Web Based System for Future MINDS (AMI) Montessori School

S.S.W.Vidanapathirana

2016

Web Based System for Future MINDS (AMI) Montessori School

S.S.W.Vidanapathirana

BIT Registration No. R082102

Index number: 0821022

Name of the supervisor – Mr. Vidura Hettiarachchi

2016

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

Declaration

Declaration

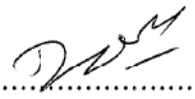
I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a Degree or Diploma in any University and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where 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 inter-library loans, and for the title and summary to be made available to outside organizations.

..... Date: 08./.09./.2016

S.S.W. Vidanapathirana

(Name of Candidate)

Countersigned by:



..... Date: 24./.09./.2016

Mr Vidura Hettiarachchi

(Name of Supervisor)

Abstract

Future MINDS (AMI) Montessori school is a leading English medium preschool which follows the Montessori Method of education .It has stepped forward significantly in a very short period of the time. Recently providing prior education for over 150 children and considerable number of staff. The internal process of the Montessori is properly managed, yet it needs of a system due to heavy paper work with inefficient and time consuming manual process.

The proposed web based system facilitates the current work process to be coordinated efficiently with lesser paper work. In fact added features convey external and internal users providing new user experiences with effective communication.

The web based system has been developed based on waterfall develop methodology. Open source server side scripting language, PHP is used for development with application of MVC – (Model – View – Controller) architecture based Code-Ignitor Framework. Further MySQL has been used as database tool, Net-Beans as IDE (Integrated Development Environment) and Apache web server to run the system.

Acknowledgment

I like to make this an opportunity to convey my gratitude to everyone who supported and encouraged me to the complete my project successfully.

First of all expenses my gratitude to BIT coordinators of the UCSC (university of Colombo School of Computing) for giving as valuable design program to follow up , providing guidance.

I am highly indicated to my project supervisor Mr Vidura Hettiarachchi for his guidance and consultant supervisor of my project despite his busy schedule.

A special gratitude I give to Mr Amindra De Silva, Mr Inshar Anas , Dr Viraj Pinto who encouraged me throughout the BIT degree program as well as kind support and guidance to successfully completion of the project .

Further mean I expect my worm thanks to Ms. Gayathri Wickramasekara Managing Directress of future MINDS Montessori School, my staff mates and my friends for giving me support to complete my final year project.

Table of Contents

List of Figures

List of Tables

List of Acronyms

AMI - Association Montessori Internationale

AJAX - Asynchronous JavaScript and XML

DBMS - Database Management System

ERD - Entity Relationship Diagram

IDE - Integrated Development Environment.

MVC - Model-View-Controller

OO - Object Oriented

OOD - Object Oriented Design

PDF - Portable Document Format

PHP - Hypertext Preprocessor

SHA - Secure Hash Algorithm

SQL - Structured Query Language

UML - Unified Modeling Language

CHAPTER 1 - INRODUCTION

1.0 Chapter Overview

This chapter will provide a brief introduction about Future Minds Montessori School, a description on its problematic areas, and motivation, objectives and benefits of the proposed system.

1.1 Introduction to Future MINDS (AMI) Montessori School

Most parents' utmost dream and effort is to give the best for their child. In that, providing their child a high quality education is one major commitment.

Among many early childhood education service providers in Matara suburbs, Future MINDS Montessori school has recorded a remarkable success in a short period of time. The Montessori Method with the correct approach of scientific observation of child, and medium of class conduct may obviously be the reason behind this other than efforts of a qualified staff and the dedicated managing directress.

Future MINDS Montessori school was established in 2011 in the heart of Matara city. At present, 150 children are taken cared by a staff of 15 teachers and assistants in this school. Primary grade one class, child care center, and extra classes including elocution class are also conducted.

1.2 Motivation for the Project

The client is influenced and strictly adhered (AMI) Montessori methods in almost every activity they carry out. They are practicing well organized and planned approaches in their work to assure high quality and best standard.

However, with increasing demand and the perceptions of the target market, it has been difficult to maintain the high standards with a manual paper based processes. For example, uniqueness is always problematic when monthly evaluation reports and lesson plans, are manually prepared.

In addition, the client is currently in need of a website to expand their service and to upgrade their standards with better quality service.

1.3 Problem Area

Current process encountered problems mentioned below.

- Difficult to manage records manually with the current high demand. Also it adds an extra cost to the institute.
- All the records being kept manually would be insecure.
- Need lot of report generations.
- Communication with parents for daily acknowledgement need to be effective and convenient for parents as most parents do not possess time to visit Montessori daily.

1.4 Objectives of the Project

- To provide better service in the competitive market

  Part of the proposed system is a website available for public with features for current and prospective customers, which will provide better interactivity and convenience. Therefore it is helpful to attract target market and improve market share.
- Effective communication

  Current practice for daily acknowledgment is a written note or mostly a chit. This may not be an effective way of communication, in a situation that the message is not delivered on time. The proposed system would enable instantaneous notifications and feedback.
- To store data in a computer system and maintain a database

  As of now all records are maintained and stored in a paper based system. In fact, a computer system, and a database to maintain and store data electronically is strongly required.
- To reduce unnecessary paper work

  The proposed system would reduce the cost of paperwork and the space used for storage. It would also enhance efficiency and convenience in securing information.
- To maintain the standards and quality

  A computerized system ensures more accurate information and it enables immediate updating of records saving time. Also it generates high quality reports with improved accuracy compared to existing procedure. In addition it will minimize data redundancy too.

Introducing a computerized system allows immediate updating of records; it is possible to save time and to improve accuracy of information system. It will minimize the data redundancy and, will also be helpful in generating more reliable reports that required for swift and accurate decision making.

- To enhance the administration tasks

  The proposed system generates various reports required for decision making. In addition it generates notifications and provides easy to retrieve data. Indeed this assures convenience, and saves time in administrative tasks compared to recent procedure.

1.5 Scope of the Project

The scope of the project is given below,

- System facilitates to store and manage student details, teachers' details and parent information.
- Provides a website which allows parent to view required details such as upcoming events.
- Evaluations can be viewed by relevant parent via web after registration.
- System provides convenience of delivering and managing monthly lesson schedules.
- Payment history can be maintained.
- Daily acknowledgement and special notice can be sent.
- Facilitates efficient monthly evaluation process.
- Facilitates reservation of new admissions.
- There are three level of users−Admin, Teachers, and Parents with appropriate user privileges.
- Can create users are access system securely.
- Report can be granted easily.

1.6 Structure of the Dissertation

1.6.1 Chapter 1 – Introduction

This chapter contains a brief introduction of client, motivation aspects, objectives and scope of the proposed project.

1.6.2 Chapter 2 – Analysis

This chapter includes requirement gathering techniques and outcome of the analysis, functional and nonfunctional requirements and details of the current system process also discussed.

1.6.3 Chapter 3– Design

This consists of database and the main interface of the system according to the analyzed requirements. Many diagrams will be used to depict the functionality.

1.6.4 Chapter 4 – Implementation

This chapter focuses on implementation procedures based on the analyzed results of the previous chapter. Development environment, used technologies and reused modules are discussed.

1.6.5 Chapter 5 – Evaluation

Applied technologies of testing in order to evaluate functionalities within the system cater user requirements and expected level of quality is explained

1.6.6 Chapter 6 – Conclusion

This chapter consists of future enhancements to the system and lessons learnt from the overall project.

1.7 Summary of the Chapter

With the overview provided regarding the purpose of the project next chapter will look in to analysis and understanding on requirements.

CHAPTER 2– ANALYSIS

2.1 Chapter Overview

Main objective of this phase is to present an over view of the business process and to identify the problem areas. Also, this chapter includes functional and non-functional requirements for the proposed system and existing similar solutions and use case diagram for current process.

2.2 Fact Finding Techniques

Fact finding process is a most crucial stage because the decisions, tools and all other phases within the system development process are based on gathered facts. Following methods of fact finding techniques were carried out to in this system process.

1. Interviews
2. Observation
3. Inspection of documents

2.2.1 Interviews

Interview is the commonly used research tool to gather information relates to proposed system. This helped to obtain a clear picture on background of the Montessori, work place situation, ongoing procedures and details as well as to grasp stakeholders' opinions and ideas in a beneficial manner.

A considerable number of interviews were conducted with Managing Directress, staff members and a few parents at the Montessori premises.

Both structured and unstructured approaches of interviewing were used.

2.2.2 Observation

Observation is another fact finding technique used, which yeilded a better understanding of the current working process. It took considerable time to visit the location and see the flow of work.

2.2.3. Inspection of Documents

Inspection of documents was the best way to get detailed description of the existing manual process and how it is carried out. It has maintained many documents in a unique way in their process. Sample copies and screen shots of existing documents and forms, were taken and analyzed. Their reports (outputs) required such as monthly progress evaluation reports, and monthly lesson plan and inputs for the system are clearly identified in this phase.

2.3 Description of the current system

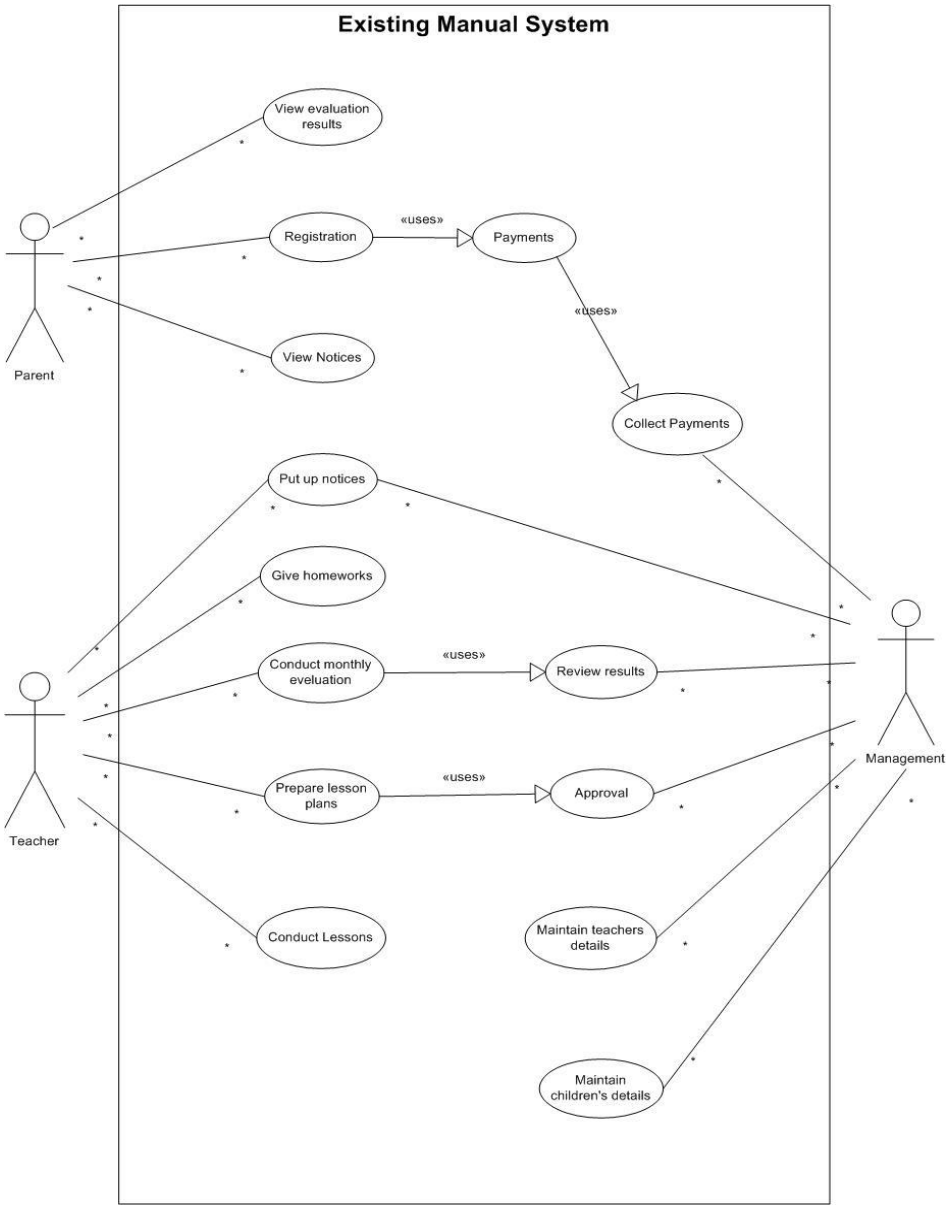


Figure 2.1: Use Case Diagram for manual system

Registration of a child is the first phase of enrolling. An application form is given to be filled out. Duly filled form is filed and maintained. At the same time, frequently used details such as parents' names, contact details are recorded in a registry.

Teachers details also kept as above.

Teachers asses and evaluate a child monthly basis on predefined rubrics accordingly. Such record is also maintained in a special register by the principal. Any parent could visit and check on the progress of their child.

Syllabus is predefined and updated by the principal.

Teachers prepare lesson plans monthly based on syllabus for a period of a month and only the approved lesson plan by the principal is conducted during the month.

Homework and special notices are informed via a 'Chit' or displayed on the notice board.

Fees and charges are collected on monthly basis and these also recorded manually.

2.4 Existing similar system

Considering the requirements of the client, it was decided to search similar systems in order to provide best possible solution for the purpose.

There are many websites related to nursery, but most of them only represent details. And there also are websites which cater to create own website with attractive designs, yet, to the best of our knowledge most of them are not sufficient to meet client requirements of the Future MINDS (AMI) Montessori School.

However, a few websites were selected for the study since they shared most of very frequent expectations of the whole spectrum of similar web sites. One of those web sites, Guiding steps International Pre-school Montessori Academy is of local nursery while all others, including The Nest and Jubilee Day Nursery are nonnative.

The study leads to obtain a good idea about the main functionalities to be focused on the needed website and how to present and arranged them in a much better way.

In fact, the first two websites appeared comparatively comprehensive and much similar to the requirements. However, it was not possible to conduct an in depth analysis of any of them due to restricted access. Therefore, exploring similar existing systems to identify relevant solutions was not satisfactory as expected. The interfaces of above discussed web systems are represented in figure 2.2, figure 2.3, and figure 2.4.

Figure 2.2: The Interface of The Nest Nursery school

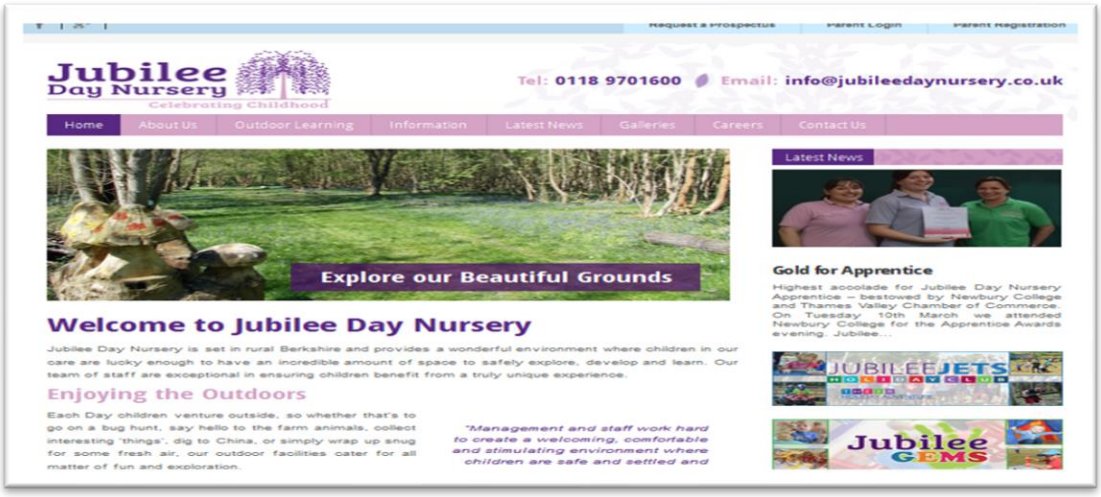


Figure 2.3: The Interface of Jubilee Day Nursery

Figure 2.4: The Interface of Guiding Step Nurs**ery**

2.5 Functional requirements

In software engineering (and systems engineering), a functional requirement defines a function of a system and its components. A function is described as a set of inputs, the behavior, and outputs. [1]

Functional requirements should depict facilities or features to be performed by the system to carry out the operations. Main functional requirements which were identified at this stage are mentioned below:

- Children and staff details maintaining module

  This is done by management or whom this privilege is granted.

  Records are maintained in a database and could create, edit, view, delete when it is necessary.

- Evaluation module

  Children are being evaluated monthly and derived results should be entered by the corresponding teacher.

  Management is analyzing above records.

  Parents can check out their child's evaluation results and progress online.

- Lesson planning module

  All the teachers should submit pre-prepared lesson plan for the next month before a closing date.

  Management checks individual lesson plans and approve them.

This module helps to maintain and update records periodically.

- New admission module

  Facilitate to apply online for enrollments when new admissions are called up.

  Help management to assess the demand for the new batches and resources.

- Payment History Management

  Facilitate management to record and analyze fees and other charges received.

- Login module

  Provision is only to access different privileges and functionalities only to authorized members of the Montessori.

  Focus on three types of user levels; Admin, Staff member, Parent.

- General website

  This is to share general information among interested parties.

  Focus to facilitate communication with parents.

  Users can log remotely and carry out specific tasks.

- Management can send notifications to target groups or specific user.
- Report generation can be done based on the set privileges and user levels.

2.6 Non Functional requirements

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. [2]

Non functional requirements describe the quality of the system. Despite achieving functional requirements, achieving non functional requirements furnish the stability and the standard of the system.

- System should be user friendly and simple.
- Security is a priority because personal details are maintained in the system. Access privileges should be maintained accurately.
- High accuracy is required due to the fact that children's records are frequently communicated and parents are very passionate about them. As a result a minor error would build up a considerable distrust among parents leading to low usage. Excellent performance of the web site is mandatory since it has a direct positive impact on the reputation of the Montessori.

2.7 Methodology for the Proposed System

A software development methodology (also known as a system development methodology, software development life cycle, software development process, and software process) is a division of software development work into distinct phases (or stages) containing activities with the intent of better planning and management. It is often considered a subset of the systems development life cycle. The methodology may include the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application.[3]

There is no ideal or specific process model for development of a given system. Different approaches are being used in software development depending on the system to be developed. The process model is selected based on the capabilities of the developing people and specific characteristics of the systems which are being developed. Many development process models are being deployed in software development industry.

Among those models Waterfall model is used as selected methodology. Due to the fact that the time is need to be well managed with the development phase, not to change the solution drastically and therefore the requirements needed to be thoroughly identified before design and developments stages. Even though Rapid Application Development could also be used due to above reasons Waterfall methodology is adopted for this.

CHAPTER 3 – DESIGN

3.1 Chapter Overview

This chapter is mainly focused on the approaches, and alternate solutions, a few diagrams including Use case diagram, Class diagram and Entity Relationship diagram depicting object-oriented approach, that have been used to design the proposed system.

3.2 Selected Solution

Web based approach is found to be the best solution for the problem domain. This approach facilitates,

- Remote access, and accessibility via a range of devices; provides convenience for users
- Easy to share information
- Easy installation
- Easy and quick content updating.

Any party who is authorized and interested in could also use the system.

3.3 Alternative Solution and Feasibility

Communication is a prior concern for the system.

Standalone system approach does not provide a better solution mainly due to its high maintenance cost, and lack of support in sharing information, compared to alternative solution approach, yet it facilitates managing documentation.

On the other hand, using several software, each one is serving solely for one or few of the requirements, is also discouraged (a website, content management system) since, these software may contain unwanted features, and would incur extra expenses.

3.4 Object Oriented Design

In order to develop the required web-based system efficiently, object-oriented analysis, and design methodology is used, since this methodology allows intended divers and complicated functions to be incorporated in to the system.

OOD allows large-scale applications to be developed in independent modules. Object-oriented decomposition provides a method to decompose a complex arrangement by

the primary objects apparent in the system. "Once the objects are defined and the system functionality is assigned, major components of the software system are developed independently. The parallel development and testing of individual modules requires strict adherence to the specification interface requirements".[4]

Further, OOD supports developing the system module-wise, and it identifies patterns to determine what components to be used repeatedly on shared characteristics. Objects, class, abstraction, encapsulation, inheritance, polymorphism are some of the concepts of object oriented world.

3.5 Design Techniques

MVC framework was used for the system development, MVC is a framework pattern that splits an application's implementation logic into three component roles.

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.[5]

- Model - The lowest level of the pattern which is responsible for maintaining data.
- View - This is responsible for displaying all or a portion of the data to the user.
- Controller - Software Code that controls the interactions between the Model and View

Figure 3.1 depicts the interactions between the components of MVC architecture.

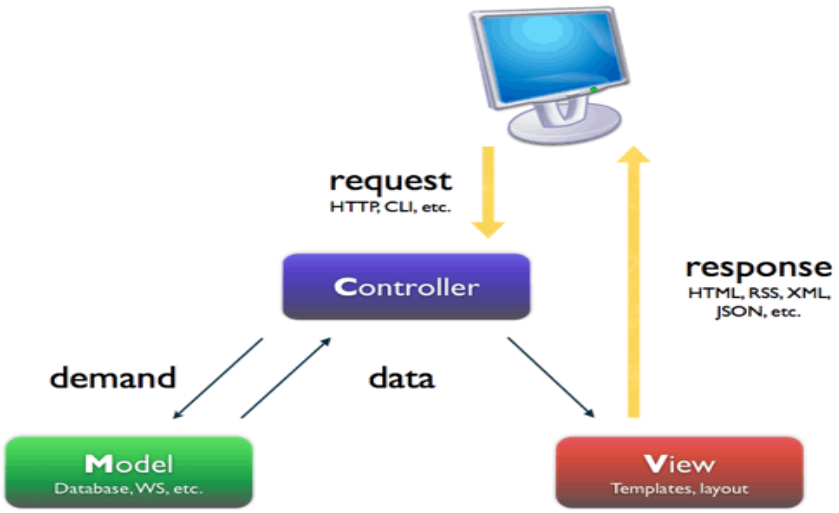


Figure 3.1: MVC architecture

PHP technology was chosen for developing the system with the help of PHP related framework called “CodeIgniter” which makes system highly maintainable, reliable and provides proper interoperability between system modules. Code Igniter design with MVC architecture has been described under section 3.5.

MySQL was used as the relational database management system and Apache server was selected as the application server which means the software bundle WampServer was used. Wamp Server is a web based platform which helps to run dynamic web pages with help of PHP and MYSQL on windows.

Bootstrap framework was chosen to create user interface including responsiveness, user-friendliness and efficiency, with the help of CSS and JavaScript.

3.6 Design Diagrams

As mentioned earlier, Object-orientation has evolved as a means to manage the complexity of software systems. After identifying objects of a system and their relationships, diagrams can be produced, which then can be converted to executables using object oriented languages.

Unified Modeling Language is used to model Object Oriented software applications. UML is not a programming language. It is a pictorial language used to make software blue prints. UML diagrams can be used to model different aspects such as static, dynamic and etc.

UML had been created by Object Management Group (OMG).

3.5.1 Use Case Diagram

The use case diagram clearly models the external and internal factors influencing the system and influence of the stakeholder to the system. This diagram consists of actors, use cases and their relationships. It depicts very high level of design.

Figure 3.2 shows the high level use case diagram for the proposed web based system

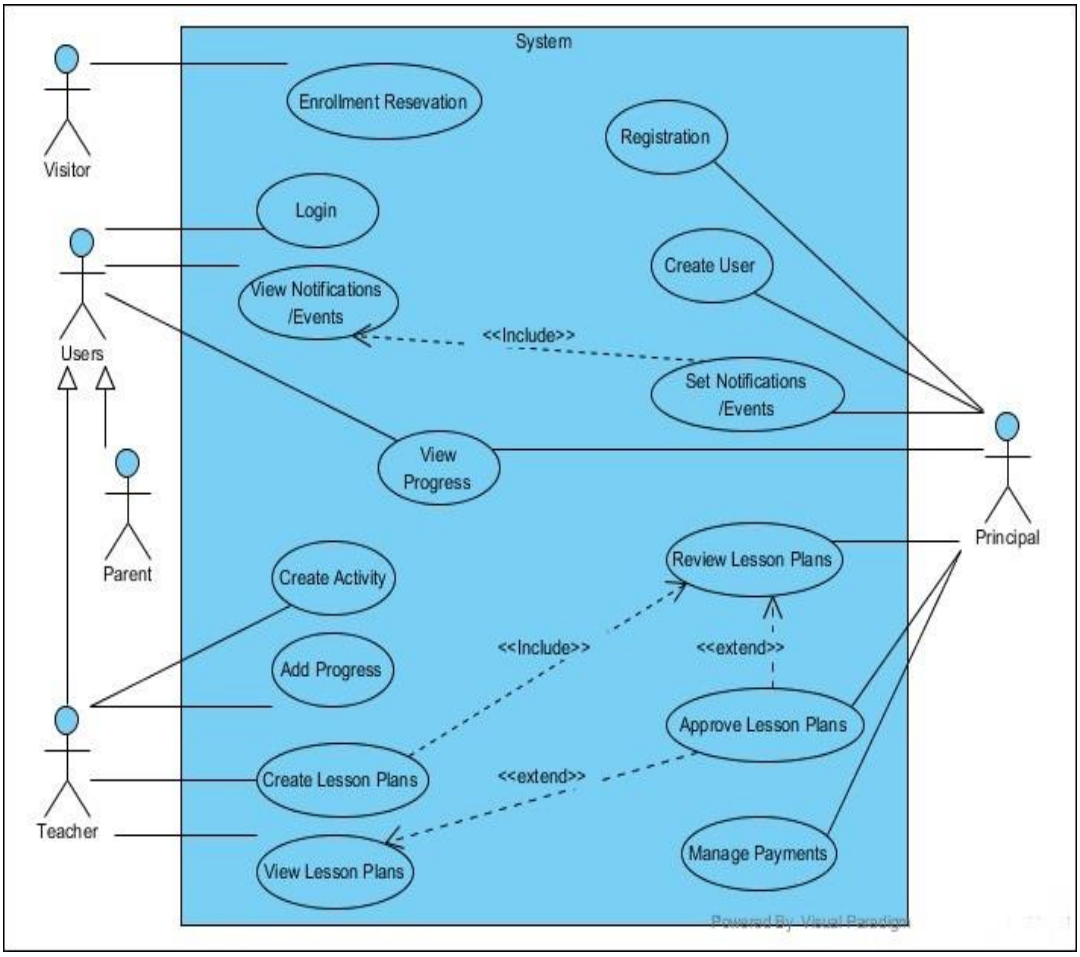


Figure 3.2: High Level Use Case Diagram for the proposed system

For better understanding, the high level use case was isolated in modules in accordance with main functionalities of the system. Few main use cases for the system and their use case narratives are discussed here.

- **Use case diagram for the Lesson Plan Module**

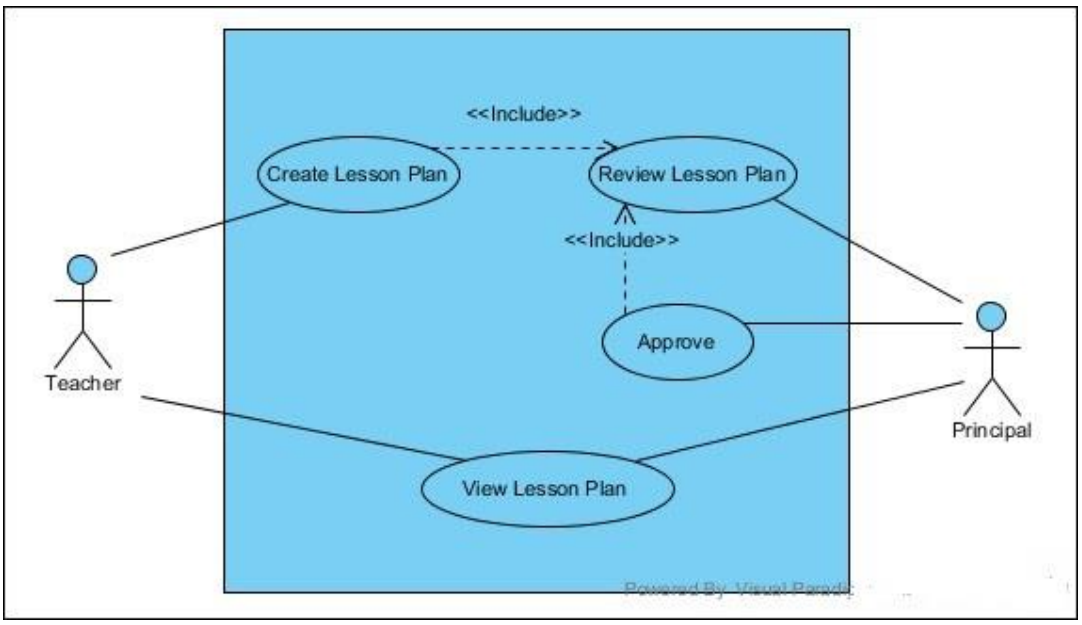


Figure 3.3: Use Case Diagram for Lesson Plan Module

Use Case Name	Lesson Plan Module
Actors	Teacher Principal
Description	Planning lesson plans for the classes
Precondition	Principal and Teacher should login to the system.
Typical Course of Events	Teacher should create a new lesson plan for a particular period. Principal then is being able to review the lesson plan accordingly and approve or reject it by adding a comment to the lesson plan. The teacher who added the lesson plan then can see the comment as well as its status.
Post Conditions	The relevant lesson plan will be saved within the system and can be viewed.

Table 3.1: Use case narrative for Lesson Plan Module

- **Use case diagram for the Evaluation Module**

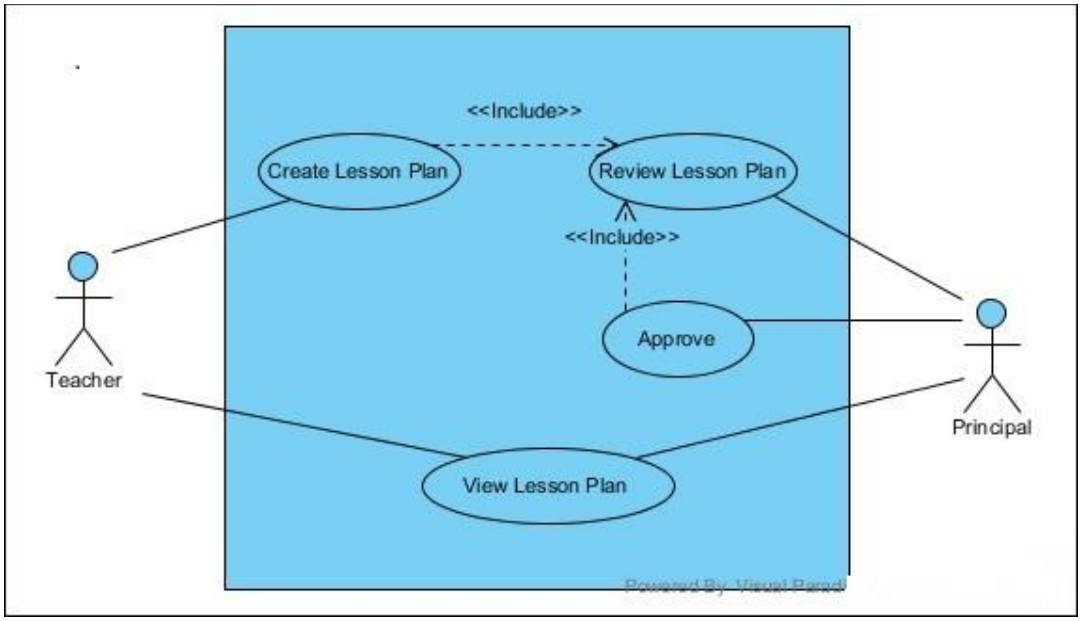


Figure 3.4: Use Case Diagram for Evaluation Module

Use Case Name	Evaluation Module
Actors	Teacher Principal Parents
Description	Teachers will be evaluating students by activities.
Precondition	Principal, Teacher, Parents should login to the system.
Typical Course of Events	Teacher should create a new activity for a particular subject. Teachers then carry out the activity in the class and will be add the level of the student to the system, based on the carried out activity. Relevant teacher, principal and the parents can view the level of the student as a progress.
Post Conditions	Progress (achieved levels for the activities) of the student will be saved within the system and can be viewed.

Table 3.2: Use case narrative for Evaluation Module

All other use case diagrams and use case narratives for the system can be found in Appendix B.

3.5.2 Class Diagram for the system

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application.[6]

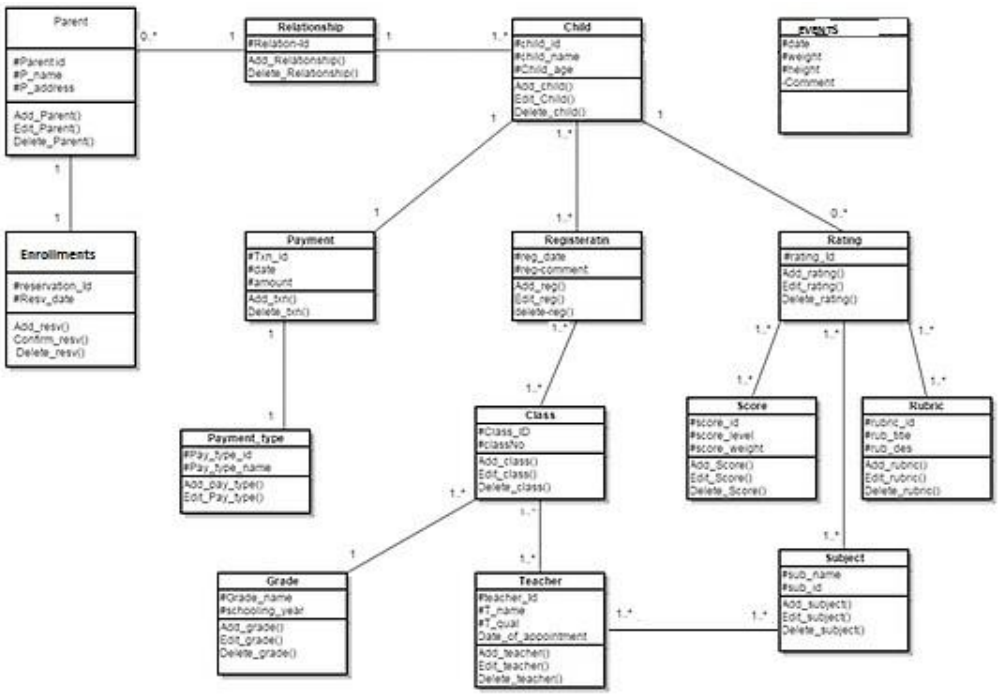


Figure 3.5: Class Diagram for the proposed system.

1.5.3 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.[7]

Figure 3.6 depicts the flow of activities occurred when a user login to the system. Further activity diagrams related to the system can be found in Appendix B - Design Documentation.

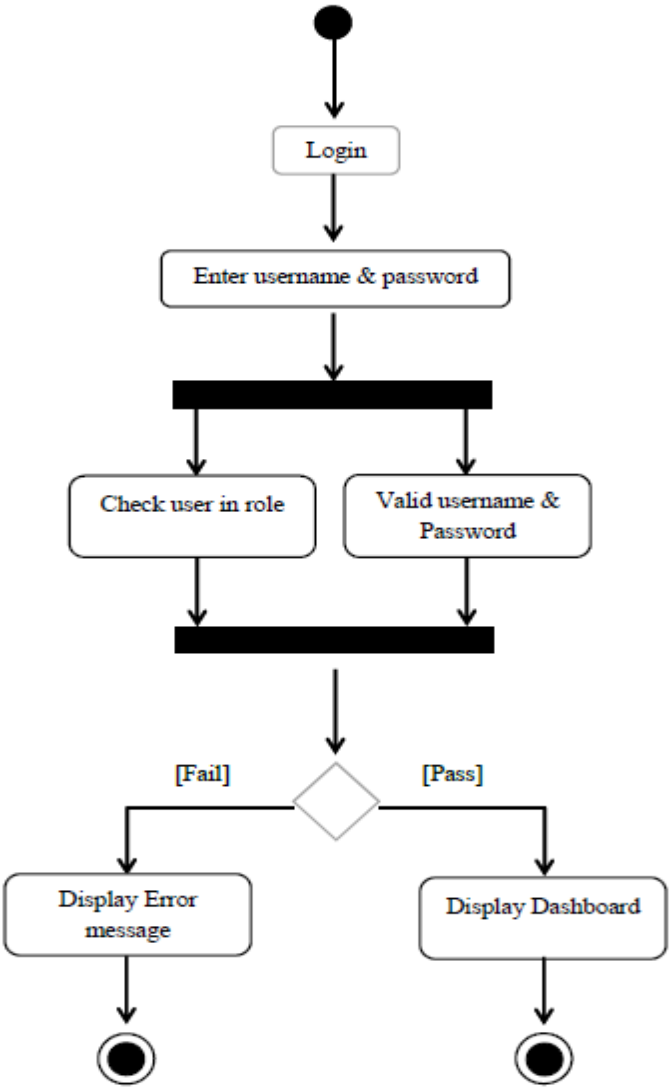


Figure 3.6: Activity Diagram for the login.

1.5.4 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct 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.[8]

Figure 3.7 shows the sequence diagram for the login module. Further sequence diagrams related to the system can be found in Appendix B - Design Documentation

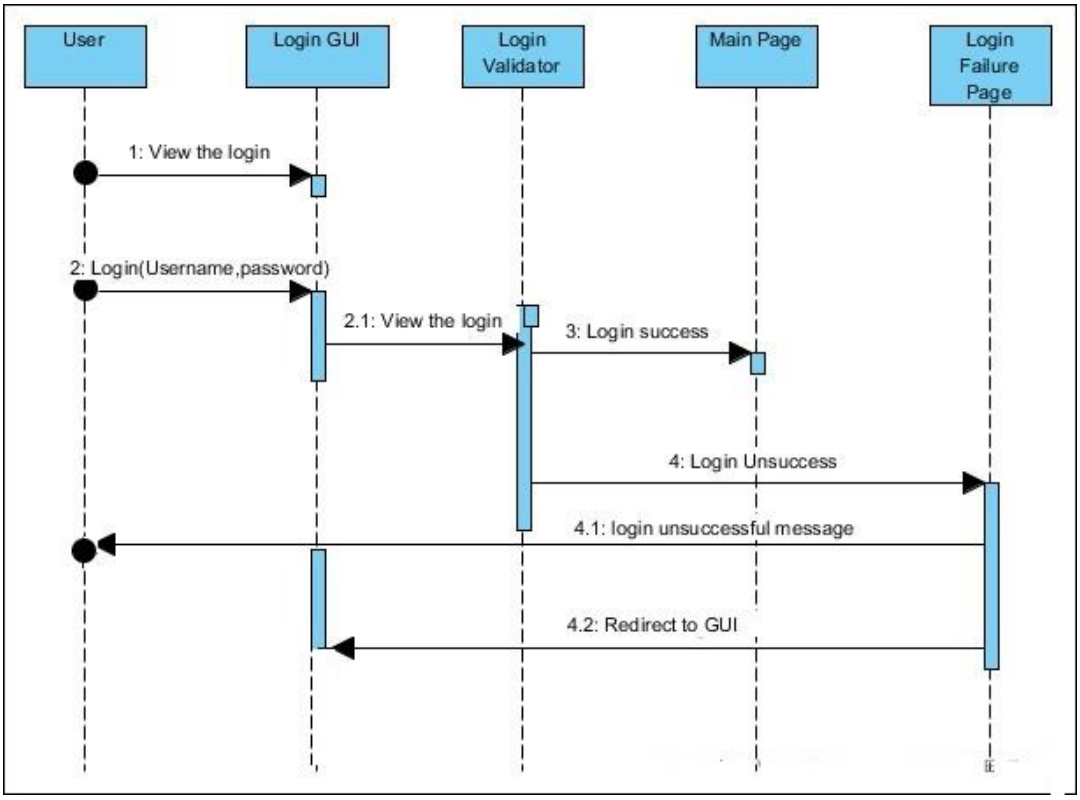


Figure 3.7: Sequence Diagram for the login

3.5.5 Entity Relationship Diagram

Entity Relationship diagram is a graphical representation of system entities and relationships among them. Entities, relationships, and attributes are element of the Entity Relationship Diagram.

Following figure 3.8 depicts the Entity Relationship Diagram of the proposed system.

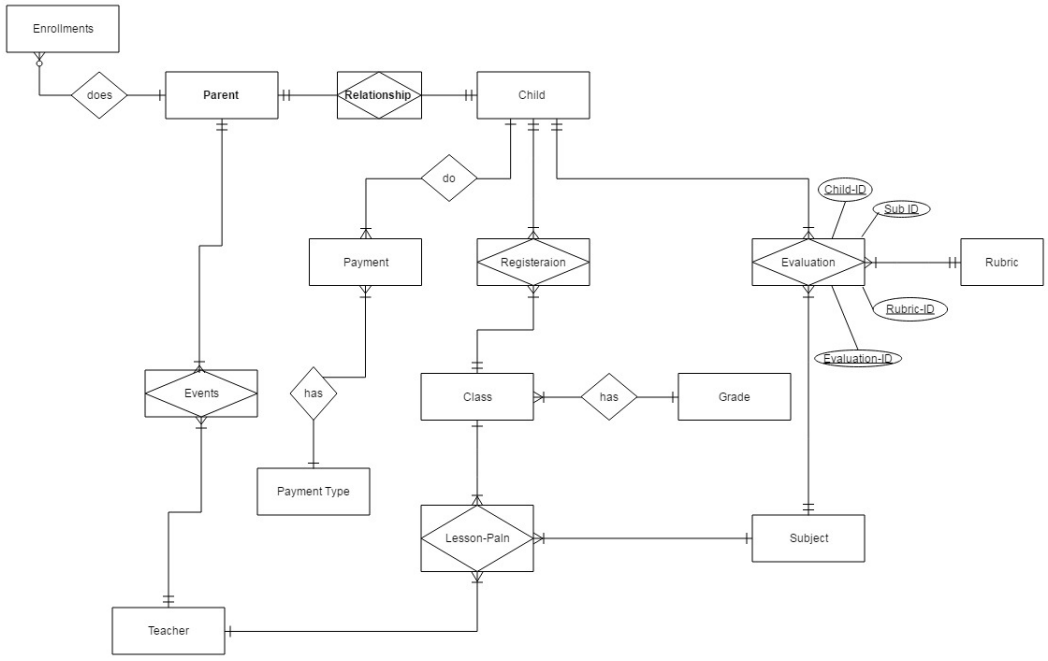


Figure 3.8: Entity Relationship diagram of the proposed system

3.6 Database Design

The following, figure 3.9 depicts the table arrangement in the database of the proposed system.

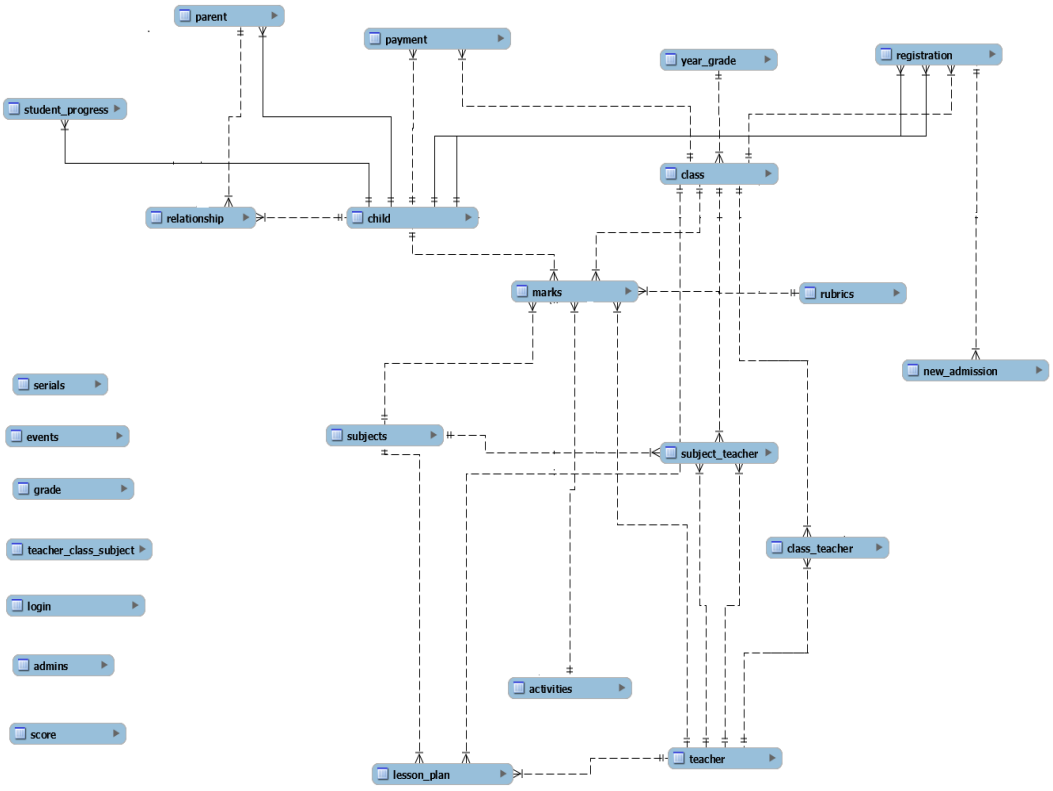


Figure 3.9: Data base design of the system

3.7 User Interface Design

Interface is the layer via which communication with the user and the system is done. User interfaces should make it convenient to the user to perform its operation without making unnecessarily complications. It should be simple and easy to use for users. As an example, icons and labels should be meaningful. According to the user requirements the user interfaces were designed.

Model-View-Controller architecture is used to design the system. In this concept software is divided into three main components; then these components communicate with each other to perform a particular task. Where View is represented date and interacts with users.

Interfaces for the proposed system are different from each other depending on the expected operations, functions and the user. This approach provides convenience to users and to meet the client requirements.

Design layout for general web site is more information oriented and colors and other facilities suggested mainly focusing external users. Figure 3.4 shows the index page of the web based system where Figure 3.5 depicts the login page for the system where that page is common for all the users. Providing user name and password any authorized users can login to the system

Figure 3.10: The index page of web based system

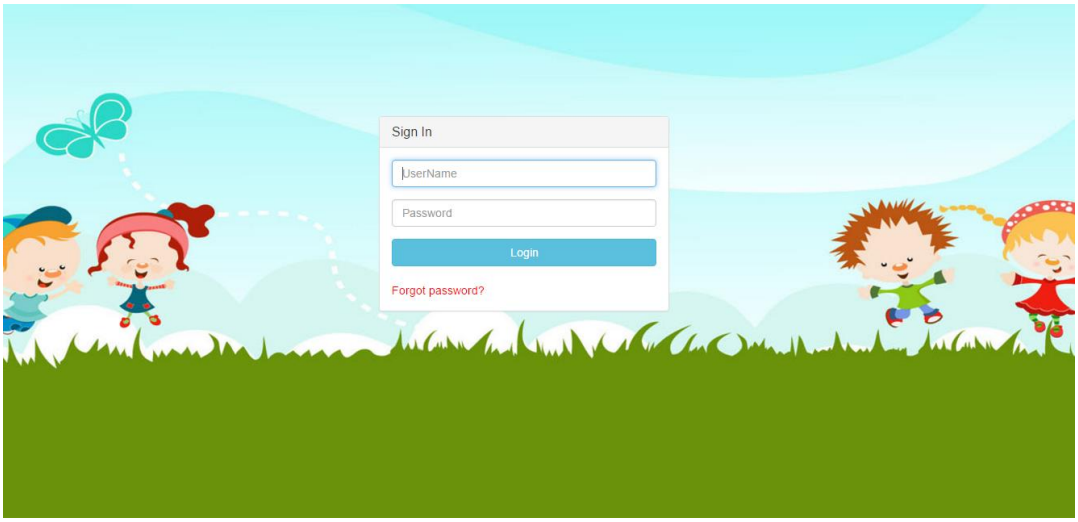


Figure 3.11 The Login Page

After a successful login internal system user interface is way different from general web pages. This is where main operations and functions are performed and users communicate with system for during a long period of time. Therefore, if not optimal, colors and effects could make the user tired.

Figure 3.12 shows the administrator's dashboard.

Figure 3.13 depicts form of, customizable system tools added which is aan administrator's function.

Figure 3.14 contains a form that is enabled for teachers to add levels to the students.

Figure 3.12: The Administrator Dashboard

In the administrator functions customizability is concerned as much as possible. Figure 3.13 depicts form of, customizable system tools added which is an administrator's view. In this form it can be restored some deleted data if it wanted again, when some conditions satisfied.

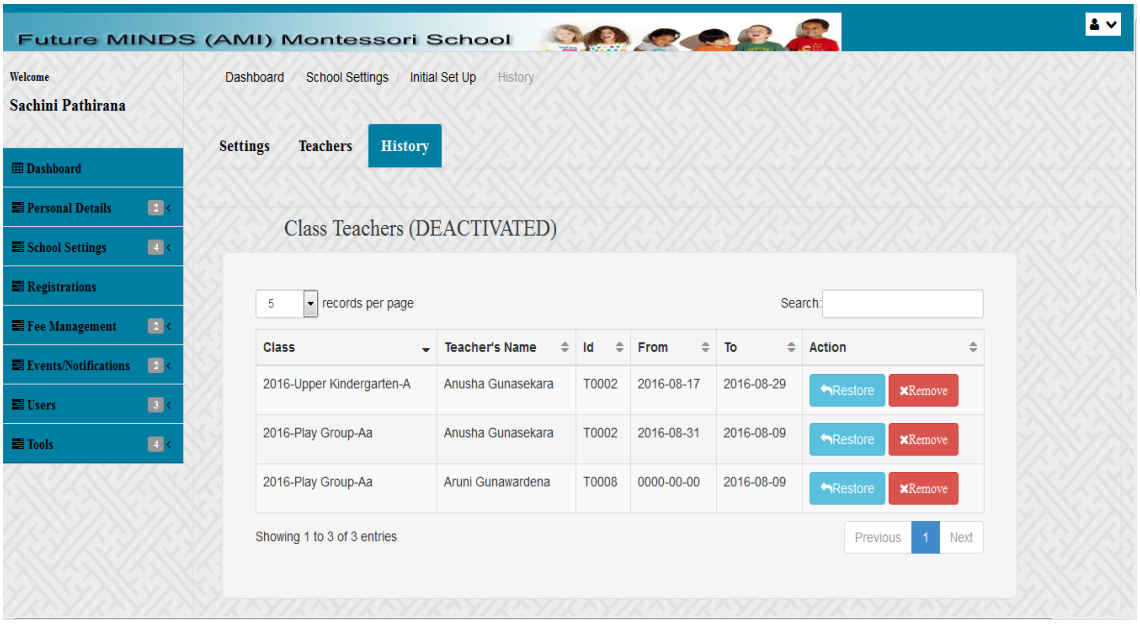


Figure 3.13: An Administrator's Function.

Figure 3.14 contains a form is enabled for teachers which is allowed to add levels to the students. It is to mention that all the forms even with a two field is validated before submitting it.

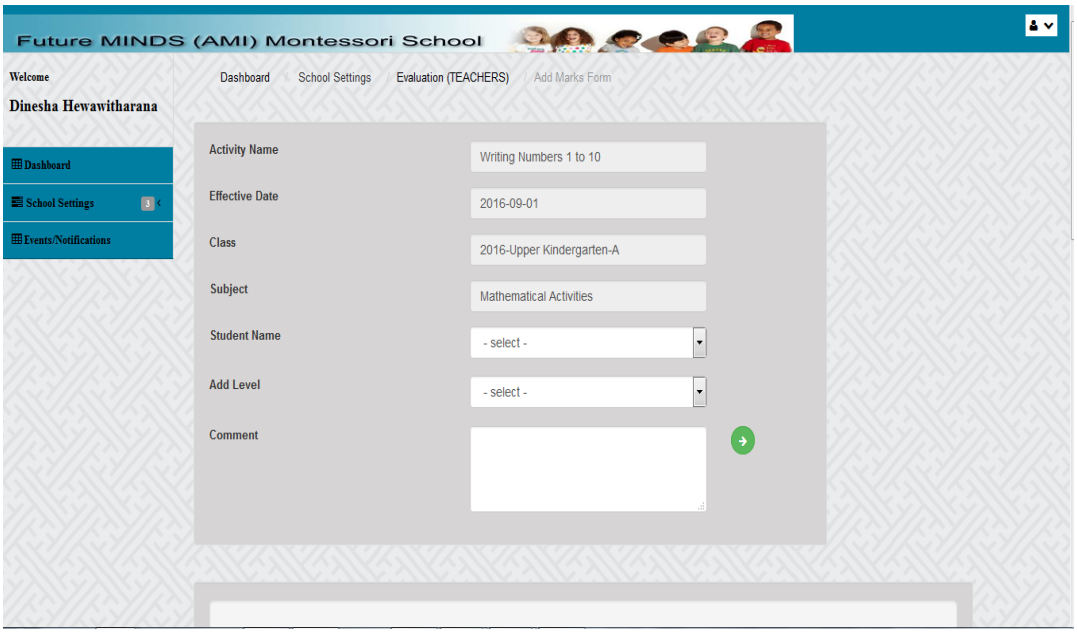


Figure 3.14: A Teacher's Function

CHAPTER 4 – IMPLIMENTATION

4.1 Chapter Overview

Implementation is the process of converting user requirements and system functionalities into the actual working system. In this chapter the facts which are necessary to implementation the system are discussed. Therefore, this chapter includes software environment, hardware environment, architectural overview, reused modules, and network diagram of the implementation environment, code structure and security.

4.2 Implementation Environment

The implementation Environment is based on various aspects such as developing software, framework and etc. Implementation environment is the collection of hardware and software tools which are used by a system developer to build software systems.

4.2.2 Software Environment

A software environment for an application include the operating system, the database system, specific development tools or compiler.

- Microsoft Windows 7 Professional - 32bit
- WAMP Version: 2.4
  - Apache Version: 2.4.4
  - MySQL Version: 5.6.12
- NetBeans IDE Version: 8.1

4.2.3 Hardware Environment

- Intel(R) Core(TM) Dual CPU T5870 @ 2.20GHz
- 2 GB RAM
- 300 GB Hard disk

This web based system is compatible with other operating system although the system was implemented under windows based operating system. And it is fully responsive with mobiles, tabs, and monitors of varying sizes.

4.2.4 Supported Tools and Technologies

- PHP is the main developing language selected for developments.
- MYSQL was chosen to handle databases queries of the system
- HTML and CSS was used to make the front end interfaces for the system.
- JavaScript was used for the client side validations such as form validation, and other simple tasks to be performed such as trigger an alert after a mouse click so on.
- AJAX, JSON, and JQuery were used to deal with the database with lesser user intervene and to make it easier for the user.
- Firebug 2.0.11 is a tool which comes with 'Firefox' web browser, and it was used to JavaScript debugging, and analyzing network usage. Further, it facilitates inspecting the HTML and modifying the HTML layout and the styles in real-time.
- Notepad++

4.3 Open Source Frameworks

4.3.1 CodeIgniter v2.2.0

CodeIgniter is an open source rapid development web application framework, for use in building dynamic web sites with PHP. It is a free, open-source, easy-to-use, object-oriented PHP web application framework, provides a ready-to-use library to use with own PHP applications, without having to create a lot of repetitive code. Also it is simple to configure and customize according to own needs.

CodeIgniter is loosely based on the popular model–view–controller (MVC) development pattern. While controller classes are a necessary part of development under CodeIgniter, models and views are optional. Codeigniter can be also modified to use Hierarchical Model View Controller (HMVC) which allows developers to maintain modular grouping of Controller, Models and View arranged in a sub-directory format. CodeIgniter is most often noted for its speed when compared to other PHP frameworks.[9]

4.3.2 Bootstrap v3.1.0

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with

front-end development only. Bootstrap is the second most-starred project on GitHub, with more than 100,000 stars and 45,000 forks.[10]

4.4 Reused Modules

The following pre-coded modules were used while developing the system.

- **Bootstrap Datepicker**

  Datepicker is a highly configurable plugin that adds datepicker functionality to your pages. You can customize the date format and language, restrict the selectable date ranges and add in buttons and other navigation options easily.[11]

- **Date Range Picker**

  Originally built for reporting at Improvely, the Date Range Picker can be attached to any webpage element to pop up two calendars for selecting dates, times, or from predefined ranges like "Last 30 Days".[12]

- **PHPMailer**

  PHP Mailer is a code library to send (transport) emails safely and easily via PHP code from a web server (MUA to the MSA server).Sending emails directly by PHP code requires a high-level familiarity to SMTP standard protocol and related issues (such as Carriage return) and vulnerabilities about Email injection for spamming. From 2001 up to 2013 PHPMailer is one of the popular solutions for these matters on PHP.[13]

- **ChartJS**

  ChartJS provides beautiful flat designs for charts. It uses HTML5 canvas element for rendering. Support for older browsers like IE7/8 is added through polyfill.ChartJS charts are responsive by default. They work well in mobiles and tablets.[14]

- **DataTable**

  DataTables is a plug-in for the jQuery Javascript library. It is a highly flexible tool, based upon the foundations of progressive enhancement, and will add advanced interaction controls to any HTML table.[15]

- **mPDF**

  mPDF is a simple and popular tool for shared hosting users to create and convert UTF-8 encoded HTML pages to PDF files. [16]

4.5 Folder Structure

The folder structure is generated by the CodeIgniter Framework tool, which is bundled with the CodeIgniter Framework. Figure 4.1 depicts the folder structure of the system.

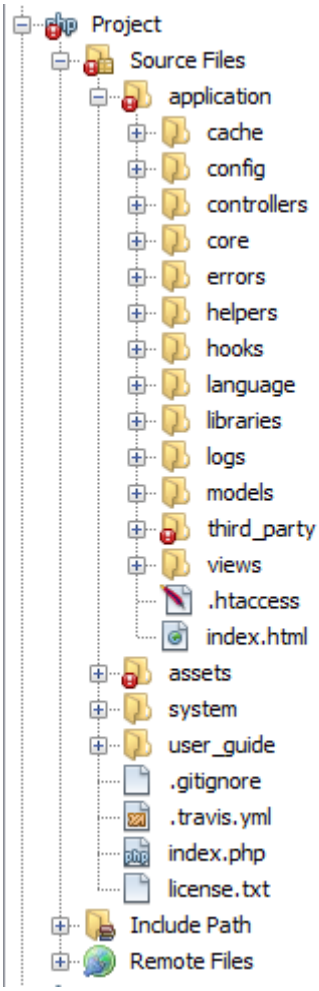


Figure 4.1: Folder structure of the system.

4.6 Major Code Segments

As mentioned in section 3.5, MVC framework used for the system development, which consists of an architectural pattern of Model-View-Controller (MVC). The MVC is a useful pattern for the reuse of object code and it does significantly reduce the time it takes to develop applications with user interfaces.

4.6.1 Model

The model manages fundamental behaviors and data of the application as well as it is the underlying, logical structure of data in the software application. It can respond to requests for information, respond to instructions to change the state of its information. In short, it is the data and data-management of the application. And it does not depend on the controller or the view.

Model includes readymade database queries such as insert, update, get and etc. before use of these queries it's compulsory to declare a class in model and the declared class should extend of CI_Model class of the CodeIgniter.

Following code fragment shows the declaring CI_Model and retrieving data from the database using get function.

```
class Generic_model extends CI_Model {

    public function getData($fieldset, $tableName, $where = '') {
        if ($where == "") {
            $this->db->select($fieldset)->from($tableName);
        } else  {
            $this->db->select($fieldset);
            $this->db->from($tableName);
            $this->db->where($where);
        }
        $query = $this->db->get();
        return $query->result();
            }
```

Insertion and updating of data to the database and joining table of the table is as follows.

```php
function insertData($tablename, $data_arr) {
    $result = $this->db->insert($tablename, $data_arr);
    return $result;
        }

public function updateData($tableName,$dataArra,$whereArr) {
    $this->db->where($whereArr);
    $result=$this->db->update($tableName, $dataArra);
    return $result;
        }

public function join($fields, $wherefieldtablefrom, $tablefrom, $tablejoin, $tablejoincondition) {
    if ($wherefieldtablefrom == "") {
        $this->db->select($fields);
        $this->db->from($tablefrom);
        $this->db->join($tablejoin, $tablejoincondition);

    } else {
        $this->db->select($fields);
        $this->db->from($tablefrom);
        $this->db->join($tablejoin, $tablejoincondition);
        $this->db->where($wherefieldtablefrom);

    }
    $query = $this->db->get();
    return $query->result();
        }
```

4.6.2 View

The view provides the user interface of the application. The things that the user can see and respond to on the screen, such as buttons, display boxes etc. The view displays the model data in a form that is suitable for the user interface. It sends user actions (e.g. button clicks) to the controller.

Following code fragment is an example code in View element in MVC.

```php
<!DOCTYPE html>
<!--Administrator Dashboard-->

<html>
<!-- BEGIN HEAD-->
<head>

    <title>Dashboard</title>
    <?php echo $this->load->view('admin_panel/template/header'); ?>
    <!-- PAGE LEVEL STYLES -->
    <link href="<?=base_url();?>assets/admin_assets/css/layout2.css" rel="stylesheet" />
    <!-- END PAGE LEVEL  STYLES -->

</head>
<!-- END  HEAD-->
    <!-- BEGIN BODY-->
<body class="padTop53" >

     <!-- MAIN WRAPPER -->
    <div id="wrap" style="background-image: url('http://localhost:8080/Project/assets/imgs/bgptn.png');
         background-repeat: repeat;">
        <!--HEADER SECTION -->
         <?php echo $this->load->view('admin_panel/template/navbar'); ?>
        <!-- END HEADER SECTION -->

        <!-- MENU SECTION -->
          <?php echo $this->load->view('admin_panel/template/menubar'); ?>
        <!--END MENU SECTION -->
```

4.6.3 Controller

The controller which represents the classes connecting the model and the view which provides model data to the view, and interprets user actions such as button clicks. It depends on the view and the model.

Following code fragment shows declaring setup class and setup class's index function.

```php
<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');
class setUp extends CI_Controller {
     function __construct() {
        parent::__construct();
        $this->load->model('Generic_model', '', TRUE);
        $this->load->model('system_model', '', TRUE);
    }
    public function index() {
                if ($this->session->userdata('logged_in')) {
            $session_data = $this->session->userdata('logged_in');
            $group_id = $session_data['group_id'];
            $username = $session_data['username'];
           // inner if
            if ($group_id === "10") {
                //get user info
                $fields = "*";//get teacher info
                $whereArr=array("user_id" => $username);
                $data['userData']=$this->Generic_model->
                        getData($fields,'admins',$whereArr);
//                  var_dump($data);
                $this->load->view('admin/School/index',$data);
       } else {
                $this->load->view('admin/accessDeniedPage');
            }
       //outer else
        } else {
            redirect('login/index');
        }
    }
```

4.6.4 Config Folder

The file in this folder is very needed in configuring the application. Files in the Config folder helps to set basic configurations for the application. It helps to retrieve

configuration preferences. Basic configurations such as Base URL configuration, Database configuration, Auto Load configuration, email configuration are very helpful in efficient developments.

Database configuration via Data Base configuration file settings are as follows.

```
$active_group = 'default';
$active_record = TRUE;

$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'root';
$db['default']['password'] = '';
$db['default']['database'] = 'future-minds';
$db['default']['dbdriver'] = 'mysql';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;
```

Base URL configuration is as follows.

```
$config['base_url']     = 'http://localhost:8080/Project/';
```

4.7 Security

Since the system contains sensitive information related to students, parents, and teacher, good security is a major concern. Passwords used in the system are encrypted using SHA1 algorithm. Each type user's access is limited according to their user role and system verifies the user in all the controllers' constructor method.

CHAPTER 5 – EVALUATION

5.1 Chapter Overview

This chapter is focused on the evaluation procedure of the developed system. Test plan, test cases, user acceptance test results were carried out in order to ensure that the all aspect of the system have been tested.

5.2 Software Testing

There are different levels of testing to be carried out while developing as well as after integration, before implanting the software in the business environment. In software testing there are four main hierarchical levels of testing. They are as follows.

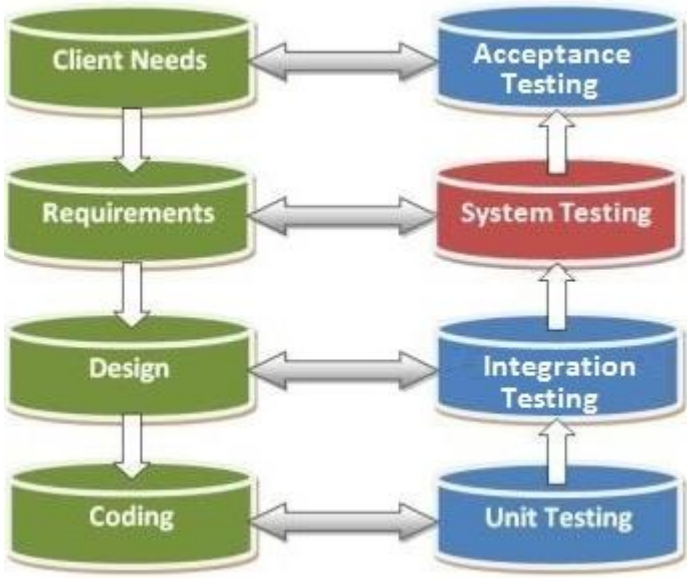


Figure 5.1: Hierarchical levels of testing

5.2.1 Unit testing

Testing is done in the development process while developer completes the unit development. The object of this testing is to verify correctness of the module. The purpose of unit testing is to check that as individual parts are functioning as expected. Basically Unit testing is typically carried out by the developer.[17]

5.2.2 Integration testing

System Integration Testing is started after the individual software modules are integrated as a group. A typical software project consists of multiple modules and these are developed by different developers. So in integration testing is focuses to check that after integrating modules is two modules are communicating with each other or not. It

is critical to test every module's effect on the entire program model. Most of the issues are observed in this type of testing. [17]

5.2.3 System testing

This is the first time end to end testing of application on the complete and fully integrated software product before it is launch to the market. [17]

5.2.4 Acceptance testing

User acceptance is a type of testing performed by the Client to certify the system with respect to the requirements that was agreed upon. This is beta testing of the product & evaluated by the actual end users. The main purpose of this testing is to validate the end to end business flow. [17]

This system was tested under above mentioned tests and those tests are listed under Test Plan and Test Cases chapter.

5.3 Test Plan and Test Case

In the testing procedure, most important part is creating test cases. Once creating the test strategy, test case creation is done. Generally test case consist description of test case, steps to test, expected output and priority. In order to mitigate the difficulty of the system, system has divided into modules. Test cases were written for each module. Following tables shows some test cases models.

5.3.1 User Login

Test No.	Test Description	Expected Result	Test priority
1	Insert invalid username or password in login form.	Display the error message	High
2	Empty the username or password in login form.	Display "Please fill out this field" message	High
3	Insert valid username and password and click login	Valid username/password and redirect to the relevant user level's dashboard	High
4	Display logged username	Show logged username, on the page	Medium
5	Click the logout button given on header	The user will be logged out from the system and redirect to the login page	High

Table 5.1: Test cases for the User Login

5.3.2 Student/Teacher Registration (Personal Details module)

Test No.	Test Description	Expected Result	Test priority
1	Insert invalid details to the fields (Ex wrong date format, text to telephone no fields, numbers to text fields) in the forms.	Display proper error messages	High
2	Empty the required fields	Display “This field is required” message, and prevent submit the form	High
3	Insert valid data in the form and click submit	Redirect to the relevant view and when display the inserted data in the applicable table. After a successful submission generate next serial number accordingly.	High

Table 5.2: Test cases for the Student/Teacher Registration

5.3.3 Lesson Planning Module

Test No.	Test Description	Expected Result	Test priority
1	Add / Edit Lesson Plans	Display the relevant interfaces and the messages.	High
2	Add / Edit Lesson Plans without valid /required fields	Display “This field is required” message, and prevent submit the form	High
3	Add / Edit Lesson Plans with valid /required fields	Redirect to the relevant view and when display the inserted data in the relevant tables.	Medium
4	Admin is to comment without selecting an “Approval status”.	Preventing submit and display “Please select an approval status”	Medium
5	Click Delete Pending approval Lesson Plans	Display “You are going to delete this record”	Low
6	List selected class’s Approved Lesson Plans	List the relevant Lesson Plans in a view	Medium

7	List selected teacher's Approved Lesson Plans	List the relevant Lesson Plans in a view	Medium

Table 5.3: Test cases for the Lesson Planning Module

5.3.4 Evaluation Module

Test No.	Test Description	Expected Result	Test priority
1	Add / Edit Activities	Display the relevant interfaces and the messages.	High
2	Add / Edit Activities without valid /required fields	Display "This field is required" message, and prevent submit the form	High
3	Add / Edit Activities with valid /required fields	Redirect to the relevant view and when display the inserted data in the relevant tables.	Medium
4	Trying to a level for already marked student.	Preventing submit and display "Level is already assigned"	Medium
5	Click Delete Activities	Display "You are going to delete this record"	Low

Table 5.4: Test cases for the Evaluation Module

5.3.5 Initial Set Up Module

Test No.	Test Description	Expected Result	Test priority
1	Add / Edit contents without valid /required fields	Display "This field is required" message, and prevent submit the form	High
2	Add / Edit contents with valid /required fields	Redirect to the relevant view and when display the inserted data in the relevant tables.	Medium
3	Restore a deleted, class/subject teacher which is still not assigned a new class/subject teacher.	Do the action and display "Successfully restored the class/subject teacher"	High

4	Restore a deleted, class/subject teacher which is already assigned a new class/subject teacher.	Display "Teacher is already Assigned"	High
5	Click Delete Activities	Display "You are going to delete this record"	Low

Table 5.5: Test cases for the Initial Set Up Module

5.3.6 Payments

Test No.	Test Description	Expected Result	Test priority
1	Add payment without valid /required fields	Display "This field is required" message, and prevent submit the form	High
2	Add payment with valid /required fields	Submit the payment and it displays in All Transactions view. After a successful submission generate next Txn number accordingly.	High
3	Print the receipt for a payment received.	Generate and save a receipt.	High
4	Change the student's class after payments.	Change the students class for the next year after successful payments under Renewal Admission/New Addmission.	High
5	View the transaction history	Ability to view all the transaction from the very beginning.No user is allowed to change the records.	High

Table 5.6: Test cases for the Payments

5.3.7 Reports

Test No.	Test Description	Expected Result	Test priority
1	Generate reports: click on the relevant option menu/button.	Generate the relevant report with required fields.	High
2	Download the report.	Download the report	High

3	Print the report.	Print the report.	High

Table 5.7: Test cases for Reports

5.3.8 Events and Notifications

Test No.	Test Description	Expected Result	Test priority
1	Add/Edit events without valid /required fields	Display “This field is required” message, and prevent submit the form	High
2	Add/Edit events with valid /required fields	Submit the form and display the relevant notification in the very index page only in the relevant user categories.	High

Table 5.8: Test cases for Events and Notifications

5.3.9 Users

Test No.	Test Description	Expected Result	Test priority
1	Select a User	List down relevant User Ids after selecting the user category.	High
2	Email validation	Submit the form after entered an email address in the correct format	High

Table 5.9: Test cases for Users

5.4 Test Data and Test Results

Regular tests were done throughout the development and the testing phase. The results derived from these tests were compared with the results to be expected. These results determined the degree of success of the project. The test results of key test cases are added to Appendix- E along with some screen shots.

5.5 Acceptance Testing

Acceptance testing was carried out at the client site to demonstrate the client that the software is capable of performing the expected tasks. There were several test rounds to fix out the remaining problems along with user training sessions. The certificate received from the client has been appended to Appendix G.

5.6 User Feedback

There were 36 selected teachers and parents who were participated as system users. The overall feedback of the system users is analyzed and is expressed using a chart as given in Figure 5.2. summery of feedback from users is shown 5.3 with the feedback from a questionnaire.

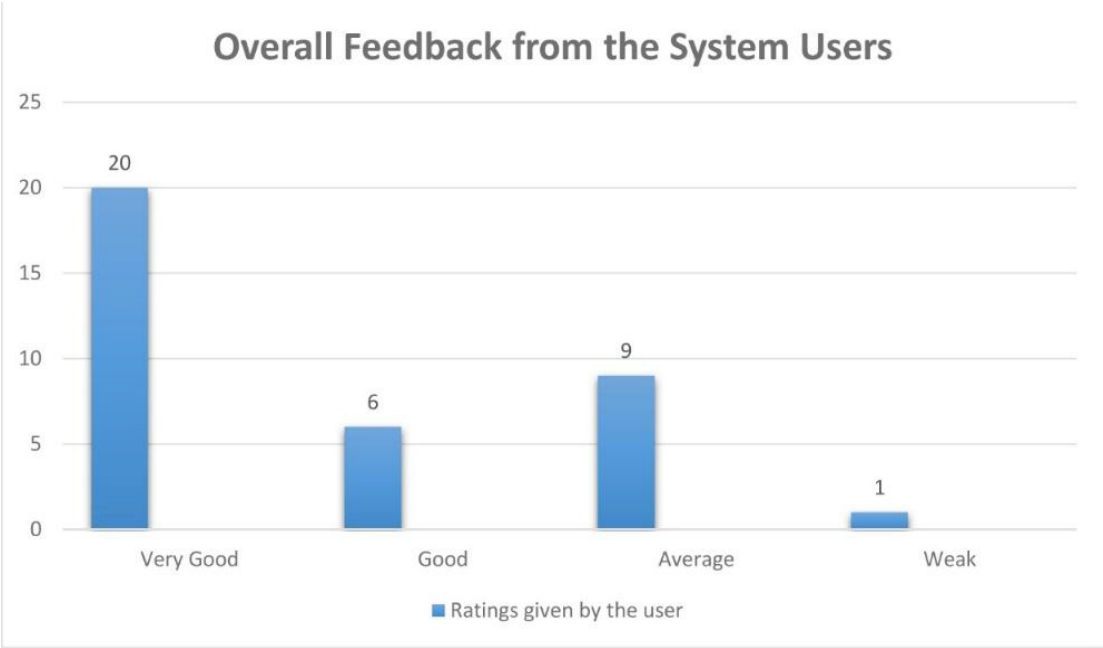


Figure 5.2: Feedback from Users

User Feedback form for Web based system for Future MINDS (AMI) Montessori School

Name: Hashini Karanayake

Designation: Teacher

No	Question	Excellent	Very Good	Good	Average	Weak
1	The design of the system.		✓			
2	Color scheme		✓			
3	Collaboration with the menu and button of the system.	✓				
4	Security and restrictions of the system			✓		
5	Easiness of using the forms of the system			✓		
6	Easiness of using the functionalities of the system.		✓			
7	Overall sense about the system	✓				
	Any Suggestions? It is good to have a system like this to make our work more efficient. This system provides to have a better communication with the principle & parents. It is better if you can improve this further.					

Hashini 28/07/2016

Signature Date

Figure 5.3: Feedback questioner from User

CHAPTER 6 - CONCLUTION

6.1 Introduction

Future MINDS (AMI) Montessori School is influenced and strictly adhered (AMI) Montessori methods in almost every activity they carrying out. Irrespectively they perform so many paper based activities, their tasks were well established and well planned. Therefore their requirement was not a heavy weight common school management system with lots of features. They needed to have technological improvement for their paper backed process with simplicity and yet to make their Montessori out standing among the rest of the native competitors. To accomplish these requirements Web based system for Future MINDS (AMI) Montessori School was presented.

In the system all the students, teachers as well as parents details also saved. Therefore one click of a button could search such details without searching in a heap of files. The system basically supports to enhance administrative tasks, cater to maintain standards and quality by adding uniformity and accuracy for the tasks they performed. Further it facilitate to share notifications among group of users .And for users it also enables perform tasks and view information after remote login. Therefore Future MINDS (AMI) Montessori School is to provide outstanding, better service among the competitive institutions.

6.2 Lessons learnt

Final year project is a remarkable experience enlighten my life which opened a path to gather vast range of knowledge and improve skills. Mainly it really affect me on how to work under pressure yet how to manage work with better time management. This project helped to utilize the lessons learnt in the past three years in the BIT Degree programme in the practical environment. It guided me to search and learn by own which is another advantage obtained. It enhance the knowledge area on concepts and technologies such as PHP, AJAX, Json, JQUERY, CSS, MVC architecture, frameworks and tools. The dissertation itself helped me to improve my skills on how to design and write technical reports.

6.3 Future improvements

- To add a health related module to manage students' health related records.
- To send SMS notifications
- To automatically generate a monthly progress report for all the students and email it to the parents.
- Allow online payments.

References

[1]"Functional requirement", *Wikipedia*, 2016. [Online]. Available: http://en.wikipedia.org/wiki/Functional_requirement. [Accessed: 23- Feb- 2016].

[2]"Non-functional requirement", *Wikipedia*, 2016. [Online]. Available: http://en.wikipedia.org/wiki/Non-functional_requirement. [Accessed: 24- Feb- 2016].

[3]"Software development process", *Wikipedia*, 2016. [Online]. Available: http://en.wikipedia.org/wiki/Software_development_process. [Accessed: 13- Mar- 2016].

[4]"I", *Umsl.edu*, 2016. [Online]. Available: http://www.umsl.edu/~sauterv/analysis/488_f01_papers/quillin.htm#2. Cackowski, David, Object Analysis in Organizational Design: A solution for matrix organizations, Project Management Journal, September, 2000. [Accessed: 25- Mar- 2016].

[5]"MVC Framework Introduction", *www.tutorialspoint.com*, 2016. [Online]. Available: http://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm. [Accessed: 25- Mar- 2016].

[6]"UML - Class Diagram", *www.tutorialspoint.com*, 2016. [Online]. Available: http://www.tutorialspoint.com/uml/uml_class_diagram.htm. [Accessed: 28- Mar- 2016].

[7]"Activity diagram", *Wikipedia*, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Activity_diagram. [Accessed: 28- Mar- 2016].

[8]"Sequence diagram", *Wikipedia*, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Sequence_diagram. [Accessed: 28- Mar- 2016].

[9]"CodeIgniter", *Wikipedia*, 2016. [Online]. Available: https://en.wikipedia.org/wiki/CodeIgniter. [Accessed: 20- Apr- 2016].

[10]2016. [Online]. Available: https://en.wikipedihttps://en.wikipedia.org/wiki/Bootstrap_front-end_framework [Accessed: 20- Apr- 2016].

[11]j. jquery.org, "Datepicker Widget | jQuery UI API Documentation", *Api.jqueryui.com*, 2016. [Online]. Available: https://api.jqueryui.com/datepicker/. [Accessed: 25- Apr- 2016].

[12]"Date Range Picker for Bootstrap", *Daterangepicker.com*, 2016. [Online]. Available: http://www.daterangepicker.com/. [Accessed: 25- Apr- 2016].

[13]"PHPMailer", *Wikipedia*, 2016. [Online]. Available: https://en.wikipedia.org/wiki/PHPMailer. [Accessed: 26- Apr- 2016].

[14]"The 15 Best JavaScript Charting Libraries", *SitePoint*, 2015. [Online]. Available: https://www.sitepoint.com/15-best-javascript-charting-libraries/. [Accessed: 26- Apr- 2016].

[15]"DataTables | Table plug-in for jQuery", *Datatables.net*, 2016. [Online]. Available: https://datatables.net/. [Accessed: 28- Apr- 2016].

[16]R. Smaizys, "mPDF: HTML to PDF introduction | Richard Smaizys", *Smaizys.com*, 2011. [Online]. Available: http://www.smaizys.com/php/mpdf-html-to-pdf-introduction/. [Accessed: 28- Apr- 2016].

[17]"System Testing: What? Why? & How? | Software Testing Class", *Softwaretestingclass.com*, 2016. [Online]. Available: http://www.softwaretestingclass.com/system-testing-what-why-how/. [Accessed: 26- Jun- 2016].

Appendix A – System Documentation

This documentation is to help the interest parties who wish to enhance this project work, to install and configure the system in technical perspective. The system documentation can be referred if there are any changes to be made in the Web based system for Future MINDS (AMI) Montessori School. In order to install the system, the device chosen for installation should meet the following prerequisites of hardware and software.

Hardware Requirements

Hardware	Minimum Requirement
Processor	Intel core 2 Duo or similar or newer Processor.
Memory	1GB RAM or more
Hard Disk	1GB free disk space or Higher.
Printer	Dot-matrix printer or Ink jet printer or Laser Printer
Internet	Minimum 512kbps ADSL connection.

Table A.1:Minimum Hardware Requirements

Software Requirements

Hardware	Minimum Requirement
Operating System	Microsoft Windows XP/Vista/Windows7 /Windows 8
Bundle Package Software	WAMP Server 2.0
Code Editor	Net Beans IDE 8.0. and Sublime Text or suitable editor.
Web Browser	Google Chrome/ Fire Fox/ Internet Explorer

Table A.2: Minimum Software Requirements

The system installation has three stages

1. Required software installations
2. Database installation
3. System installation

Required Software Installations

- Installing WAMP Server

Download and install WAMP Server windows version from http://www.wampserver.com/en/ . Give installation path to C:\ of the computer. Please

refer the installation guide provided in the above website before installing WAMP on your computer

Figure A.1: WAMP Server download page

- Installing Google chrome, Firefox web browsers

Download and install the latest version of Google chrome from www.google.com (set as a default browser), Mozilla Firefox from www.mozilla.com.

Database Installation

- Open your web browser and type http://localhost/ or http://localhost:8080/
- Select phpMyAdmin tool from the left side panel of the WAMP home page.
- In the phpMyAdmin window click in the text field named “Create new database” and type “future-minds” and press “Create” button. A new database named “future-minds” will be created.
- Now click on the “Import” tab from the tabs located in the top of the window and click on the “Chose file” button located next to the “Location of the text file” option in “File to import” section. It will give you a browsing window.
- Insert the “Transport Management System” CD in to your CD-ROM. Locate and select the “future-minds.sql” from the path: X:\ Transport Management System\Database\ future-minds.sql (Note: “X” is your CD-ROM Drive letter) and press “Open” button.
- Press “Go” button located in the bottom right hand corner

System Installation

After installing the relevant softwares, copy the “future-minds” folder given in the supplementary CD and paste it inside ‘www’ folder in the path - C:\wamp\www.

Launching the System

- Go to start Programs WAMP for Windows.
- Now click on “future-minds” under “Your Projects” in the window.

Appendix B – Design Documentation

Use Case Diagram

- Use case diagram for Registration Process

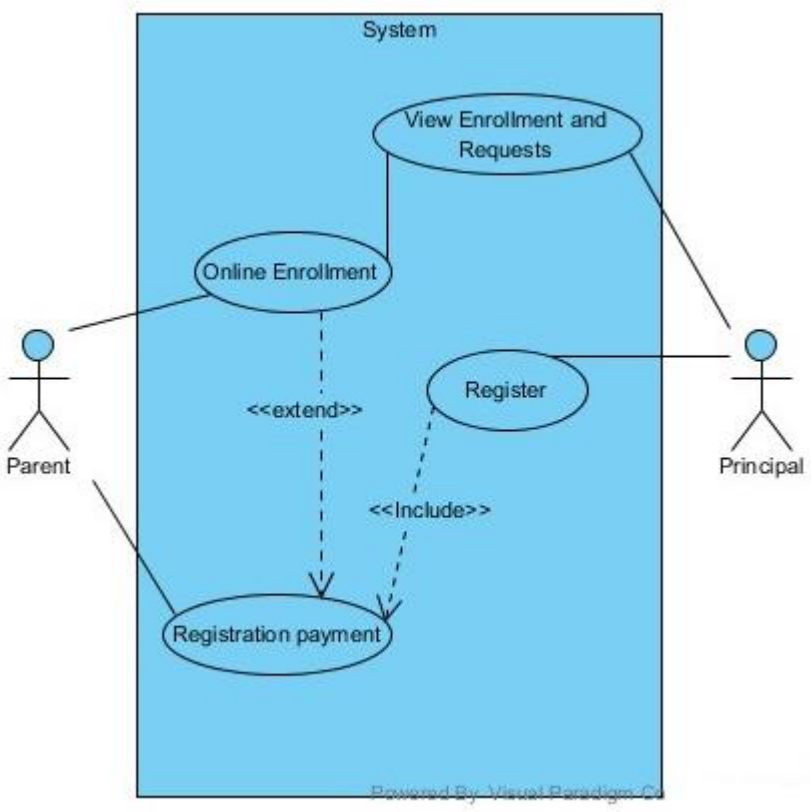


Figure B.1: Use Case Diagram for Registration Process

Use Case Name	Registration Process
Actors	Parents Principal
Description	Registering to the Montessori via system
Precondition	Principal should login to the system
Typical Course of Events	Parent who wish to enroll a child who was not there in the Montessori can visit to the web site and fill the New Admission form and submit. Principal can view the requests. After making register payments principal will register the child.
Post Conditions	Child details will be saved within the system as a student and can do further proceeds.

Table B.1: Use case narrative for Registration Process

- Use case diagram for Events and Notifications Module

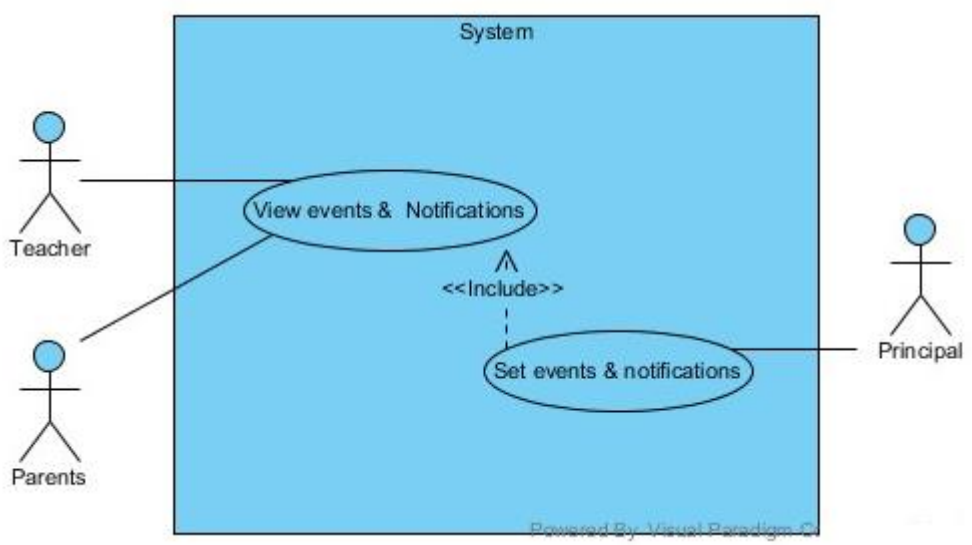


Figure B.2: Use Case Diagram for Events and Notifications Module

Use Case Name	Events and Notifications Module
Actors	Teacher Parent Principal
Description	Setting Notifications to parents and teachers
Precondition	Principal, Parent and Teacher should login to the system.
Typical Course of Events	Principal set notifications to selected groups of users, teachers and parents. When a particular user login to the system user can see new messages in the dashboard.
Post Conditions	The events and notifications will be saved within the system and can be viewed.

Table B.2: Use case narrative for Events and Notifications Module

Activity Diagram

Following diagrams depicts the activity diagrams for lesson planning and student registration process.

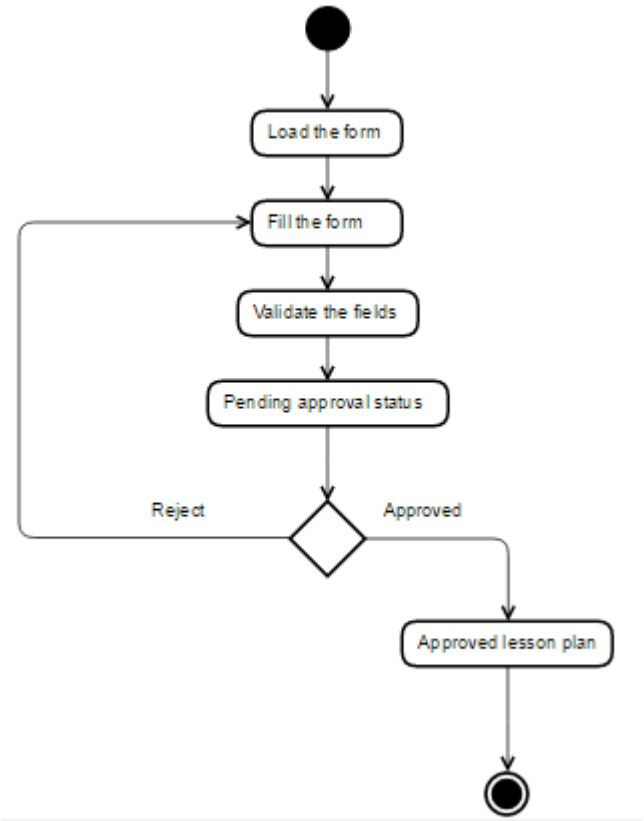


Figure B.3: Activity Diagram for Lesson Planning Module

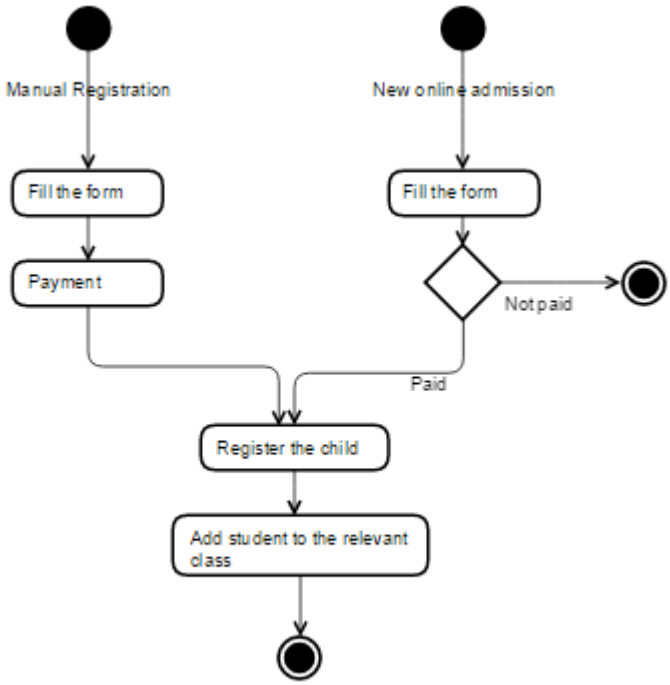


Figure B.4: Activity Diagram for Registration Process

Sequence Diagram

Figure B.5 depicts the sequence diagram for the process of restoring deleted Class Teacher record to the functions again.

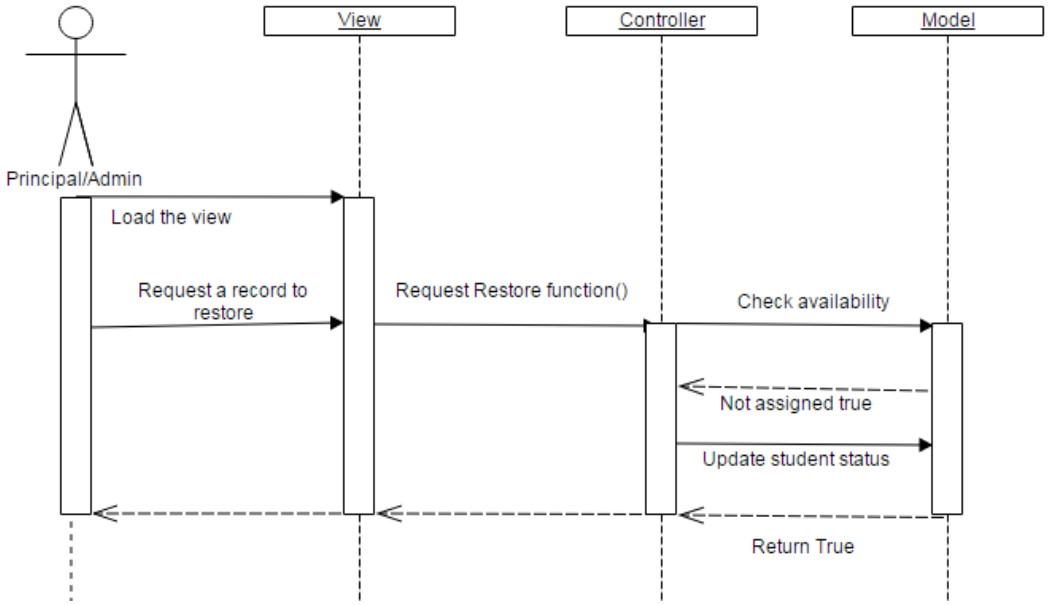


Figure B.5: Sequence diagram for the Process of restoring deleted Class Teacher

Appendix C – User Documentation

The system has three types of user roles, namely Administrator-the Principal, Teacher and Parent. Only the Administrator has all the rights to operate all the modules, and to explain all the module functions in the system, the documentation is compiled for the users who hold administrative privileges .and selected few function from teachers privileges which is not in administrator- principal module. The system has a separate views for parent module, teacher module. But they are not explained here.

This is the login page of the system. Any user can log via this page for whom the login is allowed. Figure C.1 depicts the login form for the system.

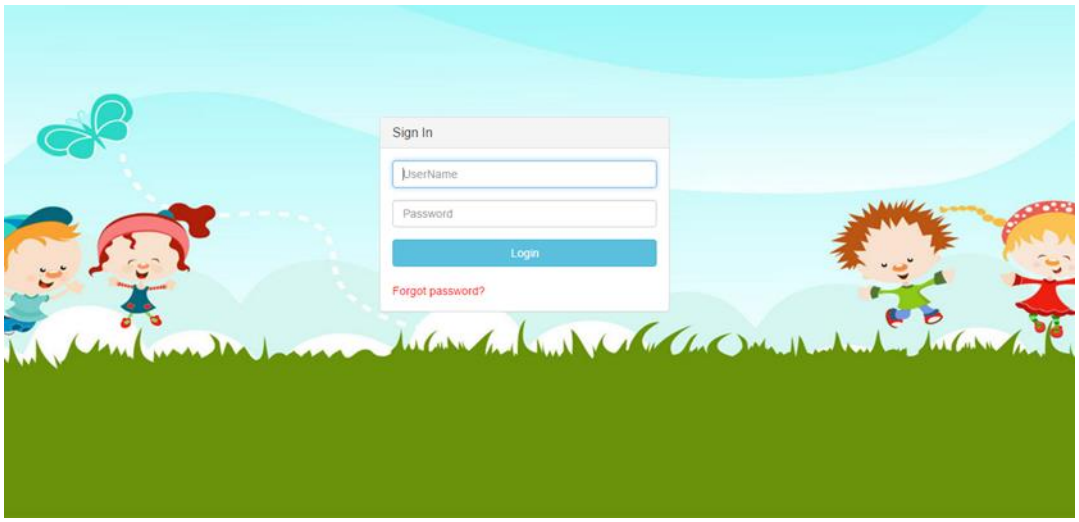


Figure C.1:The Login page

After providing the correct username and password user gain the authorized access to the system. Once logged in, the user is directed to dashboard. Figure C.2 depicts the dashboard for the Administrator – Principal.

Figure C.2:Administrator's Dashboard

Logged user name will be displayed in the left most corner. Logout link from the system is provided in the right most corner.

In the left side of the dashboard, navigation bar is shown. All the modules of the system can be browsed through tasks block and navigation bar. Sub menus are included in menu in navigation bar.

In the right side of the dashboard, all the events and notifications that is for the user are listed down.

Personal Detail Module

This module is used to register students, parents as well as teachers data to the system. A student is registered by providing student information via the registration form. Parent details also can be capture during the student registration. Click the save button to save the record to the database. Part of the registration form of the student depicted in Figure C.3.

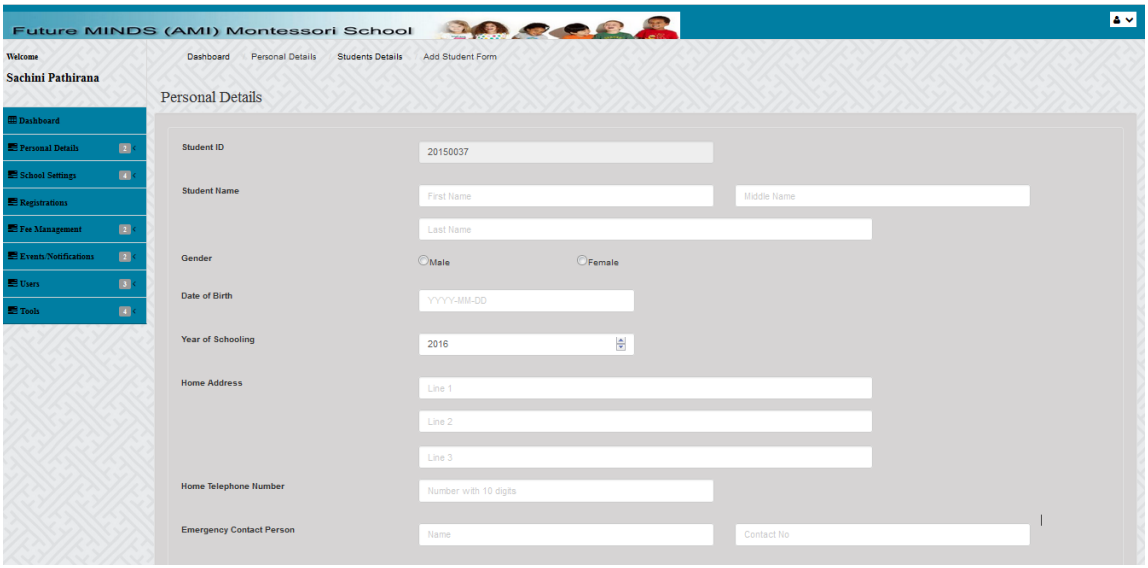


Figure C.3:Registration Form

After successful registration of a students, particular student record listed in a table which then allowed to view, edit, remove it, when required. Each of these can be done by the clicking of the relevant button under "Action" column in the table. The screen will be moved to relevant form view which is for view, edit, or to delete the student information after above button click. So as the Teachers registration section.

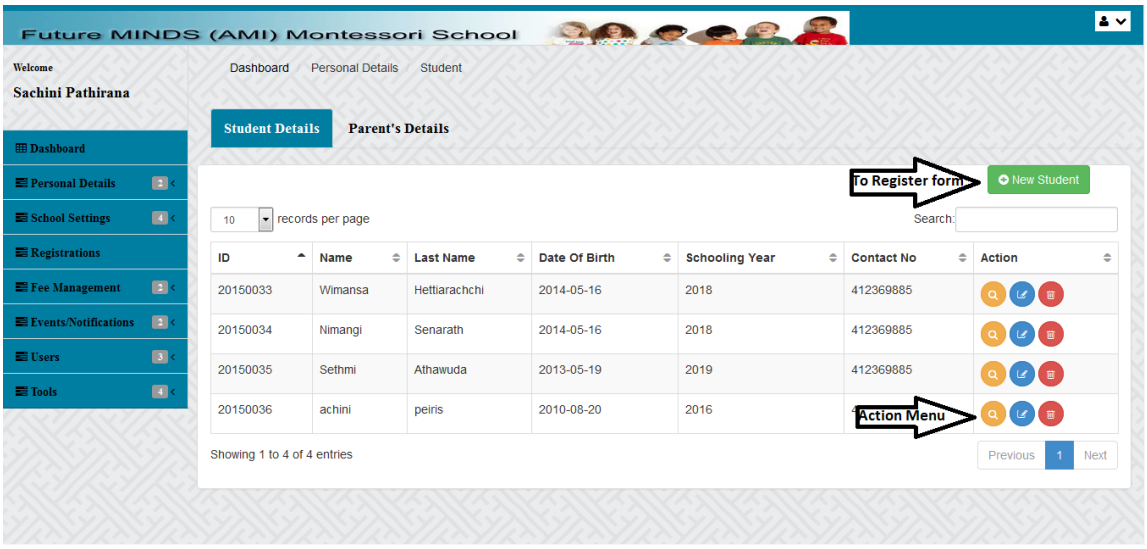


Figure C.4:Student Details Index Page

Initial Set Up

School settings-> Initial Set Up

This section is a very important part of the section which is strictly allowed only to the administrator. This section is concerned about the basic set for the school.

- "Manage Grade" section allows to add grades for the Montessori such as "Play group", "Upper Kindergarten" which is common to any year.

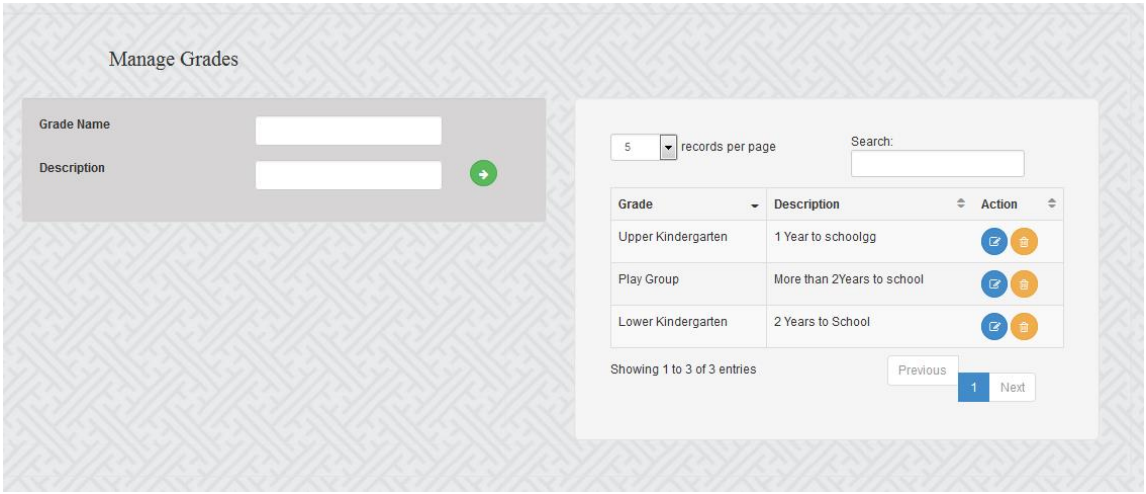


Figure C.5:Section to insert Grades to the Montessori

- "Select Grade for the Year" concerned about what are the grades which is going to be allowed in a particular year. This may be differed based on year basis. This should be first selected before adding classes to the Montessori.

  Enter the academic year -> Select a grade from drop down menu -> Submit

  Grades maintained in each year are listed in the table shown in the right side.

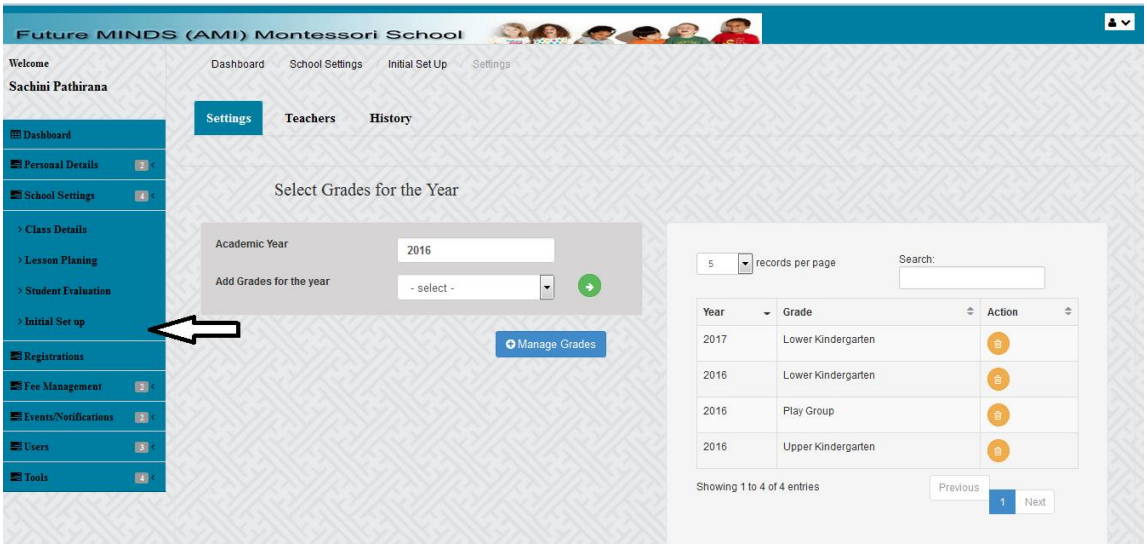


Figure C.6:Section to add a grade for a Year

- After selecting grades for a particular year, then classes should be added. Each year there are more than one class for every grade, and therefore very first class's default name should be assigned as "A". User cannot proceed other functions without having a name to the class. Class name can be edited using the form receiving after clicking the "Edit" button in the table. See figure.

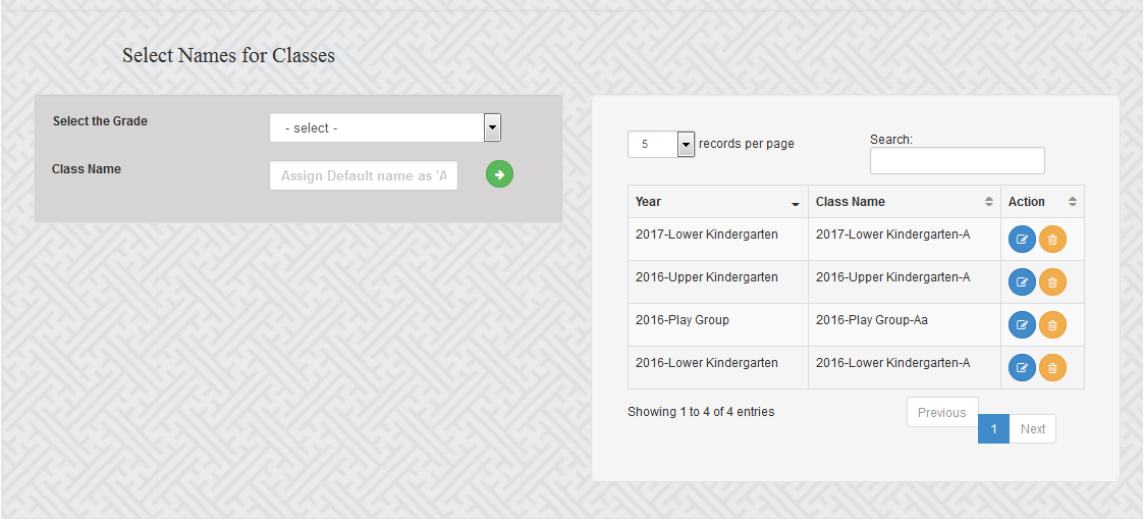


Figure C.7:Section to add Classes for a Grade

- Subject is added to each grade which have been entered for particular year.

Select a Grade and Year from drop down menu -> Enter subject -> Submit

Subject name can be edited using the form receiving after clicking the "Edit" button in the table.

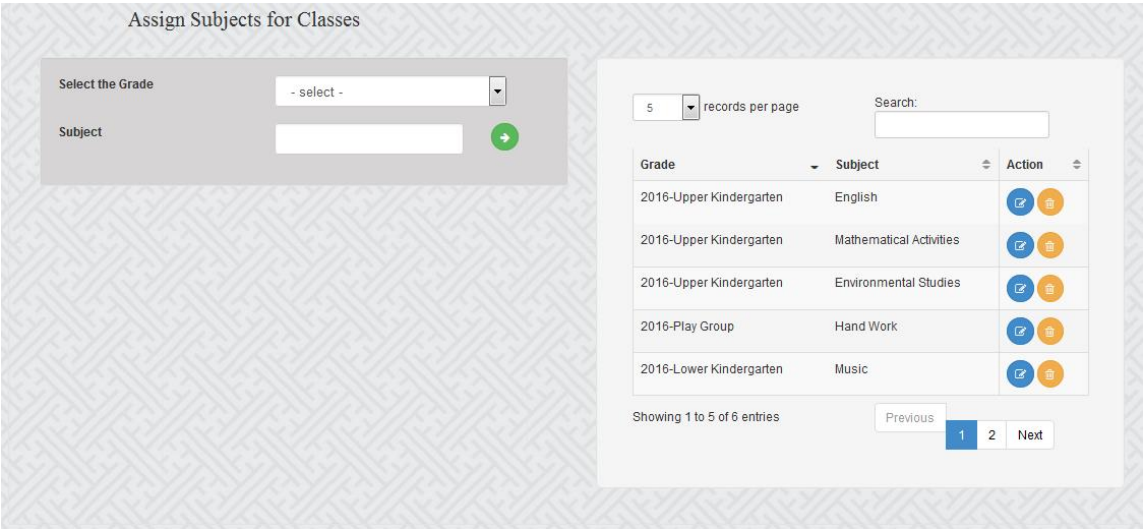


Figure C.8:Section to add Subjects for a Grade

Appendix D - Management Reports

The report statistics helps the management to get summary of the school such as student details, students improvement on subjects and other statistical information. These reports are given in PDF format which helps the user to easily browse, view and print the reports when needed.

Following Figure D.1, D.2and D.3 depicts a few sample reports generated by the system.

IFRD

Academy of International Education & Business Development

Future Minds
English Medium Montessori & Day Care Center

Student Evaluation Report

Student Name : Wethmi Gunawardane

Class 2016-Upper Kindergarten - A

Subject	Activity	Level
Mathematical Activities	Count 1 to 10	Very Good
Mathematical Activities	Identify Numbers 1 to 10	Average
Music	Song "This Old Man"	Very Good
English	Rhyming Alphabet	Good
Handwork	Clay work	Very Good
Handwork	Making "Wesak Card"	Excellent

Figure D.1: Student Evaluation Report

Academy of International Education & Business Development

Future Minds
English Medium Montessori & Day Care Center

TRANSACTION SUMMERY

Report Date 10/03/2016

Tran ID	Payment Time	Child ID	Amount	Log details
16	Monthly Fees	20160023	3000.00	2016-02-17
17	Renewal of Registration	20160032	5000.00	2016-02-18
18	New Admission	20160045	6000.00	2016-02-19
19	Misc Fees	20160023	1500.00	2016-02-21
20	Misc Fees	20160023	1000.00	2016-02-25
21	Monthly Fees	20160014	3000.00	2016-03-08
22	Monthly Fees	20160010	6000.00	2016-03-08
23	Monthly Fees	20160011	3000.00	2016-03-08

Figure D.2: Transaction Summery Report

Academy of International Education & Business Development

Future Minds
English Medium Montessori & Day Care Center

Class Details

Class 2016-Upper Kindergarten - A

Report Date: 27/08/2016

ID	Student Name	Contact No
20160010	Oshini Weerakkodi	0718194696
20160012	Wimansa Weerathunga	0412224567
20160018	Wethmi Gunawardane	0777397065
20160019	Chirani Sehansa Abeywardane	0412245523
20160023	Ishika Nimsarani Wijepala	0709873543
20160025	Suvini Darmapala	0415603245

Figure D.3: Class Details Report

Appendix E – Test Results

User Login

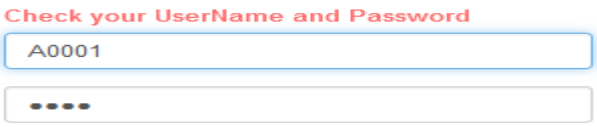
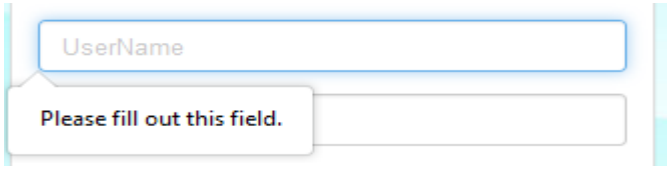
Test No	Test Data Inputs and Actual Results	Status
1	Invalid username or password	Pass
		
2	Empty the username or password	Pass
		
3	Display logged username	Pass
		

Table E.1: Test results for User Login

Student/Teacher Registration (Personal Details module)

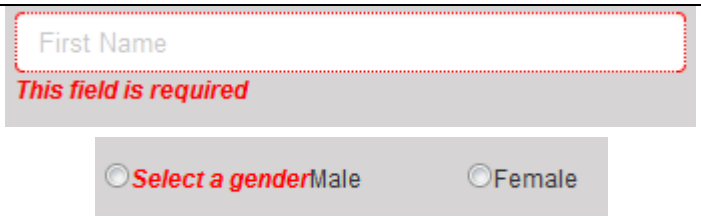
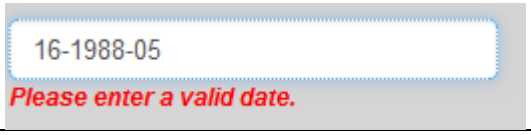
Test No	Test Data Inputs and Actual Results	Status
1	Empty the required fields	Pass
		
2	Insert invalid details to the fields (Ex wrong date format)	Pass
		

Table E.2: Test results for Student/Teacher Registration (Personal Details module)

Lesson Planning Module

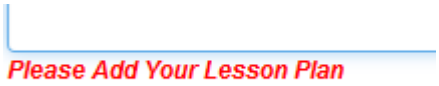
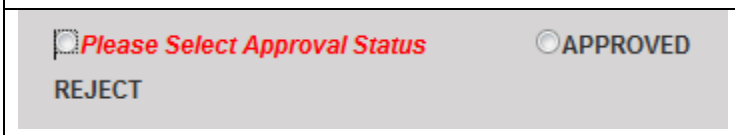
Test No	Test Data Inputs and Actual Results	Status
1	Display the relevant interfaces and the messages when add/edit LessonPlans	Pass
		
2	Admin is to comment without selecting an "Approval status".	Pass
		

Table E.3: Test results for Lesson Planning Module

Evaluation Module

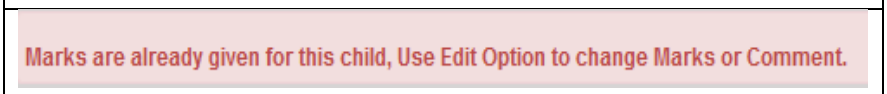
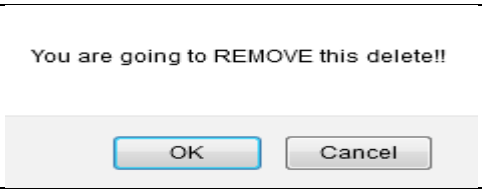
Test No	Test Data Inputs and Actual Results	Status
1	Trying to a level for already marked student.	Pass
		
2	Click Delete Activities	Pass
		

Table E.4: Test results for Evaluation Module

Initial Set Up Module

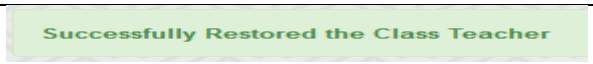
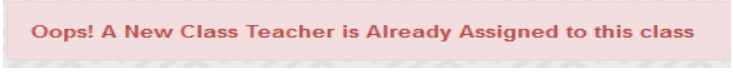
Test No	Test Data Inputs and Actual Results	Status
1	Restore a deleted, class/subject teacher which is still not assigned a new class/subject teacher.	Pass
		
2	Restore a deleted, class/subject teacher which is already assigned a new class/subject teacher.	Pass
		

Table E.5: Test results for Initial Set Up Module

Appendix F - Code Listing

Since the codes are very lengthy, only important code snippets are represented in this segment apart from the codes included in implementation chapter.

The system has developed under the MVC design pattern, codes related to each function, comprises of separate Model View and Controller.

Following code segments shows the how the events and notifications module is implemented under MVC pattern.

The following code section shows the code in View for form to insert new event.

```
<form action="<?php echo base_url(); ?>index.php/events/add_event"
id="add_user" method="POST" >
       <div class='row'>
              <div class='col-sm-5'>
              <label  class="control-label"> Target Group</label>
              </div>
              <div class='col-sm-5'>
              <select class="form-control" id="target_group"
name="target_group" required="">
                  <option value="">-- Select--</option>
                  <option value="1">ALL</option>
                  <option value="2">TEACHERS ONLY</option>
                  <option value="3">PARENTS ONLY</option>
               </select>
               </div>
        </div>
     </br>
        <div class='row'>
               <div class='col-sm-5'>
               <label  class="control-label"> Header</label>
               </div>
               <div class='col-sm-5'>
               <div class='form-group'>
               <input type="text" id="header" name="header"
class="form-control" required=""/>
               </div>
               </div>
         </div>
         <div class='row'>
               <div class='col-sm-5'><label  class="control-label">
Description</label></div>
               <div class='col-sm-5'>
               <div class='form-group'>
               <textarea id="des" name="des" rows="5" cols="100"
class="form-control"></textarea>
               </div>
               </div>
          </div>
          <div class="row">
               <div class="col-sm-offset-11">
                <div class="col-sm-3">
```

```
                    <button class="btn btn-success btn-sm pull-right"
type="submit"> SUBMIT </button>
                     </div>
                     </div>
                </div>
</form>
```

The following code section shows the code in Controller to add new event

```
public function add_event() { //add new event
  if ($this->session->userdata('logged_in')) {
      $session_data = $this->session->userdata('logged_in');
      $group_id = $session_data['group_id'];
      $username = $session_data['username'];

      // inner if
      if ($group_id === "10") {
     //get user info
      $fields = "*";//get teacher info
      $whereArr=array("user_id" => $username);
      data['userData']=$this->Generic_model->getData($fields,
'admins',$whereArr);

       $date = date('Y-m-d');
       //insert new event
       $dataArr = array(
       target_group" => $this->input->post("target_group"),
       "add_date" =>$date,
       "added_by" => $username,
       "header" => $this->input->post("header"),
       "description" => $this->input->post("des"),
                                        );
       $this->Generic_model->insertData('events', $dataArr);

redirect(base_url().'index.php/events/manageEvents');
   } else {
   $this->load->view('admin/accessDeniedPage');
            }
 //outer else
} else {
redirect('login/index');
       }
}
```

The following code section shows the code in Model, Generic_model for few basic crud operations.

```
class Generic_model extends CI_Model {

    function insertData($tablename, $data_arr) {
        $result = $this->db->insert($tablename, $data_arr);
        return $result;
             }

    public function getData($fieldset, $tableName, $where = '') {
        if ($where == "") {
            $this->db->select($fieldset)->from($tableName);
        } else  {
            $this->db->select($fieldset);
            $this->db->from($tableName);
            $this->db->where($where);
//              $this->db->where($where2);
        }
        $query = $this->db->get();
        return $query->result();
                 }
}
```

Client-Side Validation

Each view form values are validated through JavaScript (Client-side). Following code snippet shows how some of the client-side validations have been done for above form.

```
<!-- Form Validation : -->
        <script>
            $(document).ready(function () {
                $("#add_user").validate({
                    rules: {
                        target_group: {
                            required: true,
                       },
                        des: {
                            required: true,
                                 },
                        header: {
                            required: true,
                           },
                        },
                    messages: {
                        header: {
                            required: "This field is required",
                         },des: {
                            required: "This field is required",
                       },target_group: {
                            required: "Select a target group",
                        }
                    }
                });
            });
        </script>
```

Appendix G - Client Certificate

Academy of
International Education
& Business Development
15, School Lane, Rahula Rd, Matara.
Tel. 041 2225243 / 071 8359158

Future MINDS
English Medium
Montessori
AMI Method

24.09.2016

The Project Examination Board
University of Colombo School of Computing
Colombo 7.

Project – Web based System for Future MINDS (AMI) Montessori School, Matara

This is in regard with the above subject as Ms.S S W Vidanapathirana has done a web based system for the Future MINDS (AMI) Montessori School as her final year project.

She has gathered the necessary data and information from us in order to develop the said system and it is in the process of testing.

The management of Future MINDS (AMI) Montessori School has decided to accept the system to be used after completion of necessary testing.

This letter is issued on request of Ms.Vidanapathirana.

Academy of International Education & Business Development /
Future MINDS (AMI) Montessori School

Gayathri Wickramasekara
Managing Directress

IELTS Registration Centre for British Council in Matara |
British Council Registered Centre to conduct University of Cambridge-UK English Courses |
Registered Centre for Institute of Western Music & Speech-Sri Lanka
Future MINDS (AMI) Montessori School

Glossary

AJAX (Asynchronous JavaScript and XML) - is a group of interrelated web development methods used on the client-side to create interactive web applications. It is the art of exchanging data with a server, and update parts of a web page without reloading the whole page.

CSS (Cascading Style Sheet) - is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a mark-up language.

Codeigniter - is an Application Development Framework

Internet - is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide.

JavaScript - This is an object-oriented scripting language, which supports prototyping

JQuery - is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML

Local host - This is the default name describing the local computer address.

Object Oriented Development - is a standard approach to software development based on objects and its instances

PHP (Hypertext Preprocessor**) -** which is a powerful server side scripting Language

Unified Modelling Language (UML) - is a standardized general-purpose modelling language in the field of object-oriented engineering. This includes a set of graphic notation techniques to create visual models of object-oriented systems

WAMP - This is a free, open source and cross-platform web server solution Package

Index

T

U

W