Student Management System

For

Joseph Vaz Institute

K.G.R.Aruna Udara

BIT Reg No: r141660

BIT Index No: 1416601

Name of the Supervisor: Mr. G.G.N.K.Gallage

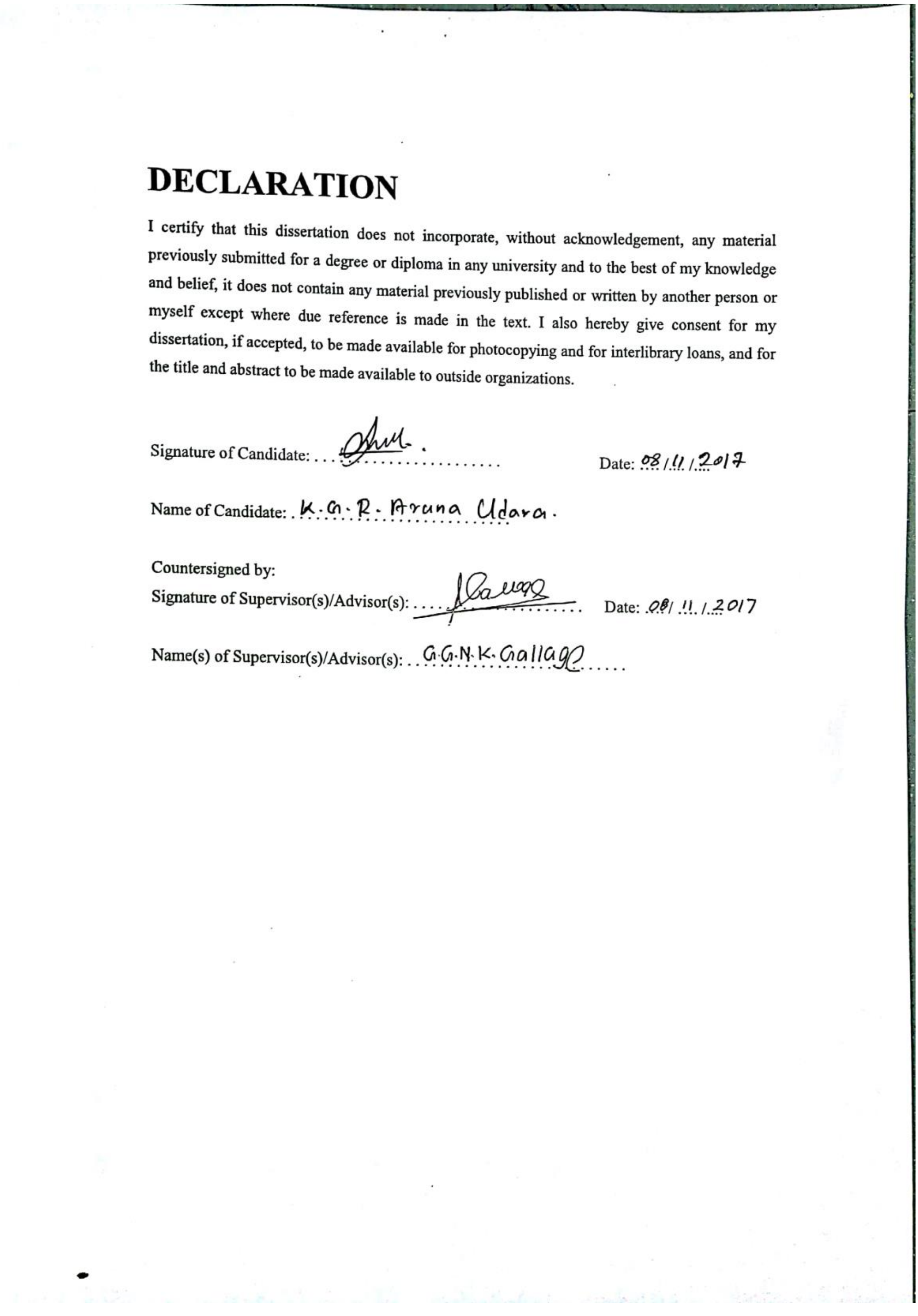
2017

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

DECLARETION



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 organizations.

Signature of Candidate: .... Date: 08/11/2017

Name of Candidate: K.G.R. Aruna Udara.

Countersigned by:

Signature of Supervisor(s)/Advisor(s): .... Date: 08/11/2017

Name(s) of Supervisor(s)/Advisor(s): G.G.N.K. Gallage

ABSTRACT

Developments in information technologies have been impacting upon educational organizations. Student Management system for Joseph Vaz Institute (JVISMS) is the approach taken by the Director of Joseph Vaz Institute to overcome the difficulties in current manual administrative process such as keep and analyzes historical data and report generation etc.

The JVISMS is to computerize the Institute Information. The Front end developed using HTML, CSS, JAVASCRIPT, BOOTSTRAP and the Back End uses MySQL Server. JVISMS was developed as five subsystems namely; Teacher Management, Student Management, Payment Management, Report Management and Security Management. These subsystems provide support in managing the organization efficiently. The time consumption in managerial processes will absolutely decrease. Furthermore, human resources of the school can be managed effectively. One of the most and important feature of the proposed system is that it is very much user friendly and accurate. Hence the system users feel much comfortable to interact with it. The modules are being regularly updated therefore the data extracted will be up to date & more accurate.

The proposed System will improve the efficiency of administrative services. Also the JVISMS can help to access, manage & report the information quickly and easily.

ACKNOWLEDGEMENT

I would like to use this opportunity to convey my sincere gratitude to every personality who has helped me during each and every step of my project's life cycle. First of all I wish to thank my project's supervisor Mr. G.G.N.K.Gallage for guiding me to handle all the tasks properly and specially for encouraging me to face strongly in some complicated situations occurred during the time of developing this project. All the advices and the support he has given me was one of the main important factor to complete the project successfully on time. And also I would like to express my grateful thanks to all the lectures at Esoft Metro Campus for brushing up my knowledge throughout the BIT degree program. The Director of Joseph Vaz Institute, Rev. Fr. Sagara Prishantha was another most important supportive character for me who has given a big cooperation throughout the project work and all the other staff members, members' of the management section, parents, students and each and every one in the Institute for helping me a lot to gather requirements as well as to clarify my doubts specially in requirements analysis phase will be highly appreciated.

Also, I like to express my gratitude for the BIT coordinators and all other academic staff members of University of Colombo School of Computing for all the guidance and support given during the project development and throughout the whole degree program.

My parents are the ones who were with me throughout this degree program by facilitating each and every resource I required and they played a vital role to produce a successful outcome as well.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

BIT - Bachelor of Information Technology

JVI – Joseph Vaz Institute

JVISMS – Joseph Vaz Institute Student Management System

ER - Entity Relationship

ERD - Entity Relationship Diagram

GB - Giga Byte

GUI - Graphical User Interface

HTML - Hypertext Markup Language

CSS – Cascade Style Sheet

IDE - Integrated Development Environment

IP - Internet Protocol

JDK - Java Development Kit

OOP - Object Oriented Programming

PC - Personal Computer

PDF - Portable Document Format

RAM - Random Access Memory

ROM - Read Only Memory

RUP - Rational Unified Process

SIMS - School Information Management System

SMS - Short Messaging System

SQL - Structured Query Language

UCSC - University of Colombo School of Computing

UML - Unified Modeling Language

CHAPTER 1 - INTRODUCTION

1.1 MOTIVATION FOR THE PROJECT

Current manual system of Joseph Vaz institute keeps written information of Teachers, Students, Nonacademic staff with many deficiencies such as hand writing errors, calculation errors etc. With the manual system it takes a long time to filter information from written documents while it leads to problematic conditions on decision making and report generation.

The workload of current manual system is very high, therefore outdated information can be arise. Hence there is a greater flexibility for inaccurate and incomplete reports.

The manager of institute have to gather and analyze information continuously and make decisions in a short time while it is not possible with the current manual system.

The privacy of teachers and students is at a level of risk because, not like a password protected database, written documents stored in cupboards can be access by others easily.

1.2 OBJECTIVE AND SCOPE OF THE PROJECT

The main objective of developing JVISMS is to reduce the paper work and save time in student management. There by the efficiency will be increased while decreasing the work load. The project provides us the information about students, teachers, and student payment and student attendance records to maintain absentees. Reports should be generated automatically and should be available to view, print or export by the system administrators with ease.

1.2.1 SCOPE IN A SUMMARIZED FORMAT

After considering the client requirements the scope of the system was decided the following main procedures,

- Manage personal information of Students, Parents and Teachers. This is one of the most required tasks after registering users.
- Handle registration of the system’s users.
- Manage attendance of students.
- Generate report such as details of all the students, Payment details etc.

- Payments Details Management.
- Generate follow-up message for student

1.2.2 GENERAL OBJECTIVE

The general objective of the project is to automate the manual Student Management System of Joseph Vaz Institute.

1.2.3 SPECIFIC OBJECTIVES

In order to attain the general objective, the following list of specific objectives was set:

- To develop a Web base teacher & student registration system,
- To produce a timetable,
- To facilitate attendance record keeping,
- To facilitate payment record keeping,
- To facilitate various report generation,
- To allow teachers, school community (with the support of system administrator, probably the Manager) to view reports on students,
- To track students records,
- To search students and teachers quickly,
- To generate official letters with the letter head of the institute for teachers and students,
- To ensure the data recoverability and security with scheduled backups,

1.3 DEVELOPMENT ENVIRONMENT

This project is implemented in a Microsoft Windows 10 environment with the help of languages such as PHP, BOOTSTRAP JavaScript, Jquery, Ajax and databases are handled in MySQL Server 5.6.24 in Xamp Server Version 3.2.1.

CHAPTER 2 – ANALYSIS

2.1 FACT GATHERING

The success of any project is dependent upon the accuracy of available data. Accurate information was collected with the help of certain fact finding techniques. The techniques used for fact finding process were; interview, observation, document review and scenarios.

Selecting the suitable people to be interviewed was a difficult task. The people selected to be interviewed should have sufficient experience with the current system. Therefore some of the administrative members, involved in the current manual system, were interviewed. Selected people were asked for a suitable place and time which will be comfortable for them and carried out the interview. Their responses were recorded and discussed about the future requirements that can be occurred.

At the time of interview a questionnaire was developed as a mechanism of data collection. In the first section of the questionnaire, there were some queries about the personal information of the school managers such as vocational experience and education level. In the second section, the queries were about information technology facilities related to the information systems of the Institute and attitudes of the institute managers about technological facilities. In the third section of the questionnaire, there were queries related to the studies done with the Student Management Systems and items about by whom these studies were done. These queries were divided into two sections as the preparation of various documents, lists and statistics, and data entry. The fourth section consisted of the contributions of managing management systems to institute management.

To get an idea about the flow of documents, working and working load of the existing manual system, the users of the system, a direct visit and an observe was carried out. The best way to analyze the existing manual system was to collect facts from existing documentation rather than from human sources. There were various kinds of documents to collect facts from existing documents. These include: record books, registration forms, manually created reports etc.

2.2 Fact Finding Techniques Used

- Interviews
- Observation
- Analyzing sample reports
- Discussions

2.2.1 Interviews

Interviewed randomly selected parents, teachers and students in-order to get a clear idea about their opinion about the current situation of the school and the proposed system. To clarify doubts about management procedures interviewed people at the managerial level of the school as well. This requirement gathering technique helped to have an overall idea about the user’s requirements.

2.2.2 Observation

Specially, observed activities related to management staffs that are doing the most important procedures like registering students, handling payments details etc. With the help of observed results could catch up the difficulties they face due to lack of a proper system and the difficulties occur during with lots of paper based manual writing activities.

2.2.3 Analyzing sample reports

Referred to some sample payment receipts, attendance records, and other distributed reports including messages, marks in order to have a full coverage about the format they expect and the methods they use.

2.2.4 Discussions

Many discussions have been arranged with different user categories in-order to grab their requirements and it was carried out within a user friendly environment.

2.3 REQUIREMENT ANALYZING

To produce a model of the system which is correct, complete and consistent it is necessary to construct the analysis model which focuses on structuring and formalizing the requirements of the system. “The analysis model describes the structure of the system or application that you are modeling. It consists of class diagrams and sequence diagrams that describe the logical implementation of the functional requirements that you identified in the use case model” [1]. For the purpose of this project the analysis model is described in terms of Use case model and class diagram.

The whole system is designed with five separate subsystems (see figure 2.1); Student Management, Teacher Management, Payment Management, Security Management and Report Management. Relationships between the subsystems are illustrated using a use case diagram. Manager, Branch manager, Teacher and Trainer are the main actors of the system.

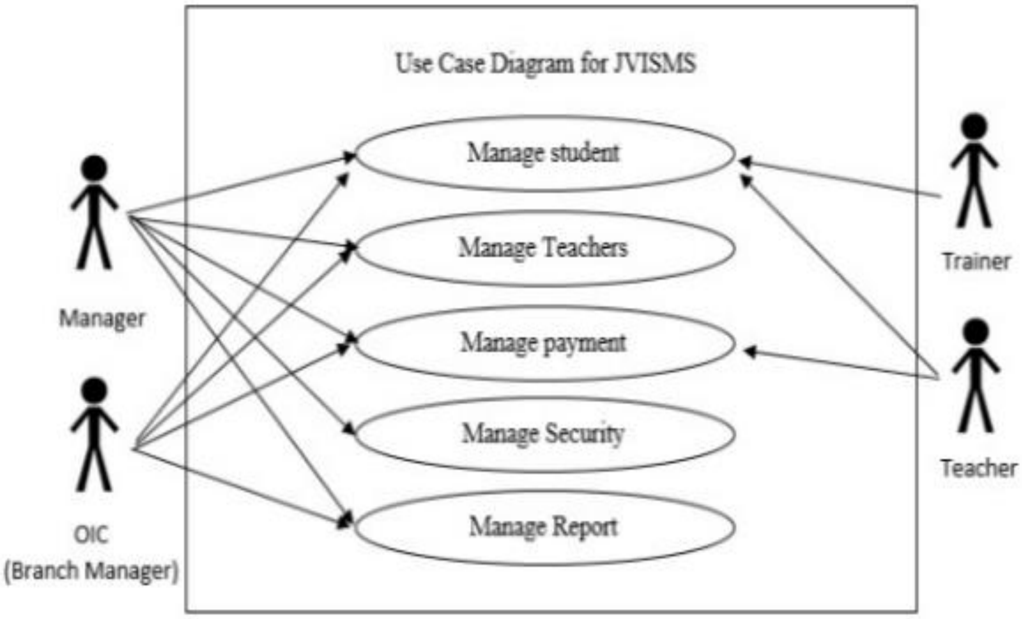


Figure 2.1 Main use case diagram of the JVISMS

Actors of the system

Name: Manager (Administrator)

Description: A Manager is a person who involved in all activities of the Institute

Name: OIC/Branch Manager

Description: A Branch Manager is a teacher assigned by the Institute Manager to handle Institute branch. She/he has the responsibility of maintaining all branch responsibilities.

Name: Teacher

Description: A Teacher is a teacher assigned by the Institute Manager to each class of students to follow the students closely. She/he has the responsibility of maintaining student activities and Payment.

Name: Trainer

Description: A Trainer is a training teacher assigned by the Institute Manager to each class of students to follow the students closely. She/he has the responsibility of maintaining student activities.

2.3.1 STUDENT MANAGEMENT SUB SYSTEM

Register student, Record Attendance, Search Students, Enroll Students are the use cases of the Student Management Subsystem. Manager, Branch Manager, Teacher and Trainer are the actors of the use case (see figure 2.2). Manager manages the entire system. Main use cases; Register student in table 2.1 and Record attendance in table 2.2 are described below

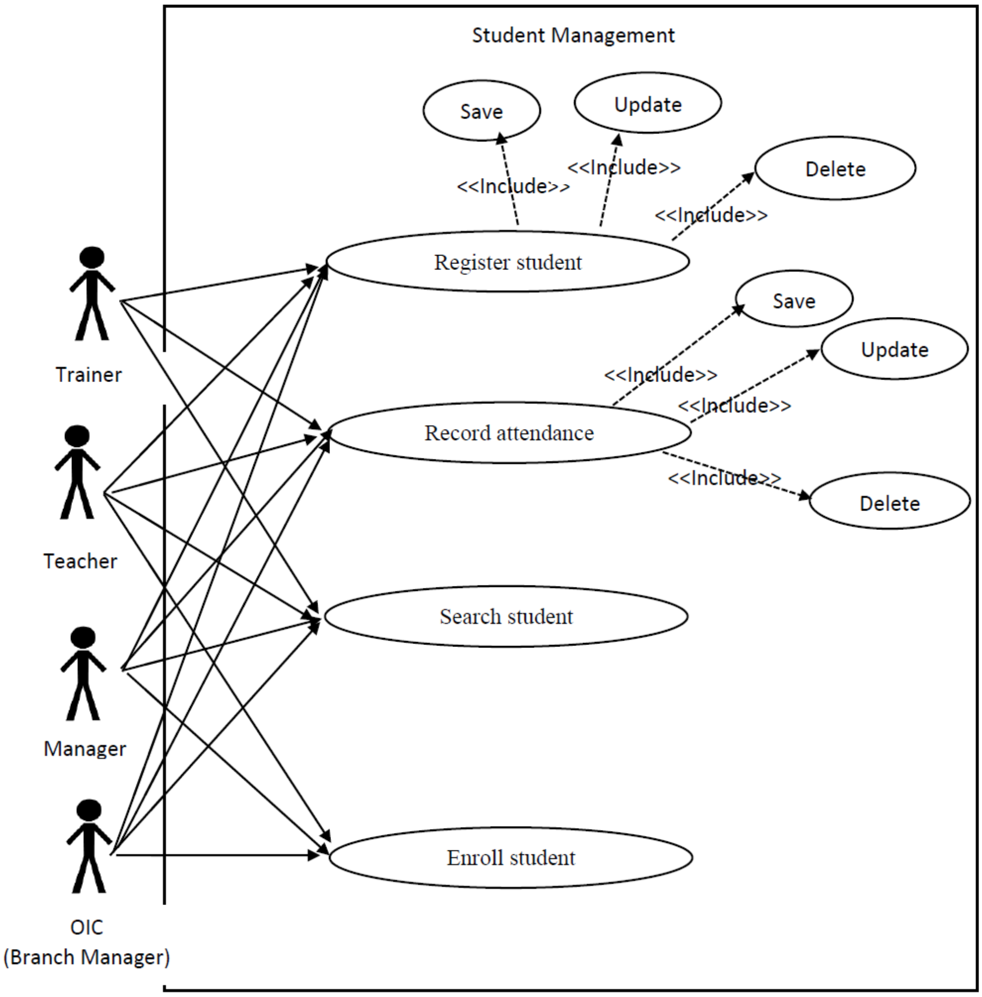


Figure 2.2: Use case for Student Management Subsystem

1. Description for the Register Student Use Case

All descriptions of the Register Student Use Case (like overview, Precondition etc.) are described below in Table 2.1.

Use case	Register Student
Actors	Manager/Branch Manager/Teacher/Trainer
Overview	
To register someone as a student of the Institute	
Preconditions	
1. Should not be already registered in the JVISMS.	
Flow of Events	
1. Student wants to be registered as a student of the Institute 2. Authorized person verifies that the student is eligible 3. Registration form will be given to the student 4. The student completes the registration form that contains student’s full name, address, parent name, Contact Number and addresses and other detail information. 5. Authorized person of the Institute checks whether the contents of the registration form is properly completed. 6. Authorized person checks if the member is already registered in JVISMS and if not fills and submits the form to the system. 7. Use case ends.	
Post Condition	
New Student Registered.	

Table 2.1: Description for the Register Student Use Case

2. Description for the Record Student Attendance Use Case

All descriptions of the Record Student Attendance Use Case (like overview, Precondition etc.) are described below in Table 2.2.

Use case	Record Student Attendance
Actors	Manager/Branch Manager/Teacher/Trainer
Overview	
To record attendance of students.	
Preconditions	
Flow of Events	
1. Teacher wants to record absentees from the class. 2. Having the attendance book the Teacher logs in to record. 3. Class teacher records absentees and submits. 4. System acknowledges. 5. Use case ends.	
Post Condition	
Absentees recorded.	

Table 2.2: Description for the Record Student Attendance Use Case

2.3.2 TEACHER MANAGEMENT SUBSYSTEM

Register Teacher, Record Attendance, Search Teacher, Create teacher level are the use cases of the teacher management subsystem. Main use cases; Register Teacher, Record Attendance are described below. Manager and Branch manager are the actors of the use case. Manager has relation to all use cases Branch manager has relationship with all use cases except the create teacher level use case (see figure 2.3)

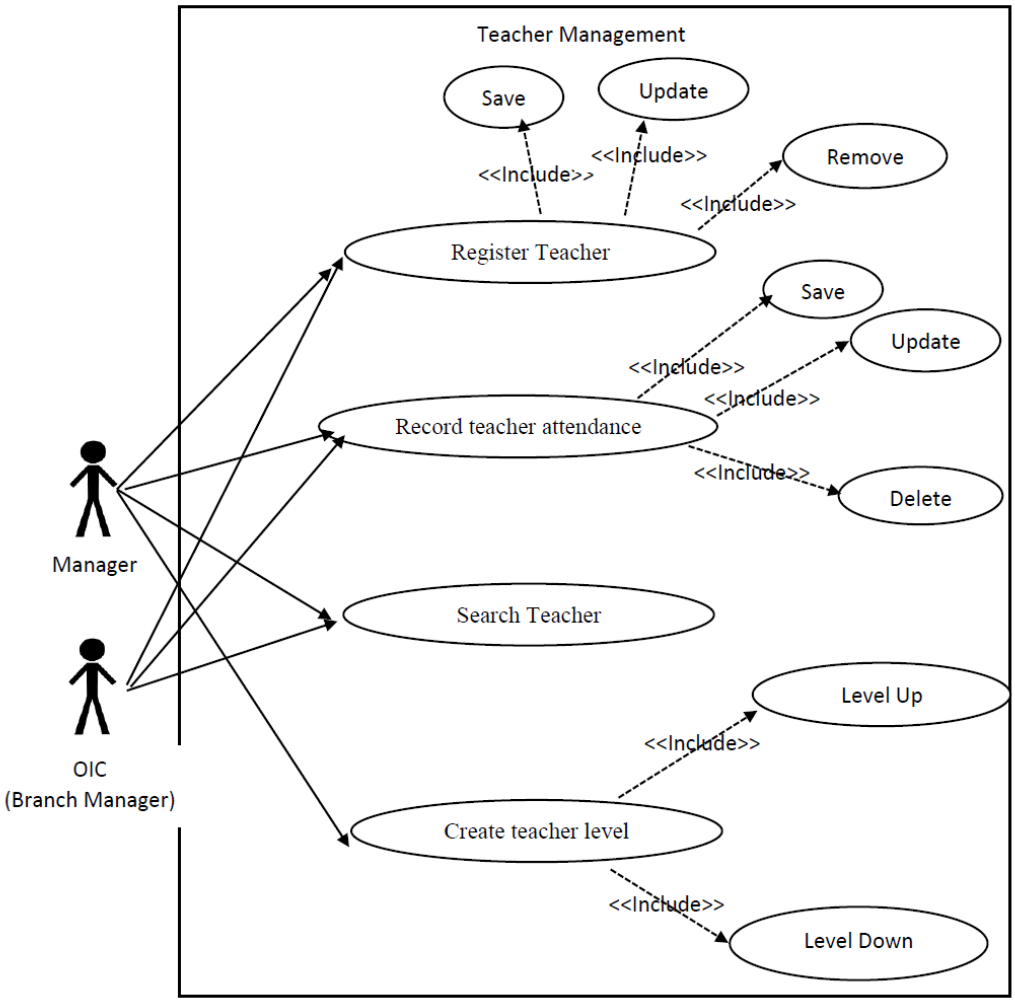


Figure 2.3: Use case for Teacher Management Subsystem

1. Description for the Register Teacher Use Case

All descriptions of the Register Teacher Use Case (like overview, Precondition etc.) are described below in Table 2.3.

Use case	Register Teacher
Actors	Manager/Branch Manager
Overview	
To register someone as a Teacher of the Institute	
Preconditions	
1. Should not be already registered in the JVISMS.	
Flow of Events	
1. Teacher wants to be registered as a Teacher of the Institute 2. Authorized person verifies that the Teacher is eligible 3. Registration form will be given to the Teacher 4. The Teacher completes the registration form that contains Teacher's full name, address, Contact Number, addresses and other detail information. 5. Authorized person of the Institute checks whether the contents of the registration form is properly completed. 6. Authorized person checks if the member is already registered in JVISMS and if not fills and submits the form to the system. 7. Use case ends.	
Post Condition	
New Teacher Registered.	

Table 2.3: Description for the Register Teacher Use Case

2. Description for the Record Teacher Attendance Use Case

All descriptions of the Teacher Attendance Use Case (like overview, Precondition etc.) are described below in Table 2.4.

Use case	Record Teacher Attendance
Actors	Manager/Branch Manager
Overview	
To record attendance of Teacher.	
Preconditions	
1. Manager must login to record attendance.	
Flow of Events	
1. Manager/Branch Manager want to record absentees of the teachers. 2. Teachers sign the attendance book at the office. 3. Having the attendance book the Manager/Branch Manager logs in the system. 4. Manager/Branch Manager records absentees and submit. 5. System acknowledges. 6. Use case ends	
Post Condition	
Absentees recorded.	

Table 2.4: Description for the Record Teacher Attendance Use Case

2.3.3 PAYMENT MANAGEMENT SUBSYSTEM

Manage Payment is the use case of the Payment Management Subsystem (see figure 2.4). And also it is the generalization of the use cases; Get Payment and Check Payment. Manager/Branch Manager/Teacher are the actors of the use case.

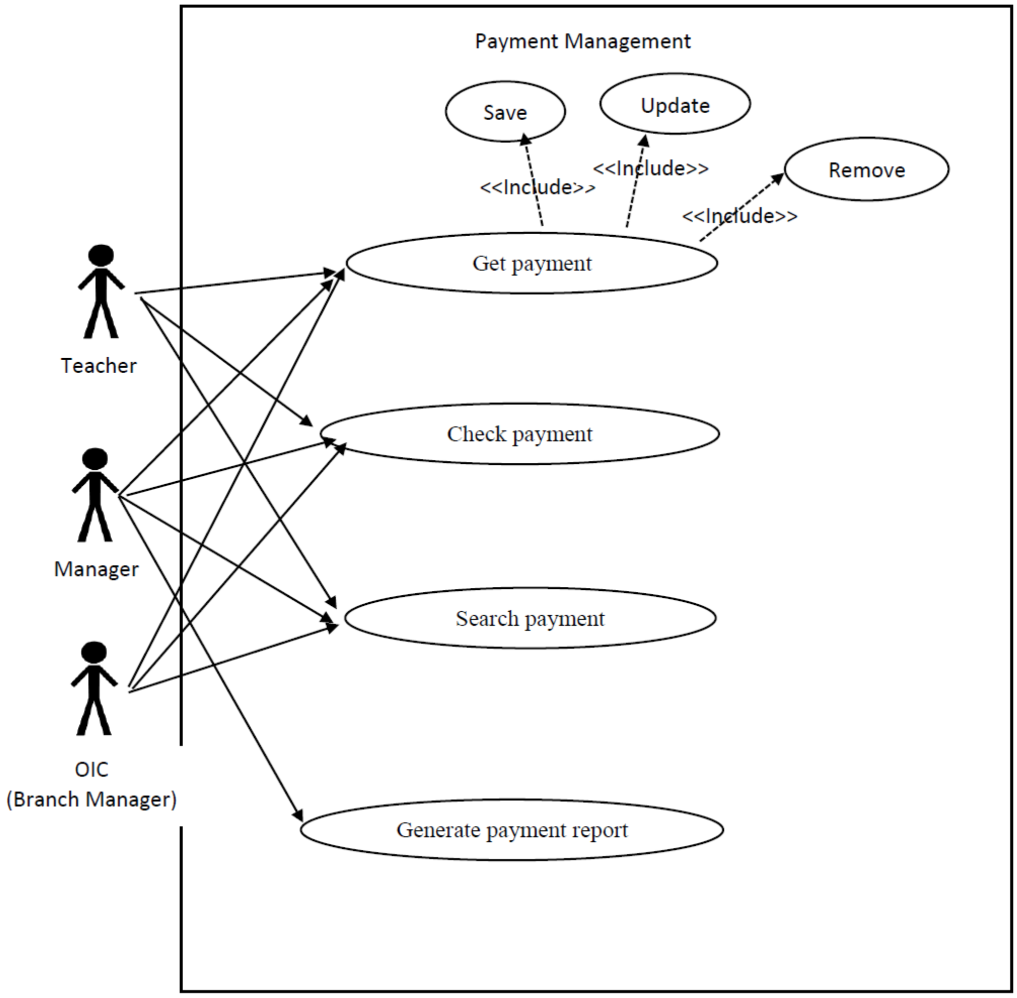


Figure 2.4: Use Case for Payment Management Subsystem

1. Description for the Get Payment Use Case

All descriptions of the Get Payment Use Case (like overview, Precondition etc.) are described below in Table 2.5.

Use case	Get Payment
Actors	Manager/Branch Manager/Teacher
Overview	
To Get Payment of students.	
Preconditions	
1. Authorized Person must login to get Payment.	
Flow of Events	
1. Manager/Branch Manager/Teacher want to get payment. 2. Student settle the payment. 3. Having the payment the Manager/Branch Manager/Teacher logs in the system. 4. Manager/Branch Manager/Teacher records Payments. 5. System acknowledges. 6. Use case ends	
Post Condition	
Settled Payment	

Table 2.5: Use Case for Payment Management Subsystem

2.3.4 SECURITY MANAGEMENT SUBSYSTEM

Security of the system is managed by the Manager. Hence use case of the Security Management Subsystem has only Manager as the actor. Manage users and Create Backup and recovery are the identified use cases of the subsystem (see figure 2.5)

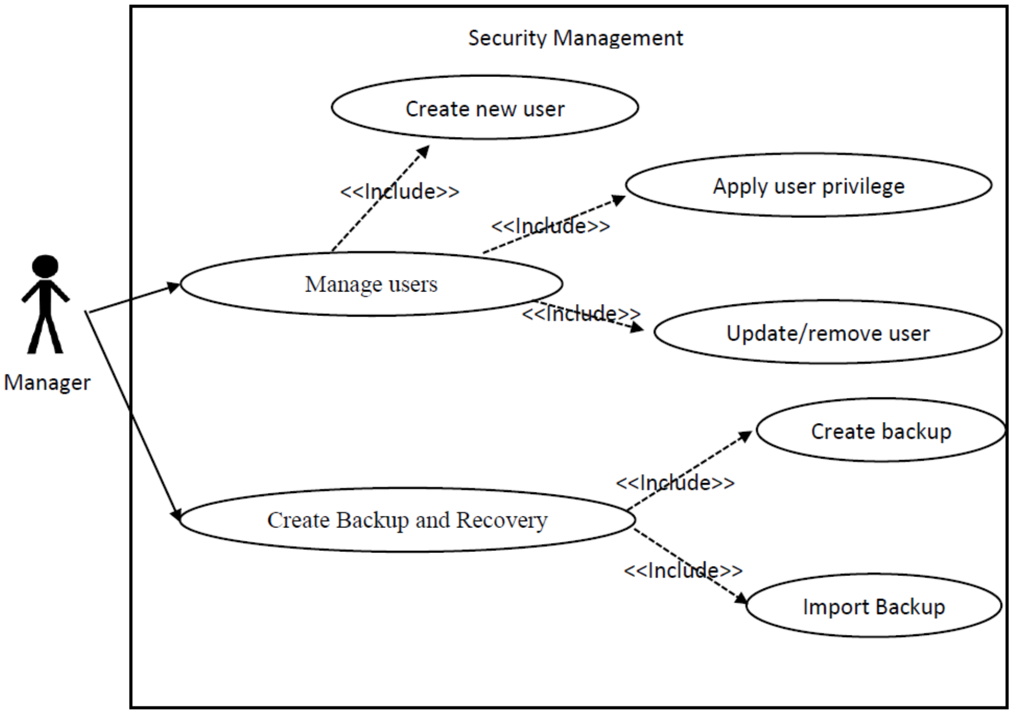


Figure 2.5: Use case for the Security Management Subsystem

2.3.5 REPORT MANAGEMENT SUBSYSTEM

All the reports are managed by the Manager and Branch Manager. Use cases of the report management subsystem are View reports, Print Reports and Save reports. View reports use case is described in order to clearly understand the use case (see figure 2.6).

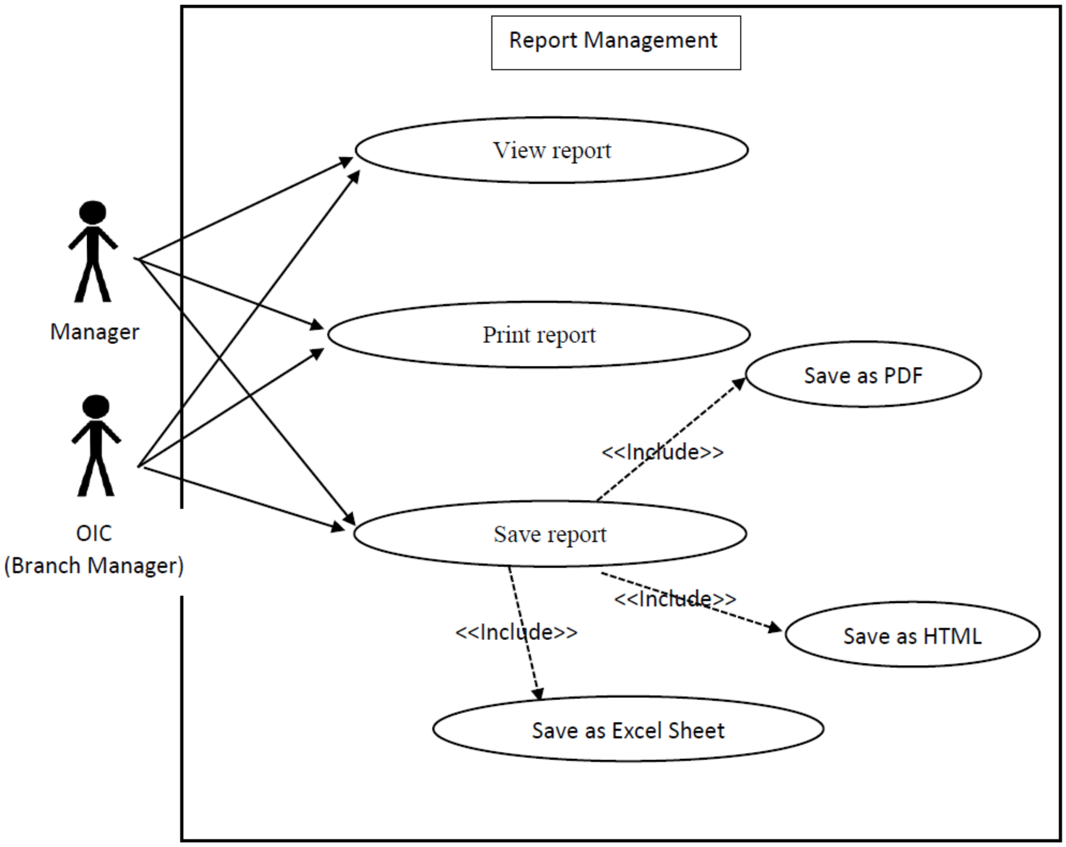


Figure 2.6: Use case for the Report Management Subsystem

1. Description for the View Reports Use Case

All descriptions of the View Reports Use Case (like overview, Precondition etc.) are described below in Table 2.6.

Use case	View Report
Actors	Manager/Branch Manager
Overview	
To Generate and View Report	
Preconditions	
1. Authorized Person must login to View Report.	
Flow of Events	
1. Authorized person wants to generate report. 2. Authorized Person logs in. 3. System displays a form for relevant criteria for the report. 4. Authorized person fills in the form and submits. 5. System displays the appropriate report. 6. Use case ends	
Post Condition	
Report Generated and Displays	

Table 2.6: Description for the View Reports Use Case

2.4 FUNCTIONAL REQUIREMENTS OF THE SYSTEM

- The system should facilitate to register a new teacher.
- The system should facilitate to register a new student.
- The system should facilitate to record attendance of students.
- The system should facilitate to record attendance of teachers.
- The system should facilitate to record payment of students.
- The system should facilitate to record payment of teachers.
- The system should keep all historical data of students.

- The system should display automatically updated grades of the students.
- System should have a calendar with daily events.
- The system should have a powerful search facility.
- The system should facilitate with super report generation.

2.5 NON-FUNCTIONAL REQUIREMENTS OF SYSTEMS

In addition to the obvious features and functions that will provide in the proposed system, there are other requirements that don't actually do anything, but they are very important for a successful system. These are called "non-functional requirements" or sometimes "Quality Attributes." For example, attributes such as performance, security, usability, compatibility aren't a "feature" of the system, but are a required characteristic.

Data will be stored in a database; hence Security requirements are very important factors in this system. User authentication & authorization will be done during login to ensure that the user is valid and that the user only has access to his or her permission data.

The system should have consistent interface formats and button sets for all forms in the application, will have a form based interface for all data entry and viewing formats, and will generate reports that are formatted in a table and that should look like the existing manual report formats for user friendliness.

Time consumption is the major problem associated with the current manual system. To overcome this situation the system should have fast response time (real time) with performance. Furthermore, the system should not be taking up too much space in memory.

The system should be easily maintainable by the developer or other authorized trained person and the system should be implemented in a way to utilize reusability and maintainability of both the design and code.

The system should have an inbuilt backup and restore facility to ensure data protection. Reliability of the system will be acquired through keeping backups of the system.

2.6 PROCESS MODEL

To develop high quality software it is needed to follow a software process model. There are many process models such as RUP (Rational Unified Process), prototyping model, waterfall Model and Rapid Application Development, in the software development industry. The RUP (Rational Unified Process) model was selected to develop the JVISMS over other process models.

"The Rational Unified Process [2] is a Software Engineering Process. It provides a disciplined approach to assigning tasks and responsibilities within a development organization. Its goal is to ensure the production of high-quality software that meets the needs of its end-users, within a predictable schedule and budget."

The figure 2.7 [3] shows how this separation is depicted in Rational Unified Process.

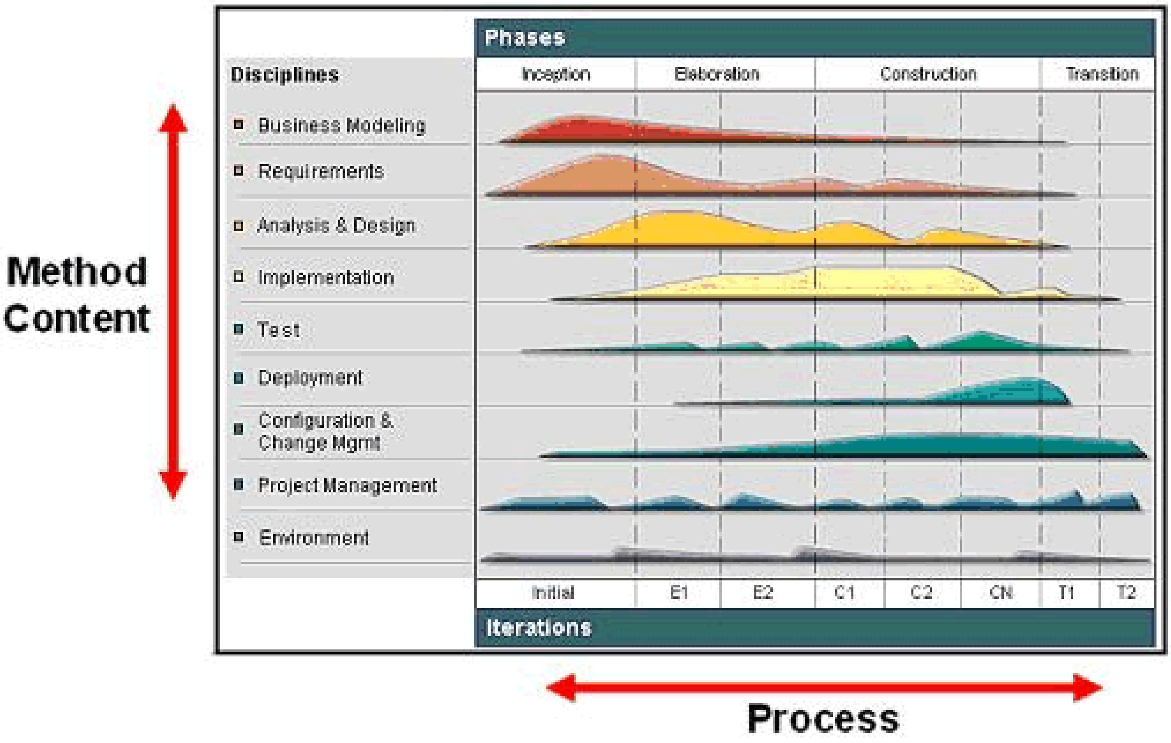


Figure 2.7: Rational Unified Process Model

2.7 LITERATURE REVIEW AND SIMILAR SYSTEMS

Some of the existing systems were followed and studied to develop JVISMS. Most of the available proprietary and open source systems consist with unnecessary functionalities and features that waste the time of the client and further it is difficult to study the system by the client.

With reference to some web based and standalone applications, most important functions and features that are most use full to the selected school were listed. For this purpose, selected web base applications like Fedena, Ecare and open source systems like openSIS, Web-School were followed.

fedena is an award winning web based school management system. It provides a free demo facility to try all features of Fedena. Provided demo was very useful to get an overall idea of the system. Important functionalities for JVISMS were identified by the trial system.

As its official site states [4] „Fedena is multipurpose school management software which is used by thousands of educational institutions worldwide for all administration, management and learning related activities. Use Fedena management information system to manage students, teachers, employees, courses and all the systems and processes related to running your institute efficiently.

Further knowledge on school management systems was gained through open source system called OpenSIS. OpenSIS community edition is freely available to download.

Reviews on these systems broadened the knowledge on school management systems and helped to get a clear idea on how to design the JVISMS.

CHAPTER 3 – DESIGN

“System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements” [5]. System design can be sub divided into Logical design and Physical Design.

Functional and non-functional requirements of the system and produced the analysis model was identified in the previous chapter. The following are discussed in this chapter: database design, process design and user interface design of the proposed system.

3.1 DATABASE DESIGN

A database design is process of transforming a higher level conceptual data into a logical data model that will be used to implement the physical database and the application. The database of the JVISMS is designed with consideration of a quality database design and will be implemented, maintained and modified in a cost effective way.

Although a use case diagram is not a database design tool, and cannot be directly contributed to a quality database, found that use case modeling helped to capture business requirements in order to design a good database in several ways. Firstly, the use case diagram is an initial tool that describes what a system does, and reflects the view of what users want. Therefore, use cases can be used to determine entities, objects, classes, and relationships for designing ERD.

3.1.1 DATABASE NORMALIZATION

- First Normal Form (1NF) – “Remove duplicate columns from the same table and generate separate tables for each group by giving a primary key for each row” [6].
- Second Normal Form (2NF) –“This requires the first normal form as the prerequisite and remove subsets of data which affects to multiple rows and include them in separate tables. Create relationships between newly created tables and their predecessors using foreign keys” [6].
- Third Normal Form (3NF) – “This requires the second normal form as a prerequisite and any non-key attribute that does not depend on the key has to be removed” [6] .

3.1.2 ENTITY RELATIONSHIP DIAGRAM

The figure 3.1 illustrates the Entity Relationship Diagram for the JVISMS. Relationships among the database Tables can be identified through ERD.

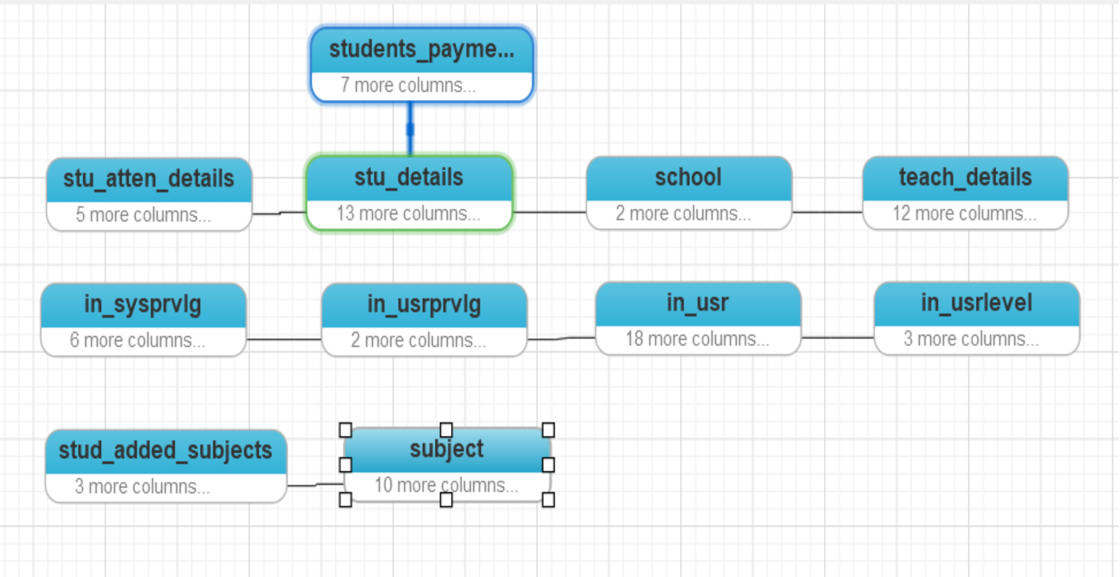


Figure 3.1: Entity Relationship Diagram of JVISMS

3.2 CLASS DIAGRAM

UML Class diagram for School Information Management System is shown below. Main Classes involved in the system are:

Classes: Student, Teacher, Parent, User, Attendance, Payment, Subject, Backup, Login.

All attributes and methods are not included in the class diagram (see figure 3.2). Main and most important attributes are shown instead.

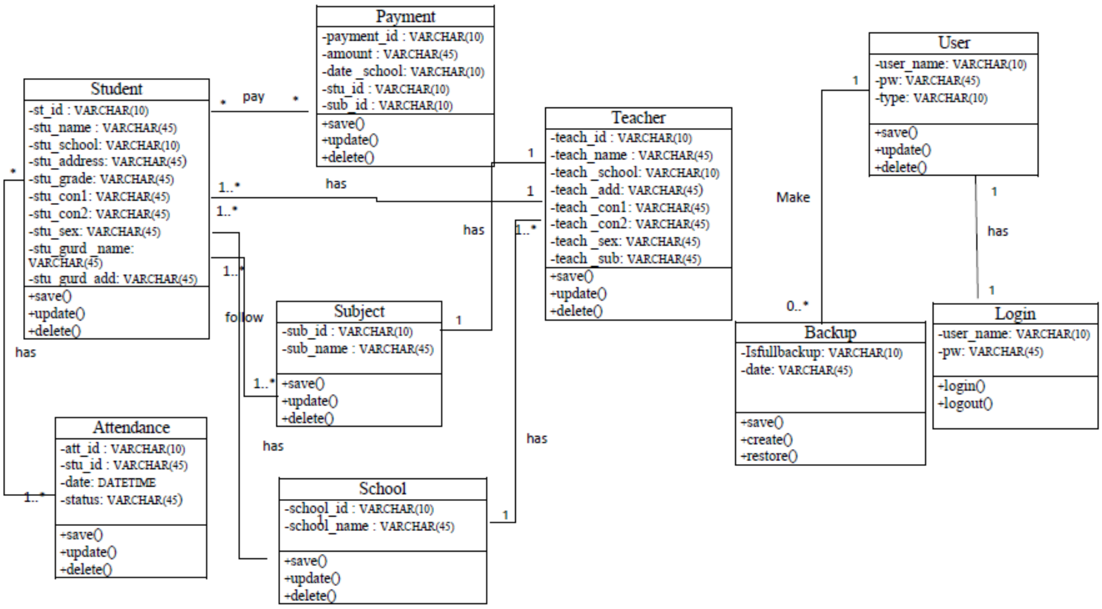


Figure 3.2: Class diagram of JVISMS

3.3 MODULAR DECOMPOSITION

Modular decompositions will help to reduce the complexity of the system. The modular decomposition of the JVISMS is as shown in Figure 3.3. These modules are further decomposed into other modules. The major modules identified are Student, Teacher, reports, Payment, Security & Backup modules.

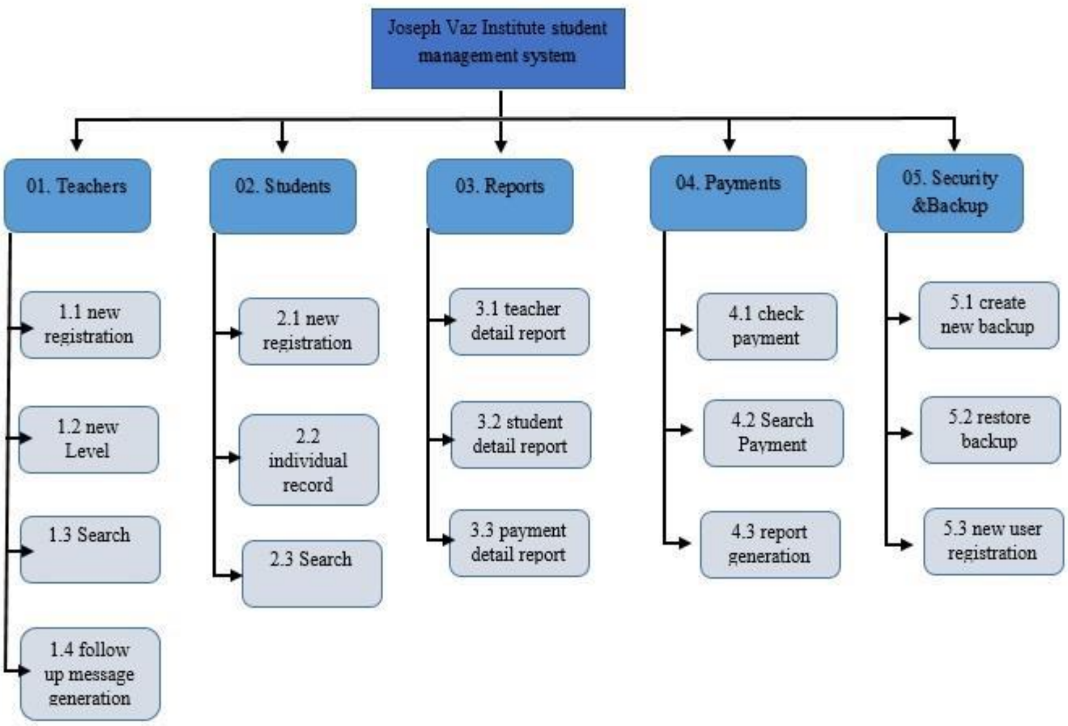


Figure 3.3: Modular Decomposition Diagram of the System

3.4 INTERFACE DESIGN

A critical aspect of the information system is the quality of the user interface. The design of the user interface defines how the user will interact with the system. Therefore user interface design requires a good understanding of user needs.

Considering the above facts this system was developed with simple user-friendly interfaces ensuring good interaction with the system.

To improve the usability of the system Shneiderman‟s [7] “Eight Golden Rules of Interface Design" was followed.

1. Strive for consistency.

2. Enable frequent users to use shortcuts.

3. Offer informative feedback.

4. Design dialog to yield closure.

5. Offer simple error handling.

6. Permit easy reversal of actions.

7. Support internal locus of control.

8. Reduce short-term memory load.

3.4.1 LOGIN WINDOW

The initial interface “User Login” shown at the startup of the program is shown in the Figure 3.4. User needs to enter their valid user credential to log into the JVISMS. In cause of invalid credentials, user will not be able to login to the system, error message will be displayed.

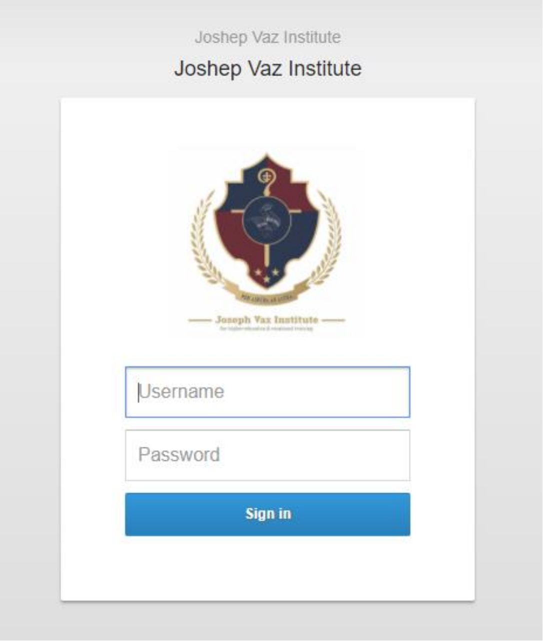
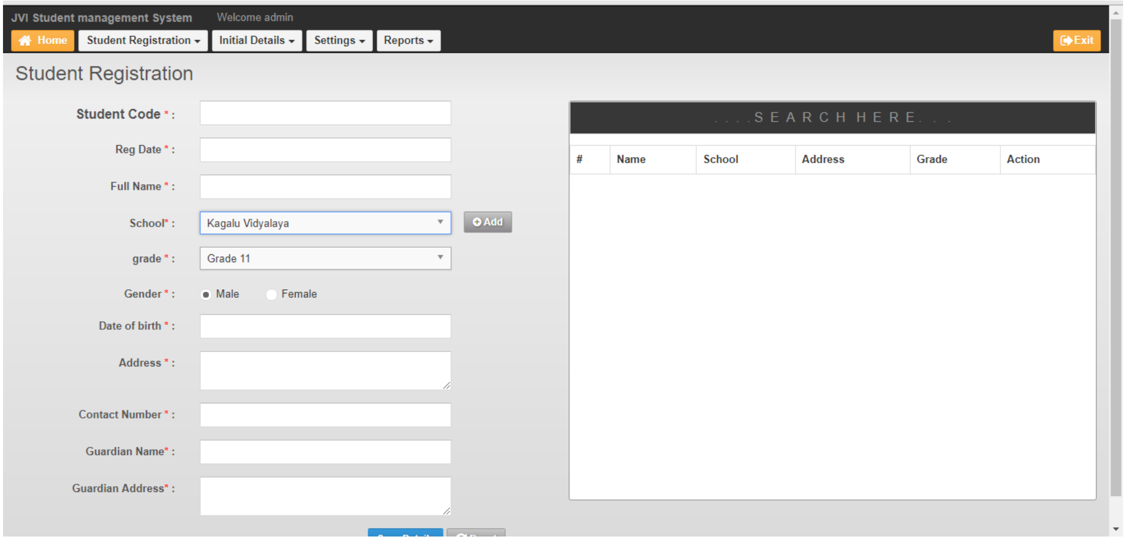


Figure 3.4: User Login Interface

3.4.2 STUDENT REGISTRATION WINDOW

Figure 3.5 shows the student registration window of JVISMS. User needs to enter valid student details in to the form in order to register a student.



Student Registration

Student Code * :

Reg Date * :

Full Name * :

School* : Kagalu Vidyalaya

grade * : Grade 11

Gender * : ● Male ○ Female

Date of birth * :

Address * :

Contact Number * :

Guardian Name* :

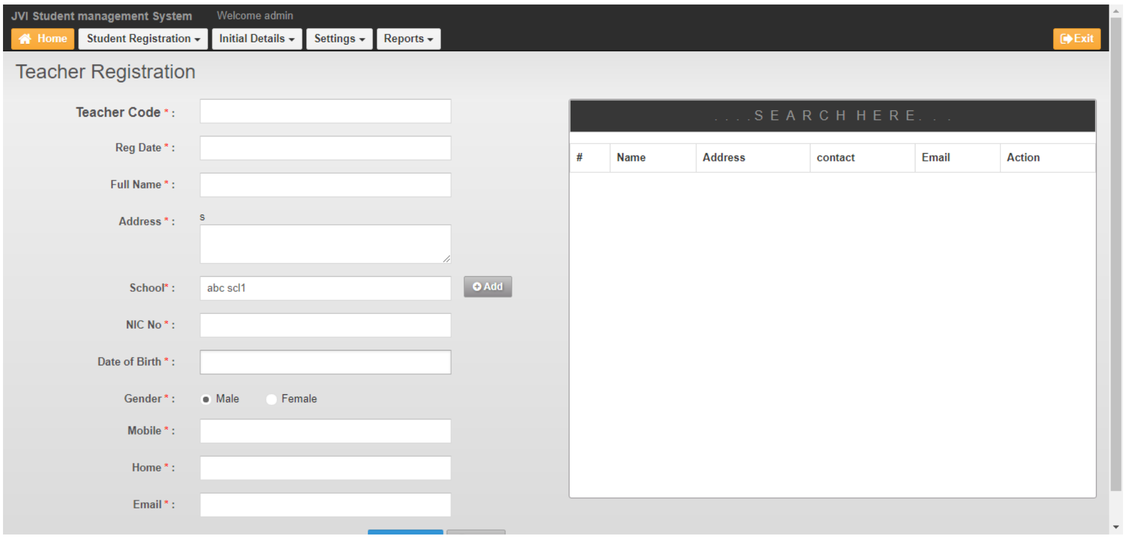
Guardian Address* :

#	Name	School	Address	Grade	Action

Figure 3.5: Student Register Interface

3.4.3 TEACHER REGISTRATION WINDOW

Figure 3.6 shows teacher registration window of the JVISMS. Form displayed in the window is grouped to make sure that the user is motivated to enter all and valid details of a teacher.



Teacher Registration

Teacher Code * :	
Reg Date * :	
Full Name * :	
Address * :	s
School* :	abc scl1
NIC No * :	
Date of Birth * :	
Gender * :	● Male ○ Female
Mobile * :	
Home * :	
Email * :	

#	Name	Address	contact	Email	Action

Figure 3.6: Teacher Register Interface

3.5 REASONS TO CHOOSE WEB APPLICATION

Although a desktop (stand-alone) student management system is an appropriate alternate solution for JVISMS, there are many disadvantages compared to web application. Following describes some of the reasons to select a Desktop Application.

- The Client request was to develop a web Application.
- Unlike traditional applications, web systems are accessible anytime, anywhere and via any PC with an Internet connection. This puts the user firmly in charge of where and when they access the application.[8]
- In addition to being customizable for user groups, content can also be customized for use on any device connected to the internet. This includes the likes of PDAs, mobile phones and tablets.[8]

CHAPTER 4 - IMPLEMENTATION

In this chapter, the implementation of the system is described. Chapter explains all major code and module structures. Also, the implementation environment (hardware and software), existing codes that was reused and tools used to develop the system. In this phase the major process is to apply technological aspects into practice to satisfy the gathered requirements and design principles as described in previous phase.

4.1 SOFTWARE REQUIREMENTS

4.1.1 LANGUAGES AND TECHNIQUES USED, HAVE BEEN DESCRIBED BELOW

- PHP(Hypertext Preprocessor)– Server side scripting language widely used for web development mainly together with HTML (Hyper Text Markup Language)
- CSS –Cascading Style Sheets have been used for styling the document written in HTML
- Java script –Used for client side validation
- Ajax(Asynchronous JavaScript Technology and XML)
- Navicat Premium-Used to implement the database
- Xamp Server-This server was used to develop the system
- Operating System-The development of the system was done using Windows 10 professional operating system
- NetBeans IDE 8.0.2-Used as a development tool for coding
- Photoshop-Used for the purpose of making enhancements to the images

4.2 HARDWARE REQUIREMENTS

- Processor - Intel® Core™ i5 @1.60 GHZ 2.30GHZ
- RAM - 4GB
- 500 GB Hard Disk

4.3 EXTERNAL LIBRARIES

Chosen (v1.8.2)

Chosen automatically sets the default field text ("Choose a country...") by reading the select element's data-placeholder value. If no data-placeholder value is present, it will default to "Select an Option" or "Select Some Options" depending on whether the select is single or multiple [9].

Datepicker

The datepicker is tied to a standard form input field. Focus on the input (click, or use the tab key) to open an interactive calendar in a small overlay. Choose a date, click elsewhere on the page (blur the input), or hit the Esc key to close. [10]

FPDF

FPDF has been used to produce documents in printable pdf format

4.4 CODES WHICH HAVE USED FREQUENTLY

4.4.1 DATABASE CONNECTIVITY

This is the code which has been used to connect with the database named as “jvisms”. This is a very important task since there’s no any other option to execute queries unless the system is connected to a database. Lots of functionalities have been assembled to the dbc.php file and if there is a need of any change have to modify only this php file and it is a very time preserving method.

```
<?php

$baseURL = '/jvi_sms/';
$db_NAME = 'jvisms';
$db_HOST = 'localhost';
$db_USER = 'root';
$db_PASS = '';

class MainConfig {

    public static function connectDB() {
        global $db_HOST, $db_USER, $db_PASS, $db_NAME;
        $link   =   mysql_connect($db_HOST,   $db_USER,   $db_PASS)   or
die("Couldn't make connection.");
        mysql_set_charset('utf8', $link);
        date_default_timezone_set('Asia/Colombo');
        $db = mysql_select_db($db_NAME, $link) or die("Couldn't select
database");
        }
      }
?>
```

4.4.2 USER MODULE

User Login

The application was developed only for the particular set of users. Valid users can provide an access point to the all valid database information.

Some controls over the users were added by using Sessions. After entering the user name and password by a user, login module will check the user in the database and if he available module will create a session for the user.

Following Code segment shows customer login

```php
<?php

require_once '../config/dbc.php'; //include database connection
require_once '../class/systemSetting.php';
$system = new setting(); // create object for system class(which include
mySql functions)

if (array_key_exists("logSystem", $_POST)) {
    if  (isset($_POST['userName'])  &&  !empty($_POST['userName'])  &&
isset($_POST['password']) && !empty($_POST['password'])) {
        $user = ($_POST['userName']);
        $pass = ($_POST['password']);
        $userQuery = "SELECT
in_usr.usrID,
in_usr.usrName,
in_usr.usrPwd,
in_usr.usrStatus,
in_usr.usrLevel,
in_usr.userBranchID
FROM
in_usr
WHERE
(in_usr.usrStatus = '1') AND
in_usr.usrName = '{$user}' LIMIT 1";
        $userAvailability = $system->getCountByQuery($userQuery);
        if ($userAvailability > 0) {
            $userDetails = $system->prepareSelectQuery($userQuery);
     $encriptedPass = sha1('ch' . $pass . 'BIT');
            foreach ($userDetails as $ud) {
                if ($ud['usrPwd'] == $encriptedPass) {
                    //Set Cookie if select remember btn
                    session_start();
                    $_SESSION['user_id'] = $ud['usrID'];
                    $_SESSION['user_name'] = $ud['usrName'];
                    $_SESSION['user_level'] = $ud['usrLevel'];
                    $_SESSION['usrStatus'] = $ud['usrStatus'];
                    $_SESSION['branch'] = $ud['userBranchID'];
                    $_SESSION['HTTP_USER_AGENT']                    =
md5($_SERVER['HTTP_USER_AGENT']);
                    echo json_encode(array(array("msgType" => 0, "msg"
=> "Successfully Logged to the System")));
                } else {
                    echo json_encode(array(array("msgType" => 1, "msg"
=> "Password was incorrect.Please Check your Password")));
```

```
                }
            }
        } else {
            echo json_encode(array(array("msgType" => 2, "msg" => "User
was not available in database,plase check your username")));
        }
    } else {
        echo json_encode(array(array("msgType" => 3, "msg" => "Please
enter username or password")));
    }
}
?>
```

User Creation

Instead of access to the system by giving user name and password any one can create a new user. It is requires to submit some little information. After filling require information and click save button. We can create new users via the system. Some code snippet of that part as follow,

```
<?php
$system = new setting();//create object for system class(which include
mySql functions)
$dbClass = new database();//create object for database class
if ($_POST['action'] == 'addNewAdminUser') {
        $time = date("h:i:s a");
        $encriptedPass = sha1('ch' . $pass . 'BIT');
        $system->prepareCommandQueryForAlertify("INSERT INTO `in_usr`
(`usrName`, `usrFName`, `usrLName`, `usrLevel`,
            `usrPwd`, `usrRegDate`, `usrStatus`, `usrAddress`,
`usrEmail`, `lstLgDate`, `lstLgTime`, `usrEmpNo`, `usrNIC`,

            `usrMobileNo`, `usrWorkTelNo`, `usrHomeTelNo`,
`userBranchID`) VALUES ('{$_POST['username']}', '{$_POST['fName']}',
            '{$_POST['lName']}', '{$_POST['selUserLevel']}',
'{$encriptedPass}', '{$_POST['date']}', '{$_POST['userStatus']}',
            '{$_POST['address']}', '{$_POST['eMail']}',
'{$_POST['date']}', '{$time}', '{$_POST['empNo']}', '{$_POST['nic']}',
            '{$_POST['mobile']}', '{$_POST['work']}',
'{$_POST['home']}', '{$_POST['branchID']}');",
            "Successfully Saved", "Sorry ..! Counld Not Be Save");
         }
    }
?>
```

4.4.3 CREATE, READ, UPDATE, DELETE (CRUD)

These are four basic functions that play a vital role in each and every script code when it comes to implementing modules. Most of the coding sections contain at least one CRUD function.

```
if ($_POST['action'] == 'get_student_details') {
        $system->prepareSelectQueryForJSON("SELECT
stu_details.stu_name,
stu_details.stu_grade,
school.school_name
FROM
stu_details
INNER JOIN school ON stu_details.stu_school = school.school_id
WHERE
stu_details.stu_id = '{$_POST['serch_data']}'");
    }
}
```

4.5 IMPLEMENTATION ENVIRONMENT

This system could be installed in a web server with the facility of Linux Operating System and this should be accessible to the users through internet with the help of a suitable web browser. Since the system needs to be accessed by the external parties having only an intranet will not sufficient enough for system implementation environment. System has been implemented accordance with the Client Server architecture. Client Server architecture facilitates a communication where server sends the request and client responds while emphasizing the flexibility of it. For testing purposes the system has been implemented on local host.

4.6 SECURITY

Security is the most powerful criteria to be measured whether the user can cope up with the system without any doubt since usually there are some situations which store external parties' sensitive data and at the same time these kinds of situations allow space for other parties to peep into other's information and grab them unethically. In-order to reduce these kinds of threats security mechanisms play a vital role. Especially security is one of the most important factors to be considered and managing it in an efficient manner will ensure the quality of the system as well. Academic Administration Management System Page 39

There should be a high level security to prevent unauthorized access, unethical modifications and illegal usages. Only the authorized users should have the access to use their profiles since separate functions have been allocated according to their user type. Granting the access will help to restrict unethical behaviors as well as passwords have encrypted using MD5 algorithm as another security mechanism as shown in the below.

```
<?php
require_once '../config/dbc.php';
require_once '../class/systemSetting.php';
$system = new setting();
if (array_key_exists("logSystem", $_POST)) {
    if (isset($_POST['userName']) && !empty($_POST['userName']) &&
isset($_POST['password']) && !empty($_POST['password'])) {
        $user = ($_POST['userName']);
        $pass = ($_POST['password']);
        $userQuery = "SELECT
in_usr.usrID,
in_usr.usrName,
in_usr.usrPwd,
in_usr.usrStatus,
in_usr.usrLevel,
in_usr.userBranchID
FROM
in_usr
WHERE
(in_usr.usrStatus = '1') AND
in_usr.usrName = '{$user}' LIMIT 1";
        $userAvailability = $system->getCountByQuery($userQuery);
        if ($userAvailability > 0) {
            $userDetails = $system->prepareSelectQuery($userQuery);
            $encriptedPass = sha1('ch' . $pass . 'BIT');
            foreach ($userDetails as $ud) {
                if ($ud['usrPwd'] == $encriptedPass) {
                    //Set Cookie if select remember btn
                    session_start();
                    $_SESSION['user_id'] = $ud['usrID'];
                    $_SESSION['user_name'] = $ud['usrName'];
                    $_SESSION['user_level'] = $ud['usrLevel'];
                    $_SESSION['usrStatus'] = $ud['usrStatus'];
                    $_SESSION['branch'] = $ud['userBranchID'];
                    $_SESSION['HTTP_USER_AGENT'] =
md5($_SERVER['HTTP_USER_AGENT']);

                    echo json_encode(array(array("msgType" => 0, "msg"
=> "Successfully Logged to the System")));
                } else {
                    echo json_encode(array(array("msgType" => 1, "msg"
=> "Password was incorrect.Please Check your Password")));
                }
            }
        } else {
            echo json_encode(array(array("msgType" => 2, "msg" => "User
was not available in database,plase check your username")));
        }
    } else {
        echo json_encode(array(array("msgType" => 3, "msg" => "Please
enter username or password")));
    }
}
```

CHAPTER 05 – EVALUATION

5.1 INTRODUCTION

Evaluation is a review of the whole project, looking at its overall value and effect. It usually takes place when the project is complete. You can use the information you've collected during monitoring. Doing an evaluation will help you see how and where your project has been successful. It will let you see how well funds have been used, and what the benefits of your project have been to those involved. You can use this information to improve your future projects [11] .

5.2 TESTING PROCESS

5.2.1 UNIT TESTING

This testing procedure is used to test individual units or the functions to check whether those small individual parts execute without any errors and this is a very simple testing mechanism that can be used to move with further steps ahead. These units have no any connection or dependencies among other modules.

5.2.2 INTEGRATION TESTING

Integration testing covers the wide portion by concentrating on different pieces of the system work together. This is used to ensure different software module implementations are suitable to cover up the functional requirements of the system. Though the testing of small portions have been passed from unit testing, when an unit integrates with another it behaves differently and there’s a high probability for errors to take place especially with the interfaces.

5.2.3 SYSTEM TESTING

After the unit testing and integration testing the whole system should be tested to ensure that it satisfies client requirements. This consists of testing the backend database of a website parallel with the front end User Interface.

5.2.4 ACCEPTANCE TESTING WITH USER EVALUATION

Acceptance testing is used to identify whether the user has fully satisfied with the system and it is a valuable measurement to check the demand of the system.

User feedback has been gathered by conducting a survey and different kind of system users have been questioned to cover up the final outcome of the system.

Sample question form is given below in the Table 5.1.

No	Question	Excellent	Medium	Low
01	I like the interface of the system			
02	Colour theme is suitable			
03	Easy to use the system			
04	System satisfies the requirement			
05	I think this system is useful			
06	This system saves time			
07	I like the system			

Please put a tick under any of the option as you prefer

Table 5.1-User evaluation form

User evaluation data has been summarized in a graph as shown in figure 5.1 and several changes have been done to the system depends on the feedback

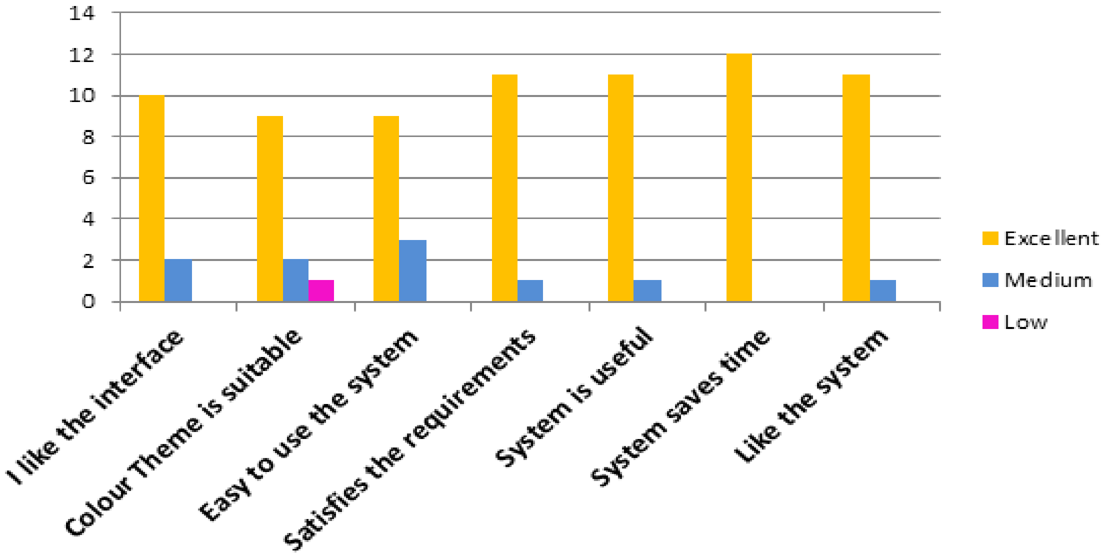


Figure 5.1-User evaluation summery

5.3 RELIABLE METHODS USED TO MINIMIZE ERRORS

Most of the manual entry methods have been reduced in order to overcome the problems that can arise due to manual processes and also to make feel the user less fatigued while using the system. To achieve this goal most of the functionalities have been granted according to user types and system will automatically display most of the relevant details without any user involvement. And also to maintain user friendliness, the use of alerts has been minimized as much as possible to deteriorate the rate at which end users get annoyed.

5.4 TEST CASES

5.4.1 USER LOGIN

Test cases for User login are shown in table 5.2.

ID	Test Case	Expected Output	Status
01	User Enters valid username with the password	Successfully Logged In to the system message	Pass
02	Empty Username and Password Fields	Please enter username or password message	Pass
03	Invalid Username with a correct password	Please enter username or password message	Pass
04	Valid username with an incorrect Password	Password was incorrect. please check your password message	Pass

Table 5.2-User login test plan

5.4.2 STUDENT REGISTRATION

Test cases for Student Registration are shown in table 5.3.

ID	Test Case	Expected Output	Status
01	User Enters Numerical Value for the Name In Full Field	Alert with an error message	Pass
02	Empty Name In Full Field	Alert with an error message	Pass
03	Empty Mobile Number	Alert with an error message	Pass
04	Empty Date Of Birth Field	Alert with an error message	Pass

Table 5.3-Student Registration test plan

5.4.3 EDIT/DELETE USERS

Test cases for the Edit/Delete user’s functionality are shown in table 5.4.

ID	Test Case	Expected Output	Status
01	Edit user record	Alert with a confirmation message	Pass
02	Delete user record	Alert with a confirmation message	Pass

Table 5.4-Edit/Delete test plan

5.4.4 ATTENDANCE MODULE

Test cases for the Attendance Module are shown in table 5.5.

ID	Test Case	Expected Output	Status
01	Click on Mark Attendance button to mark the attendance twice for the same class.	Alert saying You have already marked the attendance.	Pass
02	Click on Mark Attendance button to after start class time	Alert saying student is late	Pass

Table 5.5-Attendance module test plan

5.4.5 PAYMENT MODULE

Test cases for the Payment Module are shown in table 5.6.

ID	Test Case	Expected Output	Status
01	Student have any due payment	Alert saying student have <<how much>> of due	Pass

Table 5.6-Payment module test plan

5.4.6 COMMON FUNCTION

Test cases for the Common Function are shown in table 5.7.

ID	Test Case	Process	Expected Output	Status
01	Logout from the system	Click on exit Button	Alert saying do you want to log out	Pass
02	Go to home screen	Click home button	Redirect to the home page	Pass

Table 5.7-Common Function

5.4.7 ENROLL STUDENT

Test cases for the Enroll Student are shown in table 5.8.

ID	Test Case	Expected Output	Status
01	Enters the same subject twice for the same student for the same current grade	Error message saying this subject has been already selected	Pass
	Subjects added successfully	Message saying subjects have been added successfully	Pass

Table 5.8-Enroll Student

CHAPTER 06 – CONCLUSION

6.1 OVERVIEW

Joseph Vaz Institute which is one of the popular institute in North western province is looking forward to maintain their day to activities efficiently and effectively. The developed automated Student Management System facilitated with various activities taking place at the Institute such as student registration, teacher registration, attendance marking and Payment handling.

The implementation of Student Management System will have a huge impact on all the stake holders where it is in a guaranteed status to provide some essential required functionality to the users according to different kind of user's perspectives. Though there are so many student management systems available this has been implemented to fit with the client's unique requirements since the real situation of the institute cannot be caught up by any other systems which have implemented to cover up most of the common problems.

6.2 FUTURE ENHANCEMENTS

Staff attendance was not handled in the system since it should be integrated with a finger print scanner attendance machine which will be implemented while the system is being revised.

6.3 LESSONS LEARNT

Applying the lessons learnt to practical scenarios

This is one of the greatest experiences which have ever gained through BIT Degree Program and has persuaded my mind to be with the newest technologies and has provided an opportunity to apply the knowledge absorbed to the practical scenarios without sticking only with the theoretical side.

Work According to a timely manner

Main advantage was to know how to handle a task professionally in a timely manner. Practicing to work according to a schedule will be a value added feature to face with confidence for any future situation.

Completeness

From the Requirements Analysis phase to Evaluation phase had to maintain a proper progress in-order to get a successful outcome and all the tasks should be completed perfectly to make the final product much more effective.

Report Writing

Report writing was another privilege which was given through this project work and has improved my writing skill as well as could catch up the techniques of how to produce it in a well-structured format.

Work within a friendly environment

When it comes to requirements analysis phase had to cooperate with lots of strangers' in-order to have a better idea about what they need. Through formal procedures like interviews maintaining a friendly environment with them was a key point to grab their free thoughts and it was very helpful.

Knowledge improvement

Searched for lots of theories related to PHP and some other related languages to build up the system and it enhances the ability of self-studying.

REFERENCES

[1]Ibm.com. (2017). *IBM Knowledge Center*. [online] Available at: https://www.ibm.com/support/knowledgecenter/SS8PJ7_9.6.1/com.ibm.xtools.prodover.doc/topics/canalysismodel.html [Accessed 10 Jun. 2017].

[2]Anon, (2017). [online] Available at: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf [Accessed 20 Jun. 2017].

[3] Anon, (2017). [online] Available at: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf [Accessed 1 Jul. 2017].

[4] Fedena.com. (2017). *Fedena - School Management Software*. [online] Available at: https://www.fedena.com/ [Accessed 15 Jul. 2017].

[5] En.wikipedia.org. (2017). *Systems design*. [online] Available at: https://en.wikipedia.org/wiki/Systems_design [Accessed 15 Aug. 2017].

[6] Access-programmers.com. (2017). *Normalizing the Table Design*. [online] Available at: https://access-programmers.com/normalizing-the-table-design [Accessed 15 Aug. 2017].

[7] Faculty.washington.edu. (2017). Shneiderman's Eight Golden Rules of Interface Design.[online]Availableat:http://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html [Accessed 16 Aug. 2017].

[8] Magic Web Solutions. (2017). *The benefits of web-based applications - Magic Web Solutions*. [online] Available at: http://www.magicwebsolutions.co.uk/blog/the-benefits-of-web-based-applications.htm [Accessed 16 Aug. 2017].

[9]"Chosen: A jQuery Plugin by Harvest to Tame Unwieldy Select Boxes", Harvesthq.github.io, 2017. [Online]. Available: https://harvesthq.github.io/chosen/. [Accessed: 05- Oct- 2017].

[10]j. jquery.org, "Datepicker | jQuery UI", Jqueryui.com, 2017. [Online]. Available: https://jqueryui.com/datepicker/. [Accessed: 05- Nov- 2017].

[11]E. Trust, "Energy Saving Trust", Tools.energysavingtrust.org.uk, 2017. [Online]. Available: http://tools.energysavingtrust.org.uk/Communities/Support-and-guidance/How-to-guides/How-to-monitor-and-evaluate-your-community-project. [Accessed: 08- Oct- 2017].

APPENDIX A –SYSTEM DOCUMENTATION

This section guides to have an overall specification about all the hardware and software requirements as well as the installation procedures of Student management System for Joseph Vaz Institute.

Hardware Requirements

Personal Computer	Intel® Pentium® Dual Core or above
Processor	1.6 GHz processor
RAM	2 GB RAM
Hard Disk	160 GB With 40GB Space

Table A.1: Hardware Requirement

Software Requirements

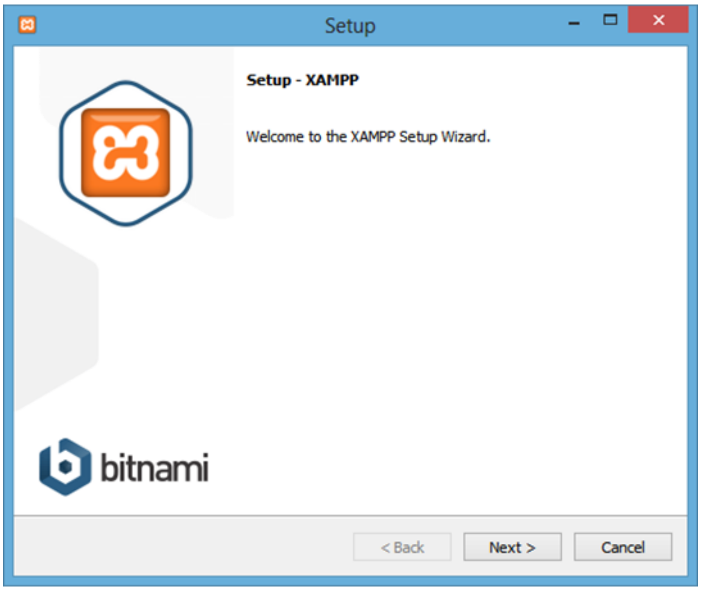
Operating System	Windows 7 or above
Browser	Java Script enable Browser

Table A.2: Software Requirement

Software Installation

Step1: Install XAMPP server stack using the interface shown in figure A.1. You can download XAMPP from

https://www. apachefriends.org/download.html



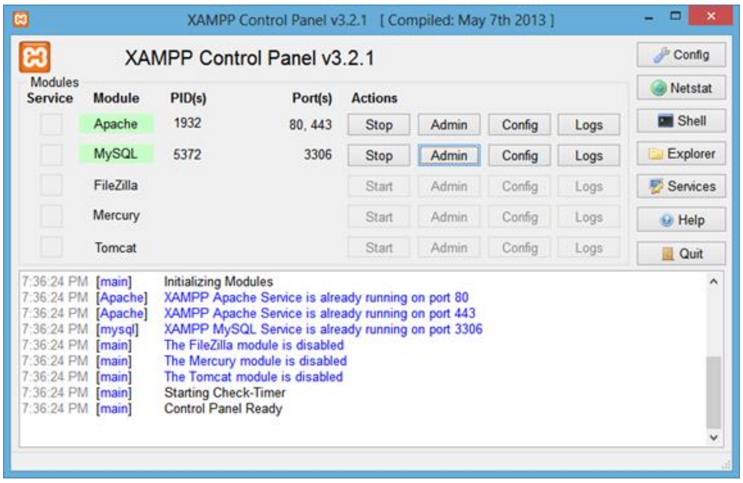
Setup

Setup - XAMPP

Welcome to the XAMPP Setup Wizard.

Figure A.1 XAMPP Setup window

Step 2: Run the XAMPP control panel and make sure Apache and MySQL services are running using the interface shown in figure A.2



XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]

XAMPP Control Panel v3.2.1

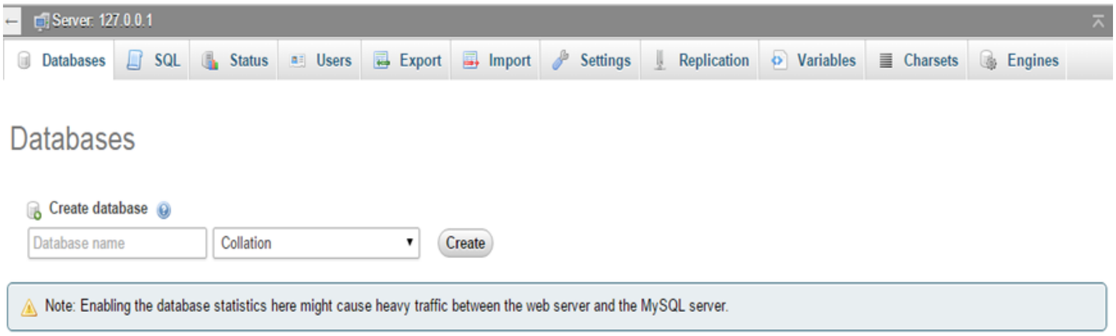
Modules

Service	Module	PID(s)	Port(s)	Actions			
□	Apache	1932	80, 443	Stop	Admin	Config	Logs
□	MySQL	5372	3306	Stop	Admin	Config	Logs
□	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
□	Tomcat			Start	Admin	Config	Logs

```
7:36:24 PM  [main]     Initializing Modules
7:36:24 PM  [Apache]   XAMPP Apache Service is already running on port 80
7:36:24 PM  [Apache]   XAMPP Apache Service is already running on port 443
7:36:24 PM  [mysql]    XAMPP MySQL Service is already running on port 3306
7:36:24 PM  [main]     The FileZilla module is disabled
7:36:24 PM  [main]     The Mercury module is disabled
7:36:24 PM  [main]     The Tomcat module is disabled
7:36:24 PM  [main]     Starting Check-Timer
7:36:24 PM  [main]     Control Panel Ready
```

Figure A.2 XAMPP control panel

Step 3: Click the ‘Admin’ button of MySQL. It will open your web browser and load phpMyAdmin application using the interface shown in figure A.3



Databases

Create database

Database name

Collation

Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.

Figure A.3 Create Database in phpMyAdmin

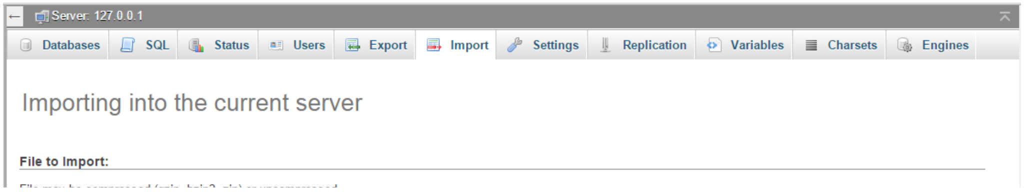
Step 4: Go to Databases tab and type jvi_sms as the database name and click on create button. Then go to import tab and browse for jvi_sms sql file located in the CD and click Go button using the interface shown in figure A.3

Configure the system

Copy the system files on CD and paste to “XAMPP\htdocs” folder on computer

How to run the system

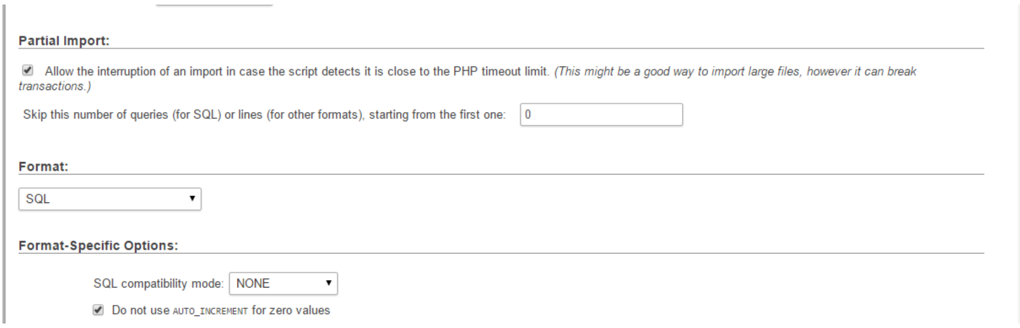
Open the web browser and type URL http://localhost /jvi_sms/ in the address bar, Then click go.



Importing into the current server

File to Import:

Figure A.4 import database



Partial Import:

☑ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a good way to import large files, however it can break transactions.)*

Skip this number of queries (for SQL) or lines (for other formats), starting from the first one: 0

Format:

SQL

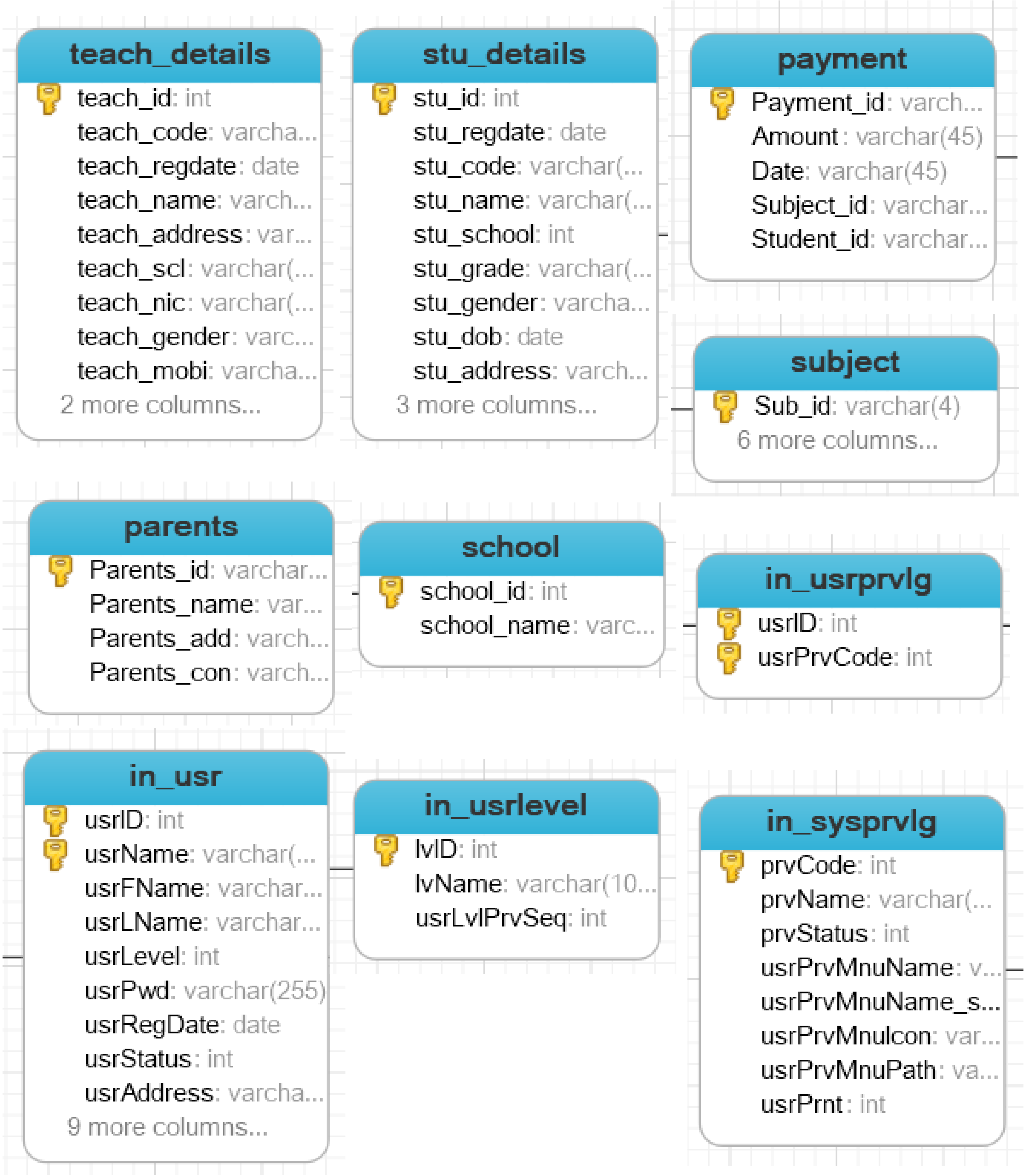
Format-Specific Options:

SQL compatibility mode: NONE

☑ Do not use AUTO_INCREMENT for zero values

APPENDIX B-DESIGN DOCUMENTATION

DATABASE TABLES



teach_details
teach_id: int
teach_code: varcha...
teach_regdate: date
teach_name: varch...
teach_address: var...
teach_scl: varchar(...
teach_nic: varchar(...
teach_gender: varc...
teach_mobi: varcha...
2 more columns...

stu_details
stu_id: int
stu_regdate: date
stu_code: varchar(...
stu_name: varchar(...
stu_school: int
stu_grade: varchar(...
stu_gender: varcha...
stu_dob: date
stu_address: varch...
3 more columns...

payment
Payment_id: varch...
Amount: varchar(45)
Date: varchar(45)
Subject_id: varchar...
Student_id: varchar...

subject
Sub_id: varchar(4)
6 more columns...

parents
Parents_id: varchar...
Parents_name: var...
Parents_add: varch...
Parents_con: varch...

school
school_id: int
school_name: varc...

in_usrprvlg
usrID: int
usrPrvCode: int

in_usr
usrID: int
usrName: varchar(...
usrFName: varchar...
usrLName: varchar...
usrLevel: int
usrPwd: varchar(255)
usrRegDate: date
usrStatus: int
usrAddress: varcha...
9 more columns...

in_usrlevel
lvID: int
lvName: varchar(10...
usrLvlPrvSeq: int

in_sysprvlg
prvCode: int
prvName: varchar(...
prvStatus: int
usrPrvMnuName: v...
usrPrvMnuName_s...
usrPrvMnuIcon: var...
usrPrvMnuPath: va...
usrPrnt: int

Figure B.1: Database Tables

Sequence Diagram for log in to the system

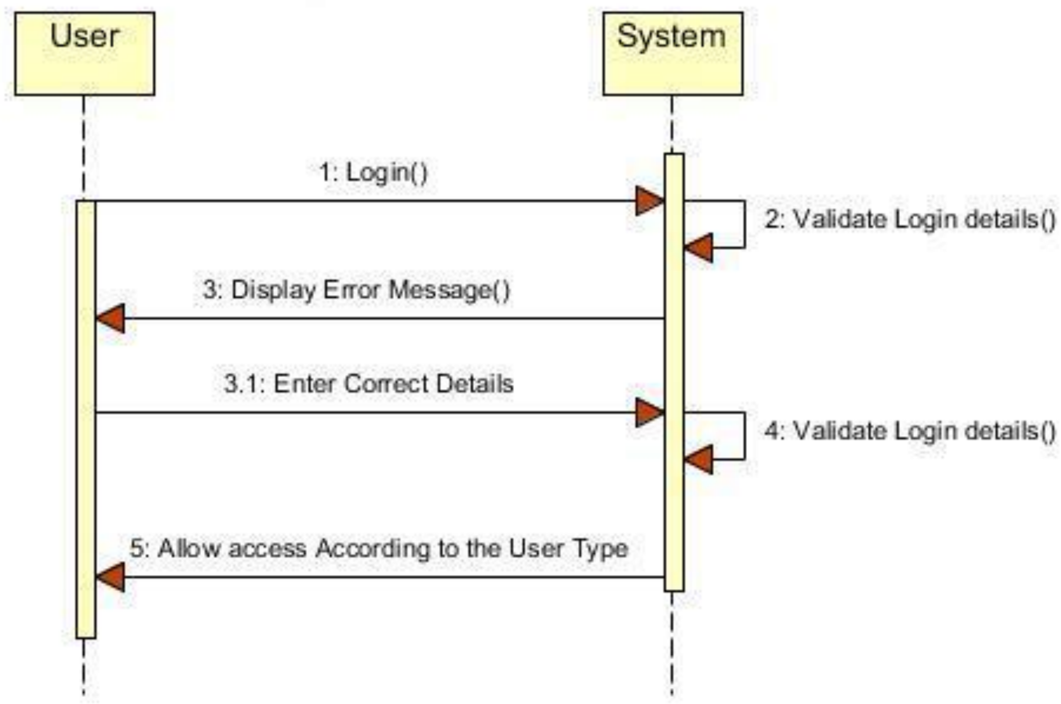

User
System
1: Login()
2: Validate Login details()
3: Display Error Message()
3.1: Enter Correct Details
4: Validate Login details()
5: Allow access According to the User Type

Figure B.2: Sequence Diagram for log in to the system

Sequence Diagram for registering a Student

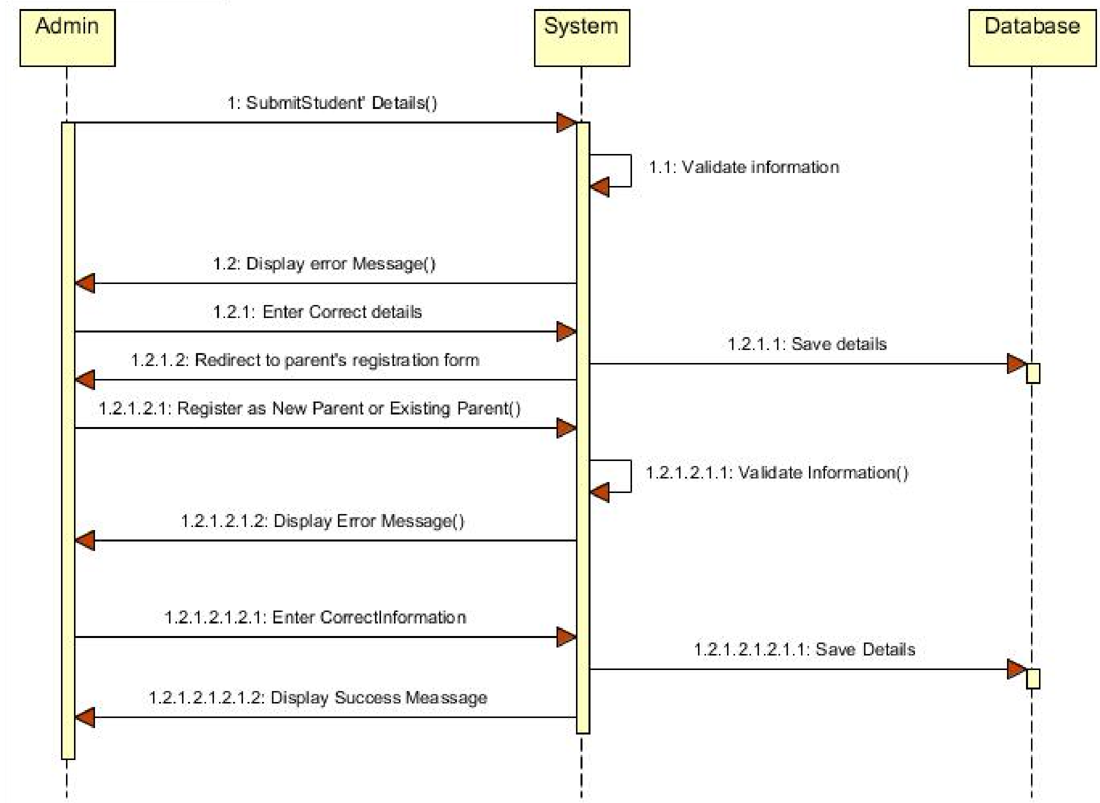

Admin
System
Database
1: SubmitStudent' Details()
1.1: Validate information
1.2: Display error Message()
1.2.1: Enter Correct details
1.2.1.1: Save details
1.2.1.2: Redirect to parent's registration form
1.2.1.2.1: Register as New Parent or Existing Parent()
1.2.1.2.1.1: Validate Information()
1.2.1.2.1.2: Display Error Message()
1.2.1.2.1.2.1: Enter CorrectInformation
1.2.1.2.1.2.1.1: Save Details
1.2.1.2.1.2.1.2: Display Success Meassage

Figure B.3: Sequence Diagram for registering a Student

APPENDIX C-USER DOCUMENTATION

JVISMS was developed with many functionalities and features in order to perform an effective and efficient student management in Joseph vaz Institute-kurunegala. The system has four types of user roles namely Administrator (Manager), Branch manager, Teacher and Trainer. Administrator has full access to all functions and features of the system.

User Login Window

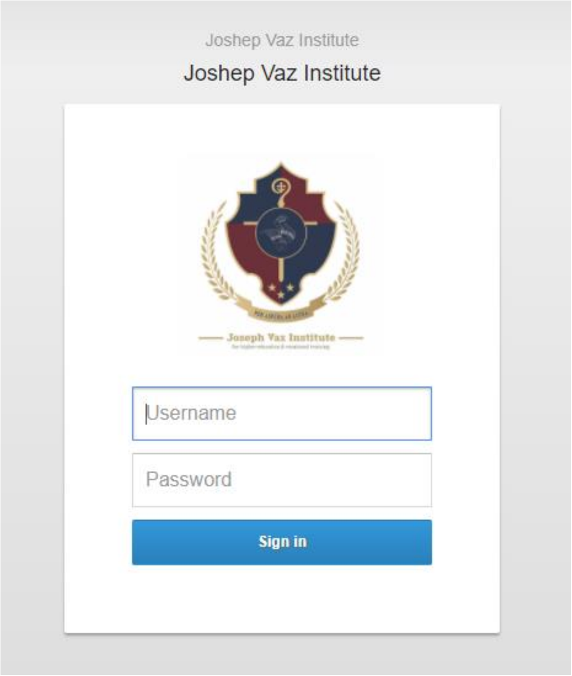
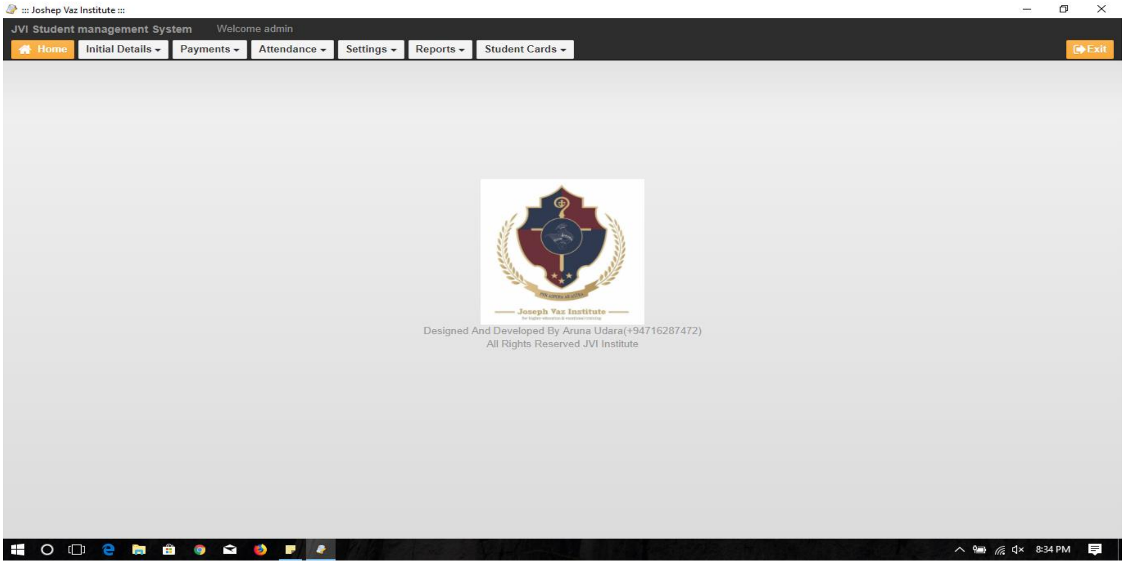


Figure C.1: User Login Window

Login is the initial window that appears when user runs the system. Only the authorized users can access the system functions. Hence user needs to be authorized by providing his/her valid user name and password. Once successfully authenticated, following Main Window (Figure C.2) will appears with the Navigation panel (Figure C.3).

Main Window



Designed And Developed By Aruna Udara(+94716287472)
All Rights Reserved JVI Institute

Figure C.2: Main Window

Main Navigation Panel

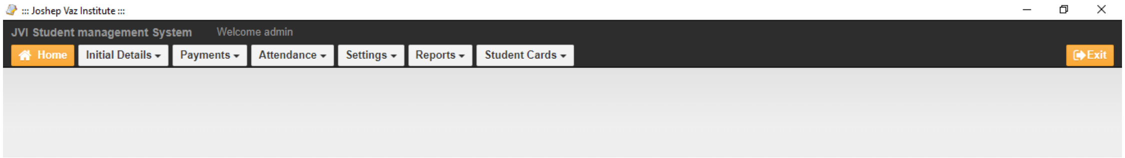


Figure C.3: Main Navigation panel

Home button: this button is use for go back to the main window.

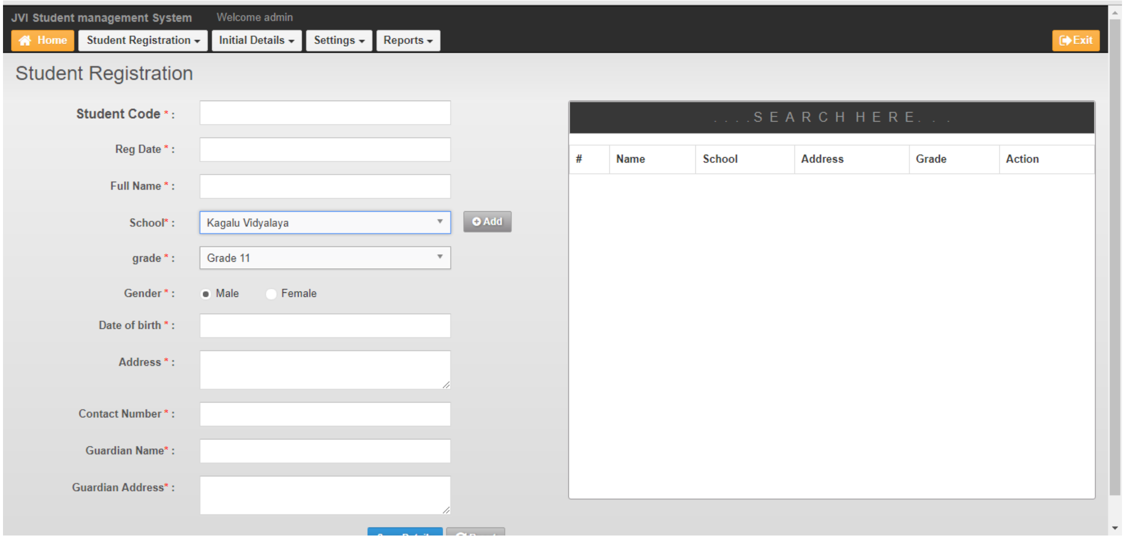
Registrations: All the registration forms are listed in this button.it is easy for the access to all register form.

Initial details: This button through user can handle other forms. (Example: student enrollment)

Setting: This button through can be accessed to management handling form (example: user management.)

Reports: This is the main embedded report generating button that is very useful in Student management process. All the management reports can be generated here.

Student Registration Form



Student Registration

Student Code * :	
Reg Date * :	
Full Name * :	
School* :	Kagalu Vidyalaya
grade * :	Grade 11
Gender * :	● Male ○ Female
Date of birth * :	
Address * :	
Contact Number * :	
Guardian Name* :	
Guardian Address* :	

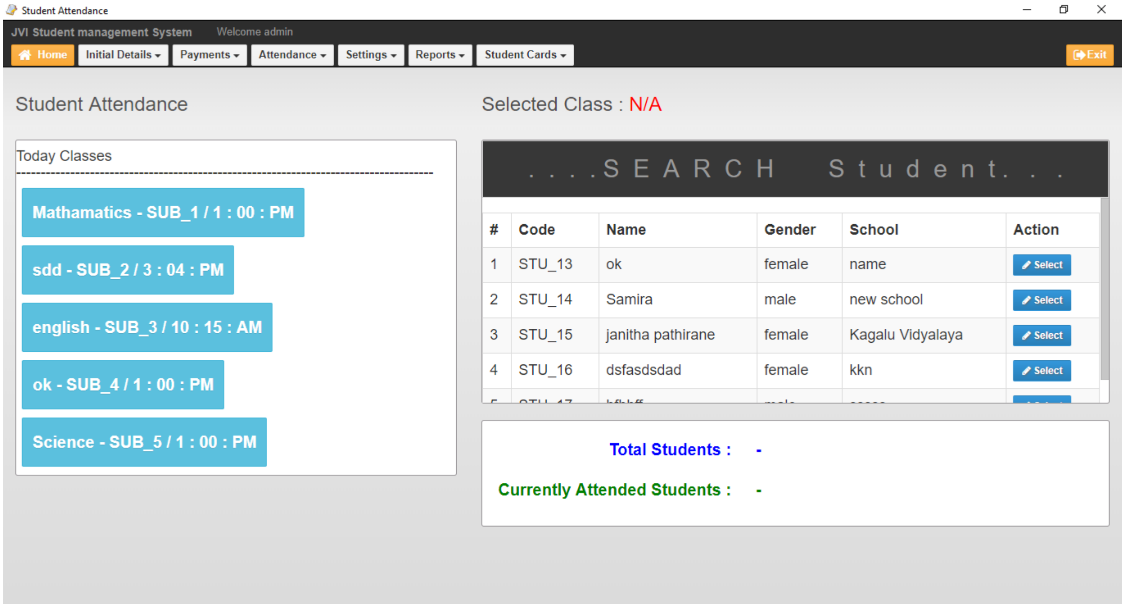
. . . . S E A R C H H E R E . . .

#	Name	School	Address	Grade	Action

Figure C.4: Student Registration Form

Students can be registered with this form (Figure C.4). Click on “Save Details” button after entering valid student details into the form.

Student Attendance Form



Student Attendance

Selected Class : N/A

Today Classes

- Mathamatics - SUB_1 / 1 : 00 : PM
- sdd - SUB_2 / 3 : 04 : PM
- english - SUB_3 / 10 : 15 : AM
- ok - SUB_4 / 1 : 00 : PM
- Science - SUB_5 / 1 : 00 : PM

. . . . S E A R C H S t u d e n t . . .

#	Code	Name	Gender	School	Action
1	STU_13	ok	female	name	Select
2	STU_14	Samira	male	new school	Select
3	STU_15	janitha pathirane	female	Kagalu Vidyalaya	Select
4	STU_16	dsfasdsdad	female	kkn	Select

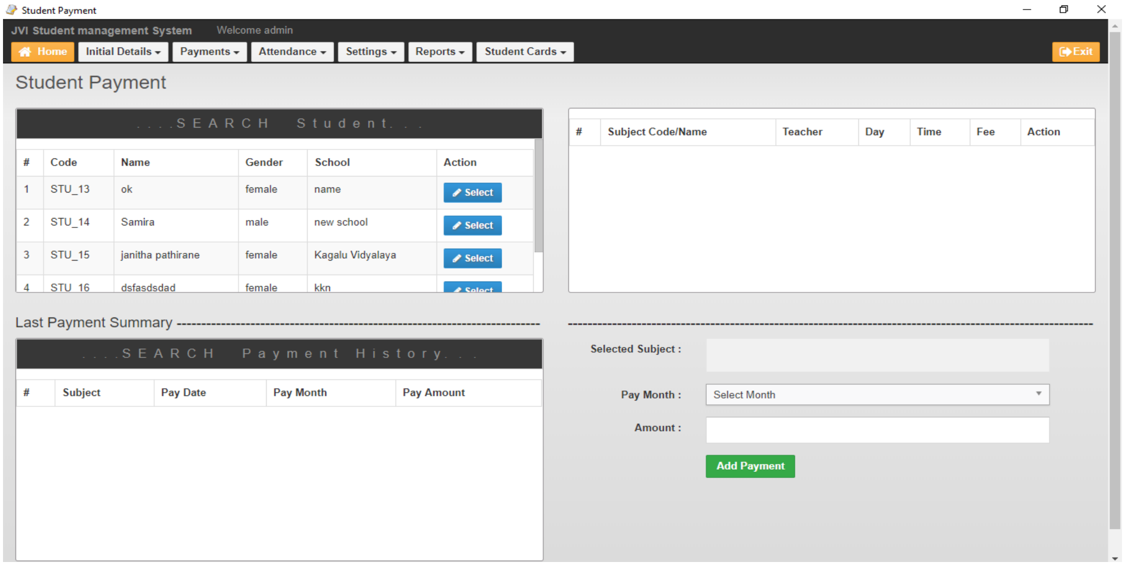
Total Students : -

Currently Attended Students : -

Figure C.5: Student Attendance Form

Attendance of students can be recorded through this window (Figure C.5). After insert student ID to “search hear” field, details will be auto load. Click on “Save” button after entering valid information into the form.

Student Payment Form



Student Payment

. . . S E A R C H   S t u d e n t . . .

#	Code	Name	Gender	School	Action
1	STU_13	ok	female	name	Select
2	STU_14	Samira	male	new school	Select
3	STU_15	janitha pathirane	female	Kagalu Vidyalaya	Select
4	STU_16	dsfasdsdad	female	kkn	Select

#	Subject Code/Name	Teacher	Day	Time	Fee	Action

Last Payment Summary

. . . S E A R C H   P a y m e n t   H i s t o r y . . .

#	Subject	Pay Date	Pay Month	Pay Amount

Selected Subject :

Pay Month : Select Month

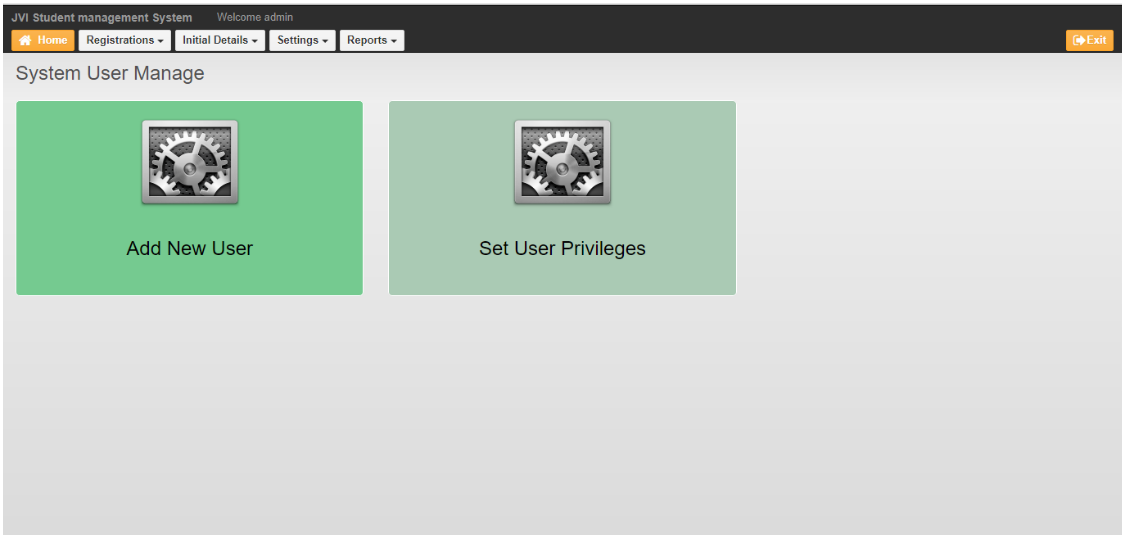
Amount :

Add Payment

Figure C.6: Student Payment Form

All subject Payment Can be handle through this window (Figure C.6). After insert student ID to “search hear” field, details will be auto load.

System User Management Form



System User Manage

Add New User

Set User Privileges

Figure C.7: System User Management Form

This window can access to only administrator (manager). That function will be useful for set user privilege to system actor.

System Setting Form

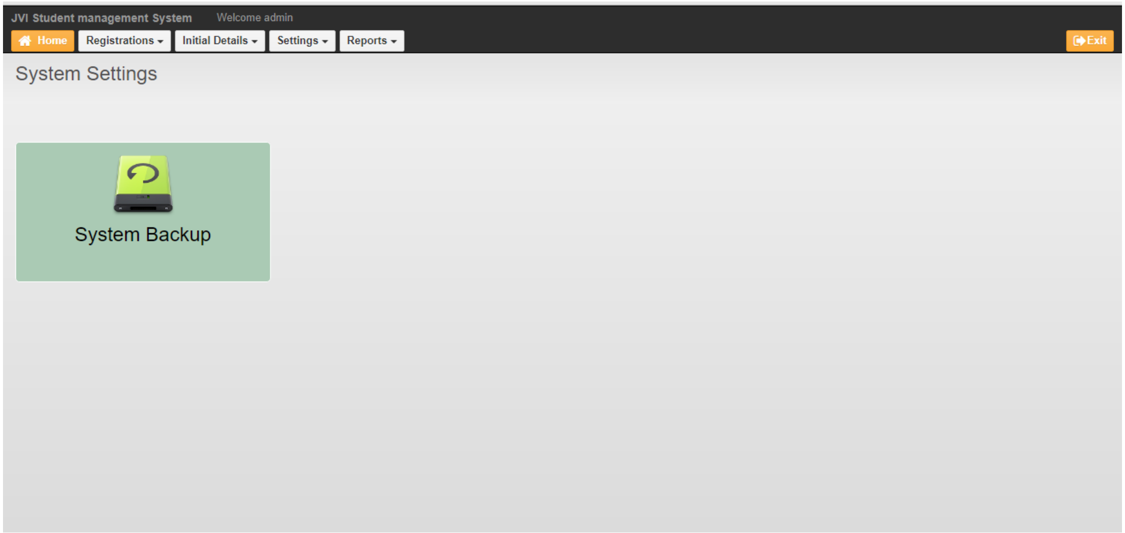
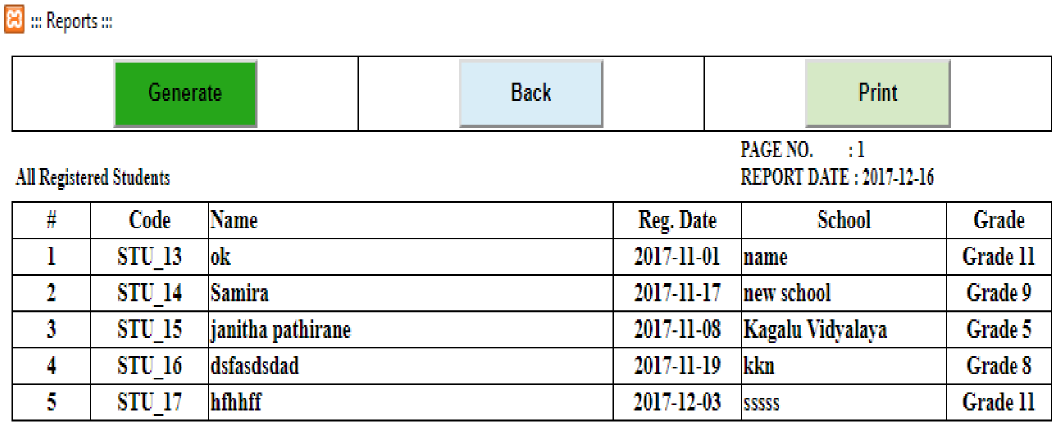


Figure C.8: System Setting Form

Backup is the important thing for system data security. If you want backup data, just click system backup icon in Figure C.8 and do some simple process. It is easy to recover data after its loss, be it by data deletion or corruption no matter your data is still safe with JVISMS backup process.

APPENDIX D-MANAGEMENT REPORTS

View All Students



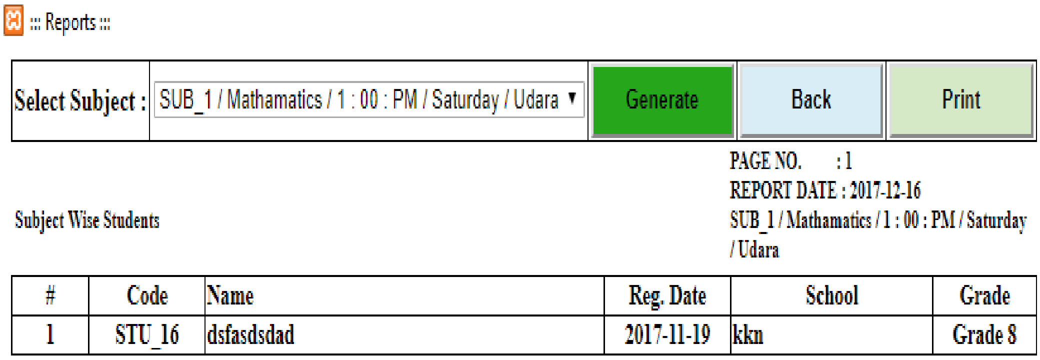
All Registered Students

PAGE NO. : 1
REPORT DATE : 2017-12-16

#	Code	Name	Reg. Date	School	Grade
1	STU_13	ok	2017-11-01	name	Grade 11
2	STU_14	Samira	2017-11-17	new school	Grade 9
3	STU_15	janitha pathirane	2017-11-08	Kagalu Vidyalaya	Grade 5
4	STU_16	dsfasdsdad	2017-11-19	kkn	Grade 8
5	STU_17	hfhhff	2017-12-03	sssss	Grade 11

Figure D.1: All Students details report

View subject vice Students



Select Subject : SUB_1 / Mathamatics / 1 : 00 : PM / Saturday / Udara

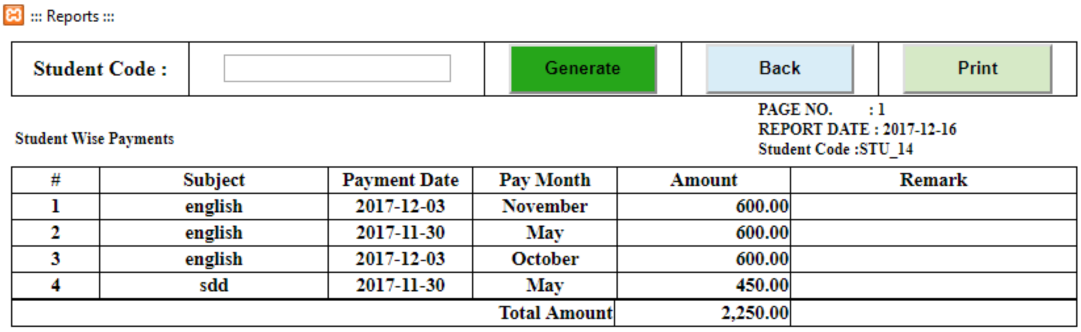
Subject Wise Students

PAGE NO. : 1
REPORT DATE : 2017-12-16
SUB_1 / Mathamatics / 1 : 00 : PM / Saturday / Udara

#	Code	Name	Reg. Date	School	Grade
1	STU_16	dsfasdsdad	2017-11-19	kkn	Grade 8

Figure D.2: subject vice Students details report

View Student vice payment details



Student Code :

Student Wise Payments

PAGE NO. : 1
REPORT DATE : 2017-12-16
Student Code :STU_14

#	Subject	Payment Date	Pay Month	Amount	Remark
1	english	2017-12-03	November	600.00	
2	english	2017-11-30	May	600.00	
3	english	2017-12-03	October	600.00	
4	sdd	2017-11-30	May	450.00	
			Total Amount	2,250.00	

Figure D.3: Student vice payment details report

APPENDIX E – TEST RESULTS

Testing is the most important procedure of any system which is the best way to attract users in a reliable manner. Some of the test results have been displayed below.

Test Results for User Login

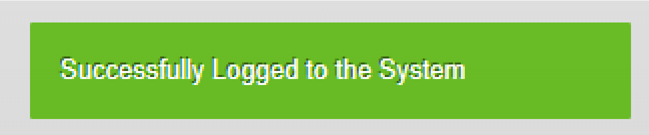
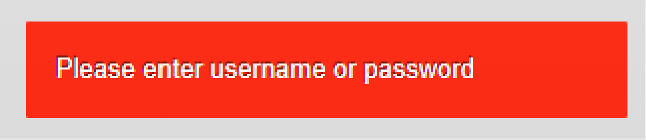
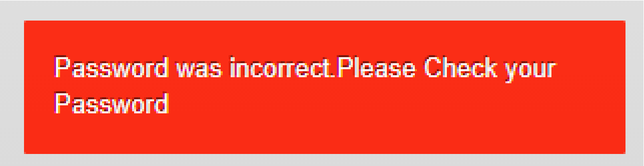
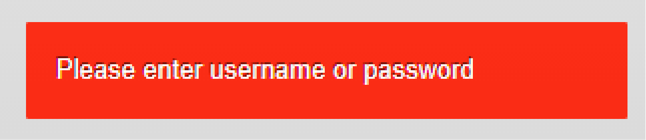
Test ID	Test Result	Status
1	Login with Correct matching Username and password  Successfully Logged to the System	Pass
2	Login with Invalid Username  Please enter username or password	Pass
3	Login with Invalid Password  Password was incorrect.Please Check your Password	Pass
4	Login with Invalid Username and Password  Please enter username or password	Pass

Table E.1: Test result for user login

Test Results for Student Registration

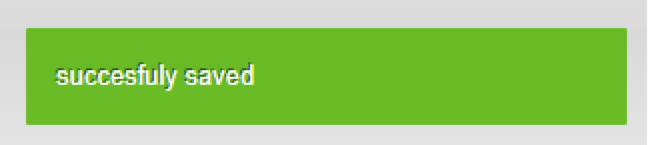
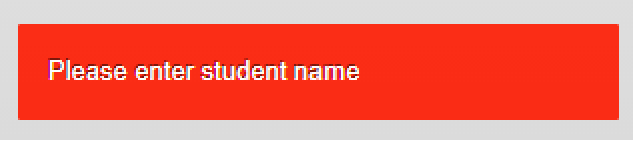
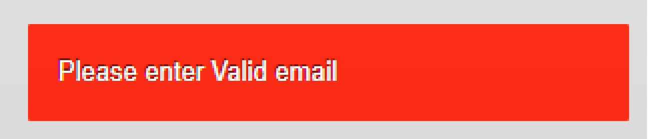
Test ID	Test Result	Status
1	Add records successfully Ex:-Register a new student successfully  succesfuly saved	Pass
2	Leave empty fields Ex:- Empty Name field  Please enter student name	Pass
3	Invalid e-mail entry  Please enter Valid email	Pass

Table E.2: Test result for Student Registration.

Test Results for System Logout

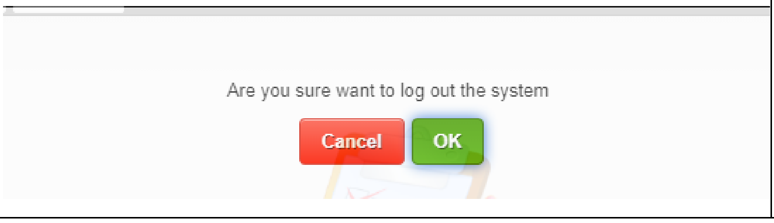
Test ID	Test Result	Status
1	Click Exit Button  Are you sure want to log out the system 	Pass

Table E.3: Test result for System Logout.

APPENDIX F – CODE LISTING

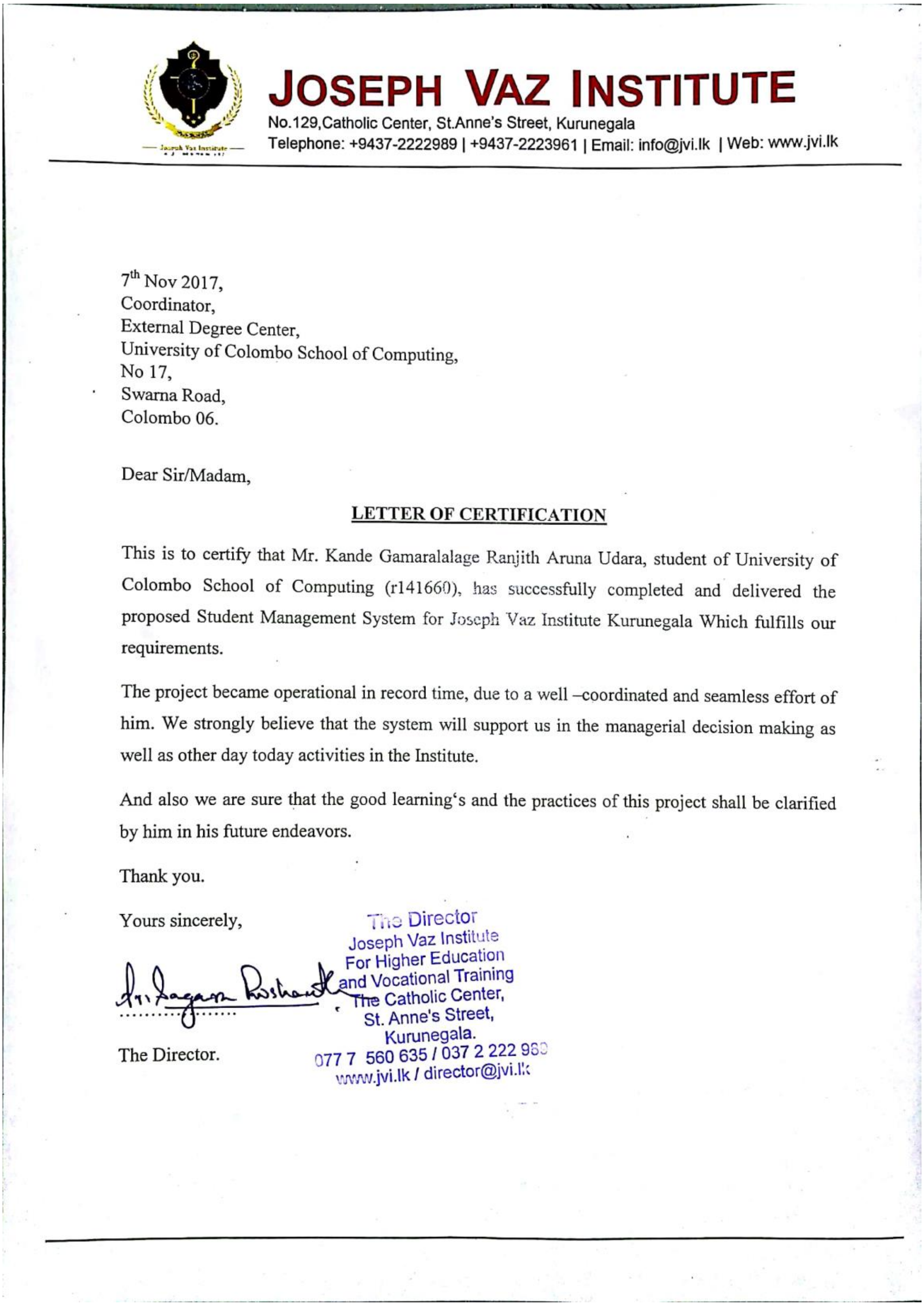
USER LOGIN MODULE

```
<?php
require_once '../config/dbc.php';//include databse connection
require_once '../class/systemSetting.php';
$system = new setting();
var userName = $('#user_name').val();
    var password = $('#pass').val();
if (array_key_exists("logSystem", $_POST)) {
    if (isset($_POST['userName']) && !empty($_POST['userName']) &&
isset($_POST['password']) && !empty($_POST['password'])) {
        $user = ($_POST['userName']);
        $pass = ($_POST['password']);
        $userQuery = "SELECT
in_usr.usrID,
in_usr.usrName,
in_usr.usrPwd,
in_usr.usrStatus,
in_usr.usrLevel
FROM
in_usr
WHERE
(in_usr.usrStatus = '1') AND
in_usr.usrName = '{$user}' LIMIT 1";
        $userAvailability = $system->getCountByQuery($userQuery);
        if ($userAvailability > 0) {
            $userDetails = $system->prepareSelectQuery($userQuery);
            $encriptedPass = sha1('ch' . $pass . 'BIT');//encrypt
password
            foreach ($userDetails as $ud) {
                if ($ud['usrPwd'] == $encriptedPass) {
                    //Set Cookie if select remember btn
                    session_start();
                    $_SESSION['user_id'] = $ud['usrID'];
                    $_SESSION['user_name'] = $ud['usrName'];
                    $_SESSION['user_level'] = $ud['usrLevel'];
                    $_SESSION['usrStatus'] = $ud['usrStatus'];
                    $_SESSION['HTTP_USER_AGENT'] =
md5($_SERVER['HTTP_USER_AGENT']);

                    if (isset($_POST['remember']) &&
$_POST['remember'] == 'r') {
                        setcookie("user_id", $_SESSION['user_id'],
time() + 60 * 60 * 24 * COOKIE_TIME_OUT, "/");
                        setcookie("user_name", $_SESSION['user_name'],
time() + 60 * 60 * 24 * COOKIE_TIME_OUT, "/");
                    }
                    echo json_encode(array(array("msgType" => 0, "msg"
=> "Successfully Logged to the System")));
                } else {
                    echo json_encode(array(array("msgType" => 1, "msg"
=> "Password was incorrect.Please Check your Password")));
                }
            }
        } else {
            echo json_encode(array(array("msgType" => 2, "msg" => "User
was not available in database,plase check your username")));
        }
    } else {
        echo json_encode(array(array("msgType" => 3, "msg" => "Please
enter username or password")));
    }
}



<script type="text/javascript">
function timelyRedirect(url, delay) {
    setTimeout(function () {
        window.location = url;
    }, delay);
}
</script>
$.post(function (e) {                     //e is reply
        if (e === undefined || e.length === 0 || e === null) {
            alertify.error("System Error Occured", 2000);
        } else {
            $.each(e, function (index, msgData) {
                if (msgData.msgType === 0) {
                    alertify.success(msgData.msg, 1500);
                    timelyRedirect("dashboard.php", 1500);
                } else if (msgData.msgType === 1) {
                    alertify.error(msgData.msg, 3000);
                } else if (msgData.msgType === 2) {
                    alertify.error(msgData.msg, 3000);
                } else if (msgData.msgType === 3) {
                    alertify.error(msgData.msg, 3000);
                }
            });
        }
    }, "json");
?>
```

STUDENT REGISTRATION

```
//main model
<?php

require_once '../config/dbc.php';
require_once '../class/systemSetting.php';
$system = new setting();

//save student details
if (array_key_exists("action", $_POST)) {
    if ($_POST['action'] == 'save_stu') {
        $system->prepareCommandQueryForAlertify("INSERT INTO
`stu_details` (`stu_regdate`, `stu_code`, `stu_name`, `stu_school`,
`stu_grade`, `stu_gender`, `stu_dob`, `stu_address`, `stu_contact`,
`stu_gurd_nm`, `stu_gurd_add`) "
                . "VALUES ('{$_POST['regdate']}', '{$_POST['code']}',
'{$_POST['name']}', '{$_POST['scl']}', '{$_POST['grd']}',
'{$_POST['gender']}', '{$_POST['dob']}', '{$_POST['add']}',
'{$_POST['con']}', '{$_POST['gurname']}', '{$_POST['guradd']}');",
"succesfuly saved", "system error");
    }

    if ($_POST['action'] == 'save_teach') {
        $system->prepareCommandQueryForAlertify("INSERT INTO
`teach_details` (`teach_code`, `teach_regdate`, `teach_name`,
`teach_address`, `teach_scl`,`teach_nic`, `teach_gender`,
`teach_mobi`, `teach_hom`, `teach_email`)"
                . " VALUES ('{$_POST['code']}', '{$_POST['regdate']}',
'{$_POST['name']}', '{$_POST['add']}',
'{$_POST['scl']}','{$_POST['nic']}', '{$_POST['gender']}',
'{$_POST['mobile']}', '{$_POST['home']}',
'{$_POST['email']}');","succesfuly saved", "system error");
    }
?>
//control
function save_stu_data() {
    var code = $('#stu_code').val();
    var regdate = $('#reg_date').val();
    var name = $('#stu_name').val();
    if (name.length == 0) {
        alert('Please enter Valid Email');
        return;
    }
    var scl = $('#stu_school').val();
    var grd = $('#stu_grade').val();
    var gender = "";
    if (document.getElementById('male').checked) {
        gender = 'male';
    } else {
        gender = 'female';
    }
    //alert(gender)
    var dob = $('#stu_dob').val();
    var add = $('#stu_address').val();
    var con = $('#stu_tp_num').val();
    var gurname = $('#stu_gurd_name').val();
    var guradd = $('#stu_gurd_address').val();

    alert('code :' + code + 'regdate:' + regdate + 'name:' + name +
'scl :' + scl + 'grd :' + grd + 'gender :' + gender + 'dob :' + dob +
'con :' + con + 'add :' + add + 'gurname :' + gurname + 'guradd :' +
guradd);

    var data_array = {action: 'save_stu', code: code, regdate: regdate,
name: name, scl: scl, grd: grd, gender: gender, dob: dob, add: add,
con: con, gurname: gurname, guradd: guradd};
    $.post('models/main_model.php', data_array, function (reply) {
        alertifyMsgDisplay(reply, 3000);
        clear_stu_data();
    }, 'json');
}

//view


<script type="text/javascript">
    $(function () {
 $('#save_stu').click(function () {
            save_stu_data();
                  }
                  }
</script>
```

APPENDIX G – CLIENT CERTIFICATE



JOSEPH VAZ INSTITUTE

No.129,Catholic Center, St.Anne's Street, Kurunegala
Telephone: +9437-2222989 | +9437-2223961 | Email: info@jvi.lk | Web: www.jvi.lk

7th Nov 2017,
Coordinator,
External Degree Center,
University of Colombo School of Computing,
No 17,
Swarna Road,
Colombo 06.

Dear Sir/Madam,

LETTER OF CERTIFICATION

This is to certify that Mr. Kande Gamaralalage Ranjith Aruna Udara, student of University of Colombo School of Computing (r141660), has successfully completed and delivered the proposed Student Management System for Joseph Vaz Institute Kurunegala Which fulfills our requirements.

The project became operational in record time, due to a well –coordinated and seamless effort of him. We strongly believe that the system will support us in the managerial decision making as well as other day today activities in the Institute.

And also we are sure that the good learning‘s and the practices of this project shall be clarified by him in his future endeavors.

Thank you.

Yours sincerely,

....

The Director.

The Director
Joseph Vaz Institute
For Higher Education
and Vocational Training
The Catholic Center,
St. Anne's Street,
Kurunegala.
077 7 560 635 / 037 2 222 989
www.jvi.lk / director@jvi.lk

GLOSSARY

- Ajax - Asynchronous JavaScript Technology and XML, which is a new way to use existing standards.
- Cascading Style Sheets (CSS) – This is a language that is used to format the content of web pages.
- Database - Collection of data that is organized.
- HTML–Hyper Text Markup Language. This is a highly used markup language for web pages.
- Java script - This is a dynamic computer programming language.
- MySQL - A relational database management system.
- Object Oriented Designing - It is a method that uses objects to develop a system.
- Use Case – A diagram which is used to show the relationship between the actor and the relevant process.
- Validation – Checking whether the system satisfies user requirements.
- Xamp Server – Xamp Server is a Windows web development environment.

INDEX