

**ONLINE PRE-SCHOOL MANAGEMENT SYSTEM
FOR
URUMPIRAI KIDS' PARK**

A.Sangeetha

2017



**ONLINE PRE-SCHOOL MANAGEMENT SYSTEM
FOR
URUMPIRAI KIDS' PARK**

A.Sangeetha

BIT Registration Number: R141669

Index Number: 1416693

Name of the supervisor

Mr K.Venugoban

December 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**

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 interlibrary loans, and for the title and abstract to be made available to outside organizations.



Date: 04/11/2017

A.Sangeetha

(Name of Candidate)

Countersigned by:



Date: 04/11/2017

Mr.K.Venugoban

(Name of Supervisor)

ABSTRACT

Jaffna is the major education city of the Northern Province of Sri Lanka. Kids' Park is one of the best model preschool in Jaffna district which has several branches located around the Jaffna Peninsula. They provide best education for children from the ages of infancy until six. It has specially trained preschool teachers and facilitate each child's learning through continuous observation and assessment.

“Kid's Park” still keeps their records manually and faces many difficulties when using manual system. This system is generally very bulky both to handle and to store. The details of the students are recorded when students are admitted to the preschool in record books. Details of staff are gathered each time a new staff is appointed, these details are kept in record books. With that, every time the details are updated. Changes of address, telephone numbers and other changes are not updated in manual procedure.

The purpose of the project is to provide an efficient and effective online system to Kids' Park to manage their details of students, staff and handle the payments process. With the help of this online system parents can easily see their student progress and maintain their relationship with the school. The administrator can also handle the admission processes, manage the registered details of students and staff.

This Online Pre-school management system has mostly been developed using some modern open sourced tools and software. PHP which is a powerful server side scripting language has been used for server side scripting along with the Apache web server and MySQL which is a relational database management system. The developed system has been properly tested using a comprehensive testing procedure which ensures a high-quality system.

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

First of all, I would like to express my special gratitude and thanks to BIT Coordinator of the UCSC and to the Project Examination Board for giving me this valuable opportunity to follow a world recognized degree program.

Foremost, I wish to express my deepest gratitude to the director and staff of the Urumpirai Kids' Park who granted me permission to undertake the software project and providing necessary information regarding the project and also for their support in completing the project.

I would like to express my special thanks of gratitude to my supervisor Mr. K.Venugoban for devoting so much time to guidance and he give me lot of ideas, and encouragements to fulfil the project work. And thank my lecturers Mr.S.Theivamainthan and Mr.Y.Yogaseelan to provide basic programming knowledge, instruction.

Finally, I would like to thank my parents who helped me a lot in finalizing this project within the limited time frame.

TABLE OF CONTENTS

DECLARATION.....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT	iv
TABLE OF CONTENTS.....	v
LIST OF FIGURES	viii
LIST OF TABLES	xii
LIST OF ACRONYMS	xiii
CHAPTER 1: INTRODUCTION	1
1.1 INTRODUCTION.....	1
1.2 MOTIVATION	1
1.3 OBJECTIVE OF THE PROJECT.....	2
1.4 SCOPE OF THE PROJECT.....	2
1.5 CRITICAL FUNCTIONALITY FOR PROJECT.....	3
1.6 STRUCTURE OF THE DISSERTATION	3
CHAPTER 2: ANALYSIS	4
2.1 EXISTING MANUAL SYSTEM.....	4
2.2 DRAWBACKS OF EXISTING MANUAL SYSTEM	5
2.3 REQUIRMENT ANALYSIS FOR MANUAL SYSTEM	5
2.4 REQUIRMENT GETHERING	5
2.4.1 FUNCTIONAL REQUIREMENTS	5
2.4.2 NON-FUNCTIONAL REQUIREMENT	6
2.4.3 RESOURCE REQUIREMENTS FOR PROJECT	7

2.5 FEASIBILITY STUDY FOR THE PROPOSED SYSTEM	7
2.5.1 OPERATIONAL FEASIBILITY	8
2.5.2 TECHNICAL FEASIBILITY	8
2.5.3 ECONOMICAL FEASIBILITY	8
2.6 LITERATURE REVIEW	8
CHAPTER 3: DESIGN OF SOLUTION	11
3.1 DEVELOPMENT METHODOLOGY	11
3.1.1 WATERFALL MODEL	11
3.2 ALTERNATE SOLUTION	13
3.3 PROCESS DESIGN OF THE SYSTEM	13
3.3.1 DATABASE DESIGN.....	13
3.3.2 ARCHITECTURE DESIGN	16
3.3.3 INTERFACE DESIGN.....	21
CHAPTER 4: IMPLEMENTATION	26
4.1 IMPLEMENTATION ENVIRONMENT	26
4.2 REUSED MODULES AND COMPONENTS	27
4.3 NETWORK IMPLEMENTATION	28
4.4 CODE AND MODULE STRUCTURE.....	28
4.5 SECURITY	35
CHAPTER 5: EVALUATION	36
5.1 TESTING PROCEDURE	36
5.1.1 UNIT TESTING	36
5.1.2 INTEGRATION TESTING.....	36
5.1.3 SYSTEM TESTING	36
5.1.4 ACCEPTANCE TESTING.....	37
5.2 TESTING PLANS FOR OUR SYSTEM	37
5.2.1 NEW USER REGISTER MODULE (WITHOUT LOGIN)	37

5.2.2 ADMINISTRATION (DIRECTOR) MODULE	38
5.2.3 BRANCH HEAD MODULE.....	39
5.2.4 STAFF MODULE	41
5.2.5 PARENT MODULE.....	42
5.2.6 COMMON FUNCTION MODULE.....	43
5.3 TEST DATA AND TEST RESULTS	45
CHAPTER 6: CONCLUSION	49
6.1 CONCLUSION.....	49
6.2 LESSION LEARNT	49
6.3 FUTURE WORK.....	50
REFERENCES	51
APPENDIX.....	52
APPENDIX A - SYSTEM DOCUMENTATION.....	52
APPENDIX B - DESIGN DOCUMENTATION	56
APPENDIX C – USER DOCUMENTATION	61
APPENDIX D - MANAGEMENT REPORT	96
APPENDIX E - TEST RESULTS	101
APPENDIX F - CODE LISTINGS.....	103
APPENDIX G - CLIENT CERTIFICATE.....	112
GLOSSARY	113

LIST OF FIGURES

Figure 2. 1: Use-Cases for Student Management System in Existing Manual system...	4
Figure 2. 2: KinderCare	9
Figure 2. 3: Little Amigos.....	9
Figure 2. 4: Rainbow preschool	10
Figure 3. 1: Waterfall model.....	12
Figure 3. 2: ER Diagram	15
Figure 3. 3: Use case diagram for student management	16
Figure 3. 4: Use case diagram for pre-school management.....	17
Figure 3. 5: Activity diagram for user login	19
Figure 3. 6: Sequence diagram for student attendance	20
Figure 3. 7: Data Flow Diagram Level 0.....	21
Figure 3. 8: Home Page	22
Figure 3. 9: Login Page.....	23
Figure 3. 10: Student Registration Form.....	24
Figure 3. 11: Report of students' details – class wise.....	25
Figure 3. 12: Print Report	25
Figure 4. 1: Network implementation	28
Figure 4. 2: Module Structure	29
Figure 4. 3: Directory.....	30
Figure 5. 1: User Acceptance chart.....	46
Figure 5. 2: Client Acceptance Report.....	47
Figure 5. 3: User Acceptance Report.....	48
Figure A. 1: WampServer Installation Step 1	52
Figure A. 2: WampServer Installation Step 2	53
Figure A. 3: WampServer Installation Step 3	53
Figure A. 4: WampServer Installation Step 4	54
Figure A. 5: WampServer Installation Step 5	54
Figure A. 6: WampServer Installation Step 6	55
Figure A. 7: WampServer Installation Step 7	55
Figure B. 1: Activity diagram for student attendance.....	56

Figure B. 2: Activity diagram for staff leave management	57
Figure B. 3: Sequence diagram for student performance	58
Figure B. 4: Sequence diagram for staff leave.....	59
Figure B. 1: Data Flow Diagram Preschool management system level 1.....	60
Figure C. 1: Index Page	62
Figure C. 2: Login page with wrong user name.....	63
Figure C. 3: Incorrect Password.....	63
Figure C. 4: About Us Menu.....	64
Figure C. 5: Gallery Menu	64
Figure C. 6: Contact Us Menu	65
Figure C. 7 Director Interface	66
Figure C. 8: Dropdown User Menu	67
Figure C. 9: Profile	67
Figure C. 10: Change Password.....	68
Figure C. 11: Staff menu.....	68
Figure C. 12: View Staff Details.....	69
Figure C. 13 Add New Staff	70
Figure C. 14: Data successfully added message	70
Figure C. 15: Print staff details	71
Figure C. 16: Print page	71
Figure C. 17: View staff additional information.....	72
Figure C. 18: Edit Staff	73
Figure C. 19: Delete Staff	73
Figure C. 20: Staff leave confirmation table.....	74
Figure C. 21: Staff's salary	74
Figure C. 22: Payment menu	75
Figure C. 23: Class Rate	75
Figure C. 24: Student Event Payment Report.....	76
Figure C. 25: Print event payment report.....	76
Figure C. 26: Add event payment	77
Figure C. 27: Students' class payment.....	77
Figure C. 28: Add Class Payment.....	78
Figure C. 29: Students Details	79

Figure C. 30: Add New Staff	80
Figure C. 31: Select parent.....	80
Figure C. 32: Add new parent	81
Figure C. 33: Student personal information.....	82
Figure C. 34: Add Class Information.....	83
Figure C. 35: Mange branch detail	83
Figure C. 36: Subject detail	84
Figure C. 37: Class Subject Map	84
Figure C. 38: Branch head profile menu.....	85
Figure C. 39: Manage Class Details	85
Figure C. 40: Staff Profile Menu	85
Figure C. 41: Expense Details	86
Figure C. 42: Apply Leave.....	86
Figure C. 43: Message	87
Figure C. 44: Teacher Interface	88
Figure C. 45: Students Menu	89
Figure C. 46: Students' attendance	89
Figure C. 47: Add your student marks.....	90
Figure C. 48: Students' Marks Report	90
Figure C. 49: Print student report	91
Figure C. 50: Parent Interface	92
Figure C. 51: Kid's detail in parent interface	93
Figure C. 52: Kid's attendance report.....	93
Figure C. 53: Absent detail	94
Figure C. 54: Add new absent.....	94
Figure C. 55: Kid's marks report	95
Figure D. 1: Student Report Class Wise	96
Figure D. 2: Student Report Division Wise	97
Figure D. 3: Student Report Branch Wise	97
Figure D. 4: Staff Branch wise	98
Figure D. 5: Student attendance report	98
Figure D. 6: Expense Report.....	99
Figure D. 7: Student marks branch wise	99

Figure D. 8: Staff Salary 100

LIST OF TABLES

Table 3. 1: Alternate Solution	13
Table 3. 2: Use case Narrative	18
Table 4. 1: Implement Requirement	26
Table 5. 1: New User Module	37
Table 5. 2: Director Module.....	39
Table 5. 3: Branch Head Module	40
Table 5. 4: Staff Module	42
Table 5. 5: Parent Module.....	43
Table 5. 6: Common Module	45
Table E. 1: Test Result.....	101
Table E. 2: Common Test Result.....	102
Table E. 3: Report Test	102

LIST OF ACRONYMS

HTML – Hypertext Markup Language

PHP – Hypertext Preprocessor

CSS – Cascading Style Sheets

SQL – Structured Query Language

SMS – Short Message Service

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

Jaffna is the major education city of the Northern Province of Sri Lanka. Kids' Park is one of the best model preschool in Jaffna district which has several branches located around the Jaffna Peninsula. They provide best education for children from the ages of infancy until six. It has specially trained preschool teachers and facilitate each child's learning through continuous observation and assessment.

1.2 MOTIVATION

Preschool education focuses on educating children from the ages of infancy until six. Kids' Park is one of the famous model preschool in Jaffna district which has several branches located around the Jaffna Peninsula. It has specially trained preschool teachers and facilitate each child's learning through continuous observation and assessment. It also manages the evening courses for kids such as CALSDA, Art and Design, etc.

“Kid's Park” still keeps their records manually and faces many difficulties when using manual system. The details of the students are recorded when students are admitted to the preschool in record books. Details of staff are gathered during the appointment to the preschool in record books and kept. With that, every time the details are updated. Changes of address, telephone numbers and other changes are not updated in manual procedure.

Windows based or web based management systems are suitable to manage this preschool but geographic location of the branches and users are different, so the web based system is most suitable. The system is planned to be developed using Open Source tools, such as PHP & MySQL to build the system and HTML, CSS, JavaScript and AJAX for designing and validation.

The purpose of my project is to provide an efficient and effective online system to Kids' Park to manage the details of students and staff and handle the payments process.

With the help of this online system parents can also see their student progress and maintain their relationship with the Preschool. The Administrator (Director) can also handle the admission processes, manage the registered details of students and staff easily.

1.3 OBJECTIVE OF THE PROJECT

Preschool administration can be able to:

- Manage the details of student and staff and maintain up-to-date information.
- Manage students' attendance. If a student is absent system automatically check with parents' report and alert them via sms.
- Manage staff leave with director or branch head approvals.
- Create staff's monthly salary report.
- Manage students' event, class payment details.
- Teachers can enter their students' marks and their extracurricular activities and easily generate the student's report.
- Manage events organized by the school and can convey event details to the parents.
- Easy to communicate with system users through message (SMS).
- Generate daily, monthly, annual and timely report with print option.

Parents can be able to:

- Find their students' overall performance including their extracurricular activities.
- Easily inform through absent report about the absent of the particular student to School.
- Manage their event or monthly payment details.
- Easy to communicate with teachers or administration of the school through message (SMS).

1.4 SCOPE OF THE PROJECT

The scope of this project is to develop an online Preschool management system for Kids' Park, which support the core functioning of managing details of students and

staff, students attendance, staff leave, salary and student payment in order to carry out the administration effectively and efficiently.

1.5 CRITICAL FUNCTIONALITY FOR PROJECT

- The users who lack knowledge of information technology may face some problem using this system.
- All users may not have internet facilities at their location.
- Initial cost may be expensive for the client.

1.6 STRUCTURE OF THE DISSERTATION

This dissertation will describe about this project as follows. The structure of dissertation: The introduction is discussed in the chapter1, chapter 2 talks about analysis; in this chapter2 we discussed about how we analysed the problem and list down the functional and non-functional requirements. Chapter 3 talks about design; how the user interact with system and is shown by diagram. Chapter 4 talks about implementation; how the system modules and structure and chapter 5 talks about testing, how to evaluate the system with details study. Chapter 6 is about conclusion; appendix is annexed additional document about this project and finally glossary is included.

CHAPTER 2: ANALYSIS

System Analysis is an important phase of the software development lifecycle. System analysis as a problem-solving technique that breaks down a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose. This chapter deals with the present situation of the “Urumpirai Kids’ Park”. It based on the gathering techniques used to gather problems of current manual system of the institute and achieve the project goal. Finally we will compare the existing system with the functional and non-functional requirement of the proposed system.

2.1 EXISTING MANUAL SYSTEM

Urumpirai Kids’ Park management system is still managed manually. So, this system contains human errors and data redundancy.

Use-case diagram for existing manual system is shown in the following Figures 2.1:

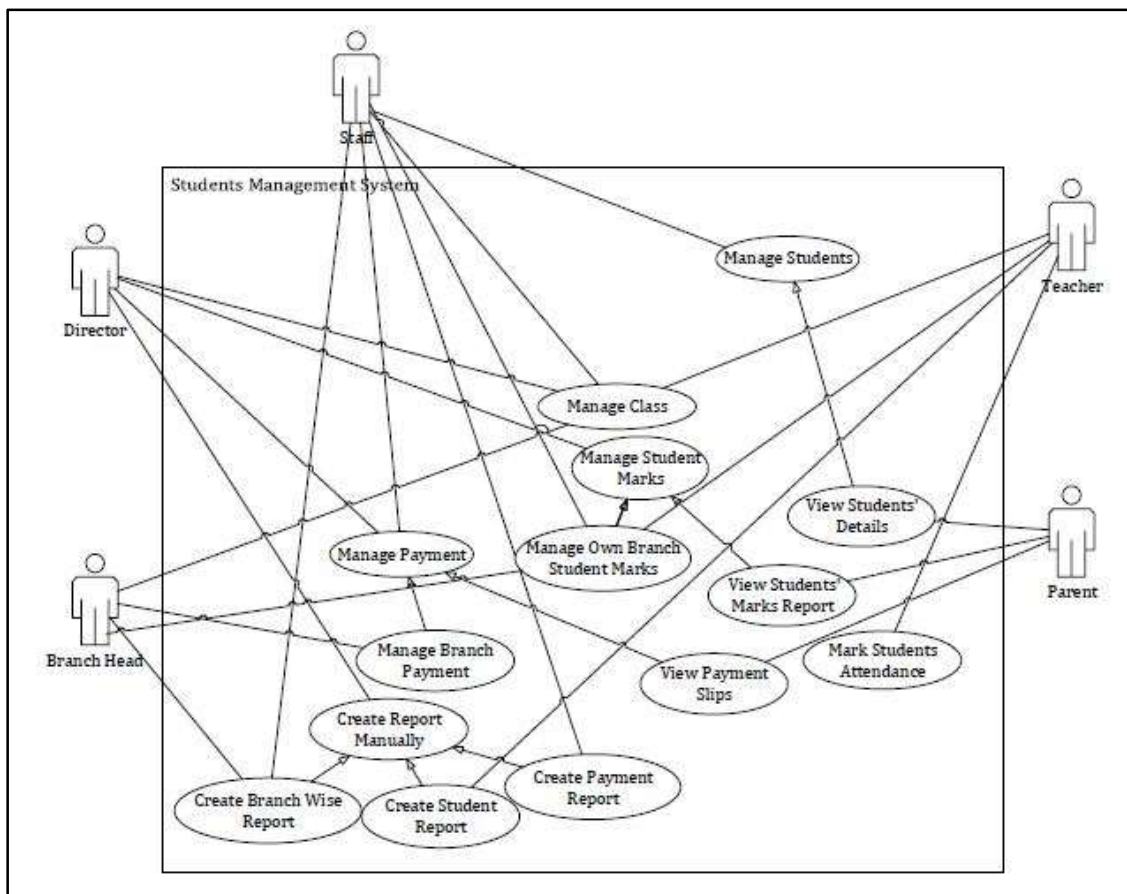


Figure 2. 1: Use-Cases for Student Management System in Existing Manual system

2.2 DRAWBACKS OF EXISTING MANUAL SYSTEM

The following major drawbacks have been identified in the existing manual management system.

- The risk of errors is greater.
- No backups for the confidential data.
- Quality of output is less consistent and often not well-designed.
- Suffering and time wasting with paper works.
- Poor relationship between parents and the management.
- Paper based systems are generally very bulky both to handle and to store, and office space is expensive.
- Collaboration on documents is extremely hard when working with paper versions.

2.3 REQUIRMENT ANALYSIS FOR MANUAL SYSTEM

Preschool management system for Urumpirai Kids' park is done by manually with loads of paper work. Paper is very vulnerable to valuable data being lost. In cases of fire or flood, school systems that rely on paper files are left without a back-up plan; the data is simply destroyed. And everybody's handwriting is different, and what is legible to one person is not always legible to the next.

2.4 REQUIRMENT GETHERING

Requirement gathering or requirement eliciting is the process of addressing the needs and conditions of the new system.

2.4.1 FUNCTIONAL REQUIREMENTS

A functional requirement document defines the functionality of a system or one of its subsystems. It also depends upon the type of software, expected users and the type of system where the software is used. Functional user requirements may be high-level statements of what the system should do but functional system requirements should also describe clearly about the system services in detail.

Requirements of the Urumpirai kids' park

- Entering and storing the details of students and staff.
- Kids' park can store and enter their day to day activity details.
- Parents can view their student's performance and attendance.
- Director and staffs can manage class, payment and students marks.
- Director can create and view all kinds of report.
- Manage expenditure.
- Teacher can generate their students' reports.
- Get parents' feedback.
- Make friendly Graphical User Interface (GUI)

2.4.2 NON-FUNCTIONAL REQUIREMENT

Non-functional requirements “in systems engineering and requirement 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 behaviours.”

- Security: It is a very important thing. Only the authorized persons should be allowed to operate the system with their own username and password only.
- Accuracy and consistency: It is very important. In this project there exist quite a few functionalities which accuracy is very vital. Entering data should be validated by the system.
- Reliability: The system should accurately performs student registration, report generation, manage staff's leave and students' payment.
- User friendliness: It is very important to the success of the system. In this system most of the users related to know basic computer skill. Seven though User interfaces should be designed as attractive.
- Portability of the system: This system will work on all the Windows based operating systems well, which are above Windows XP.
- Response time & processing time: Although this was not specified by the users, in general it is needed that the system should give response to the user without much delay.

- Availability: The system are available to users anytime, anywhere, just need a PC and Internet Connection. Also the system work in multiple web browsers like (Chrome, Mozilla, Opera, and Internet Explorer).

2.4.3 RESOURCE REQUIREMENTS FOR PROJECT

Hardware requirements:

- Pentium 4 computer or above.
- Basic printer for report printing.

Software requirements:

For implementation purpose

- Any operating system.
- Any browsers.

For development purpose

- Windows Operating Systems
- XAMPP Sever v2.3.3
- Adobe Dreamweaver CC
- Microsoft Visio 2013
- MySQL Workbench 6.0.8 CE
- CSS
- JavaScript
- MS word
- Adobe Photoshop CS2

Special requirements:

- Internet facility
- Web hosting and domain registration

2.5 FEASIBILITY STUDY FOR THE PROPOSED SYSTEM

Feasibility study is a very process in order to find out the strengths, weaknesses, opportunities and threats of a proposed system to full fill the main requirements. A detailed feasibility study was carried out regarding this system as following facts.

2.5.1 OPERATIONAL FEASIBILITY

The functions of proposed system were analysed to see whether they accomplish the requirements.

2.5.2 TECHNICAL FEASIBILITY

It is a measure of the practicality of a specific technical solution and the availability of technical resources and expertise.

2.5.3 ECONOMICAL FEASIBILITY

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the institute that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis.

2.6 LITERATURE REVIEW

This section summarizes various techniques that have been employed closely related with this system. This system has some literature reviews that are on preschool management system. The following similar systems were used for detailed study.

1. KinderCare

KinderCare [1] is using an online web based system. In this system they provide larger amount of their data. They provide online registration, online payment, view all branches and so own. They listed all of their staff. This system is private. The “KinderCare” front page following Figure 2.2 KinderCare:

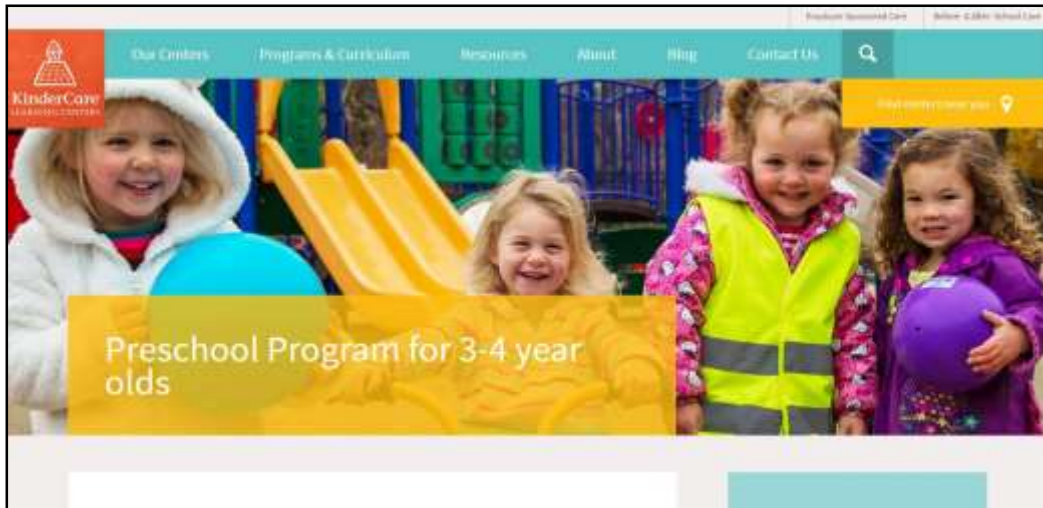


Figure 2. 2: KinderCare

2. Little Amigos

Little Amigos [2] is using an online an online web based system. They provide online registration, online payment, parents review, and mailing facilities. They listed all of their staff. The “Little Amigos” front page following Figure 2.3 Little Amigos:



Figure 2. 3: Little Amigos

3. Rainbow preschool

Rainbow [3] is using an online web based system. They provide admission contact form, payment form, and their contact details. The “Rainbow preschool” front page following Figure 2.4 Rainbow preschool:



Figure 2. 4: Rainbow preschool

In best of my knowledge, the above three systems are not suitable for Smart Kids Pre-School. Because most of the functionality of Smart Kids Pre-School not available in these given three management systems.

CHAPTER 3: DESIGN OF SOLUTION

In the phase of design the system architecture has been developed. The design phase has been initialized with the required document delivered by the information or requirement gathering phase and map the gathered requirements in the information gathering phase into architecture. This architecture defines the components, their interfaces and behaviours of that particular newly developing system. The architecture is describes a plan of the system using by diagrams.

3.1 DEVELOPMENT METHODOLOGY

The Software Development Methodology (SDM) to be used in this project is the Software Development Life cycle (SDLC). Where there is a structured set of activities required to develop a software system required to develop a software system.

- Specification
- Design
- Validation
- Evolution

3.1.1 WATERFALL MODEL

Waterfall model is an example of Sequential model. In this model, the software development activity is divided into different phases and each phase consists of series of tasks and has different objectives. The water fall is shown in the following Figure 3.1: Waterfall model

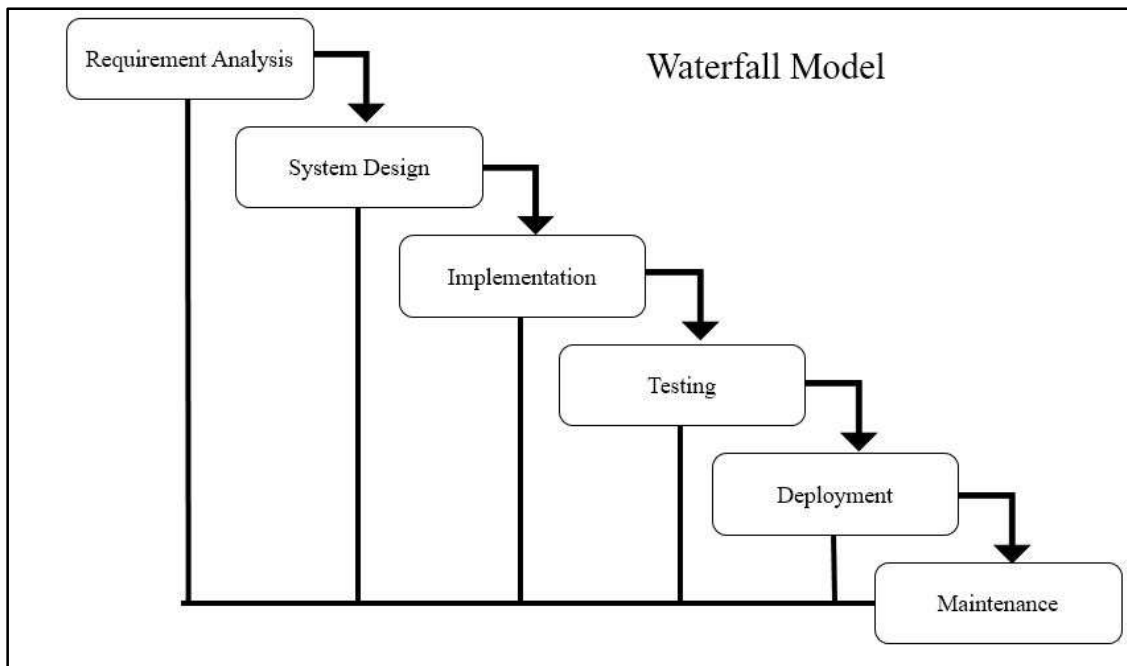


Figure 3. 1: Waterfall model

- **Requirement analysis:** The first phase involves understanding what you need to design and what is its function, purpose etc. All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
- **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
- **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
- **System Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **System Deployment:** Once the functional and non-functional testing is done.

- **System Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released.

3.2 ALTERNATE SOLUTION

In software development all the system basically divided into three categories that are standalone, network based and web based system.

Alternate solutions	Standard alone	Web based	Network based
Access form anywhere	X	√	Limited
Graphical user interface	√	√	√
Multiple users	X	√	√
Find overall management from one place	X	√	Limited
Sharing facilities	X	√	√

Table 3. 1: Alternate Solution

In this system the web based prechool management system for “Kids’ park” has branches in various places, they want to access the system in their places. They like to centralized the backend (database) and provide the system to all users in simultaneously. For these reasons based on above Table 3. 1 the institute has decided to develop the system in web based.

3.3 PROCESS DESIGN OF THE SYSTEM

The design gives the solution for requirements analysis, based on this design to develop the system. This design was divided into three stages:

- Database design
- Application architecture design
- Interface design

3.3.1 DATABASE DESIGN

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. This is an important phase in the

design process, which helps to design the system structure in a robust manner. By incorporating a transformation process is called normalization. It can then be converted to the physical database.

Normalization

In the field of relational database design, normalization is a systematic way of ensuring that a database structure is suitable for general-purpose querying and free of certain undesirable characteristics such as insertion, update, and deletion anomalies. That could lead to a loss of data integrity.

First Normal Form (1NF)

First Normal form sets the very basic rules for an organized database. These eliminate duplicative columns from the same table. This can be creating separate tables for each group of related data and identify each row with a unique column or set of columns (the primary key).

Second Normal Form (2NF)

Second normal form addresses the concept of removing up locative data. Meet all the requirements of the first normal form. Remove subsets of data that apply to multiple rows of a table and place them in separate tables. Create relationships between these new tables and their predecessors through the use of foreign keys.

Third Normal Form (3NF)

Third normal form goes one large step further meet all the requirements of the second normal form. Remove columns that are not dependent upon the primary key.

After third normal form most of the table mostly avoids the data redundancy, if any additional other normal form will be used. I drew the Entity Diagram by MySQL Workbench 6.0.8 CE. Our system ER diagram shown following Figure 3.2: ER Diagram:

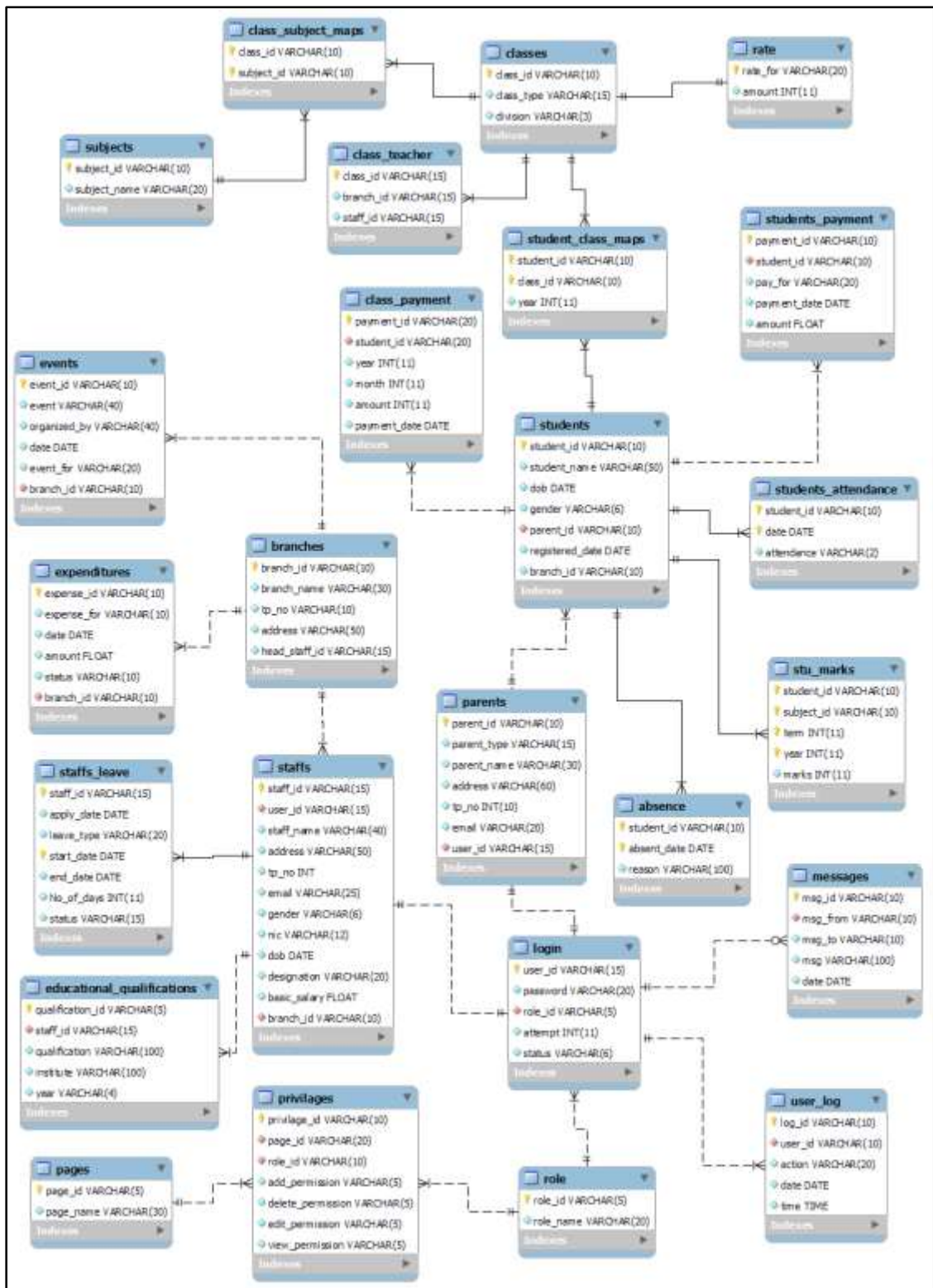


Figure 3. 2: ER Diagram

3.3.2 ARCHITECTURE DESIGN

In this design level, the basic structural framework that identifies the major components of a system and the communications between these components. Below is a general model for the whole system.

Use-case diagram:

Use-case diagram is the simplest that illustrates the user's interaction with the system. It portrays the different type of users in the system and various ways that they interact with the system. We drew use-case diagram for preschool management and student management. Use-case diagram for "Online Pre-school management system for Urumpirai Kids' Park" as follow Figure 3.3 and Figure 3.4

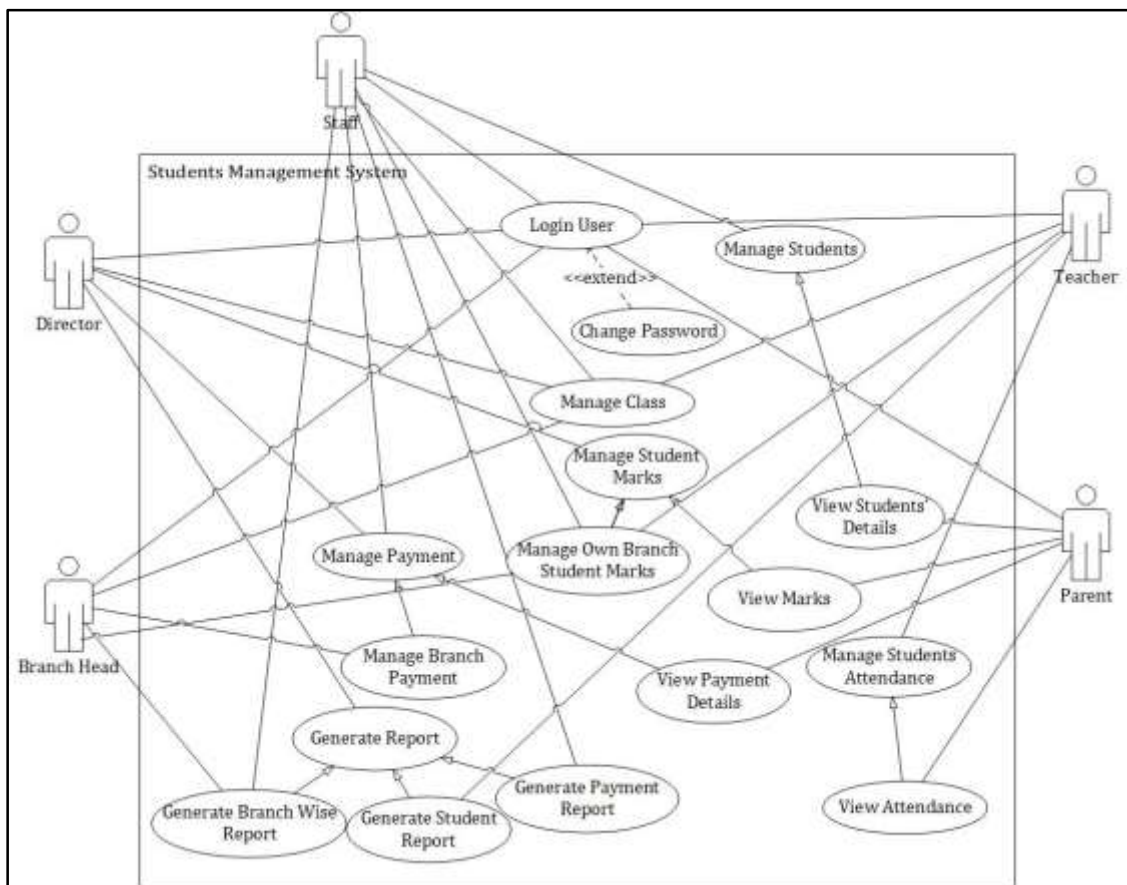


Figure 3. 3: Use case diagram for student management system

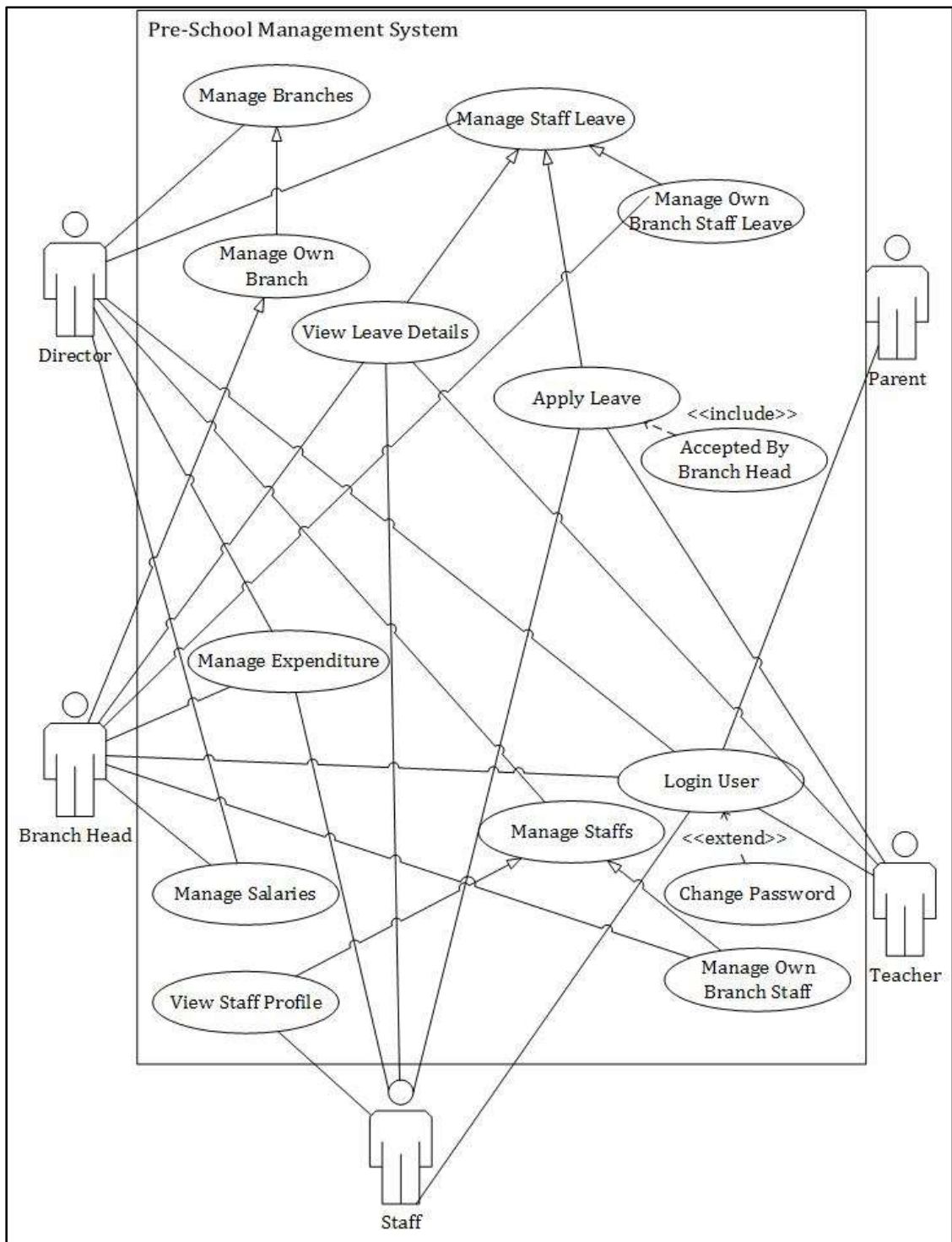


Figure 3. 4: Use case diagram for pre-school management

Use-case	Login User
Actor	Staff, Director, Branch Head, Parent, Teacher
Description	Only already registered users can access the system. Others cannot, if they want to access; want register to this system. Registered users when login; system validate the username and password; and authorized that is legal or illegal; if correct authorized user system allowed to access the system.
Use-case	Manage Class
Actor	Director, Staff, Teacher, Branch Head
Description	Manage the class details by director and branch head, others staff and teacher view the class details
Use-case	Manage Student Marks
Actor	Director, Staff, Teacher, Branch Head, Parent
Description	Manage the all the student marks details by director and branch head, others staff and teacher manage their branch student marks details, parents only view his/her children marks details
Use-case	Manage Leave Details
Actor	Director, Staff, Teacher, Branch Head
Description	Manage all the staff leave details by director and branch head manage his/her branch staff leave details, others staff and teacher view his/her leave details and apply for the leave

Table 3. 2: Use case Narrative

Activity diagram:

This diagram represents the graphical view of workflows of stepwise activities in the system. We drew activity diagram for user login. Refer Figure 3.5.

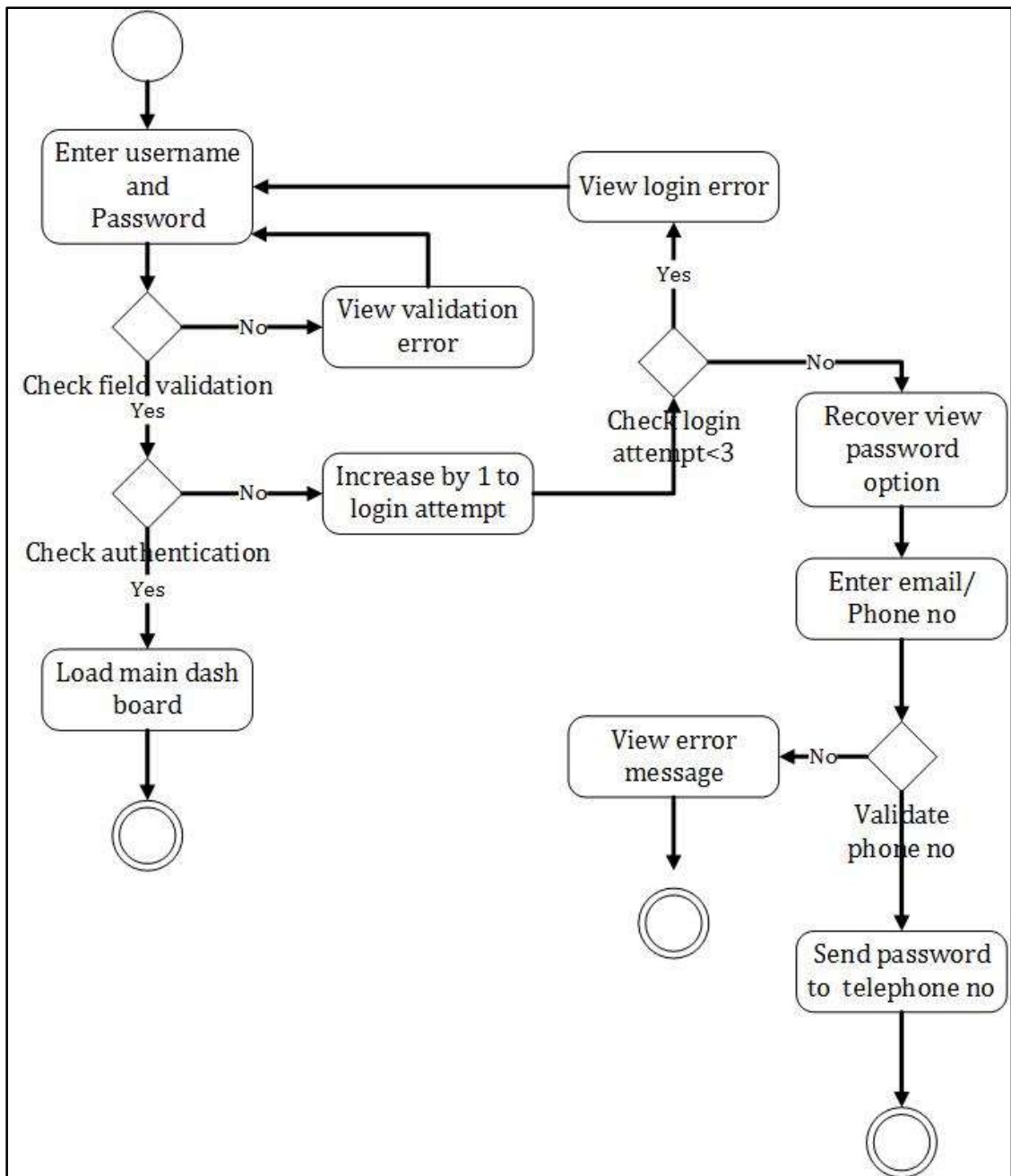


Figure 3. 5: Activity diagram for user login

Sequence diagram:

This diagram represents how objects interact in given situation or activity. I drew sequence diagram for student attendance. Refer Figure 3.6.

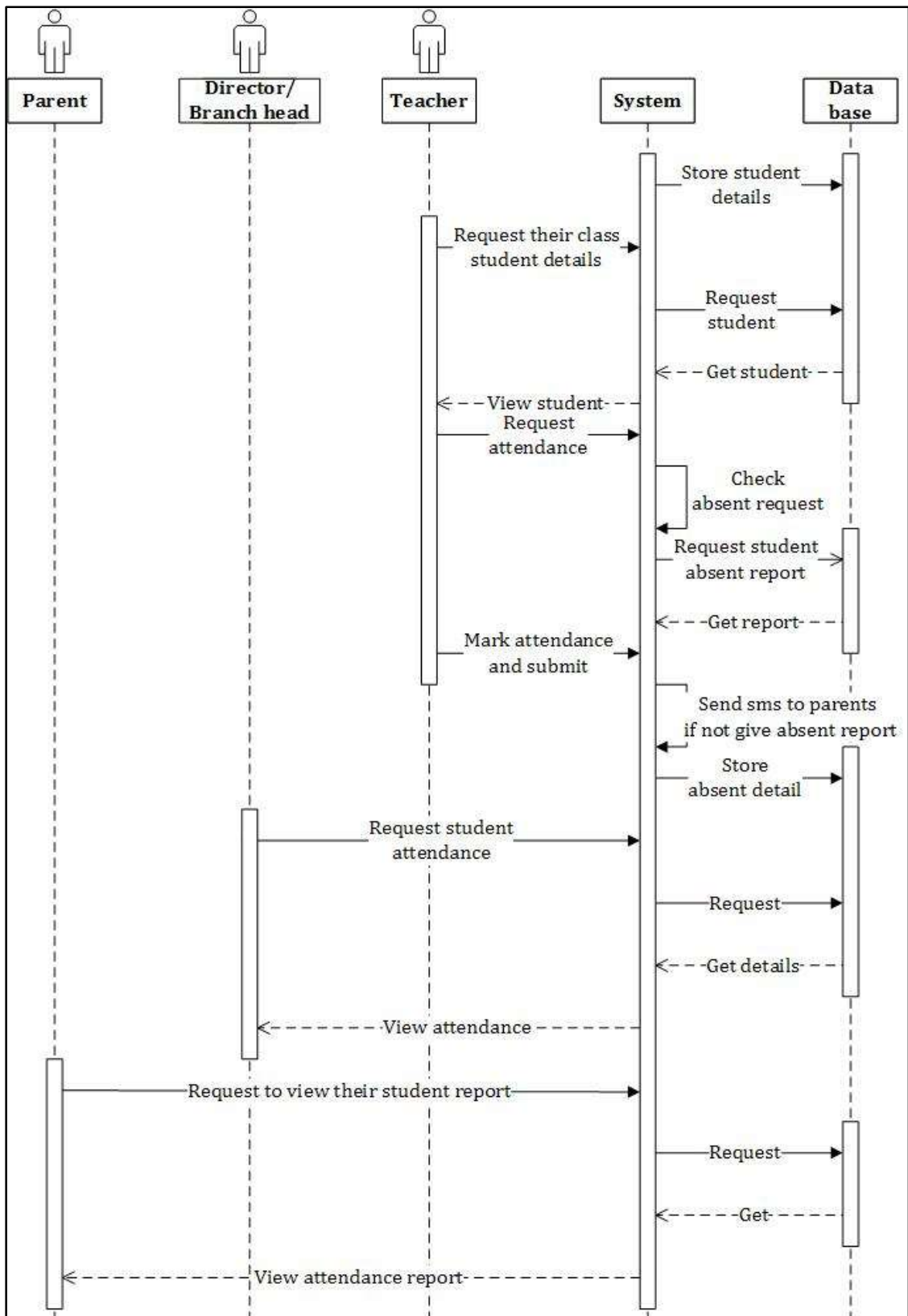


Figure 3. 6: Sequence diagram for student attendance

Date Flow Diagram:

A date flow diagram (DFD) is a graphical representation of the “flow” of data through An information system, modelling its process aspects.

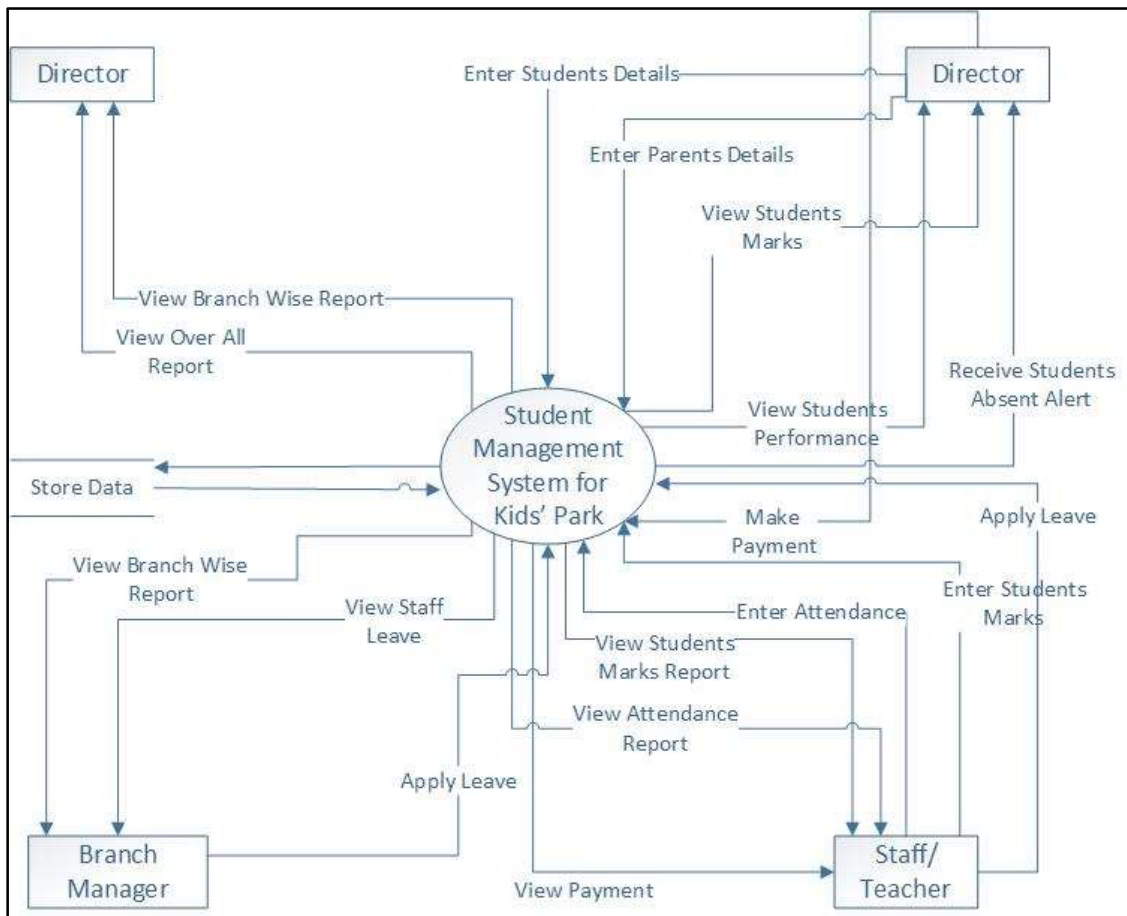


Figure 3.7 Data Flow Diagram for Pre School Management System Level 0

3.3.3 INTERFACE DESIGN

Interface design is the design of the user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals [4].

Main User Interfaces

This section makes available for using interfaces of the online pre-school management for Urumpirai kid's park to provide the structure and design of the system. Please refer Appendix C – User Documentation for the rest of the interface designs.

Home Page

The default homepage makes adequate to see about us, events in this institute, gallery, contact details and login to the system. This home page will be changed after a user logged in to the system according to the user type. Refer following Figure 3.8: Home Page.



Figure 3. 8: Home Page

Login Page

This page allows users who are properly registered in this system to login and also this page gives authority for a password if the users find difficult by forgetting their password. Refer Figure 3.9: Login Page.

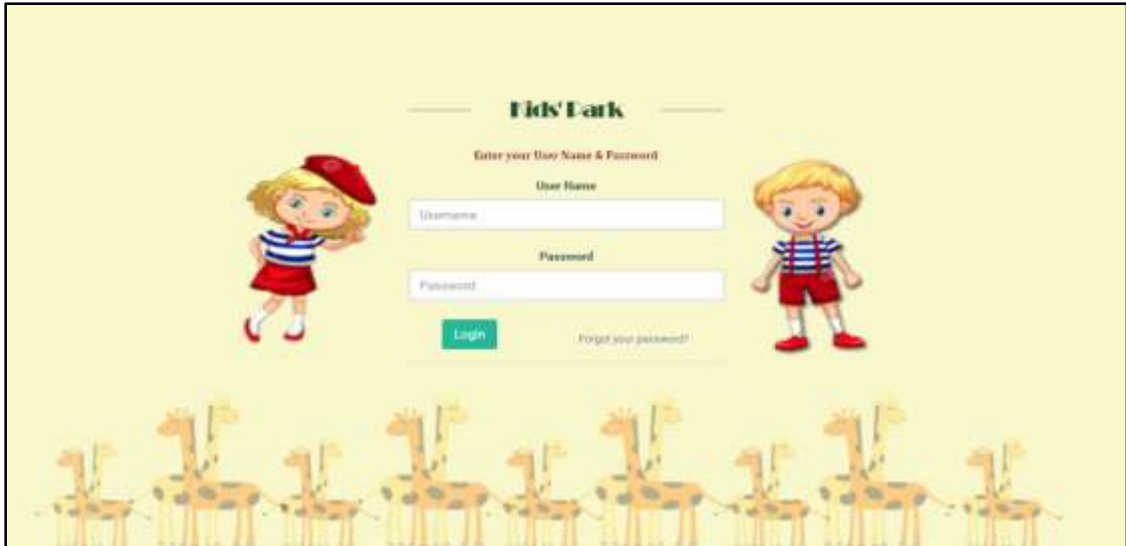


Figure 3. 9: Login Page

Registration Form

Student Registration form of online pre-school management system includes the students personal details. Student id automatically generate in the text box. Refer in Figure 3.10: Student Registration Form

Add New Student

Student ID	<input type="text" value="STU0012"/>	
Parent ID	<input type="text" value="type here"/>	<input type="button" value="» Select"/>
Student Name	<input type="text" value="type here"/>	
Date Of Birth	<input type="text" value="yyyy-mm-dd"/>	
Gender	<input type="button" value="Male"/> <input checked="" type="button" value="Female"/>	
Registered Date	<input type="text" value="yyyy-mm-dd"/>	
Branch	<input type="text" value="Thinnaveli"/>	

Figure 3. 10: Student Registration Form

Report Generator

This report contains student's detail in class wise. When user select the proper option student's report view on this page. Administrator (Director), branch head and staff are able to print this report. Refer Figure 3.11: Report of students' details – class wise.

Report of Students' Details - Class wise

Branch: ▼
 Class: ▼
 Year: ▼

Report of Students' Details - Class wise

Description	
Branch	Urumpirai
Class Type	4Plus
Year	2017

Student ID	Student Name	DOB	Gender	Parent ID	Registered Date	Branch ID
STU0003	Akash	2012-07-13	male	P0004	2017-01-04	B02
STU0008	Sajeesan	2012-08-08	male	P0005	2017-01-03	B02

[Print](#)

Figure 3. 71: Report of students' details – class wise.

After click on the print button that report will be as follow. Refer Figure 3.12.



Urumpirai Kids' Park
 Model Pre-School
 Do Right Be Bright
 A nonprofit making service organization

TP NO: 021 223 0754 / 077 656 5258 GANAVAJRAVAR SWAMY KOVIL ROAD, URUMPIRAI

Report of Students' Details - Class wise

Description	
Branch	Urumpirai
Class Type	4Plus
Year	2017

Student ID	Student Name	DOB	Gender	Parent ID	Registered Date	Branch ID
STU0003	Akash	2012-07-13	male	P0004	2017-01-04	B02
STU0008	Sajeesan	2012-08-08	male	P0005	2017-01-03	B02

[Print](#)

Figure 3. 82: Print Report

CHAPTER 4: IMPLEMENTATION

An implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programming and deployment [5]. In this chapter implementation we describe what activities were carried out during the development of our online pre-school management system. After design stage, the result of the design stage is transformed into physical design and we implemented that result of design stage.

For the development and coding PHP and MySQL are basic fundamentals. Some tools such as Notepad ++, Bootstraps are in operation for a development. All coding were designed to produce the system to be most functional in the future.

4.1 IMPLEMENTATION ENVIRONMENT

Hardware	Software
Pentium 4 computer	Wamp/XAMPP Server
Basic printer for printing	Web browser
	Operating System

Table 4. 1: Implement Requirement

Online pre-school management system was developed on a Windows 10 Home Premium computer and to usually compatible with the Windows 8, Windows 7, Windows Vista, and Linux systems such as Ubuntu, Fedora and Red Hat. (Furthermore this system is mostly compatible with any Operating System only tested Operating Systems are mentioned here).

This system was tested under the following screen resolutions on various computers.

- 1366 x 768 (pixels)
- 1280 x 768 (pixels)
- 1024 x 768 (pixels)

Development Tools used for this system development

- Windows operating system
- WampServer Version 2.4 for Windows
- Apache Web Server Version 2.4.4
- PHP Script Language Version 5.4.16
- MySQL Database Version 5.6.12
- phpMyAdmin Database Manager Version 4.0.4
- MySQL Workbench 6.0.8 CE
- Microsoft Visio 2013
- Adobe Photoshop Version 14.0
- Internet Explore/Google Chrome/Mozilla Firefox

Technologies used for development

- MySQL was designed to deal all the development with regard to the database.
- PHP was the main development language used to develop the main system and its logics.
- HTML 5 was designed to create the base Interfaces of the system.
- JavaScript was used to code all the client-side validation.
- JQuery which is also based on JavaScript was used to implement the pre-coded time picker module, Date Picker module, and password validate.
- CSS was designed to have the plain HTML 5 interfaces with more attraction and user friendly, which also decided easy to access the system.

4.2 REUSED MODULES AND COMPONENTS

The following pre-coded modules were used while developing the system. **Gentelella - Bootstrap Admin Template [6]** is a cascading style sheet developed by Colorlib which is used for designing of tables and forms. This template is free can download from web. This template is only used for user interface design.

4.3 NETWORK IMPLEMENTATION

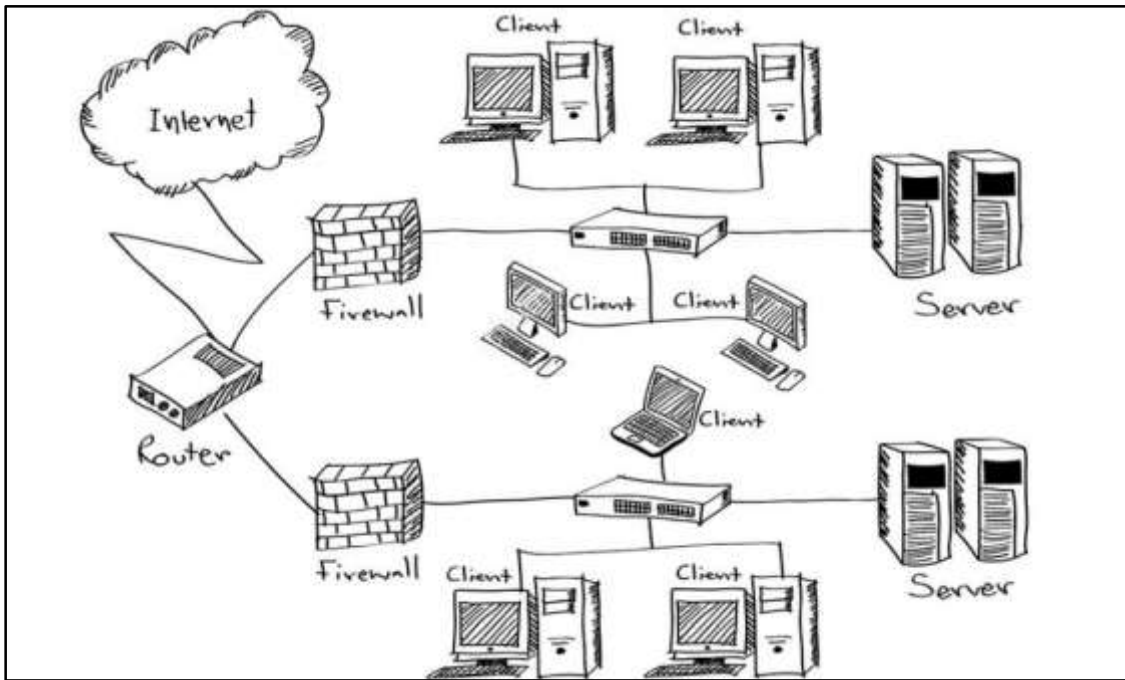


Figure 4. 1: Network implementation

Web applications run on the Web Server. This web based system is accessed through web clients. It could be installed on a web server and a dedicated database server is used to control the system's database. The above Figure 4.1 shows a Dedicated Web server and a Database Server are installed in the institute but public servers also can be used to implement the system. It will reduce much initial cost and maintenance host. Director, branch head, staff, teachers and parents are able to use the system through the Internet connection. To carryout users requirements they need Personnel computer or Laptop with Internet connection without any interruption.

4.4 CODE AND MODULE STRUCTURE

Online pre-school management system has many sub categories. There are student, branch, staff, login and payment. Each category contain following items. (Figure 4.2)

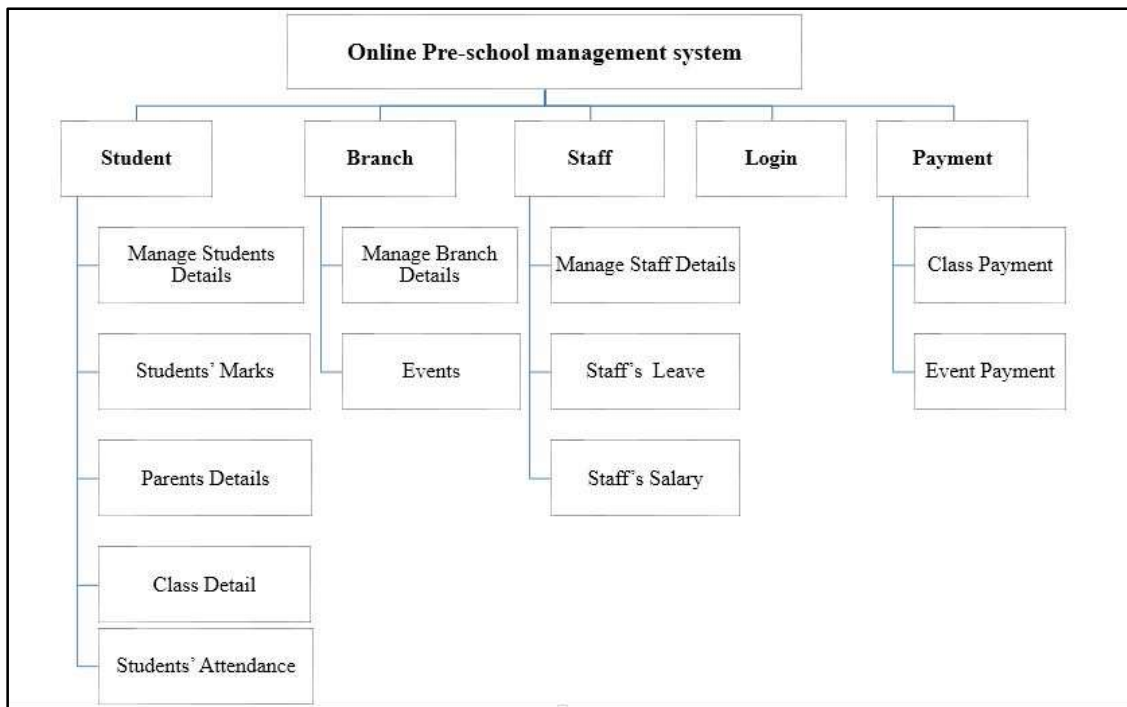


Figure 4. 2: Module Structure

Student management: This system considers about manage students' details, students' marks, parents' details, class details' and students' attendance.

Branch management: This system considers about manage branch details and event.

Staff management: This system considers about manage staff's details, staff's leave and staff's salary.

Login management: This system considers about user login only.

Payment management: This system considers about class payment and event payment.

WAMP Server

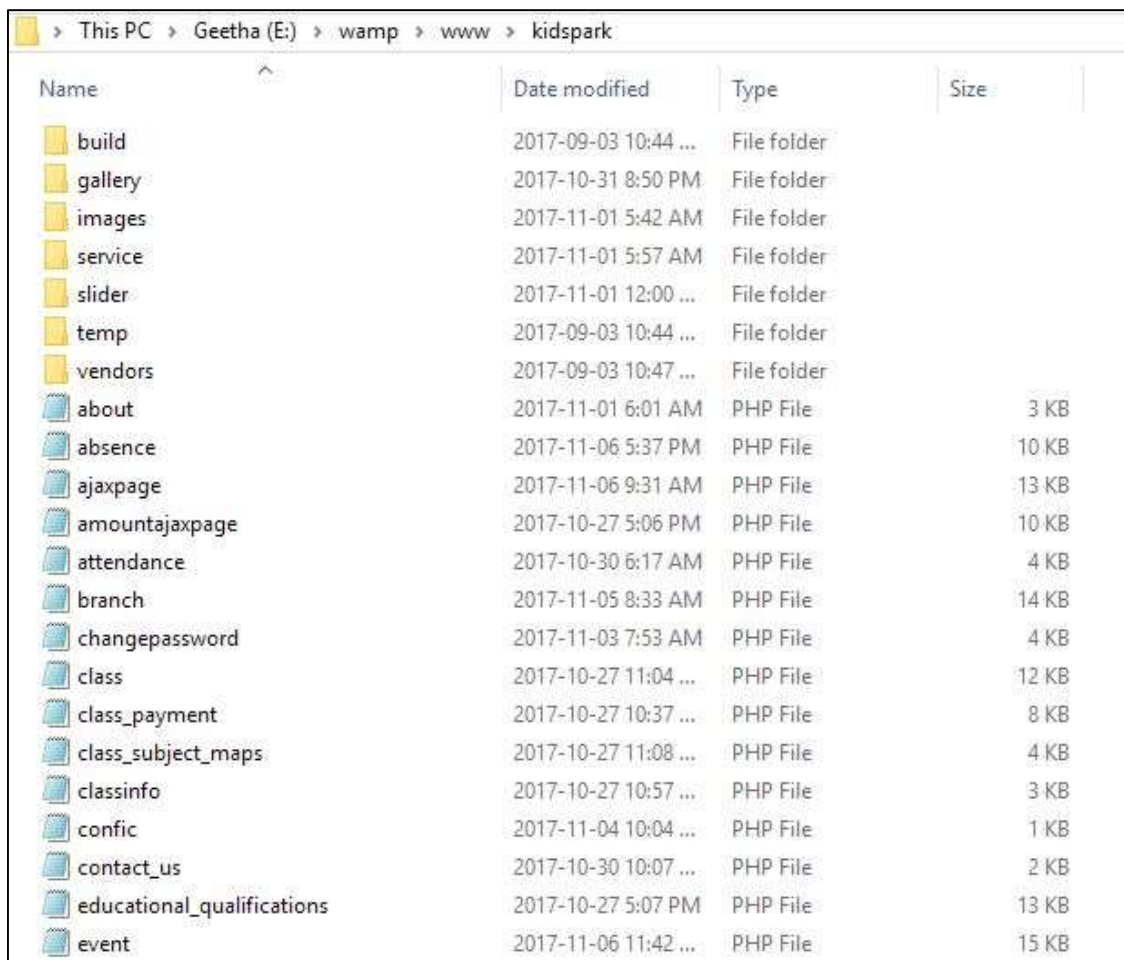
WAMP is an acronym for Window, Apache, MySQL, and PHP. This is a software bundle consisting of the before mentioned four software. WAMP server is solid and stable high-performance platform for windows

Gentelella - Bootstrap Admin Template is a cascading style sheet developed by Colorlib used for designing of tables and buttons.

Other Software

Notepad++ was used in designing the website as well as scripting, while phpmyadmin was used as the interface for interacting with MySQL. MySQL was used to create database. Adobe Photoshop was used to create banner and image editing.

The directory of this system is arranged like on Figure 4.3:



Name	Date modified	Type	Size
build	2017-09-03 10:44 ...	File folder	
gallery	2017-10-31 8:50 PM	File folder	
images	2017-11-01 5:42 AM	File folder	
service	2017-11-01 5:57 AM	File folder	
slider	2017-11-01 12:00 ...	File folder	
temp	2017-09-03 10:44 ...	File folder	
vendors	2017-09-03 10:47 ...	File folder	
about	2017-11-01 6:01 AM	PHP File	3 KB
absence	2017-11-06 5:37 PM	PHP File	10 KB
ajaxpage	2017-11-06 9:31 AM	PHP File	13 KB
amountajaxpage	2017-10-27 5:06 PM	PHP File	10 KB
attendance	2017-10-30 6:17 AM	PHP File	4 KB
branch	2017-11-05 8:33 AM	PHP File	14 KB
changepassword	2017-11-03 7:53 AM	PHP File	4 KB
class	2017-10-27 11:04 ...	PHP File	12 KB
class_payment	2017-10-27 10:37 ...	PHP File	8 KB
class_subject_maps	2017-10-27 11:08 ...	PHP File	4 KB
classinfo	2017-10-27 10:57 ...	PHP File	3 KB
confic	2017-11-04 10:04 ...	PHP File	1 KB
contact_us	2017-10-30 10:07 ...	PHP File	2 KB
educational_qualifications	2017-10-27 5:07 PM	PHP File	13 KB
event	2017-11-06 11:42 ...	PHP File	15 KB

Figure 4. 3: Directory

All core components separated from the style sheets, images and scripts. The Directories are arranged to the appropriate files according their file extension as Figure 4.3.

The functionalities of the index.php, menu.php plus the other pages are explained below Files are separated from images, programming files, CSS, JS. To make the system easily managed by anyone in the future CSS, JS, other tool files are located into the folder named as build.

The important functionalities of the confic.php, login.php, Add form and some other pages are explained below.

Database connectivity

When developing the system, there is a need for database to store the data. Every time when run SQL coding, want to connect database so write a database connection coding in single page (confic.php) and call (include('confic.php')) the connection page when run the

SQL coding.Used the “mysql_connect()” code to connect the local host. After successful of the connection of local host we want to connect the database. We used the “mysql_select_db()” code for connect the database.

Database connection code:

```
<?php
$hostname='localhost'; // host name
$username='root'; // user name
$password=''; // password
$dbname='kidspark_db'; // db name
$con=mysql_connect($hostname,$username,$password);
if(!$con)
{
    echo "mysql could not connected".mysql_error();
}
$select=mysql_select_db($dbname);
if(!$select)
{
    echo "database could not connected".mysql_error();
}
?>
```

Login to system

This login has to used authenticate the users. When we enter username and password correctly system displays their user interface. If we enter username or password wrong the system will provide a message box with meaningful message. If we enter wrong username or password more than three then systems automatically go to forget password page.

```
$msg="";
if(isset($_POST['btnlogin'])){
    $user_id=$_POST['user_id'];
    $password=$_POST['password'];
    $sql1 = "SELECT *
    FROM `login`
```

```

WHERE user_id='$user_id';

$result1=mysql_query($sql1)or die("error in login:".mysql_error());

if(mysql_num_rows($result1)==1){

$row1=mysql_fetch_assoc($result1);

$sql2 ="SELECT *

FROM `login`

WHERE user_id='$user_id'

AND password='$password'";

$result2=mysql_query($sql2)or die("error in login:".mysql_error());

if(mysql_num_rows($result2)==1){

$_SESSION['user_id']=$user_id;

$row2=mysql_fetch_assoc($result2);

$role_id=$row2['role_id'];

$usertypesql = "SELECT branch_id

FROM `staffs`

WHERE user_id='$user_id'";

$usertypesresult=mysql_query($usertypesql)or die("error in login:".mysql_error());

$row3=mysql_fetch_assoc($usertypesresult);

$branch_id=$row3['branch_id'];

$_SESSION['role_id']=$role_id;

$_SESSION['branch_id']=$branch_id;

$sql3="UPDATE login SET attempt=0 WHERE user_id='$user_id'";

$result3=mysql_query($sql3)or die("error in login1:".mysql_error());

header('location:index.php');

}

elseif($row1['attempt']<3){

$msgg="your password is incorrect, please try again";

$sql4 ="UPDATE login

SET attempt=attempt+1

WHERE user_id='$user_id'";

$result4=mysql_query($sql4)or die("error in login2:".mysql_error());

```

```

    }
    else{
        header('location:forgetpassword.php');
    }
}
else{
    $msg="your userid is not registered in the db";
}
}

```

New register code

If the user enters all correct information the system insert the details into staff table and student table. If we enter wrong the system provide an alert message box with meaningful message.

New register code:

```

$sqlinsertstudent="INSERT INTO `students`(`student_id`, `student_name`, `dob`, `gender`,
`parent_id`, `registered_date`, `branch_id`)
VALUES ('".mysql_real_escape_string($_POST['student_id'])."',
'".mysql_real_escape_string($_POST['student_name'])."',
'".mysql_real_escape_string($_POST['dob'])."',
'".mysql_real_escape_string($_POST['gender'])."',
'".mysql_real_escape_string($_POST['parent_id'])."',
'".mysql_real_escape_string($_POST['registered_date'])."',
'".mysql_real_escape_string($_POST['branch_id'])."'
)";

$resultinsertstudent = mysql_query($sqlinsertstudent)or die("error in insert student
part:".mysql_error());

$student_id=$_POST['student_id'];
$action="Addnewstudent:".$student_id;

include('userlog.php');

if($resultinsertstudent)
{
    echo '<script> alert("your data added successfully");</script>'; }

```

Validation code

Code segments given below validate user inputs using JavaScript technology and handles user input errors before it is transmitted to the server.

```
//nic validation
```

```
function checknicno(){
```

```
    var nicno=document.getElementById("nic").value;
```

```
    if(nicno.length==10)
```

```
    {
```

```
        var nicformat1=/^[0-9]{9}[a-zA-Z0-9]{1}$/;
```

```
        if(nicno.match(nicformat1)){
```

```
            var nicformat2=/^[0-9]{9}[VvXx]{1}$/;
```

```
            if(nicno.match(nicformat2)){
```

```
                document.getElementById("nicerrormsg").innerHTML="";
```

```
            }
```

```
        else{
```

```
            document.getElementById("nicerrormsg").innerHTML="last charactor must be V or X";
```

```
            document.getElementById("nic").focus();
```

```
        }
```

```
    }
```

```
    else{
```

```
        document.getElementById("nicerrormsg").innerHTML="first 9 charactor must be numbers";
```

```
        document.getElementById("nic").focus();
```

```
    }
```

```
}
```

```
else if (nicno.length==12){
```

```
    var nicformat=/^[0-9]{12}$/;
```

```
    if(nicno.match(nicformat)){
```

```
        document.getElementById("nicerrormsg").innerHTML="";
```

```
    }
```

```
    else{
```



```

        document.getElementById("nicerrormsg").innerHTML="All charactor must be
numbers";
        document.getElementById("nic").focus();
    }
}
else if (nicno.length==0){
    var nicformat=/^[0-9]{12}$/;
    if(nicno.match(nicformat)){
        document.getElementById("nicerrormsg").innerHTML="";
    }
}
else {
    document.getElementById("nicerrormsg").innerHTML="your NIC no must be 10 or 12
charactor";
    document.getElementById("nic").focus();
}
}
// end nic validation

```

4.5 SECURITY

This online web based preschool management system is accessed by parents and Urumpirai kids' park manager level users from various places and transform many data, so it is important to maintain the security. The administration (director) has full privilege to access the system, in other hand other users have less privilege than director. From our system registered user only access the system, the registered user uses their user id and password to login and access the system. After register in our system the new user can access the system.

If user try to login with wrong password, our system allow only three times. More than three times system automatically load the forget password page. From forget password web page the system verify the user id and registered hand phone number then system send a password to user's registered phone number.

CHAPTER 5: EVALUATION

The evaluation process carried out entire software development life cycle to achieve the user friendly system with satisfaction of user requirements. From this phase the all aspect of online based preschool management system was tested with sample data and validates all.

The testing is very important to identify and detect error, check if the system is working properly or not, validating the system and verification all activities of the software implement correctly or not.

5.1 TESTING PROCEDURE

Testing procedures are the Testing is the process of evaluating a system or its component(s) with the intention to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

This involves testing the system using different types of system tests that were performed on these system. This is aimed at uncovering errors and measuring the system capability. The following system tests are do in this system.

5.1.1 UNIT TESTING

A unit is the smallest testable part of an application like functions, classes, procedures, interfaces. Unit testing is a method by which individual units of source code are tested to determine if they are fit for use. The goal of unit testing is to segregate each part of the program and test that the individual parts are working correctly.

5.1.2 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing. The objective is to take unit-tested module and build a program structure that has been dictated by design.

5.1.3 SYSTEM TESTING

System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing. For example, login interface, register details and printing function etc.

5.1.4 ACCEPTANCE TESTING

The software has been tested with the realistic data given by the client and produced fruitful results. The client satisfying all the requirements specified by them has also developed the software within the time limitation specified. A demonstration has been given to the client and the end-user giving all the operational features.

5.2 TESTING PLANS FOR OUR SYSTEM

Testing was done each and every part of the system. The entire test plans were done before implementation on client side. From our test pan help to us identify the error and fixed it.

Our test modules were designed as follows:

- New user register module (Table 5. 1)
- Administrator module (Director) (Table 5. 2)
- Branch Head module (Table 5. 3)
- Staff module (Table 5. 4)
- Parent module (Table 5. 5)
- Common function module (Table 5. 6)

5.2.1 NEW USER REGISTER MODULE (WITHOUT LOGIN)

Test No	Test Description	Steps to test	Expected Result	Status
1	Login without register	Cannot access the system	Did not access the system	Pass
2	Register new staff or parent	System allow only phone number and send verification code to phone number to verify.	System allowed the phone number only and send a verification code to phone number and after verification allow to access	Pass

Table 5. 1: New User Module

5.2.2 ADMINISTRATION (DIRECTOR) MODULE

Test No	Test Description	Steps to test	Expected Result	Status
1	Add new staff	Complete all the fields correctly and click save button.	Display popup message “Your data added successfully”.	Pass
2	Add new student	Complete all the fields correctly and click save button.	Display popup message “Your data added successfully”.	Pass
3	Add new branch	Complete all the fields correctly and click save button.	Display popup message “Your data added successfully”.	Pass
4	Add new class	Complete all the fields correctly and click save button.	Display popup message “Your data added successfully”.	Pass
5	Generate salary report	Select appropriate options and click print.	Popup open with relevant data in print preview.	Pass
6	Generate students’ attendance report	Select the appropriate	Popup open with relevant	Pass

		option (branch, class, from date and to date) and click print.	data in print preview.	
7	Generate students' payment report	Select the appropriate option (branch, event and class) and click print.	Popup open with relevant data in print preview.	Pass
8	Message	Complete all the fields correctly and click send button.	Message send to relevant person	Pass
9	Leave Confirmation	Click confirm/ reject button in the status field.	Data view on staff's leave page	Pass

Table 5. 2: Director Module

5.2.3 BRANCH HEAD MODULE

Test No	Test Description	Steps to test	Expected Result	Status
1	Add new staff	Complete all the fields correctly and click save button.	Display popup message "Your data added successfully".	Pass

2	Add new student	Complete all the fields correctly and click save button.	Display popup message “Your data added successfully”.	Pass
3	Add new class	Complete all the fields correctly and click save button.	Display popup message “Your data added successfully”.	Pass
4	Generate students’ attendance report	Select the appropriate option (branch, event and class) and click print.	Print page open with relevant data in print preview.	Pass
5	Generate students’ payment report	Select the appropriate option (branch, event and class) and click print.	Popup open with relevant data in print preview.	Pass
6	Message	Complete all the fields correctly and click send button.	Message send to relevant person	Pass
7	Apply Leave	Apply for the leave.	Conformation send from director.	Pass

Table 5. 3: Branch Head Module

5.2.4 STAFF MODULE

Test No	Test Description	Steps to test	Expected Result	Status
1	Add new event	Complete all the fields correctly and click save button.	Display popup message "Your data added successfully".	Pass
2	Add new parent	Complete all the fields correctly and click save button.	Display popup message "Your data added successfully".	Pass
3	Add new expenditure	Complete all the fields correctly and click save button.	Display popup message "Your data added successfully".	Pass
4	Add new expenditure	Complete all the fields correctly and click save button.	Display popup message "Your data added successfully".	Pass
5	Add new subject	Complete all the fields correctly and click save	Display popup message "Your data	Pass

		button.	added successfully”.	
6	View staff details	Select the view button.	Staff can be able to view staff details.	Pass
7	View branch details	Select the view button.	Staff can be able to view branch details.	Pass
8	Message	Complete all the fields correctly and click send button.	Message send to relevant person	Pass
9	Apply Leave	Apply for the leave.	Conformation send from director.	Pass
10	Profile	Click the profile menu top navigation menu.	Staff can be able to view their profile and able to edit / print profile	Pass

Table 5. 4: Staff Module

5.2.5 PARENT MODULE

Test No	Test Description	Steps to test	Expected Result	Status
1	View their kid's (student) details	Select the student's details in the menu.	Parent can be able to view their kid's details.	Pass
2	View their kid's	Select the	Parent can be	Pass

	(student) attendance	student's attendance in the menu and select the appropriate option.	able to view their kid's attendance.	
3	Add new absence	Complete all the fields correctly and click save button.	Data saved in the database	Pass
4	View their kid's (student) marks	Select the student's marks in the menu and select the appropriate option.	Parent can be able to view their kid's attendance marks.	Pass
5	View event details	Select the events details in the menu.	Parent can be able to view event detail.	Pass
6	Profile	Click the profile menu top navigation menu.	Parent can be able to view their profile and able to edit / print profile	Pass

Table 5. 5: Parent Module

5.2.6 COMMON FUNCTION MODULE

Test No	Test Description	Steps to test	Expected Result	Status
1	Login testing	Enter wrong	Display error	Pass

	(negative)	username or password.	message as follows (your user id is not registered in the db).	
2	Login testing (positive)	Enter a valid registered username and password	Redirect to appropriate page based on user id.	Pass
3	Forget Password	Click forgot password on the login page.	Ask phone number for verification if provided correctly and send password to registered email.	Pass
4	Logout	Click logout link.	Logout user and direct to index page (home page).	Pass
5	Test browser support	Access and load system into various browsers (Internet Explorer, Firefox, Google Chrome).	The system should act and display well-formed in all browsers.	Pass
6	Test Database	Import	System should	Pass

	connections	database into various system and change the values in database connection coding.	able to connect to database.	
7	Form validation	Enter wrong parameters for the field	Display error message before submit the form using JavaScript.	Pass
8	Test OS compatibility	Access the system from various Windows /Linux environments	System should be able to work finely in all OSs	Pass

Table 5. 6: Common Module

5.3 TEST DATA AND TEST RESULTS

Sample data and some past record from the pre-school were entered into the system to all modules which need to be tested in order to ensure the correctness. Each data in the all forms were tested to find out whether the system will provide the appropriate error/success message according to the data is entered before submitting the form to avoid in-appropriate data to be entered into the database

Users Acceptance Chart

The system was tested by different types of users and got feedback from them. The user evaluation chart is shown in the following Figure 5.1:

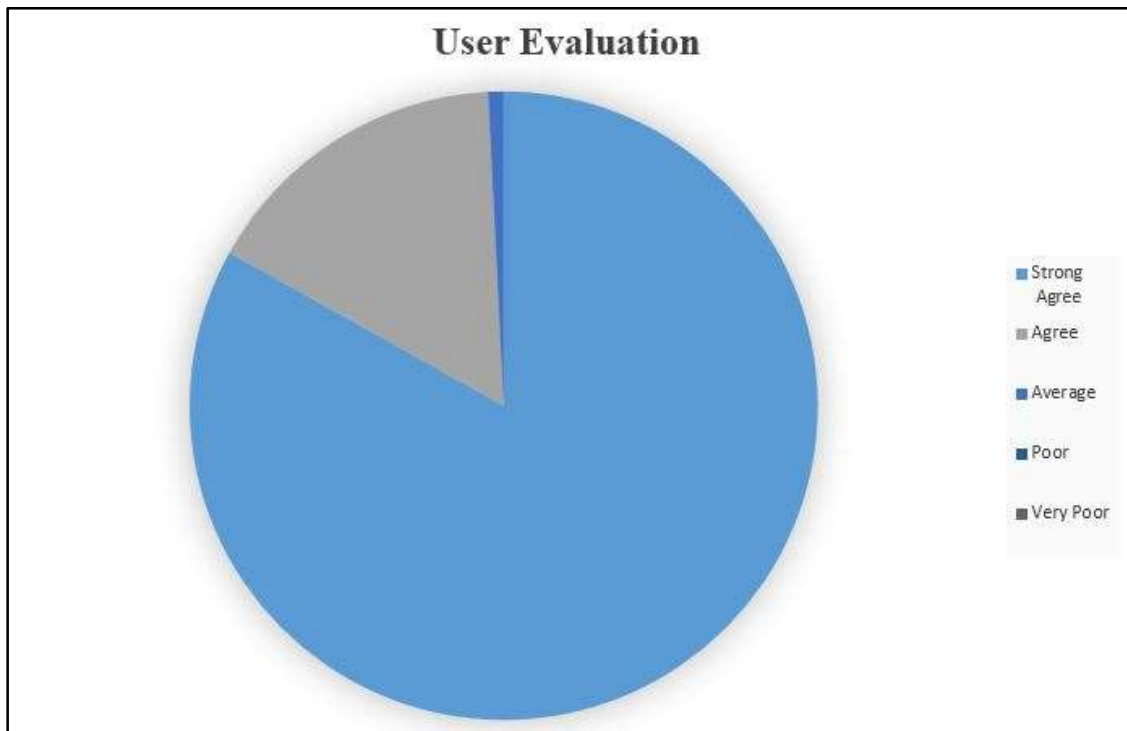



Figure 5. 1: User Acceptance chart

CLIENT ACCEPTANCE REPORT

The client acceptance test result is shown in the following Figure 5.2:



Urumpirai Kids' Park

Model Pre-school
Do Right Be Bright
உரும்பிராய் கிட்ஸ் பார்க்
A nonprofit making service organization

GNANAVIRAVA SWAMY KOVIL ROAD,
URUMPIRAI

...../...../201.....

Admin Evaluation Report

Test Case	Test Case	Strong Agree	Agree	Average	Poor	Very Poor
1	New User Register	✓				
2	Administrator Module (Director)	✓				
3	Branch Head Module	✓				
4	Staff Module	✓				
5	Parent Module	✓				
6	Common Function Module	✓				
7	User Friendly of System	✓				
8	Clear of error messages, validation and verification code		✓			
9	Access the system		✓			
10	Overall perform of the system	✓				

How do you feel about the system?

Evaluated By: *Panchalingam*

Signature: *[Signature]*

Date: *30/10/2017*

URUMPIRAI KIDS' PARK
URUMPIRAI

President : A. Panchalingam 021- 223-0754 / 077 656 5258
Secretary : M. Sriharan 077 711 0186
Treasurer : (Ms) M.Selvakumar 077-9906271

Figure 5. 2: Client Acceptance Report

USERS ACCEPTANCE RESULT

Evaluation report about my system are obtain from the director of the Urumpirai kids' park which indicates point Figure 5. 3:

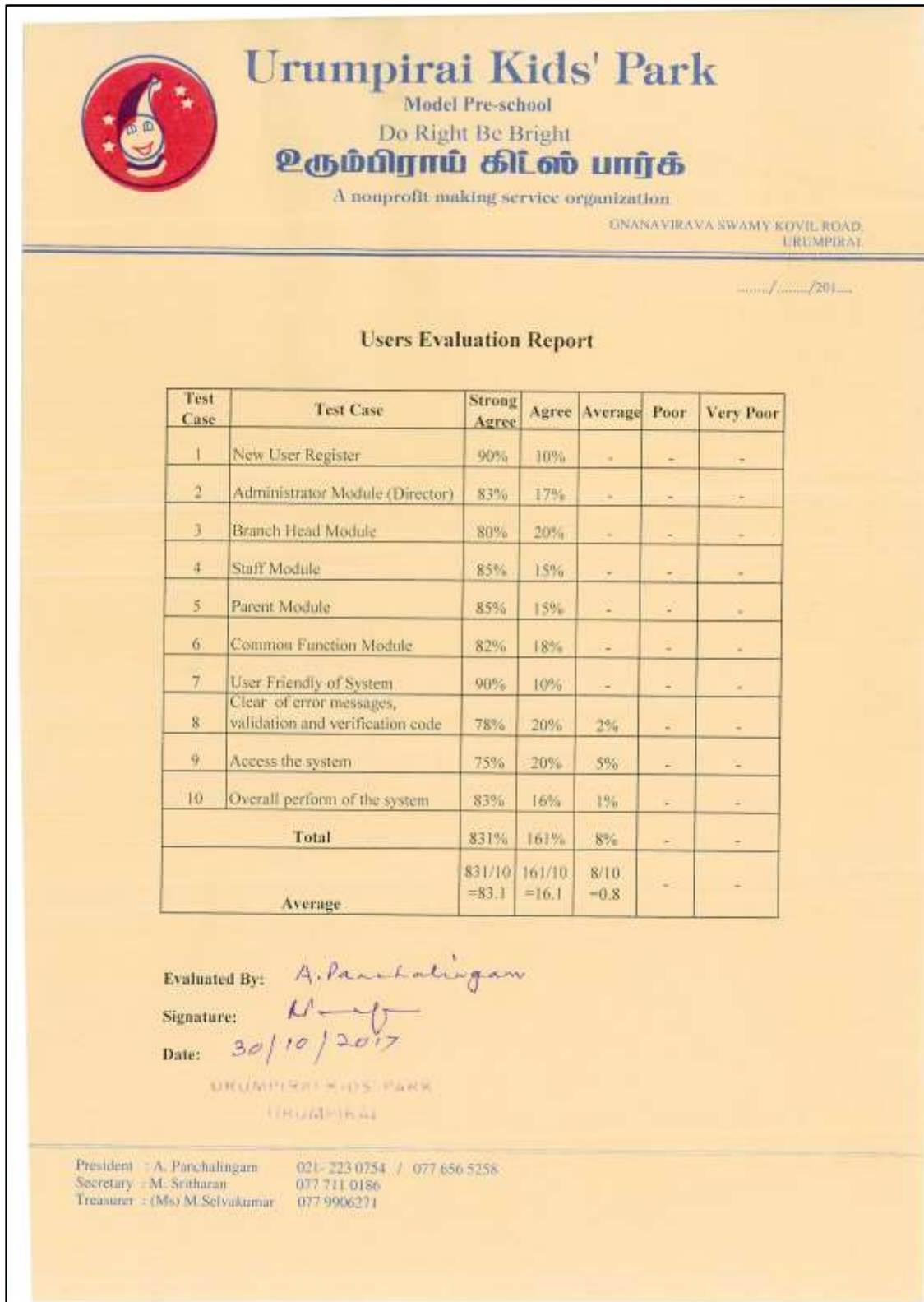


Figure 5. 3: User Acceptance Report

CHAPTER 6: CONCLUSION

This project was successfully completed within the time period. The project has been developed in php. All the modules are tested separately via testing procedures and put together to form the main kidspark system. Finally the system is tested with real data and everything worked successfully. Thus the system has fulfilled the entire objects identified.

6.1 CONCLUSION

From the start of the institute they keeps their records manually and faces many difficulties when using manual system. The details of the students are recorded when students are admitted to the preschool in record books and kept. Staff's details are gathered during the appointment to the preschool, the details of the staff are recorded in record books and kept. With that, every time the details are updated. This online pre-school management system was developed to address those drawbacks as well as to optimize their overall process and through this to help them to keep their institute name at the top level. This system developed based on PHP, HTML, CSS and JavaScript with back end MySQL.

At the end, system successfully satisfied the user requirements including functional and non-functional requirements. User with minimum knowledge about computers can also operate the system easily. By automating some processes, the workload of the user was reduced and it helped in saving the time of the client efficiently. Moreover, this automated system helps in reducing common mistakes that might occur by the user. Getting user feedback and properly testing the system helped to validate the overall system and to avoid conflicts that can occur.

6.2 LESSION LEARNT

This online pre-school management system for Urumpirai kids' park provide good knowledge to us and helps to forward our carrier life in easy way. The project give practical knowledge of software development and also helped to gain the experience of how to apply the theories that had learnt in previous semester of Bachelor of Information Technology (BIT) degree in practical wise.

In the start of the project proposal I did not have sufficient knowledge to carry out the project. I gained valuable knowledge in doing a successful professional system development project while following the detailed process and the guidelines provided by the university. The scheduled processes helped me in managing time efficiently. The implementation phase was the toughest and most interesting phase of the project, as it allowed me to apply the practical knowledge that I've gained on programming languages such as PHP, Java Scripts, CSS and many more development tools and techniques. Writing the dissertation was an exciting task of the project and it was a practice to my report writing. It helped me in developing my skills on writing and designing technical reports.

6.3 FUTURE WORK

This system fully satisfies the client requirements, but in future, some modules of this system can be replaced with the upcoming technology. Some of the future work is listed down here:

- ✓ Allow the parents to settle their kids' payment online.
- ✓ Settle the staff salary through bank.
- ✓ Add discussion forms and online exam facilities.
- ✓ Develop a mobile application integrated with this system.

REFERENCES

- [1] KinderCare, [online],
Available: <http://www.kindercare.com/programs-curriculum/programs-by-age/preschool>, [Accessed on: 13.07.2017]
- [2] Little Amigos, [online]
Available: <https://www.littleamigos.org/>, [Accessed on: 13.07.2017]
- [3] Rainbow preschool, [online]
Available: <http://www.rainbow-preschool.de/> , [Accessed on: 13.07.2017]
- [4] Wikipedia – Interface Design [online]
Available: https://en.wikipedia.org/wiki/User_interface_design, [Accessed on: 24.09.2017]
- [5] Wikipedia – Implementation [online]
Available: <https://en.wikipedia.org/wiki/Implementation>, [Accessed on: 24.09.2017]
- [6] Gentelella - Bootstrap Admin Template [online]
Available: <https://colorlib.com/polygon/gentelella/> , [Accessed on: 24.06.2017]

APPENDIX

APPENDIX A - SYSTEM DOCUMENTATION

We give instruction how to install our system and database, and it is help to client to installation and maintenance.

Step 1: Double click on download file WampServer 2.4 and just follow the instruction. Everything is automatic.

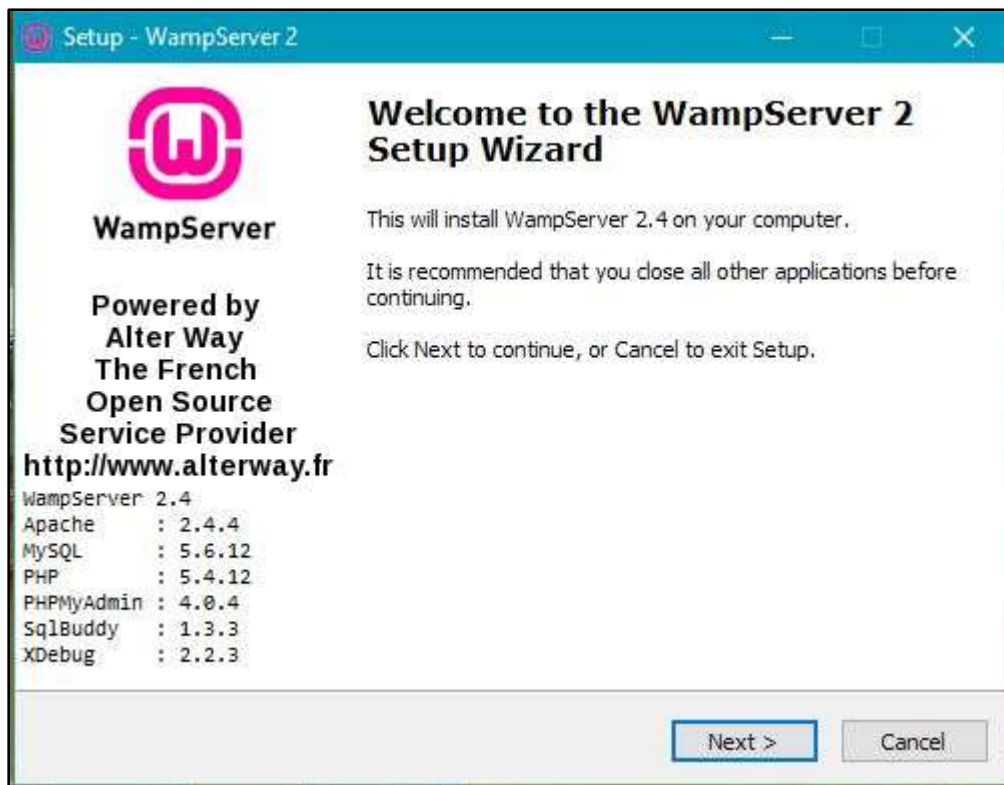


Figure A. 1: WampServer Installation Step 1

Step 2: After click next, accept the agreement and click next.

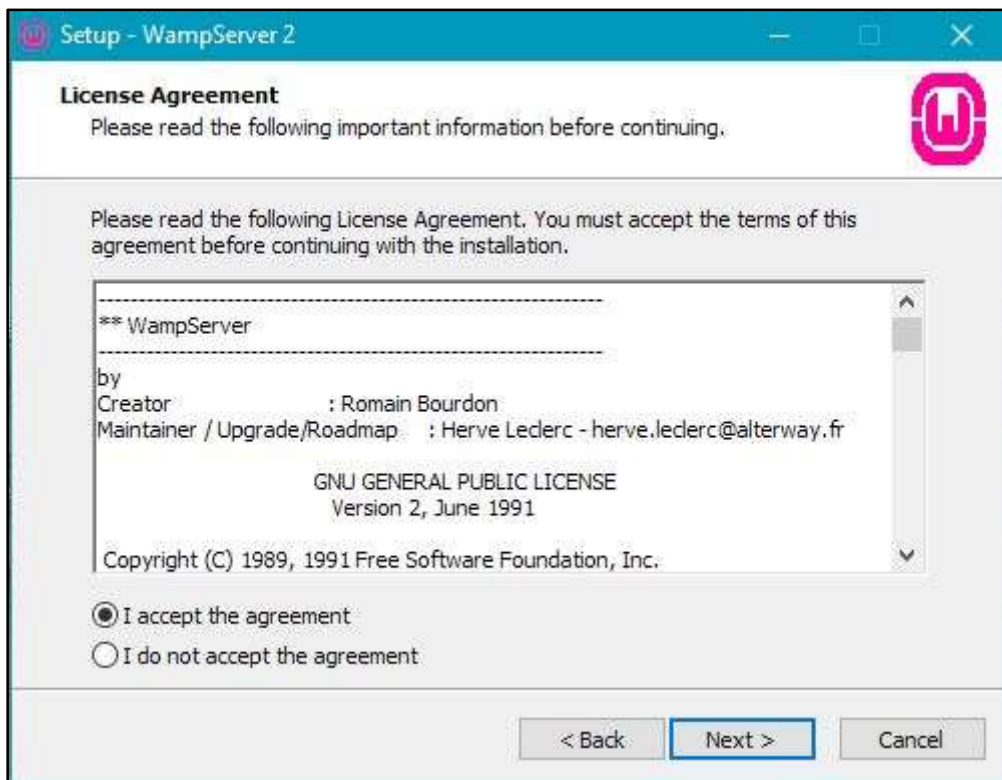


Figure A. 2: WampServer Installation Step 2

Step 3: Select the installation location of WampServer.

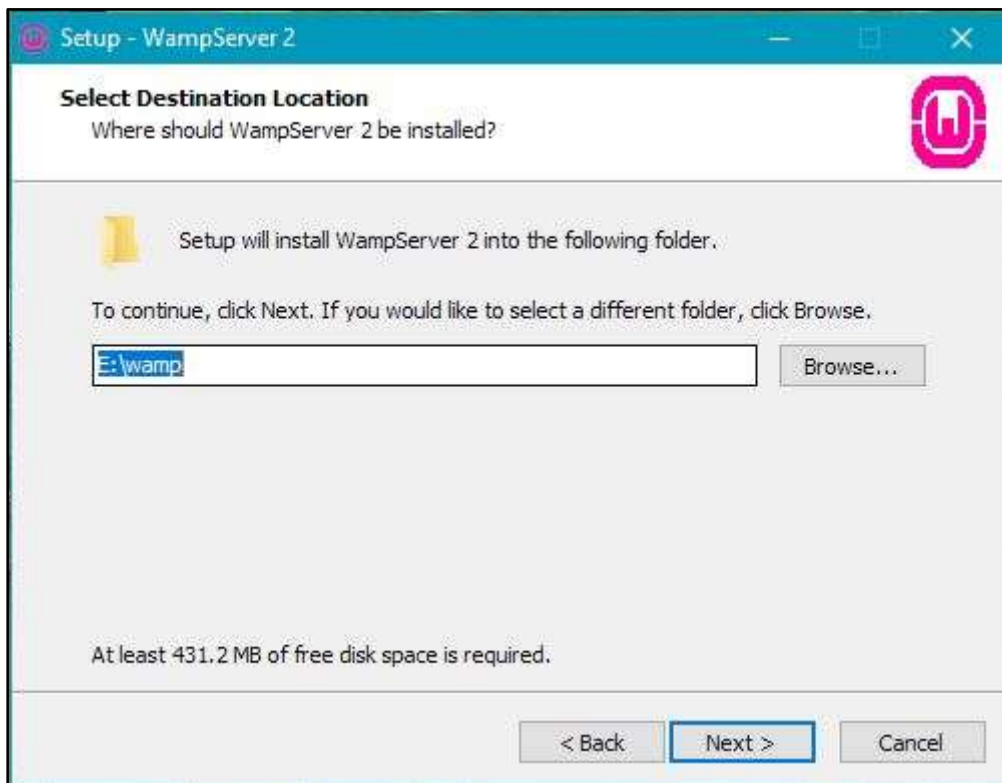


Figure A. 3: WampServer Installation Step 3

Step 4: Select additional tasks of WampServer.

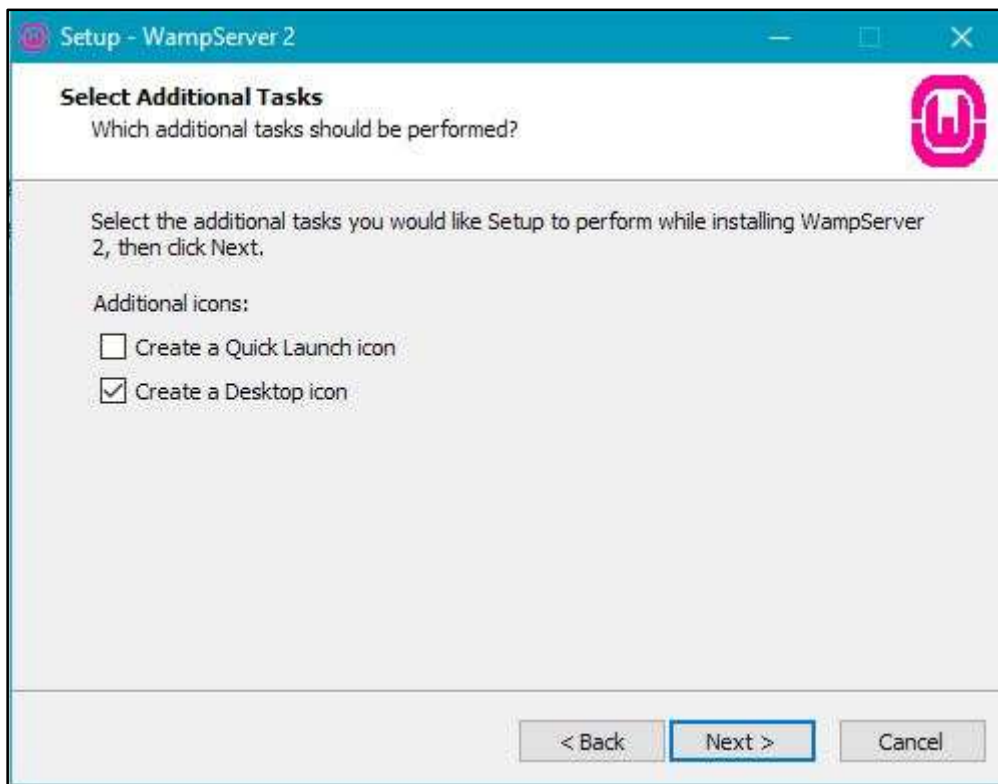


Figure A. 4: WampServer Installation Step 4

Step 5: Click Install and It finish automatically.

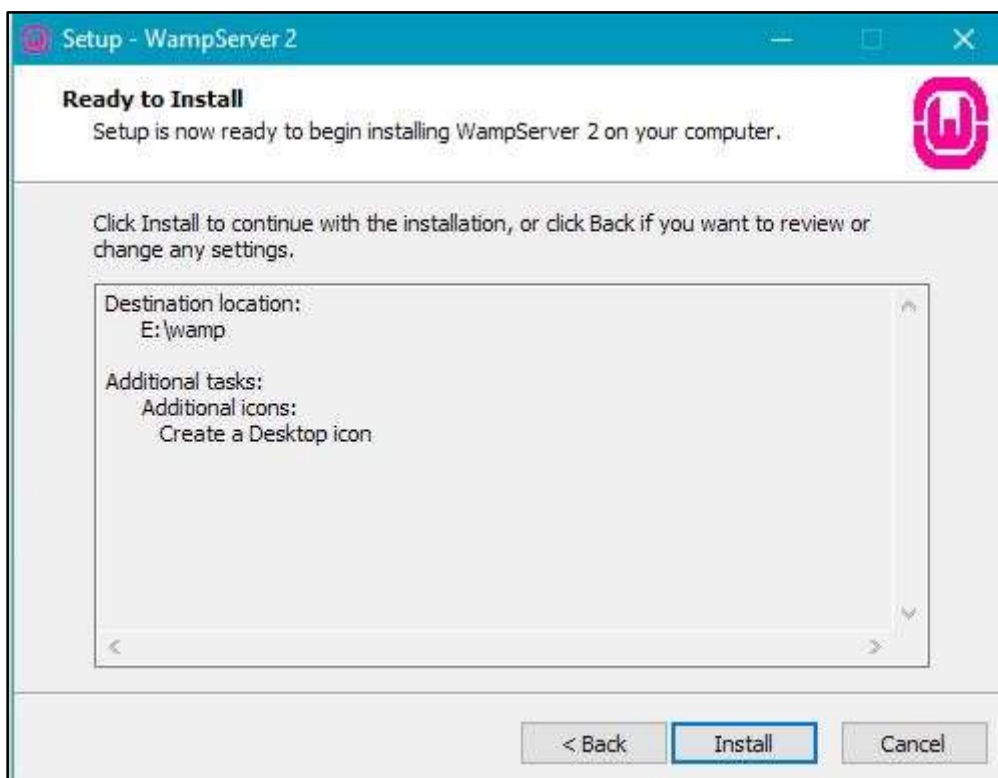


Figure A. 5: WampServer Installation Step 5

Step 6: After Install, start the WampServer in taskbar WampServer visible in green colour. After green colour, go and type in browser's address bar "localhost or 127.0.0.1".



Figure A. 6: WampServer Installation Step 6

Step 7: In browser, there is phpmyadmin under Tools heading, click that link and type username as "root" and password is blank. And click go button.



Figure A. 7: WampServer Installation Step 7

APPENDIX B - DESIGN DOCUMENTATION

The additional diagram such as activity diagram and sequence diagram of this system include in this section.

Activity diagram for student attendance is shown on following Figure B.1 Activity diagram for student attendance.

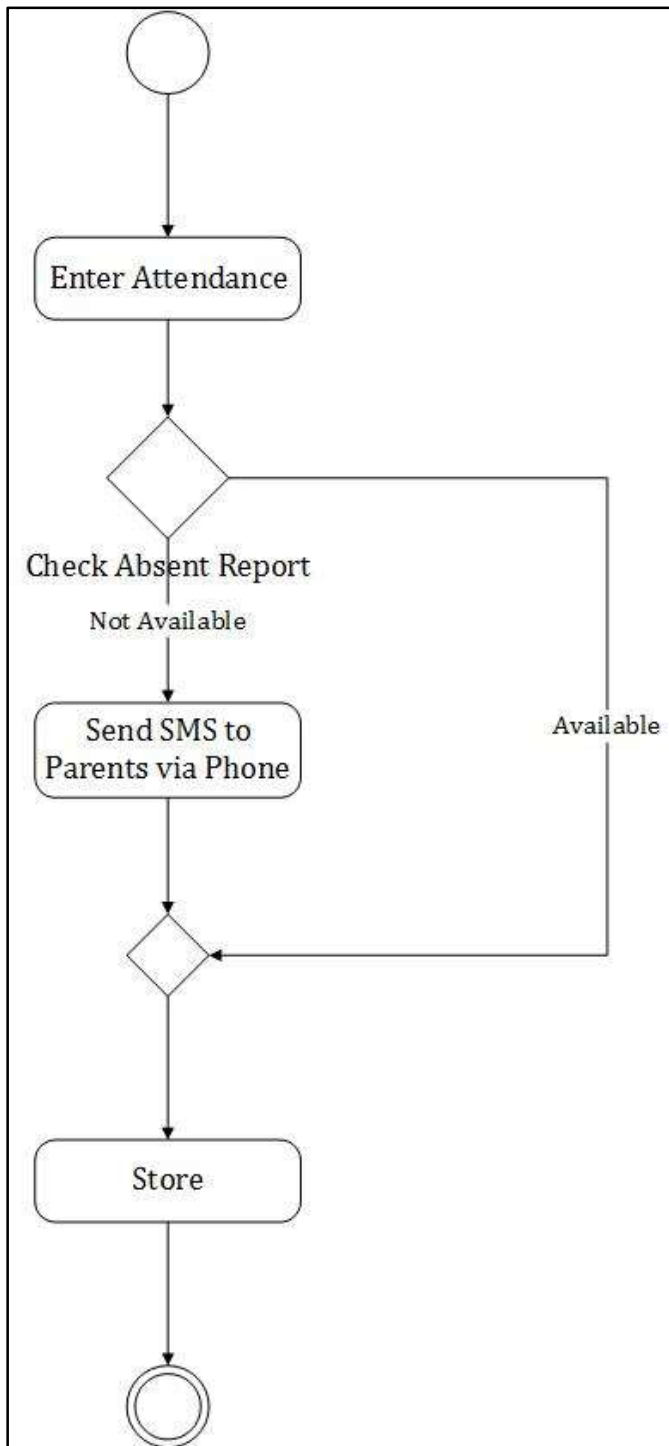


Figure B. 2: Activity diagram for student attendance

Activity diagram for staff leave management is shown on following Figure B.2:

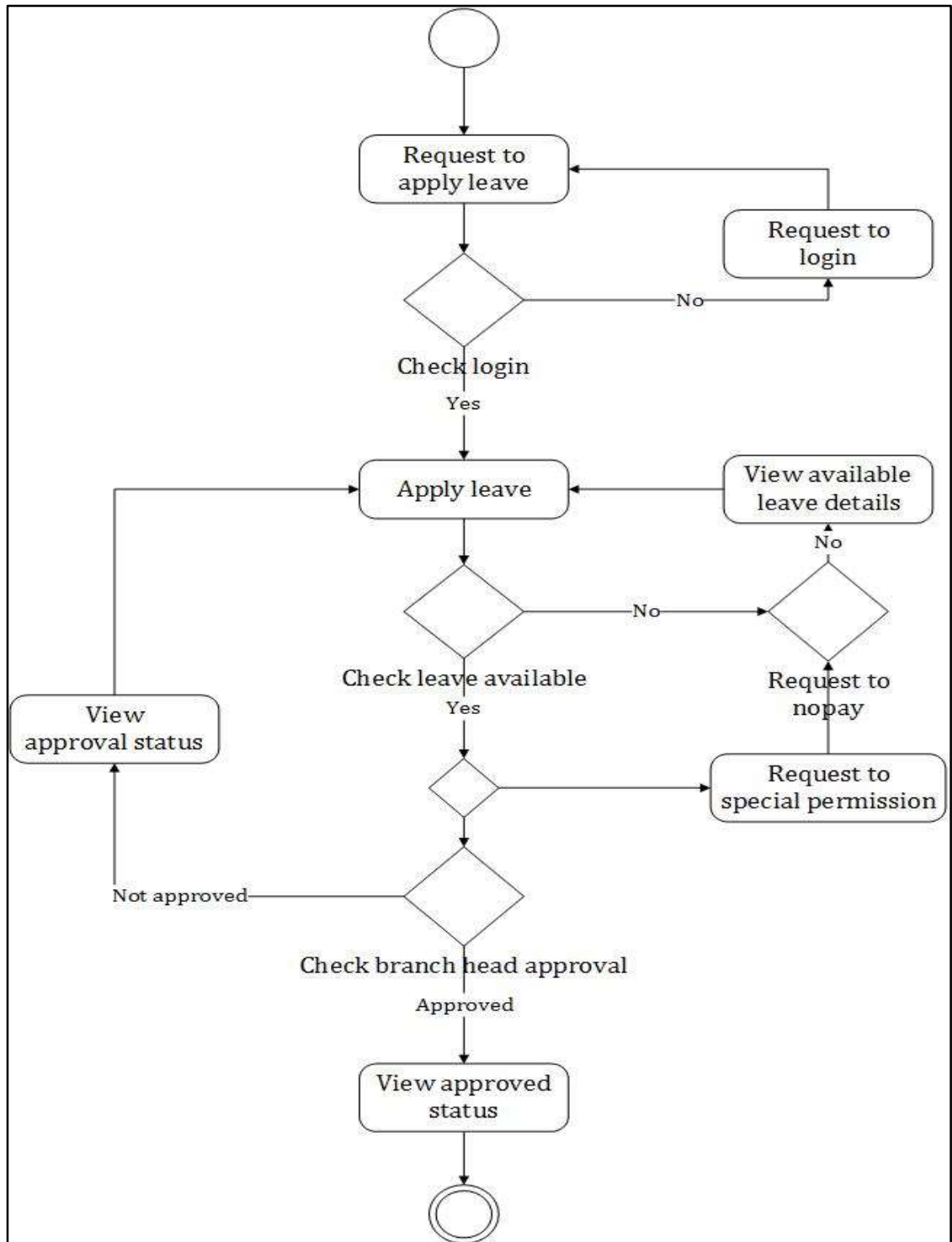


Figure B. 3: Activity diagram for staff leave management

Sequence diagram for student performance is shown on Figure B.3 Sequence diagram for student performance.

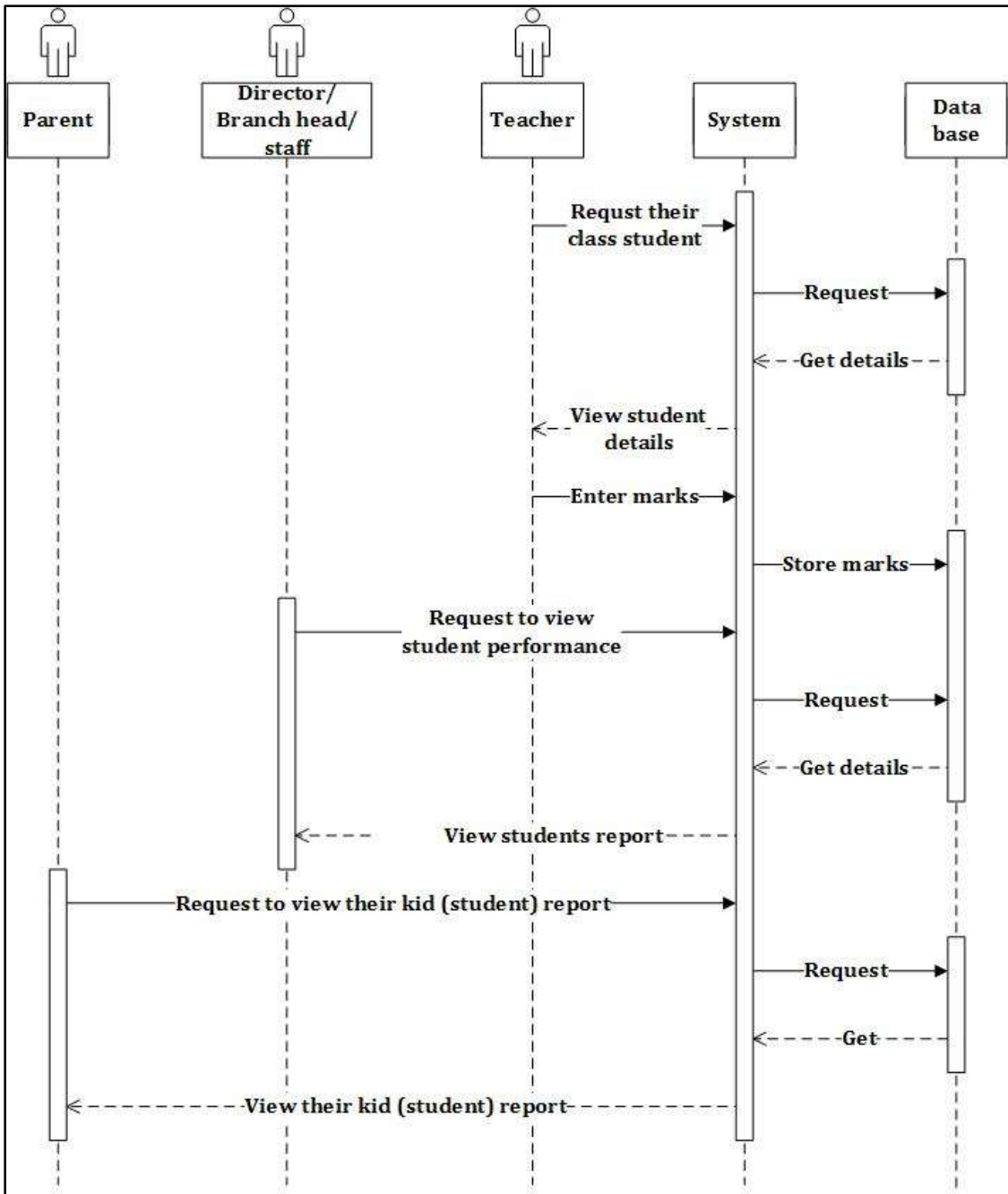


Figure B. 4: Sequence diagram for student performance

Sequence diagram for staff leave is shown on Figure B.4:

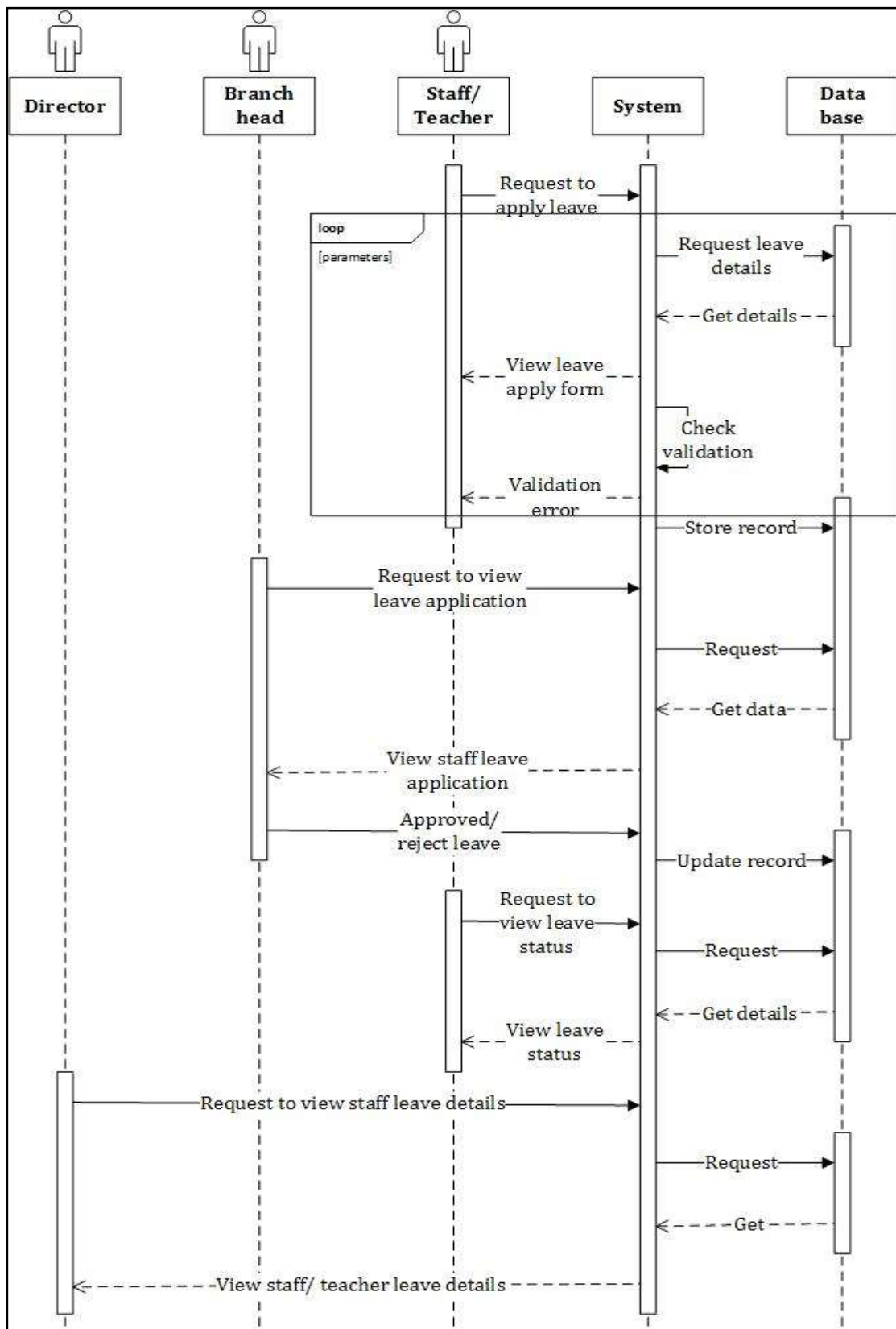


Figure B. 5: Sequence diagram for staff leave

Data Flow Diagram for preschool management system is shown on Figure B.5:

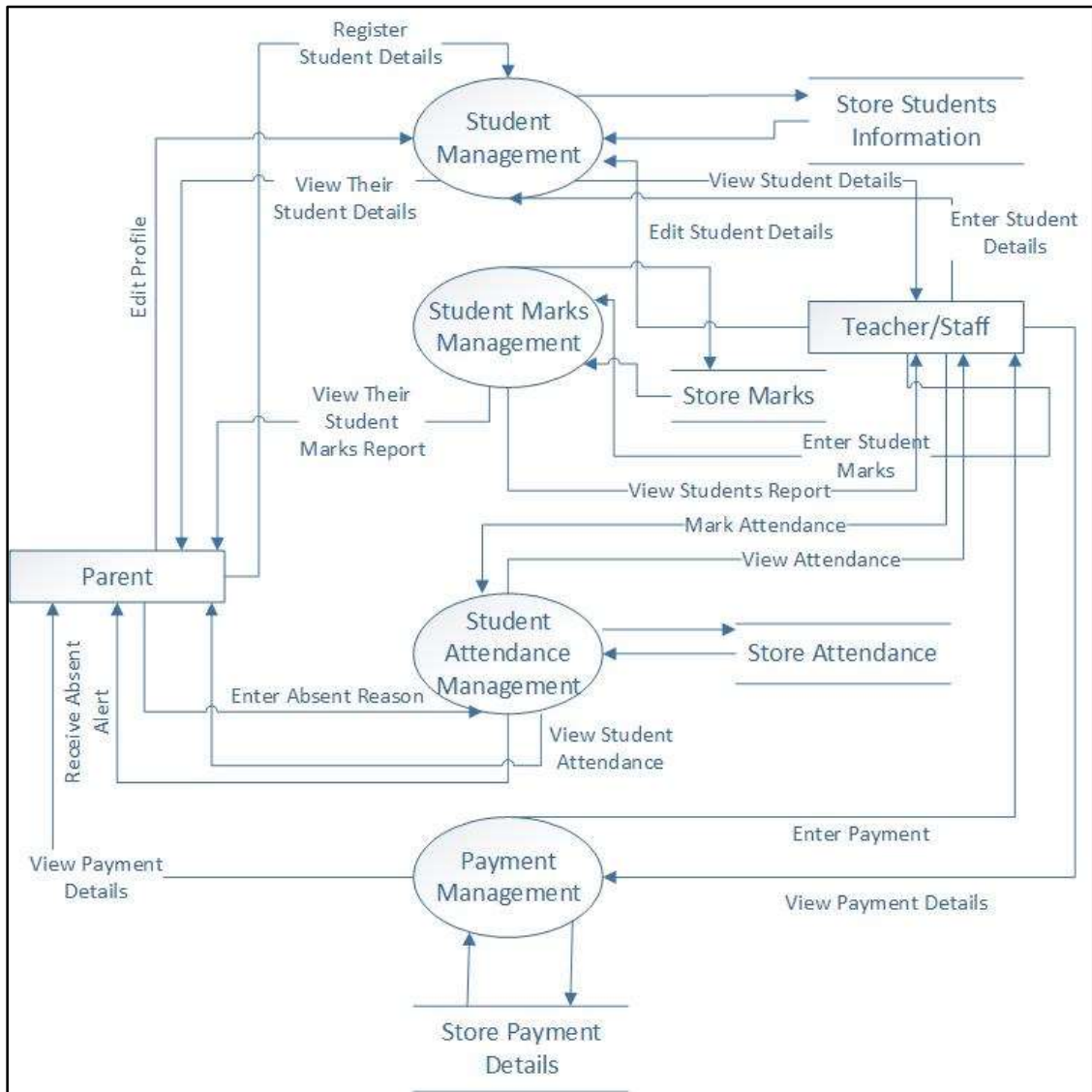


Figure B. 6: Data Flow Diagram Preschool management system level 1

APPENDIX C – USER DOCUMENTATION

User documentation refers to the documentation for this online preschool management system. The user documentation is designed to assist director, branch head, teachers, staff and parents to use the system. User levels are available to each and every member and different profile pages are provided for their management.

Index Page (Home menu)

An index page is generally the main page. Is used to facilitate navigation to other pages. Type the URL <http://localhost/kidspark/> and navigate. After navigation, user can find the Login page following Figure C.1 Index page gives overall structure of the system. In this page contains the following menus: home, about us, gallery, contact us and login. Home page refers as index page, it contains ‘what kids learn in kids’ park’, ‘our courses’, events and contacts. ‘About us’ page gives a short description of this institute. ‘Gallery’ page shows some image of students’ performance. ‘Contact us’ page facilitate the visitor to contact information.



Figure C. 1: Index Page

Login Page

Login page provides an individual gains access to a computer system by identifying and authenticating themselves. The following Figure C.2 shows the Login Page where the user will type wrong user name and password.

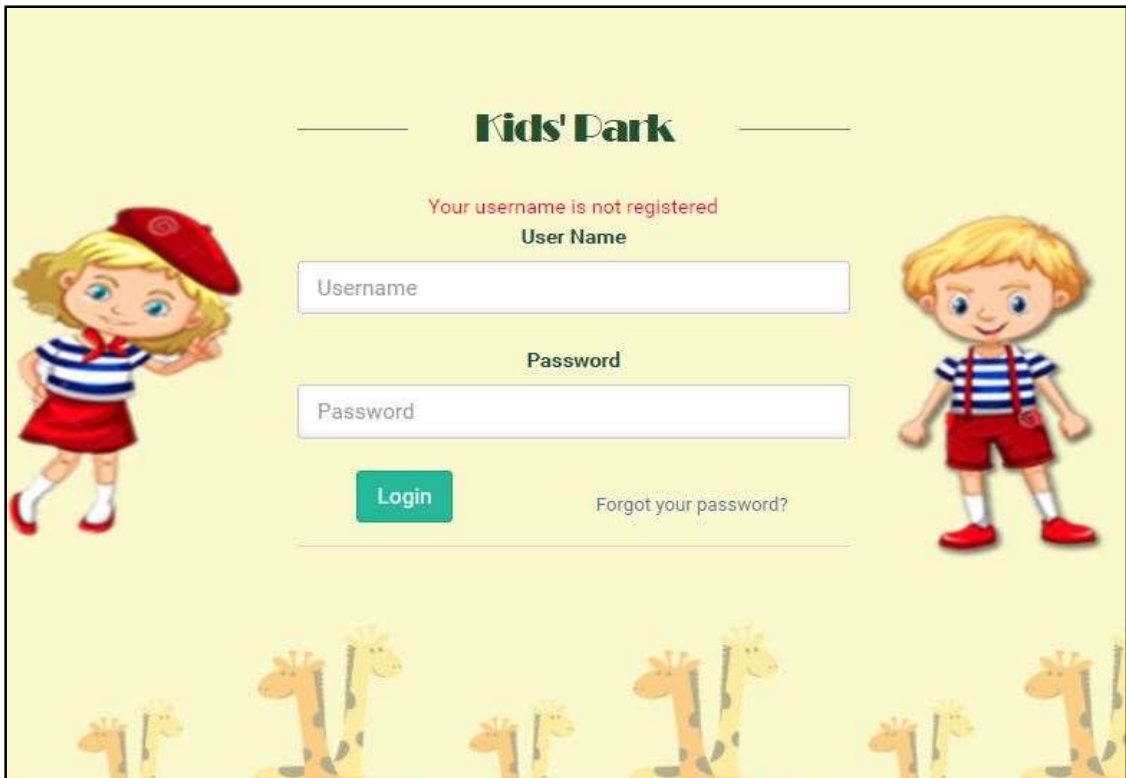


Figure C. 2: Login page with wrong user name

When enter correct username and wrong password, the system shows the error message that the password is incorrect, please try again. This error message is shown in the following Figure C.3 Incorrect Password:



Figure C. 3: Incorrect Password

If registered user attempt more than three times login, system provides a message and redirect to forget password form.

About Us Menu

It contains short introduction about Urumpirai kids' park. Refer Figure C.4: About Us Menu



Figure C. 4: About Us Menu

Gallery Menu

It contains number of galleries of Urumpirai kids' park. Refer the figure C.5: Gallery Menu



Figure C. 5: Gallery Menu

Contact Us Menu

Contact us page shows the contact details about this institute and its show the location of the institute in Figure C.6: Contact Us Menu



Figure C. 6: Contact Us Menu

Administrator (Director) Interface

After login as director, the system shows the home page and director menu. The home page is same as index page that already illustrated about this page under index page. In this administrator interface, it include following menu such as home, about us, gallery, contact us, students, staff, payment, branch and report. The director interface is shown in the following Figure C.7: Director Interface:



Figure C. 7 Director Interface

From home link top navigation dropdown user menu contain as following: profile, change password and log out. Every user interface have this link with same function. The following Figure C.8 shows the dropdown user menu of the system.



Figure C. 8: Dropdown User Menu

When we click the log out link the system logout and the system destroys all the session details and shows the public menu and home page. Profile link shows the personal detail of that the user log in the system. It allows edit and print privilege to the user. The following Figure C.9 shows the profile.

<u>Profile</u>	
Staff Id	S001
User Id	S001
Staff Name	Panchalingam
Address	Urumpirai
Telephone No	772998814
Email	panchalingam@gmail.com
Gender	male
NIC No	928191249V
Date Of Birth	1992-11-14
Designation	Director
Basic Salary	20000
Branch ID	B02

[Edit](#) [Print](#)

Figure C. 9: Profile

The change password link provides a form to change user password. In the form user has to enter the current password, new password and confirm new password and submit it. The change password interface is shown in the following Figure C.10 Change Password:



The image shows a web form titled "Change your password" in a dark red font. The form has a light yellow background. It contains four input fields: "User ID" with the value "S002", "Current Password" with the placeholder "type here", "New Password" with the placeholder "type here", and "Confirm New Password" with the placeholder "type here". Below the input fields are two buttons: a green "Save" button with a floppy disk icon and an orange "Cancel" button with an 'X' icon.

Figure C. 10: Change Password

Staff menu

From the director interface staff menu contain following pages: manage staff details, staff's leave and salary. Branch head interface have this menu with same page. Following Figure C.11 shows staff menu:

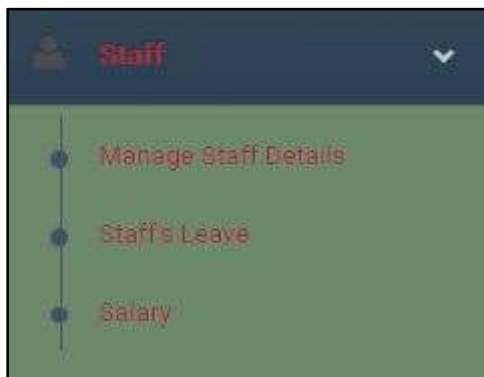


Figure C. 11: Staff menu

View staff details

Please see the Figure C.12 to get an idea about it and the arrow heads under the figure. This belongs to <http://localhost/kidspark/index.php?pg=staff.php&option=view> link

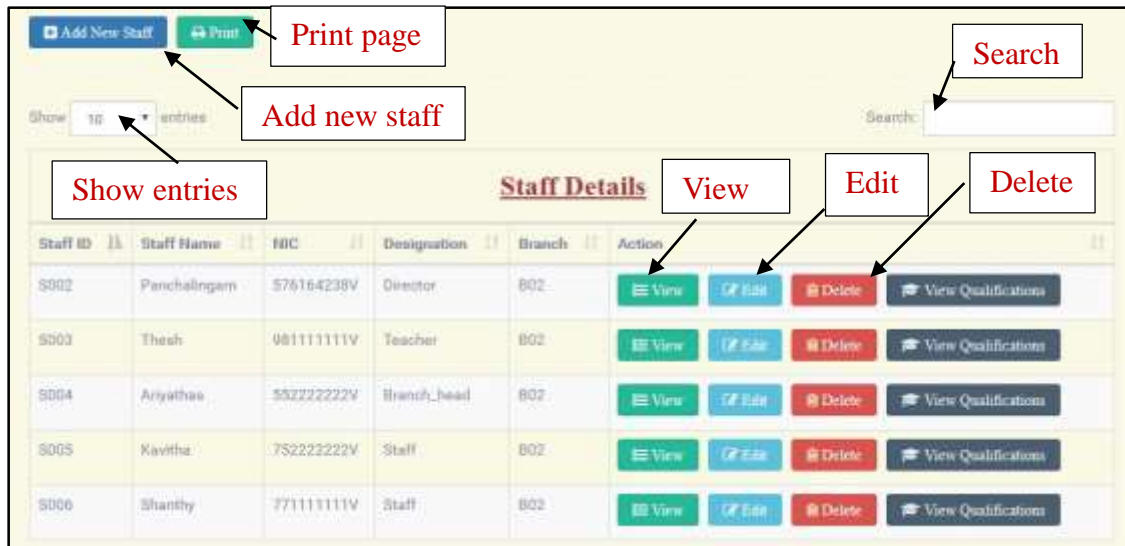


Figure C. 12: View Staff Details

- **Add new staff:** This helps to add new staff.
- **Print:** Print button help to print data.
- **Show Entries:** This helps to manage how many records should appear at a time. If there are more records than selected value, user can navigate to next set of items using which is located at the bottom right hand side of the table.
- **Search Box:** Data or set of data wanted by the user can be found with the help of this.
- **View:** Additional information belonging to a particular record can be viewed with the help of this.
- **Edit:** It helps to edit particular details.
- **Delete:** A particular data item can be deleted by using this.

Add New Staff

Addition of new staff to the system is allowed to the director by this page. Refer figure C.13: Add new staff

Path: Staff → Manage staff details → Click add new staff button

Fill all the details correctly and click on save button to add data. If the director cancel it click on cancel button.

Add New Staff

Staff ID	<input type="text" value="S010"/>
Staff Name	<input type="text" value="type here"/>
Address	<input type="text" value="type here"/>
Telephone No	<input type="text" value="eg: 77 000 0000"/>
Email	<input type="text" value="eg: a@gmail.com"/>
Gender	<input type="radio"/> Male <input checked="" type="radio"/> Female
NIC No	<input type="text" value="type here"/>
Date Of Birth	<input type="text" value="yyyy-mm-dd"/>
Designation	<input type="text" value="Select the designation"/>
Basic Salary	<input type="text"/>
Branch	<input type="text" value="Select the branch"/>

Figure C. 13 Add New Staff

After click save button a message will be display in this page. Refer Figure C.14

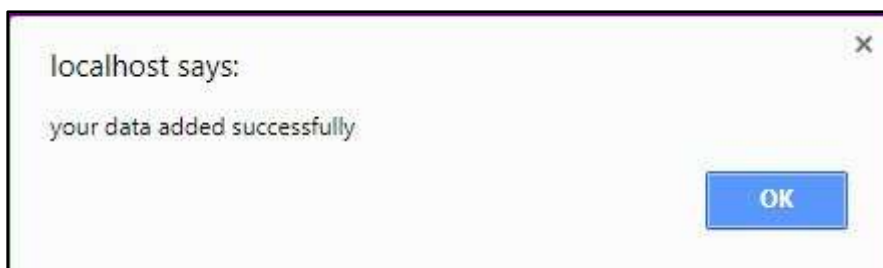


Figure C. 14: Data successfully added message

Print (staff's details)

Print staff button help print staff details. Refer figure C.15



Staff ID	Staff Name	NIC	Designation	Branch
S001	Geetha	828191249V	Teacher	B02
S002	Panchalingam	576164238V	Director	B02
S003	Thesh	081111111V	Teacher	B02
S004	Aziyofas	552222222V	Branch_head	B01
S005	Kavitha	752222222V	Staff	B02
S006	Shanthi	771111111V	Staff	B02
S007	Thushi	942222222V	Branch_head	B01

Figure C. 15: Print staff details

After that click the print button it load the print page. Refer figure C.16

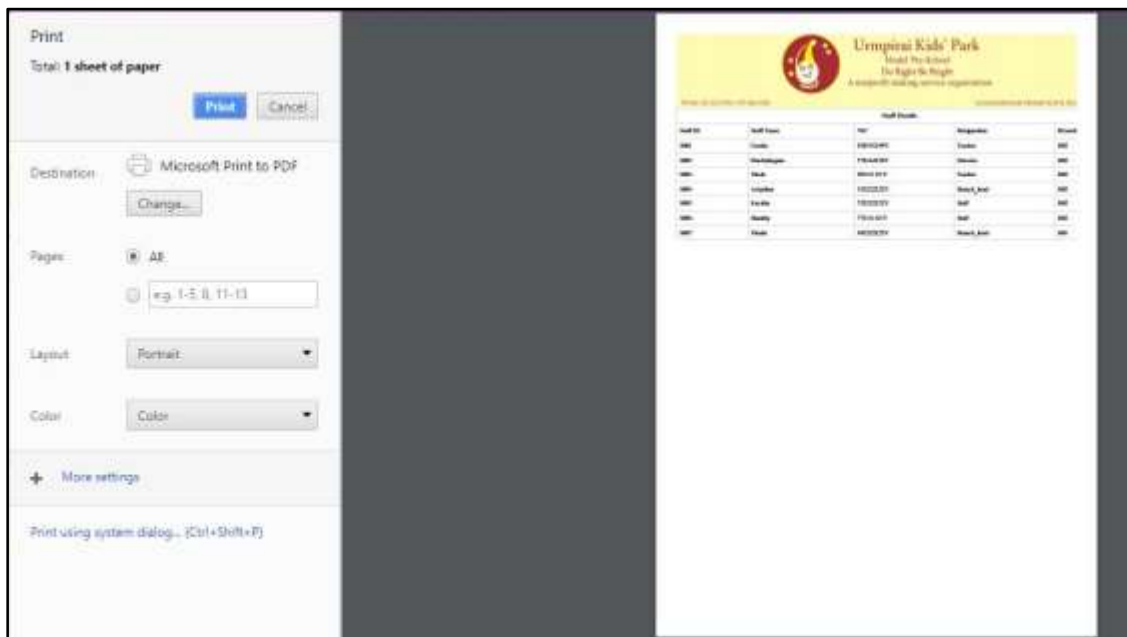


Figure C. 16: Print page

View particular information of staff

This page contain all personal information and educational qualification of staff. It contain following button: add educational qualification, edit, go back and print. Edit button help to edit information.

When click the go back button it goes to the view page. Refer figure C.17

Path: Staff → Manage staff details → Click view button

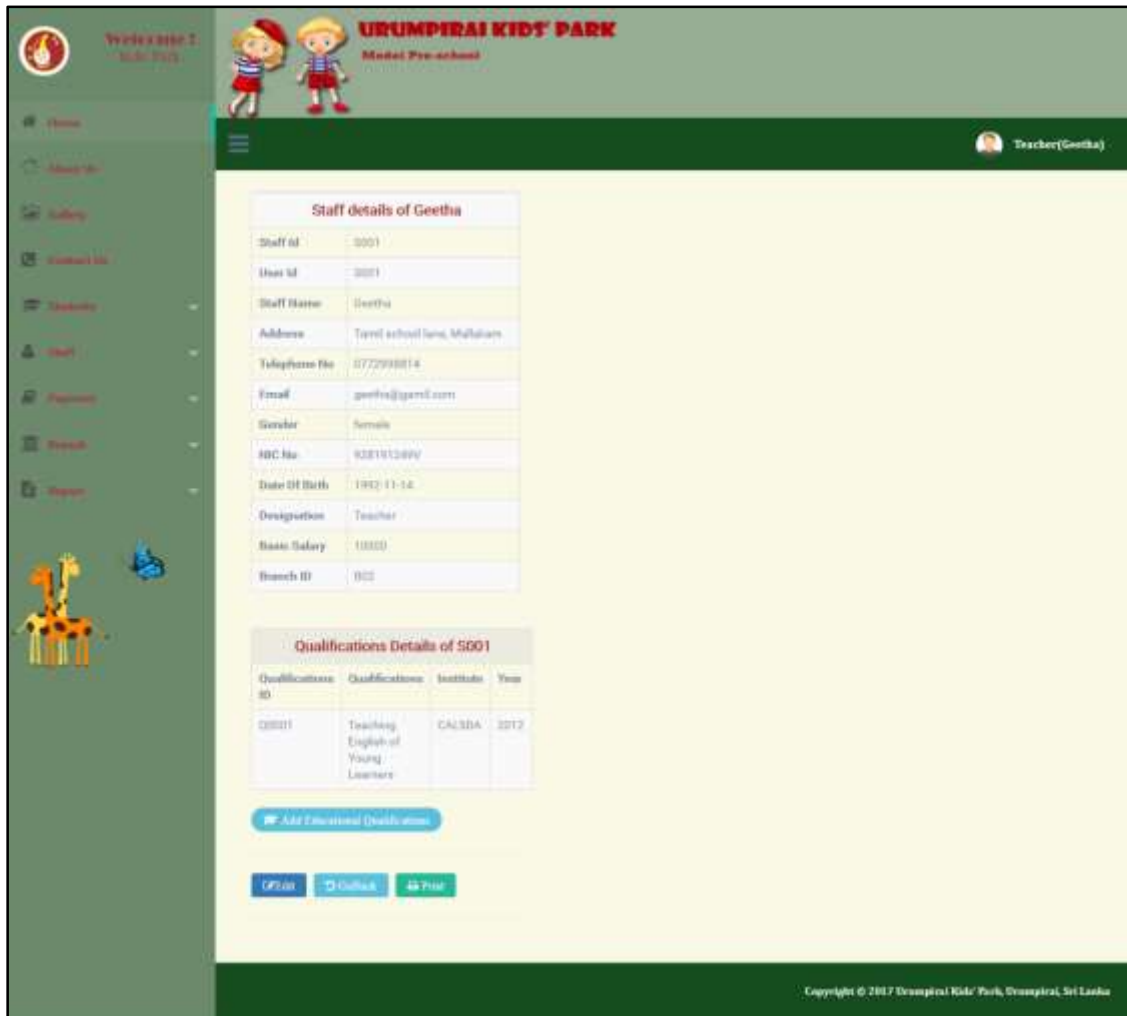


Figure C. 17: View staff additional information

Edit staff

Edit staff detail facilitate to edit information. Director and branch head have the permission to edit this page. After editing the information click the update button or if you wish to make no changes click the cancel button. Refer figure C.18.

Path: Staff → Manage staff details → Edit

Edit Staff

Staff ID	S002
Staff Name	Panchalingam
Address	Mallakam, Jaffna
Telephone No	0212241085
Email	panchalingam@yahoo.com
Gender	<input type="radio"/> Male <input checked="" type="radio"/> Female
NIC No	576164238V
Date Of Birth	1957-04-25
Designation	Director
Basic Salary	25000
Branch	Urumpirai
<input type="button" value="Update"/> <input type="button" value="Cancel"/>	

Figure C. 18: Edit Staff

Delete staff

It helps delete particular data. If we click the delete button there is an alert will display. Refer figure C.19. If we click the ok button that record will be delete or click the cancel button it will not be delete.

Path: Staff → Manage staff details → Delete

localhost says:

Do you want to delete this record?

Figure C. 19: Delete Staff

Staff's leave page

This page contain staff's leave details and status field has two button to confirm or reject the leave applied by the staff. Branch head interface has this page with same function. Refer figure C.20.

Path: Staff → Staff's leave

Apply Date	Leave Type	Start Date	End Date	No of Days	Status
2017-09-01	Sick Leave	2017-09-01	2017-09-07	5	<input checked="" type="checkbox"/> Confirm <input type="checkbox"/> Reject
2017-09-18	Casual Leave	2017-09-18	2017-09-21	3	<input checked="" type="checkbox"/> Confirm <input type="checkbox"/> Reject

Figure C. 20: Staff leave confirmation table

Salary Page

This page refer the salary details of staff. Select the branch, year and month then click print button to print this page. Refer Figure C.21.

Path: Staff → Salary

Staff ID	Name	Basic Salary	Additional Class	EPF	ETF	Total Earning	Total Deduction	Net Salary
S001	Geetha	10000	5000	800	200	15000	1000	14000
S002	Panchalingam	25000	5000	2000	500	30000	2500	27500
S003	Theesh	10000	0	800	200	10000	1000	9000
S004	Ariyathaa	20000	0	1600	400	20000	2000	18000
S005	Kavitha	12000	0	960	240	12000	1200	10800
S006	Shanthi	10000	0	800	200	10000	1000	9000

Figure C. 21: Staff's salary

Payment menu

From the director interface payment menu contain following pages: Class rate, student event payment and student class payment. Branch head interface has this page with same function. Following Figure C.22 shows payment menu:



Figure C. 22: Payment menu

Class rate page

Class rate define the rate information of particular class. Refer Figure C.23

Path: Payment → Class Rate



The screenshot shows the 'Class Rate Details' page. At the top, there are two buttons: 'Add New Rate' (blue) and 'Print' (green). Below them is a 'Show' dropdown set to '10' and a search box. The main content is a table with the following data:

Rate for	Amount	Action
CLS01	1550	Edit Delete
CLS02	1250	Edit Delete
CLS03	1350	Edit Delete

Figure C. 23: Class Rate

Student Event Payment page

Path: Payment → Students' event payment

This page has three select element such as: branch, event and class. If the director select any branch, it load the event and class element. After select the event and class, we can see the students' event payment report. Staff interface has this page with same function. Refer Figure C.24

If any student did not make any event payment, there is a add payment button shows under the amount field. Director can be able to print the report too.

Students' Event Payment

Branch:

Event:

Class:

Students Event Payment Report

Student ID	Student Name	Amount	Payment Date
STU0001	Santhos	200	2017-09-12
STU0005	Anojan	<input type="button" value="Make Payment"/>	Null
STU0007	Arush	<input type="button" value="Make Payment"/>	Null

Figure C. 24: Student Event Payment Report

When we click the print button, the image will be: Refer Figure C.25



Urumpirai Kids' Park
 Model Pre-School
 Do Right Be Bright
 A nonprofit making service organization

TP NO: 021 223 0754 / 077 656 5258 GANAVAIRAVAR SWAMY KOVIL ROAD, URUMPIRAI

Students Event Payment Report

Student ID	Student Name	Amount	Payment Date
STU0001	Santhos	200	2017-09-12
STU0005	Anojan	Null	Null
STU0007	Arush	Null	Null

Figure C. 25: Print event payment report

When click the make payment button, it goes to the add payment page. Refer figure C.26

Add Event Payment

Payment ID

Student ID

Pay for

Payment Date

Amount

Figure C. 26: Add event payment

Student Class Payment page

Path: Payment → Student class Payment

This page has three select element such as: branch, class and year. If the director select any branch, it load the class and year element. After select the class and year, we can see the students' class payment report. Staff interface has this page with same function. Refer Figure C.27.

If any student did not make any class payment, there is a add payment button shows under the amount field. Director can be able to print the report too.

Students' Class Payment

Branch

Class

Year

Students Class Payment Report

Student ID	Student Name	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total	Make Payment
STU0001	Senthes	1000	1000											2000	<input type="button" value="MakePayment"/>
STU0005	Anojan													0	<input type="button" value="MakePayment"/>
STU0007	Arush													0	<input type="button" value="MakePayment"/>

Figure C. 27: Students' class payment

When click the make payment button, it goes to the add payment page. Refer Figure C.28

Add Class Payment

Payment ID: PAY00004

Student ID: STU0004

Year: 2017

Month: Select Month ▼

Amount: type here

Payment Date: 2017-11-30

Save GoBack

Figure C. 28: Add Class Payment

Manage Students' Details

Path: Students → Manage Students' Details

Director can see all the students detail in this view table. This page has view, edit, delete, add new students and print button. Staff and branch head user interface have this page with same function. Refer Figure C.29.

Student ID	Student Name	Date Of Birth	Parent ID	Action
STU0001	Santha	2014-05-12	P0002	View Edit Delete
STU0002	Geetha	2013-05-04	P0003	View Edit Delete
STU0003	Akash	2012-07-13	P0004	View Edit Delete
STU0005	Arojan	2014-03-29	P0001	View Edit Delete
STU0006	Charmiyan	2014-10-15	P0004	View Edit Delete
STU0007	Arush	2014-11-12	P0006	View Edit Delete

Figure C. 29: Students Details

Add new student

Path: Students → Manage Students' Details → Add New Student

In this page director have to fill all the details properly. Refer Figure C.30

Select link redirect to select parent page.

Add New Student

Student ID:

Parent ID: [Select](#)

Student Name:

Date Of Birth:

Gender: Male Female

Registered Date:

Branch:

Figure C. 30: Add New Staff

Select parent

In this page, if director click the select button parent id will be filled in the parent id field in student add new page. Refer Figure C.31

If this page have not particular student parent detail director has to click add new button.

Refer Figure C.32



Parent ID	Parent Name	Telephone No	User ID	Action
P0001	Kuniam	212241085	P0001	Select
P0002	Parameswaran	778854791	P0002	Select
P0003	Jegatheeshwaran	776998683	P0003	Select
P0004	Sakthi	772998814	P0004	Select
P0005	Sriharan	772998814	P0005	Select
P0006	Pushpanathan	772998814	P0006	Select
P0007	Ranjan	770000000	P0007	Select

Figure C. 31: Select parent

Add New Parent

Parent ID	<input type="text" value="P0010"/>
Parent Type	<input type="text" value="--Select--"/>
Parent Name	<input type="text" value="type here"/>
Address	<input type="text" value="type here"/>
Telephone No	<input type="text" value="eg: 77 000 0000"/>
Email	<input type="text" value="eg: a@gmail.com"/>
User ID	<input type="text" value="P0010"/>

Figure C. 32: Add new parent

After fill all the details click save button. It will be save in the database and redirect the select parent page.

View Particular Student details

Path: Students → Manage Students' Details → View

Director can see student details and class detail of a particular student. Refer figure C.33

If a student is not added any class click add new class button. Refer figure C.34

Student's Personal Information

Student ID	STU0001
Student Name	Santhos
Date Of Birth	2014-05-12
Gender	male
Parent ID	P0002
Registered Date	2017-05-11
Branch	B02

Class Detail Of Santhos

Class ID	Class Type	Division	Class Teacher	Year
CLS01	Playgroup	A	S003	2017

[+ Add class information](#)

✎ Edit

↶ GoBack

🖨 Print

Figure C. 33: Student personal information

Add Class Information

Student ID:

Class:

Year:

Figure C. 34: Add Class Information

Manage Branch Details

Path: Branch → Manage Branch Details

Director can see all branches detail in this view table. This page has view, edit, delete, add new branch and print button. Refer Figure C.35

Branch Details

Branch ID	Branch Name	Telephone No	Head Staff ID	Action
001	Thiruvalli	0211111111	S007	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
002	Urumpirai	0212222222	S004	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

Showing 1 to 2 of 2 entries

Figure C. 35: Mange branch detail

Subject

Path: Students → Manage Subject details

Director can see all subject details in this view table. This page has view, edit, delete, add new subject and print button. Branch head interface also has this page with same function. Refer Figure C.36



Figure C. 36: Subject detail

Class Subject Map

Path: Students → Class Subject map

This page illustrates add more subject in selected class. See the steps as follow:

Select Class → View table → Select Subject → Click add (it will be add in table)



Figure C. 37: Class Subject Map

Branch Head Interface

After login as branch head, the system shows the home page will be same as director.

Profile Menu changed as branch head. Refer figure C.38

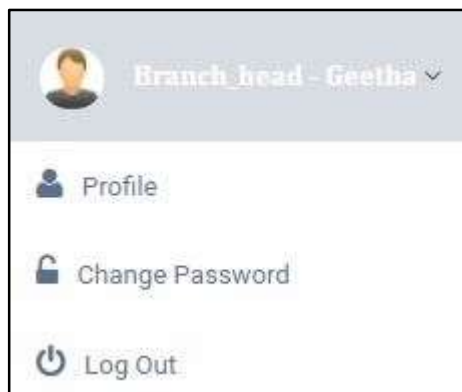


Figure C. 38: Branch head profile menu

Manage Class Details

Path: Students → Manage Class Detail

Branch head can see all the class detail in this view table. This page has view, edit, delete, add new students and print button. Director interface also has this page with same function.



The screenshot displays a web interface for managing class details. At the top, there are buttons for 'Add New Class' and 'Print'. Below these is a search bar and a dropdown menu set to '10 entries'. The main content is a table titled 'Class Details' with the following data:

Class ID	Class Type	Division	Staff ID	Action
CLS01	Playgroup	A	5003	View Edit Delete
CLS02	3Plus	A	5002	View Edit Delete
CLS03	4Plus	A	5001	View Edit Delete
CLS04	Playgroup	B	5005	View Edit Delete

At the bottom, it shows 'Showing 1 to 4 of 4 entries' and navigation buttons for 'Previous' and 'Next'.

Figure C. 39: Manage Class Details

Staff Interface

After login as staff, the system shows the home page will be same as director.

Profile Menu changed as staff. Refer figure C.40

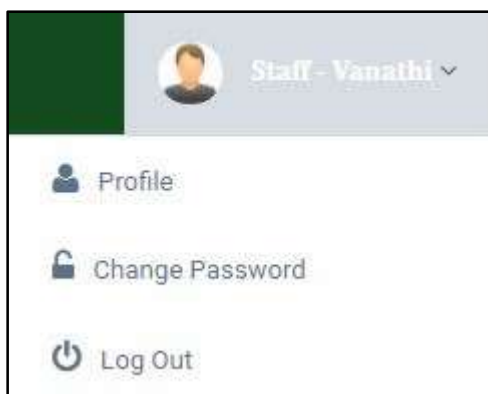


Figure C. 40: Staff Profile Menu

Manage Expenditure Details

Path: Branch → Manage Expenditure Details

Staff can see all the expense detail in this view table. This page has view, edit, delete, add new expense and print button. Director and branch head interface also have this page with same function.

Expense ID	Expense For	Date	Amount	Branch ID	Action
EXP0001	Stationary	2017-07-06	250	802	View Edit Delete
EXP0002	Printing	2017-07-14	150	802	View Edit Delete

Figure C. 41: Expense Details

Apply Leave

Path: Staff → Apply leave

Staff can see all their leave detail in this view table. This page has view, edit, delete, add new expense and print button. Teacher interface also have this page with same function.

Apply Date	Leave Type	Start Date	End Date	No of Days	Status
2017-09-01	Sick Leave	2017-09-01	2017-09-07	5	confirm

Figure C. 42: Apply Leave

Message

Path: Branch → Message

Staff can send message to select the message to option (other staff / director/ teacher /branch head /parent) and type the message in message text area after click send button.

Director, Branch head and Teacher interface also have this page with same function.

Refer Figure C.43



The screenshot shows a web form titled "Message" with a light yellow background. The form contains the following fields and controls:

- Message ID:** A text input field containing "MG0002".
- Message From:** A text input field containing "Panchalingam".
- Message To:** A dropdown menu with the text "Select The Person" and a downward arrow.
- Message:** A large text area with the placeholder text "type here" and a small icon in the bottom right corner.
- Date:** A text input field containing "2017/11/08".
- Send Button:** A green button with a white arrow icon and the text "Send".

Figure C. 43: Message

Teacher Interface

After login as teacher, the system shows the home page and teacher menu. The home page is same as index page that already illustrated under index page. This teacher interface includes following menu such as home, about us, gallery, contact us, students, staff, branch and report. The teacher interface is shown in the following Figure C.44 teacher Interface:



Figure C. 44: Teacher Interface

Student menu

From the student interface student menu contains following pages: manage students' details, students' attendance, add your students' marks, and view your students' marks and absence. Following Figure C.45 shows student menu:

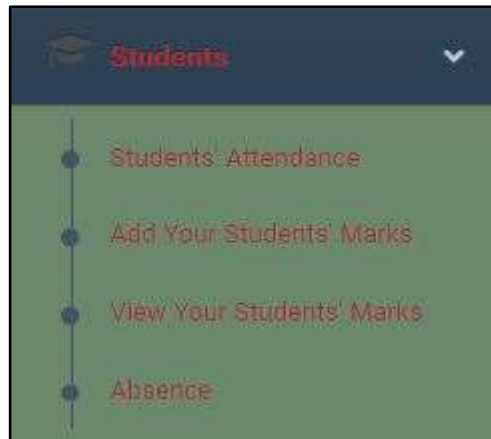


Figure C. 45: Students Menu

Students' Attendance page

Path: Students → Manage Students' Details

In this page teacher can be mark their class students' attendance in date wise. S/he can select the date from the date option. These steps include the following:

Select Class → Select Date → Select the Present/ absent in the status field → Click save button.

Refer Figure C.46 Students' attendance.

 A screenshot of a web form titled "Mark Your Students Attendance" in red text. The form has a light yellow background. At the top, there are two input fields: "Class" with a dropdown menu showing "4Plus (CL503-A)" and "Date" with a text input showing "2017-11-03". Below these fields is a table with four columns: "Student ID", "Student Name", "Status", and "Message". The table contains two rows of data. The first row has "STU0003" for Student ID, "Alanh" for Student Name, and "Present Absent" for Status. The second row has "STU0006" for Student ID, "Sajwan" for Student Name, and "Present Absent" for Status. At the bottom left of the form, there is a green "Save" button.

Student ID	Student Name	Status	Message
STU0003	Alanh	Present Absent	
STU0006	Sajwan	Present Absent	

Figure C. 46: Students' attendance

In this students' attendance table, status field refer the teacher can be able to click present or absent button. Message field represents if a student informed about their absent via message there is message shown under the message field.

Add Your Students' Marks page

Path: Students → Add Your → Students → Marks

The form that helps to teacher add their class students' marks. The steps are as follows:

Select class → Select subject → Select term → add the marks → click save button

Refer figure C.47

Add Your Students Marks

Class: 4Plus (CLS03-A) ▾

Subject: Select Subject ▾

Term: 3rd Term ▾

Student ID	Student Name	Marks
STU0003	Akash	<input type="text"/>
STU0006	Rajeevan	<input type="text"/>

Figure C. 47: Add your student marks

View Your Students' Marks page

Path: Students → View your students' marks

Teacher can able to see their class students' marks. These steps include the following:

Select Class → Select Year → Select Term, after teacher can be able to see their students' marks table.

Refer Figure C.48

Students' Marks Report

Class: Playgroup (CLS01-A) ▾

Year: 2017 ▾

Term: 1st Term ▾

Student ID	Student Name	Tamil	English	Art	Average	Rank	Action
STU0006	Theshpriyan	A	A	A	80	3	<input type="button" value="Edit"/>
STU0007	Priyanka	A	A	A	90	1	<input type="button" value="Edit"/>
STU0008	Dilaxson	A	A	A	85	2	<input type="button" value="Edit"/>

Figure C. 48: Students' Marks Report

Teacher can print this report too. The report will be: Refer the Figure C.49



Urmpirai Kids' Park
Model Pre-School
Do Right Be Bright
A nonprofit making service organization

TP NO: 021 223 0754 / 077 656 5258 GANAVAJRAVAR SWAMY KOVIL ROAD, URUMPIRAI

Students Marks Report

Description	
Class	Playgroup(A)
Year	2017
Term	1

Student ID	Student Name	Tamil	English	Art	Average	Rank
STU0006	Thesigeyan	A	A	A	80	3
STU0007	Priyanka	A	A	A	90	1
STU0008	Dhruvan	A	A	A	93	2

[Print](#)

Figure C. 49: Print student report

Parent Interface

After login as parent, the system shows the home page and parent menu. The home page is same as index page that already illustrated under index page. This parent interface includes following menu such as home, about us, gallery, contact us, student's details, student's attendance, absence, view students marks and events. The parent interface is shown in the following Figure C.50 parent Interface:



Figure C. 50: Parent Interface

Students' Details menu

In this page parent can be able to see their kid's/kids' detail and also see their detail.

First parent should be select their kid (student) because parent's one or more kid's study in this institute. Refer figure C.51

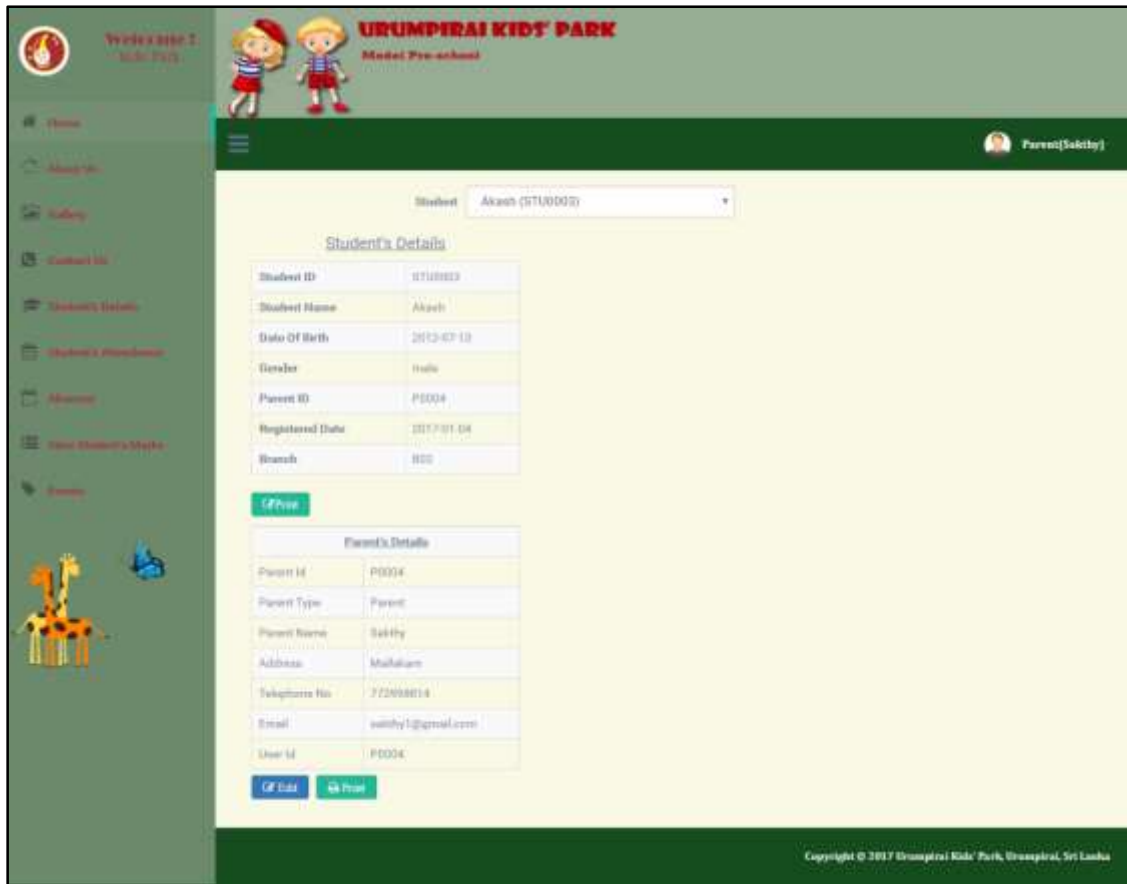


Figure C. 51: Kid's detail in parent interface

In this page parent can be able to print their kid's details and edit/print their details.

Students' attendance menu

In this page parents can see their student's attendance. See the following steps:

Select Student → Select from → Select To, after teacher can be able to see your kid's attendance. Refer figure C.52



Figure C. 52: Kid's attendance report

Absence

Parent can see their kid's (student) details. And they can inform (apply) their kid absent via add new button. Refer Figure C.53



Student ID	Absent Date	Reason	Action
STU0001	2017-07-19	Sick of fever	View
STU0001	2017-10-02	Sick of fever	View
STU0001	2017-12-03	Sick of fever	View

Figure C. 53: Absent detail

Add new absent

Path: Absence → Add new



Add New Absence Information

Student ID: Akash (STU0003)

Absent Date: 2017-11-07

Reason: Because of heavy rain

[Save](#) [Cancel](#)

Figure C. 54: Add new absent

After that, if you want to save the details click save button otherwise click cancel button.

View student's marks

Parent can be able to see their kid's marks after select following option such as:

Select class → Select year → Select Term



The screenshot displays a web interface titled "Your Kid's Marks Report". It features three dropdown menus for filtering: "Class" (set to "4Plus (CLS03-A)"), "Year" (set to "2017"), and "Term" (set to "1st Term"). Below these filters is a table with the following data:

Student ID	Student Name	Tamil	English	Average
STU0003	Akash	A	A	89.5

Figure C. 55: Kid's marks report

APPENDIX D - MANAGEMENT REPORT

The system allows the director, branch head, teacher and staff to generate reports with their appropriate power in different categories. They can view the following reports: students' details (class/ division/ branch wise), staff details, students' attendance, Expense, students marks.

Students' details - class wise report

System Administrator (director) and branch head to view students' details (class wise) by selected following steps:

Select branch → Select class → Select Year, after that details will be displayed.

Report of Students' Details - Class wise

Branch: Urungirai
Class: 4Plus
Year: 2017

Report of Students' Details - Class wise

Description	
Branch	Urungirai
Class Type	4Plus
Year	2017

Student ID	Student Name	DOB	Gender	Parent ID	Registered Date	Branch ID
STU0005	Akash	2012-07-13	male	P0004	2017-01-04	B02
STU0008	Sajeevan	2012-08-08	male	P0005	2017-01-03	B02

Print

Figure D. 1: Student Report Class Wise

Students' details - division wise report

System Administrator (director), branch head and teacher to view students' details (division wise) by selected following steps:

Select branch → Select division → Select Year, after that details will be displayed.

Report of Students' Details - Division wise

Branch: ▾

Division: ▾

Year: ▾

Report of Students' Details - Division wise

Description	
Branch	Urumpirai
Division	Playgroup(B)
Year	2017

Student ID	Student Name	DOB	Gender	Parent ID	Registered Date	Branch ID
STU0006	Charmyan	2014-10-15	female	P0004	2016-09-07	B02
STU0009	Thandini	2014-03-24	female	P0006	2017-01-06	B02

Figure D. 2: Student Report Division Wise

Students' details - branch wise report

System Administrator (director) and branch head to view students' details (branch wise) by selected following steps:

Select branch → Select Year, after that details will be displayed.

Report of Students' Details - Branch wise

Branch: ▾

Year: ▾

Report of Students' Details - Branch wise

Description	
Branch	Thinnaveli
Year	2017

Student ID	Student Name	DOB	Gender	Parent ID	Registered Date
STU0004	Sanju	2013-05-22	male	P0005	2017-01-04
STU0010	Rineeta	2014-10-02	female	P0017	2017-01-26

Figure D. 3: Student Report Branch Wise

Staff's details - branch wise report

System Administrator (director) and branch head to view staff's details (branch wise) by selected branch, after that details will be displayed.

Report of Staffs' Details - Branch wise

Branch:

Report of staffs' Details - Branch wise

Description	
Branch	B01

Staff ID	User ID	Staff Name	Address	TP No	Email	Gender	NIC	DOB	Designation	Basic Salary
S007	S007	Thiuthi	Jaffna	0733333333	thiuthi@gmail.com	female	942222222V	1994-02-27	Branch_head	20000

Figure D. 4: Staff Branch wise

Students' attendance report

System Administrator (director) to view students' attendance report by selected following steps:

Select branch → Select class → Select from date → select to date, after that details will be displayed.

Students Attendance Report

Branch:

Class:

From Date:

To Date:

Students' Attendance Report

Branch : Urumpirai	
Class : Playgroup(A)	
Attendance Period	
From	2017-10-01
To	2017-10-04

Student ID	Student Name	2017-10-01	2017-10-02	2017-10-03
STU0001	Santhos	1	1	1
STU0005	Anojan	1	1	0
STU0007	Anush	1	1	0

Figure D. 5: Student attendance report

Expense report

System Administrator (director), branch head and staff to view expense report by selected following steps:

Select branch → Select from date → select to date, after that details will be displayed.

Expense Report - Branch wise

Branch:

From Date:

To Date:

Expense Report

Branch: Urumpirai	
Expense Period	
From:	2017-07-01
To:	2017-07-29

Date	Expense ID	Expense For	Amount
2017-07-06	EXP0001	Stationary	250
2017-07-14	EXP0002	Printing	150
Total			400

Figure D. 6: Expense Report

Students' marks report

System Administrator (director), branch head and staff to view students' report by selected following steps:

Select branch → Select class → Select year → Select term, after that details will be displayed.

Students' Marks Report - Branch Wise

Branch:

Class:

Year:

Term:

Student ID	Student Name	Tamil	English	Average
STU0003	Akash	A	A	89.5
STU0008	Sajesan	A	A	98.5

Figure D. 7: Student marks branch wise

Staff's salary report

System Administrator (director) and branch head to view staffs' salary by selected following steps:

Select branch → Select year → Select month, after that details will be displayed.

Staffs' Salary

Branch: ▼

Year: ▼

Month: ▼

Staff ID	Name	Basic Salary	Additional Class	EPF	ETF	Total Earning	Total Deduction	Net Salary
S001	Gesha	10000	5000	800	200	15000	1000	14000
S002	Panchalingam	25000	5000	2000	500	30000	2500	27500
S003	Theethi	10000	0	800	200	10000	1000	9000
S004	Arijathaa	20000	0	1600	400	20000	2000	18000
S005	Kavitha	12000	0	960	240	12000	1200	10800
S006	Shanthi	10000	0	800	200	10000	1000	9000

Figure D. 8: Staff Salary

APPENDIX E - TEST RESULTS

In this online pre-school management system view appropriate error messages and success messages and information were displayed to notify the user about the activities.

General Forms


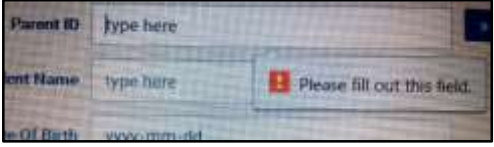
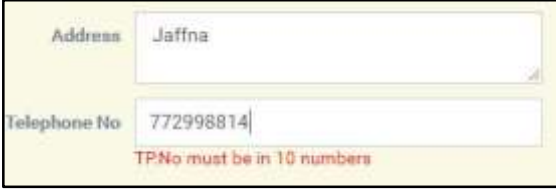


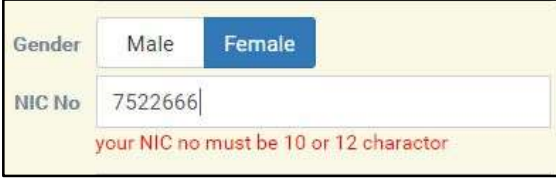
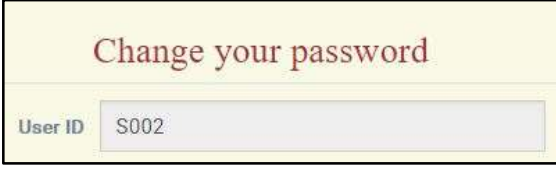
Test No	Testing step	Expected result	Status
1	Select after present date in Date of Birth field		Pass
2	Save without fill the field		Pass
3	Type phone number in incorrect format		Pass
4	Type email in incorrect format		Pass
5	Click the delete button		Pass
6	Type NIC number in incorrect format		Pass
7	Automatically show the user id when we change password		Pass

Table E. 1: Test Result

Common Test Cases





Test No	Testing step	Expected result	Status
1	Before enter user name and password		Pass
2	Enter correct user name and wrong password		Pass
3	Enter incorrect user name and incorrect password		Pass
4	Enter wrong Database name on config.php		Pass

Table E. 2: Common Test Result

Report Module

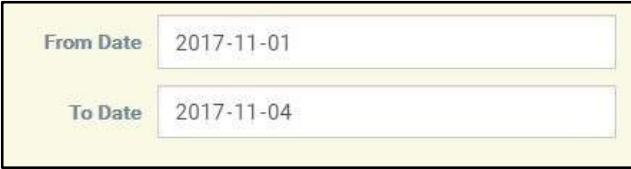

Test No	Testing step	Expected result	Status
1	Display date and time to view the report		Pass
2	Show the Print preview dialog box with print button		Pass

Table E. 3: Report Test

APPENDIX F - CODE LISTINGS

This section provides more detailed view of the coding to the users. Since the code section is very lengthy, only important section of the coding are provided below with appropriate comments to make the user more comfortable with the coding.

Form

HTML coding for create a form

```
<form action="" method="post" id="demo-form2" data-parsley-validate class="form-horizontal form-label-left">
```

```
    <div class="form-group">
```

```
        <label class="control-label col-md-3 col-sm-3 col-xs-12" for="msg_id"> Message ID </label>
```

```
        <div class="col-md-4 col-sm-6 col-xs-12">
```

```
            <?php
```

```
                //generate msg id
```

```
                $sql="SELECT msg_id
```

```
                FROM `messages`
```

```
                ORDER BY msg_id DESC";
```

```
                $result=mysql_query($sql)or die ("error in msg_id:".mysql_error());
```

```
                $row=mysql_fetch_assoc($result);
```

```
                $msg_id=$row['msg_id'];
```

```
                if(mysql_num_rows($result)>0){
```

```
                    $msg_id=++$msg_id;
```

```
                }
```

```
                else{
```

```
                    $msg_id="MG0001";
```

```
                }
```

```
                //end generate msg id
```

```
            ?>
```

```
            <!-- /load branch -->
```

```
            <input type="text" name="msg_id" id="msg_id" required="required" value="<?php echo $msg_id; ?>" readonly class="form-control col-md-7 col-xs-12" >
```

```
        </div>
```

```
    </div>
```

```

<div class="form-group">
  <label class="control-label col-md-3 col-sm-3 col-xs-12" for="msg_from">
Message From </label>
  <div class="col-md-4 col-sm-6 col-xs-12">
    <?php
      $Sql1="SELECT user_id,staff_name
        FROM staffs
        WHERE user_id='$user_id'";
      $result1=mysql_query($Sql1)or die ("error in sql1:".mysql_error());
      $row1=mysql_fetch_assoc($result1);
      $stf_name=$row1['staff_name'];
    ?>
    <input type="text" name="msg_from" id="msg_from" required="required"
value="<?php echo $stf_name; ?>" readonly class="form-control col-md-7 col-xs-12" >
  </div>
</div>

<div class="form-group">
  <label class="control-label col-md-3 col-sm-3 col-xs-12" for="msg_togrp">
Message To </label>
  <div class="col-md-4 col-sm-6 col-xs-12">
    <select name="msg_togrp" id="msg_togrp" class="form-control" required
onChange="getmsgto()">
      <option value="none" selected>Select The Person</option>
      <option value="Director">Director</option>
      <option value="Teacher">Teacher</option>
      <option value="Parent">Parent</option>
      <option value="Branch_head">Branch Head</option>
      <option value="Staff">Staff</option>
    </select>
  </div>
</div>

<div id="loadmsgto" class="form-group">

```

```

</div>

<div class="form-group">
  <label class="control-label col-md-3 col-sm-3 col-xs-12"> Message </label>

  <div class="col-md-4 col-sm-6 col-xs-12">
    <textarea name="msg" id="msg" class="resizable_textarea form-control"
placeholder="type here"></textarea>
  </div>
</div>

<div class="form-group">
  <label class="control-label col-md-3 col-sm-3 col-xs-12" for="stuyear"> Date
</label>

  <div class="col-md-4 col-sm-6 col-xs-12">
    <input type="text" name="date" id="date" required="required" value="<?php
echo date("Y/m/d");?>" class="form-control col-md-7 col-xs-12" readonly>
  </div>
</div>

<div class="form-group">
  <div class="col-md-4 col-sm-6 col-xs-12 col-md-offset-3">
    <button type="submit" name="btncsend" id="btncsend" class="btn btn-success
fa fa-save"> Send </button>
  </div>
</div>
</form>

```

Generate ID automatically

This coding automatically increase the Id in the Id field when add new detail. It is used in many id generate such as: generate student id, branch id, expense id, subject id, event id and class id.

<!-- generate staffid automatically start -->

```

<?php
    $sql1="SELECT staff_id
    FROM staffs
    ORDER BY staff_id DESC";
    $result1=mysql_query($sql1)or die ("error in staff id:".mysql_error());

```

```

        $row=mysql_fetch_assoc($result1);
        $staff_id=$row['staff_id'];
        if(mysql_num_rows($result1)>0){
            $staff_id=++$staff_id;
        }
        else{
            $staff_id="S001";
        }
    ?>
    <!-- /generate staffid automatically start -->

```

Email Validation

This coding contain the email format, if we type wrong format it display error message.

```

//validation for email
function checkemail(){
    var email=document.getElementById("email").value;
    var emailformat=/^([a-zA-Z0-9_\.|-]+)@+((([a-zA-Z0-9\-.]+\.)+([a-z0-9A-Z]{2,4})+)$);
    if (email.match(emailformat)){
        document.getElementById("emailerrmsg").innerHTML="";
    }
    else{
        document.getElementById("emailerrmsg").innerHTML="Invalid Email";
        //document.getElementById("email").focus();
    }
}
// end validation for email

```

Edit/Update Record

Update coding helps to update after edit the records, following pages include this coding such as: edit student detail, edit staff detail, edit class details, edit subject details, edit event details, edit parent details and edit profile.

```

if(isset($_POST['btnupdate']))

```



```

{
    $expense_id=$_POST['expense_id'];
    $sqlupdate="UPDATE expenditures SET
        expense_for=".mysql_real_escape_string($_POST['expense_for']).",
        date=".mysql_real_escape_string($_POST['date']).",
        amount=".mysql_real_escape_string($_POST['amount']).",
        status=".mysql_real_escape_string($_POST['status']).",
        branch_id=".mysql_real_escape_string($_POST['branch_id'])."
        WHERE expense_id='$expense_id'";
    $resultupdate=mysql_query($sqlupdate) or die("error in update
    expenditures part:".mysql_error());
    if($resultupdate){
        header('location:index.php?pg=expenditures.php&option=new');
    }
}

```

Call a function to get information (get student detail-class wise)

```
<head>
```

```
<script>
```

```

function getstudentdetailclasswise(){
    var class_type=document.getElementById("class_type").value;
    var year=document.getElementById("year").value;
    var branch_id=document.getElementById("branch_id").value;

    var xmlhttp=new XMLHttpRequest();
    xmlhttp.onreadystatechange=function(){
        if(xmlhttp.readyState==4 & xmlhttp.status==200){
            document.getElementById("loadstudentdetailclasswise").innerHTML=xmlhttp.responseText;
        }
    }
}

```

```

    }
}

xmlhttp.open("GET","reportajaxpage.php?option=loadstudentdetailclasswise&class_type="+class_type+"&year="+year+"&branch_id="+branch_id,true);

xmlhttp.send();

}

</script>
</head>

```

View report

This coding helps to view report in particular page.

```

//start studentdetail class wise-----
else if($_GET['option']=="loadstudentdetailclasswise"){
    $class_type=$_GET['class_type'];
    $year=$_GET['year'];
    $branch_id=$_GET['branch_id'];

    $stuvievwclzwise="SELECT S.student_id, S.student_name, S.dob, S.gender, S.parent_id,
S.registered_date, S.branch_id

        FROM students as S , student_class_maps as SCM , classes as C

        WHERE S.student_id=SCM.student_id

        AND SCM.class_id=C.class_id

        AND C.class_type='$class_type'

        AND S.branch_id='$branch_id'

        AND SCM.year='$year'";

    $stuvievwresultclzwise=mysql_query($stuvievwclzwise)or die("error in
stuvievwclzwise part:".mysql_error());

?>
<div class="x_title" >
<div class="row">
    <div class="col-md-9">

```



```

        </table>
    </div>
</div>
</div>
<?php
    echo'<table id="datatable" class="table table-striped table-bordered">
        <thead>
            <tr>
                <th> Student ID </th>
                <th> Student Name </th>
                <th> DOB </th>
                <th> Gender </th>
                <th> Parent ID </th>
                <th> Registered Date </th>
                <th> Branch ID </th>
            </tr>
        </thead>
        <tbody>';
        while($row5=mysql_fetch_assoc($stuvviewresultclzwise))
        {
            echo'<tr>
                <td>'.$row5['student_id'].'</td>
                <td>'.$row5['student_name'].'</td>
                <td>'.$row5['dob'].'</td>
                <td>'.$row5['gender'].'</td>
                <td>'.$row5['parent_id'].'</td>
                <td>'.$row5['registered_date'].'</td>
                <td>'.$row5['branch_id'].'</td>';
            echo'</tr>';
        }
        echo'</tbody>

```

```

</table>;

if(!isset($_GET['pr'])){//if get print

echo '<hr> ';

echo '<a href="print.php?pr=reportajaxpage.php&option=loadstudentdetailclasswise&class_type='.$class_type.'&year='.$year.'&branch_id='.$branch_id.'" target="_blank">
Print</a>';

</a>;

} // end print }

//end studentdetail class wise -----

```

Log out

In this coding `session_destroy()`; destroys all of the data.

```

<?php

include('config.php');

if (!isset($_SESSION)){

session_start();

}


session_destroy();

header('location:index.php');

?>

```

APPENDIX G - CLIENT CERTIFICATE



Urumpirai Kids' Park

Model Pre-school
Do Right Be Bright
உரும்பிராய் கிட்ஸ் பார்க்
A nonprofit making service organization

GNANAVIRAVA SWAMY KOVIL ROAD,
URUMPIRAI

30/10/2017

BIT Coordinator,
University of Colombo School of Computing,
Colombo – 07.

Dear Sir/Madam,

LETTER OF CLARIFICATION

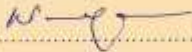
This is to clarify that Miss. Sangeetha.Ariyathan successfully designed and developed an online Pre-school management system for Urumpirai Kids' Park. The project was undertaken by her partial fulfillment of a requirement for the Bachelor of Information Technology Degree Program.

The system fully satisfies with our requirements and this system, web based management system would be useful to us and parents. From this system we can manage students, staff, students' marks, students' attendance and parents can be able to view their kid's details, marks and events. This can enhance our relationship with parents. I have pleasure to commend that the system developed by Miss.Sangeetha.Ariyathan fulfills the requirements of the institution.

This commendation is issued at the request of Miss.Sangeetha.Ariyathan.

Thanking you.

Yours faithfully,


.....

URUMPIRAI KIDS' PARK
URUMPIRAI

President : A. Panchalingam	021-223 0754 / 077 656 5258
Secretary : M. Sriharan	077 711 0186
Treasurer : (Ms) M.Selvukumar	077 9906271

GLOSSARY

Apache - is an open source web server. Mostly for UNIX, Linux and Solaris platforms.

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.

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

Domain - is knowledge about the environment in which the target system operates.

Firewall - is a solution that is used to enforce security policies. This can be a Hardware or Software.

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

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

Iterative and Incremental Development - is a cyclic approach to software development in which activities are repeated in a structured manner and the software is developed in increments.

JavaScript - is a prototype-based, object-oriented client-side scripting language that is dynamic. It is also considered as a functional programming language.

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

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

PHP Hyper-text Pre-processor (PHP) - is a server-side programming language.

Rapid Application Development (RAD) - is a software development methodology that uses minimal planning in favor of rapid prototyping.

Structured Query Language - is a database computer declarative language designed for managing data in relational database management systems (RDBMS).

Unified Modeling Language (UML) - is a standardized general-purpose modeling language in the field of object-oriented engineering. This includes a set of graphic

notation techniques to create visual models of object-oriented software-intensive systems.

Web Browser - is a software which allows the user to access WWW.

WWW (World Wide Web) - is a system of interlinked hypertext documents accessed via the Internet.