.3 : Use Case Diagram for existing system

2.4 Functional Requirements of the System

In software engineering, a functional requirement defines a function of a software system or its components. A function is described as a set of inputs, the behavior, and outputs (see also software). Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. [2]

Following are the functional requirements of the Human Resources Management System.

- Manage the employees personal information, academic qualifications and working experience
- System user levels and privileged management
- Manage the employees leave
- Manage teacher's seminar / training
- Manage subject allocation for teachers
- Manage teacher's performance
- Generate various types of reports

2.5 Nonfunctional Requirements of the System

Nonfunctional requirement describes the features that system should have. It is very difficult to manage nonfunctional requirement than functional requirements because each user have different perspectives. Nonfunctional requirements which are to be offered with the proposed system are listed below.

- User friendliness
- Interoperability with other software
- Accuracy
- Maintainability
- Security
- Usability
- Simplicity

2.6 Study of a Similar System

To understand the procedure of the Human Resource Management system, study and research some similar systems. It was help for in understanding the design of this project. There were two systems.

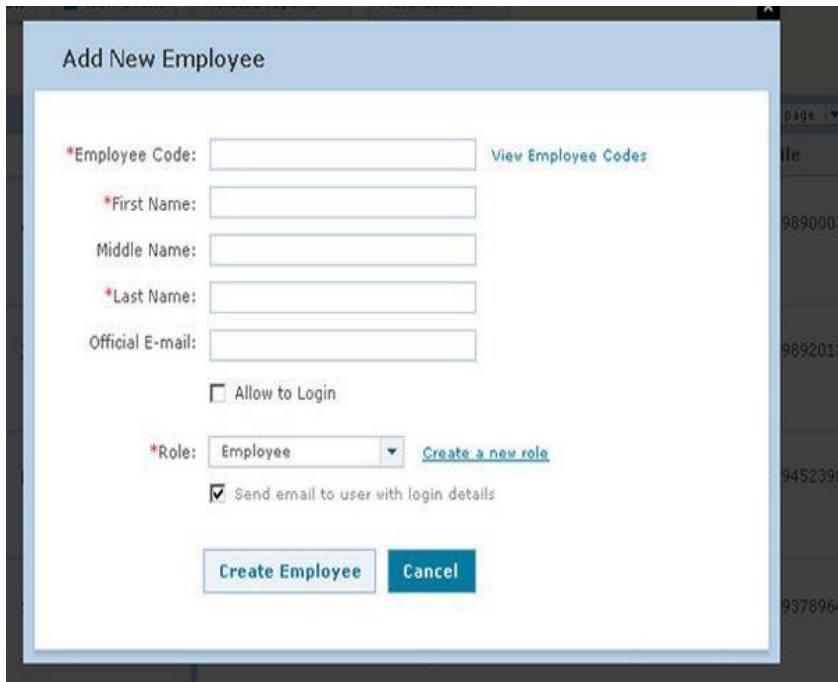
2.6.1 Negombo International School Teacher Management System

It is also used as a standalone system to handle their day to day human resources activities. They also manage their Staff Member details by using that system. Study of this system was very helpful to get an idea about this Human Resources Management system.

2.6.2 Behance HR system

Behance system is a Human Resources Management system build in India. Some of the user interfaces are listed here. [3]

Figure 2.6 given below shows Behance system, how to add new employee



The screenshot shows a web form titled "Add New Employee". The form contains the following elements:

- *Employee Code:** A text input field with a "View Employee Codes" link to its right.
- *First Name:** A text input field.
- Middle Name:** A text input field.
- *Last Name:** A text input field.
- Official E-mail:** A text input field.
- Allow to Login
- *Role:** A dropdown menu currently showing "Employee", with a "Create a new role" link to its right.
- Send email to user with login details
- Two buttons at the bottom: "Create Employee" (light blue) and "Cancel" (dark blue).

Figure 2.6 : Behance -Add new employee

Figure 2.6 given below shows Beahnce system, how to add employee details

The screenshot displays the 'Employee Details' form for Shubroto Bhattacharya. The interface includes a navigation bar with tabs for Employee, Leave, Salary, Report, and Travel. Below this, there are sub-tabs for Personal, Job, Salary, Other, and Document. The form is divided into several sections:

- Personal Information:** Includes a profile picture of Shubroto Bhattacharya, a 'Change Photo' link, and a 'Remove Photo' link. Below the photo is a section for uploading a new photo with a 'Browse' button and 'Save'/'Cancel' options. Employee details listed include: Employee Code: 93639, Role: Employee, Joining Date: 09-May-2008, Designation: Manager, and Department: Finance.
- Contact Details:** Contains fields for Present Address 1, 2, and 3; Permanent Address 1, 2, and 3; City; Zipcode; Office Phone No. (with Area Code and Phone# sub-fields); Office Extension; Office Mobile No.; Personal Phone No. (with Area Code and Phone# sub-fields); Personal Mobile No.; and Personal Email.
- Tax Related:** Includes fields for PAN No., PF No., and Social Security No.
- Bank Details:** Includes fields for Bank, Branch, and Account No., along with an Account Type selection (Current, Savings, Other) where 'Savings' is selected.

Figure 2.6 : Beahnce - Employee Details

2.6.3 Comparison between above similar systems

Software Features	Behance Software	Teacher Management Software
Manage Staff Details	✓	✓
Manage allocation of subject and classes	✗	✓
Manage Staff Leaves	✓	✗
Manage Staff training & seminar	✗	✓
Manage Staff performance	✓	✓
Report Generating	✓	✓
Notification	✓	✓
Manage Backups	✗	✗
User Friendliness	✓	✓

Table 2.6.3. 1: Comparison between similar systems

2.7 Selected Methodology

When analyzing the problem domain, opportunities and software engineering theories RUP is the most suitable methodology for systems, where requirements rapidly changed.

The characteristics of unified process model are use-case driven, architecture centric, risk focused and iterative and incremental development. This process divides the development process into four phases that are inception, elaboration, construction and transition. [4]

Figure 2.76 given below shows Rational Unified Process Model

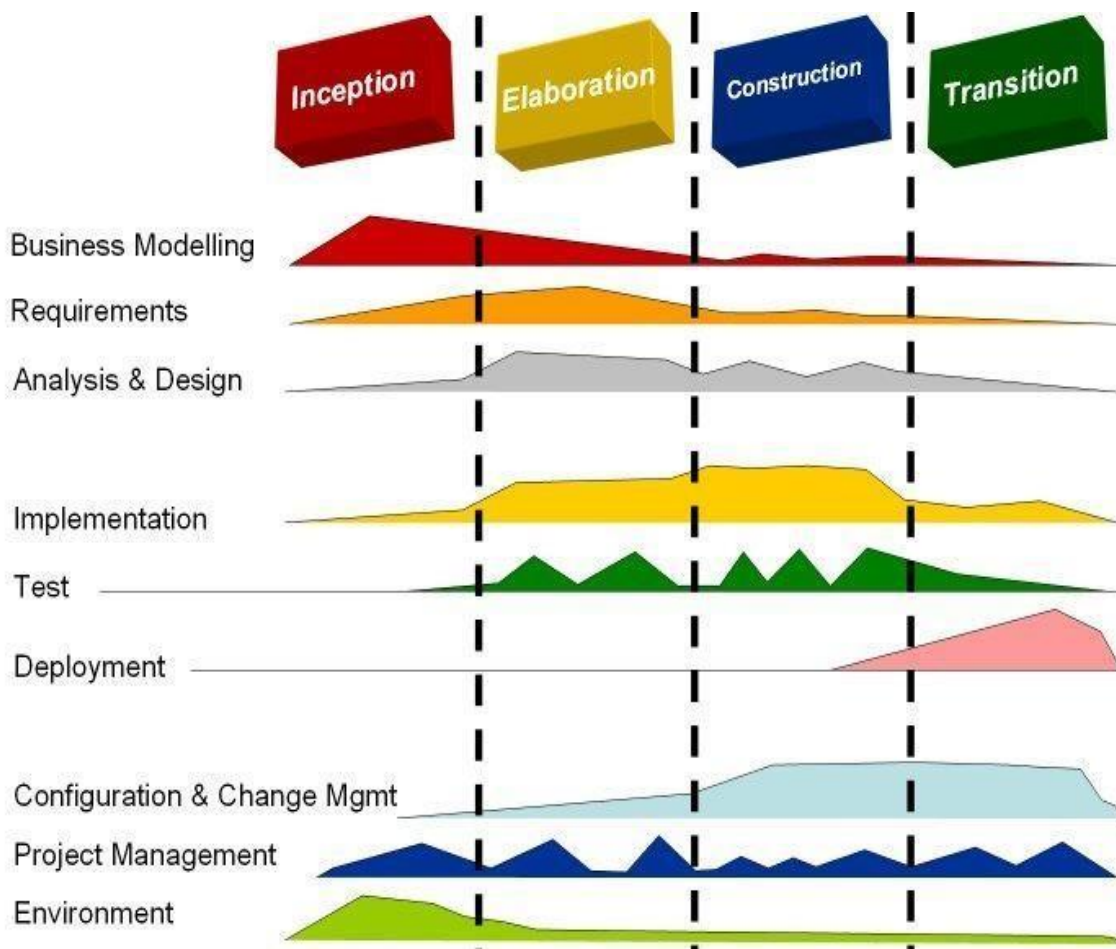


Figure 2.7: Rational Unified Process Model

Description for each phase

Inception

A discover phase, where an initial problem statement and functional requirements are created. The business case is defined which includes the business context of the project.

Elaboration

Architecture of the project gets its basic form and construction cycles are planned.

Construction

The development of components and other features of the system being designed

Transition

The software is deployed to the users. Final adjustments or updates are made based on feedback from end users.

CHAPTER 3 – DESIGN

Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.

For assessing user requirements, an SRS (Software Requirement Specification) document is created whereas for coding and implementation, there is a need of more specific and detailed requirements in software terms. The output of this process can directly be used into implementation in programming languages. [5]

This chapter will give a clear idea about alternate solutions and their evaluation, description and justification of selected solution, relevant design diagrams and overall idea about user interface design

3.1 Alternate solutions evaluation

St. Nicholas' International College expects a solution to manage the internal human resources activities especially about teachers of the school. There are no huge numbers of users for the system and they are not requested to work remotely. Therefore, standalone system is more suitable for this system. Furthermore, this is their first computerized system and have no experienced staff to handle complex system like this web based program. Because this is their first computerized system they do not need to spend more for this system. The following reasons are also to be considered.

- The client particularly requested for a standalone system
- System would be platform independent
- Maintenance is easy
- Can perform much faster than web based system
- No need to pay additional cost for web hosting

3.2 Object Oriented Designing

Object Oriented Designing (OOD) is a method that uses objects to develop a system. Each object interacts with each other and they have their own states and operations. Unified Modeling Language is one of the most popular methods used in order to develop systems with OOD concept. There are some object models defined by UML such as Use Case

Diagrams, Class Diagrams, and Activity Diagrams that aid system designing process. The following object models were used for the designing process of the system.

- Use Case Diagram
- Class Diagram
- Activity Diagram
- Sequence Diagram
- Entity Relationship Diagram

3.3 Use Case Diagram for the proposed system

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and would often be accompanied by other types of diagrams as well. [6]

In this system there are five critical users as below.

- Human Resources Manager (HR)
- Administrator
- Principal
- Vice Principal
- Head Teacher
- Teacher

Figure 3.3.1 given below shows the use case diagram for the proposed system

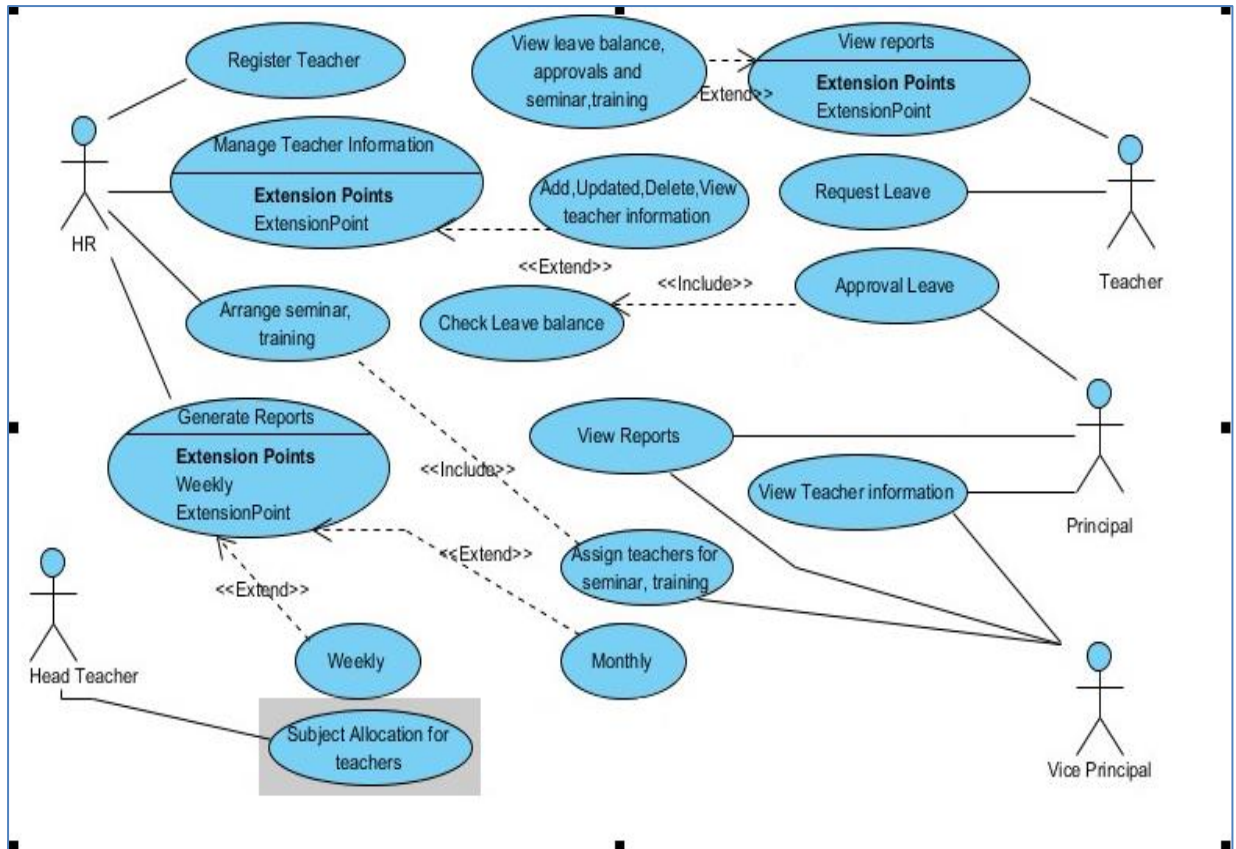


Figure 3.3 Use Case diagram of the proposed system

3.4 Use Case Narratives for the proposed system

3.4.1 Use Case narratives for the Apply Leave

Use Case	Apply Leave
Actors	User
Description	Apply leave
Precondition	<ul style="list-style-type: none"> • User log in to the system • Leave category, type, duration and reason should be added
Flow of event	<ul style="list-style-type: none"> • Select name and requested date • Select leave category and type • Enter reason for leave • Click Save
Post Condition	Request approval for leave

Table 3.4. 1: Use case narrative for leave apply

3.4.2 Use Case narratives for the seminar allocation

Use Case	Assign seminar, training
Actors	Vice principal, Human resource manager
Description	Allocate teachers for seminar and training program
Precondition	<ul style="list-style-type: none"> • Log in to the system • Select teacher • Select section
Flow of event	<ul style="list-style-type: none"> • Select subject • Select seminar type • Select conducted by • Enter description • Select the date/ time and venue
Post Condition	Reserve teacher for seminar / training program

Table 3.4. 2: Use case narrative for seminar allocation

3.5 Activity Diagram

Activity diagrams are presenting detailed operations of the use cases which are identified in use case diagram. The control flow of a use case in the system can be identified which turns user requirements to system requirements. Activity is a particular operation of the system.

Figure 3.5 given below shows Activity Diagram for Logging to the System

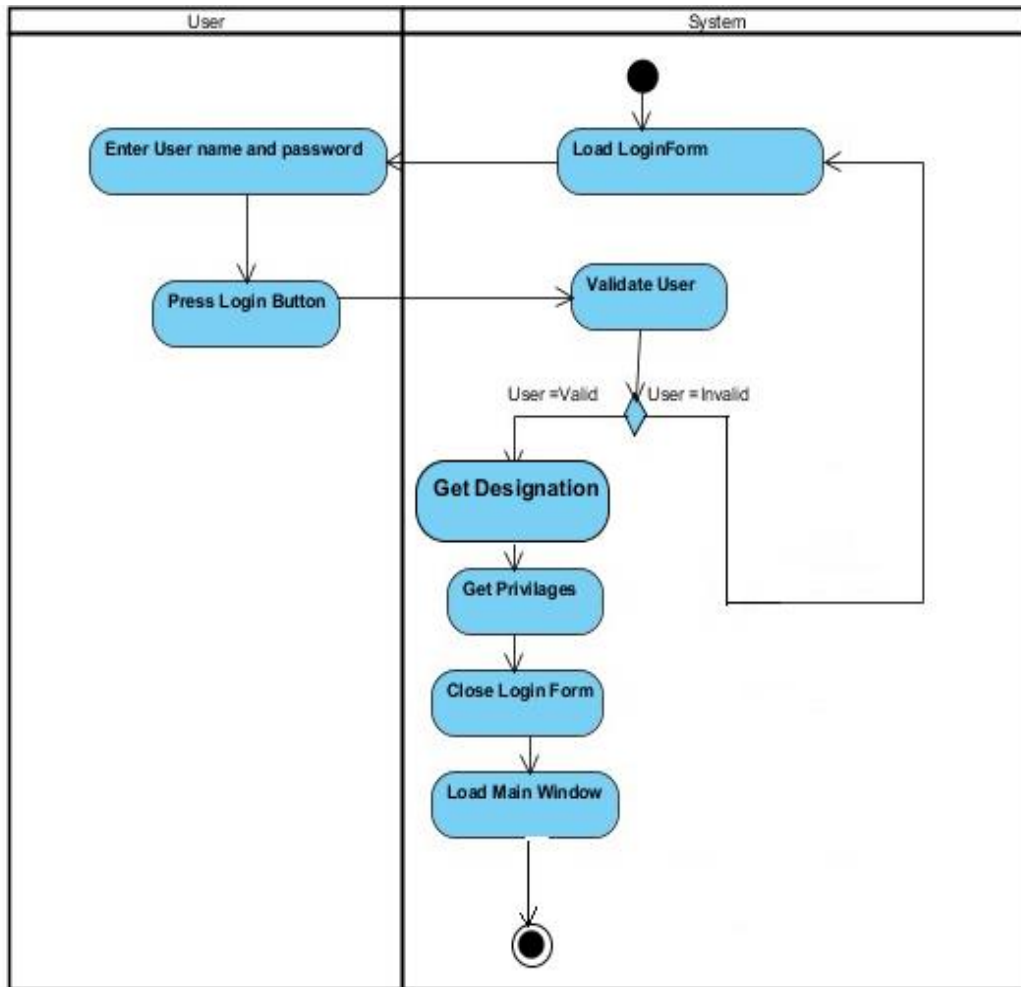


Figure 3.5 Activity Diagram for Logging to the system

3.6 Class Diagram

Class Diagram provides an overview of the target system by describing the objects and classes inside the system and the relationships between them. It provides a wide variety of usages; from modeling the domain-specific data structure to detailed design of the target system. With the share model facilities, you can reuse your class model in the interaction diagram for modeling the detailed design of the dynamic behavior. It shows how the different entities relate to each other with the static structure of the system. [7]

Figure 3.6 given below shows Class Diagram for the system.



Figure 3.6 Class diagram of the system

3.7 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construction of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. [8]

Figure 3.7 given below shows Sequence Diagram for Logging

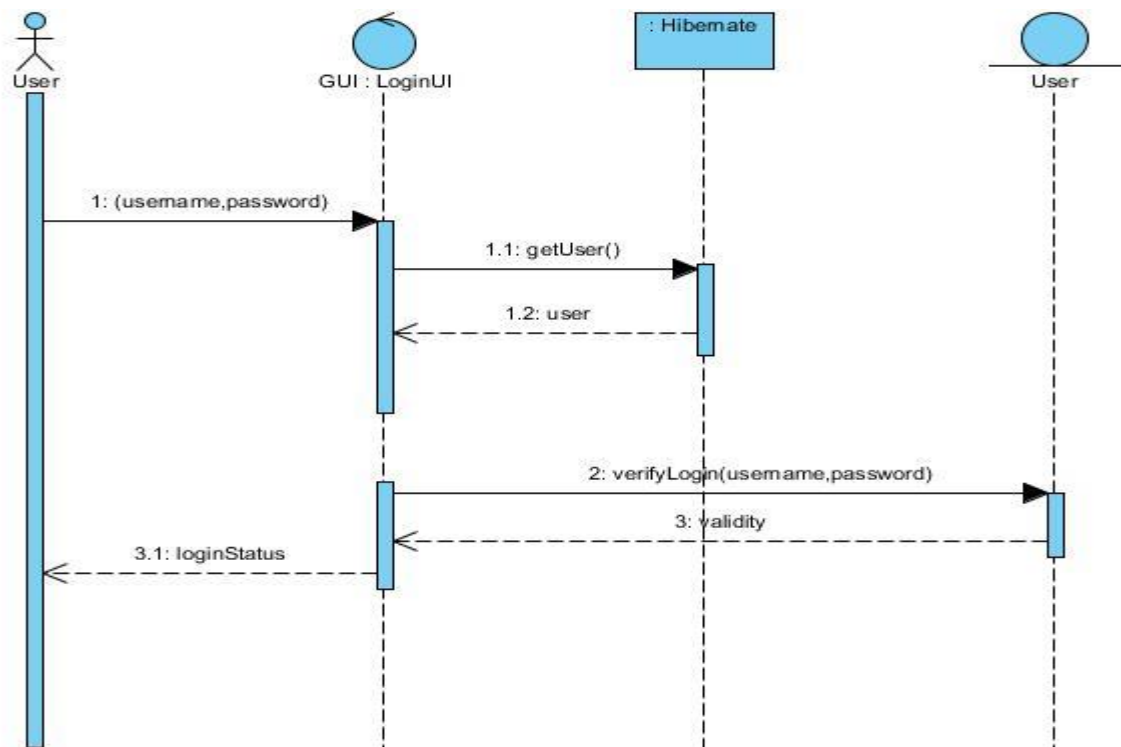


Figure 3.7 Sequence diagram for logging

Figure 3.8 given below shows Sequence Diagram for Check User duplicate

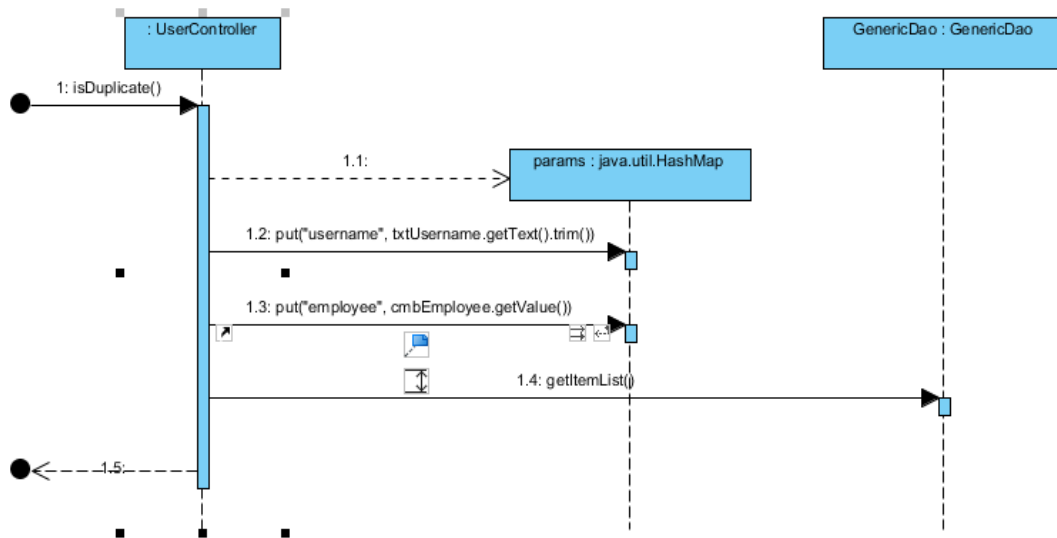


Figure 3.8 Sequence Diagram for Check User duplicate

Figure 3.9 given below shows Select and Set image for the Employee User Interface

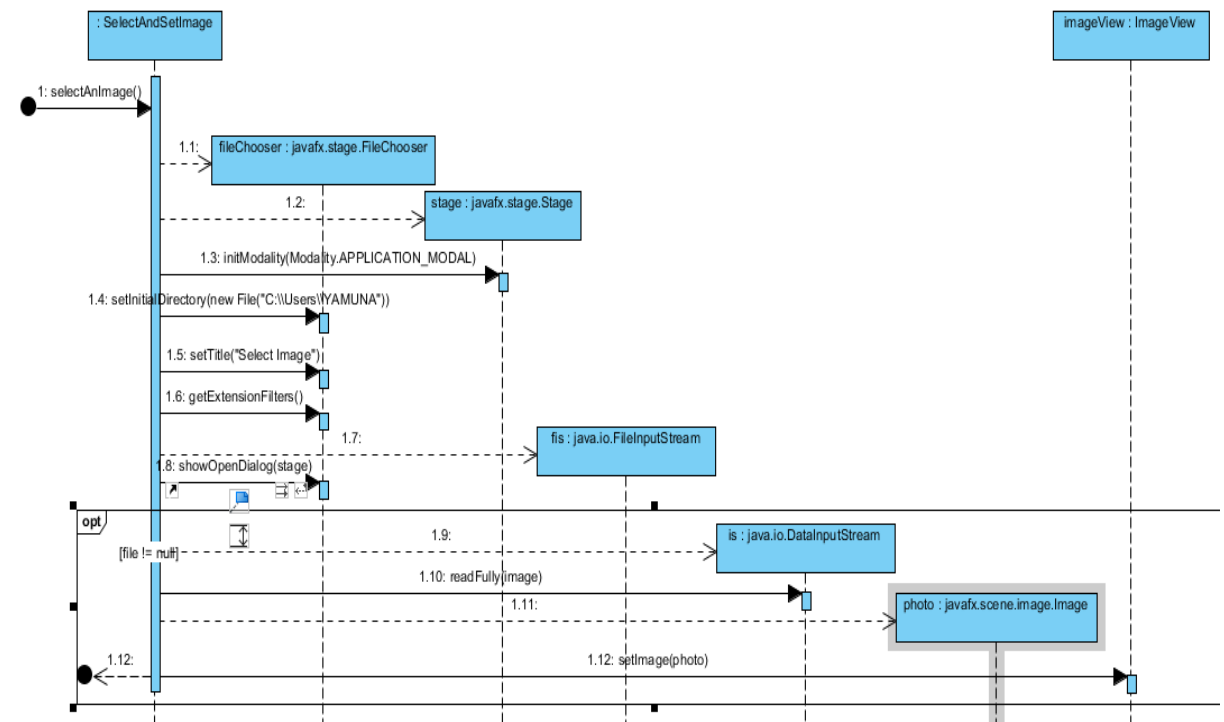


Figure 3.9 Sequence Diagram for Select and Set Image for Employee UI

3.8 Database Design

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. [9]

Since data is a critical component of a software development project, storing data and information in proper manner is very important for the success of the project. When structuring data for storing, data redundancy becomes major problem to cause anomalies in database. Therefore, normalization was used for this project to solve this problem. Database normalization is the process of organizing the attributes and tables of a relational database to minimize data redundancy. Database for the system has normalized up to the Third Normal Form.

3.9 Entity Relationship Diagram

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. It does not define the business processes; it only presents a business data scheme in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities. [10]

And Figure 3.10 given below shows the complete ER diagram for the database design.



Figure 3.10 : Entity Relationship diagram for database design

3.10 User Interface Design

When developing a software system, it should be friendly, interactive and pleasant to users. User interfaces carry a major duty for the above purpose. Therefore, user interface should not be complex and user should be able to understand and use the system easily. Also it should be easy to locate the activities and navigate through them.

In this system prototyping is used to design user interfaces and get feedback on the design. Paper based sketches were used as throwaway prototyping in the project. Evolutionary prototyping also used in this system to design and evaluate. When designing user interfaces Seen Builder software was used. They can be used to build up interfaces, present them to users to get responses, and could be used to develop up to the complete product. Figure 3.11 given below shows Subject Allocation User Interface.

SUBJECT ALLOCATION

Please select a teacher.
Teacher
* Gayathri Perera


* Academic Year
 Middle School Upper School

* Grade

Subject	Year	Main	Remove
No content in table			

Add

TEACHER DETAILS



NIC : 907032295V
EPF No. : 9
Designation : Teacher
Category : Probation
App.Date : 2017-06-11
Address : Kochichikade

Save

Figure 3.11 Subject Allocation User Interface

CHAPTER 4 – IMPLEMENTATION

During the Development phase, the design specifications produced during the design phase are translated into system components. The developed system is validated through a sequence of individual unit testing for usability and integration testing with the objective of ensuring that the system functions as expected and satisfies all requirements. This phase is usually carried out in parallel with the development of testing documents and user documentation. [11]

This is also an important phase of the project, because in this phase the system specification done in design phase transformed to an executable system that the system should be able to satisfy the used requirement. This chapter will explain how the project was implemented according to the system design, and will describe coding and other activities carried out in the implementation process. Also development environment and software tools as well as technologies used are being discussed.

4.1 Implemented Environment

Functional & non-functional requirement of the intended system and selected programming language are considered when selecting software and hardware resources. It is ensured that the Human Resource Management system develops with these configurations and the system is fully compatible with windows 8.1

Hardware & software configuration used,

4.1.1 Hardware Requirements

- Personal Computer with processing power similar or higher than 2.20 GHz
- RAM with 2GB or above capacity.
- Hard Disk with 160 GB or above capacity.

4.1.2 Software Requirements

- MySQL Server 5.5
- MySQL Workbench
- NetBeans IDE

- JavaFX Scene Builder
- Adobe Photoshop
- Microsoft Office Package
- Windows 8.1
- Visual Paradigm

4.2 Development Tools

4.2.1 NetBeans IDE 8.2

NetBeans 8.2 is an IDE (Integrated Development Environment) which means, a software application that provides various facilities for developers and computer programmers for software development. NetBeans 8.2 was used as the IDE for the development of the system. It provides all the tools necessary to develop desktop, web and mobile applications. Also it allows developers to add new plug-ins for their needs. It is a primary tool for developing Java application, but also supports other languages such as, HTML, PHP, JavaScript and C/C++. The NetBeans IDE available for all kinds of operating systems such as Windows, OS X, Linux, Solaris and etc.

There are many features in NetBeans 8.2 that facilitate to easy and speedy development in software. Design GUIs for applications quickly and smoothly by using editors and drag-and-drop tools in the IDE. The NetBeans Editor indents lines, matches words and brackets, and highlights source code syntactically and semantically. It also provides code templates, coding tips, and refactoring tools.

4.2.2 Java Language

Java is a programming language and a computing platform first released by Sun Microsystems in 1995, which has merged into Oracle Corporation in 2010.

Java is a high-level programming language as well as it is an Object oriented language, but simplified to eliminate language features that cause common programming errors. Java source code files (files with a .java extension) are compiled into a format called byte code (files with a .class extension), which can then be executed by a Java interpreter. Compiled Java code can run on most computers because Java interpreters and runtime environments, known as Java Virtual Machines

(VMs), exist for most operating systems, including UNIX, the Macintosh OS, and Windows. Byte code can also be converted directly into machine language instructions by a just-in-time compiler (JIT). [12]

To manage memory in the object lifecycle Java uses an automatic garbage collector. The programmer has to decide when objects are created, and recovering the memory, is a responsibility of the Java runtime once objects are no longer in use. Once no references to an object remain, the unreachable memory will automatically be cleared by the garbage collector.

Java is a general purpose programming language with a number of features that make the language well suited for use for standalone system moreover on the World Wide Web.

4.2.3 MySQL Server

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL) and runs on virtually all platforms, including Linux, UNIX, and Windows. It ships with no GUI tools to administer MySQL databases or manage data contained within the databases.

Users may use the included command line tools, or use MySQL “front-ends”, desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records.

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs which provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. It saves the design time for designers and programmers by forward engineering and synchronizing the ER diagram into a database schema.

4.2.4 JavaFX Scene Builder

JavaFX Scene Builder is a visual layout tool that lets users quickly design JavaFX application user interfaces, without coding. Users can drag and drop UI components to a work area, modify their properties, apply style sheets, and the FXML code for the

layout that they are creating is automatically generated in the background. The result is an FXML file that can then be combined with a Java project by binding the UI to the application's logic. [13]

JavaFX Scene Builder was used to design user interfaces and to get user feedbacks about the design and user friendliness. Furthermore, CSS and Adobe Photoshop were used for the styling of the system interfaces.

4.2.5 Visual Paradigm

Visual Paradigm is a software design tool designed for agile software projects which supports modelling standards such as UML, SysML, ERD, DFD, BPMN, ArchiMate, etc. It is facilitated for building software and systems that excel in user experience by supporting effective use case identification, requirements gathering, flow of events, wire framing, requirement specification generation, etc.

4.2.6 Hibernate

Hibernate was used as an object-relational mapping (ORM) library for the Java language in this project. Hibernate provides a framework to mapping a traditional relational database to an object - oriented domain model.

Hibernate is a high-performance Object/Relational persistence and query service which is licensed under the open source GNU Lesser General Public License (LGPL) and is free to download. Hibernate not only takes care of the mapping from Java classes to database tables (and from Java data types to SQL data types), but also provides data query and retrieval facilities. [14]

There are two ways using to object/relational mappings.

- XML Document - mapping file instructs Hibernate how to map the defined class or classes to the database tables. Though many Hibernate users choose to write the XML by hand, a number of tools exist to generate the mapping document. These include XDoclet, Middlegen and AndromDA for advanced Hibernate users.

- Annotation - Hibernate Annotations is the powerful way to provide the metadata for the Object and Relational Table mapping. Annotations can be used in addition to or as a replacement of XML mapping metadata.

Annotations are used in this project in Object/relational mappings.

Hibernate Query Language (HQL) that is similar in appearance to SQL is provided by Hibernate to query the database. In this project Named queries that can be defined in Hibernate mapping files or through the use of JPA annotations were used for this purpose.

Figure 4.2.6.1 given below shows the Architecture of Hibernate.

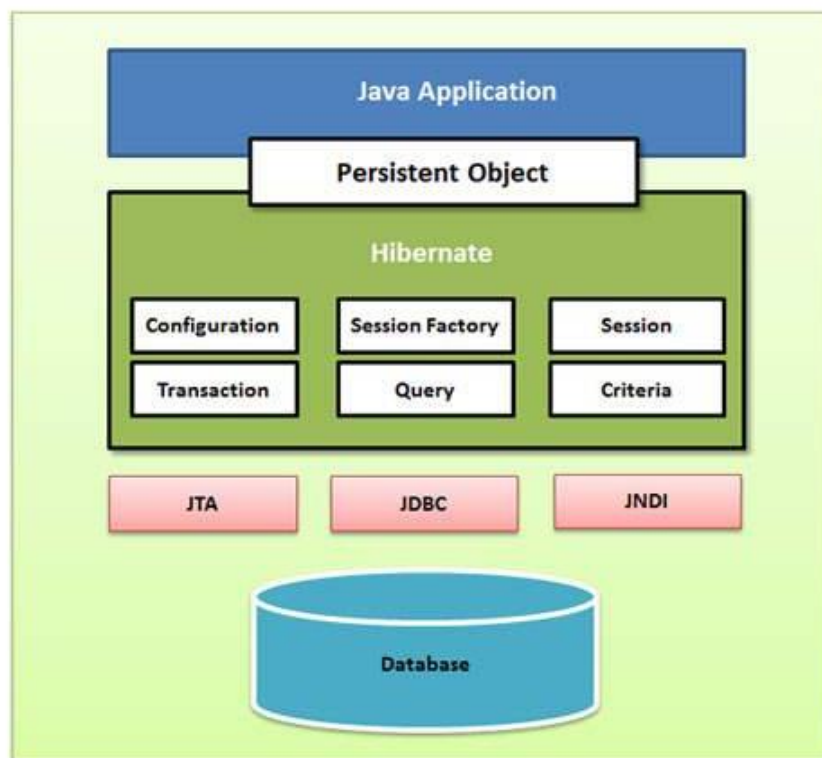


Figure 4.2 : Architecture of Hibernate

4.2.7 Jasper Report

Jasper Report is an open source reporting library that can be embedded into any Java application. It provides the necessary features to generate dynamic reports, including data retrieval using JDBC (Java Database Connectivity), as well as support for parameters, expressions, variables, and groups.

It also includes advanced features, such as custom data sources, script lets, and sub reports. It has a flexible report layout and can present data textually or graphically. It is capable of exporting reports to a variety of formats and developers can supply data in multiple ways.

4.3 Implementation

MVC model was used as the architecture of the system with some adjustments. MVC refers to Model, view and controller. It allows to separate application logic, user interface layer and data layer. In this project view split into two parts, FXML and UI controller, because we use JavaFX for this project. And controller is split into hibernate and Dao packages.

- Model: Represents Business entity that store data retrieved to the controller and displayed in the view. Used to transfer data.
- View: User interacts with this layer split into two parts, FXML and UI controller due to use of JavaFX.
- Controller: send commands to the model to update the model's state. It can also send commands to its associated view to change the view's presentation of the model.

Figure 4.3.1 given below shows the MVC architecture.

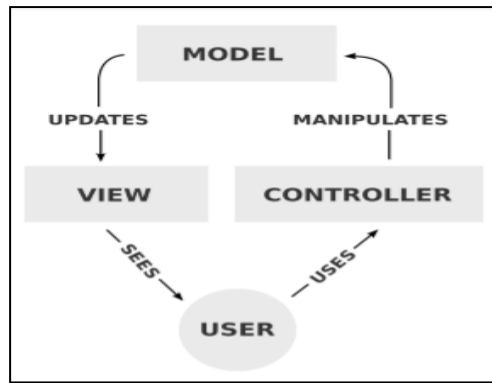


Figure 4.3 : MVC Architecture

Figure 4.3.2 shows the how classes are organizing in the system.

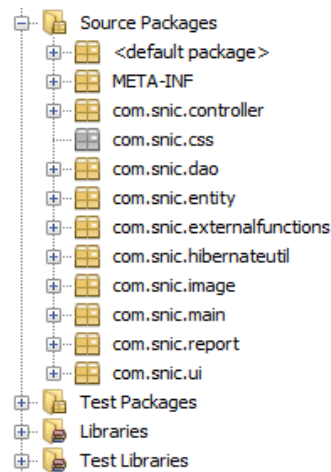


Figure 4.3 : Classes Organizing in the System

4.4 Data Layer Implementation

Selected DBMS for this project MySQL Server 5.5 and MySQL Workbench were used as a graphical tool to develop the database.

In this layer hibernate was used as an ORM library, which is used to map database tables with the model classes. They have relationships between them such as one-to-one, one-to-many, many-to-many, etc. And CRUD operations (which means Create, Retrieve, Update, delete) can be performed on them.

4.4.1 Hibernate Configuration

We have to create a Hibernate configuration file named hibernate.cfg.xml which includes all the information about mapping such as database tables to java classes and a set of configuration settings related to database and other related parameters. This task was done by using Hibernate Configuration Wizard. Following shows the xml codes use for configuring hibernate (code 4.1).

```

<hibernate-configuration>
  <session-factory>
    <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
    <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/hr?zeroDateTimeBehavior=convertToNull</property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.connection.password">123</property>
    <mapping class="com.snic.entity.Civilstatus"/>
    <mapping class="com.snic.entity.Designation"/>
    <mapping class="com.snic.entity.Gender"/>
    <mapping class="com.snic.entity.Grade"/>
    <mapping class="com.snic.entity.Leavecategory"/>
    <mapping class="com.snic.entity.Leaves"/>
    <mapping class="com.snic.entity.Leavestatus"/>
    <mapping class="com.snic.entity.Leavetype"/>
    <mapping class="com.snic.entity.Performance"/>
    <mapping class="com.snic.entity.Qualification"/>
    <mapping class="com.snic.entity.QulificationType"/>
    <mapping class="com.snic.entity.Salary"/>
    <mapping class="com.snic.entity.Section"/>
    <mapping class="com.snic.entity.SectionGrade"/>
    <mapping class="com.snic.entity.SectionSubject"/>
    <mapping class="com.snic.entity.Seminar"/>
    <mapping class="com.snic.entity.SeminarTeacher"/>
    <mapping class="com.snic.entity.Seminarconduct"/>
    <mapping class="com.snic.entity.SeminarType"/>
    <mapping class="com.snic.entity.Subject"/>
    <mapping class="com.snic.entity.Teacher"/>
    <mapping class="com.snic.entity.TeacherPerformance"/>
    <mapping class="com.snic.entity.TeacherQulification"/>
    <mapping class="com.snic.entity.TeacherSalary"/>
  </session-factory>
</hibernate-configuration>

```

Code 4.1 hibernate.cfg.xml

4.4.2 Create Java Entity Classes

In this section mainly discuss about entity classes that can be auto generated from the database by using NetBeans options. Named queries are also created to retrieve data from database.

Following shows the entity class codes and named queries of Leave entity. (Code 4.2)

```

package com.snic.entity;

import java.io.Serializable;
import java.math.BigDecimal;
import java.util.Date;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.xml.bind.annotation.XmlRootElement;

@Entity
@Table(name = "leaves")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Leaves.findAll", query = "SELECT l FROM Leaves l ORDER BY l.teacherId.name ASC")
    , @NamedQuery(name = "Leaves.findById", query = "SELECT l FROM Leaves l WHERE l.id = :id")
    , @NamedQuery(name = "Leaves.findByName", query = "SELECT l FROM Leaves l WHERE l.name = :name")
    , @NamedQuery(name = "Leaves.findByRequestdate", query = "SELECT l FROM Leaves l WHERE l.requestdate = :requestdate")
    , @NamedQuery(name = "Leaves.findByDescription", query = "SELECT l FROM Leaves l WHERE l.description = :description")
    , @NamedQuery(name = "Leaves.findByStartdate", query = "SELECT l FROM Leaves l WHERE l.startdate = :startdate")
})

```

Code 4.2 Leaves Entity

Mapping detail in entity level was done using annotations in these classes. All entity classes implement the Serializable Interface. Following shows the annotations that is used to map other tables and attributes of Leaves entity. (Code 4.3)


```

public class Leaves implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "id")
    private Integer id;
    @Column(name = "name")
    private String name;
    @Column(name = "requestdate")
    @Temporal(TemporalType.DATE)
    private Date requestdate;
    @Column(name = "description")
    private String description;
    @Column(name = "startdate")
    @Temporal(TemporalType.DATE)
    private Date startdate;
    @Column(name = "enddate")
    @Temporal(TemporalType.DATE)
    private Date enddate;
    @Column(name = "count")
    private BigDecimal count;
    @Column(name = "approveddate")
    @Temporal(TemporalType.DATE)
    private Date approveddate;
    @Column(name = "approvedby")
    private String approvedby;
    @JoinColumn(name = "leavecategory_id", referencedColumnName = "id")
    @ManyToOne(optional = false)
    private Leavecategory leavecategoryId;
}

```

Code 4.3 Leaves Entity Mapping details

Entity class was used to store objects or instances in database tables. Following shows some setters and getters of the Leaves entity and the implementation of override methods. (Code 4.4)

```

@Override
public int hashCode() {
    int hash = 0;
    hash += (id != null ? id.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof Leaves)) {
        return false;
    }
    Leaves other = (Leaves) object;
    if ((this.id == null && other.id != null) || (this.id != null && !this.id.equals(other.id))) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "entity.Leaves[ id=" + id + " ]";
}

```

Code 4.4 Leaves Entity – Override Methods

4.4.3 Hibernate Session

Configuration object is used to create a Session Factory object which in turn configures Hibernate for the application using the supplied configuration file and allows for a Session object to be instantiated. The Session Factory is a thread safe object and used by all the threads of an application.

The Session Factory is a heavyweight object so usually it is created during application start-up and kept for later use. You would need one Session Factory object per database using a separate configuration file. So if you are using multiple databases then you would have to create multiple Session Factory objects.

Following shows the java code for create session factory. (Code 4.5)

```

import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;

public class HibernateUtil {

    private static final SessionFactory SESSIONFACTORY;

    static {
        try {
            Configuration configuration = new Configuration();
            configuration.configure();
            ServiceRegistry serviceRegistry = new StandardServiceRegistryBuilder().applySettings(configuration.getProperties()).
                buildServiceRegistry();
            SESSIONFACTORY = configuration.configure().buildSessionFactory(serviceRegistry);
        } catch (HibernateException ex) {
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return SESSIONFACTORY;
    }

    public static void closeFactory() {
        if (SESSIONFACTORY != null) {
            try {
                SESSIONFACTORY.close();
            } catch (HibernateException ex) {
                System.out.println("Couldn't close session factory");
            }
        }
    }
}

```

Code 4.5 Hibernate Util

4.5 Business Logic Layer Implementation

Business logic layer provides the link between data Layer and interface layer. It can be recognized as ‘Controller’ in MVC architecture.

Business logic is the part of the program that encodes the real-world business rules that determine how data can be created, displayed, stored, and changed. It is contrasted with the remainder of the software which might be concerned with lower-level details of managing a database or displaying the user interface, system infrastructure, or generally connecting various parts of the program. [15]

4.5.1 DAO (Data Access Object)

In computer software, a data access object (DAO) is an object that provides an abstract interface to some type of database or other persistence mechanism. By mapping application calls to the persistence layer, DAO provides some specific data operations without exposing details of the database. This isolation supports the Single responsibility principle. It separates what data accesses the application needs, in terms

of domain-specific objects and data types (the public interface of the DAO), from how these needs can be satisfied with a specific DBMS, database schema, etc. (the implementation of the DAO).

This project uses common class for CRUD operations in every form. CRUD stands for create, read, update and delete. Following codes shows method in Generic Dao to insert objects to database. (Code 4.6)

```
public class GenericDao {  
  
    private static Session session;  
    private static ObservableList resultSet;  
  
    public static void save(Object object) {  
        try {  
            session = HibernateUtil.getSessionFactory().openSession();  
            session.beginTransaction();  
            session.save(object);  
            session.getTransaction().commit();  
        } catch (HibernateException e) {  
            session.getTransaction().rollback();  
            System.out.println("Insertion failed! Transaction rolledback successfully." + e);  
        } finally {  
            session.close();  
        }  
    }  
}
```

Code 4.6 Generic Dao – Insert / Save method

Following codes shows method in Generic Dao to update set of objects from the database according to given conditions. (Code 4.7)

```

public static void update(Object object) {
    try {
        /* Open transaction initialized in HibernateUtil class */
        session = HibernateUtil.getSessionFactory().openSession();
        session.beginTransaction();
        /* Update object */
        session.update(object);
        /* Commit changes to the database */
        session.getTransaction().commit();
    } catch (HibernateException e) {
        /* Rollback changes in case of exception */
        session.getTransaction().rollback();
        System.out.println("Update failed! Transaction rolledback successfully. " + e);
    } finally {
        /* Close the session */
        session.close();
    }
}
}

```

Code 4.7 Generic Dao – Update method

Following codes shows method in Generic Dao to delete set of objects from the database according to given conditions. (Code 4.8)

```

public static void delete(Object object) {
    try {
        /* Open transaction initialized in HibernateUtil class */
        session = HibernateUtil.getSessionFactory().openSession();
        session.beginTransaction();

        /* Update object */
        session.delete(object);

        /* Commit changes to the database */
        session.getTransaction().commit();
    } catch (HibernateException e) {
        /* Rollback changes in case of exception */
        session.getTransaction().rollback();
        System.out.println("Update failed! Transaction rolledback successfully. " + e);
    } finally {
        /* Close the session */
        session.close();
    }
}
}

```

Code 4.8 Generic Dao – Delete method

Following codes shows method in Generic Dao to get set of objects from the database according to given conditions. (Code 4.9)

```

public static ObservableList getItemList(String namedQuery)
    throws HibernateException {
    try {
        session = HibernateUtil.getSessionFactory().openSession();
        resultSet = null;
        Query query = session.getNamedQuery(namedQuery);
        List tempResultSet = query.list();
        resultSet = FXCollections.observableArrayList(tempResultSet);
    } catch (HibernateException e) {
        System.out.println("Insertion failed! Transaction rolledback successfully." + e);
    } finally {
        session.close();
    }
    return resultSet;
}

```

Code 4.9 Generic Dao – Get Item List method

4.5.2 Event Handlers

To create a usable system, it must respond to commands from the user. To do so, Java programs rely on events that describe user actions. These are known as Event Handlers. Following codes shows how event handler used in Employee Registration UI. (Code 4.10)

```

@FXML
private void btnSaveOA(ActionEvent event) {
    if (isValid()) { //checking update or save
        Alert popup = new Alert(Alert.AlertType.CONFIRMATION, "Are you sure you want to save the changes?");
        popup.showAndWait();
        if (popup.getResult() == ButtonType.OK) {
            Teacher teacher;
            if (isupdate) {
                teacher = getteacher(); //Update
                setData(teacher);
                try {
                    GenericDao.update(teacher);
                    TeacherviewController c = new TeacherviewController();
                    c = c.getInstance();
                    c.refreshData();
                } catch (HibernateException e) {
                    System.out.println(e.getMessage());
                }
                btnSave.getScene().getWindow().hide();
                Notifications.create().text("Successfully saved.").darkStyle().position(Pos.TOP_RIGHT).showInforma
            } else {
                teacher = new Teacher(); //Save
                setData(teacher);
                GenericDao.save(teacher);
                /* Calculate salary */
                calculateSalary(teacher);
                formClear();
                Notifications.create().text("Successfully updated.").darkStyle().position(Pos.TOP_RIGHT).showInfor
            }
        }
    }
}

```

Code 4.10 Generic Dao – Get Item List method

4.6 Interface Layer Implementation

The user interface layer represents the front end of the system, and contains the actual GUI elements that users view and click.

To develop user interfaces JavaFX Scene Builder was used. And Adobe Photoshop was used to create and edit images for the interfaces. Developers can easily add UI controllers to user interfaces by the drag and drop facility provided by Scene Builder.

JavaFX Scene Builder will automatically generate the necessary xml code for UI components which was added by the developer to the user interface. This facility saves the developing time.

Following shows the auto generated xml codes for creating Create Employee interface using JavaFX Scene Builder (Code 4.11), (Code 4.12)

```

<?xml version="1.0" encoding="UTF-8"?>

<?import com.jfoenix.controls.JFXButton?>
<?import com.jfoenix.controls.JFXComboBox?>
<?import com.jfoenix.controls.JFXDatePicker?>
<?import com.jfoenix.controls.JFXListView?>
<?import com.jfoenix.controls.JFXRadioButton?>
<?import com.jfoenix.controls.JFXTextArea?>
<?import com.jfoenix.controls.JFXTextField?>
<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIconView?>
<?import javafx.geometry.Point3D?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.ScrollPane?>
<?import javafx.scene.control.ToggleGroup?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>

<AnchorPane id="AnchorPane" maxHeight="435.0" prefWidth="1025.0" styleClass="white-pane" stylesheets="@../css/mate
  <children>
    <Pane maxWidth="1025.0" minWidth="1025.0" prefHeight="55.0" styleClass="blue-title-pane">
      <children>
        <Label layoutX="30.0" layoutY="17.0" styleClass="label-title-main" text="CREATE EMPLOYEE" />
        <Label fx:id="lb1EPF" layoutX="248.0" layoutY="13.0" prefHeight="35.0" prefWidth="190.0" />
      </children>
    </Pane>
    <ScrollPane fitToWidth="true" hbarPolicy="NEVER" layoutY="54.0" maxHeight="480.0" prefViewportWidth="1025.0"
      <content>
        <AnchorPane styleClass="white-pane">
          <children>
            <AnchorPane layoutX="166.0" layoutY="27.0">
              <children>

```

Code 4.11 XML codes for create Employee

```

<font></Label>
<ComboBox fx:id="cmbYear" layoutX="183.0" layoutY="130.0" onAction="#cmbYearAP" prefHeight="25.0" prefWidth="209.0" promptText="Select
<ComboBox fx:id="cmbForm" layoutX="183.0" layoutY="172.0" onAction="#cmbFormAP" prefHeight="25.0" prefWidth="209.0" promptText="Select
<ComboBox fx:id="cmbClassTeacher" layoutX="183.0" layoutY="306.0" onAction="#cmbClassTeacherAP" prefHeight="25.0" prefWidth="209.0" pr
<TextArea fx:id="txaRemarks" layoutX="183.0" layoutY="508.0" onKeyReleased="#txaRemarksKR" prefHeight="68.0" prefWidth="209.0" prompt
<TextField fx:id="txtSearchClassName" layoutX="764.0" layoutY="96.0" onKeyReleased="#txtSearchClassNameKR" prefHeight="25.0" prefWidth
<Button fx:id="btnAdd" layoutX="147.0" layoutY="652.0" mnemonicParsing="false" onAction="#btnAddAP" onKeyPressed="#btnAddKP" text="Add
<Button fx:id="btnUpdate" layoutX="201.0" layoutY="652.0" mnemonicParsing="false" onAction="#btnUpdateAP" onKeyPressed="#btnUpdateKP"
<Button fx:id="btnDelete" layoutX="272.0" layoutY="652.0" mnemonicParsing="false" onAction="#btnDeleteAP" onKeyReleased="#btnDeleteKR"
<Button fx:id="btnClear" layoutX="346.0" layoutY="652.0" mnemonicParsing="false" onAction="#btnClearAP" onKeyReleased="#btnClearKR" te
<Label layoutX="48.0" layoutY="210.0" prefHeight="17.0" prefWidth="51.0" text="Class" textFill="WHITE">
<font></Label>
<ComboBox fx:id="cmbCls" layoutX="183.0" layoutY="211.0" onAction="#cmbClsAP" prefHeight="25.0" prefWidth="209.0" promptText="Select C
<TextField fx:id="txtClassName" layoutX="183.0" layoutY="264.0" onKeyReleased="#txtClassNameKR" prefHeight="25.0" prefWidth="209.0" />
<Label layoutX="52.0" layoutY="405.0" text="Maximum Count" textFill="WHITE">
<font></Label>
<TextField fx:id="txtMaxCount" layoutX="183.0" layoutY="401.0" onKeyReleased="#txtMaxCountKR" prefHeight="25.0" prefWidth="209.0" prom
<Label layoutX="52.0" layoutY="357.0" text="Class Status" textFill="WHITE">
<font></Label>
<ComboBox fx:id="cmbSelectStatus" layoutX="183.0" layoutY="357.0" onAction="#cmbSelectStatusAP" prefHeight="25.0" prefWidth="209.0" pr
<Label layoutX="545.0" layoutY="105.0" text="Search By Class Name" textFill="WHITE">
<font></Label>
<Label layoutX="545.0" layoutY="151.0" text="Search By Status" textFill="WHITE">
<font></Label>
<Label layoutX="545.0" layoutY="193.0" text="Search By Class Teacher" textFill="WHITE">
<font></Label>
<ComboBox fx:id="cmbSearchStatus" layoutX="765.0" layoutY="146.0" onAction="#cmbSearchStatuAP" prefHeight="25.0" prefWidth="248.0" pro
<ComboBox fx:id="cmbSearchTeacher" layoutX="765.0" layoutY="193.0" onAction="#cmbSearchTeacherAP" prefHeight="25.0" prefWidth="248.0"
<Button fx:id="btnSearchClear" layoutX="969.0" layoutY="277.0" mnemonicParsing="false" onAction="#btnSearchClearAP" onKeyReleased="#bt
<Pagination fx:id="pagination" layoutX="508.0" layoutY="365.0" prefHeight="339.0" prefWidth="526.0" />
<TableView>
<TextField fx:id="txtNoofStudents" layoutX="183.0" layoutY="446.0" onKeyReleased="#txtNoofStudentsKR" prefHeight="25.0" prefWidth="209
<Label layoutX="47.0" layoutY="51.0" prefHeight="17.0" prefWidth="280.0" styleClass="topic" text="Class Teacher Assign" textFill="WHIT
<font>
</Label>
</children>
</AnchorPane>

```

Code 4.12 XML codes for create Employee

To manage interface, event handlers and some other controlling methods were added to a file called UI Controller which is auto generated by NetBeans. Following shows the java code Create Employee interface controller class (Code 4.13), (Code 4.14), (Code 4.15), (Code 4.16)


```

import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;
import javafx.scene.control.ToggleGroup;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import org.controlsfx.control.Notifications;
import org.hibernate.HibernateException;

/**
 * FXML Controller class
 *
 * @author YAMUNA
 */
public class TeacherregistrationController implements Initializable {

    @FXML
    private TextField txtName;
    @FXML
    private TextField txtNIC;
    @FXML
    private DatePicker dtpdob;
    @FXML
    private RadioButton rdbmale;
    @FXML
    private RadioButton rdbfemale;
    @FXML
    private RadioButton rdbmarried;
    @FXML
    private RadioButton rdbunmarried;
    @FXML
    private TextArea txtAddress;

```

Code 4.13 UI Controller

```

@Override
public void initialize(URL url, ResourceBundle rb) {
    formClear();
}

@FXML
private void btnSaveOA(ActionEvent event) {
    if (isValid()) { //checking update or save
        Alert popup = new Alert(Alert.AlertType.CONFIRMATION, "Are you sure you want to save the changes?");
        popup.showAndWait();
        if (popup.getResult() == ButtonType.OK) {
            Teacher teacher;
            if (isupdate) {
                teacher = getteacher(); //Update
                setData(teacher);
                try {
                    GenericDao.update(teacher);
                    TeacherviewController c = new TeacherviewController();
                    c = c.getInstance();
                    c.refreshData();
                } catch (HibernateException e) {
                    System.out.println(e.getMessage());
                }
            }
            btnSave.getScene().getWindow().hide();
            Notifications.create().text("Successfully saved.").darkStyle().position(Pos.TOP_RIGHT).showInformation();
        } else {
            teacher = new Teacher(); //Save
            setData(teacher);
            GenericDao.save(teacher);
        }
    }
}

```

Code 4.14 UI Controller

```

private void formClear() {
    //<editor-fold defaultstate="collapsed" desc="Code">
    btnSave.setText("Save"); //after saving a record form clear will be there
    txtName.clear();
    txtNIC.clear();
    //txtEPF.clear();
    txtEmail.clear();
    txtLandno.clear();
    txtmobileno.clear();

    rdbmale.setSelected(true);
    rdbmarried.setSelected(true);
    rdbDiploma.setSelected(true);

    txtaAddress.clear();
    txtaExperience.clear();

   .dtpappointmentdate.setValue(LocalDate.now());
   .dtpdob.setValue(LocalDate.now());

    imgPhoto.setImage(new Image("/com/unic/image/user.png")); //imgPhoto clear

    cmbDesignation.getSelectionModel().clearSelection();
    cmbTeacherStatus.getSelectionModel().clearSelection();
    lstLeft.getItems().clear(); //list clear
    lstRight.getItems().clear();

    rdbDiplomaOA(null);
    formValidation();
    FormFill();
//</editor-fold>
}

```

Code 4.15 UI Controller

```

private void formFill() { //loading contents to the combo box
    cmbDesignation.setItems(GenericDao.getItemList("Designation.findAll"));
    cmbTeacherStatus.setItems(GenericDao.getItemList("TeacherStatus.findAll"));
    setEPFNo();
}

private void formValidation() {
    //<editor-fold defaultstate="collapsed" desc="code">
    validation = new HashMap<>();
    msgAddress.setVisible(false);
    msgAppointment.setVisible(false);
    msgDob.setVisible(false);
    msgEmail.setVisible(false);
    msgExperience.setVisible(false);
    msgLand.setVisible(false);
    msgMobile.setVisible(false);
    msgName.setVisible(false);
    msgNic.setVisible(false);

    txtName.focusedProperty().addListener((obs, oldVal, newVal)
        -> {
            if (!newVal) {
                if (txtName.getText().trim().matches("^[_A-Za-z]{3}[_A-Za-z]*([_A-Za-z]*\\s[_A-Za-z]*)*$")) {
                    Validate.isValid(imgName, msgName);
                    validation.put(txtName.getPromptText().trim().replace(" ", ""), Boolean.TRUE);
                } else {
                    Validate.isInvalid(imgName, msgName, "Hmm!Name seems to be invalid :(");
                    validation.put(txtName.getPromptText().trim().replace(" ", ""), Boolean.FALSE);
                }
            }
        });
   .dtpdob.focusedProperty().addListener((obs, oldVal, newVal)
        -> {

```

Code 4.16 UI Controller

Furthermore, CSS file can be created to style the interface with various colors, images and effects. Following shows the CSS code for user interface CSS file (Code 4.17)

```
/* Text field */
.jfx-text-field-login
{
    -jfx-focus-color: #eff0f1;
    -jfx-unfocus-color: #a0a2ab;
    -fx-prompt-text-fill: #a0a2ab;
}

.text-field-login
{
    -fx-font-family: "Calibri";
    -fx-font-size:14px;
    -fx-text-fill: #eff0f1; /*input color*/
    -fx-highlight-fill: #4eb769; /*Green*/
}

.jfx-text-field
{
    -jfx-focus-color: #686868;
    -jfx-unfocus-color: #a0a2ab;
    -fx-prompt-text-fill: #686868;
}

.text-field
{
    -fx-font-family: "Calibri";
    -fx-font-size:14px;
    -fx-prompt-text-fill: #686868; /*input color*/
    -fx-highlight-fill: #4eb769; /*Green*/
}

/* Text area */
.jfx-text-area
```

Code 4.17 CSS codes for UI styles

CHAPTER 5 – EVALUATION

Testing is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements.

In software testing, verification and validation is the process of checking that a software system meets specifications and that it fulfills its intended purpose. Verification is a process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. Validation checks that the product design satisfies or fits the intended use or the software meets the user requirements.

5.1 Testing Methods

There are various types of testing that can be used to test the system. Generally, four recognized levels of tests: unit testing, integration testing, system testing, and acceptance testing. In this system mainly these four testing methods were used to ensure that the system is steady and stable. Testing was carried out while development process was going on and completed.

5.1.1 Unit Testing

Unit testing is performed by the respective developers on the individual units of source code assigned areas. It was done while implementing the system to check whether the source codes are accurate and working properly. Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. [16]

5.1.2 Integration Testing

Integration testing is defined as the testing of combined parts of an application to determine that they function correctly and gives the intended results. Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. [17]

5.1.3 System Testing

System Testing is a software testing process which is performed on the completed system using system specification created in the analysis phase. System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. [18]

5.2 Test Cases

A test case, in software engineering, is a set of conditions under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do. The mechanism for determining whether a software program or system has passed or failed such a test is known as a test oracle. In some settings, an oracle could be a requirement or use case, while in others it could be a heuristic. It may take many test cases to determine that a software program or system is considered sufficiently scrutinized to be released. Test cases are often referred to as test scripts, particularly when written - when they are usually collected into test suites. [19]

The following shows some test cases conducted in this system

5.2.1 Test cases and test results for Login Module

No	Test Case	Expected Result	Pass /Fail
1	Login correct username and password entered in the relevant fields	Login to the system show Welcome Window	Pass
2	No values entered in the username field	Display error message "Login Fail"	Pass
3	No values entered in the password field	Display error message "Login Fail"	Pass

Table 5.2. 1 Test cases and Test results for Login Module

5.2.2 Test cases and test results for Subject Allocation Module

No	Test Case	Expected Result	Pass /Fail
1	Select the teacher form the combo box	Screen should be visible to enter data	Pass
2	Enter academic year	Next to the academic year green color tick mark will be visible	Pass
3	Select section from the combo box	Section should be selected	Pass
4	Select grade from the combo box	Grade should be selected	Pass
5	Select subjects from the left list view	Subjects should be viewed in right list view	Pass
6	Enter all subjects which allocate for the teacher and click add button	Added data will be shown in the table	Pass
7	Then press save button	The message will be displayed "Do you want to save data?" If click "yes" record will be saved. . If click "no" save will be cancelled	Pass

Table 5.2. 2 Test Cases and Test results for subject Allocation module

5.2.3 Test cases and test results for Development Module

No	Test Case	Expected Result	Pass /Fail
1	Select the teacher form the combo box	Teacher should be selected	Pass
2	Select the subject form the combo box	Subject should be selected	Pass
3	Select conduct organization from the combo box	organization should be selected	Pass
4	Enter venue	Next to the venue green color tick mark will be visible	Pass
5	Enter description	Next to the description green color tick mark will be visible	Pass
6	Select the seminar type form the combo box	Seminar type should be selected	Pass
7	Select the date	Selected date should be displayed	Pass
8	Select the time	Selected time should be displayed	Pass
9	Enter all information correctly and click save	The message will be displayed “Do you want to save data?” If click “yes” record will be saved. . If click “no” save will be cancelled	Pass

Table 5.2. 3 Test Cases and Test Results for Development module

5.2.4 Test cases and test results for Performance Module

No	Test Case	Expected Result	Pass /Fail
1	Select the teacher form the combo box	Teacher should be selected	Pass
2	Enter academic year	Next to the academic year green color tick mark will be visible	Pass
3	Click the toggle buttons about syllabus covered and Books corrected	Toggle button's color will be change in to blue color	Pass
4	Enter no. of weak students	Next to the weak students green color tick mark will be visible	Pass
5	Select special values from the left list view	Special values should be viewed in right list view	Pass

6	Enter increment for the teacher	Next to the increment green color tick mark will be visible	Pass
7	Enter comments for the teacher	Next to the comments green color tick mark will be visible	Pass
8	Enter all information correctly and click save	Confirmation message will be displayed “Do you want to save changes” if click “yes” again another message will be displayed if that record already exist. Otherwise record will be saved. If click “no” save will be cancelled	Pass

Table 5.2 .4 Test Cases and Test Results for Performance module

5.3 User Evaluation

Finally, the system is tested by the user for acceptance testing. The acceptance testing is carried out by implementing the system at the real working environment along with the real test data and available conditions in the actual background. Then the user can decide if all the functional and non-functional requirements are included in the system or not. The response of the user has shown that all the expected system modules were implemented and that he/she was satisfied with the system. The performance & drawbacks were monitored as well as the test results & the user friendliness of the system. Over all valuation of user feedback was done as the final stage.

5.3.1 Results respect to the Objectives

The main objective of this system is to provide an automated procedure to manage the teachers' human resources activities which were handled manually. As well as increase their efficiency and correctness of their work by means of that procedure.

The solution we provide through this project was fairly fixed with these objectives. Automation of activities previously done manually increased the efficiency of day to day activities. Also minimize the errors/mistakes when activities doing manually and increase the correctness of the information and calculations.

Given below are some objectives expected by this project and how we fairly achieved them.

- Improving the efficiency of subject allocation for the teachers
Due to the system doing the most of the time wasting manual activities efficiency were improved.
- Reduce the possibility of deriving errors:
Due to automating of recording teachers' information user no longer want to worry about correctness of their information.
- Increasing the accuracy of data:
By the validating the correctness of entering data over the system and providing privileges to users that can enter and change the data the system increases the accuracy of data.
- Provide supportive information for management:
By providing daily, weekly, monthly statistical reports and suggestions in some activities the system provides supportive information for management.
- Improve security, including easy access:
Tasks and information are easily available for authorized users and the system prevents them from reaching the wrong people.

When considering all the objectives of the project and other client requests were able to achieve fairly. The feedback from the client, end users and the other related persons were favorable and motivated.

5.3.2 Client Feedback

After implementing the system successfully, it was evaluated by the different users of the school. Tests were carried out by implementing the system at the real working environment along with the real test data in the actual background. All the users were asked to carry out their relevant activities and test the system. Finally, their observations were collected and analyzed. Figure 5.1 given below shows the sample User Evaluation form used to collect user feedback.

User Evaluation Form

Human Resource Management System for St Nicholas' International College

1. Name Mr. K.K.D. Kumara.....

2. Designation H.R. Manager.....

3. How this program affect to your current system?
a. Became easy b. Became difficult c. Nothing special

4. How user friendly is our software interfaces?
a. Became easy b. Became difficult c. Nothing special

5. How do you feel about the functionalities of this system?
a. Very successful b. Successful c. Moderate

6. Is it easy to understand the program features?
a. Became easy b. Became difficult c. Nothing special

7. Does the program useful for your daily works?
a. Very successful b. Successful c. Weak

8. Does it gives user guidance?
a. Yes b. No

9. What is the level of learnability of this system?
a. Very successful b. Successful c. Weak

10. Are you satisfies with our software?
a. Satisfied b. Moderately satisfied c. Not satisfied



.....
Signature

Figure 5. 1 Sample User Evaluation Form

From the feedback forms most users were in a satisfied situation about the system. Their answers were evaluated and the result was as follow figure 5.2

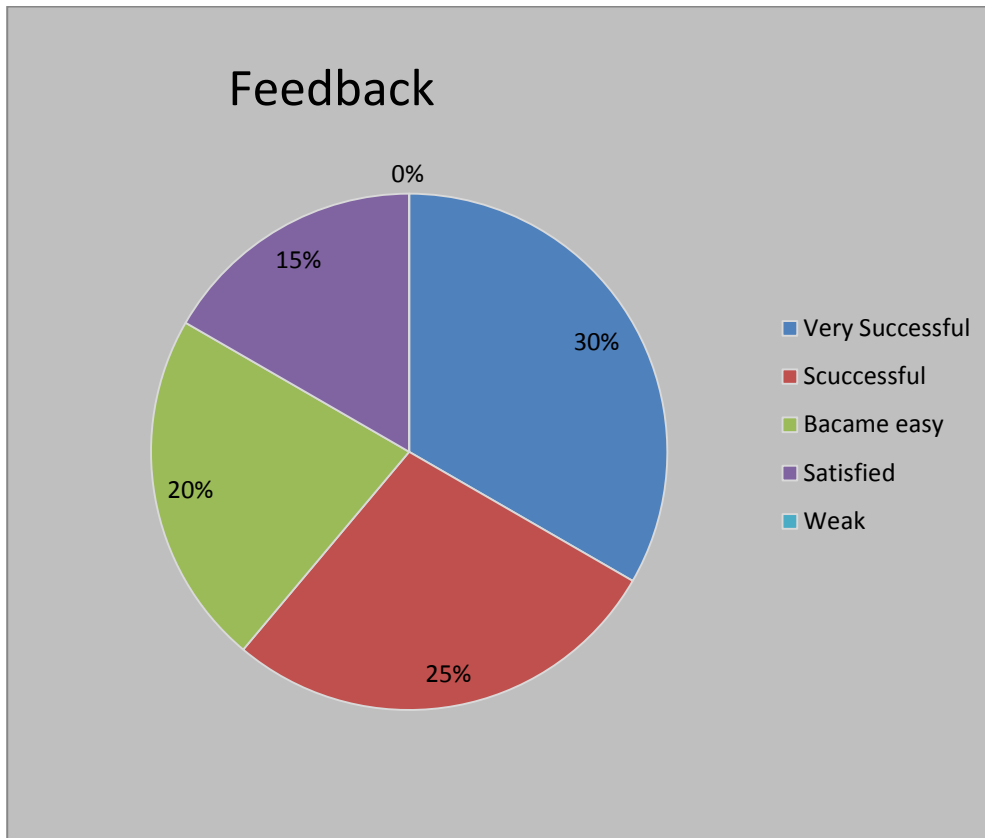


Figure 5.2 User Feedback Results

The overall results express that system work properly and users were satisfied about the system. According to the feedbacks the system proves that it can be managed easily and user-friendly environment can be made for the users.

Some bugs were pointed out during this process and they were revised and also few features were changed according to the user suggestions for better performance.

Finally, user feedback has determined that the system was implemented successfully with all the objectives and user satisfaction.

CHAPTER 6 - CONCLUSION

The “The Human Resource Management System” was tailor made software, design for St. Nicholas’ International College which is developing and a well-known Catholic International College in and around Negombo area. The main objective of this system is to provide an automated procedure to manage the day to day human resources activities of the school which were handled manually.

All the functionalities related with teacher information, subject allocation, leave management, and evaluation performance were accomplished in this project. Additionally, relevant reports that support the management were provided. User can do their day to day activities easily, efficiently and correctly using this system.

System was developed according to the MVC architecture and Object Oriented techniques were used. Together with the suitable procedures, techniques and tools were used to deliver this project.

Since the system achieved all the proposed functional and nonfunctional requirements along with some additional features, it can be measured as a successful project. Furthermore, the feedbacks given by end users establish that the system favorably satisfies the user and their requirements.

6.1 Lessons Learnt

From this project myself has got a wide knowledge about project, software engineering and many other areas. How to apply learned theories in practical scenario, what are the problems occurred when applying theories and how they were solved were learned in this project. As a student this is a good opportunity to get some practical experience about developing a system. Knowledge about data gathering, analysis and designing suitable solutions for problem domain was acquired.

Sound level of technical knowledge was collected by implementing this project in areas such as; MVC architecture, Java, JavaFX, Hibernate, MySQL, iReports, etc.

Additionally, the skills like finding appropriate knowledge sources were developed. Together with building some relationships with real world software engineers and professionals was accomplished through this project.

This project work has helped, not only to improve the knowledge but also it assists to improve many other skills.

6.2 Future Improvement

In future the following features are planned to be add to the newly built system as further

Improvements:

- Web based system which can be used to interact with staff members from the Colombo branch.
- System can be enhanced by providing online leave apply system.
- Android based system to insert daily updates into the system

REFERENCE

- [1] Wikipedia - The free encyclopedia – Requirements analysis
http://en.wikipedia.org/wiki/Requirements_analysis [Last Accessed 23rd June, 2017]
- [2] Wikipedia - The free encyclopedia – Functional requirement
http://en.wikipedia.org/wiki/Functional_requirement [Last Accessed 23rd June, 2017]
- [3] Behance site
<https://www.behance.net/HR-SOFTWARE-USER-INTERFACE/11869407>
[Last Accessed 27th June, 2017]
- [4] Wikipedia - The free encyclopedia – Unified Process
http://en.wikipedia.org/wiki/Unified_Process [Last Accessed 02nd July, 2017]
- [5] Tutorials point
http://www.tutorialspoint.com/software_engineering/software_design_basics.htm
[Last Accessed 02nd July, 2017]
- [6] Wikipedia - The free encyclopedia – Use Case Diagram
https://en.wikipedia.org/wiki/Use_Case_Diagram [Last Accessed 03rd July, 2017]
- [7] Visual paradigm – Visual paradigm diagrams
<http://www.visual-paradigm.com/VPGallery/diagrams/Class.html>
[Last Accessed 05th July,2017]
- [8] Wikipedia - The free encyclopedia - Sequence diagram
https://en.wikipedia.org/wiki/Sequence_diagram [Last Accessed 05th July, 2017]
- [9] Datanamic - The free encyclopedia – DB modeling
<http://www.datanamic.com/support/lt-dez005-introduction- dbmodeling.html>
[Last Accessed 07th July 2017]
- [10] Wikipedia - The free encyclopedia – Relationship model
https://en.wikipedia.org/wiki/Entity/E2/80/93relationship_model
[Last Accessed 09th July 2017]
- [11] http://www.bestpractices.osi.ca.gov/system_development/development.shtml
[Last Accessed 15th July 2017]
- [12] <http://www.webopedia.com/TERM/J/Java.html> [Last Accessed 20th July 2017]
- [13] <http://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilderinfo-2157684.html> [Last Accessed 22nd July 2017]
- [14] <http://www.tutorialspoint.com/hibernate> [Last Accessed 24th July 2017]

[15] <http://whatis.techtarget.com/definition/business-logic>

[Last Accessed 10th August 2017]

[16] <https://github.com/FrontendMasters/front-endhandbook/blob/maslearning/test.md>

[Last Accessed 15th August 2017]

[17] <https://github.com/FrontendMasters/front-endhandbook/blob/master/learning/test.md>

[Last Accessed 15th August 2017]

[18] <http://www.aptest.com/testtypes.html> [Last Accessed 16th August 2017]

[19] https://wiki.openoffice.org/wiki/QA/Testcase/How_to_write_test_case.

[Last Accessed 20th August 2017]

APPENDIX

APPENDIX – A

System Documentation

This documentation provides guidelines for preceding activities to the setup of the Human Resource Management System. This is to sustain the users to install this System in their PC's, Servers or Workstations. The system documentation can be referred if there are any changes to be made in the System.

Hardware & software configuration requirements,

Hardware Requirements:

- Personal Computer with processing power similar or higher than 3.06 GHz
- RAM with 4 GB or above capacity.
- Hard Disk with 250 GB or above capacity.
- Printer

Software Requirements:

- MySQL Server 5.5
- NetBeans IDE 8.2
- Windows 8.1

A.1 SOFTWARE INSTALLATION AND CONFIGURATION

A.1.1 Install Java Run Time on Client Machine



Figure A. 1: Installation progress of JVM (Step1)



Figure A. 2: Installation progress of JVM (Step 2)

Select Optional features and click next

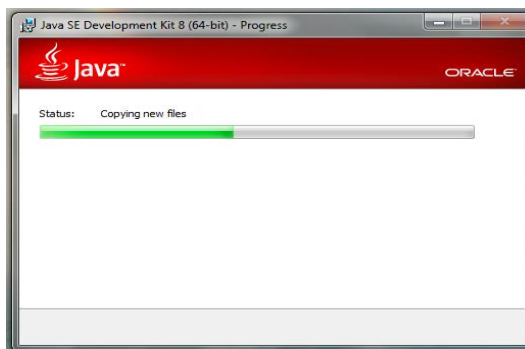


Figure A. 3: Installation progress of JVM (Step 3)

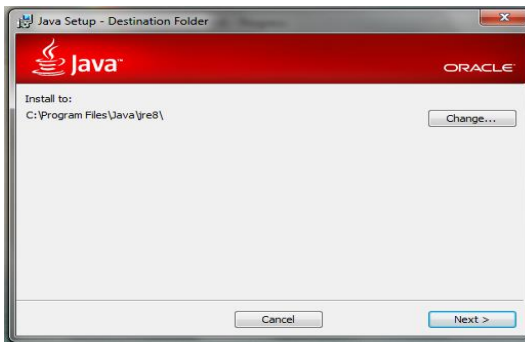


Figure A. 4: Installation progress of JVM (Step 4)



Figure A. 6: Installation progress of JVM (step 5)



Figure A. 5: Installation progress of JVM (step 6)

"click Finish to complete the installation"

A.1.2 Installing My SQL Server 5.5

Step 1: Download MySQL Community Server 5.5 installation file appropriate for the platform. Open installation file for MySQL Community Server and press “Next”.



Figure A. 7: Installation Progress MYSQL server (Step 1)

Step 2: Choose “Typical” setup type, click “Next” and “Install”.



Figure A. 8: Installation Progress MYSQL server (Step 2)

Step 3: Choose “Typical” setup type, click “Next” and “Install”.

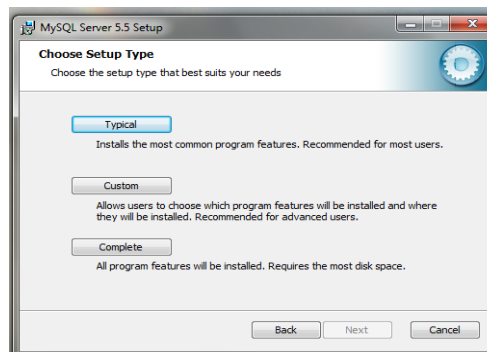


Figure A. 9: Installation Progress MYSQL server (Step 3)

Step 4: Click “Install” to begin the installation.

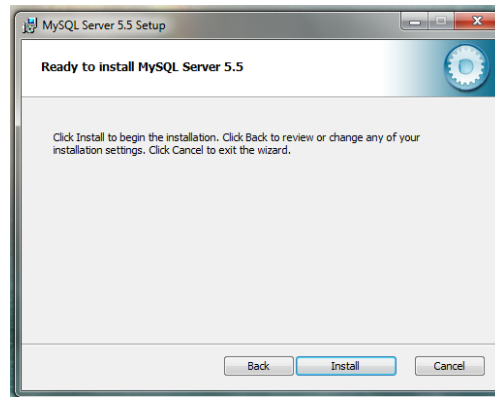


Figure A. 10: Installation Progress MYSQL server (Step 4)

Step 5: After installation process is completed, check “Launch the MySQL Instance Configuration Wizard” and click “Finish”.

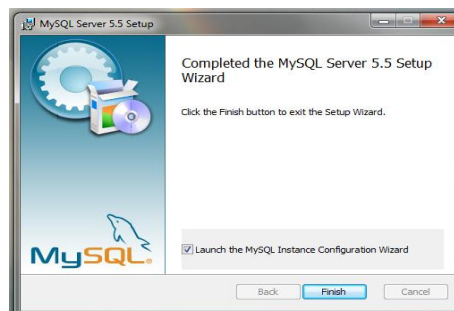


Figure A. 11: Installation Progress MYSQL server (Step 5)

Step 6: Click “Next” when “Configuration Wizard” appears.

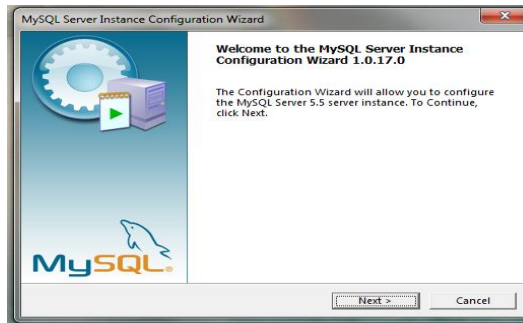


Figure A. 12: Installation Progress MYSQL server (Step 6)

Step 7: Choose “Standard Configuration” and click “Next”.

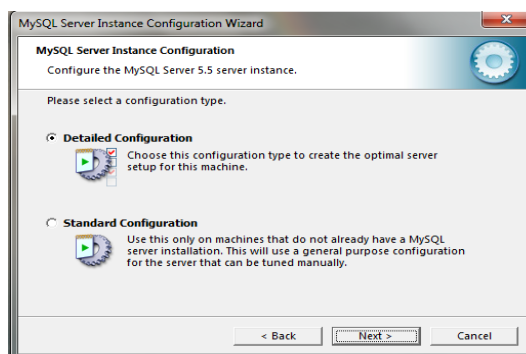


Figure A. 13: Installation Progress MYSQL Server (Step 7)

Step 8: Check “Install as Windows Service”, select service name “MySQL”. Check “Launch the MySQL Server automatically”. Then check “Include Bin Directory in Windows PATH” and click “Next”.

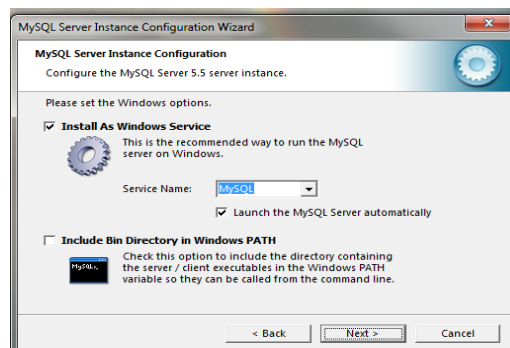


Figure A. 14 Installation Progress MYSQL Server (Step 8)

Step 9: Check “Modify Security Settings”, Set a new password to the “root” user. Check “Enable root access from remote machines”. Click “Next” and then “Execute”.



Figure A. 15: Installation Progress MYSQL Server (Step 9)

Step 10: After configuration process is completed click “Finish”.

A.1.3 Installing My SQL Server 5.5

Step 1: Download MySQL Community Server 5.5 installation file appropriate for the platform. Open installation file for MySQL Community Server and press “Next”.

Step 2: Choose “Typical” setup type, click “Next” and “Install”.



Figure A. 16: Installation Progress MYSQL server (Step 1)

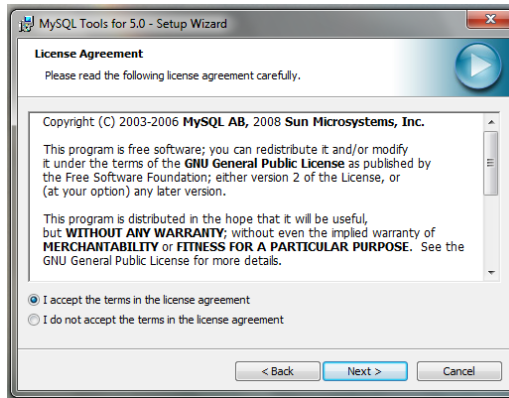


Figure A. 17: Installation Progress MYSQL server (Step -2) –Accept the License agreement

Step 3: Choose “Typical” setup type, click “Next” and “Install”.

Step 4: Click “Install” to begin the installation.

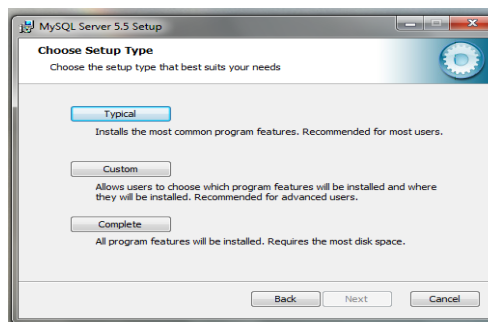


Figure A. 18: Installation Progress MYSQL server (Step 3)

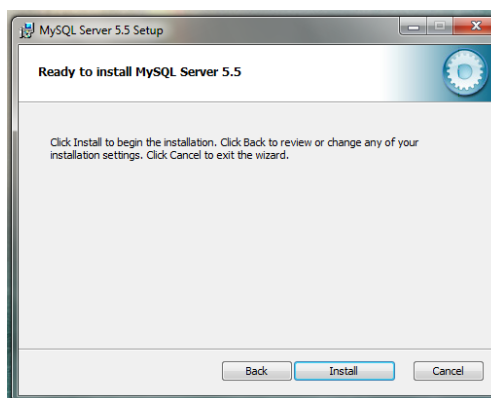


Figure A. 19: Installation Progress MUQL server (Step 4)

Step 5: After installation process is completed, check “Launch the MySQL Instance

Configuration Wizard” and click “Finish”.

Step 6: Click “Next” when “Configuration Wizard” appears.

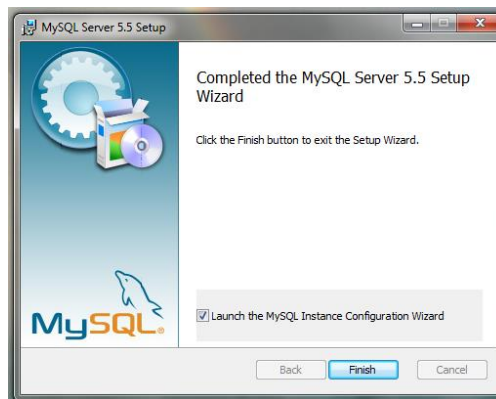


Figure A. 20: Installation Progress MYSQL server (Step 5)

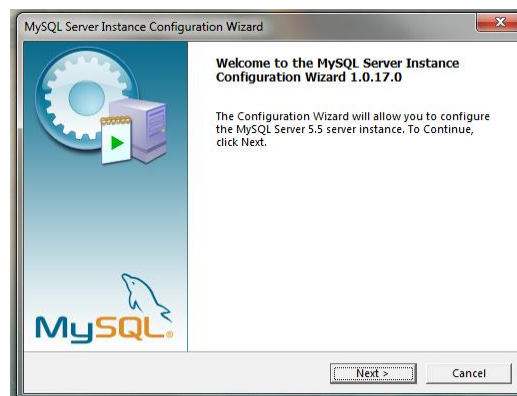


Figure A. 21: Installation Progress MYSQL server (Step 6)

Step 7: Choose “Standard Configuration” and click “Next”.

Step 8: Check “Install as Windows Service”, select service name “MySQL”. Check “Launch the MySQL Server automatically”. Then check “Include Bin Directory in Windows PATH” and click “Next”.

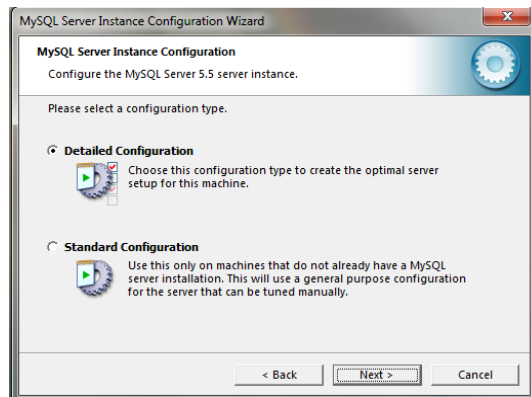


Figure A. 22: Installation Progress MYSQL Server (Step 7)

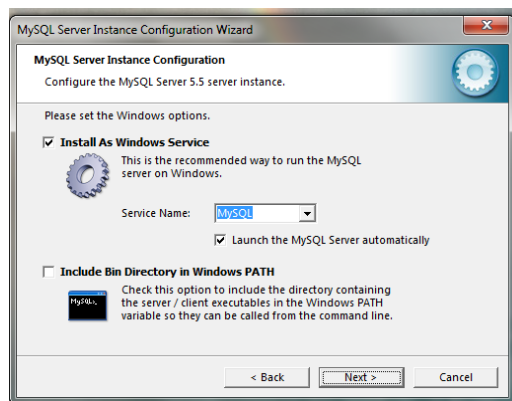


Figure A. 23: Installation Progress MYSQL Server (Step 8)

Step 9: Check “Modify Security Settings”, Set a new password to the “root” user. Check “Enable root access from remote machines”. Click “Next” and then “Execute”.

Step 10: After configuration process is completed click “Finish”.

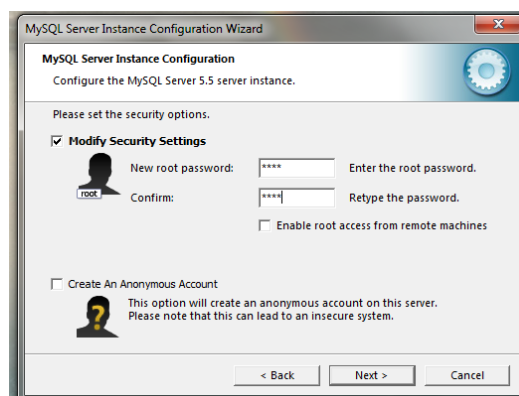


Figure A. 24: Installation Progress MYSQL Server (Step 9)

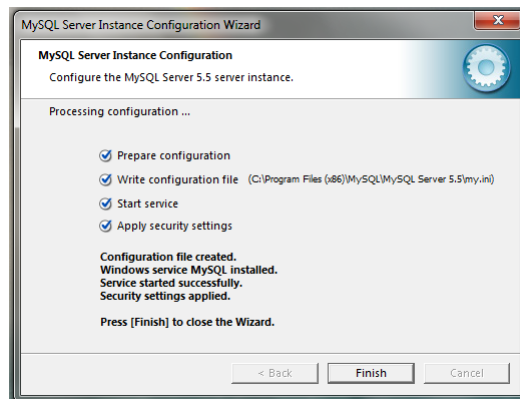


Figure A. 25: Installation Progress MYSQL Server (Step 10)

A.1.4 Installing Microsoft SQL Query Browser



Figure A. 26: Installation Progress MYSQL Query Browser (Step 1)

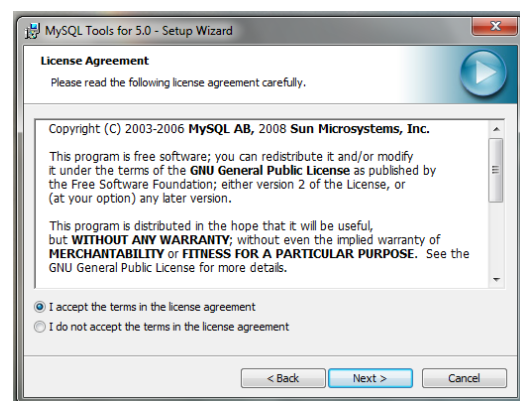


Figure A. 27: Installation Progress MYSQL Query Browser (Step 2)

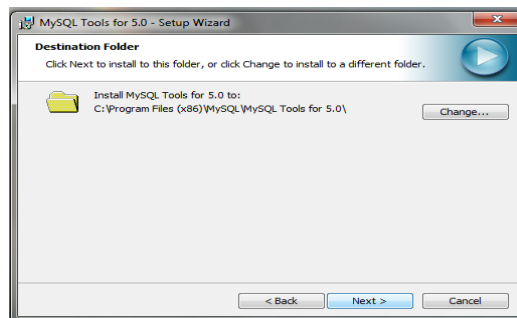


Figure A. 28: Installation Progress MYSQL Query Browser (Step 3)

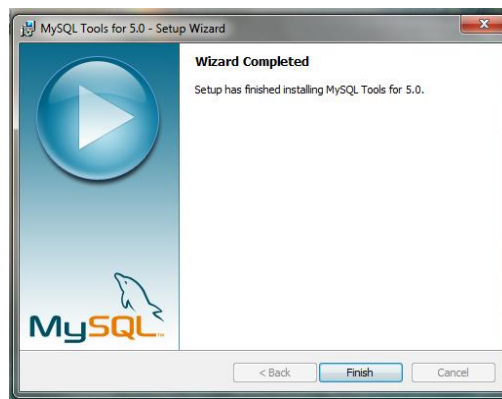


Figure A. 29: Installation Progress MYSQL Query Browser (Step 4)

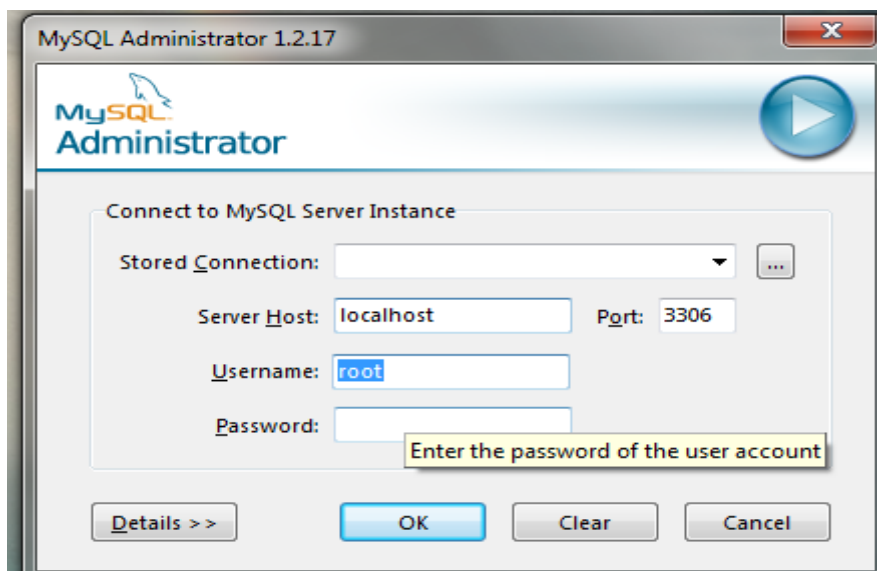


Figure A. 30: Open MYSQL Administrator & Insert server host, port & the user name as given above & enter the given password (Step 5).

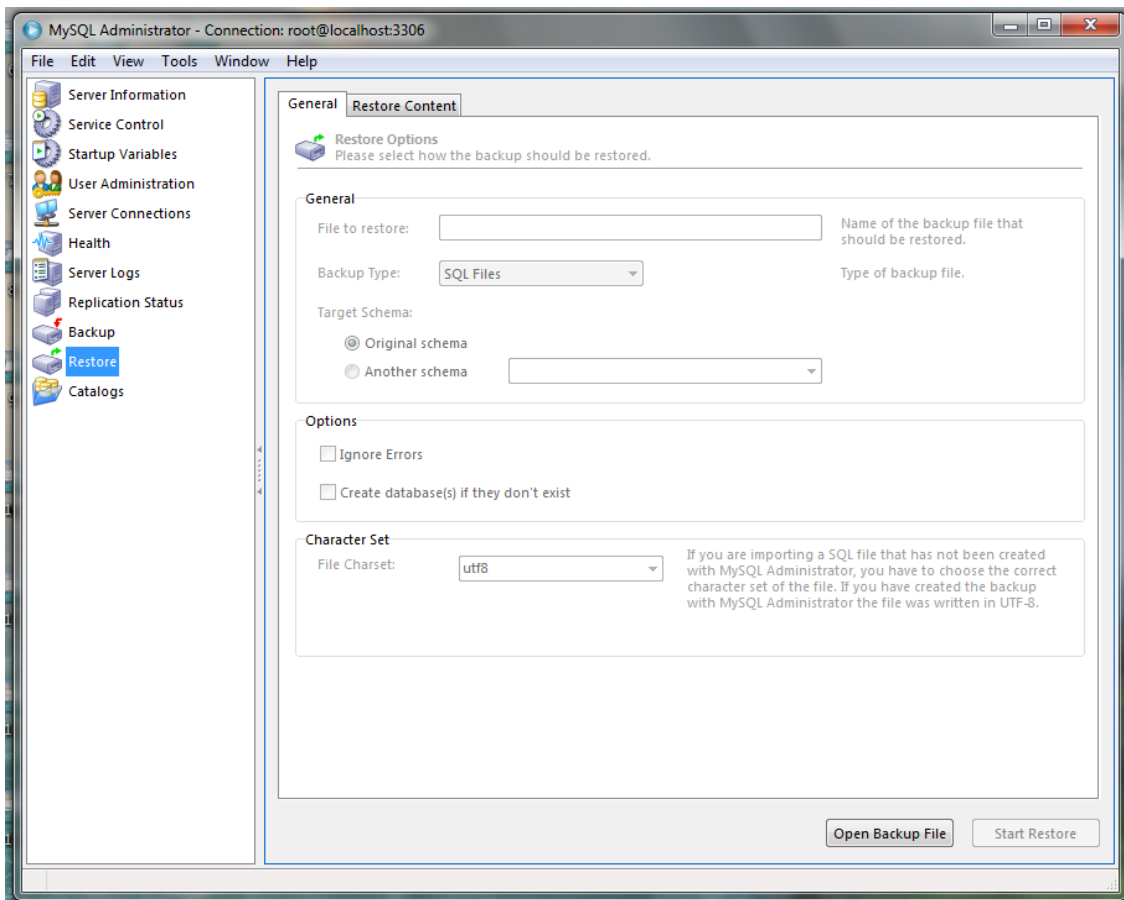


Figure A. 31: To restore the database create a new database the select restore database & last set database to system database (Step6)

A.1.5 Human Resources Management System

After setting up the database,

Select from device

Click Add &

Browse the **Human Resources Management System** file from CD

Run the setup.exe file located in CD.

APPENDIX – B

B.1 Design Documentation

B.1.1 Activity Diagram

Activity diagram for Insert new user is given in Figure B.1

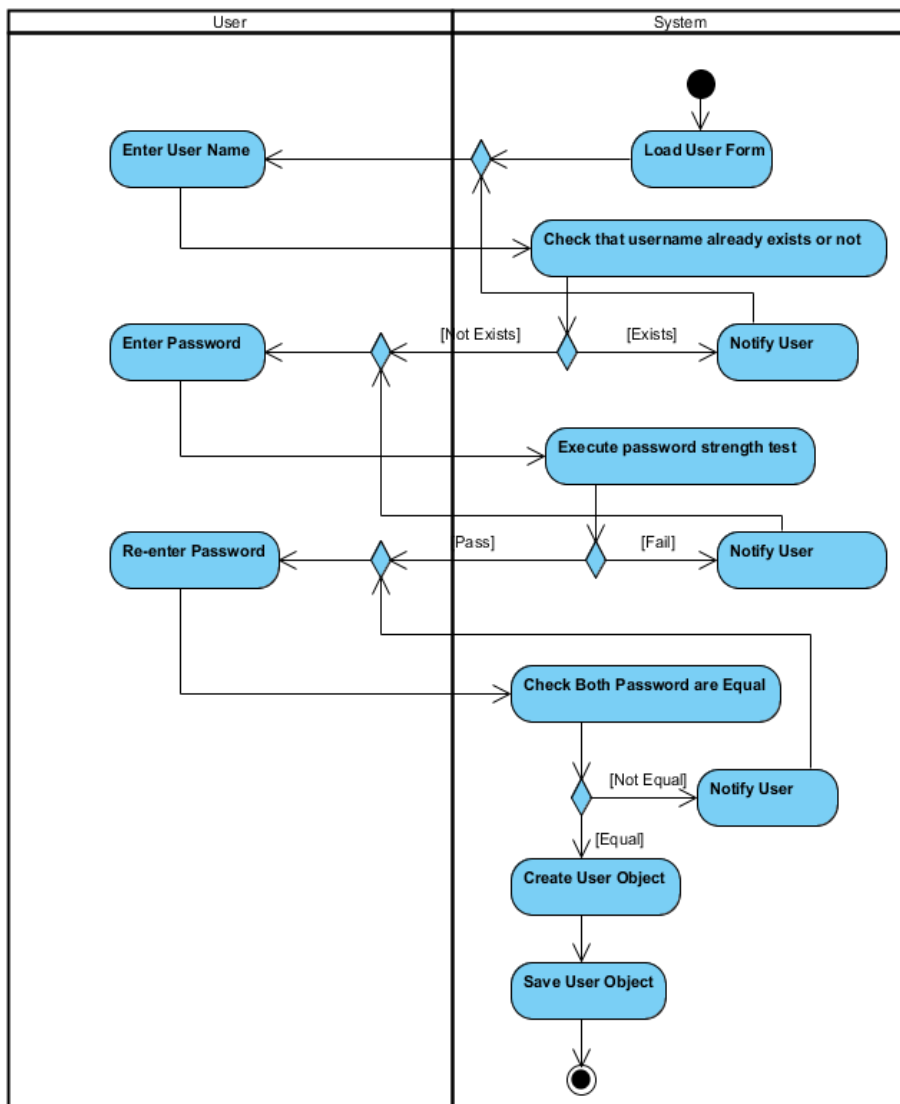


Figure B. 1: Activity Diagram for Insert New User

Activity Diagram for Apply leave is given in Figure B.2

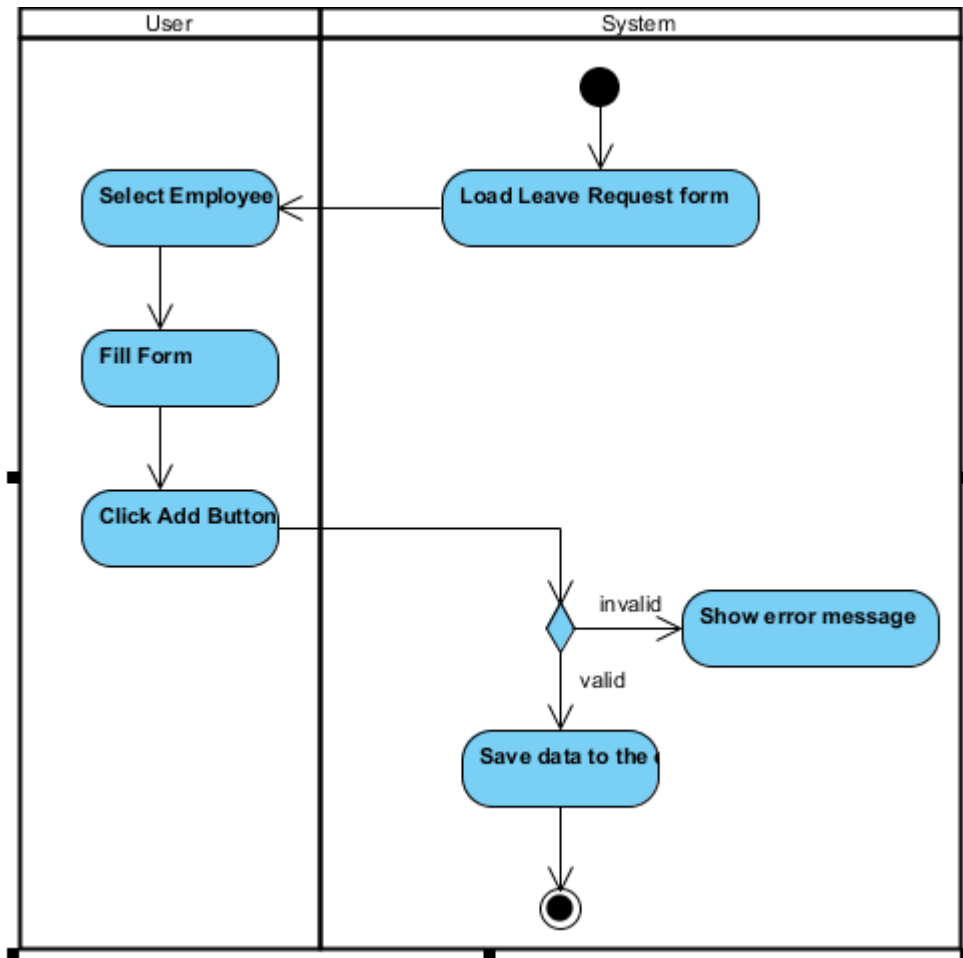


Figure B. 2 : Activity Diagram for Apply Leave

B.1.2 Sequence diagram

Sequence diagram for Report Generation of the system is given in Figure B.3

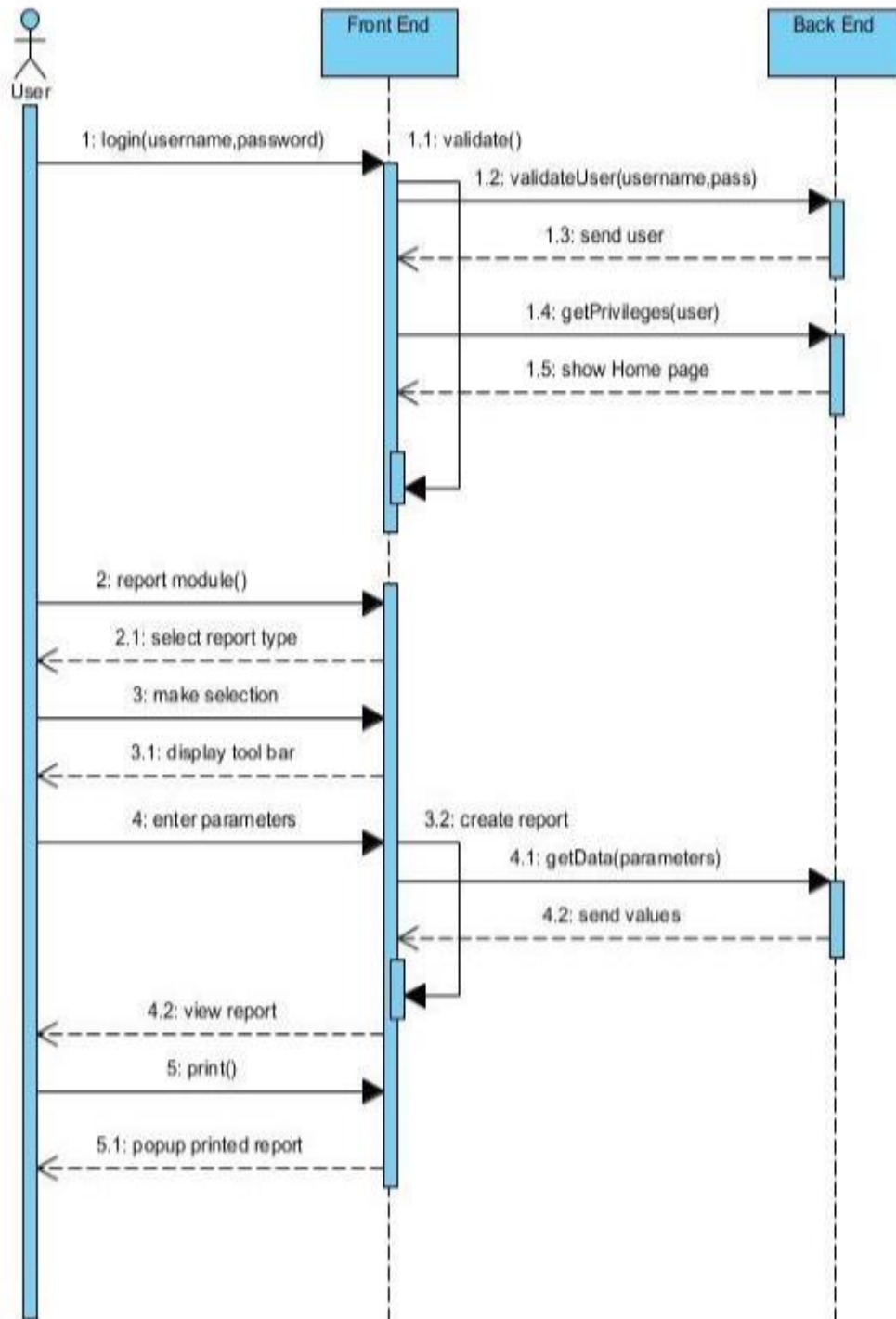


Figure B. 3: Sequence Diagram for Report generation of the system

B.1.3 Use Case diagram

Figure B.4 shows a use case diagram for Principal Requirements

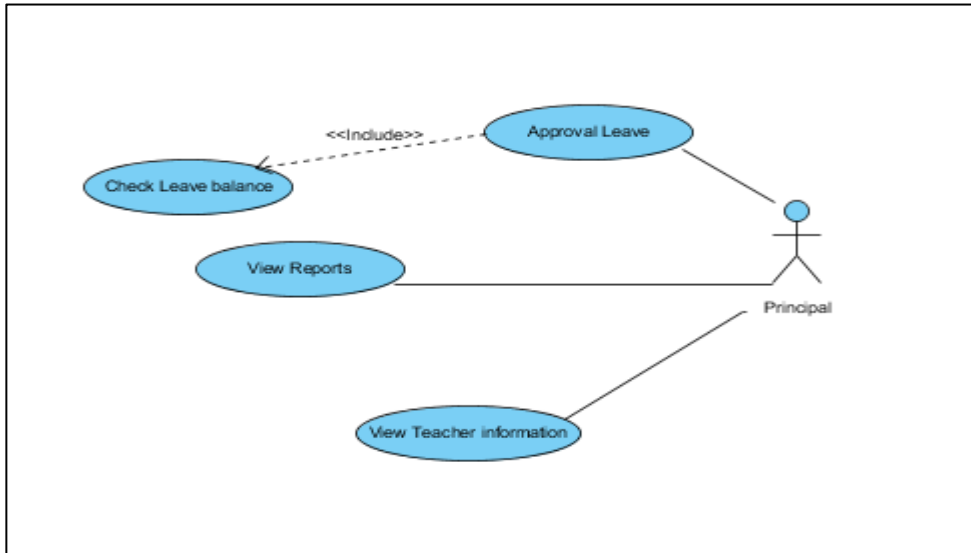


Figure B. 4 Use case diagram for Principal Requirements

Figure B.5 shows a use case diagram for Human Resource Manager Requirement

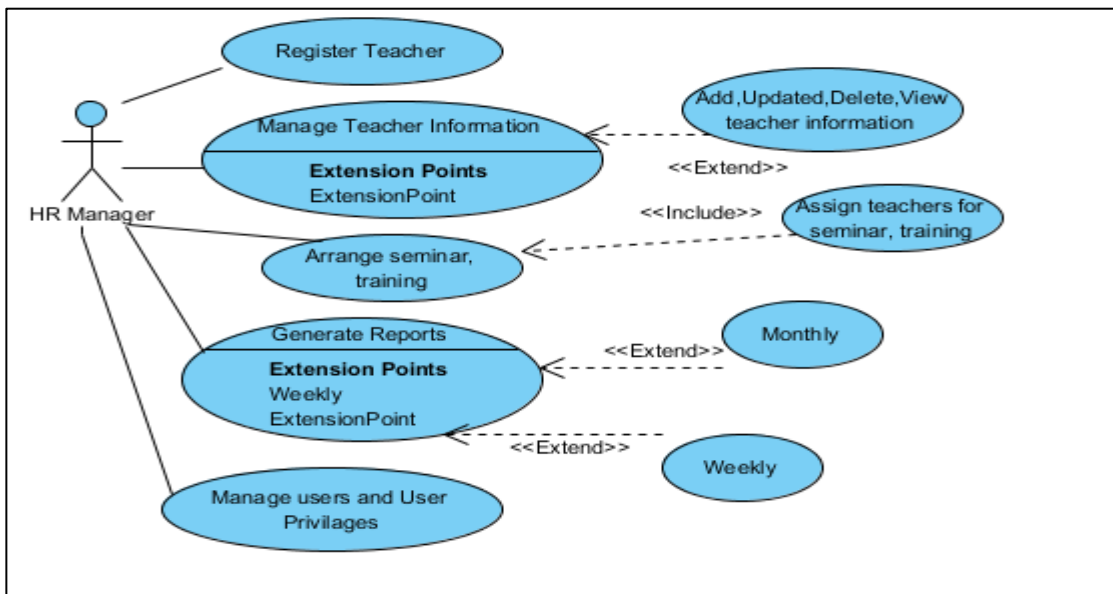


Figure B. 5 Use case diagram for Human Resource Manager Requirement

Use case description Add user is given in Table B.1

Use case	Add User
Actors	Administrator
Description	Insert a new user
Precondition	<ul style="list-style-type: none"> I. User must login to the system II. Load all Employees from the database III. Load all Users from the database
Flow of event	<ul style="list-style-type: none"> I. Select an employee II. Enter an username III. Check whether the entered username already exists IV. If exists notify user V. Enter the password VI. Execute password strength test VII. If test fail notify user VIII. Re-enter the password IX. Check whether entered passwords are matching or not X. Press save button XI. Show confirmation box, if user confirm a new will add to the system
Post condition	New user will be added to the system

Table B. 1 Add User

Use Case Description for Approval of Leave is given in Table B.2

Use Case	Approval Leave
Actors	Principal
Description	Approval leave
Precondition	<ul style="list-style-type: none"> Principal log in to the system Pending leaves will be displayed
Flow of event	<ul style="list-style-type: none"> Check whether the leave has applied in correct way and then Click Approve button if no, Click Reject button and give comments for reject leave
Post Condition	Approval of leave

Table B. 2 Approval of leave

B.2 Database Design

Figure B.6 shows the structure of Teacher Table in database

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
epf_no	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
name	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
dob	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
nic	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
address	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
mobile	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
land	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
email	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
appointment_date	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
image	BLOB	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
experience	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
gender_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
civilstatus_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
teacher_status_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
designation_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 6Teacher Table

Figure B.7 shows the structure of Civil Status Table in database



Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 name	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 7 Civil Status Table

Figure B.8 shows the structure of Designation Table in database



Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 name	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 8 Designation Table

Figure B.9 shows the structure of Gender Table in database



Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 name	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 9 Gender Table

Figure B.10 shows the structure of Qualification Table in database




Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 name	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 qualification_type_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 10 Qualification Table

Figure B.11 shows the structure of Qualification Type Table in database



Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 name	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 11 Qualification Type Table

Figure B.12 shows the structure of Salary Table in database




Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 amount	DECIMAL(10,3)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 qualification_type_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 12 Salary Table

Figure B.13 shows the structure of Teacher Salary Table in database








Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 increment	DECIMAL(10,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 effective_date	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 salary_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 teacher_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 basicsalary	DECIMAL(10,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 initial	BOOLEAN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 13 Teacher Salary Table

Figure B.14 shows the structure of Leaves Table in database














Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 name	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 requestdate	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 description	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 startdate	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 enddate	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 count	DECIMAL(5,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 approveddate	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 approvedby	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 teacher_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 leavestatus_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 leavetype_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 leavecategory_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 14 Leaves Table

Figure B.15 shows the structure of Leave Type Table in database

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
🔑 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
🔗 name	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 15 Leave Type Table

Figure B.16 shows the structure of Leave Statues Table in database

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
🔑 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
🔗 name	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 16 Leave Statues Table

Figure B.17 shows the structure of Leave Category Table in database

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
🔑 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
🔗 name	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 17 Leave Category Table

Figure B.18 shows the structure of Section Table in database

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
🔑 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
🔗 name	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 18 Section Table

Figure B.19 shows the structure of Section Grade Table in database

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
🔗 section_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
🔗 grade_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
🔑 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figure B. 19 Section Grade Table

Figure B.20 shows the structure of Subject Table in database

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
🔑 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
◇ name	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 20 Subject Table

Figure B.21 shows the structure of User Table in database

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
🔑 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
◇ username	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
◇ password	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
◇ teacher_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 21 User Table

Figure B.22 shows the structure of User Privilege in database

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
◇ privilege_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
🔑 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
◇ designation_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 22 User Privilege Table

Figure B.23 shows the structure of Module in database

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
🔑 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
◇ value	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 23 Module Table

Figure B.24 shows the structure of Performance in database

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
🔑 id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
◇ correctionofbooks	BOOLEAN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
◇ syllabuscovered	BOOLEAN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
◇ weakstudents	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
◇ increment	DECIMAL(10,2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
◇ comments	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
◇ year	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure B. 24 Performance Table

APPENDIX – C

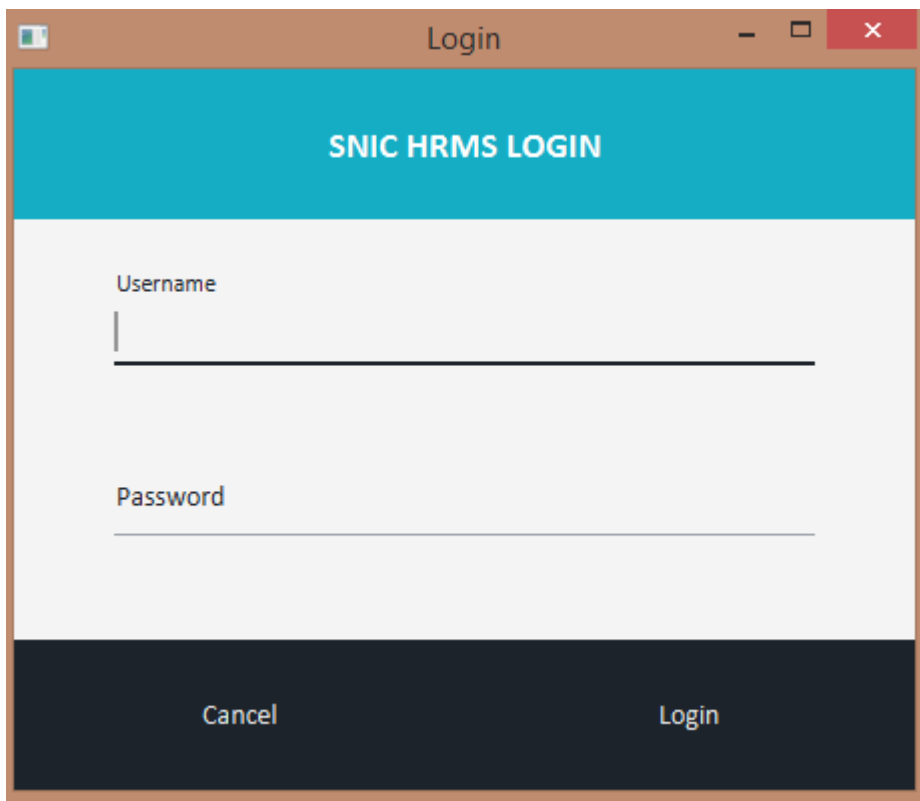
USER DOCUMENTATION

This section describes how the user operates this system.

How to login to the system

- Start the program.

Following window will appear.



The image shows a screenshot of a web browser window titled "Login". The window has a teal header bar with the text "SNIC HRMS LOGIN" in white. Below the header, there are two input fields: "Username" and "Password". The "Username" field is currently empty, and the "Password" field is also empty. At the bottom of the window, there are two buttons: "Cancel" and "Login". The window's title bar is brown and contains the text "Login" and standard window control icons (minimize, maximize, close).

Figure C. 1 Login Window

How Register Employee to the system

- Click on Employee menu

The following window will appear

CREATE EMPLOYEE

Browse
Clear

EPF NUMBER - 21

Name

Date of birth

NIC Number

Gender : Male Female

Civil Status : Single Married

Status

Appointment Date

Designation

Mobile

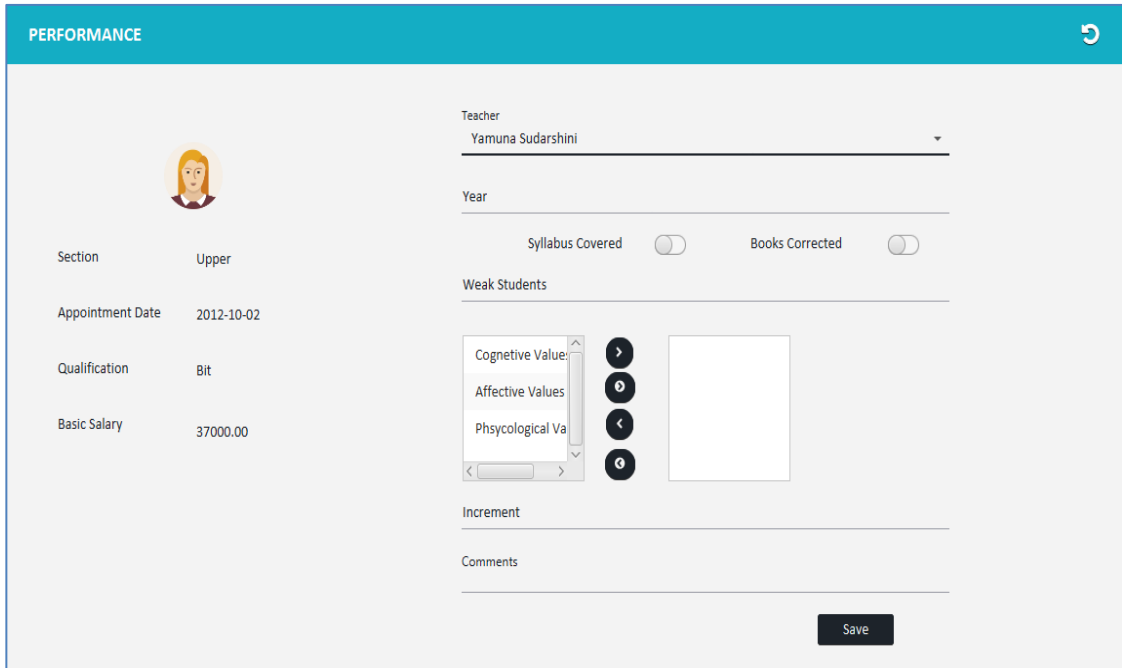
Figure C. 2 Create Employee Window

- This will provide facility to register employee to the system.
- After registration of the employee can proceed all other modules.

How evaluate the performance of the teacher

- Click on Performance menu

The following window will appear



The screenshot shows a web interface titled "PERFORMANCE" with a refresh icon in the top right corner. On the left, there is a profile card for a teacher with a placeholder icon. The card lists the following details:

Section	Upper
Appointment Date	2012-10-02
Qualification	Bit
Basic Salary	37000.00

On the right side of the form, there are several input fields and controls:

- Teacher:** A dropdown menu showing "Yamuna Sudarshini".
- Year:** A text input field.
- Syllabus Covered:** A toggle switch.
- Books Corrected:** A toggle switch.
- Weak Students:** A section containing a list of values: "Cognitive Value:", "Affective Values", and "Phycological Va". To the right of this list are four circular buttons with arrows (right, down, left, up) and a large empty rectangular box.
- Increment:** A text input field.
- Comments:** A text input field.
- Save:** A dark button at the bottom right.

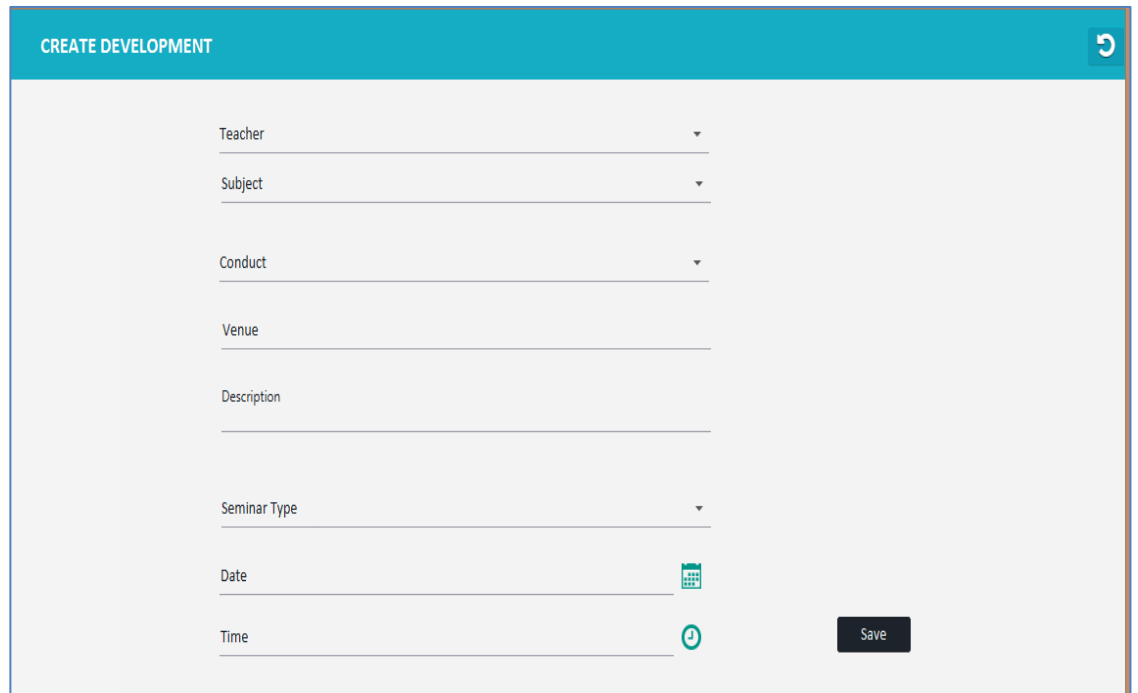
Figure C. 3 Performance Window

- This will provide the evaluate performance of the teacher.
- So it helps to decide the increment for the teacher in each academic year.

How assign seminar / training for the teacher

- Click on Development menu

The following window will appear



The screenshot shows a web form titled "CREATE DEVELOPMENT". The form contains the following fields:

- Teacher (dropdown menu)
- Subject (dropdown menu)
- Conduct (dropdown menu)
- Venue (text input)
- Description (text input)
- Seminar Type (dropdown menu)
- Date (calendar icon)
- Time (clock icon)

A "Save" button is located at the bottom right of the form.

Figure C. 4 Development Window

- This will provide the assign seminar / training of the teacher.

APPENDIX – D

MANAGEMENT REPORT

The system allows users such as Principal, Vice Principal HR & Head Teacher to generate various types of reports in order to use for the decision making process of the school. As system gives daily, monthly & yearly reports after manipulating them, management can use them to analyze & identify the trends, patterns & seasonal variations of the school.

Figure D.1 shows the report for Teachers Performance according to the year

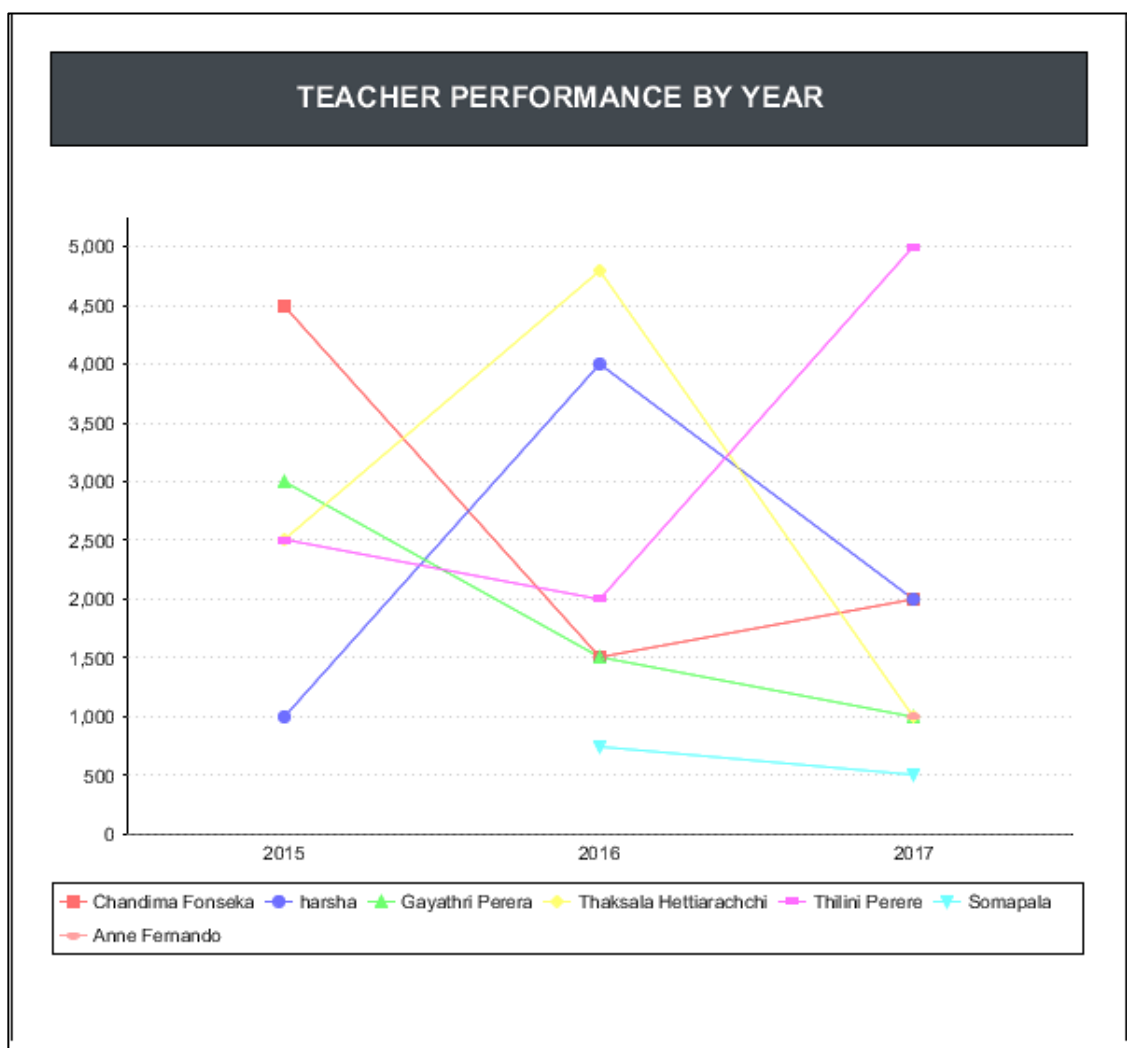


Figure D. 1 Teachers Performance according to the year

Figure D.2 shows the report for Seminar Details By Teacher

SEMINAR DETAILS BY TEACHER						
Name : Anjalo Perera						
EPF No : 17						
Date	Time	Venue	Conducted By	Section	Subject	Type
13/12/2017	10:58:00	Junior School	School	Upper	English	Out Bound
11/01/2018	09:00:00	Renuka Hotel	Cambridge	Upper	Physics	Face To Face
13/12/2017	10:58:00	British Council	British Council	Upper	English	Face To Face

Figure D. 2 Seminar Details By Teacher

APPENDIX-E

TEST RESULTS

E.1 Test case and test results for Create Employee module

Test No	Test Description	Expected Result	Pass / Fail
1	When required fields are not entered	There will be red tick marks and notice to enter the required fields.	Pass
2	Enter invalid Name	There will be red tick mark and notice to re-enter.	Pass
3	Enter invalid NIC	There will be red tick mark and notice to re-enter.	Pass
4	Enter Mobile number or Land number without 10 digits	There will be red tick mark and notice to re-enter.	Pass
5	Select Academic qualification List	Academic qualification list will be appeared and user can select qualification	Pass
6	Select Designation combo box	Designation list will be appeared and user can select a designation	Pass
7	Enter invalid Date of Birth	There will be red tick mark and notice to re-enter.	Pass
8	Enter the Email address in invalid format	There will be red tick mark and notice to re-enter. "Email address should be in correct format, It should have "@" symbol and end with "." and 2 or 3 characters.	Pass
9	Select Status combo box	Status list will be appeared and user can select a status	Pass
10	Enter invalid Appointment Date	There will be red tick mark and notice to re-enter.	Pass
11	Enter invalid Address	There will be red tick mark and notice to re-enter.	Pass
12	Enter all information correctly and click Save	Confirm Message box will be shown as "Do you want to save? If click "YES" button notification will be shown as "Employee Saved Successfully	Pass

Table E. 1 Test Results for Create Employee Module

E.2 Test case and test results for Apply Leave module

Test No	Test Description	Expected Result	Pass / Fail
1	When required fields are not entered	There will be red tick marks and notice to enter the required fields.	Pass
2	Select Leave Type from the combo box	Selected leave type will be loaded in to the combo box	Pass
3	Select leave category form radio button	Selected radio button will be selected	Pass
4	Select from date	Selected date will be display	Pass
5	Select to date	The selected date should be future date for the from date. If it is a future date, green color tick mark will be displayed. If not red color tick mark will be displayed.	Pass
6	Leave count will be calculated automatically	It any wrong date has selected red color tick mark will be displayed next to the leave count.	Pass
7	Enter valid reason	There will be green color tick mark next to the reason.	Pass
8	Enter all information correctly and click Save	Confirm Message box will be shown as "Do you want to save? If click "YES" button notification will be shown as "Leave Saved Successfully. Otherwise Save will be cancelled.	Pass

Table E 2 Test results for Apply Leave module

E.3 Test case and test results for Approval Leave module

Test No	Test Description	Expected Result	Pass / Fail
1	When required fields are not entered	There will be red tick marks and notice to enter the required fields.	Pass
2	Without selecting an applicant of leave from the dialog box	Whole applicants will be displayed in the table	Pass
3	Select an applicant of leave	Selected applicants details will be displayed	Pass
4	Select from date	Selected date will be display	Pass
5	Select to date	The selected date should be future date for the from date. If it is a future date, green color tick mark will be displayed. If not red color tick mark will be displayed.	Pass
6	Click Search button	According to the selected applicant details will be displayed in the table	Pass
7	Click Approve button	Selected applicant's leave will be approved and that record will be disappeared form the table	Pass
8	Click Reject button	Selected applicant's leave will be rejected and should enter reject reason in the dialog box	Pass

Table E 3 Test results for Approval Leave module

APPENDIX F

CODE LISTING

F.1 Code for Login

Code F.1 Login Controller.java

```
public class LoginController implements Initializable {

    @FXML
    private TextField txtUsername;
    @FXML
    private PasswordField txtPassword;
    @FXML
    private Button btnLogin;
    @FXML
    private JFXButton btnExit;

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        formClear();
    }

    @FXML
    private void btnLoginOA(ActionEvent event) {
        if (txtPassword.getText().length() > 0 && txtUsername.getText().length() > 0) {
            String username = txtUsername.getText();
            String password = txtPassword.getText();
            if (new LoginHandler().Validate(username, password)) {
                closeLogin();
                StageLoader.LoadStage("/com/snic/ui/Welcome.fxml", Modality.NONE, StageStyle.DECORATED, "", true);
            } else {
                Alert a = new Alert(Alert.AlertType.ERROR, "Not a valid user");
                a.show();
                System.out.println("Not a valid user");
            }
        }
    }
}
```



```
private void closeLogin() {
    Stage stage = (Stage) btnLogin.getScene().getWindow();
    stage.hide();
}

@FXML
private void btnExitOA(ActionEvent event) {
    System.exit(0);
}

private boolean formClear() {
    txtPassword.setText("");
    txtUsername.setText("");
    return true;
}
}
```

APPENDIX G

CLIENT CERTIFICATE



	ST. NICHOLAS' INTERNATIONAL COLLEGE Katana Road, Nelumpitiya, Negombo, Sri Lanka.
30 th October, 2017	
BIT Coordinator, External Degree Centre, No.17, Swarna Road, Colombo 6.	
Dear Sir/Madam,	
<u>CERTIFICATION FOR HUMAN RESOURCE MANAGEMENT SYSTEM</u>	
We do hereby certify that Mrs. R. A. Y. S. Ranasinghe (Registration No.R111734) has successfully developed a Human Resource Management System for our college, St. Nicholas' International College, Negombo.	
We appreciate the attention that Mrs. Ranasinghe has shown towards her work, and the feasibility of implementing her system in our college. She had carefully studied every aspect of our school's administration and had successfully developed a system incorporating all the requested functionalities through hard work and dedication.	
Whilst complimenting Mrs. Ranasinghe for the task accomplished, we wish her for success in her academic endeavours.	
Yours faithfully,	
	
Fr. B. Anselm Shiran, Rector	
Rev. Fr. B. Anselm Shiran RECTOR St, Nicholas' International College Negombo Sri Lanka	
<i>"LEARNED SHINE LIKE STARS"</i>	
Tel : 0312221607, E-mail : principal@snic.lk, Web : www.snic.lk	

Figure G. 1 Client Certificate

GLOSSARY

Database - is an organized collection of data for one or more purposes, usually in digital form.

UML – refers to Unified Modeling Language is a standardized general-purpose modeling language in the field of object-oriented engineering. This includes a set of graphic notation techniques to create visual models of object-oriented software intensive systems.

Graphical User Interface - is a type of user interface that allows users to interact with electronic devices with images rather than text commands.

Relational database management system - database management system (DBMS) that is based on the relational model (that all data is represented in terms of tuples, grouped into relations.)

JavaFX- is a software platform for creating and delivering rich internet applications (RIAs are) that can run across a wide variety of devices

Object-relational mapping - is a programming technique for converting data between incompatible type systems in object-oriented programming languages.

INDEX

activity, 18, 19,72,73
analysis, 5, 56
configuration,25, 35,28,32,68
controller, 31, 38
design, 4, 10, 12, 13, 24, 30, 31, 33, 38, 45
diagram, 4, 13, 14, 19, 22, 24, 33, 66
documentation, 57
entity, 4, 12, 24, 26
error, 92
hibernate, iii, 34, 36, 41, 44, 54
implementation, 4, 10, 32, 42
integration, 46
interfaces, 13, 26, 38
inventory, i, iii, 12, 15, 16, 17, 19, 20, 21, 22, 24, 45, 53, 57, 65
jasper report, 35
java, 13, 31, 32, 34, 35, 36, 39, 57
javafx, iii, 30, 33, 38, 44, 54
methodology, 12, 13
netbeans, iii, 30, 31, 32, 57
object oriented, iii
relationship, 26
requirements, iii, 4, 5, 6, 7, 8, 12, 14, 26, 30, 33, 45, 46, 50, 53, 57
scope, 2, 3
security, 2, 53
sequence, 66
testing, 45, 46, 96
validation, 27, 28, 45, 46
visual paradigm, 30, 33