COMPUTER UNIVERSITY (MANDALAY)

FINAL YEAR PROJECT REPORT

ON

ESTIMATING THE EFFORT COST REQUIRED FOR A SOFTWARE PROJECT

Bachelor of Computer Science (B. C. Sc.)

Presented by Group-15

2014-2015

CONTENTS

Acknowledgements

We would like to express our deepest gratitude and sincere appreciation to the following persons who have supported directly or indirectly towards the success of this project and helped make this dissertation possible.

We would respectfully like to thank Dr. Win Aye, the Rector of University of Computer Studies (Mandalay), for her kind permission to carry out this project, general guidance and workable environment during the period of study.

We would like to enthusiastically thank my supervisor Daw Thein Thein, Lecturer, Information Science Department, University of Computer Studies (Mandalay), for her valuable guidance, encouragement, superior suggestions and supervision on the accomplishment of this thesis.

Our sincere thanks go to Dr. Zar Zar Linn, Associated Professor, and Software Department, University of Computer Studies (Mandalay) for her administrative support, recommendations and suggestions.

We are also thankful to Daw Win Min Soe, Lecturer, English Department,University of Computer Studies (Mandalay) for editing this project form the English language point of view.

Thanks are also extended especially to all our teachers of University of Computer Studies (Mandalay) for their help in providing the necessary for our project and presentation.

Group Member List

Sr.No	Name	Roll No.
1	Ma Thet Wai Aung	4CS-18
2	Ma Thet Myat Noe	4CS-66
3	Mg Ye Htet Aung	4CS-108
4	Ma May Thu Soe	4CS-170

Supervisor

Name : : Daw Thein Thein

Rank : : Lecturer

Department : : Information Science Department

Computer University (Mandalay)

Project Schedule

Project Proposal : : Estimating the Effort Cost Required For a Software Project

First Seminar : : 9.6.2015

Second Seminar : : 14.7.2015

Third Seminar : : 18.8.2015

Time Schedule	March 2015	May 2015	June 2015	July 2015	August 2015
Project Proposal					
First Seminar					
Second Seminar					
Third Seminar					
Book Submission					

Abstract

Software development effort estimation is one of the most major activities in software project management.Algorithmic cost modelling uses a mathematical formula to predict project costs based on estimates of a project sizes, other process and product factors.The project will calculate software effort cost estimation based on COCOMO2 model of Early Design Model.Early Design Model used during early stages of the system design after the requirements have been established.Estimates are based on function points, which are then converted to a number of lines of source code. The formula follows the standard form with a simplified set of seven multipliers.The project will be implemented with PHP programming language.

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Introduction

The project presents a program using algorithmic cost modelling. A number of models have been proposed to construct a relationship between software size and effort. However there are many problems. Algorithmic cost modelling is primarily used to make estimates of software development costs.COCOMO model is empirical models that are derived by collecting data from software project to calculate the cost of a particular project initially.The effort cost of the software development is estimated based on existing project.Estimates can be made after the requirements have been agreed.

1.2 Objective of the Project

- To achieve the estimate effort cost
- To get easy and fast calculation when effort cost is estimated to develop of new software project
- To assist project manager in taking best decision for a project

1.3 Project Requirements

1.3.1 Software Requirements

1. Microsoft Word 2010
2. Xampp database
3. PHP

1.3.2 Hardware Requirements

1. Processor-Pentium P4 1.8GHz
2. Memory- 4GB DDR3
3. HDD-500GB

CHAPTER 2

THEORY BACKGROUND

2.1 Estimation Techniques

2.1.1 Algorithmic Cost Model

Algorithmic cost modeling uses a mathematical formula to predict project costs based on estimates of the project size, the number of software engineers, and other process and product factors. Algorithmic cost models are primarily used to make estimates of software development costs.Algorithmic cost modeling is one of cost estimated techniques to make software effort and cost estimates.

Algorithmic cost model estimates the effort based on number of line of source code as code size. As the size of the software increases, extra costs are incurred because of the communication overhead of larger teams, more complex configuration management, more difficult system integration. Therefore, the larger the system, the larger the value of this exponent.

2.1.2 Expert Judgment

Several experts on the proposed software development techniques and the application domain are consulted. They each estimate the project cost. The experts provide estimates using their own methods and experience.

2.1.3 Estimation by Analogy

This technique is applicable when other objects in the same application domain have been completed. The cost of a new project is estimated by analogy with these completed projects. Reference gives a very clear description of this approach.

2.1.4 Parkinson's Law

Parkinson's Law states that work expands to fill the time available. The cost is determined by available resources rather than by objective assessment. If the software has to be delivered in 12 months and 5 people are available, the effort required is estimated to be 60 person-months.

2.1.5 Price-to-win

The software cost is estimated to be the best price to win the project. The estimation is based on the customer's budget instead of the software functionality. For example, if a reasonable estimation for a project costs 100 person-months but the customer can only afford 60 person-months, it is common that the estimator is asked to modify the estimation to fit 60 person months' effort in order to win the project. This is again not a good practice since it is very likely to cause a bad delay of delivery or force the development team to work overtime.

2.2 COCOMOModel (Constructive Cost Model)

The COCOMO model is an algorithmic cost estimation model developed by Barry Boehm. The model uses a basic regression formula, with parameters that are derived from historical project data and current project characteristics. COCOMO model is an empirical model that is derived by collecting data from software project.

The first version of COCOMO model (COCOMO81) was a three-level model where the levels corresponded to the detail of the analysis of the cost estimate. The first level provided an initial rough estimate; the second level modified this using a number of project and process multipliers; and the most detailed level produced estimates for different phases of the project. COCOMO 81 assumed that the software is developed according to the waterfall process using standard programming language.COCOMO81 through various instantiation to COCOMOII.

2.2 Introduction to COCOMOII

The COCOMO models are comprehensive, with a large number of parameters that can take a range of values. There are several COCOMO models:

a) Basic COCOMO

The basic COCOMO model is simple and easy to use. As many cost factors are not considered, it can only be used as a rough estimate.

b) Intermediate COCOMO and Detailed COCOMO

While both basic and intermediate COCOMOs estimate the software cost at the system level, the detailed COCOMO works on each sub-system separately and has an obvious advantage for large systems that contain non-homogeneous subsystems.

c) COCOMOII

COCOMOII is a model that allows one to estimate the cost, effort, and schedule when planning a new software development activity. COCOMOII is useful for a much wider collection of techniques and technologies. COCOMOII includes the four sub-models: the application- composition model, the early design model, the reuse model, and the post- architecture model.

2.3.1 The Application-Composition Model

The application-composition model is introduced into COCOMOII to support the estimation of effort required for prototyping projects and for projects where the software is developed by composing existing components. Software size estimates are based on application point, and a simple size / productivity formula is used to estimate the effort required.

Programmerproductivity also depends on the developer's experience and capability. The formula for effort computation for system prototypes is:

PM= (NAP*(1-%reuse/100))/PROD

PM is the effort estimate in person months. NAP is the total number of application points in the delivered system. %reuse is an estimate of the amount of reused code in the development. PROD is the object point productivity.

As another definitional point, note that the use of the term object points are NOT the same as object classes. The number of object-points in a program is considered as a weighted estimate of 3 elements:

- The number of separate screens that are displayed
- The number of reports that are produced by the system

- The number of 3GL modules that must be developed to supplement the 4GL code

The counting procedure for object point is the following:

- Assess object-counts in the system: number of screens, reports, and 3GL.
- Assess complexity level for each object: simple, medium and difficult.

2.3.2 The Early Design Model

An early design model is used during early stages of the system design after the requirements have been established. Estimates are based on function points, which are converted to number of lines of source code.

The standard formula for algorithmic models, namely:

$$Effort = A * Size^{B} * M$$

Boehm proposes that the coefficient A should be 2.94. The size of thesystem is expressed in KSLOC, which is the number of thousands of line of source code. KSLOC can calculate by estimating the number of function point in the software. The exponent B reflects the increased effort required as the size of the project increases. This is not fixed for different types of systems, as in COCOMO 81, but can vary from 1.1 to 1.24 depending on the novelty of the project.

The multiplier M in COCOMO II is based on a simplified set of seven project and process characteristics that influence the estimate. These can increase or decrease the effort required. These characteristics used in the early design model are:

- PERS= Personal Capability
- RCPX= Product Reliability and Complexity
- RUSE= Reuse Required

- PDIF= Platform Difficulty
- PREX= Personal Experience
- FCIL= Support Facilities
- SCED= Schedule

The result in an effort computation as follows:

$$PM=2.94 * Size^{B} * M$$

Where:

M=PERS* RCPX * RUSE* PDIF * PREX * FCIL * SCED

2.3.3 The Reuse Model

A reuse model is used for effort to integrate reusable components or automatically generated code. COCOMOII considers reused code to be of two types. Black-box code is code that can be reused without understanding the code or making changes to it. The development effort for black-box code is taken to be zero. Code that has to be adapted to integrate it with new code or other reused components is called white-box code.

2.3.4The Post-Architecture Model

The post- architecture model is the most detailed of the COCOMOII model. A post-architecture model is used for development effort based on system design specification. It is based on number of lines of source code.

The estimate produce as the post-architecture level is based on the same basic formula ($PM=A * Size^{B} * M$) used in the early design estimates. However, the size estimate for the software should be more accurate by this state in the

estimation process. In addition, a much more extensive set of product,process and organizational attributes (17 rather than 7) are used to refine the initial effort computation. It is possible to use more attributes at this stage.

CHAPTER 3

DESIGN AND IMPLEMENTATION

3.1 System Flow Diagram

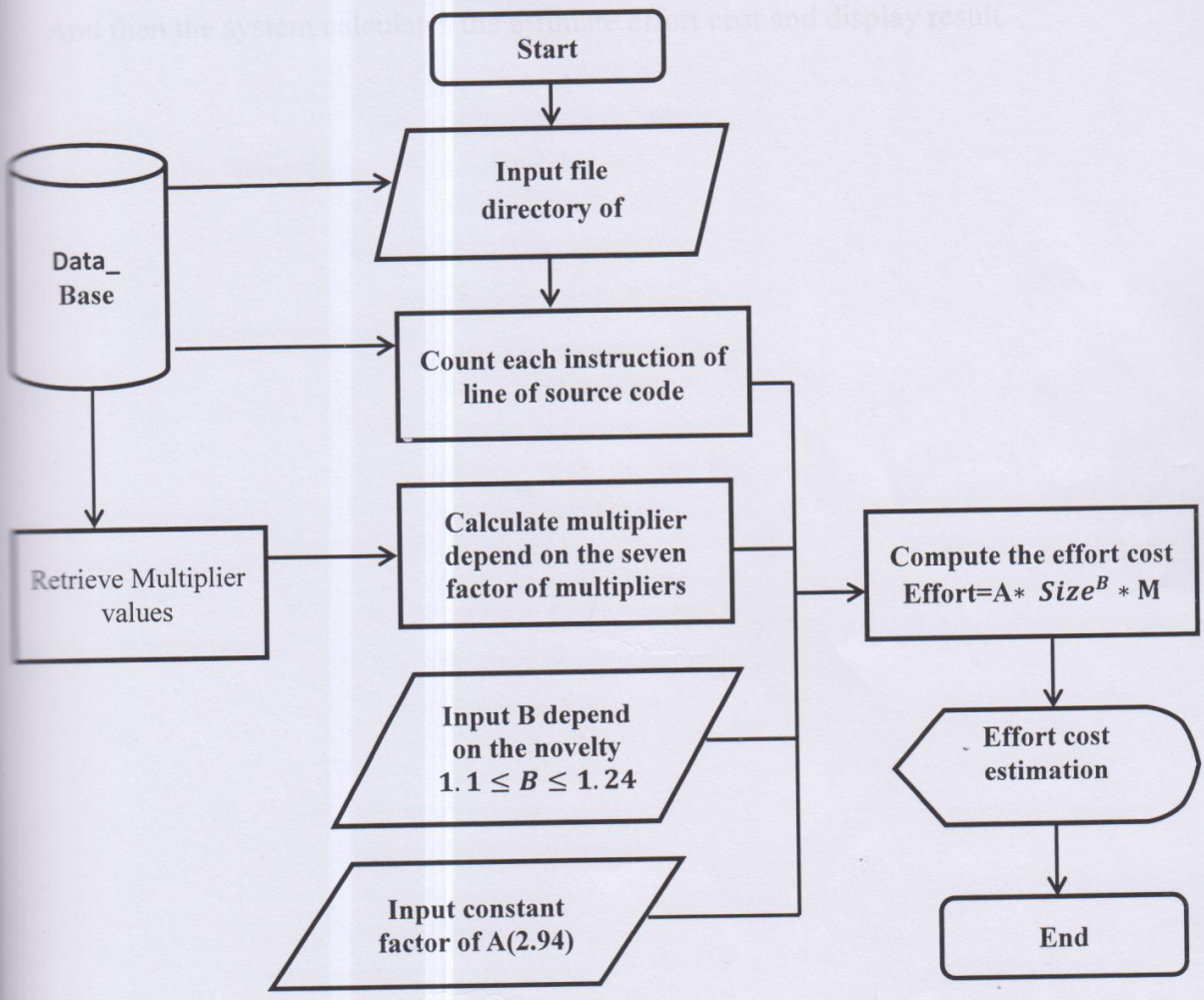

Figure 3.1: System Flow Diagram

In figure3.1, the user enters the file directory of the input existing java program.Then the system counts the number of line of source code.Then the user chooses the multiplier values depended on the seven factors of multiplier and then system calculate the multiplier value (M).The user enters the value B depended on the novelty (between 1.1 and 1.24) and constant factor A (2.94). And then the system calculates the estimate effort cost and display result.

3.2 Use Case Diagram

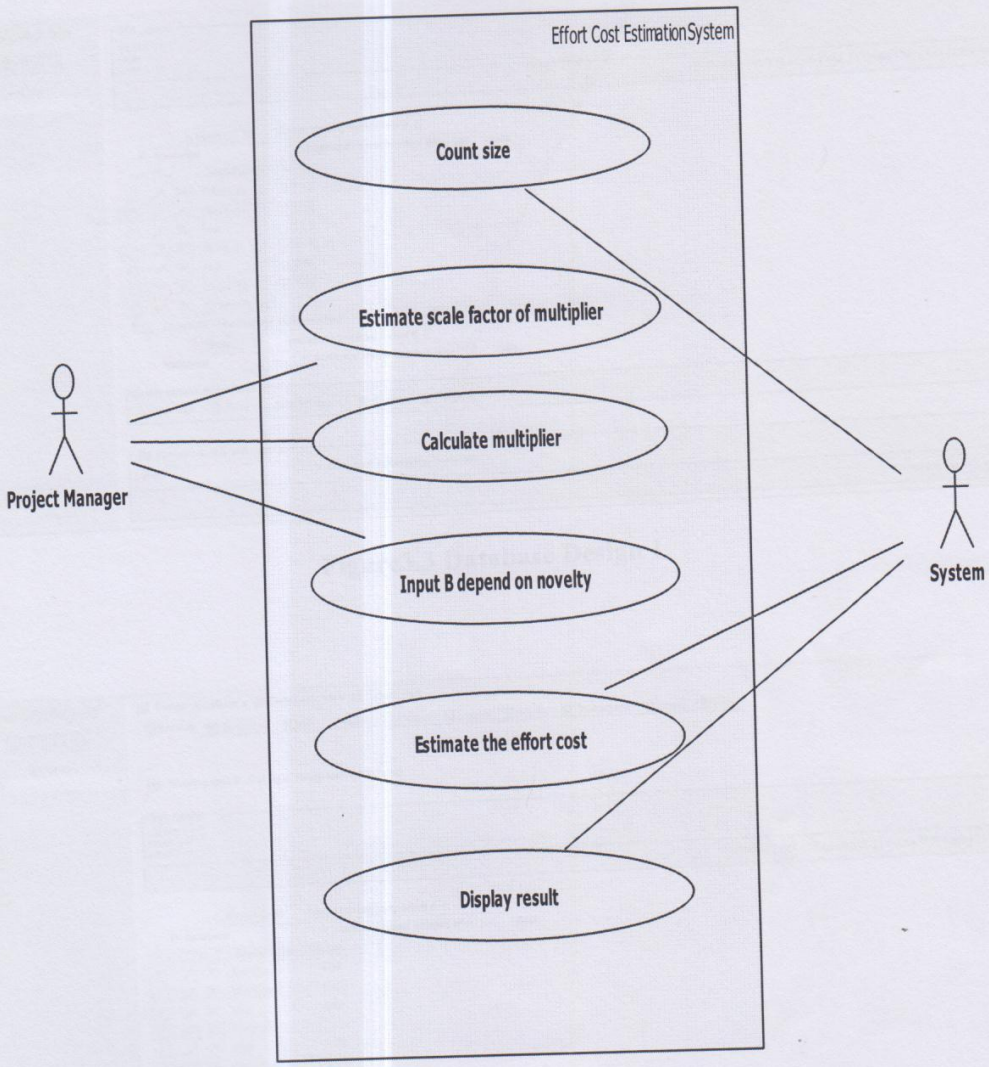

Figure 3.2 Use Case Diagram of the system

3.3 Database Design

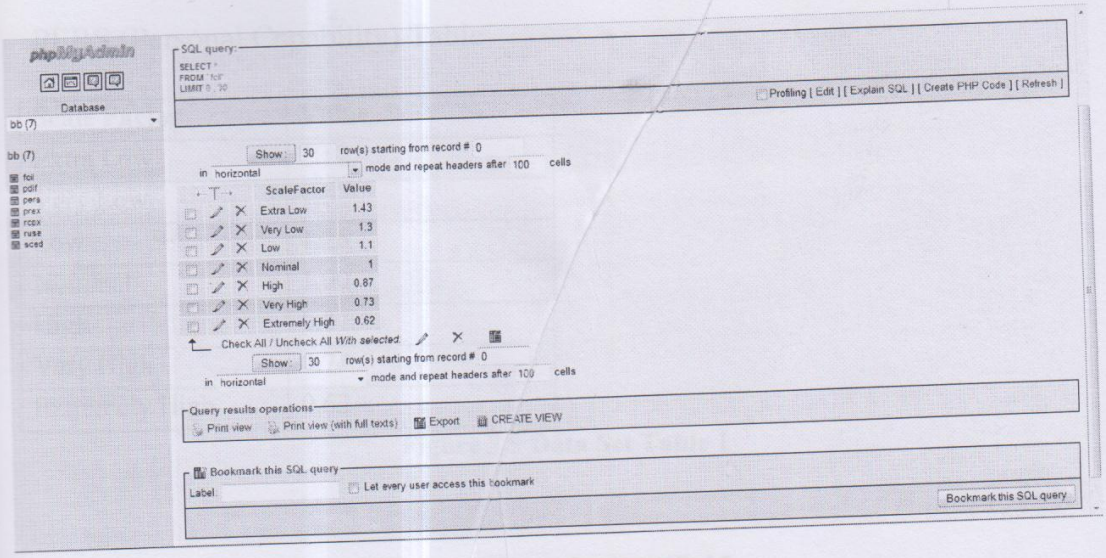

Figure3.3 Database Design 1

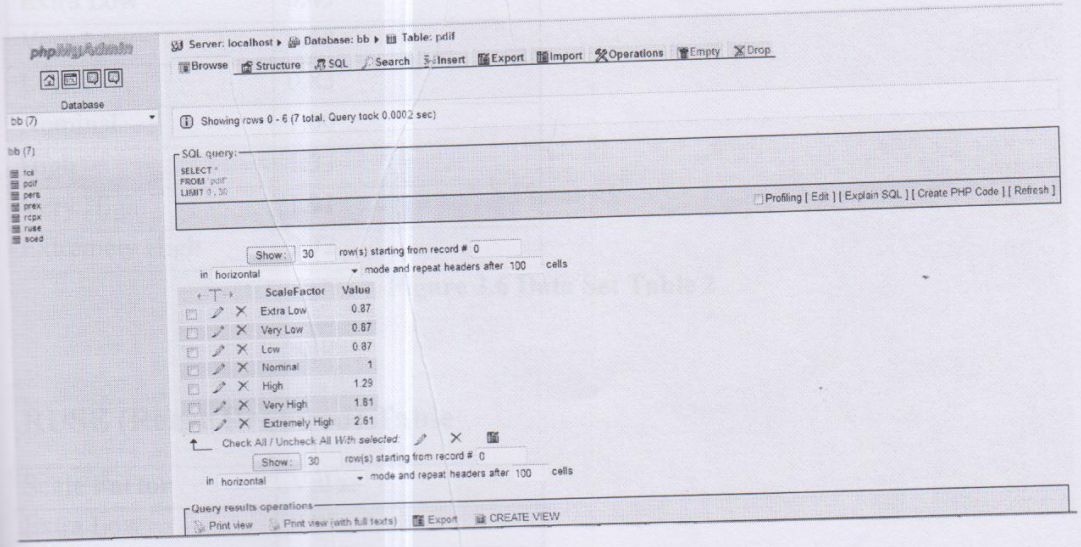

Figure 3.4 Database Design 2

3.4 Data Set Tables

PERS (Personal Capability) Table

Scale Factor	Value
Extra Low	2.12
Very Low	1.62
Low	1.26
Nominal	1.00
High	0.87
Very High	0.74
Extremely High	0.62

Figure 3.5 Data Set Table 1

RCPX (Product Reliability and Complexity) Table

Scale Factor	Value
Extra Low	0.49
Very Low	0.60
Low	0.83
Nominal	1.00
High	1.33
Very High	1.91
Extremely High	2.72

Figure 3.6 Data Set Table 2

RUSE (Reuse Required) Table

Scale Factor	Value
Extra Low	0.95
Very Low	0.95
Low	0.95
Nominal	1.00
High	1.07
Very High	1.15
Extremely High	1.24

Figure 3.7 Data Set Table 3

PDIF (Platform Difficulty) Table

Scale Factor	Value
Extra Low	0.87
Very Low	0.87
Low	0.87
Nominal	1.00
High	1.29
Very High	1.81
Extremely High	2.61

Figure 3.8 Data Set Table 4

PREX (Personal Experience) Table

Scale Factor	Value
Extra Low	1.59
Very Low	0.60
Low	1.22
Nominal	1.00
High	0.87
Very High	0.74
Extremely High	0.62

Figure 3.9 Data Set Table 5

FCIL (Support Facilities) Table

Scale Factor	Value
Extra Low	1.43
Very Low	1.30
Low	1.10
Nominal	1.00
High	0.87
Very High	0.73
Extremely High	0.62

Figure 3.10 Data Set Table 6

SCED (Schedule) Table

Scale Factor	Value
Extra Low	1.00
Very Low	1.00
Low	1.00
Nominal	1.00
High	1.14
Very High	1.43
Extremely High	1.43

Figure 3.11 Data Set Table 7

3.5 Implementation of the Project

The project is implemented by using PHP programming language and Xampp Database. Xampp Database includes PERS (Personal Capability), RCPX (Product Reliability and Complexity),RUSE(Reuserequired), PDIF(Platform Difficulty), PREX (Personal Experience), FCIL (Support Facilities), and SCED (Schedule).

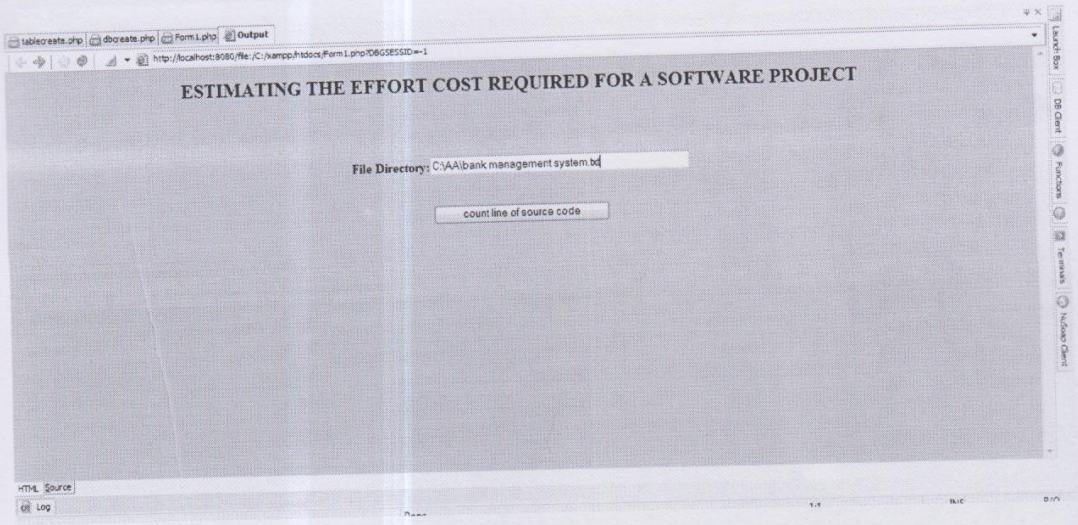

Figure 3.12 Input Form (Directory of the file)

In figure 3.12, the user enters the file directory of existing java program and then 'CountLine of source code' button is used to calculate the size of the program.

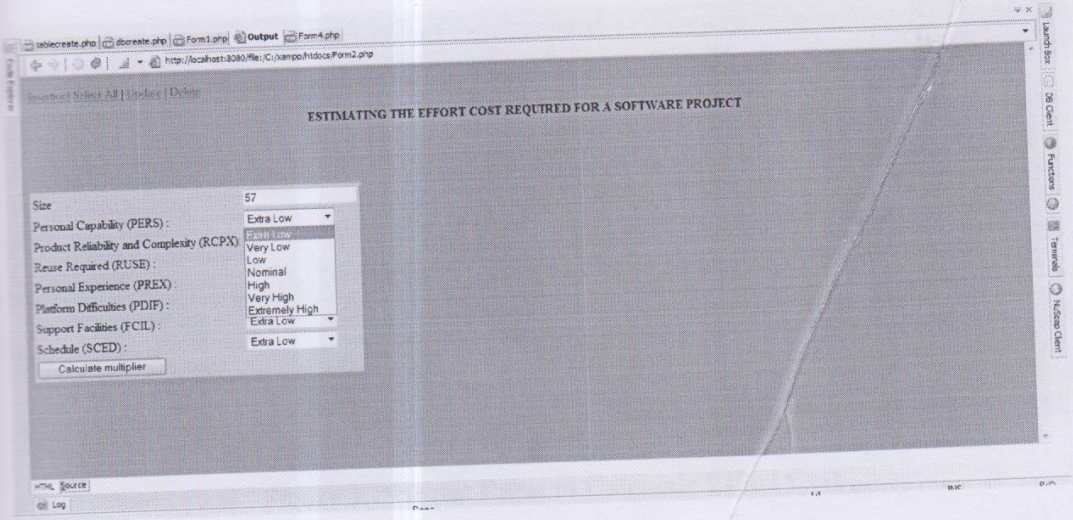

Figure 3.13 Choose Form

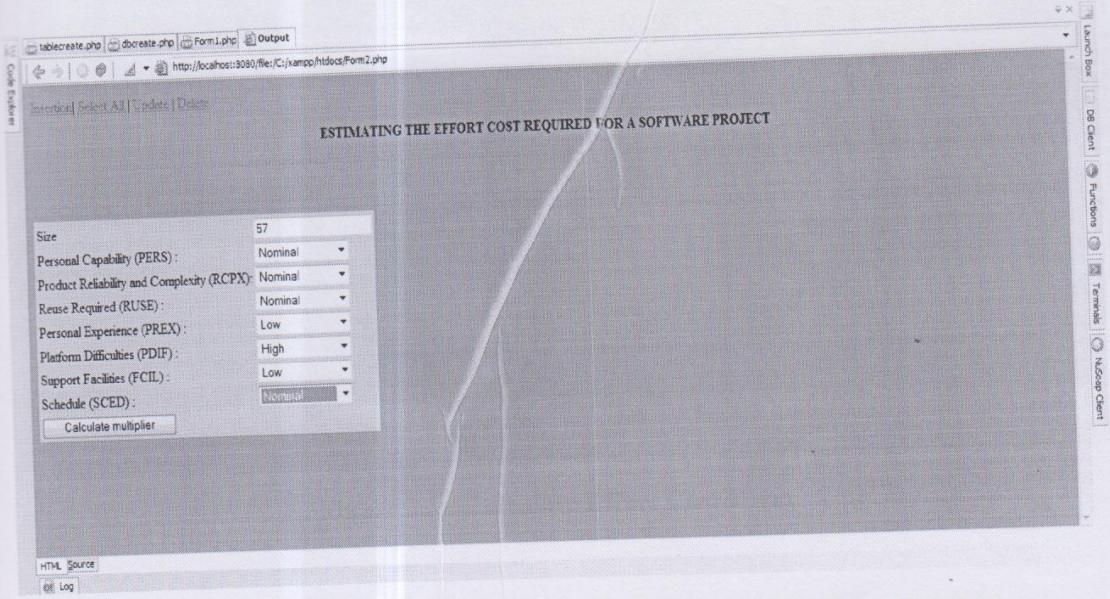

Figure 3.14 Display Size result and Choose Form of seven factors of Multiplier

In figure 3.13 and figure3.14, the user chooses the 'Count line of source code' button in Form1 the system itself calculates the program size and display in Form2. Then the user chooses the seven factors of multiplier and then click 'Calculate multiplier' button to implement next process.

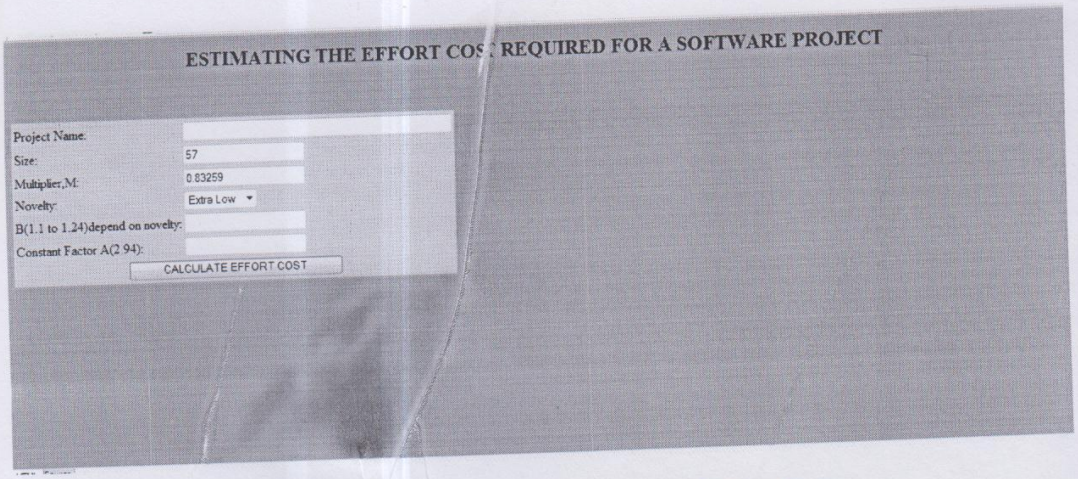

Figure 3.15 Input Form to Calculate Effort Cost

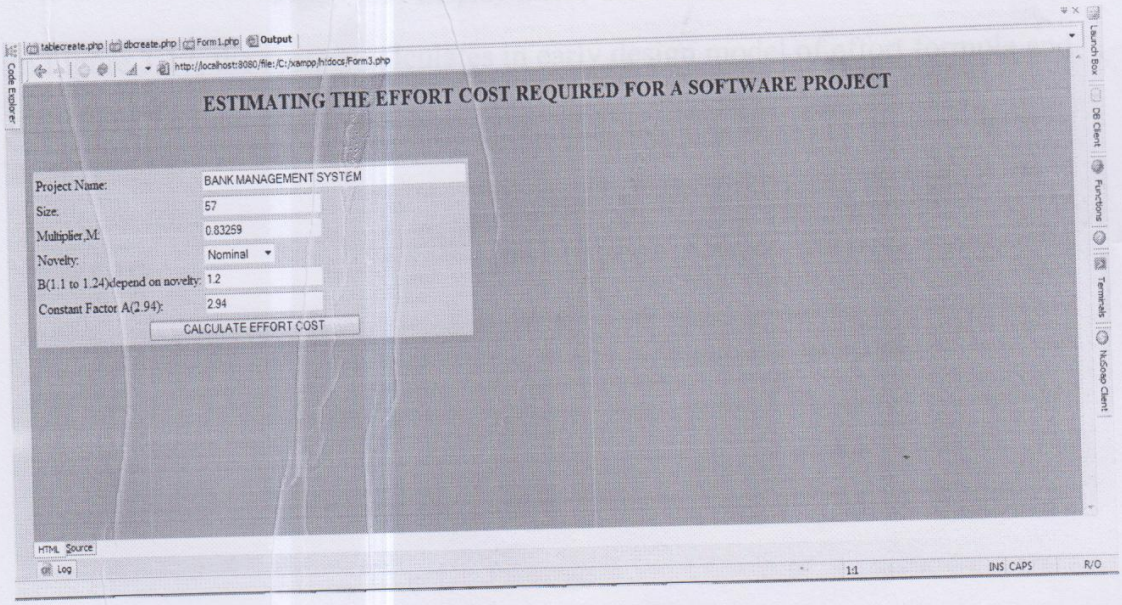

Figure 3.16 Calculate Effort Cost Form

In Figure 3.15 and figure 3.16, the user input the project name, value B depend on the novelty, the constant factor of A. Then the user click 'CALCULATE EFFORT COST' button to calculate the total effort cost.

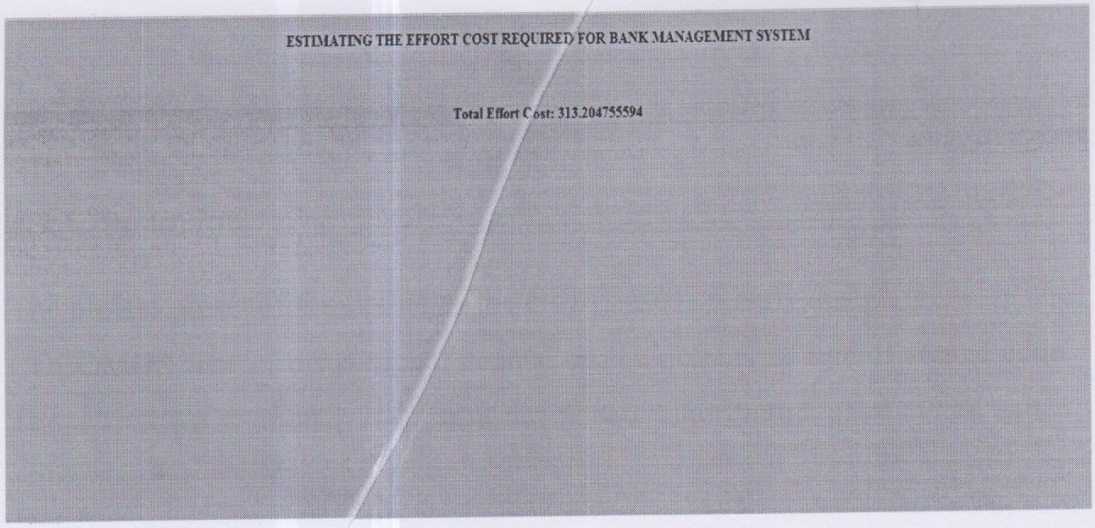

Figure 3.17 Display Effort Cost Result Form

In Figure 3.17, the system calculates in early design model of effort formula and displayed the total effort cost.

CHAPTER 4

CONCLUSION

4.1 Conclusion

The project present formulating of the effort based on early design model of the COCOMO2. The effort compute depend on the number of line of source code. The COCOMO2 model is a well-developed algorithmic cost model when estimating the effort.

The prediction of software development costs is to make the good management decisions and determine how much effort cost of a project required for project managers as well as system analysts and developers.

In the project, the existing project is used for the case study to count the lines of source code.

4.2 Advantages of the Project

The system can estimate development effort cost required.The system can calculate quickly effort cost estimation to develop of software project.

4.3 Further Extension and Limitation

The system can further estimate the software development cost using COCOMO2 of methods. The system can run only for java programs. The user must choose the values of multiplicity (cost drivers) therefore who must have an experience of the software development.The system can only estimate approximate effort cost and cannot calculate effort cost.

References

References Order

1. Dr.Barry Boehm,"COCOMOII Model Definition Manual 1997",University of Southern California.
2. I.Sommerville, "Software Engineering-Eight Edition", ISBN 0-321-31379-8, Pearson Education Limited.
3. Hamlet, Dick, Maybee, Joe, "The Engineering of Software", Addison-Wesley, Inc, USA, 2001.
4. Barry Boehm. Software Engineering Economics. Englewood Cliffs, NJ: Prentice-Hall, 1981. ISBN 0-13-822122-7.
5. http://www.Wikipedia.com/COCOMO