**TABLE OF CONTENT**

CONTENT PAGES

DECLARATION………………………………………………………………………...I

CERTIFICATION………………………………………………………………….........II

DEDICATION………………………………………………………………………......III

ACKNOWLEDGEMENT………………………………………………………………IV

TABLE OF CONTENT…………………………………………………………………V

ABSTRACT……………………………………………………………………….……X

**CHAPTER ONE – INTRODUCTION**

1.0 BACKGROUND OF THE STUDY……………………………………………………….1

1.1 STATEMENT OF THE PROBLEM……………………………………………………….1

1.2 MOTIVATION………………………………………………………………..2

1.3 AIMS AND OBJECTIVE………………………………………………………2

1.4 PURPOSE OF THE STUDY……………………………………………………3

1.5 SIGNIFICANCE OF THE STUDY…………………………………………….3

1.6 ORHANISATION OF THE WORK……………………………………………4

1.7 ORHANISATION OF THE WORK……………………………………………4

**CHAPTER TWO – LITERATURE REVIEW**

2.0 INTRODUCTION………………………………………………………………6

2.1 RECORDS………………………………………………………………………6|

2.2 PATIENTS………...……………………………………………………………7

2.3 EXISTING SYSTEM …………………………………………………………….7

2.4…………………………………………………………………………………….

2.3.3 PHP…………………………………………………………………………………….

2.3.4 WHAT DOES PHP CODE LOOK LIKE?.........................................................

............

2.3.5 BASIC PHP SYNTAX…………………………………………………………………

2.3.6 ESCAPING FROM HTML…………………………………………………………….

2.3.7 INSTRUCTION SEPARATION……………………………………………………….

2.3.8 WHAT CAN PHP DO?...................................................................................................

2.3.9 THE THREE MAIN AREAS WHERE PHP SCRIPTS ARE USED………………….

2.3.10 PHP SECURITY……………………………………………………………………...

2.3.11 MYSQL (DBMS)…………………………………………………………………….

2.3.12 HIGH AVAILABILITY……………………………………………………………...

2.3.13 ROBUST TRANSACTIONAL SUPPORT……………………………………………

2.3.14 WEB AND DATA WAREHOUSE STRENGHTS……………………………………

2.3.15 STRONG DATA PROTECTION………………………………………………………

2.3.16 COMPREHENSIVE APPLICATION DEVELOPMENT…………………………….

2.3.17 MANAGEMENT EASE………………………………………………………………

2.3.18 APACHE SERVER……………………………………………………………………

2.3.19 PERFORMANCE OF APACHE SERVER……………………………………………

**CHAPTER THREE – METHODOLOGY AND SYSTEM ANALYSIS**

3.1 INTRODUCTION…………………………………………………………………………

3.2 SOFTWARE DEVELOPMENT LIFE CYCLE……………………………………

3.2.1 WATERFALL MODEL…………………………………………………………………

3.2.2 PROTOTYPE MODEL………………………………………………………………….

3.2.3 V-SHAPED MODEL……………………………………………………………………

3.2.4 ITERATIVE AND INCREMENTAL MODEL…………………………………………

3.2.5 AGILE MODEL…………………………………………………………………………

3.3 THE AGILE PROCESS FLOW…………………………………………………………

3.4 BENEFITS OF ADHERING TO SOFTWARE DEVELOPMENT METHODOLOGIES……

3.5 SOFTWARE METHODOLOGY USED IN DEVELOPING THE SYSTEM…………………

3.6 ANALYSIS OF CURRENT SYSTEM…………………………………………………………

3.7 LAPSES WITH CURRENT SYSTEM…………………………………………………………

3.8 OBSERVATION……………………………………………………………………………….

3.9 EFFORT DISTRIBUTION DIAGRAM (EDD)……………………………………………….

3.10 ENTITY RELATIONSHIP DIAGRAM……………………………………………………...

3.11 DATA FLOWCHART DIAGRAM (DFD)………………………………………………….

3.12 USE CASE DIAGRAM………………………………………………………………………

3.15 SYSTEM REQUIREMEMT………………………………………………………………....

3.15.1 HARDWARE REQUIREMENT……………………………………………………………

3.15.2 SOFTWARE REQUIREMENT………………………………………………………….

3.16 CONCLUTION………………………………………………………………………………

**CHAPTER FOUR – DESIGN, TESTING AND IMPLEMENTATION**

4.1 DESIGN…………………………………………………………………………………….

4.2 SYSTEM MODULE ARCHITECTURE…………………………………………………...

4.2.1 STUDENT…………………………………………………………………………………

4.2.2 H.O. D……………………………………………………………………………………

4.2.3 HALL MASTER………………………………………………………………………….

4.2.4SECURITY……………………………………………………………………………….

4.2.5 ADMIN…………………………………………………………………………………….

4.3 IMPLEMENTATION……………………………………………………………………….

4.4 CHOICE OF PROGRAMMING LANGUAGE…………………………………………….

4.5 DATABASE FILE DESIGN…………………………………………………………………

4.6 INPUT DESIGN………………………………………………………………………………

4.7 OUTPUT…………………………………………………………………………………….

4.8 SYSTEM TESTING………………………………………………………………………….

4.9 TESTING METHODS……………………………………………………………………….

4.10 INSTALLATION AND TESTING SCREENSHOOTS……………………………………

**CHAPTER FIVE – SUMMARY, RECOMMENDATION AND CONCLUTION**

5.1 SUMMARY…………………………………………………………………………………

5.2 RECOMMENDATIONS…………………………………………………………………….

5.3 CONCLUSION………………………………………………………………………………

REFERENCE……………………………………………………………………………………

APPENDIX(SOURCE CODE)………………………………………………………………….

**LIST OF FIGURES**

**Figures Page**

Fig1

Fig2

Fig 3

Fig 4

Fig 5

Fig 6

Fig 7

Fig 8

Fig 9

Fig 10

Fig 11

Fig 12

Fig 13

Fig 14

Fig 15

Fig 16

Fig 17

Fig 18

**ABSTRACT**

Exeat as a term means permission from a college, boarding school, or other institution for temporary absence. It is probably known widely as a “pass”, “exit permission” or “leave of absence”. In an academic environment, it is implemented to prevent un-authorized exits and to keep track of student movement in relation to time and destination of travel.

This paper presents the design and development of an electronic exeat system which automates the manual system which is known to be cumbersome, challenging, difficult, unsecure and expensive. The case study is Bingham University, a privately owned institution of higher learning. The software developed proved to be very efficient for the school administrators. In our case study institution, the current manual method of exeat scheduling and authentication has proven to be challenging, difficult, unsecured and expensive due to the following reasons:

1. The paper-based manual system promotes a recurrent waste of scarce resources which could be directed to other productive activities.
2. Difficulty in analyzing exeats and authorization on the part of the Hall Administrators.
3. Data could be somewhat unreliable regarding date and time of proposed exeat.
4. The issue of impersonation.