



## Department of Software Engineering

---

University of Gujrat Sialkot Sub Campus

# SOFTWARE PROJECT ESTIMATOR



Session: BSSE Spring 2014-2018

**Project Advisor:** **MS MAHUM ADILL**

**Submitted By**

*ARSLAN AHMED*                    *14061598-027*

*AHMED RIAZ*                    *14061598-063*

*AHMAD QADEER*                    *14061598-026*

---

Department of Software Engineering  
University of Gujrat Sialkot sub Campus

---

## STATEMENT OF SUBMISSION

This is certified that Arslan Ahmed Roll No. 14061598-027 and Ahmed Riaz Roll No. 14061598-063 and Ahmad Qadeer Roll No. 14061598-026 has successfully completed the final year project named as Software Project Estimator at the Department of Computing and Information Technology, University of Gujrat, to fulfill the requirement of the degree of **BS in Software Engineering**.

---

Project Supervisor

---

Project Coordination Office  
Faculty of C&IT -UOG

---

Head of the Department

### Acknowledgement

We truly acknowledge the cooperation and help make by Ms. Mahum Adil, Assistant Professor, University of Gujrat. He has been a constant source of guidance throughout the course of this project. We would also like to thank Ms. Faiza Tahir for his help and guidance throughout this project. We are also thankful to our friends and families whose silent support led us to complete our project.

1- Arslan Ahmed

2- Ahmed Riaz

3- Ahmad Qadeer

Date: July 30, 2018

## Abstract

Pre-planning and estimation are an important part for making things successful literally software industry is no exception to it. The execution of a productive and successful software or project is only possible if proper beforehand estimation is done. The sad part, however, is that the software industry still lacks effective estimation. The methods that are used do not give accurate results or in some other cases, the results are not availed properly. Consequently, the software projects suffer from under estimation and over estimation. Both under estimation and over estimation become the reason of inadequate staff, low quality software products, excessive use of resources, delayed project submissions and as a result project failure occur which caused the client's trust is lost forever. There are many reasons for such a large number of project failures. The basic reason being that there are no proper courses offered to students that could help them understand the processes of effective software estimation. Although there are a number of estimation tools and websites available over the internet but they are dispersed. Users either don't find an estimation technique that would support the process model of their software or they just perform the estimation using the wrong technique. To resolve these problems, a tool named 'Software Project Estimator' is developed. It would provide effective software estimation using three different estimation tools that accommodate different design process models. Estimator minimizes the chance of project failure by estimating the size, cost, effort and time of a project beforehand using three different techniques i.e. COCOMO I, COCOMO II and SLIM. The estimation is done on different phases of design and implementation. The gained results would be clear and consequently the project submitted will be on time and within budget without compromising on the quality.

## TABLE OF CONTENTS

### CHAPTER 1: PROJECT FEASIBILITY REPORT.....**10**

1.1. INTRODUCTION .....	11
1.2. PROJECT/PRODUCT FEASIBILITY REPORT .....	11
1.2.1. <i>Technical Feasibility</i> .....	11
1.2.2. <i>Operational Feasibility</i> .....	11
1.2.3. <i>Economic Feasibility</i> .....	12
1.2.4. <i>Schedule Feasibility</i> .....	12
1.2.5. <i>Specification Feasibility</i> .....	12
1.2.6. <i>Information Feasibility</i> .....	12
1.2.7. <i>Motivational Feasibility</i> .....	12
1.2.8. <i>Legal &amp; Ethical Feasibility</i> .....	12
1.3. PROJECT/PRODUCT SCOPE .....	12
1.4. PROJECT/PRODUCT COSTING .....	13
1.4.1. <i>Project Cost Estimation by Function Point Analysis</i> .....	13
1.5. TASK DEPENDENCY TABLE .....	15
1.6. CPM - CRITICAL PATH METHOD.....	16
1.7. GANTT CHART .....	17
1.8. TASK AND MEMBER ASSIGNMENT TABLE.....	18
1.9. TOOLS AND TECHNOLOGY WITH REASONING .....	18
1.10. VISION DOCUMENT .....	19
1.10.1. <i>Scope</i> .....	19
1.10.2. <i>Problems in Current System</i> .....	19
1.10.3. <i>Stake Holders</i> .....	19
1.10.4. <i>Assumption</i> .....	19
1.10.5. <i>Constraints</i> .....	19
1.10.6. <i>Key Benefits</i> .....	19
1.11. RISK LIST .....	20
1.12. PRODUCT FEATURES/ PRODUCT DECOMPOSITION .....	20

### CHAPTER 2: SOFTWARE REQUIREMENT SPECIFICATION (FOR OBJECT ORIENTED APPROACH) .....21

2.1 INTRODUCTION: .....	22
<i>Systems Specifications</i> .....	23
2.1.1. <i>Introduction</i> .....	23
2.1.2. <i>Existing System</i> .....	23
2.1.4. <i>Scope of the System</i> .....	24
2.1.5. <i>Summary of Requirements:</i> .....	24
<i>The user can login or register their self to manage their estimated projects as, save, view, print, delete.</i> .....	24
2.1.6. <i>Identifying External Entities</i> .....	24
2.1.7. <i>Context Level Data Flow Diagram:</i> .....	25
2.1.8. <i>Capture "shall" Statements:</i> .....	25
2.1.9. <i>Allocate Requirements:</i> .....	28
2.1.10. <i>Prioritize Requirements:</i> .....	33
2.1.11. <i>Requirements Trace-ability Matrix:</i> .....	41
2.1.12. <i>High Level Use Case Diagram:</i> .....	46
2.1.13. <i>Analysis Level Use Case Diagram:</i> .....	47
2.2.14. <i>Use case Description</i> .....	52

<b>CHAPTER 3: DESIGN DOCUMENT.....</b>	<b>83</b>
3.1. INTRODUCTION.....	84
3.2. DOMAIN MODEL .....	84
3.3. SYSTEM SEQUENCE DIAGRAM .....	85
3.4. SEQUENCE DIAGRAM .....	86
3.4.1. Sequence Diagram for Basic Stage COCOMO I.....	86
3.4.2. Sequence Diagram for Intermediate Stage COCOMO I.....	87
3.4.3. Sequence Diagram for Advanced Stage COCOMO I.....	88
Figure 3.2. 3: Advanced Stage COCOMO I.....	88
3.4.4. Sequence Diagram for Application Composition Model COCOMO II.....	89
Figure 3.2. 4: Application Composition Model COCOMO II.....	89
3.4.5. Sequence Diagram for Early Design Model COCOMO II.....	90
3.4.6. Sequence Diagram for Post Architecture Model COCOMO II.....	91
3.4.7. Sequence Diagram for Software Lifecycle Management Model SLIM.....	92
3.4.8. Sequence Diagram for Estimated Projects Management .....	93
3.5. DESIGN CLASS DIAGRAM.....	94
3.6. DATA MODEL.....	94
<b>CHAPTER 4: USER INTERFACE DESIGN .....</b>	<b>96</b>
4.1. INTRODUCTION.....	97
4.2. SITE MAPS .....	97
4.3. STORY BOARDS .....	98
4.4. NAVIGATIONAL MAPS: .....	99
4.5 TRACE-ABILITY MATRIX.....	117
<b>CHAPTER 5: SOFTWARE TESTING.....</b>	<b>123</b>
5.1. INTRODUCTION: .....	124
5.2. TEST PLAN .....	124
5.2.1. Purpose .....	124
5.2.2. Outline.....	124
HARDWARE REQUIREMENTS .....	127
SOFTWARE REQUIREMENTS .....	127
5.3. TEST CASE SPECIFICATION.....	128
5.3.1. Purpose .....	128
5.3.2. Outline.....	128
5.4. TEST PROCEDURE SPECIFICATION.....	156
5.4.1. Purpose .....	156
5.4.2 Outline.....	156
5.5. TEST ITEM TRANSMITTAL REPORT .....	158
5.5.1. Purpose .....	158
5.5.2. Outline.....	158
5.6. TEST LOG .....	159
5.6.1. Purpose .....	159
5.6.2. Outline.....	159
5.7. TEST INCIDENT REPORT .....	160
5.7.1. Purpose .....	160
5.7.2. Outline.....	160
5.8. TEST SUMMARY REPORT.....	161
5.8.1. Purpose .....	161

5.8.2. <i>Outline</i> .....	161
-----------------------------	-----

## LIST OF TABLES

### **CHAPTER 1**

Table 1. 1: Task Dependency Table .....	15
Table 1. 2: Critical Path Method.....	16
Table 1. 3: Task and Member Assignment Table .....	18

### **CHAPTER 2**

Table 2. 1: Scope of Software Project Estimator.....	24
Table 2. 2: Shall Statements.....	25
Table 2. 3: Allocation of Requirements with Use Case.....	28
Table 2. 4: Requirement Prioritization.....	33
Table 2. 5: Requirements Trace-Ability Matrix .....	41

## LIST OF FIGURES

### CHAPTER 1

Figure 1. 1: Project Estimation .....	15
Figure 1. 2: Gantt Chart .....	17

### CHAPTER 2

Figure 2. 1: System Specification .....	22
Figure 2. 2: Organization Chart .....	23
Figure 2. 3: Data Flow Diagram .....	25
Figure 2. 4: High Level Use Case Diagram .....	47

#### Analysis Level Use Case

Figure 2.5. 1: Project Details Module .....	48
Figure 2.5. 2: COCOMO I Module.....	49
Figure 2.5. 3: COCOMO II Module .....	50
Figure 2.5. 4: SLIM Module .....	51
Figure 2.5. 5: Estimated Project Management Module .....	52

### CHAPTER 3

Figure 3. 1: Domain Model of the system .....	84
---	----

#### Sequence Diagram

Figure 3.2. 1: Basic Stage COCOMO I .....	86
Figure 3.2. 2: Intermediate Stage COCOMO I .....	87
Figure 3.2. 3: Advanced Stage COCOMO I .....	88
Figure 3.2. 4: Application Composition Model COCOMO II.....	89
Figure 3.2. 5: Early Design Model COCOMO II .....	90
Figure 3.2. 6: Post Architecture Model COCOMO II .....	91
Figure 3.2. 7: Software Lifecycle Management Model SLIM.....	92
Figure 3.2. 8: Estimated Projects Management .....	93

#### Class Diagram

Figure 3. 3: Class Diagram of the system .....	94
--	----

#### Entity Relationship Diagram

Figure 3. 5: Entity Relationship Diagram of the system.....	95
---	----

## **Chapter 1: Project Feasibility Report**

---

### **1.1. Introduction**

Software Project Estimator helps in the estimation of size, effort, cost and development time of a project at different phases of design and implementation, that too, beforehand. The estimation will be done using three techniques; COCOMO I, COCOMO II and SLIM. All three of these accommodate different design process models. The motivation behind the project was to provide a single platform to the users for effective software estimation purposes. The website is a combination of three different estimation techniques; COCOMO1, COCOMO2 and SLIM that would estimate the size, effort, cost and development time beforehand. This will result in timely project submissions, quality work, adequate resources, and there would be less chances of project failure.

Incremental methodology is used because of a number of reasons. A working software is obtained quickly and early during the software lifecycle. Risk management becomes easier because erroneous parts of the software are identified at each iteration. Smaller iterations enable easy testing and debugging and flexibility of the model provides low cost if scope and requirements are to be changed. Coming to the object-oriented approach, it is used because of its useful aspects like code reusability, flexibility, reliability.

### **1.2. Project/Product Feasibility Report**

When a project is started the first matter to establish is to assess the feasibility of a project or product. Feasibility means the extent to which appropriate data and information are readily available or can be obtained with available resources such as staff, expertise, time, and equipment. It is basically used as a measure of how practical or beneficial the development of a software system will be to you (or organization). This activity recurs throughout the life cycle.

There are many types of feasibilities:

- Technical
- Operational
- Economic
- Schedule
- Specification
- Information
- Motivational
- Legal and Ethical

#### **1.2.1. Technical Feasibility**

We use web technologies. Web based applications are easy to develop and implement. Web based applications are far more compatible across platforms than traditional installed software's. Web based systems need only be installed on the server placing minimal requirements on the end user workstation. Due to the manageability and cross platform support deploying web applications to the end user is far easier. Typically, in larger more complex systems data is stored and moved around separate systems and data sources. In web-based systems these systems and processes can often be consolidated reducing the need to move data around. More ever web-based apps contain more secured live data and reduce costs.

#### **1.2.2. Operational Feasibility**

Pre-planning and estimation are an important part for making things successful and software industry is no exception to it. The execution of a productive and successful software or project is only possible if proper beforehand estimation is done. The sad part,

however, is that the software industry still lacks effective estimation. The methods that are used do not give accurate results or in some other cases, the results are not availed properly. Consequently, the software projects suffer from under estimation and over estimation. Both under estimation and over estimation become the reason of inadequate staff, low quality software products, excessive use of resources, delayed project submissions and as a result project failure occur and the clients' trust is lost forever. Software Project Estimator will help to estimate their projects with user friendly interface.

### **1.2.3. Economic Feasibility**

Our project is economically feasible to us because here we have use the efficient method of cost analysis and budget of our project. It helps in identifying profit against investment expected from a project. It Increased agency revenue, agency costs and other public benefits. We will demonstrate the net benefit of a proposed project for accepting or disbursing electronic funds/benefits, taking into consideration the benefits and costs to the agency, other state agencies, and the general public as a whole. Without proper pre-planning and estimation, a project couldn't be successful. Currently there is no software that can estimate a software project that too in early development stages. Our project helps the project manager to estimate size, time, cost of their projects during early stages. So, we can gain more economical benefit from that project.

### **1.2.4. Schedule Feasibility**

Time is an important factor to achieve the success. Our staff will present our deliverables and we will meet our milestones and checkpoints within time. Our project has a high probability to be completed on-time and we can also include as an alternate for the timelines so we can meet our deadlines on time. We ensure that a project will be completed before the project or technology becomes obsolete or unnecessary.

### **1.2.5. Specification Feasibility**

All the specification of our project is achievable logically and feasibly within available resources and time. Software Project Estimator minimizes the chance of project failure by estimating the size, cost, effort and time of a project beforehand using three different techniques i.e. COCOMO I, COCOMO II and SLIM. The estimation is done on different phases of design and implementation. The gained results would be clear and consequently the project submitted will be on time and within budget without compromising on the quality.

### **1.2.6. Information Feasibility**

It would provide effective software estimation using three different estimation models i.e. COCOMO I, COCOMO II and SLIM. All of these are worldwide approved software estimation model.

### **1.2.7. Motivational Feasibility**

The motivation behind the project was to provide a single platform to the users for effective software estimation purposes. The website is a combination of three different estimation techniques; COCOMO I, COCOMO I and SLIM that would estimate the size, effort, cost and development time beforehand. This will result in timely project submissions, quality work, adequate resources, and there would be less chances of project failure. Our team has good problem-solving skills. Our mentor is skill full and motivated.

### **1.2.8. Legal & Ethical Feasibility**

Our project is feasible legally and ethically. All information will be confidential and not being provided to other organizations. We do not copy paste the others data.

## **1.3. Project/Product Scope**

The scope of this project is extremely high because in Pakistan most software house

doesn't estimate their projects. The most important part for making a successful software project is pre-planning and estimation. Unfortunately, the software industry lacks effective project estimation. Most of the time, the projects are either not estimated well, or the gained estimates are not used properly. The methods to determine the project progress are poor and lack real-time visibility and control. This leads to problems like under-estimation or over-estimation which causes failure or low-quality software products.

Serious problems occur in cases where a project is under-estimated. The staff required for the project becomes inadequate that exhausts the staff members and even diminishes their interest in the project. Not having an estimation of the project's scope also leads to the risk of delivering low quality product to the users. Ineffective time estimates are equally drastic in the software development process because not meeting the deadlines can cause the client to lose trust. Moreover, in the case where the project is over-estimated; the excessive use of resources will result in more cost, longer delivery time and an overall negative impact on the future projects.

### **1.4. Project/Product Costing**

#### **1.4.1. Project Cost Estimation by Function Point Analysis**

##### **4 Unadjusted Function Points**

Elements	Complexity Weighting Factors		
	Low	Average	High
External Inputs (EI)	30	10	3
External Outputs (EO)	20	5	2
External Inquiries (EQ)	3	10	1
External Interface files (EIF)	0	0	0
Internal Logical files (ILF)	6	6	1
<b>Total:</b> 439			

## 5

## Value Adjustment Factors

Data Communication: 3	Distributed Data Processing: 2	Performance: 5
Heavily Used Configuration: 1	Online Data Entry: 1	Transaction Rate: 1
End User Efficiency: 5	Complex Processing: 1	Online Updates: 1
Resuabilty: 3	Installation Ease: 4	Operational Ease: 5
Multiple Sites: 2	Facilitate Change: 1	
<b>Total: 1.0001</b>		

## 6

## Effort Adjustment Factor

### PRODUCT ATTRIBUTES

1. Reliability (RELY): High
2. Data (DATA): Nominal
3. Product Complexity (CPLX): Very Low

### PLATFORM ATTRIBUTES

1. Execution time and Constraint(TIME): Nominal
2. Main Storage Constraint (STOR): Nominal
3. Virtual Machine Volatility (VIRT): Low
4. Computer Turnaround Time (TURN): Very High

### PERSONAL ATTRIBUTES

1. Analyst Capability (ACAP): Nominal
2. Application Experience (AEXP): Nominal
3. Programmer Capability (PCAP): Nominal
4. Virtual Machine Experience (VEXP): Nominal
5. Language Experience (LEXP): Nominal

### PROJECT ATTRIBUTES

1. Modern Programming Practices (MODP): Low
2. Software Tools (TOOL): Nominal
3. Development Schedule (SCED): Nominal

**Total:** 0.89

## 7 Final Estimations

**Size:** 439 KLOC

**Effort:** 2.4 person/month

**Development Time:** 3 months

**Figure 1. 1: Project Estimation**

### 1.5. Task Dependency Table

Software Project Estimator has following tasks and their dependencies.

**Table 1. 1: Task Dependency Table**

Task No #	Activity	Dependencies
T1	Interest Discussion	-
T2	Field Study	-
T3	Group Discussion	-
T4	Selection	T1, T2, T3
T5	Platform Investigation	T4
T6	Technology Analysis	T5
T7	Resource Evaluation	T6
T8	Literature Research	T7
T9	Existing product research	T3
T10	Potential users question	T9
T11	Requirements	T10
T12	SRS	T11
T13	Design	T12

<b>T14</b>	Use Case Diagram	T13
<b>T15</b>	Entity Relation Diagram	T14
<b>T16</b>	Graphical User Interface Diagram	T14, T15
<b>T17</b>	Object Modeling	T16
<b>T18</b>	Front End Implementation	T17
<b>T19</b>	Back End Implementation	T17
<b>T20</b>	Testing	T19

### **1.6. CPM - Critical Path Method**

In 1957, DuPont developed a project management method designed to address the challenge of shutting down chemical plants for maintenance and then restarting the plants once the maintenance had been completed. Given the complexity of the process, they developed the Critical Path Method (CPM) for managing such projects.

CPM provides the following benefits:

- Provides a graphical view of the project.
- Predicts the time required to complete the project.
- Shows which activities are critical to maintaining the schedule and which are not.

The critical path method table of Software Project Estimator is Follows:

**Table 1. 2: Critical Path Method**

Task No #	Activity	Duration	ES	EF	LS	LF	TS	FS
<b>T1</b>	Interest Discussion	1W	0	1W	0	1W	0	0
<b>T2</b>	Field Study	1W	0	1W	0	1W	0	0
<b>T3</b>	Group Discussion	1W	0	1W	0	1W	0	0
<b>T4</b>	Selection	2W	1W	3W	1W	3W	0	0
<b>T5</b>	Platform Investigation	1W	3W	4W	3W	4W	0	0
<b>T6</b>	Technology Analysis	2W	4W	6W	4W	6W	0	0
<b>T7</b>	Resource Evaluation	2W	6W	8W	6W	8W	0	0
<b>T8</b>	Literature Research	2W	8W	10W	8W	10W	0	0
<b>T9</b>	Existing product research	1W	3W	4W	11W	12W	8	0
<b>T10</b>	Potential users question	2W	10W	12W	10W	12W	0	0
<b>T11</b>	Requirements	4W	12W	16W	12W	16W	0	0
<b>T12</b>	SRS	1W	16W	17W	16W	17W	0	0
<b>T13</b>	Design	2W	17W	19W	17W	19W	0	0

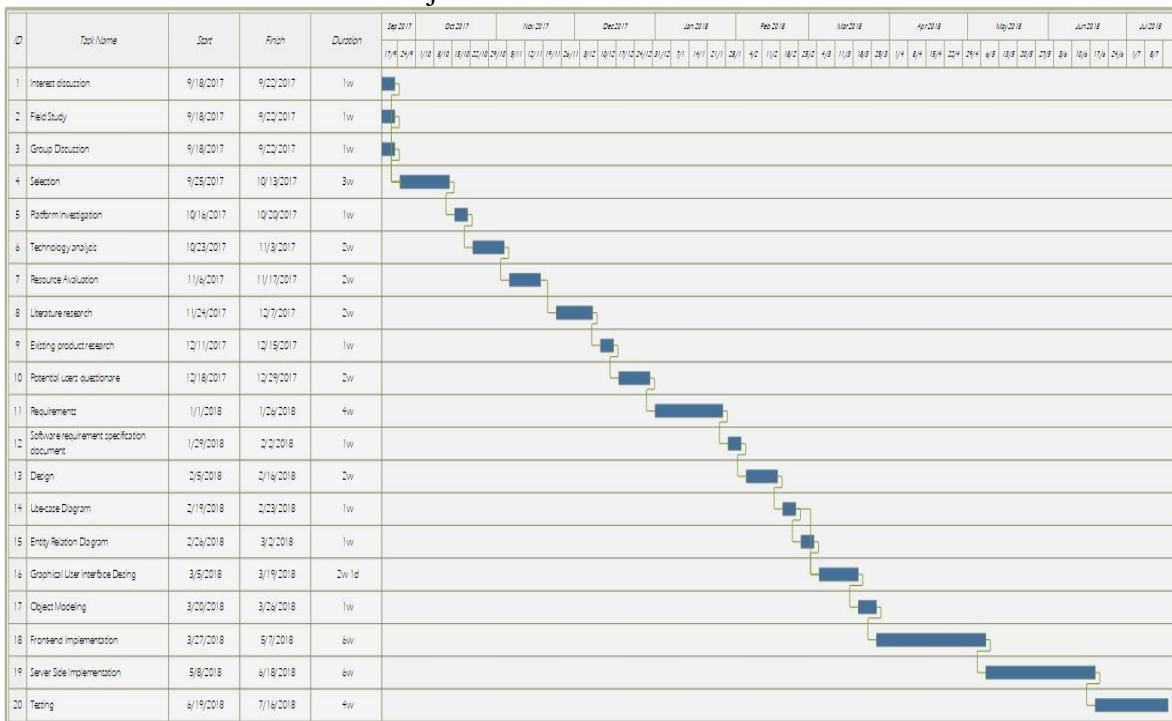
## Software Project Estimator

<b>T14</b>	Use Case Diagram	1W	19W	20W	19W	20W	0	0
<b>T15</b>	Entity Relation Diagram	1W	20W	21W	20W	21W	0	0
<b>T16</b>	Graphical User Interface Diagram	2W	21W	23W	21W	23W	0	0
<b>T17</b>	Object Modeling	1W	23W	24W	23W	24W	0	0
<b>T18</b>	Front End Implementation	6W	24W	30W	24W	30W	0	0
<b>T19</b>	Back End Implementation	6W	24W	30W	24W	30W	0	0
<b>T20</b>	Testing	4W	30W	34W	30W	34W	0	0

### 1.7. Gantt chart

The Gantt chart enumerates the activities to be performed on the vertical axis and their corresponding duration on the horizontal axis. It is possible to schedule activities by either early start or late start logic. In the early start approach, each activity is initiated as early as possible without violating the precedence relations. In the late start approach, each activity is delayed as much as possible as long as the earliest finish time of the project is not compromised. Based on the Work Breakdown Structure (WBS), a timeline or Gantt chart showing the allocation of time to the project phases or iterations should be developed. This Gantt chart would identify major milestones with their achievement criteria. It must contain duration estimation of all the necessary activities to be carried out during the project development along with the human resources responsible for the respective tasks. Activity dependencies are also required to be mentioned in it.

The Gantt chart of Software Project Estimator is follows:



**Figure 1. 2: Gantt Chart**

### **1.8. Task and Member Assignment Table**

This table consists of a list of tasks and correspondingly allocation of members to that task. The basic aim of this table would be an indication of the amount of work the members would be performing.

**Table 1. 3: Task and Member Assignment Table**

<b>Task No #</b>	<b>Activity</b>	<b>Engineer</b>	<b>Duration</b>
<b>T1</b>	Interest Discussion	Ahmed Riaz	1W
<b>T2</b>	Field Study	Arslan Ahmed	1W
<b>T3</b>	Group Discussion	Arslan Ahmed, Ahmed Riaz, Ahmad Qadeer	1W
<b>T4</b>	Selection	Arslan Ahmed, Ahmed Riaz, Ahmad Qadeer	2W
<b>T5</b>	Platform Investigation	Arslan Ahmed	1W
<b>T6</b>	Technology Analysis	Ahmed Qadeer	2W
<b>T7</b>	Resource Evaluation	Ahmed Riaz	2W
<b>T8</b>	Literature Research	Arslan Ahmed	2W
<b>T9</b>	Existing product research	Ahmed Riaz	1W
<b>T10</b>	Potential users question	Ahmed Qadeer	2W
<b>T11</b>	Requirements	Arslan Ahmed, Ahmed Riaz, Ahmad Qadeer	4W
<b>T12</b>	SRS	Arslan Ahmed	1W
<b>T13</b>	Design	Arslan Ahmed, Ahmed Riaz, Ahmad Qadeer	2W
<b>T14</b>	Use Case Diagram	Ahmed Qadeer	1W
<b>T15</b>	Entity Relation Diagram	Ahmed Riaz	1W
<b>T16</b>	Graphical User Interface Diagram	Arslan Ahmed	2W
<b>T17</b>	Object Modeling	Ahmed Riaz	1W
<b>T18</b>	Front End Implementation	Arslan Ahmed, Ahmed Riaz, Ahmad Qadeer	6W
<b>T19</b>	Back End Implementation	Arslan Ahmed, Ahmed Riaz	6W
<b>T20</b>	Testing	Arslan Ahmed, Ahmed Riaz, Ahmad Qadeer	4W

### **1.9. Tools and Technology with reasoning**

The application tools, which are used to develop front and back end of the system are follows:

- HTML5
- CSS3
- Bootstrap 3
- ASP.NET MVC
- C#
- Javascript
- SQL Server

As with rapid increase in use of new web languages we are including HTML5, CSS3, Bootstrap 3 for designing web terminal of our project. C# and Javascript with best combination of backend support of SQL server. These languages and techniques are used as if our client any time demands for an application that would have specifications to run at android as well as on IOS. Then we can easily convert our site to these platform's friendly code.

### **1.10. Vision Document**

Estimator helps in the estimation of size, effort, cost and development time of a project at different phases of design and implementation, that too, beforehand. The estimation will be done using three techniques; COCOMO I, COCOMO II and SLIM. All three of these accommodate different design process models.

#### **1.10.1. Scope**

The tool can be used in any software house, or locally by project manager.

#### **1.10.2. Problems in Current System**

There are some systems that provide estimation for software projects but that estimation supports just one type of estimation model. There is no such system that would provide estimation using three different models.

#### **1.10.2.1. Solution**

The Software Project Estimator, provide a single platform to the users for effective software estimation purposes. The website is a combination of three different estimation techniques; COCOMO I, COCOMO II and SLIM that would estimate the size, effort, cost and development time beforehand. This will result in timely project submissions, quality work, adequate resources, and there would be less chances of project failure.

#### **1.10.3. Stake Holders**

The people which are involved in development of Software Project Estimator called Stake Holders, which are follows:

- Project Manager
- Learners / Teachers
- Development Team

#### **1.10.4. Assumption**

- The system will be available 24 hours.
- The system saves the estimated project details for each user.
- The system will display estimated results near to accuracy.
- The system should be able to estimate project less than 5 minutes.
- User may have basic knowledge of estimation techniques.

#### **1.10.5. Constraints**

- System must support different platforms e.g. Windows, Android, IOS, Linux, and UNIX standard browsers.
- System must be developed in .NET MVC with c#.

#### **1.10.6. Key Benefits**

- Level of accuracy in this system will be higher. All operation would be done correctly and it insure all information is accurate.
- No information is repeated anywhere in storage.
- The system would be easy to operate and it can be developed within short period of time within budget.
- Avoid errors and track every single detail
- Improve data security

- Friendly user interface
- Saving paper work
- Data can be easily insert update and delete.

### **1.11. Risk List**

The possibility of suffering harm or loss in terms of danger is called risk. Regarding the importance of risks, a list is to be maintained. Risk list is a sorted list of known, open risks to the project, sorted in decreasing order of importance, associated with specific mitigation or contingency actions. The risks are following that may occur are following:

- If any or team member cannot continue with this project there is no substitute of it, because it takes more development time to start from beginning.
- Development time if less then pressure would be high.
- Low motivation and moral support.
- Detail reporting would take more development time.
- Adding extra functionalities increase cost of project.
- People assignment do not match their strength.
- Team members needs extra time to learn unfamiliar tools or technique.
- Conflicts among team member's ideas result in poor performances, more meetings and extra rework.
- Development of extra functionalities that are not required will expand the schedule.
- Development team may deliver components of low quality and more time will require to improve quality.
- Lack of experiences in project management, especially in testing, risk management and changes management in existing team.
- All team member need time for preparation of midterm, final and other work.
- Resources are not enough.

### **1.12. Product Features/ Product Decomposition**

The core feature of this tool is the estimation of size, effort, cost and development time of a project at different phases of design and implementation, that too, beforehand. The estimation will be done using three techniques. All three of these accommodate different design process models. The tool will be beneficial for all kind of software project developers i.e. software house or locally user.

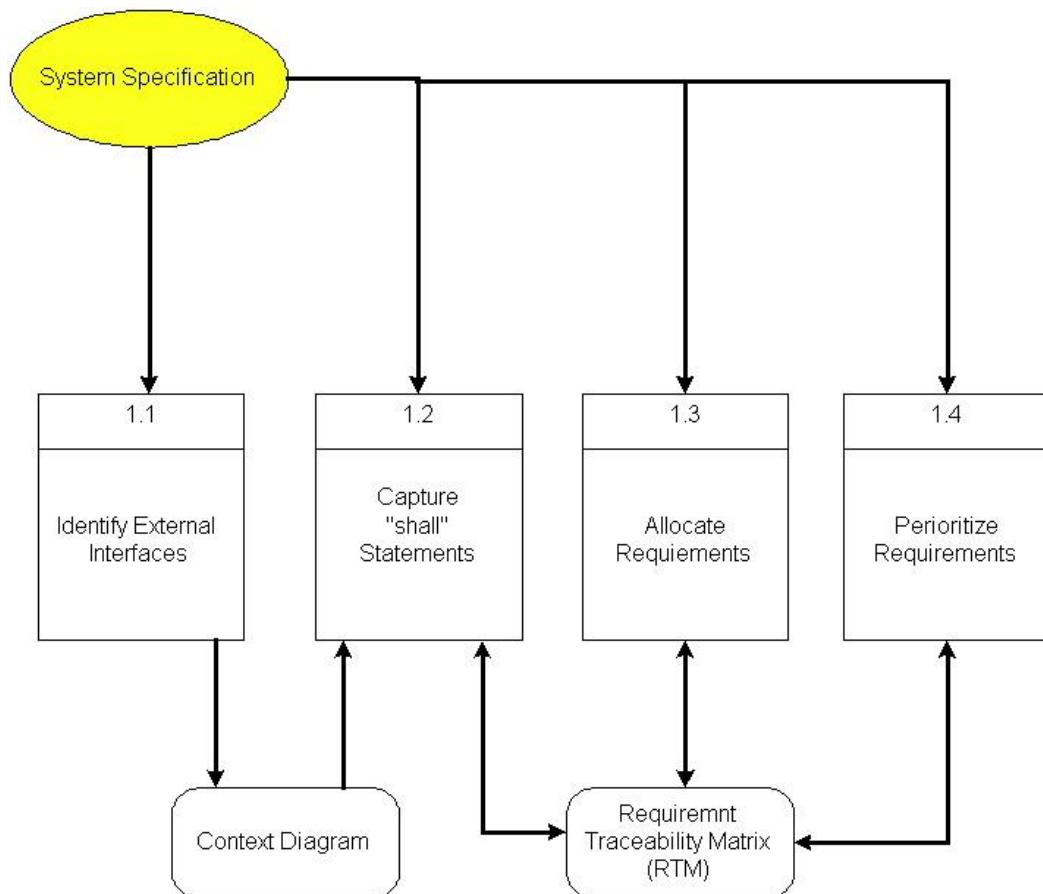
## **Chapter 2: Software Requirement Specification (For Object Oriented Approach)**

---

## 2.1 Introduction:

Requirements engineering process provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are transformed into an operational system. The task of capturing, structuring, and accurately representing the user's requirements so that they can be correctly embodied in systems which meet those requirements (i.e. are of good quality).

- Requirements elicitation
- Requirements analysis and negotiation
- Requirements specification
- System modeling
- Requirements validation
- Requirements management



**Figure 2. 1: System Specification**

Here, requirements specification is to be discussed. Requirements specification would lead to the following four steps:

- Identify external interfaces
- Development of context diagram
- Capture “shall statements
- Allocate requirements

## Software Project Estimator

- Prioritize requirements
- Development of requirements traceability matrix

### Systems Specifications

The following are the clauses that describe the system specifications of Software Project Estimator.

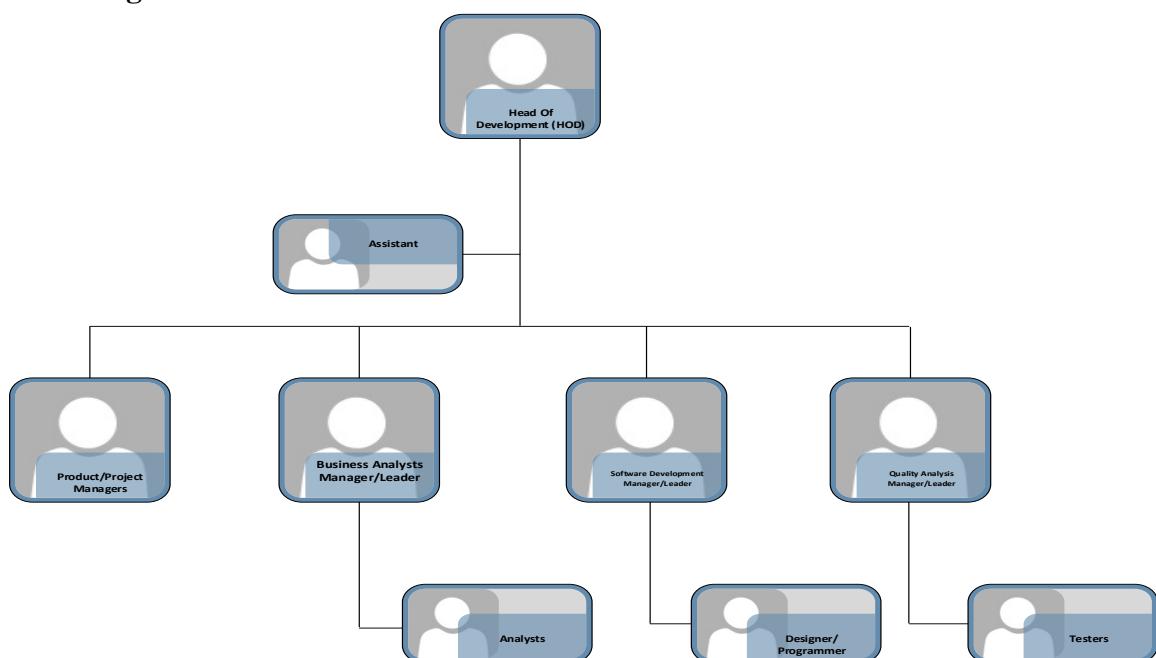
#### 2.1.1. Introduction

Software Project Estimator helps in the estimation of size, effort, cost and development time of a project at different phase of design and implementation, that too, beforehand. The estimation will be done using three techniques; COCOMO I, COCOMO II and SLIM. All three of these accommodate different design process models. There are some systems that provide estimation for software projects but that estimation supports just one type of estimation model. There is no such system that would provide estimation using three different models. The motivation behind the project was to provide a single platform to the users for effective software estimation purposes. This will result in timely project submissions, quality work, adequate resources, and there would be less chances of project failure.

#### 2.1.2. Existing System

Even though there are many software estimation tools available, but they are in a scattered form and they provide estimation for just one type of software product. Users are not able to properly judge the software estimation technique that is suitable for their project because there is no single platform available that provides estimation for all types of software products i.e. System Star Estimation Tool which only provides the estimation of COCOMO. Estimating all the features required for the software development process, that too, at a single platform, is the biggest faced challenge for the software industry.

#### 2.1.3. Organizational Chart



**Figure 2. 2: Organization Chart**

#### **2.1.4. Scope of the System**

Software Project Estimator contains following features.

**Table 2. 1: Scope of Software Project Estimator**

For	Project Managers, Developers, Learners, Teachers, Computer Scientists and IT related persons.
What	Project Size Estimation. Project Effort Estimation. Project Development Time Estimation. Project Cost Estimation.
The	Software Project Estimator
Is	A Web Application
That	Helps in the estimation of size, effort, cost and development time of a project at different phases of design and implementation, that too, beforehand. The estimation will be done using three techniques; COCOMO I, COCOMO II and SLIM. All three of these accommodate different design process models.

#### **2.1.5. Summary of Requirements:**

##### **2.1.5.1. Project Details**

The user shall provide the project name and project process model, the system will automatically redirect to best suited estimation technique i.e. COCOMO I COCOMO II or SLIM.

##### **2.1.5.2. COCOMO I**

There are three Modes Organic, Semidetached, Embedded and also three Stages Basic, Intermediate, Advanced of COCOMO I. User shall have to select both mode and stage one by one. User shall provide the values of UFP, VAF, EAF and will get the estimated size, effort and development time.

##### **2.1.5.3. COCOMO II**

There are three Models Application Composition, Early Design, and Post Architecture of COCOMO II. User shall have to select one model and moved for further estimation. User shall provide the values of OP, NOP, 3GL Modules, Reuse Percentage, Productivity Rate, Scale Component, UFP, VAF, EAF and will get the estimated size, effort and development time at different phases of SDLC.

##### **2.1.5.4. SLIM**

SLIM Model used for large size projects which may have 70,000 plus line of code (LOC). User shall have to provide the values of Process Productivity, UFP, VAF and will get the estimated size, total effort, effort at delivery time and manpower build up. The complete report will be generated at the end of each stage and model. The print option and save option will be available. User can print at the spot and can save for future.

##### **2.1.5.5. Estimated Project Management**

The user can login or register their self to manage their estimated projects as, save, view, print, delete.

#### **2.1.6. Identifying External Entities**

##### **a. Over Specify Entities from Abstract:**

## Software Project Estimator

The following are the entities for software estimation tool.

- Project Managers
- Learners and Teachers
- Computer Scientists and IT related persons.

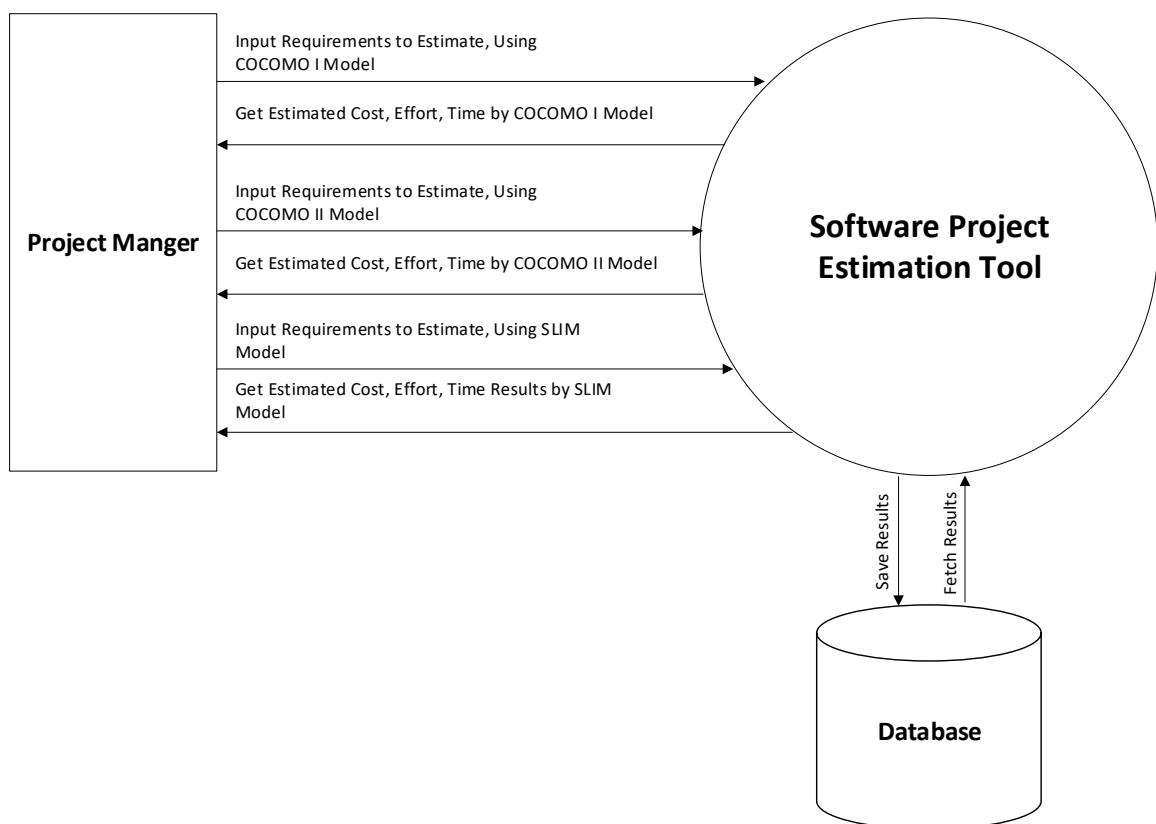
### b. Perform Refinement:

The following entities are result in to be more related to Business Logic.

- Project Managers
- Learners and Teachers

#### 2.1.7. Context Level Data Flow Diagram:

All external entities are shown on the context diagram of Software Project Estimator as well as major data flow to and from them.



**Figure 2. 3: Data Flow Diagram**

#### 2.1.8. Capture "shall" Statements:

The “shall” statements of the Software Project Estimator are shown in Table 2.2 as they would be the functional requirements.

**Table 2. 2: Shall Statements**

Para #	Initial Requirements
1.1	The system shall allow user to enter the project name.
1.2	The system shall allow user to move best estimation technique by selecting a process model.
2.1	The system shall allow user to enter External Inputs (EI) weight as low, average, high and multiply with constants as 3, 4, 6 respectively.

## Software Project Estimator

2.2	The system shall allow user to enter External Outputs (EO) weight as low, average, high and multiply with constants as 4, 5, 7 respectively.
2.3	The system shall allow user to enter External Enquiries (EQ) weight as low, average, high and multiply with constants as 3, 4, 6 respectively.
2.4	The system shall allow user to enter External Interface Files (EIF) weight as low, average, high and multiply with constants as 5, 7, 10 respectively.
2.5	The system shall allow user to enter Internal Logical Files (ILF) weight as low, average, high and multiply with constants as 7, 10, 15 respectively.
2.6	The system shall calculate unadjusted function point using EI, EO, EQ, EIF, and ILF.
3.1	The system shall allow user to rate 14 value adjustment factors.
4.1	The system shall allow user to rate product attributes RELY, DATA, and CPLX as extra high, very high, high, nominal, low, very low.
4.2	The system shall allow user to rate platform attributes TIME, STOR, VIRT, and TURN as extra high, very high, nominal, low, very low.
4.3	The system shall allow user to rate personnel attributes ACAP, AEXP, PCAP, VEXP, and LEXP as extra high, very high, nominal, low, very low.
4.4	The system shall allow user to rate project attributes MODP, TOOL, and SCED as extra high, very high, nominal, low, very low.
5.1	The system shall allow user to select the project mode of COCOMO I.
5.2	The system shall allow user to select the stage of COCOMO I.
5.3	The system shall estimate the size using unadjusted function point and value adjustment factors in KLOC at basic stage.
5.4	The system shall take effort adjustment factor as 1 at basic stage.
5.5	The system shall estimate effort using size, effort adjustment factor and constant values (a b) of selected mode at basic stage.
6.1	The system shall estimate the size using unadjusted function point and value adjustment factors in KLOC at intermediate stage.
6.2	The system shall calculate effort adjustment factor using product, platform, personnel and project attributes rating weights intermediate stage.
6.3	The system shall estimate effort using size, effort adjustment factor and constant values (a b) of selected mode at intermediate stage.
6.4	The system shall estimate development time using effort and constant values (c d) at intermediate stage.
7.1	The system shall estimate the size using unadjusted function point and value adjustment factors in KLOC at advanced stage.
7.2	The system shall allow user to rate product, platform, personnel, project attributes for requirement planning and product design phase at advanced stage.
7.3	The system shall allow user to rate product, platform, personnel, project attributes for detailed design phase at advanced stage.
7.4	The system shall allow user to rate product attributes for code and unit test phase at advanced stage.
7.5	The system shall allow user to rate product, platform, personnel, project attributes for integration and test phase at advanced stage.

7.6	The system shall calculate effort adjustment factor for all phases at advanced stage.
7.7	The system shall estimate effort for all phases using size, effort adjustment factor of all phases and constant values (a b) of selected mode at advanced stage
7.8	The system shall estimate development time using efforts of all phases and constant values (c d) at intermediate stage.
8.1	The system shall allow user to rate Precedence (PREC) as very low, low, nominal, high, very high, extra high.
8.2	The system shall allow user to rate Development/Flexibility (FLEX) as very low, low, nominal, high, very high, extra high.
8.3	The system shall allow user to rate Architecture/Risk Resolution (RESL) as very low, low, nominal, high, very high, extra high.
8.4	The system shall allow user to rate Team Cohesion (TEAM) as very low, low, nominal, high, very high, extra high.
8.5	The system shall allow user to rate Process Maturity (PMAT) as very low, low, nominal, high, very high, extra high.
9.1	The system shall allow user to select the model of COCOMO II.
9.2	The system shall allow user to enter the no of screens in their project at application composite model.
9.3	The system shall allow user to enter the no of reports in their project at application composite model.
9.4	The system shall calculate object point by taking no of reports and no of screens at application composite model.
9.5	The system shall allow user to select developer's skill as very high, high, nominal, low, very low to calculate productivity rate.
9.6	The system shall calculate scale factor using Precedence, Development/Flexibility, Architecture/Risk Resolution, Team Cohesion, and Process Maturity.
9.7	The system shall estimate effort by taking object point and productivity rate at application composite model.
10.1	The system shall allow user to select programming language to calculate language factor.
11.1	The system shall estimate the size using unadjusted function point, value adjustment factors and language factor in KLOC at early design model.
11.2	The system shall calculate effort adjustment factor using product, platform, personnel and project attributes rating weights at early design model.
11.3	The system shall estimate effort using size, effort adjustment factor and a constant value 2.45 at early design model. ( $\text{Effort} = 2.45 * \text{Size} * \text{EAF}$ )
12.1	The system shall calculate function point at post architecture model.
12.2	The system shall estimate the size using unadjusted function point, value adjustment factors and language factor in KLOC at post architectural model.
12.3	The system shall calculate effort adjustment factor using product, platform, personnel and project attributes rating weights at post architectural model.
12.4	The system shall calculate scale factor using Precedence, Development/Flexibility, Architecture/Risk Resolution, Team Cohesion, and Process Maturity.
12.5	The system shall estimate effort for Management, Requirement, Design, Implementation, and Deployment phases at post architectural model.

## Software Project Estimator

12.6	The system shall estimate development time for Management, Requirement, Design, Implementation, and Deployment phases at post architectural model.
13.1	The system shall allow user to select their project type at software life cycle management (SLIM) model.
13.2	The system shall estimate the size using unadjusted function point, value adjustment factors in LOC at software life cycle management (SLIM) model.
13.3	The system shall take development time 12 months at software life cycle management (SLIM) model.
13.4	The system shall estimate the total effort (B) using size, development time and application type weight at software life cycle management (SLIM) model.
13.5	The system shall estimate the effort at delivery time using total effort and constant value 0.393513 at software life cycle management (SLIM) model.
13.6	The system shall calculate the Process Productivity of project
13.7	The system shall calculate the manpower acceleration of project.
14.1	The system shall calculate cost by using time and effort.
15.1	The system shall allow user to register himself.
15.2	The system shall allow user to login.
16.1	The system shall allow user to generate report of each technique.
16.2	The system shall allow user to print report.
16.3	The system shall allow user to save report.
16.4	The system shall allow user to view report.
16.5	The system shall allow user to delete report.

### **Non-Functional Requirements**

<b>Usability</b>	
US-1	The interface of application will be consistent. In case of error occurrence, the website will notify the users by displaying an error message. For example, if the unadjusted function point, values entered by user is not valid, the error message will notify the users to enter another valid value.
US-2	The website will be developed user friendly.
<b>Accuracy</b>	
AR-1	User will get the accurate information of the selected option.
<b>Performance</b>	
PR-1	User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Supportability</b>	
SR-1	User will get proper help about each model by the user manual and the help provided by the system.

### **2.1.9. Allocate Requirements:**

All “shall” requirements are allocated with use case as shown in Table 2.3

**Table 2. 3: Allocation of Requirements with Use Case**

Para #	Initial Requirements	Use Case Name
1.1	The system shall allow user to enter the project name.	UC_Enter_Project_Name

## Software Project Estimator

1.2	The system shall allow user to move best estimation technique by selecting a process model.	UC_Select_Process_Model
2.1	The system shall allow user to enter External Inputs (EI) weight as low, average, high and multiply with constants as 3, 4, 6 respectively.	UC_Cal_EI
2.2	The system shall allow user to enter External Outputs (EO) weight as low, average, high and multiply with constants as 4, 5, 7 respectively.	UC_Cal_EO
2.3	The system shall allow user to enter External Enquiries (EQ) weight as low, average, high and multiply with constants as 3, 4, 6 respectively.	UC_Cal_EQ
2.4	The system shall allow user to enter External Interface Files (EIF) weight as low, average, high and multiply with constants as 5, 7, 10 respectively.	UC_Cal{EIF}
2.5	The system shall allow user to enter Internal Logical Files (ILF) weight as low, average, high and multiply with constants as 7, 10, 15 respectively.	UC_Cal_ILF
2.6	The system shall calculate unadjusted function point using EI, EO, EQ, EIF, and ILF.	UC_Cal_Unadjusted_Function_Point
3.1	The system shall allow user to rate 14 value adjustment factors.	UC_Cal_Value_Adjustment_Factor
4.1	The system shall allow user to rate product attributes RELY, DATA, and CPLX as extra high, very high, high, nominal, low, very low.	UC_Cal_Product_Attributes
4.2	The system shall allow user to rate platform attributes TIME, STOR, VIRT, and TURN as extra high, very high, nominal, low, very low.	UC_Cal_Platform_Attributes
4.3	The system shall allow user to rate personnel attributes ACAP, AEXP, PCAP, VEXP, and LEXP as extra high, very high, nominal, low, very low.	UC_Cal_Personnel_Attributes
4.4	The system shall allow user to rate project attributes MODP, TOOL, and SCED as extra high, very high, nominal, low, very low.	UC_Cal_Project_Attributes
5.1	The system shall allow user to select the project mode of COCOMO I.	UC_Select_Mode_COCOMOI

## Software Project Estimator

5.2	The system shall allow user to select the stage of COCOMO I.	UC_Select_Stage_COCOMOI
5.3	The system shall estimate the size using unadjusted function point and value adjustment factors in KLOC at basic stage.	UC_Est_Size_Basic_COCOMOI
5.4	The system shall take effort adjustment factor as 1 at basic stage.	UC_Cal_Cost_Drivers
5.5	The system shall estimate effort using size, effort adjustment factor and constant values (a b) of selected mode at basic stage.	UC_Est_Effort_Basic_COCOMOI
6.1	The system shall estimate the size using unadjusted function point and value adjustment factors in KLOC at intermediate stage.	UC_Est_Size_Intermediate_COCOMOI
6.2	The system shall calculate effort adjustment factor using product, platform, personnel and project attributes rating weights intermediate stage.	UC_Cal_Cost_Drivers
6.3	The system shall estimate effort using size, effort adjustment factor and constant values (a b) of selected mode at intermediate stage.	UC_Est_Effort_Intermediate_COCOMOI
6.4	The system shall estimate development time using effort and constant values (c d) at intermediate stage.	UC_Est_Time_Intermediate_COCOMOI
7.1	The system shall estimate the size using unadjusted function point and value adjustment factors in KLOC at advanced stage.	UC_Est_Size_Advanced_COCOMOI
7.2	The system shall allow user to rate product, platform, personnel, project attributes for requirement planning and product design phase at advanced stage.	UC_Cal_RPD
7.3	The system shall allow user to rate product, platform, personnel, project attributes for detailed design phase at advanced stage.	UC_Cal_DD
7.4	The system shall allow user to rate product attributes for code and unit test phase at advanced stage.	UC_Cal_CUT
7.5	The system shall allow user to rate product, platform, personnel, project attributes for integration and test phase at advanced stage.	UC_Cal_IT
7.6	The system shall calculate effort adjustment factor for all phases at advanced stage.	UC_Cal_Cost_Drivers_By_Phase

7.7	The system shall estimate effort for all phases using size, effort adjustment factor of all phases and constant values (a b) of selected mode at advanced stage	UC_Est_Effort_Advanced_COCOMOI
7.8	The system shall estimate development time using efforts of all phases and constant values (c d) at intermediate stage.	UC_Est_Time_Advanced_COCOMOI
8.1	The system shall allow user to rate Precedence (PREC) as very low, low, nominal, high, very high, extra high.	UC_Cal_PREC
8.2	The system shall allow user to rate Development/Flexibility (FLEX) as very low, low, nominal, high, very high, extra high.	UC_Cal_FLEX
8.3	The system shall allow user to rate Architecture/Risk Resolution (RESL) as very low, low, nominal, high, very high, extra high.	UC_Cal_RESL
8.4	The system shall allow user to rate Team Cohesion (TEAM) as very low, low, nominal, high, very high, extra high.	UC_Cal_TEAM
8.5	The system shall allow user to rate Process Maturity (PMAT) as very low, low, nominal, high, very high, extra high.	UC_Cal_PMAT
9.1	The system shall allow user to select the model of COCOMO II.	UC_Select_Model_COCOMOII
9.2	The system shall allow user to enter the no of screens in their project at application composite model.	UC_Cal_Screens
9.3	The system shall allow user to enter the no of reports in their project at application composite model.	UC_Cal_Reports
9.4	The system shall calculate object point by taking no of reports and no of screens at application composite model.	UC_Cal_New_Object_Point
9.5	The system shall allow user to select developer's skill as very high, high, nominal, low, very low to calculate productivity rate.	UC_Cal_Productivity_Rate
9.6	The system shall calculate scale factor using Precedence, Development/Flexibility, Architecture/Risk Resolution, Team Cohesion, and Process Maturity.	UC_Cal_Scale_Factor
9.7	The system shall estimate effort by taking object point and productivity rate at application composite model.	UC_Est_Effort_ACN_COCOMOII

## Software Project Estimator

10.1	The system shall allow user to select programming language to calculate language factor.	UC_Cal_Language_Factor
11.1	The system shall estimate the size using unadjusted function point, value adjustment factors and language factor in KLOC at early design model.	UC_Est_Size_EDM_COCOMOII
11.2	The system shall calculate effort adjustment factor using product, platform, personnel and project attributes rating weights at early design model.	UC_Cal_Cost_Drivers
11.3	The system shall estimate effort using size, effort adjustment factor and a constant value 2.45 at early design model. (Effort = 2.45 * Size * EAF)	UC_Est_Effort_EDM_COCOMOII
12.1	The system shall calculate function point at post architecture model.	UC_Cal_Function_Point
12.2	The system shall estimate the size using unadjusted function point, value adjustment factors and language factor in KLOC at post architectural model.	UC_Est_Size_PAM_COCOMOII
12.3	The system shall calculate effort adjustment factor using product, platform, personnel and project attributes rating weights at post architectural model.	UC_Cal_Cost_Drivers
12.4	The system shall calculate scale factor using Precedence, Development/Flexibility, Architecture/Risk Resolution, Team Cohesion, and Process Maturity.	UC_Cal_Scale_Factor
12.5	The system shall estimate effort for Management, Requirement, Design, Implementation, and Deployment phases at post architectural model.	UC_Est_Effort_PAM_COCOMOII
12.6	The system shall estimate development time for Management, Requirement, Design, Implementation, and Deployment phases at post architectural model.	UC_Est_Time_PAM_COCOMOII
13.1	The system shall allow user to select their project type at software life cycle management (SLIM) model.	UC_Select_Application_Type
13.2	The system shall estimate the size using unadjusted function point, value adjustment factors in LOC at software life cycle management (SLIM) model.	UC_Est_Size_SLIM
13.3	The system shall take development time 12 months at software life cycle management (SLIM) model.	US_Take_Development_Time

13.4	The system shall estimate the total effort (B) using size, development time and application type weight at software life cycle management (SLIM) model.	UC_Est_Total_Effort_SLIM
13.5	The system shall estimate the effort at delivery time using total effort and constant value 0.393513 at software life cycle management (SLIM) model.	UC_Est_Delivery_Effort_SLIM
13.6	The system shall calculate the Process Productivity of project	UC_Cal_Process_Productivity_SLIM
13.7	The system shall calculate the manpower acceleration of project.	UC_Cal_Manpower_Equation_SLIM
14.1	The system shall calculate cost by using time and effort.	UC_Cal_Cost
15.1	The system shall allow user to register himself.	UC_Registration
15.2	The system shall allow user to login.	UC_Login
16.1	The system shall allow user to generate report of each technique.	UC_Genrate_Report
16.2	The system shall allow user to print report.	UC_Print_Report
16.3	The system shall allow user to save report.	UC_Save_Report
16.4	The system shall allow user to view report.	UC_View_Report
16.5	The system shall allow user to delete report.	UC_Delete_Report

#### 2.1.10. Prioritize Requirements:

Allocated requirements have been ranked according to their priority in the Software Project Estimator as highest, medium or lowest as shown in Table 2.4.

**Table 2. 4: Requirement Prioritization**

Para #	Rank	Initial Requirements	Use Case Id	Use Case Name
1.1	Lowest	The system shall allow user to enter the project name.	UC-1	UC_Enter_Project_Name
1.2	Highest	The system shall allow user to move best estimation technique by selecting a process model.	UC-2	UC_Select_Process_Model
2.1	Highest	The system shall allow user to enter External Inputs (EI) weight as low, average, high and multiply with constants as 3, 4, 6 respectively.	UC-3	UC_Cal_EI

## Software Project Estimator

2.2	Highest	The system shall allow user to enter External Outputs (EO) weight as low, average, high and multiply with constants as 4, 5, 7 respectively.	UC-4	UC_Cal_EO
2.3	Highest	The system shall allow user to enter External Enquiries (EQ) weight as low, average, high and multiply with constants as 3, 4, 6 respectively.	UC-5	UC_Cal_EQ
2.4	Highest	The system shall allow user to enter External Interface Files (EIF) weight as low, average, high and multiply with constants as 5, 7, 10 respectively.	UC-6	UC_Cal{EIF}
2.5	Highest	The system shall allow user to enter Internal Logical Files (ILF) weight as low, average, high and multiply with constants as 7, 10, 15 respectively.	UC-7	UC_Cal_ILF
2.6	Highest	The system shall calculate unadjusted function point using EI, EO, EQ, EIF, and ILF.	UC-8	UC_Cal_Unadjusted_Function_Point
3.1	Highest	The system shall allow user to rate 14 value adjustment factors.	UC-9	UC_Cal_Value_Adjustment_Factor
4.1	Highest	The system shall allow user to rate product attributes RELY, DATA, and CPLX as extra high, very high, high, nominal, low, very low.	UC-10	UC_Cal_Product_Attributes
4.2	Highest	The system shall allow user to rate platform attributes TIME, STOR, VIRT, and TURN as extra high, very high, nominal, low, very low.	UC-11	UC_Cal_Platform_Attributes

## Software Project Estimator

4.3	Highest	The system shall allow user to rate personnel attributes ACAP, AEXP, PCAP, VEXP, and LEXP as extra high, very high, nominal, low, very low.	UC-12	UC_Cal_Personnel_Attributes
4.4	Highest	The system shall allow user to rate project attributes MODP, TOOL, and SCED as extra high, very high, nominal, low, very low.	UC-13	UC_Cal_Project_Attributes
5.1	Medium	The system shall allow user to select the project mode of COCOMO I.	UC-14	UC_Select_Mode_COCOMOI
5.2	Medium	The system shall allow user to select the stage of COCOMO I.	UC-15	UC_Select_Stage_COCOMOI
5.3	Highest	The system shall estimate the size using unadjusted function point and value adjustment factors in KLOC at basic stage.	UC-16	UC_Est_Size_Basic_COCOMOI
5.4	Medium	The system shall take effort adjustment factor as 1 at basic stage.	UC-17	UC_Cal_Cost_Drivers
5.5	Highest	The system shall estimate effort using size, effort adjustment factor and constant values (a b) of selected mode at basic stage.	UC-18	UC_Est_Effort_Basic_COCOMOI
6.1	Highest	The system shall estimate the size using unadjusted function point and value adjustment factors in KLOC at intermediate stage.	UC-19	UC_Est_Size_Intermediate_COCOMOI

## Software Project Estimator

6.2	Highest	The system shall calculate effort adjustment factor using product, platform, personnel and project attributes rating weights intermediate stage.	UC-20	UC_Cal_Cost_Drivers
6.3	Highest	The system shall estimate effort using size, effort adjustment factor and constant values (a b) of selected mode at intermediate stage.	UC-21	UC_Est_Effort_Intermediate_COCOMOI
6.4	Highest	The system shall estimate development time using effort and constant values (c d) at intermediate stage.	UC-22	UC_Est_Time_Intermediate_COCOMOI
7.1	Highest	The system shall estimate the size using unadjusted function point and value adjustment factors in KLOC at advanced stage.	UC-23	UC_Est_Size_Advanced_COCOMOI
7.2	Medium	The system shall allow user to rate product, platform, personnel, project attributes for requirement planning and product design phase at advanced stage.	UC-24	UC_Cal_RPD
7.3	Medium	The system shall allow user to rate product, platform, personnel, project attributes for detailed design phase at advanced stage.	UC-25	UC_Cal_DD
7.4	Medium	The system shall allow user to rate product attributes for code and unit test phase at advanced stage.	UC-26	UC_Cal_CUT

## Software Project Estimator

7.5	Medium	The system shall allow user to rate product, platform, personnel, project attributes for integration and test phase at advanced stage.	UC-27	UC_Cal_IT
7.6	Highest	The system shall calculate effort adjustment factor for all phases at advanced stage.	UC-28	UC_Cal_Cost_Drivers_By_Phase
7.7	Highest	The system shall estimate effort for all phases using size, effort adjustment factor of all phases and constant values (a b) of selected mode at advanced stage	UC-29	UC_Est_Effort_Advanced_COCOMOI
7.8	Highest	The system shall estimate development time using efforts of all phases and constant values (c d) at intermediate stage.	UC-30	UC_Est_Time_Advanced_COCOMOI
8.1	Medium	The system shall allow user to rate Precedence (PREC) as very low, low, nominal, high, very high, extra high.	UC-31	UC_Cal_PREC
8.2	Medium	The system shall allow user to rate Development/Flexibility (FLEX) as very low, low, nominal, high, very high, extra high.	UC-32	UC_Cal_FLEX
8.3	Medium	The system shall allow user to rate Architecture/Risk Resolution (RESL) as very low, low, nominal, high, very high, extra high.	UC-33	UC_Cal_RESL
8.4	Medium	The system shall allow user to rate Team Cohesion (TEAM) as very low, low, nominal, high, very high, extra high.	UC-34	UC_Cal_TEAM

## Software Project Estimator

8.5	Medium	The system shall allow user to rate Process Maturity (PMAT) as very low, low, nominal, high, very high, extra high.	UC-35	UC_Cal_PMAT
9.1	Medium	The system shall allow user to select the model of COCOMO II.	UC-36	UC_Select_Model_COCOMOII
9.2	Medium	The system shall allow user to enter the no of screens in their project at application composite model.	UC-37	UC_Cal_Screens
9.3	Medium	The system shall allow user to enter the no of reports in their project at application composite model.	UC-38	UC_Cal_Reports
9.4	Highest	The system shall calculate object point by taking no of reports and no of screens at application composite model.	UC-39	UC_Cal_New_Object_Point
9.5	Medium	The system shall allow user to select developer's skill as very high, high, nominal, low, very low to calculate productivity rate.	UC-40	UC_Cal_Productivity_Rate
9.6	Highest	The system shall calculate scale factor using Precedence, Development/Flexibility, Architecture/Risk Resolution, Team Cohesion, and Process Maturity.	UC-41	UC_Cal_Scale_Factor
9.7	Highest	The system shall estimate effort by taking object point and productivity rate at application composite model.	UC-42	UC_Est_Effort_ACM_COCOMOII

## Software Project Estimator

10.1	Medium	The system shall allow user to select programming language to calculate language factor.	UC-43	UC_Cal_Language_Factor
11.1	Highest	The system shall estimate the size using unadjusted function point, value adjustment factors and language factor in KLOC at early design model.	UC-44	UC_Est_Size_EDM_COCOMOII
11.2	Medium	The system shall calculate effort adjustment factor using product, platform, personnel and project attributes rating weights at early design model.	UC-45	UC_Cal_Cost_Drivers
11.3	Highest	The system shall estimate effort using size, effort adjustment factor and a constant value 2.45 at early design model. (Effort = 2.45 * Size * EAF)	UC-46	UC_Est_Effort_EDM_COCOMOII
12.1	Medium	The system shall calculate function point at post architecture model.	UC-47	UC_Cal_Function_Point
12.2	Highest	The system shall estimate the size using unadjusted function point, value adjustment factors and language factor in KLOC at post architectural model.	UC-48	UC_Est_Size_PAM_COCOMOII
12.3	Medium	The system shall calculate effort adjustment factor using product, platform, personnel and project attributes rating weights at post architectural model.	UC-49	UC_Cal_Cost_Drivers

## Software Project Estimator

12.4	Medium	The system shall calculate scale factor using Precedence, Development/Flexibility, Architecture/Risk Resolution, Team Cohesion, and Process Maturity.	UC-50	UC_Cal_Scale_Factor
12.5	Highest	The system shall estimate effort for Management, Requirement, Design, Implementation, and Deployment phases at post architectural model.	UC-51	UC_Est_Effort_PAM_COCOMOII
12.6	Highest	The system shall estimate development time for Management, Requirement, Design, Implementation, and Deployment phases at post architectural model.	UC-52	UC_Est_Time_PAM_COCOMOII
13.1	Medium	The system shall allow user to select their project type at software life cycle management (SLIM) model.	UC-53	UC_Select_Application_Type
13.2	Highest	The system shall estimate the size using unadjusted function point, value adjustment factors in LOC at software life cycle management (SLIM) model.	UC-54	UC_Est_Size_SLIM
13.3	Highest	The system shall take development time 12 months at software life cycle management (SLIM) model.	UC-55	US_Take_Development_Time
13.4	Highest	The system shall estimate the total effort (B) using size, development time and application type weight at software life cycle management (SLIM) model.	UC-56	UC_Est_Total_Effort_SLIM

13.5	Highest	The system shall estimate the effort at delivery time using total effort and constant value 0.393513 at software life cycle management (SLIM) model.	UC-57	UC_Est_Delivery_Effort_SLIM
13.6	Medium	The system shall calculate the Process Productivity of project	UC-58	UC_Cal_Process_Productivity_SLIM
13.7	Medium	The system shall calculate the manpower acceleration of project.	UC-59	UC_Cal_Manpower_Equation_SLIM
14.1	Highest	The system shall calculate cost by using time and effort.	UC-60	UC_Cal_Cost
15.1	Highest	The system shall allow user to register himself.	UC-61	UC_Registration
15.2	Highest	The system shall allow user to login.	UC-62	UC_Login
16.1	Highest	The system shall allow user to generate report of each technique.	UC-63	UC_Genrate_Report
16.2	Medium	The system shall allow user to print report.	UC-64	UC_Print_Report
16.3	Medium	The system shall allow user to save report.	UC-65	UC_Save_Report
16.4	Medium	The system shall allow user to view report.	UC-66	UC_View_Report
16.5	Medium	The system shall allow user to delete report.	UC-67	UC_Delete_Report

### 2.1.11. Requirements Trace-ability Matrix:

In Trace-ability Matrix, requirements have been categorized in Table 2.5.

**Table 2. 5: Requirements Trace-Ability Matrix**

Sr#	Para #	System Specification Text	Build	Use Case Name	Category
1	1.1	The system shall allow user to enter the project name.	B1	UC_Enter_Project_Name	Business
2	1.2	The system shall allow user to move best estimation technique by selecting a process model.	B2	UC_Select_Process_Model	Business

## Software Project Estimator

3	2.6	The system shall calculate unadjusted function point using EI, EO, EQ, EIF, and ILF.	B3	UC_Cal_Unadjusted_Function_Point	Business
4	3.1	The system shall allow user to rate 14 value adjustment factors.	B4	UC_Cal_Value_Adjustment_Factor	Business
5	5.3	The system shall estimate the size using unadjusted function point and value adjustment factors in KLOC at basic stage.	B4	UC_Est_Size_Basic_COCOMOI	Business
6	5.4	The system shall take effort adjustment factor as 1 at basic stage.	B6	UC_Cal_Cost_Drivers	Business
7	5.5	The system shall estimate effort using size, effort adjustment factor and constant values (a b) of selected mode at basic stage.	B7	UC_Est_Effort_Basic_COCOMOI	Business
8	6.1	The system shall estimate the size using unadjusted function point and value adjustment factors in KLOC at intermediate stage.	B8	UC_Est_Size_Intermediate_COCOMOI	Business
9	6.3	The system shall estimate effort using size, effort adjustment factor and constant values (a b) of selected mode at intermediate stage.	B9	UC_Est_Effort_Intermediate_COCOMOI	Business
10	6.4	The system shall estimate development time using effort and constant values (c d) at intermediate stage.	B10	UC_Est_Time_Intermediate_COCOMOI	Business
11	7.1	The system shall estimate the size using unadjusted function point and value adjustment factors in KLOC at advanced stage.	B11	UC_Est_Size_Advanced_COCOMOI	Business

## Software Project Estimator

12	7.6	The system shall calculate effort adjustment factor for all phases at advanced stage.	B12	UC_Cal_Cost_Drivers_By_Phase	Business
13	7.7	The system shall estimate effort for all phases using size, effort adjustment factor of all phases and constant values (a b) of selected mode at advanced stage	B13	UC_Est_Effort_Advanced_COCOMOI	Business
14	7.8	The system shall estimate development time using efforts of all phases and constant values (c d) at intermediate stage.	B14	UC_Est_Time_Advanced_COCOMOI	Business
15	9.4	The system shall calculate object point by taking no of reports and no of screens at application composite model.	B15	UC_Cal_New_Object_Point	Business
16	9.5	The system shall allow user to select developer's skill as very high, high, nominal, low, very low to calculate productivity rate.	B16	UC_Cal_Productivity_Rate	Business
17	9.6	The system shall calculate scale factor using Precedence, Development/Flexibility, Architecture/Risk Resolution, Team Cohesion, and Process Maturity.	B17	UC_Cal_Scale_Factor	Business
18	9.7	The system shall estimate effort by taking object point and productivity rate at application composite model.	B18	UC_Est_Effort_ACN_COCOMOII	Business

## Software Project Estimator

19	10.1	The system shall allow user to select programming language to calculate language factor.	B19	UC_Cal_Language_Factor	Business
20	11.1	The system shall estimate the size using unadjusted function point, value adjustment factors and language factor in KLOC at early design model.	B20	UC_Est_Size_EDM_COCOMOII	Business
21	11.3	The system shall estimate effort using size, effort adjustment factor and a constant value 2.45 at early design model. (Effort = 2.45 * Size * EAF)	B21	UC_Est_Effort_EDM_COCOMOII	Business
22	12.1	The system shall calculate function point at post architecture model.	B22	UC_Cal_Function_Point	Business
23	12.2	The system shall estimate the size using unadjusted function point, value adjustment factors and language factor in KLOC at post architectural model.	B23	UC_Est_Size_PAM_COCOMOII	Business
24	12.4	The system shall calculate scale factor using Precedented, Development/Flexibility, Architecture/Risk Resolution, Team Cohesion, and Process Maturity.	B24	UC_Cal_Scale_Factor	Business
25	12.5	The system shall estimate effort for Management, Requirement, Design, Implementation, and Deployment phases at post architectural model.	B25	UC_Est_Effort_PAM_COCOMOII	Business

## Software Project Estimator

26	12.6	The system shall estimate development time for Management, Requirement, Design, Implementation, and Deployment phases at post architectural model.	B26	UC_Est_Time_PAM_COCOMOII	Business
27	13.1	The system shall allow user to select their project type at software life cycle management (SLIM) model.	B27	UC_Select_Application_Type	Business
28	13.2	The system shall estimate the size using unadjusted function point, value adjustment factors in LOC at software life cycle management (SLIM) model.	B28	UC_Est_Size_SLIM	Business
29	13.3	The system shall take development time 12 months at software life cycle management (SLIM) model.	B29	US_Take_Development_Time	Business
30	13.4	The system shall estimate the total effort (B) using size, development time and application type weight at software life cycle management (SLIM) model.	B30	UC_Est_Total_Effort_SLIM	Business
31	13.5	The system shall estimate the effort at delivery time using total effort and constant value 0.393513 at software life cycle management (SLIM) model.	B31	UC_Est_Delivery_Effort_SLIM	Business
32	13.6	The system shall calculate the Process Productivity of project	B32	UC_Cal_Process_Productivity_SLIM	Business
33	13.7	The system shall calculate the manpower acceleration of project.	B33	UC_Cal_Manpower_Equation_SLIM	Business
34	14.1	The system shall calculate cost by using time and effort.	B34	UC_Cal_Cost	Business

36	15.1	The system shall allow user to generate report of each technique.	B36	UC_Genrate_Report	Business
----	------	---	-----	-------------------	----------

### **2.1.12. High Level Use Case Diagram:**

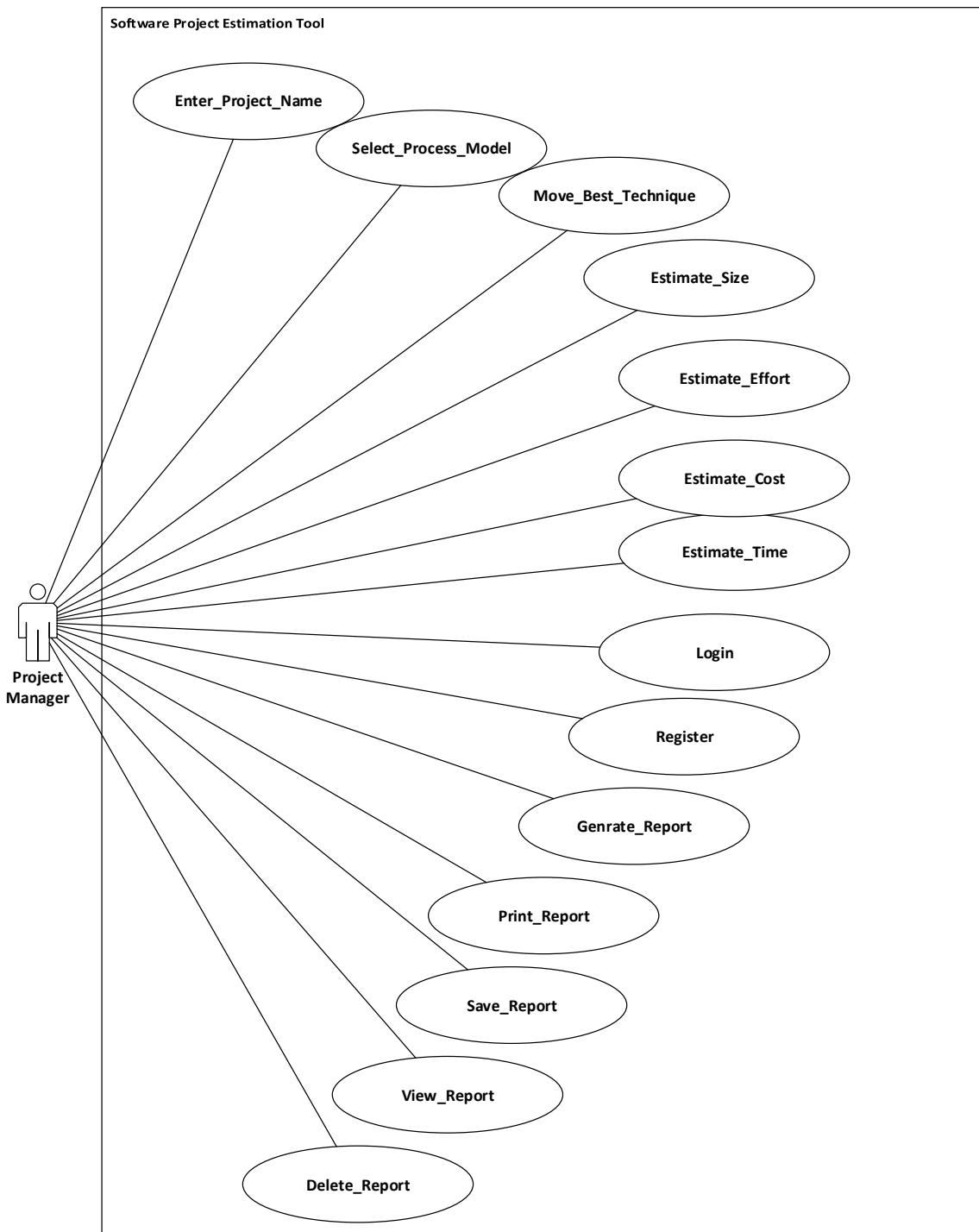
The high-level use case diagram shows the working of whole system. It includes one actor that is involved in the system. The purpose of this diagram is to demonstrate how objects will interact with Software Project Estimator and map out the basic functionality of the system. A use case scenario is a visual description, typically written in structured English or point form, of a potential business situation that a system may or may not be able to handle. A use case defines a goal-oriented set of interactions between external actors and the system under consideration.

The use case of our purposed system involves one actor.

- Project Manager

High level Use case diagram describes the overall working of the tool as shown in figure.

## Software Project Estimator

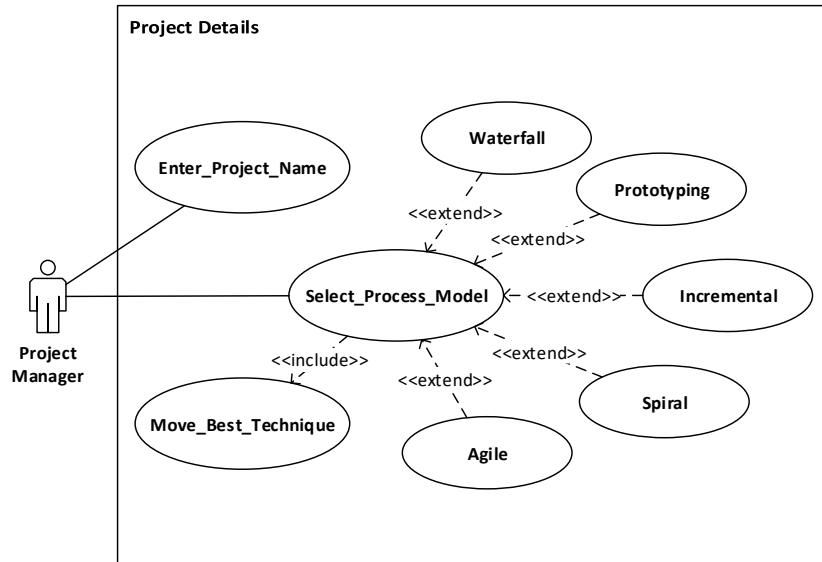


**Figure 2. 4: High Level Use Case Diagram**

### 2.1.13. Analysis Level Use Case Diagram:

Analysis level use case diagram is actually the explanation of high level use cases diagram. In this diagram high level use cases are expanded in a way that exhibit how high-level use cases will reach to their functionality. Analysis level use case diagram explains the detailed working of tool as shown in Figure 2.1.

### 2.1.13.1. Project Details



**Figure 2.5. 1: Project Details Module**

### 2.1.13.2. COCOMO I

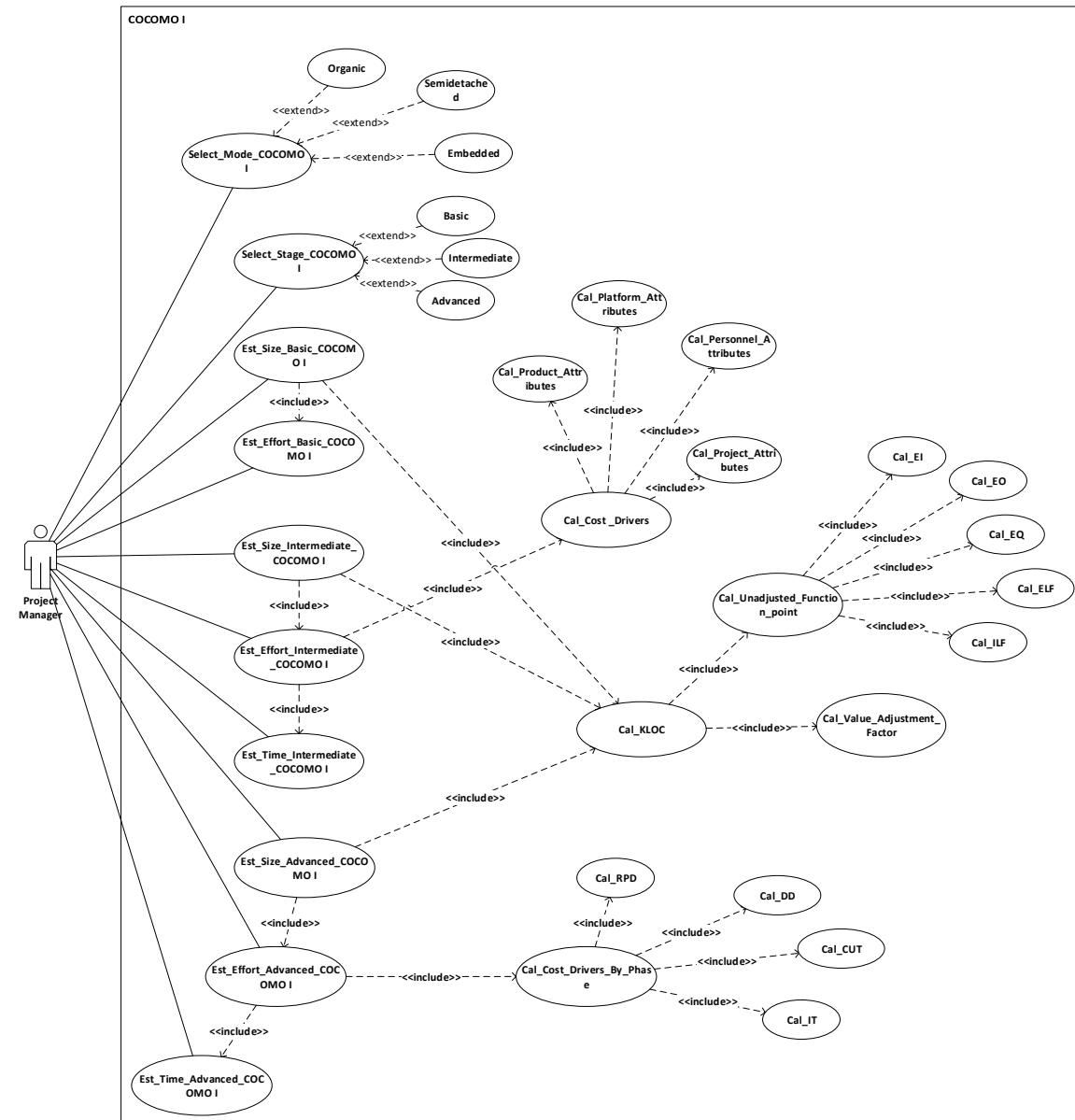
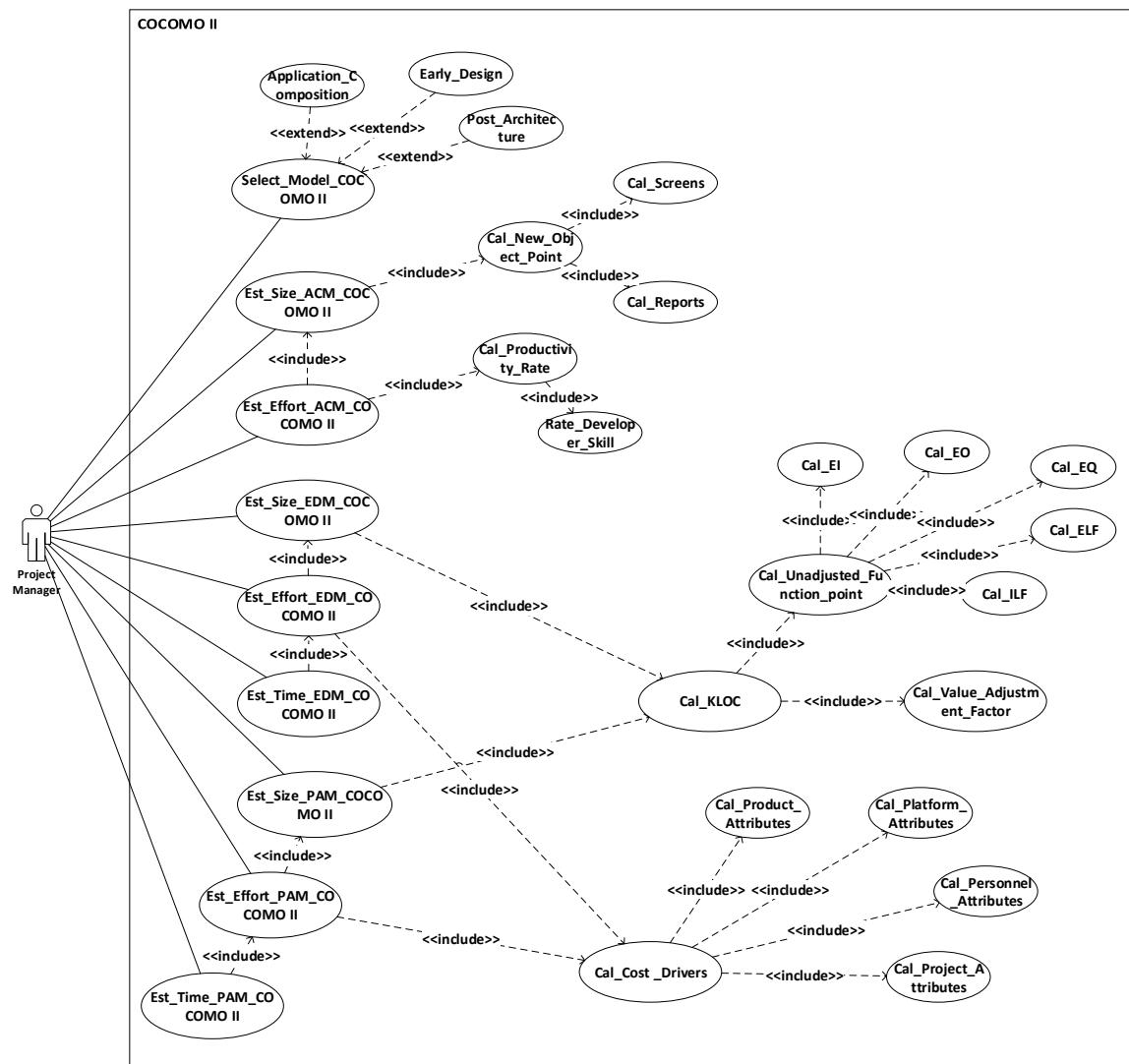


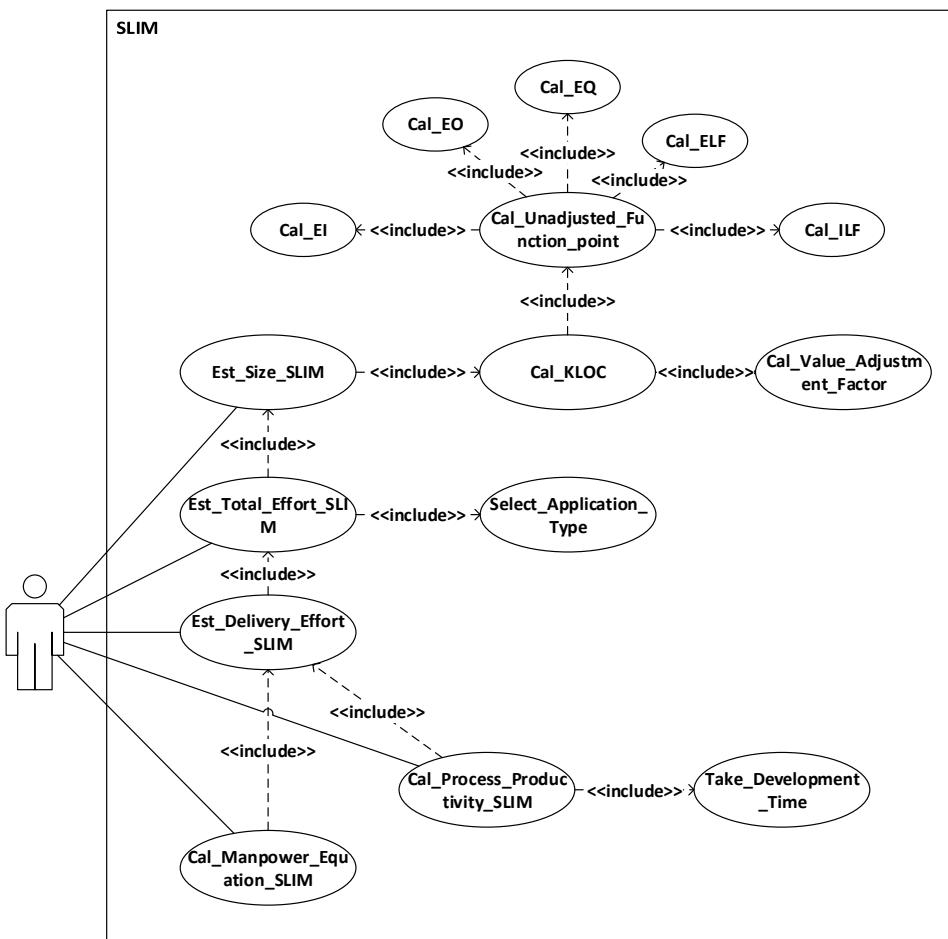
Figure 2.5. 2: COCOMO I Module

### 2.1.13.3. COCOMO II



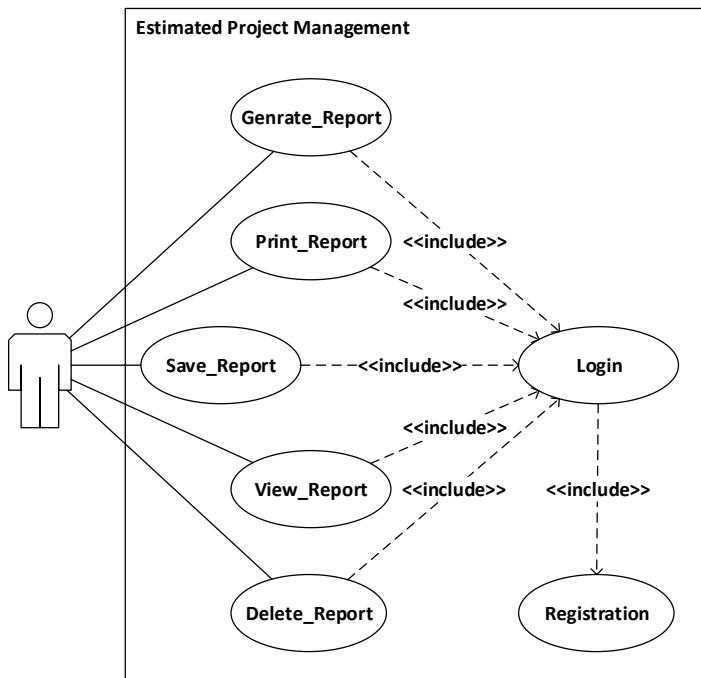
**Figure 2.5. 3: COCOMO II Module**

#### 2.1.13.4. SLIM



**Figure 2.5. 4: SLIM Module**

### 2.1.13.5. Estimated Project Management



**Figure 2.5. 5: Estimated Project Management Module**

### 2.2.14. Use case Description

While technically not part of UML, use case documents are closely related to UML use cases. A use case document is text that captures the detailed functionality of a use case. The use case description of Software Project Estimator is follows:

Table 2.2.1. Use case for Project Name

<b>Use Case ID:</b>	UC-1
<b>Use Case Name:</b>	Enter_Project_Name
<b>Actors:</b>	User (Project Manager or Team lead)
<b>Description:</b>	The user enters the project name to know the values of the specific project.
<b>Trigger:</b>	N/A
<b>Preconditions:</b>	User visits the website.
<b>Post conditions:</b>	Project mode will be selected leading to size estimation.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User visits website.</li> <li>2. User enters the project name.</li> </ol>
<b>Alternative Flows:</b>	N/A
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. Webpage closes unexpectedly.</li> <li>2. Internet connection lost.</li> </ol>
<b>Includes:</b>	N/A
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>1. Supportability: User will get proper help about each model by the user manual and the help provided by the system.</li> <li>2. Accuracy: User will get accurate information of the selected model.</li> <li>3. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.</li> </ol>
<b>Assumptions:</b>	N/A

<b>Notes and Issues:</b>	N/A
--------------------------	-----

Table 2.2.2. Use Case for Process Model

<b>Use Case ID:</b>	UC-2
<b>Use Case Name:</b>	Select_Process_Model
<b>Actors:</b>	User (Project Manager or Team lead)
<b>Description:</b>	To select the design process model of the project which will automatically lead the user to one of the estimation technique i.e. COCOMO I, COCOMO II or SLIM.
<b>Trigger:</b>	User selects one design process model out of the options available.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. User visits the website.</li> <li>2. User selects any one model according to the project.</li> </ol>
<b>Post conditions:</b>	Project mode will be selected leading to size estimation.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User visits website.</li> <li>2. User enter the project name.</li> <li>3. User selects the design process model of his project out of options that are available.</li> <li>4. User will be redirected to the estimation technique according to the process model selected.</li> </ol>
<b>Alternative Flows:</b>	N/A
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. Webpage closes unexpectedly.</li> <li>2. Internet connection lost.</li> </ol>
<b>Includes:</b>	N/A
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>1. Supportability: User will get proper help about each model by the user manual and the help provided by the system.</li> <li>2. Accuracy: User will get accurate information of the selected model.</li> <li>3. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.</li> </ol>
<b>Assumptions:</b>	Users know about the design process model of their project.
<b>Notes and Issues:</b>	N/A

Table 2.2.3. Use Case for External Input (EI)

<b>Use Case Id</b>	UC-3
<b>Use Case Name</b>	Cal_EI
<b>Actors:</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to let user enter values for external inputs. These inputs are an elementary process that processes data or control information that comes from outside the application boundary.
<b>Trigger</b>	User needs to calculate UFP.
<b>Pre conditions</b>	- Project mode should be selected. Estimation stage should be selected.

## Software Project Estimator

<b>Post-conditions</b>	Low, average, and high complexity external inputs are entered by user.
<b>Normal Flow</b>	User will enter the number of input files of low, average and high complexity.
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values are not entered. Invalid values are entered.
<b>Includes</b>	None
<b>Special Requirements</b>	None.
<b>Assumptions</b>	None
<b>Notes and Issues</b>	None

Table 2.2.4. Use Case for External Output (EO)

<b>Use Case Id</b>	UC-4
<b>Use Case Name</b>	Cal_EO
<b>Actors:</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to let user enter values for external outputs. These outputs are an elementary process that sends data or control information outside the application boundary.
<b>Trigger</b>	User needs to calculate UFP.
<b>Pre conditions</b>	- Project mode should be selected. Estimation stage should be selected.
<b>Post-conditions</b>	Low, average, and high complexity external outputs are entered by user.
<b>Normal Flow</b>	User will enter the number of output files of low, average and high complexity.
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values are not entered. Invalid values are entered.
<b>Includes</b>	None
<b>Special Requirements</b>	None
<b>Assumptions</b>	None
<b>Notes and Issues</b>	None

Table 2.2.5. Use Case for External Inquiries (EQ)

<b>Use Case Id</b>	UC-5
<b>Use Case Name</b>	Cal_EQ
<b>Actors:</b>	User (Project Manager Team Lead)

## Software Project Estimator

<b>Description</b>	The purpose of this use case is to let user enter values for external inquiries. These inquiries are an elementary process that sends data or control information outside the application boundary.
<b>Trigger</b>	When actor needs to calculate UFP.
<b>Pre conditions</b>	- Project mode should be selected. Estimation stage should be selected.
<b>Post-conditions</b>	Low, average, and high complexity external inquiries are entered by user.
<b>Normal Flow</b>	User will enter the number of inquiries files of low, average and high complexity.
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values are not entered. Invalid values are entered.
<b>Includes</b>	None
<b>Special Requirements</b>	None
<b>Assumptions</b>	None
<b>Notes and Issues</b>	None

Table 2.2.6. Use Case for External Interface Files (EIF)

<b>Use Case Id</b>	UC-6
<b>Use Case Name</b>	Cal_EIF
<b>Actors:</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to let user enter values for external interface files. These files are a user identifiable group of logically related data or control information referenced by the application but maintained within the boundary of another application.
<b>Trigger</b>	When user need to calculate UFP.
<b>Pre conditions</b>	- Project mode should be selected. Estimation stage should be selected.
<b>Post-conditions</b>	Low, average, and high complexity external interface files are entered by user.
<b>Normal Flow</b>	User will enter the number of file updating in the external database of low, average and high complexity.
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values are not entered. Invalid values are entered.
<b>Includes</b>	None
<b>Special Requirements</b>	None
<b>Assumptions</b>	None
<b>Notes and Issues</b>	None

Table 2.2.7. Use Case for Internal Logical Files (ILF)

<b>Use Case Id:</b>	UC-7
<b>Use Case Name:</b>	Cal_ILF
<b>Actor:</b>	User (Project Manager Team Lead)
<b>Description:</b>	The purpose of this use case is to let user enter values for internal logical files. These files are a user-identifiable group of logically related data or control information maintained within the boundary of the application
<b>Trigger:</b>	User needs to calculate UFP.
<b>Pre conditions</b>	- Project mode should be selected. Estimation stage should be selected.
<b>Post-conditions</b>	Low, average, and high complexity internal logical files are entered by user.
<b>Normal Flow</b>	User will enter the number of files updating in the internal database of low, average and high complexity.
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values are not entered. Invalid values are entered.
<b>Includes</b>	None
<b>Special Requirements</b>	None
<b>Assumptions</b>	None
<b>Notes and Issues</b>	None

Table 2.2.8. Use Case for Calculate Unadjusted Function Point

<b>Use Case Id:</b>	UC-8
<b>Use Case Name:</b>	Cal_Unadjusted_Function_Point
<b>Actors:</b>	User (Project Manager Team Lead)
<b>Description:</b>	To calculate the unadjusted function point that by making taking five inputs from the user that are; External inputs, external outputs, external interface files, internal logical files, and external inquiries.
<b>Trigger:</b>	To calculate function point for size estimation in KLOC.
<b>Pre conditions</b>	- Project mode should be selected.
<b>Post-conditions</b>	UFP is calculated.

<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. Calculate External Input (EI).</li> <li>2. Calculate External Output (EO).</li> <li>3. Calculate External Inquiries (EQ).</li> <li>4. Calculate External Interface Files (EIF).</li> <li>5. Calculate Internal Logical Files (ILF).</li> <li>6. Calculate the sum of five complexity weighting factor (EI, EO, EQ, EIF, and ILF) separately.</li> <li>7. <math>UFP = \text{Sum} ( EI + EO + EQ + EIF + ILF )</math></li> </ol>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values are not inserted.
<b>Includes</b>	UC-3, UC-4, UC-5, UC-6, UC-7
<b>Special Requirements</b>	<p>Accuracy: User will get the accurate results as per the values entered.</p> <p>Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.</p>
<b>Assumptions</b>	N/A
<b>Notes and Issues</b>	If EI, EO, EQ, EIF, ILF is not present, UFP cannot be calculated.

Table 2.2.9. Use Case for Calculate Value Adjustment Factor (VAF)

<b>Use Case Id</b>	UC-9
<b>Use Case Name</b>	Cal_Value_Adjustment_Factor
<b>Actor</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to calculate Value Adjustment Factors and multiply it with the unadjusted function point (UFP) to estimate the size in KLOC.
<b>Trigger</b>	Calculation of VAF is required.
<b>Pre conditions</b>	- Function points need to be calculated.
<b>Post-conditions</b>	VAF is calculated.
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. Weight Data Communication.</li> <li>2. Weight Distributed Data processing.</li> <li>3. Weight Performance.</li> <li>4. Weight Heavily Used Configuration.</li> <li>5. Weight Transaction rate.</li> <li>6. Weight On-line Date Entry.</li> <li>7. End-User Efficiency.</li> <li>8. Weight On-line Update.</li> <li>9. Weight Complex Processing.</li> <li>10. Weight Reusability.</li> <li>11. Weight Installation Ease.</li> <li>12. Weight operational Ease.</li> <li>13. Weight Multiple Site.</li> <li>14. Weight Facilitate Changes.</li> <li>15. Multiply all 14 components to get VAF value.</li> </ol>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	<p>All values are not entered.</p> <p>Invalid values are entered.</p>

<b>Includes</b>	None
<b>Special Requirements</b>	Accuracy: User will get the accurate calculation as per values entered. Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	None
<b>Notes and Issues</b>	Required value should be inserted.

Table 2.2.10. Use Case for Calculate Product Attributes

<b>Use Case Id</b>	UC-10
<b>Use Case Name</b>	Cal_Product_Attributes
<b>Actor</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to calculate Product factor.
<b>Trigger</b>	When actor need to calculate EAF.
<b>Pre conditions</b>	- EAF need to be calculated.
<b>Post-conditions</b>	Product Factor is calculated.
<b>Normal Flow</b>	Assigning weight in terms of very low, low, nominal, high, very high and extra high to factors listed below: <ul style="list-style-type: none"><li>o Reliability (RELY)</li><li>o Database Size (DATA)</li><li>o Complexity (CPLX)</li></ul>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All value in product factor should be inserted.
<b>Includes</b>	None
<b>Special Requirements</b>	Accuracy: User will get the accurate information of the selected Mode/Models Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	None
<b>Notes and Issues</b>	Factor should be correctly placed to get correct result.

Table 2.2.11. Use Case Calculate Platform Attributes

<b>Use Case Id</b>	UC-11
<b>Use Case Name</b>	Cal_Platform_Attributes
<b>Actor</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to calculate Platform factor.

## Software Project Estimator

<b>Trigger</b>	When actor need to calculate EAF.
<b>Pre conditions</b>	EAF need to calculate.
<b>Post-conditions</b>	Platform factors is calculated
<b>Normal Flow</b>	<p>Assigning weight in terms of very low, low, nominal, high, very high and extra high to factors listed below:</p> <ul style="list-style-type: none"> <li>○ Execution Time Constraint (TIME)</li> <li>○ Main Storage Constraint (STOR)</li> <li>○ Virtual Machine Volatility (VIRT)</li> <li>○ Computer Turn Around Time (TURN)</li> </ul>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All value should be inserted.
<b>Includes</b>	None
<b>Special Requirements</b>	<p>Accuracy: User will get the accurate information of the selected Mode/Models</p> <p>Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.</p>
<b>Assumptions</b>	None
<b>Notes and Issues</b>	Factor should be correctly assigned to get correct result.

Table 2.2.12. Use Case Calculate Personnel Attributes

<b>Use Case Id</b>	UC-12
<b>Use Case Name</b>	Cal_Personnel_Attributes
<b>Actor</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to calculate Personal factor.
<b>Trigger</b>	When actor need to calculate EAF.
<b>Pre conditions</b>	EAF need to calculate.
<b>Post-conditions</b>	Personal Factor is calculated.

<b>Normal Flow</b>	Assigning weight in terms of very low, low, nominal, high, very high and extra high to factors listed below: <ul style="list-style-type: none"> <li>○ Analysts Capability (ACAP)</li> <li>○ Application Experience (AEXP)</li> <li>○ Programmer Capability (PCAP)</li> <li>○ Virtual Machine Experience (VEXP)</li> <li>○ Language Experience (LEXP)</li> </ul>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values should be inserted in weights.
<b>Includes</b>	None
<b>Special Requirements</b>	Accuracy: User will get the accurate information of the selected Mode/Models Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	None
<b>Notes and Issues</b>	All value should be properly placed to get factor result, missing value may lead to errors.

Table 2.2.13. Use Case for Project Attributes

<b>Use Case Id</b>	UC-13
<b>Use Case Name</b>	Cal_Project_Attributes
<b>Actor</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to calculate Project factor.
<b>Trigger</b>	When actor need to calculate EAF.
<b>Pre conditions</b>	EAF need to be calculated.
<b>Post-conditions</b>	Project Factor is calculated.
<b>Normal Flow</b>	Assigning weight in terms of very low, low, nominal, high, very high and extra high to factors listed below: <ul style="list-style-type: none"> <li>○ Modern Programming Practices (MODP)</li> <li>○ Software Tool (TOOL)</li> <li>○ Development Schedule (SCED)</li> </ul>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values should be inserted.
<b>Includes</b>	None

<b>Special Requirements</b>	Accuracy: User will get the accurate information of the selected Mode/Models Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	None
<b>Notes and Issues</b>	All values should be properly placed to get desired result.

Table 2.2.14. Use Case for Select Mode COCOMO I

<b>Use Case ID:</b>	UC-14
<b>Use Case Name:</b>	Select_Mode_COCOMOI
<b>Actors:</b>	User (Project Manager or Team lead)
<b>Description:</b>	To select the project mode out of three modes of COCOMO I. The estimation will proceed based on the selected mode.
<b>Trigger:</b>	User selects one mode out of the three options given.
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. User enters the design process model of the project.</li> <li>2. User selects COCOMO I model for estimation of the project.</li> </ul>
<b>Post conditions:</b>	Project mode will be selected leading to size estimation.
<b>Normal Flow:</b>	<ul style="list-style-type: none"> <li>1. User selects the design process model that leads to COCOMO I model.</li> <li>2. User clicks on the required project mode out of three options.</li> <li>3. Project mode is selected.</li> </ul>
<b>Alternative Flows:</b>	User selects any other mode.
<b>Exceptions:</b>	<ul style="list-style-type: none"> <li>1. Webpage closes unexpectedly.</li> <li>2. Internet connection lost.</li> </ul>
<b>Includes:</b>	N/A
<b>Special Requirements :</b>	<ul style="list-style-type: none"> <li>1. Accuracy: User will get accurate information of the selected mode.</li> <li>2. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.</li> </ul>
<b>Assumptions:</b>	<ul style="list-style-type: none"> <li>1. The users know which design process model is suitable for their projects.</li> <li>2. The user has an understandability of the three modes.</li> </ul>
<b>Notes and Issues:</b>	User should have the knowledge of the project mode they choose.

Table 2.2.15. Use Case for Select Stage COCOMO I

<b>Use Case Id:</b>	UC-15
<b>Use Case Name:</b>	Select_Stage_COCOMOI
<b>Actors:</b>	User (Project Manager, Team Lead)
<b>Description</b>	To select the COCOMO I stage for which estimation needs to be done. The three stages are; Basic COCOMO, Intermediate COCOMO, and Advanced COCOMO.
<b>Trigger</b>	To select the stage of COCOMO I for which estimation needs to be done.
<b>Pre conditions</b>	- Project mode should be selected.

## Software Project Estimator

<b>Post-conditions</b>	COCOMO I stage is selected.
<b>Normal Flow</b>	1. User clicks on the desired stage out of the three options given. 2. Stage is selected and user proceeds to estimation process.
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values are not inserted.
<b>Includes</b>	UC-3, UC-4, UC-5, UC-6, UC-7, UC-8
<b>Special Requirements</b>	Accuracy: User will get the accurate results as per the values entered. Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	N/A
<b>Notes and Issues</b>	If EI, EO, EQ, EIF, ILF is not present, UFP cannot be calculated.

Table 2.2.16. Use Case Estimate Size Basic COCOMO I

<b>Use Case Id</b>	UC-16
<b>Use Case Name</b>	Est_Size_Basic_COCOMOI
<b>Actor</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to calculate function point that will give the estimated size in KLOC.
<b>Trigger</b>	When actor needs to calculate function points.
<b>Pre conditions</b>	1. Unadjusted function points (UFP) are calculated. 2. Value adjustment factor (VAF) is calculated.
<b>Post-conditions</b>	Function points are calculated and shown to the user.
<b>Normal Flow</b>	1. Calculate UFP. 2. Calculate VAF. 3. Click on show results button to calculate function points.
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values are not entered. Invalid values are entered.
<b>Includes</b>	UC-8, UC-9
<b>Special Requirements</b>	Accuracy: User will get the accurate results as per values entered. Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	None
<b>Notes and Issues</b>	UFP and VAF should be calculated before.

Table 2.2.17. Use Case for Estimate Effort Basic COCOMO I

<b>Use Case ID:</b>	UC-18
<b>Use Case Name:</b>	Est_Effort_Basic_COCOMOI
<b>Actors:</b>	User (Project Manager or Team lead)

<b>Description:</b>	Effort is calculated and shown to user using the formula; $a (S)^b * EAF$ <p>where 'a' and 'b' are based on the selected mode i.e., test case 2.      S is the size in KLOC found through function points i.e., test case 11.      EAF is set as 1 for this stage.</p>
<b>Trigger:</b>	User clicks on show results button.
<b>Preconditions :</b>	<ol style="list-style-type: none"> <li>Project mode is selected.</li> <li>Size in KLOC is known.</li> </ol>
<b>Post conditions:</b>	Effort in person-month is calculated.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>Calculate Basic COCOMO 1 button is clicked.</li> <li>Effort in person-month is calculated and displayed on screen.</li> </ol>
<b>Alternative Flows:</b>	N/A
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>User clicks on back button.</li> <li>Webpage closes unexpectedly.</li> <li>Internet connection lost.</li> </ol>
<b>Includes:</b>	UC-14, UC-16
<b>Special Requirements :</b>	N/A
<b>Assumptions:</b>	Size is correctly calculated.
<b>Notes and Issues:</b>	Positive values should be entered. EAF is taken as 1 for the basic model.

Table 2.2.18. Use Case Calculate Cost Drivers

<b>Use Case ID:</b>	UC-20
<b>Use Case Name:</b>	Cal_Cost_Drivers
<b>Actors:</b>	User (Project Manager or Team lead)
<b>Description:</b>	Effort adjustment factor is calculated using the effort multiplier which came as a result of the rating of 15 cost drivers. It is calculated according to the following formula: $EAF = \prod_{i=1}^{15} EM_i$
<b>Trigger:</b>	User clicks on calculate button.
<b>Preconditions :</b>	<ol style="list-style-type: none"> <li>Project mode is selected.</li> <li>Each cost driver is rated.</li> <li>Effort multiplier is known.</li> </ol>
<b>Post conditions:</b>	Effort adjustment factor is calculated.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>Effort adjustment factor button is clicked.</li> <li>Cost drivers are rated.</li> <li>EM is multiplied with the results.</li> <li>Effort adjustment factor is calculated.</li> </ol>

## Software Project Estimator

<b>Alternative Flows:</b>	N/A
<b>Exceptions:</b>	<ul style="list-style-type: none"> <li>1. Cost drivers are not rated correctly.</li> <li>2. Effort multiplier is not known.</li> <li>3. Internet connection is lost.</li> </ul>
<b>Includes:</b>	UC-10, UC-11, UC-12, UC-13
<b>Special Requirements :</b>	<ul style="list-style-type: none"> <li>1. Accuracy: User will get accurate results according to the effort multiplier.</li> <li>2. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.</li> </ul>
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	N/A

**Table 2.2.19. Use Case for Calculate Effort Intermediate COCOMO I**

<b>Use Case ID:</b>	UC-21
<b>Use Case Name:</b>	Cal_Effort_Intermediate_COCOMOI
<b>Actors:</b>	User (Project Manager or Team lead)
<b>Description:</b>	Effort is calculated using the program size and a set of cost drivers. It is calculated as follows: $E = a(KLOC)b \times EAF$
<b>Trigger:</b>	User clicks on calculate effort button.
<b>Preconditions :</b>	<ul style="list-style-type: none"> <li>1. Rate 15 cost drivers.</li> <li>2. Calculate effort multiplier based on the rating.</li> <li>3. Calculate effort adjustment factor using effort multiplier.</li> </ul>
<b>Post conditions:</b>	Effort is calculated.
<b>Normal Flow:</b>	<ul style="list-style-type: none"> <li>1. Rate 15 cost drivers.</li> <li>2. Calculate effort multiplier based on the rating.</li> <li>3. Calculate effort adjustment factor using effort multiplier.</li> <li>4. Effort is calculated.</li> </ul>
<b>Alternative Flows:</b>	N/A
<b>Exceptions:</b>	<ul style="list-style-type: none"> <li>1. Cost drivers are not rated correctly.</li> <li>2. Effort multiplier is not known.</li> <li>3. EAF is not known.</li> <li>4. Internet connection is lost.</li> </ul>
<b>Includes:</b>	UC-19, UC-20,
<b>Special Requirements :</b>	<ul style="list-style-type: none"> <li>1. Accuracy: User will get accurate results according to the effort adjustment factor.</li> <li>2. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.</li> </ul>
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	N/A

**Table 2.2.20. Use Case for Estimate Time Intermediate COCOMOI**

<b>Use Case ID:</b>	UC-22
---------------------	-------

## Software Project Estimator

<b>Use Case Name:</b>	Est_Time_Intermediate_COCOMOI
<b>Actors:</b>	User (Project Manager or Team lead)
<b>Description:</b>	Development time is calculated using the following formula: $cE^d$ where c and d are based on the selected project mode. E is the effort calculated.
<b>Trigger:</b>	User clicks on show results button.
<b>Preconditions :</b>	<ol style="list-style-type: none"> <li>1. Rate 15 cost drivers.</li> <li>2. Calculate effort multiplier based on the rating.</li> <li>3. Calculate effort adjustment factor using effort multiplier.</li> <li>4. Click on show results button.</li> </ol>
<b>Post conditions:</b>	Development time is calculated.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Rate 15 cost drivers.</li> <li>2. Calculate effort multiplier based on the rating.</li> <li>3. Calculate effort adjustment factor using effort multiplier.</li> <li>4. Effort is calculated.</li> </ol>
<b>Alternative Flows:</b>	N/A
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. Cost drivers are not rated correctly.</li> <li>2. Effort multiplier is not known.</li> <li>3. EAF is not known.</li> <li>4. Internet connection is lost.</li> </ol>
<b>Includes:</b>	UC-21
<b>Special Requirements :</b>	<p>Accuracy: User will get accurate results according to the effort adjustment factor and effort.</p> <p>Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.</p>
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	N/A

Table 2.2.2. Use Case for Calculate Cost Drivers by Phase

<b>Use Case ID:</b>	UC-28
<b>Use Case Name:</b>	Cal_Cost_Drivers_By_Phase
<b>Actors:</b>	User (Project Manager or Team lead)
<b>Description:</b>	To rate the 15 cost drivers again into very high, high, nominal, low, very low. Each cost driver is broken down by phase. These phases are requirements, planning and product design (RPD), detailed design (DD), code and unit test (CUT), and integration and test (IT).
<b>Trigger:</b>	User clicks on the button of Rate Cost Drivers in Phases.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Cost drivers are rated in intermediate mode.</li> <li>2. User rates the cost drivers again according to the four phases that are; requirements, planning and product design (RPD), detailed design (DD), code and unit test (CUT), and integration and test (IT). The rating is done from the options of very high, high, nominal, low, and very low.</li> </ol>

<b>Post conditions:</b>	The cost drivers are rated according to the four phases of advanced model.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User rates the cost drivers from the options of very high, high, nominal, low, and very low.</li> <li>2. Cost drivers are rated.</li> </ol>
<b>Alternative Flows:</b>	N/A
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. Cost drivers are not rated correctly.</li> <li>2. Internet connection is lost.</li> </ol>
<b>Includes:</b>	UC-24, UC-25, UC-26, UC-27
<b>Special Requirements :</b>	1. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	N/A

Table 2.2.22. Use Case for Estimate Effort Advanced COCOMO I

<b>Use Case ID:</b>	UC-29
<b>Use Case Name:</b>	Est_Effort_Advanced_COCOMOI
<b>Actors:</b>	User (Project Manager or Team lead)
<b>Description:</b>	Effort is calculated using the program size and a set of cost drivers weighted according to each phase of software lifecycle. It is calculated as follows: $E = a(KLOC) \times b \times EAF$
<b>Trigger:</b>	User clicks on calculate effort button.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Rate 15 cost drivers.</li> <li>2. Calculate effort multiplier based on the rating.</li> <li>3. Calculate effort adjustment factor using effort multiplier.</li> <li>4. User rates the cost drivers again according to the four phases that are; requirements, planning and product design (RPD), detailed design (DD), code and unit test (CUT), and integration and test (IT). The rating is done from the options of very high, high, nominal, low, and very low.</li> </ol>
<b>Post conditions:</b>	Effort is calculated.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Rate 15 cost drivers.</li> <li>2. Calculate effort multiplier based on the rating.</li> <li>3. Calculate effort adjustment factor using effort multiplier.</li> <li>4. User rates the cost drivers again according to the four phases that are; requirements, planning and product design (RPD), detailed design (DD), code and unit test (CUT), and integration and test (IT). The rating is done from the options of very high, high, nominal, low, and very low.</li> <li>5. Effort is calculated.</li> </ol>
<b>Alternative Flows:</b>	N/A

## Software Project Estimator

<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. Cost drivers are not rated correctly.</li> <li>2. Effort multiplier is not known.</li> <li>3. EAF is not known.</li> <li>4. Internet connection is lost.</li> </ol>
<b>Includes:</b>	UC-28
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>1. Accuracy: User will get accurate results according to the effort adjustment factor.</li> <li>2. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.</li> </ol>
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	N/A

Table 2.2.23. Use Case for Select Model COCOMO II

<b>Use Case ID:</b>	UC-36
<b>Use Case Name:</b>	Select_Model_COCOMOII
<b>Actors:</b>	User (Project Manager or Team lead)
<b>Description:</b>	To select the project model out of three models of COCOMO II. The estimation will proceed based on the selected model.
<b>Trigger:</b>	User selects one model out of the three options given.
<b>Preconditions :</b>	User enters the design process model of the project. User selects COCOMO II model for estimation of the project.
<b>Post conditions:</b>	Project model will be selected leading to size estimation.
<b>Normal Flow:</b>	User selects the design process model that leads to COCOMO II models. User clicks on the required project model out of three options. Project model is selected.
<b>Alternative Flows:</b>	User selects any other model.
<b>Exceptions:</b>	Webpage closes unexpectedly. Internet connection lost.
<b>Includes:</b>	N/A
<b>Special Requirements:</b>	Accuracy: User will get accurate information of the selected mode. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.
<b>Assumptions:</b>	The users know which design process model is suitable for their projects. The user has an understandability of the three models.
<b>Notes and Issues:</b>	User should have the knowledge of the COCOMO II models they choose.

Table 2.2.24. Use Case Calculate New Object Point

<b>Use Case Id:</b>	UC-39
<b>Use Case Name:</b>	Cal_Calculate_New_Object_Point
<b>Actors:</b>	User (Project Manager Team Lead)

<b>Description</b>	The purpose of this use case is to calculate initial measure for size in the early development cycle. It will take no of screens and reports and perform calculations.
<b>Trigger</b>	Actor need to calculate object point.
<b>Pre conditions</b>	- Screen or reports should be there.
<b>Post-conditions</b>	Object point should be calculated.
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. Estimate the number of “screens” of the system and the approximate number of data entries for each screen.</li> <li>2. Estimate the number of “reports” that will be generated by the system (Including any output file, writing to database, etc.) And the approximate number of data entries for each report.</li> <li>3. For each “screen” and “report” and their corresponding views, use table to determine whether that the screen or report is “simple”, “medium” or “difficult”.</li> <li>4. Weight the screens and reports using the weight table and sum them up to get the OP number.</li> <li>5. If acquired component, give it the weight of 10 and add it to the OP.</li> <li>6. If any part of system is reused from a previous generation, define a percentage of reuse and then adjust the value of NOP accordingly.</li> </ol>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values are not inserted.
<b>Includes</b>	UC-37, UC-38
<b>Special Requirements</b>	Accuracy: User will get the accurate information of the selected Mode/Models Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	Actor have understanding of screen and reports.
<b>Notes and Issues</b>	If the user does not have reports and screen no they won't be able to calculate object point.

Table 2.2.25. Use Case Calculate Productivity Rate

<b>Use Case Id:</b>	UC-40
<b>Use Case Name:</b>	Cal_Productivity_Rate
<b>Actors:</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to calculate productivity rate.
<b>Trigger</b>	When actor require to calculate effort (per-month)
<b>Pre conditions</b>	- None
<b>Post-conditions</b>	Productivity rate is calculated successfully.

<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. Estimate low, low, average, high and very high skill of developer and assign weight to it.</li> <li>2. Select any one value from very low, low, average, high and high weight that will be the productivity rate.</li> <li>3. Also add the percentage if any part of system is reused from a previous generation, define a percentage of reuse.</li> </ol>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values are not inserted.
<b>Includes</b>	None
<b>Special Requirements</b>	Accuracy: User will get the accurate information of the selected Mode/Models Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	None
<b>Notes and Issues</b>	If the weights are not given correctly or out of range, wrong productivity rate will be calculated.

Table 2.2.26. Use Case Calculate Scale Factor

<b>Use Case Id</b>	UC-41
<b>Use Case Name</b>	Cal_Scale_Factor
<b>Actor</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to calculate Scale factor.
<b>Trigger</b>	When actor need to calculate SF.
<b>Pre conditions</b>	- Require to evaluate project in Post Architectural Model.
<b>Post-conditions</b>	SF is calculated.
<b>Normal Flow</b>	<p>Assign weight in terms of very low, low, nominal, high, very high and extra high to factors given below.</p> <ul style="list-style-type: none"> <li>o Precedence (PREC)</li> <li>o Development/Flexibility (FLEX)</li> <li>o Architecture/Risk Resolution (RESL)</li> <li>o Team Cohesion (TEAM)</li> <li>o Process Maturity (PMAT)</li> </ul>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All weights should be inserted.
<b>Includes</b>	UC 31, UC 32, UC 33, UC 34, UC 35
<b>Special Requirements</b>	Accuracy: User will get the accurate information of the selected Mode/Models Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.

## Software Project Estimator

<b>Assumptions</b>	None
<b>Notes and Issues</b>	All weights should be correctly place to get desired result.

Table 2.2.27. Use Case for Language Factor (LF)

<b>Use Case Id</b>	UC-43
<b>Use Case Name</b>	Cal_Language_Factor
<b>Actor</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to calculate Language Factor
<b>Trigger</b>	When actor need to calculate UFP.
<b>Pre -conditions</b>	KLOC is required to calculate.
<b>Post-conditions</b>	None
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. Select any language.</li> <li>2. Value against selected language shown.</li> </ol>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	No language is selected.
<b>Includes</b>	None
<b>Special Requirements</b>	Accuracy: User will get the accurate information of the selected Mode/Models Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	None
<b>Notes and Issues</b>	All weighting factor should be multiple with correct weights.

Table 2.2.28. Use Case for Estimate Size Early Design Model COCOMO II

<b>Use Case ID:</b>	UC-44
<b>Use Case Name:</b>	Est_Size_EDM_COCOMOII
<b>Actors:</b>	User (Project Manager or Team lead)
<b>Description:</b>	The purpose of this use case is to estimate the size in KLOC.
<b>Trigger:</b>	User clicks on the space provided for size estimation.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Language Factor is calculated.</li> <li>2. Function point is calculated.</li> </ol>
<b>Post conditions:</b>	Estimated size is entered for further calculations.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Calculate Language Factor.</li> <li>2. Calculate Function Point.</li> <li>3. <math>KLOC = \text{Language Factor} * \text{Function Point}</math>.</li> </ol>
<b>Alternative Flows:</b>	N/A
<b>Exceptions:</b>	FP and Language Factor is not calculated.
<b>Includes:</b>	UC-41, UC-42

## Software Project Estimator

<b>Special Requirements :</b>	1. Accuracy: User will get accurate information of the selected mode. 2. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.
<b>Assumptions:</b>	User enters an accurate value.
<b>Notes and Issues:</b>	Further calculations will not be done in case of an inaccurate value entered.

Table 2.2.29. Use Case for Calculate Cost Drivers Early Design Model COCOMO II

<b>Use Case Id</b>	UC-45
<b>Use Case Name</b>	UC_Cal_Cost_Drivers
<b>Actor</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to calculate 15 cost drivers that come under 4 factors.
<b>Trigger</b>	When actor need to calculate EAF.
<b>Pre conditions</b>	- 15 cost drivers should be calculated before.
<b>Post-conditions</b>	EAF is calculated.
<b>Normal Flow</b>	1. Calculate Product Factor 2. Calculate Platform Factor 3. Calculate Personal Factor 4. Calculate Project Factor 5. All factors are weighted in terms of very low, low, nominal, high, very high and extra high.
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All factors should be calculated.
<b>Includes</b>	UC-10, UC-11, UC-12, UC-13
<b>Special Requirements</b>	Accuracy: User will get the accurate information of the selected Mode/Models Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	Actor is familiar with cost drivers.
<b>Notes and Issues</b>	Factors should be correctly measure to get correct result.

Table 2.2.30. Use Case for Estimate Effort Early Design Model COCOMO II

<b>Use Case Id:</b>	UC-46
<b>Use Case Name:</b>	UC_Est_Effort_EDM_COCOMOII
<b>Actors:</b>	User (Project Manager Team Lead)
<b>Description</b>	This model gets estimates of a project's cost and duration before its entire architecture is determined. It uses a small set of Cost Drivers. Based on Unadjusted Function Points
<b>Trigger</b>	Actor need to Early Design Model
<b>Pre conditions</b>	- Actor need to calculate COCOMO II.

## Software Project Estimator

<b>Post-conditions</b>	Early Design Model is calculated.
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. Calculate KLOC.</li> <li>2. Calculation of EAF using 15 Cost Drivers.</li> <li>3. Apply <math>E=2.45 \times \text{KLOC} \times \text{EAF}</math></li> </ol>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values are not inserted.
<b>Includes</b>	UC-44, UC-45
<b>Special Requirements</b>	Accuracy: User will get the accurate information of the selected Mode/Models Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	Actor is able to assign values in English Language.
<b>Notes and Issues</b>	Missing KLOC or EAF can lead to wrong result and errors.

Table 2.2.31. Use Case for Calculate Function Point COCOMO II

<b>Use Case Id</b>	UC-47
<b>Use Case Name</b>	UC_Cal_Function_Point
<b>Actor</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to calculate function point that will give the estimated size in KLOC.
<b>Trigger</b>	When actor needs to calculate function points.
<b>Pre conditions</b>	<ol style="list-style-type: none"> <li>1. Unadjusted function points (UFP) are calculated.</li> <li>2. Value adjustment factor (VAF) is calculated.</li> </ol>
<b>Post-conditions</b>	Function points are calculated and shown to the user.
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. Calculate UFP.</li> <li>2. Calculate VAF.</li> <li>3. Click on show results button to calculate function points.</li> </ol>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values are not entered. Invalid values are entered.
<b>Includes</b>	UC-8, UC-9
<b>Special Requirements</b>	Accuracy: User will get the accurate results as per values entered. Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	None
<b>Notes and Issues</b>	UFP and VAF should be calculated before.

Table 2.2.32. Use Case for Estimate Size Post Architecture Model COCOMO II

<b>Use Case ID:</b>	UC-48
<b>Use Case Name:</b>	Est_Size_PAM_COCOMOII
<b>Actors:</b>	User (Project Manager or Team lead)

## Software Project Estimator

<b>Description:</b>	The purpose of this use case is to estimate the size in KLOC.
<b>Trigger:</b>	User clicks on the space provided for size estimation.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Language Factor is calculated.</li> <li>2. Function point is calculated.</li> </ol>
<b>Post conditions:</b>	Estimated size is entered for further calculations.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Calculate Language Factor.</li> <li>2. Calculate Function Point.</li> <li>3. <math>KLOC = \text{Language Factor} * \text{Function Point}</math>.</li> </ol>
<b>Alternative Flows:</b>	N/A
<b>Exceptions:</b>	FP and Language Factor is not calculated.
<b>Includes:</b>	UC-8, UC-9, UC-43, UC-47
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>1. Accuracy: User will get accurate information of the selected mode.</li> <li>2. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.</li> </ol>
<b>Assumptions:</b>	User enters an accurate value.
<b>Notes and Issues:</b>	Further calculations will not be done in case of an inaccurate value entered.

Table 2.2.33. Use Case for Calculate Cost Drivers Post Architecture Model COCOMO II

<b>Use Case Id</b>	UC-49
<b>Use Case Name</b>	UC_Cal_Cost_Drivers
<b>Actor</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to calculate 15 cost drivers that come under 4 factors.
<b>Trigger</b>	When actor need to calculate EAF.
<b>Pre conditions</b>	- 15 cost drivers should be calculated before.
<b>Post-conditions</b>	EAF is calculated.
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. Calculate Product Factor</li> <li>2. Calculate Platform Factor</li> <li>3. Calculate Personal Factor</li> <li>4. Calculate Project Factor</li> <li>5. All factors are weighted in terms of very low, low, nominal, high, very high and extra high.</li> </ol>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All factors should be calculated.
<b>Includes</b>	UC-10, UC-11, UC-12, UC-13
<b>Special Requirements</b>	Accuracy: User will get the accurate information of the selected Mode/Models Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	Actor is familiar with cost drivers.

<b>Notes and Issues</b>	Factors should be correctly measure to get correct result.
-------------------------	--

Table 2.2.34. Use Case for Calculate Scale Factor for Post Architecture Model COCOMOII

<b>Use Case Id</b>	UC-50
<b>Use Case Name</b>	UC_Cal_Scale_Factor
<b>Actor</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to calculate Scale factor.
<b>Trigger</b>	When actor need to calculate SF.
<b>Pre conditions</b>	- Require to evaluate project in Post Architectural Model.
<b>Post-conditions</b>	SF is calculated.
<b>Normal Flow</b>	<p>Assign weight in terms of very low, low, nominal, high, very high and extra high to factors given below.</p> <ul style="list-style-type: none"> <li>○ Precedence (PREC)</li> <li>○ Development/Flexibility (FLEX)</li> <li>○ Architecture/Risk Resolution (RESL)</li> <li>○ Team Cohesion (TEAM)</li> <li>○ Process Maturity (PMAT)</li> </ul>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All weights should be inserted.
<b>Includes</b>	UC-31, UC-32, UC-33, UC-34, UC-35
<b>Special Requirements</b>	Accuracy: User will get the accurate information of the selected Mode/Models Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	None
<b>Notes and Issues</b>	All weights should be correctly place to get desired result.

Table 2.2.35. Use Case for Estimate Effort for Post Architecture Model COCOMO II

<b>Use Case Id:</b>	UC-51
<b>Use Case Name:</b>	UC_Est_Effort_PAM_COCOMOII
<b>Actors:</b>	User (Project Manager Team Lead)
<b>Description</b>	This is the most detailed COCOMO II model. It is used after developing project's overall architecture. It has new cost drivers and new equations.

<b>Trigger</b>	When actor is calculating COCOMO II.
<b>Pre conditions</b>	COCOMO II need to be calculated.
<b>Post-conditions</b>	Post Architecture Model is calculated.
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. Calculate EAF.</li> <li>2. Calculate KLOC.</li> <li>3. Calculate Scale Factor.</li> <li>4. Apply Formula: <math>E = 2.45 \times (KLOC)^b * EAF</math></li> </ol>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values are not inserted.
<b>Includes</b>	UC-48, UC-49, UC-50
<b>Special Requirements</b>	Accuracy: User will get the accurate information of the selected Mode/Models Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.
<b>Assumptions</b>	
<b>Notes and Issues</b>	EAF, b and KLOC should be calculated before.

Table 2.2.36. Use Case for Estimate Product Productivity for SLIM

<b>Use Case ID:</b>	UC-58
<b>Use Case Name:</b>	UC_Cal_Process_Productivity_SLIM
<b>Actors:</b>	User (project manager, team leader)
<b>Description:</b>	There are many applications type like business system, microcode, firmware (ROM) etc and product index values lies from 1 to 40 and standard values of product productivity which users will use.
<b>Trigger:</b>	User select the application type to use the standard values of productivity parameters.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. User should know the application type.</li> <li>2. User should know the environment of the project to select the productivity index (PI) high or low values.</li> </ol>
<b>Post conditions:</b>	Product productivity will have calculated.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1- User select the application type.</li> <li>2- User select the environment of productivity index from the given range.</li> <li>3- Productivity parameter standard value will be given to the user.</li> </ol>
<b>Alternative Flows:</b>	None.
<b>Exceptions:</b>	User enter wrong productivity index value.
<b>Includes:</b>	None
<b>Special Requirements :</b>	<ol style="list-style-type: none"> <li>1. Accuracy: User will get accurate information of the selected mode.</li> <li>2. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.</li> </ol>
<b>Assumptions:</b>	User knows the environment to do the project.
<b>Notes and Issues:</b>	None.

Table 2.2.37. Use Case for Estimate Size for SLIM

<b>Use Case Id</b>	UC-54
<b>Use Case Name</b>	UC_Est_Size_SLIM
<b>Actor</b>	User (Project Manager Team Lead)
<b>Description</b>	The purpose of this use case is to calculate function point that will give the estimated size.
<b>Trigger</b>	When actor needs to calculate function points.
<b>Pre conditions</b>	<ul style="list-style-type: none"> <li>1. Unadjusted function points (UFP) are calculated.</li> <li>2. Value adjustment factor (VAF) is calculated.</li> </ul>
<b>Post-conditions</b>	Function points are calculated.
<b>Normal Flow</b>	<ul style="list-style-type: none"> <li>1. Calculate UFP.</li> <li>2. Calculate VAF.</li> <li>3. FP=UFP*VAF.</li> </ul>
<b>Alternative Flow</b>	None
<b>Exceptions</b>	All values are not inserted.
<b>Includes</b>	UC-7, UC-8
<b>Special Requirements</b>	<p>Accuracy: User will get the accurate results as per values entered.</p> <p>Performance: User will get the result in less than a minute (preferably 10 sec) depending on the bandwidth of the internet connected.</p>
<b>Assumptions</b>	None
<b>Notes and Issues</b>	UFP and VAF should be calculated before.

Table 2.2.38. Use Case for Estimate Total Effort for SLIM

<b>Use Case ID:</b>	UC-56
<b>Use Case Name:</b>	UC_Est_Total_Effort_SLIM
<b>Actors:</b>	User (project manager, team lead)
<b>Description:</b>	B (total effort staff-years) will be calculated to find out the total effort required to complete the specific project at the given time with the limited number of staff.
<b>Trigger:</b>	User click on calculate button.
<b>Preconditions :</b>	<ul style="list-style-type: none"> <li>1. User should estimate the size of the project.</li> <li>2. User should estimate time which is given by client to complete the project.</li> <li>3. User should know the value of process productivity.</li> </ul>
<b>Post conditions:</b>	B will be calculated.
<b>Normal Flow:</b>	<ul style="list-style-type: none"> <li>1. User calculate the estimated size of the project.</li> <li>2. User enters the estimated time to complete the project.</li> <li>3. User calculate the process productivity value.</li> <li>4. B is calculated.</li> </ul>
<b>Alternative Flows:</b>	None.

## Software Project Estimator

<b>Exceptions:</b>	1. User enter wrong estimated time.
<b>Includes:</b>	UC-53, UC-54, UC-55
<b>Special Requirements :</b>	1. Accuracy: User will get accurate information of the selected mode. 2. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.
<b>Assumptions:</b>	User knows the estimated size and time.
<b>Notes and Issues:</b>	Have an estimate of the required staff and estimated time.

Table 2.2.39. Use Case for Estimate Delivery Effort SLIM

<b>Use Case ID:</b>	UC-57
<b>Use Case Name:</b>	UC_Est_Delivery_Effort_SLIM
<b>Actors:</b>	User (project manager, team lead)
<b>Description:</b>	The purpose of this use case is to calculate the effort required at the delivery time to complete the project.
<b>Trigger:</b>	User click on calculate to calculate E.
<b>Preconditions :</b>	1. User knows the total effort required to complete the project.
<b>Post conditions:</b>	E will be calculated.
<b>Normal Flow:</b>	1. User enter the total effort of the project. 2. E is calculated
<b>Alternative Flows:</b>	None.
<b>Exceptions:</b>	1. User enter wrong values. 2. User does not know the values. 3. User does not enter all values.
<b>Includes:</b>	UC-56
<b>Special Requirements:</b>	1. Accuracy: User will get accurate information of the selected mode. 2. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.
<b>Assumptions:</b>	User knows the time (deadline) and size of the project
<b>Notes and Issues:</b>	Have an estimated time and size of the project.

Table 2.2.40. Use Case for Calculate Manpower Equation for SLIM

<b>Use Case ID:</b>	UC-59
<b>Use Case Name:</b>	UC_Cal_Manpower_Equation_SLIM
<b>Actors:</b>	User (project manager, team lead)
<b>Description:</b>	The purpose of this use case is to find out the manpower acceleration using effort and time.
<b>Trigger:</b>	User click on calculate to calculate D.
<b>Preconditions :</b>	1. User know the total effort required to complete the project. 2. User know the estimated time to complete the project.

<b>Post conditions:</b>	D will be calculated.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User calculate the value of B (total effort).</li> <li>2. User enters the estimated time of the project.</li> <li>3. D is calculated.</li> </ol>
<b>Alternative Flows:</b>	None.
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. User enter wrong values.</li> <li>2. User does not know the values.</li> <li>3. User does not enter all values.</li> </ol>
<b>Includes:</b>	UC-56, UC-57, UC-58
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>1. Accuracy: User will get accurate information of the selected mode.</li> <li>2. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.</li> </ol>
<b>Assumptions:</b>	User knows the time (deadline) to complete the project.
<b>Notes and Issues:</b>	Have an estimated time and total effort.

Table 2.2.41. Use Case for Estimate Cost

<b>Use Case ID:</b>	UC-60
<b>Use Case Name:</b>	UC_Cal_Cost
<b>Actors:</b>	User (project manager, team lead)
<b>Description:</b>	The purpose of this use case is to calculate the cost required to complete the project.
<b>Trigger:</b>	User click on calculate to calculate Cost.
<b>Preconditions :</b>	<ol style="list-style-type: none"> <li>1. User knows the total effort and time required to complete the project.</li> </ol>
<b>Post conditions:</b>	Cost will be calculated.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User enter the total effort of the project.</li> <li>2. User enter the total time required.</li> <li>3. Cost is calculated</li> </ol>
<b>Alternative Flows:</b>	None.
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. User enter wrong values.</li> <li>2. User does not know the values.</li> <li>3. User does not enter all values.</li> </ol>
<b>Includes:</b>	UC-1 UC-2
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>3. Accuracy: User will get accurate information of the selected mode.</li> <li>4. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.</li> </ol>
<b>Assumptions:</b>	User knows the time (deadline) and size of the project
<b>Notes and Issues:</b>	Have an estimated time and size of the project.

Table 2.2.42. Use Case for Registration

<b>Use Case ID:</b>	UC-61
<b>Use Case Name:</b>	UC_Registration
<b>Actors:</b>	User (project manager, team lead)
<b>Description:</b>	The purpose of this use case is to register the user, so he/she can manage his/her reports.
<b>Trigger:</b>	User click on Sign Up.
<b>Preconditions :</b>	<ol style="list-style-type: none"> <li>1. User visits signup page.</li> </ol>
<b>Post conditions:</b>	User account will be created.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User fill registration form.</li> <li>2. User click signup.</li> <li>3. System validate information.</li> <li>4. Account is created.</li> </ol>
<b>Alternative Flows:</b>	None.
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. User enter invalid information.</li> </ol>
<b>Includes:</b>	None
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>1. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.</li> </ol>
<b>Assumptions:</b>	User have required information.
<b>Notes and Issues:</b>	None.

Table 2.2.43. Use Case for Login

<b>Use Case ID:</b>	UC-62
<b>Use Case Name:</b>	UC_Login
<b>Actors:</b>	User (project manager, team lead)
<b>Description:</b>	The purpose of this use case is to access the profile, so he/she can manage his/her reports.
<b>Trigger:</b>	User click on Login.
<b>Preconditions :</b>	<ol style="list-style-type: none"> <li>1. User visits Login page.</li> </ol>
<b>Post conditions:</b>	User loges into the system.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User enter name and password.</li> <li>2. User click on login.</li> <li>3. System validate information.</li> <li>5. Account is created.</li> </ol>

## Software Project Estimator

<b>Alternative Flows:</b>	None.
<b>Exceptions:</b>	1. User enter invalid information.
<b>Includes:</b>	None
<b>Special Requirements:</b>	1. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.
<b>Assumptions:</b>	User have required information.
<b>Notes and Issues:</b>	None.

Table 2.2.44. Use Case for Generate report

<b>Use Case ID:</b>	UC-63
<b>Use Case Name:</b>	UC_Genrate_Report
<b>Actors:</b>	User (project manager, team lead)
<b>Description:</b>	The purpose of this use case is present all the estimated data in the form of report.
<b>Trigger:</b>	User click on generate report.
<b>Preconditions :</b>	1. User complete all estimation steps.
<b>Post conditions:</b>	Report will be generated.
<b>Normal Flow:</b>	1. User click on generate 2. Report is generated.
<b>Alternative Flows:</b>	None.
<b>Exceptions:</b>	None
<b>Includes:</b>	None
<b>Special Requirements:</b>	1. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.
<b>Assumptions:</b>	None.
<b>Notes and Issues:</b>	None.

Table 2.2.45. Use Case for Print report

<b>Use Case ID:</b>	UC-64
<b>Use Case Name:</b>	UC_Print_Report
<b>Actors:</b>	User (project manager, team lead)
<b>Description:</b>	The purpose of this use case is present all the estimated data in printed form.
<b>Trigger:</b>	User click on print report.
<b>Preconditions :</b>	1. Report is generated

## Software Project Estimator

<b>Post conditions:</b>	Report will be Printed.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User click on Print</li> <li>2. Report will be printed.</li> </ol>
<b>Alternative Flows:</b>	None.
<b>Exceptions:</b>	None
<b>Includes:</b>	None
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>3. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.</li> </ol>
<b>Assumptions:</b>	None.
<b>Notes and Issues:</b>	None.

Table 2.2.46. Use Case for Save report

<b>Use Case ID:</b>	UC-65
<b>Use Case Name:</b>	UC_Save_Report
<b>Actors:</b>	User (project manager, team lead)
<b>Description:</b>	The purpose of this use case is save the report for future use.
<b>Trigger:</b>	User click on Save report.
<b>Preconditions :</b>	<ol style="list-style-type: none"> <li>1. Report is generated</li> </ol>
<b>Post conditions:</b>	Report will be saved.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User click on save</li> <li>2. Report is saved.</li> </ol>
<b>Alternative Flows:</b>	None.
<b>Exceptions:</b>	None
<b>Includes:</b>	None
<b>Special Requirements:</b>	<ol style="list-style-type: none"> <li>1. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.</li> </ol>
<b>Assumptions:</b>	User is logged in.
<b>Notes and Issues:</b>	None.

Table 2.2.47. Use Case for View report

<b>Use Case ID:</b>	UC-66
<b>Use Case Name:</b>	UC_View_Report
<b>Actors:</b>	User (project manager, team lead)
<b>Description:</b>	The purpose of this use case is to view saved reports.
<b>Trigger:</b>	User click on view report.

## Software Project Estimator

<b>Preconditions :</b>	1. User is logged in.
<b>Post conditions:</b>	Report will be displayed.
<b>Normal Flow:</b>	1. User click on view report. 2. Report is displayed.
<b>Alternative Flows:</b>	None.
<b>Exceptions:</b>	None
<b>Includes:</b>	None
<b>Special Requirements:</b>	1. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.
<b>Assumptions:</b>	None
<b>Notes and Issues:</b>	None.

Table 2.2.48. Use Case for Delete report

<b>Use Case ID:</b>	UC-67
<b>Use Case Name:</b>	UC_Delete_Report
<b>Actors:</b>	User (project manager, team lead)
<b>Description:</b>	The purpose of this use case is present all the estimated data in the form of report.
<b>Trigger:</b>	User click on delete report.
<b>Preconditions :</b>	1. User is logged in.
<b>Post conditions:</b>	Report will be deleted.
<b>Normal Flow:</b>	1. User click on delete report. 2. Report is deleted.
<b>Alternative Flows:</b>	None.
<b>Exceptions:</b>	None
<b>Includes:</b>	None
<b>Special Requirements:</b>	1. Performance: User will get the results in less than a minute (preferably 10 seconds), depending on the bandwidth of the internet connection.
<b>Assumptions:</b>	None.
<b>Notes and Issues:</b>	None.

## **Chapter 3: Design Document**

---

### 3.1. Introduction

A software design document is a written description of a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. This document gives a detailed description of the software architecture of the Software Project Estimator. It specifies the structure of the modules discussed in the SRS.

### 3.2. Domain Model

Domain Model forms the foundation for the design of the software. The domain model is created during object-oriented exploration to crumble the domain into concepts or objects in the real world. In software engineering, a domain model is a conceptual model of the domain that integrates both behavior and data. This model can help to solve domain related concerns. It is visual representation of conceptual classes or objects in a domain. In UML, the Domain Model is illustrated with a set of class diagrams without methods. It epitomizes real-world concepts, not software components. Domain model of Software Project Tool is follows.

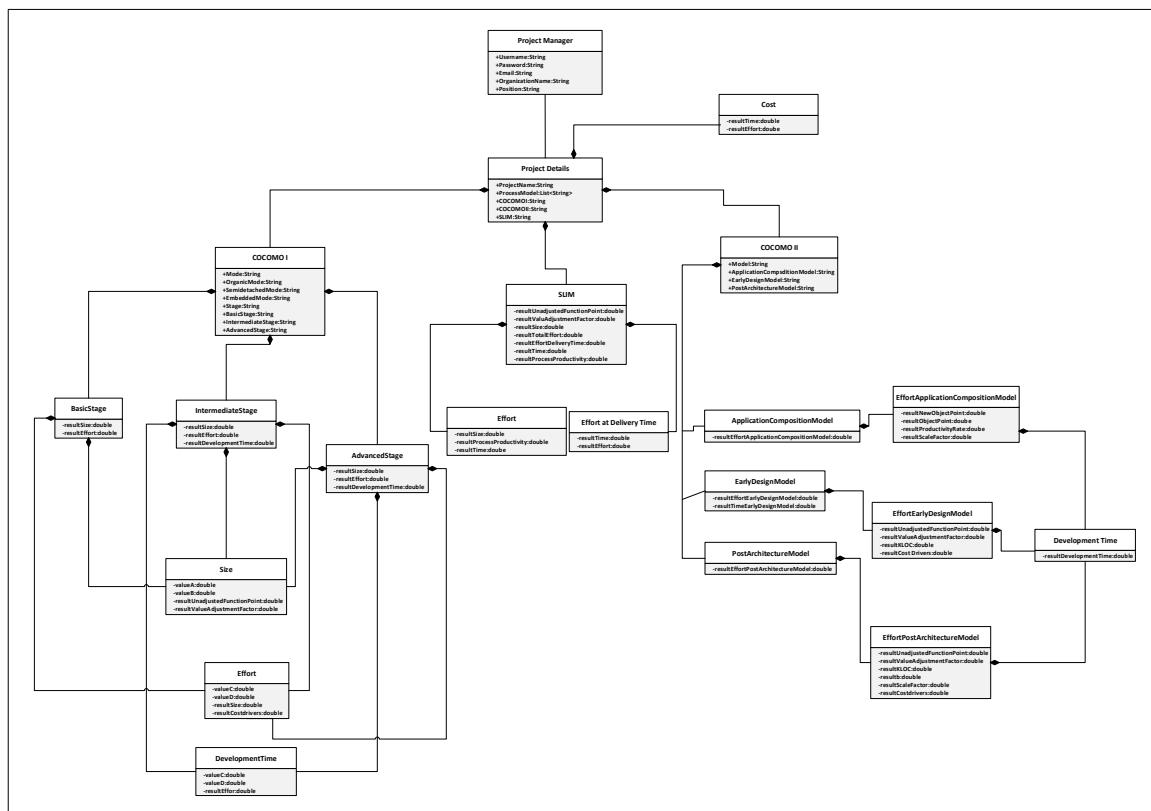
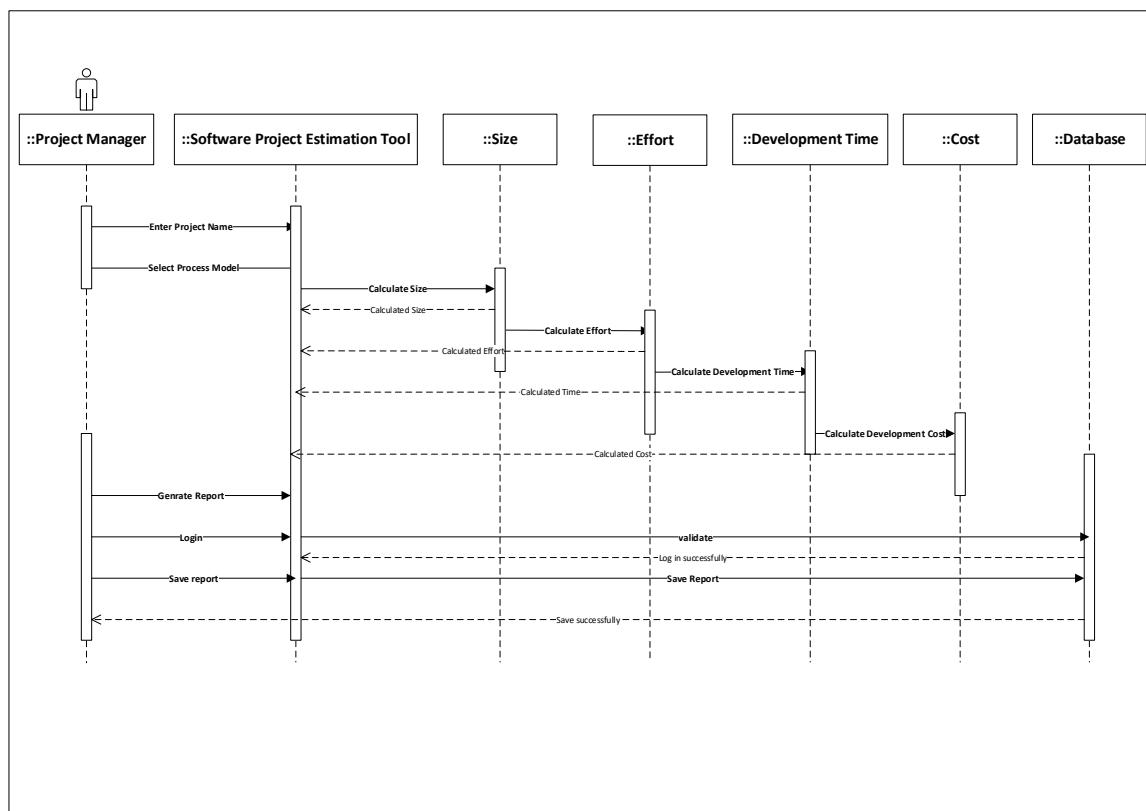


Figure 3. 1: Domain Model of the system

### 3.3. System Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It tells about interactions among classes in terms of a give-and-take of messages over time. Also known as event diagrams. The sequence diagram is used largely to show the connections between objects in the sequential order. The horizontal axis shows the elements that are involved in the interaction. The vertical axis represents time proceedings.

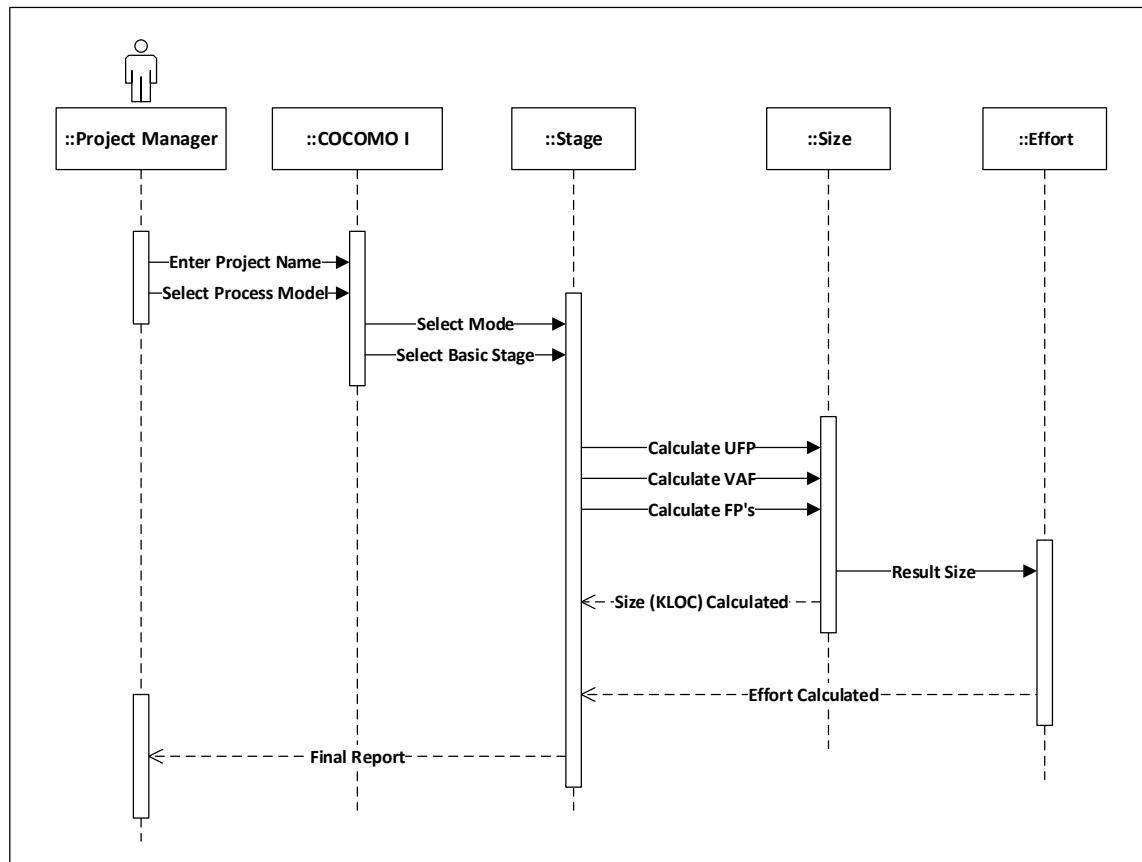


### 3.4. Sequence Diagram

The sequence diagram for “Software Project Estimator” is as follows.

#### 3.4.1. Sequence Diagram for Basic Stage COCOMO I

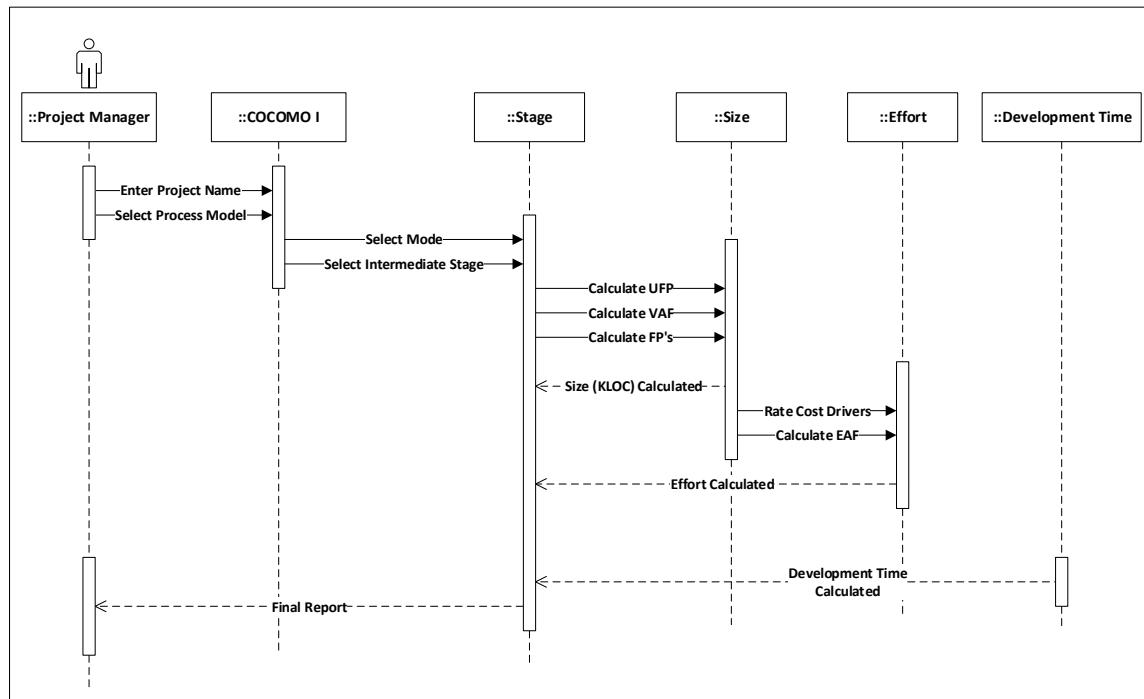
Figure 3.2.1. shows the sequence diagram for Basic Stage COCOMO I of the purposed system. This diagram shows the interaction between Project Manager and application. It demonstrating that how project manager calculates size and effort at basic stage of COCOMO I, into the system.



**Figure 3.2. 1: Basic Stage COCOMO I**

### 3.4.2. Sequence Diagram for Intermediate Stage COCOMO I

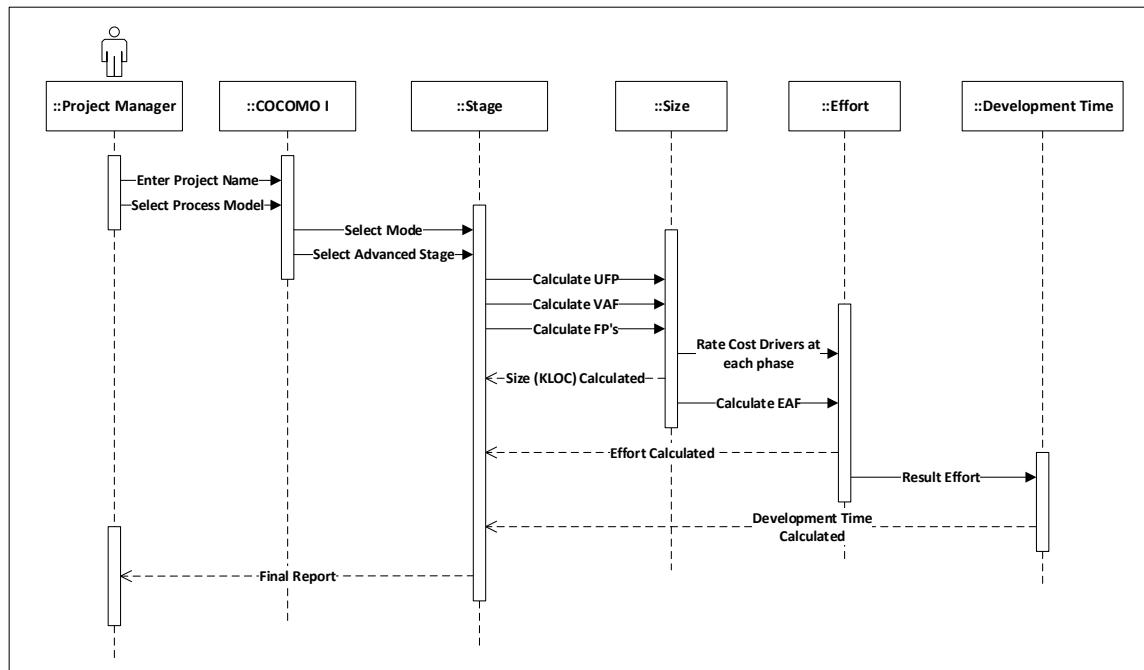
Figure 3.2.2. shows the sequence diagram for Intermediate Stage COCOMO I of the purposed system. This diagram shows the interaction between Project Manager and application. It demonstrating that how project manager calculates size and effort at intermediate stage of COCOMO I, into the system.



**Figure 3.2. 2: Intermediate Stage COCOMO I**

### 3.4.3. Sequence Diagram for Advanced Stage COCOMO I

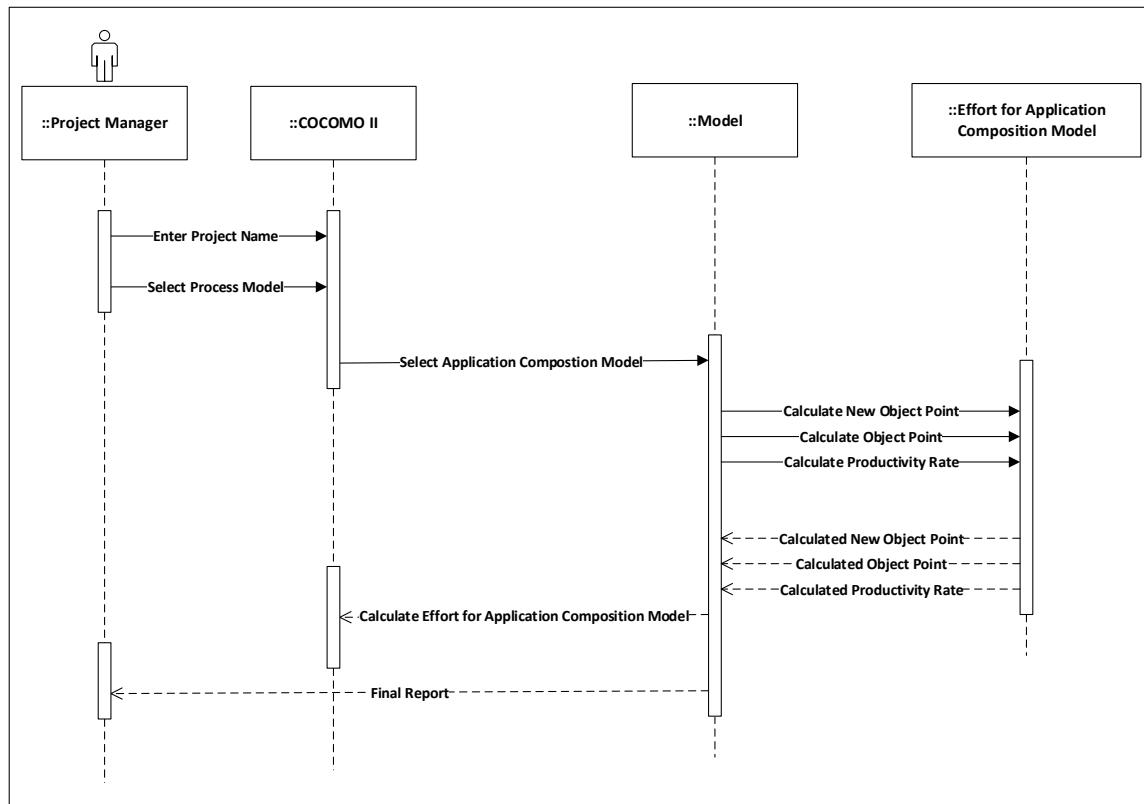
Figure 3.2.3. shows the sequence diagram for Advanced Stage COCOMO I of the purposed system. This diagram shows the interaction between Project Manager and application. It demonstrating that how project manager calculates size and effort at advanced stage of COCOMO I, into the system.



**Figure 3.2. 3: Advanced Stage COCOMO I**

### 3.4.4. Sequence Diagram for Application Composition Model COCOMO II

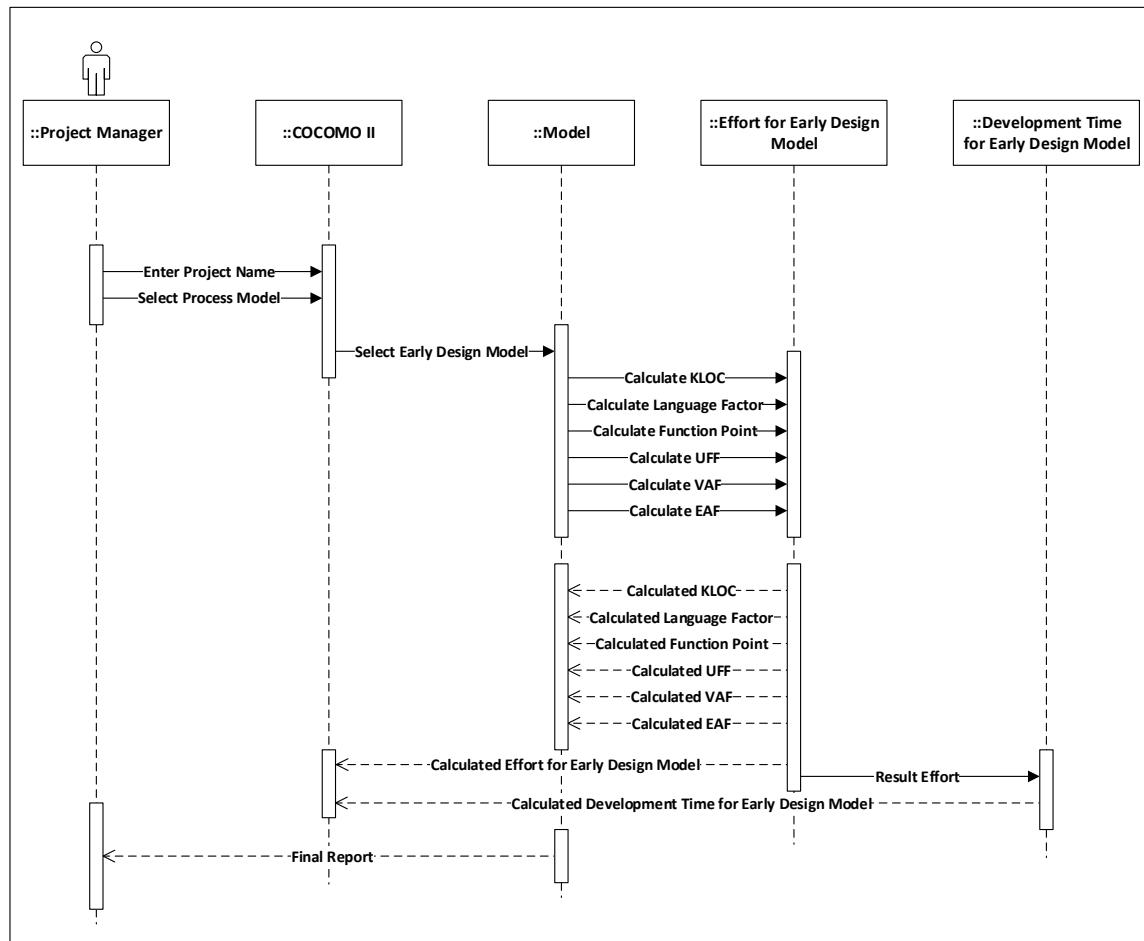
Figure 3.2.4. shows the sequence diagram for Application Composition Model COCOMO II of the purposed system. This diagram shows the interaction between Project Manager and application. It demonstrating that how project manager calculates size and effort at application composition model of COCOMO II, into the system.



**Figure 3.2. 4: Application Composition Model COCOMO II**

### 3.4.5. Sequence Diagram for Early Design Model COCOMO II

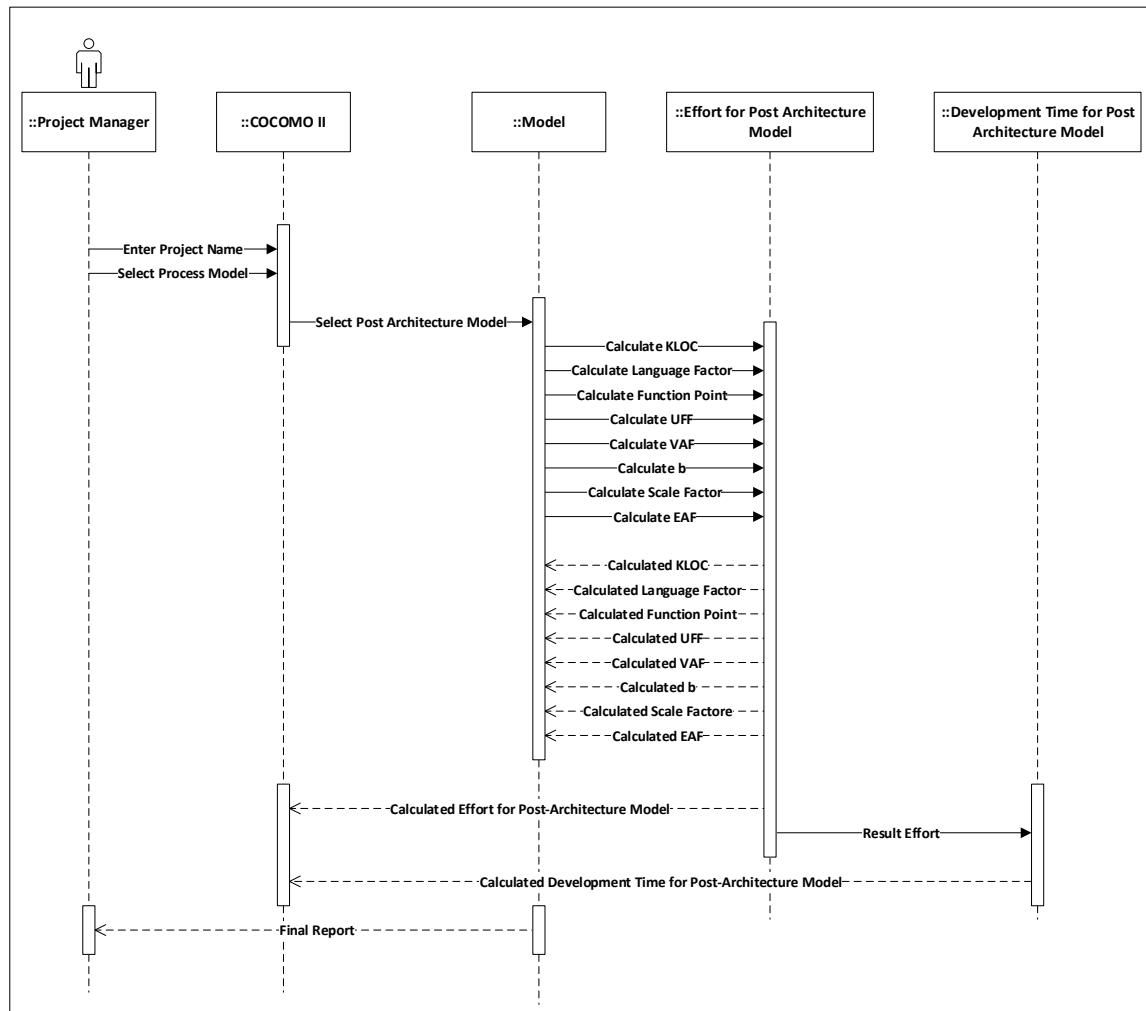
Figure 3.2.5. shows the sequence diagram for Early Design Model COCOMO II of the purposed system. This diagram shows the interaction between Project Manager and application. It demonstrating that how project manager calculates size and effort at early design model of COCOMO II, into the system.



**Figure 3.2. 5: Early Design Model COCOMO II**

### 3.4.6. Sequence Diagram for Post Architecture Model COCOMO II

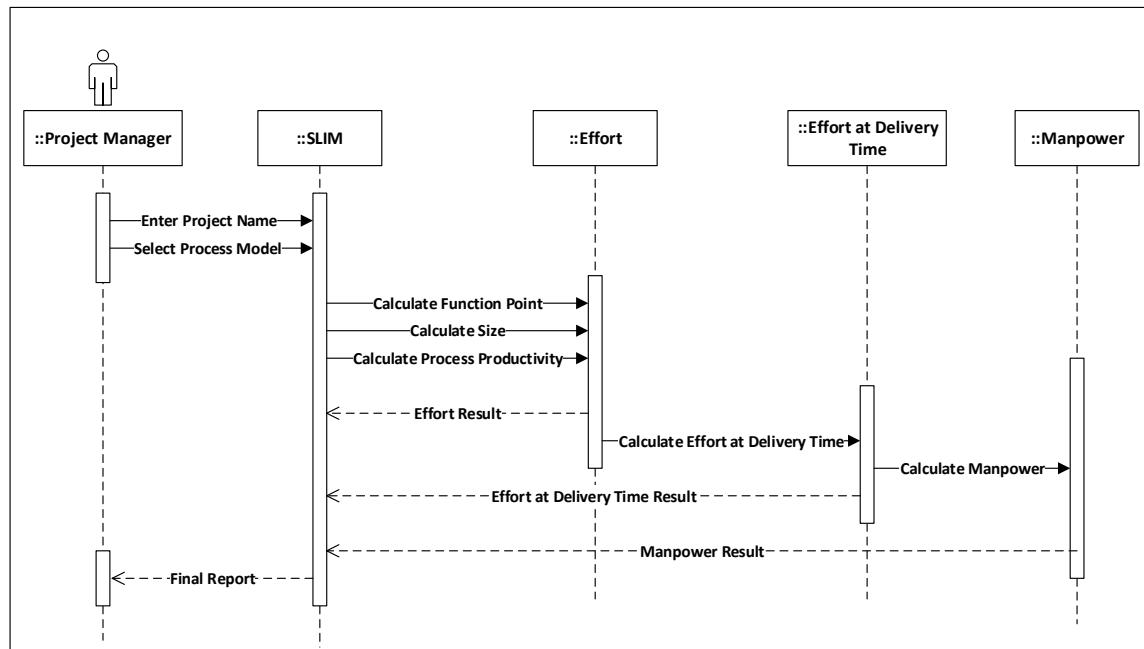
Figure 3.2.6. shows the sequence diagram for Post Architecture Model COCOMO II of the purposed system. This diagram shows the interaction between Project Manager and application. It demonstrating that how project manager calculates size and effort at post architecture model of COCOMO II, into the system.



**Figure 3.2. 6: Post Architecture Model COCOMO II**

### 3.4.7. Sequence Diagram for Software Lifecycle Management Model SLIM

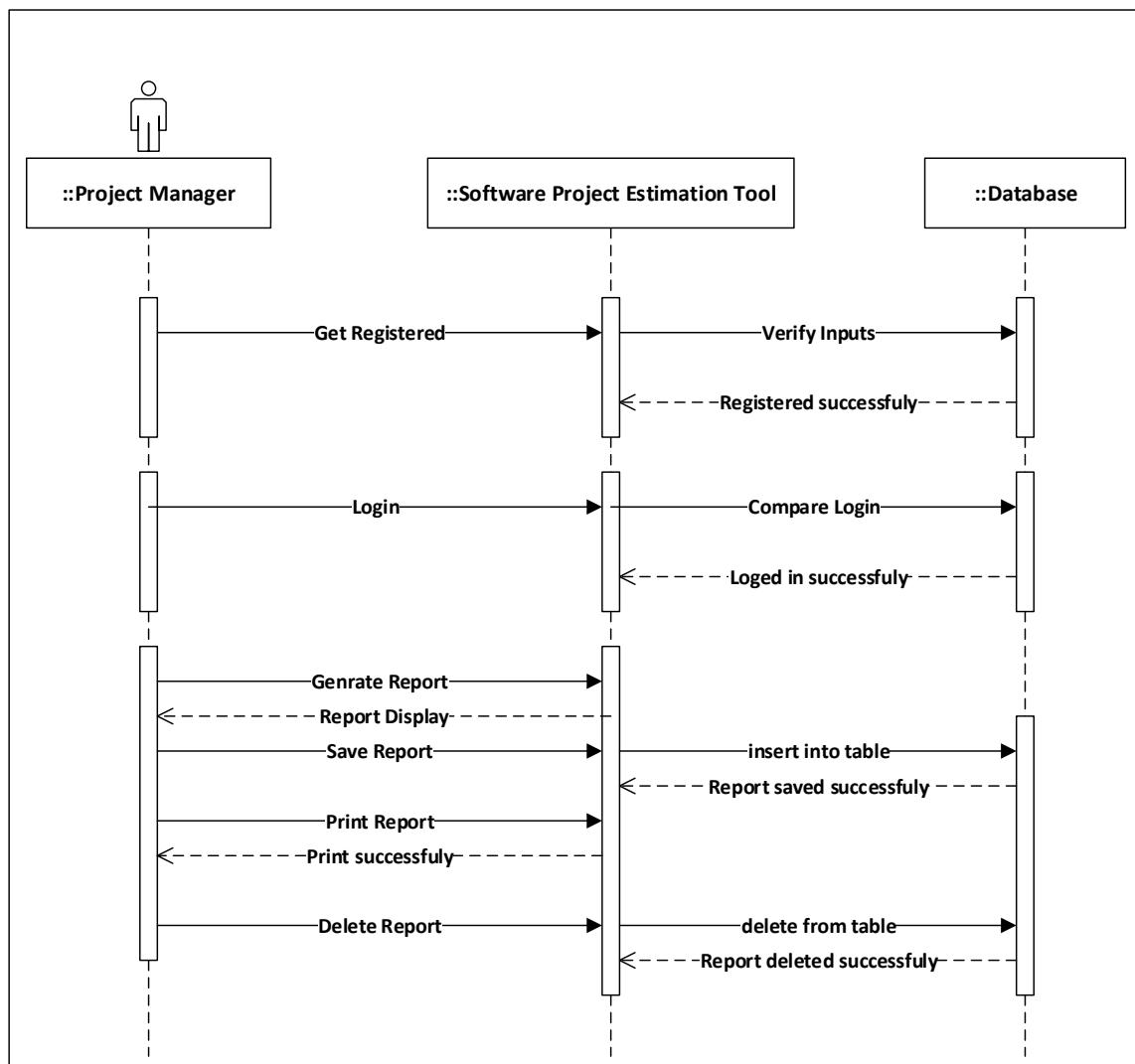
Figure 3.2.7. shows the sequence diagram for SLIM Model of the purposed system. This diagram shows the interaction between Project Manager and application. It demonstrating that how project manager calculates size, total effort, effort at delivery time, and manpower equation at slim model, into the system.



**Figure 3.2. 7: Software Lifecycle Management Model SLIM**

### 3.4.8. Sequence Diagram for Estimated Projects Management

Figure 3.2.8. shows the sequence diagram for Estimated Projects Management module of the purposed system. This diagram shows the interaction between Project Manager and application. It demonstrating that how project manager login, register, and manage their estimated projects, into the system.



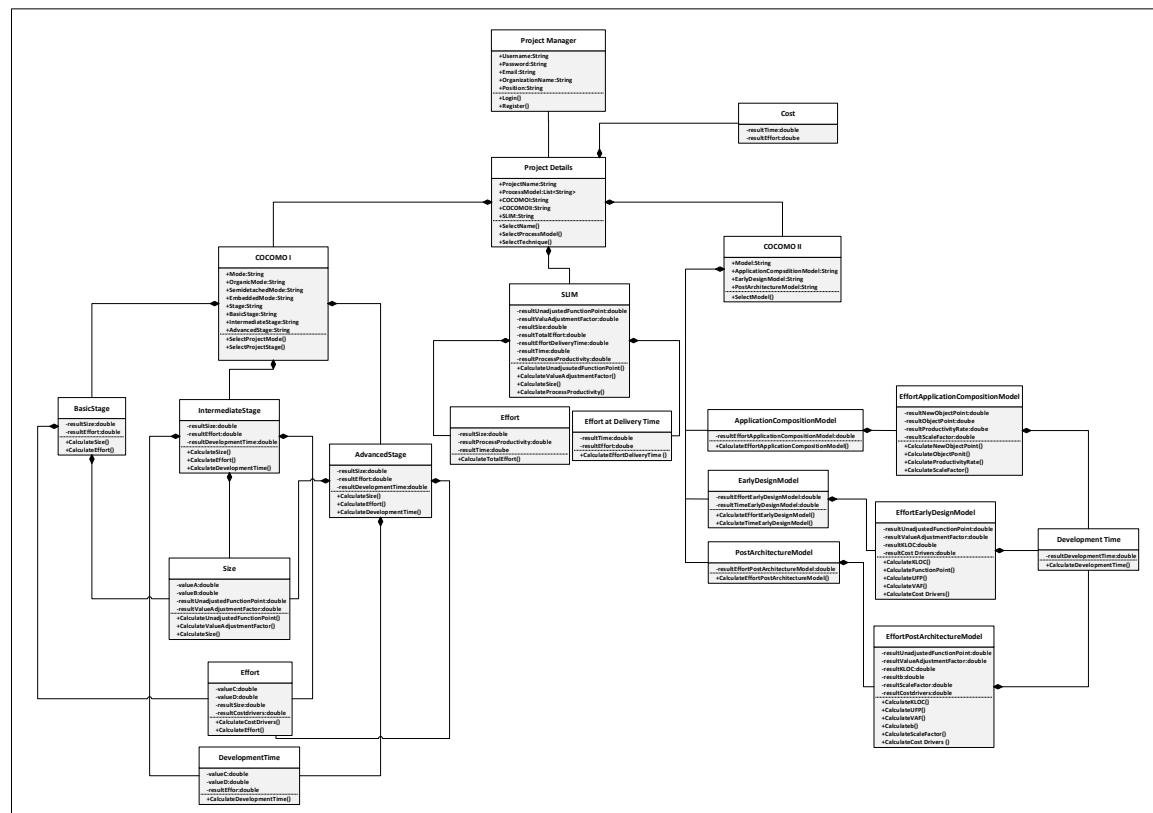
**Figure 3.2. 8: Estimated Projects Management**

### **3.5. Design Class Diagram**

A class diagram in the (UML) is a type of stagnant structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among those objects. The Class Diagram for “Software Project Estimator” shown in figure --.

In class diagram, classes are symbolized with boxes that contain three partitions:

- The top partition encloses the name of the class. It is printed in bold and centered.
  - The middle partition encompasses the attributes of the class.
  - The bottom section holds the operations the class can execute.



**Figure 3. 2: Class Diagram of the system**

### **3.6. Data Model**

The Data Modeling Diagram is used to create or view graphical models of relational database system schemas including a range of database objects. The diagrams can be drawn at a logical or a physical level. Tables, Views, Stored Procedures and other objects are connected showing the way they are related to each other. A data model identifies the structure of the data itself in detail. The main purpose of having the data models is to support the information systems development in a way of providing the definition as well as the format of some data. There are many kinds of data models most important one Relational Data Model.

### ○ Identify Entities

The entities involved in the system are as follows.

1. User
  2. Project

- **Find Relationships**

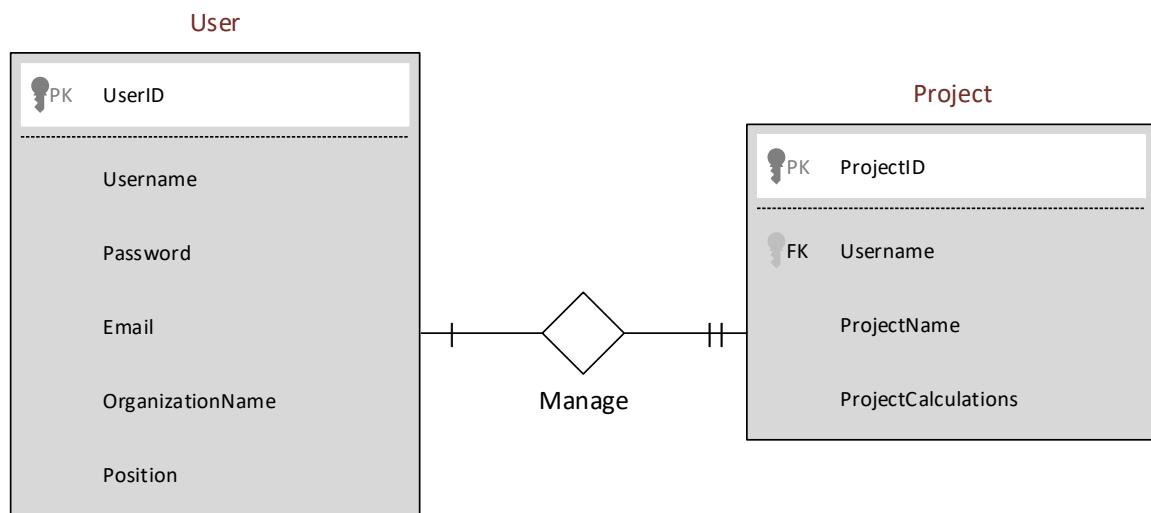
After identifying entities, the relationships between the entities would be drawn.

- **Draw ERD**

Entity Relationship Diagram sketch will be drawn to connect entities.

- **Fill in Cardinality**

In final step cardinality is filled



**Figure 3. 3: Entity Relationship Diagram of the system**

## Chapter 4: User Interface Design

---

## 4.1. Introduction

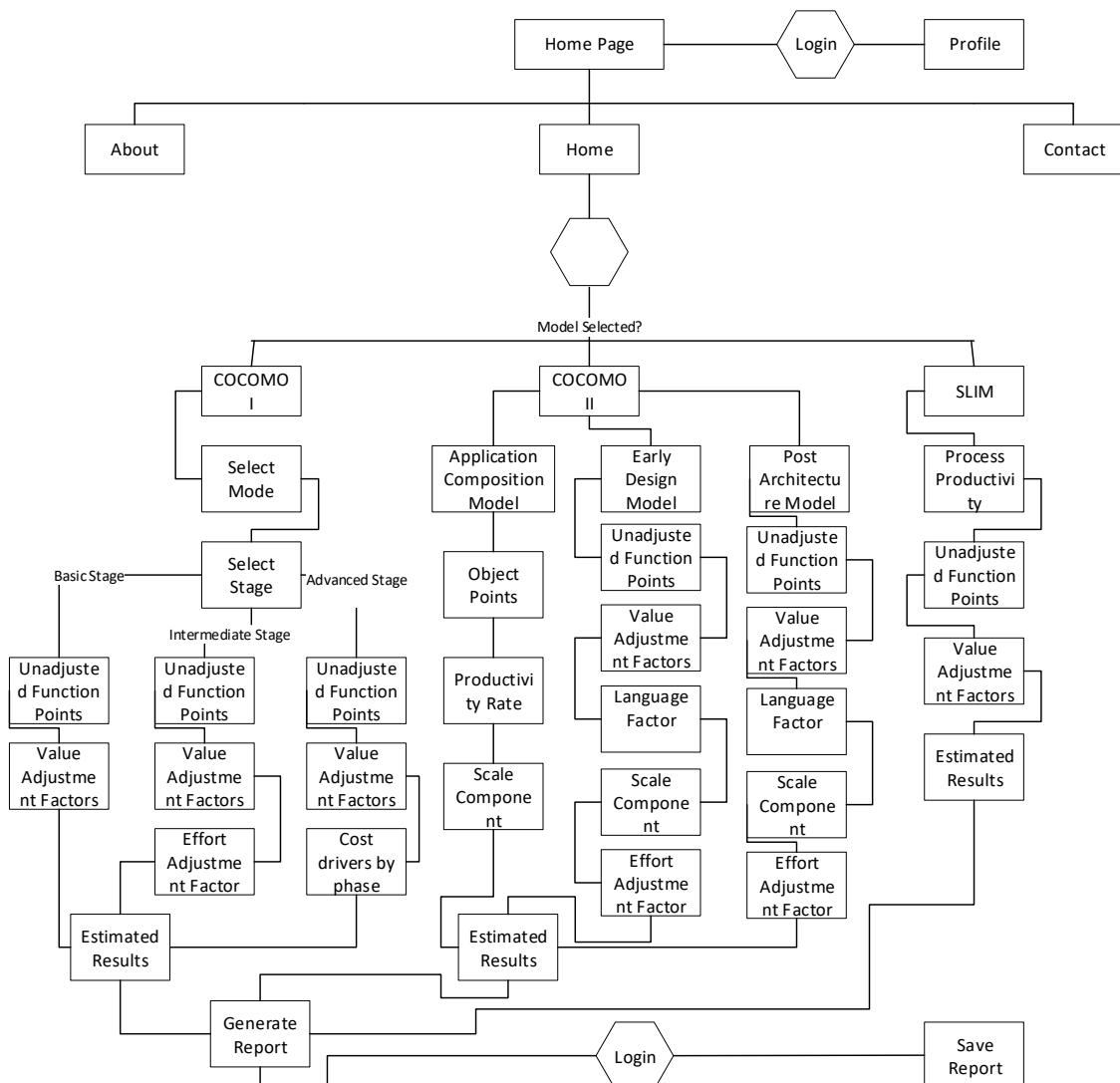
A user interface design consists of three main parts:

Page elements should be visualized on paper before building them in the computer. Just as you draw a site map to plan the site, use cartoons and storyboards to begin blocking out the site's appearance and navigational scheme.

1. Site maps
2. Storyboards
3. Navigational maps
4. Traceability Matrix

## 4.2. Site Maps

A site map's main benefit is to give users an overview of the site's areas in a single glance by dedicating an entire page to a visualization of the information architecture. If designed well, this overview can include several levels of hierarchy, and yet not be so big that users lose their ability to grasp the map as a whole.



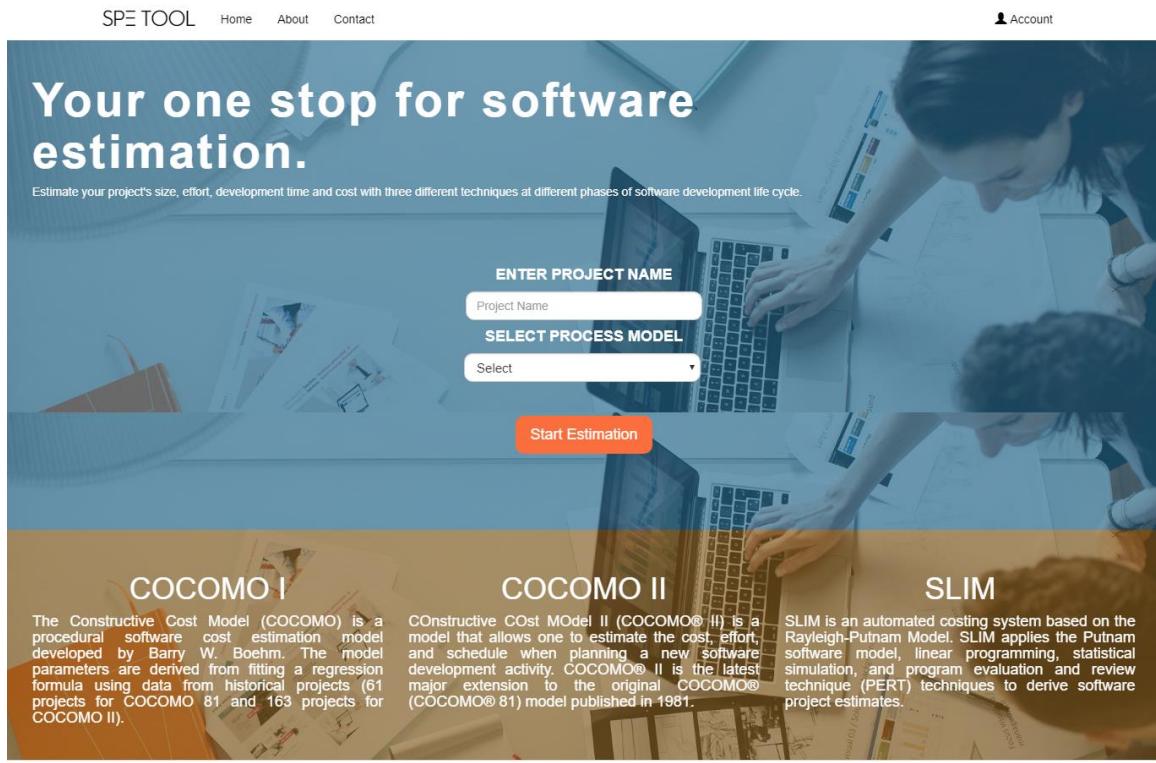
**Figure 4. 1: Site Map of the system**

### 4.3. Story boards

A storyboard is a sequence of single images, each of which represents a distinct event or narrative. It is also a visual representation of the script illustrating the interaction between the user and the machine. It can also be imagined as a film in visual-outline form. There are different attributes involved in story boards to represent our system describe below.

- **Environment:**

The web application “Software estimation tool” is used in Software houses for the better and efficient estimation of their projects. (UI ID #1)



- **User input:**

The system is designed in such a manner that it can be used anywhere like on PC, laptops, your smart phones etc. User give no. of screens, reports, language used, complexity level, productivity rate etc.

- **Machine output:**

The system responds to the user estimated size, development time and cost.

- **Technology:**

System Front end is being designed in Bootstrap and on the backend, we have worked on MVC .NET with C#, java script and SQL server.

- **Quality of experience:**

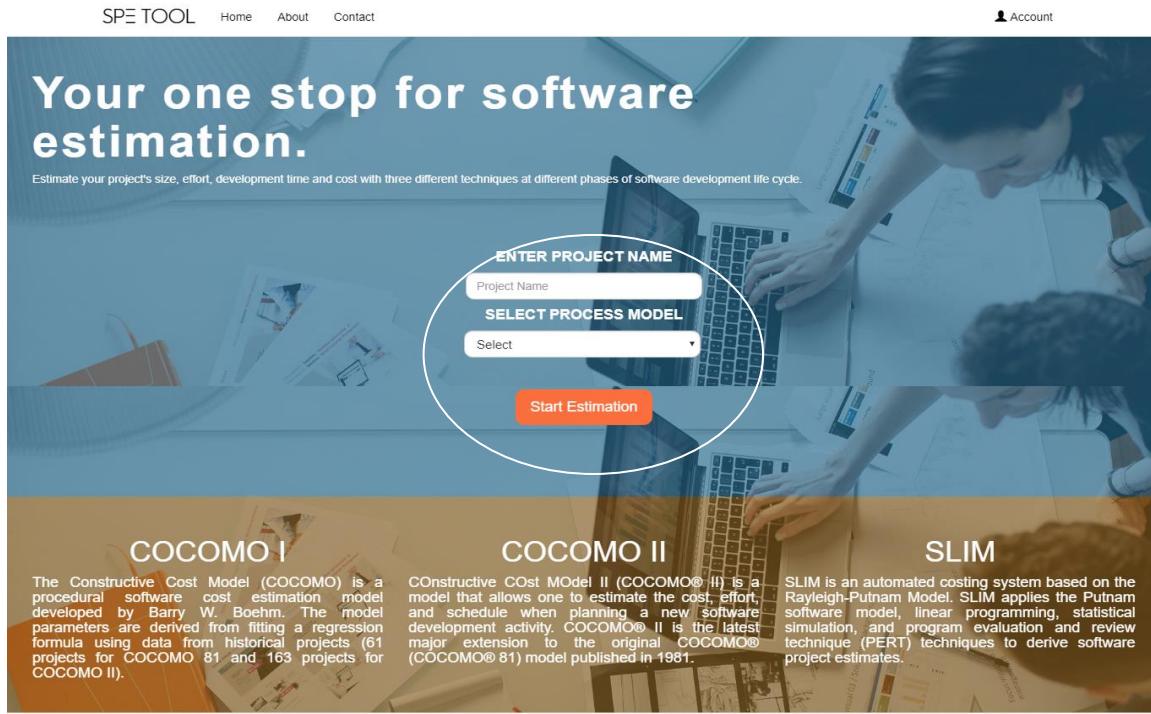
In the process of developing this system we will be able to learn new and different technologies. Our system will prove the best for the better management of software houses.

#### **4.4. Navigational maps:**

The next step is of navigational maps. In these maps, the storyboards are used as an input. The different display buttons or action buttons show the navigation from one screen to the other.

- **Home**

User will enter project name and select a process model from the dropdown. (UI ID #2)

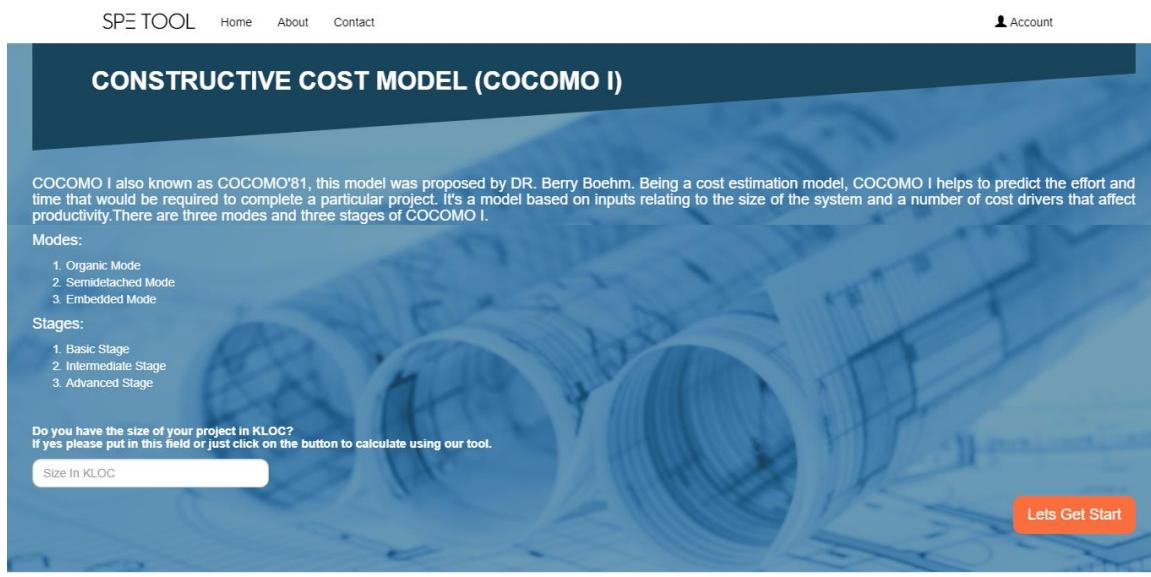


© 2018 - Software Project Estimation Tool

Base on selected process model, user is navigated to either COCOMO I, COCOMO II or SLIM.

- **COCOMO I**

(UI ID #3)



© 2018 - Software Project Estimation Tool

# Software Project Estimator

## ○ Select Mode

(UI ID #4)

The screenshot shows a web page titled 'SELECT MODE' with a sub-instruction: 'Please select a mode, according to your project type and requirements.' Below this, there are three cards representing different modes:

- ORGANIC MODE**: Describes a small team developing a well-understood application. It includes a 'SELECT' button.
- SEMITDETACHED MODE**: Describes a mixture of experienced and inexperienced staff, with team members having limited experience on related systems. It includes a 'SELECT' button.
- EMBEDDED MODE**: Describes complex hardware and software systems where team members may not have deep application experience. It includes a 'SELECT' button.

The background features a blue-toned architectural blueprint of a building's structural framework.

© 2018 - Software Project Estimation Tool

## ○ Select Stage

(UI ID #5)

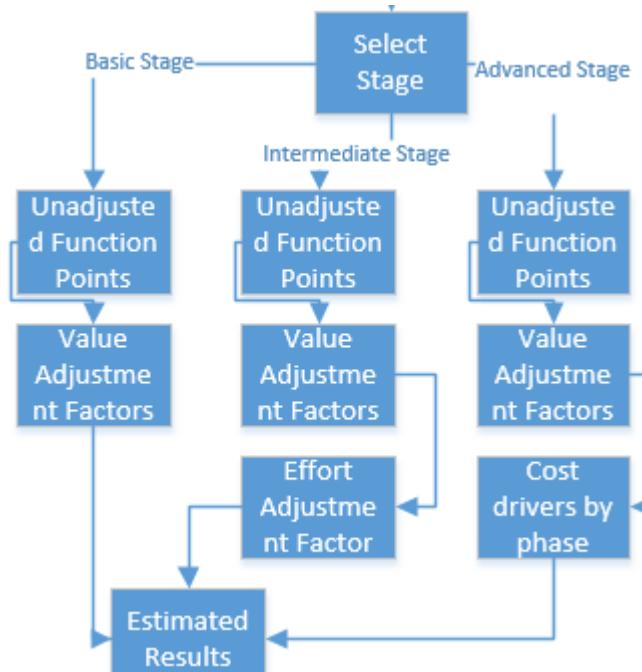
The screenshot shows a web page titled 'SELECT STAGE' with a sub-instruction: 'Please select the stage of your project, at which you want to perform estimation.' Below this, there are three cards representing different stages:

- BASIC STAGE**: Describes application early in the project, estimating parameters quickly. It includes a 'SELECT' button.
- INTERMEDIATE STAGE**: Describes application after requirements acquisition, using intermediate stage computation. It includes a 'SELECT' button.
- ADVANCED STAGE**: Describes application when design is complete, using a set of cost drivers weighted by phase. It includes a 'SELECT' button.

The background features a blue-toned architectural blueprint of a building's structural framework.

© 2018 - Software Project Estimation Tool

For different stages, different estimation steps take place.



- **Unadjusted Function Points**  
(UI ID #6)

SPE TOOL    Home    About    Contact   

### UNADJUSTED FUNCTION POINT (Basic Stage)

Enter the values in given fields according to your project requirements.

<b>EXTERNAL INPUTS</b>	<b>EXTERNAL OUTPUTS</b>	<b>EXTERNAL INQUIRIES</b>
Low	Low	Low
Average	Average	Average
High	High	High
<b>EXTERNAL INTERFACE FILES</b>		
<b>INTERNAL LOGICAL FILES</b>		
Low	Low	
Average	Average	
High	High	

**CALCULATE**

© 2018 - Software Project Estimation Tool

# Software Project Estimator

## ○ Value Adjustment Factors (UI ID #7)

SPE TOOL Home About Contact  Account

### VALUE ADJUSTMENT FACTOR (Basic Stage)

*Rate the following factors from 1-5, 1 means, the component is not relevant to the system and 5 means, component is essential:*

DATA COMMUNICATION	Select	ONLINE DATA ENTRY	Select	INSTALLATION EASE	Select
DISTRIBUTED DATA PROCESSING	Select	END USER EFFICIENCY	Select	OPERATIONAL EASE	Select
PERFORMANCE	Select	ONLINE UPDATES	Select	MULTIPLE SITES	Select
HEAVILY USED CONFIGURATION	Select	COMPLEX PROCESSING	Select	FACILITATE CHANGE	Select
TRANSACTION RATE	Select	REUSABILITY	Select		

**CALCULATE**

© 2018 - Software Project Estimation Tool

## ○ Effort Adjustment Factor (Cost Drivers) (UI ID #8)

SPE TOOL Home About Contact  Account

### EFFORT ADJUSTMENT FACTOR (Intermediate Stage)

*Please rate the following cost drivers according to your project specifications.*

PRODUCT ATTRIBUTES		PLATFORM ATTRIBUTES		PERSONNEL ATTRIBUTES		PROJECT ATTRIBUTES	
Cost Drivers	Weights	Cost Drivers	Weights	Cost Drivers	Weights	Cost Drivers	Weights
RELY	Select	TIME	Select	ACAP	Select	MODP	Select
DATA	Select	STOR	Select	AEXP	Select	TOOL	Select
CPLX	Select	VIRT	Select	PCAP	Select	SCED	Select
		TURN	Select	VEXP	Select		
				LEXP	Select		

**CALCULATE**

© 2018 - Software Project Estimation Tool

## Software Project Estimator

### ○ Effort Adjustment Factors (Cost Drivers by Phase) (UI ID #9)

SPE TOOL Home About Contact  Account

#### EFFORT ADJUSTMENT FACTOR (Advanced Stage)

The cost drivers are broken down by phase for advanced/detailed stage of COCOMO I, which are following below.

1. Requirements planning and product design (RPD)  
2. Detailed design (DD)  
3. Code and Unit Test (CUT)  
4. Integration and Test (IT)

##### Requirement Planning and Product Design

Product Attribute	Platform Attribute	Personnel Attribute	Project Attribute
Cost Drivers	Weights	Cost Drivers	Weights
RELY	Select	TIME	Select
DATA	Select	STOR	Select
CPLX	Select	VIRT	Select
		TURN	Select

##### Detailed Design (DD)

### ○ Result (Basic Stage) (UI ID #11)

SPE TOOL Home About Contact  Account

#### FINAL ESTIMATIONS (Basic Stage)

SIZE	EFFORT	DEVELOPMENT TIME
1359 KLOC	4.7 person/month	Not defined at this stage

[Generate Report](#)

© 2018 - Software Project Estimation Tool

# Software Project Estimator

## ○ Result (Intermediate Stage) (UI ID #12)

SPE TOOL Home About Contact  Account

### FINAL ESTIMATIONS (Intermediate Stage)



**SIZE:**  
1313 KLOC

**EFFORT:**  
12.5 person/month

**DEVELOPMENT TIME:**  
7 months

*6 hours per day needed to complete the project with in 7 months.*

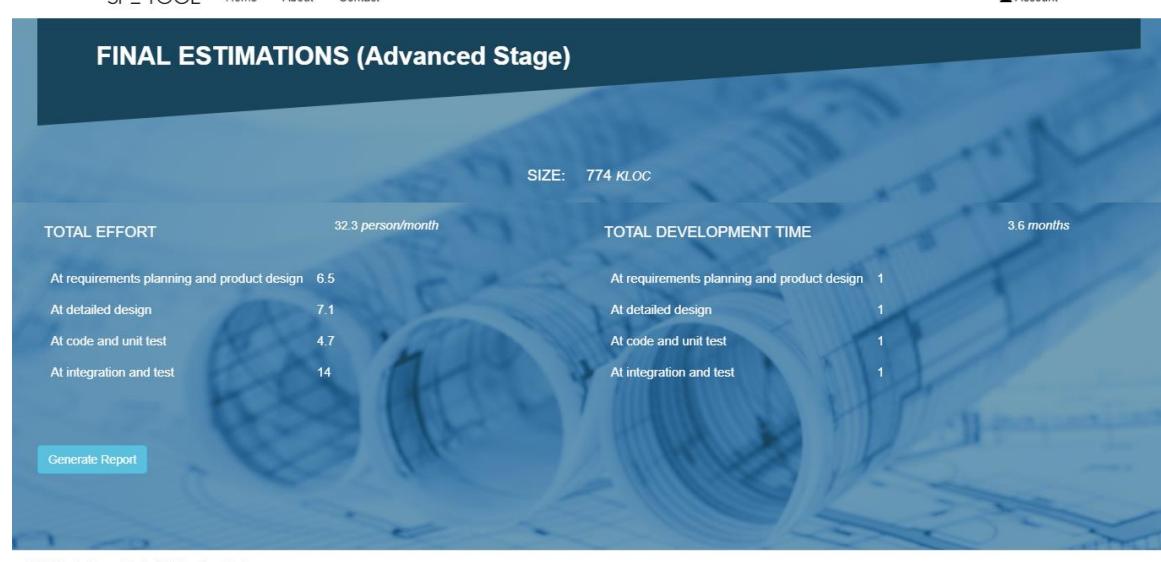
[Generate Report](#)

© 2018 - Software Project Estimation Tool

## ○ Result (Advance Stage) (UI ID #13)

SPE TOOL Home About Contact  Account

### FINAL ESTIMATIONS (Advanced Stage)



**SIZE:** 774 KLOC

TOTAL EFFORT	32.3 person/month	TOTAL DEVELOPMENT TIME	3.6 months
At requirements planning and product design	6.5	At requirements planning and product design	1
At detailed design	7.1	At detailed design	1
At code and unit test	4.7	At code and unit test	1
At integration and test	14	At integration and test	1

[Generate Report](#)

© 2018 - Software Project Estimation Tool

## Software Project Estimator

### o COCOMO II

(UI ID #14)

The screenshot shows the COCOMO II estimation tool. At the top, there's a navigation bar with 'SPE TOOL' and links for 'Home', 'About', and 'Contact'. On the right is an 'Account' icon. Below the navigation is a dark header bar with the text 'CONSTRUCTIVE COST MODEL (COCOMO II)' in white. The main content area features a background image of architectural blueprints. A text block explains the updated version of COCOMO, mentioning object-oriented software, reuse, and new systems. It also lists three models: Application Composition Model, Early Design Model, and Post Architecture. A message below states: 'You just have to choose a model according to your project requirements. The system will guide you at every step of the Estimation.' A green 'Lets Get Start' button is visible. At the bottom left, a copyright notice reads: '© 2018 - Software Project Estimation Tool'.

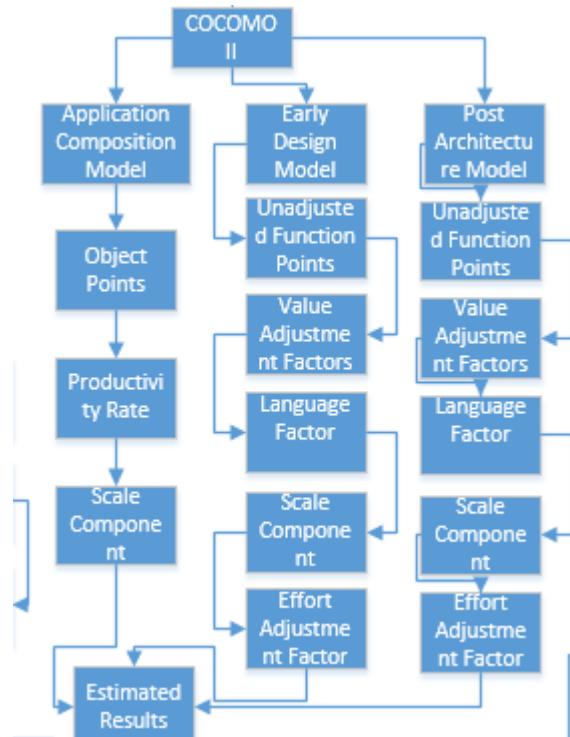
### o Select Mode

(UI ID #15)

The screenshot shows the 'Select Mode' section of the estimation tool. At the top, it has a navigation bar with 'SPE TOOL', 'Home', 'About', 'Contact', and an 'Account' icon. Below is a dark header bar with 'SELECT MODEL' in white. A sub-instruction 'Select a model, according to your project type & requirements.' is displayed. The main area contains three cards: 'APPLICATION COMPOSITION', 'EARLY DESIGN', and 'POST ARCHITECTURE'. Each card has a list of benefits and a 'SELECT' button. The 'APPLICATION COMPOSITION' card lists: 1. Use in early development stages. 2. Project build with modern GUI tools. 3. Intended for prototype projects, where the project is built by composing components called 'Object Points'. The 'EARLY DESIGN' card lists: 1. Use this model to get rough estimation of a project's cost and duration before determining its entire architecture. 2. It uses a small set of new Cost Drivers and new estimating equations. The 'POST ARCHITECTURE' card lists: 1. This is the most detailed COCOMO II model. 2. It will be used after development of project's overall architecture. 3. It has new Cost Drivers, new Line Counting Rules and new Equations. At the bottom left, a copyright notice reads: '© 2018 - Software Project Estimation Tool'.

## Software Project Estimator

For different Modes, different estimation steps take place.



### ○ Application Composition Model (UI ID #16)

SPE TOOL Home About Contact Account

## APPLICATION COMPOSITION MODEL

**NEW OBJECT POINT**

1. Enter Screens, Reports and Component Acquired.
2. Calculate Object Point.
3. Calculate Object Point and Component Acquired.

**PRODUCTIVITY RATE**

1. Weights developer's skills.
2. Rate Developer's skills Very Low to Very High.
3. Calculate productivity rate.

**EFFORT (ACM)**

1. Get calculated new object point value.
2. Get calculated Productivity Rate.
3. Calculate Effort.

Go

© 2018 - Software Project Estimation Tool

This screenshot shows the user interface for the Application Composition Model. It features three main input fields: 'NEW OBJECT POINT', 'PRODUCTIVITY RATE', and 'EFFORT (ACM)'. Each field contains a numbered list of steps to calculate its respective value. A 'Go' button is located in the bottom right corner of the input area. The background of the page is a blurred image of architectural blueprints.

# Software Project Estimator

## o Early Design Model

(UI ID #17)

The screenshot shows the 'EARLY DESIGN MODEL' section of the software. It features three main steps in a flow: 'KILO LINE OF CODE (KLOC)', 'EFFORT ADJUSTMENT FACTOR', and 'EFFORT (EDM)'. Each step has a list of tasks and a 'Go' button at the bottom right. The background is a blurred technical drawing of a mechanical part.

**KILO LINE OF CODE (KLOC)**

1. Calculate Language Factor.
2. Calculate Function Point through Unadjusted Function Point and Value Adjustment Factor.
3. Calculate KLOC using:  $KLOC = UFP * VAF$ .

**EFFORT ADJUSTMENT FACTOR**

1. Enter Product Factor, Personal Factor, Project Factor, and Platform Factor.
2. Calculate 15 Cost Drivers by entering different factors.
3. Calculate EAF by Taking product of all Cost Drivers.

**EFFORT (EDM)**

1. Get calculated KLOC value.
2. Get Calculated EAF value.
3. Calculate Effort using  $E = 2.45 * KLOC * EAF$ .

Go

© 2018 - Software Project Estimation Tool

## o Post Architecture Model

(UI ID #18)

The screenshot shows the 'POST ARCHITECTURE MODEL' section of the software. It features three main steps in a flow: 'KILO LINE OF CODE (KLOC)', 'SCALE COMPONENT (b), & EAF', and 'EFFORT (PAM)'. Each step has a list of tasks and a 'Go' button at the bottom right. The background is a blurred technical drawing of a mechanical part.

**KILO LINE OF CODE (KLOC)**

1. Calculate Language Factor.
2. Calculate Function Point through Unadjusted Function Point and Value Adjustment Factor.
3. Calculate KLOC using:  $KLOC = UFP * VAF$ .

**SCALE COMPONENT (b), & EAF**

1. Calculate (b) by entering Scale Factor's weight, using:  $b=0.91 + 0.01 * SF$ .
2. Enter Product Factor, Personal Factor, Project Factor, and Platform Factor.
3. Calculate EAF by Taking product of all Cost Drivers.

**EFFORT (PAM)**

1. Get calculated KLOC value.
2. Get Calculated (b) value.
3. Get Calculated EAF value.
4. Calculate Effort using  $E = 2.45 * (KLOC)^b * EAF$ .

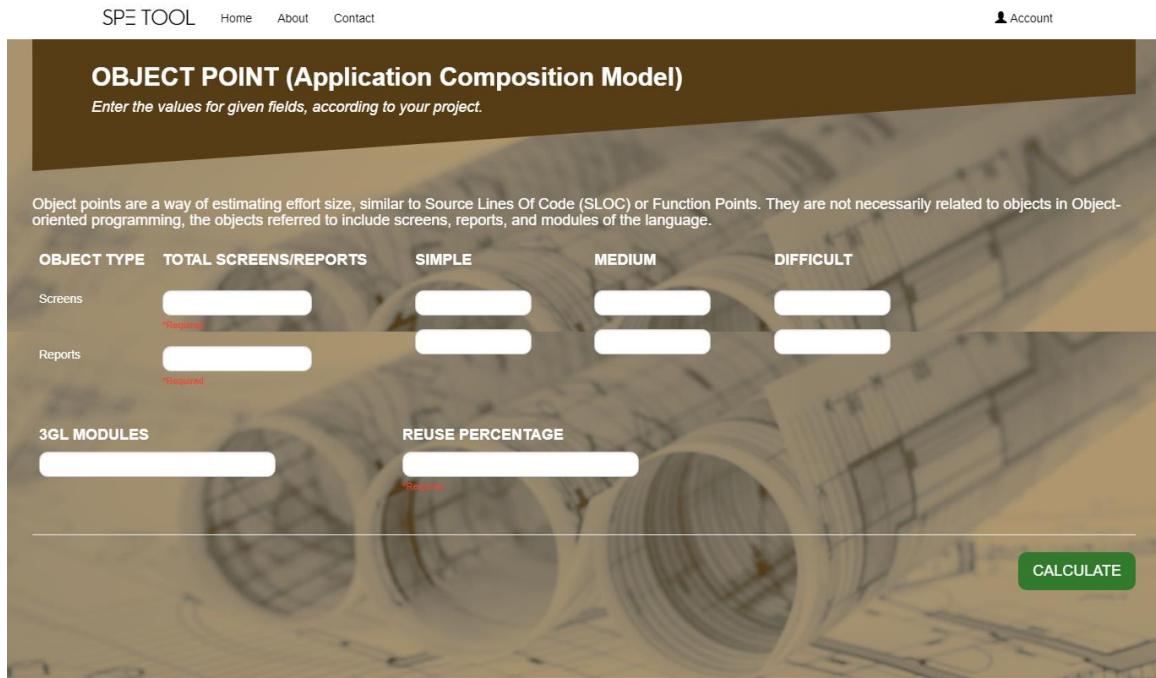
Go

© 2018 - Software Project Estimation Tool

## Software Project Estimator

### o Object Points

(UI ID #19)



SPE TOOL Home About Contact Account

### OBJECT POINT (Application Composition Model)

Enter the values for given fields, according to your project.

Object points are a way of estimating effort size, similar to Source Lines Of Code (SLOC) or Function Points. They are not necessarily related to objects in Object-oriented programming, the objects referred to include screens, reports, and modules of the language.

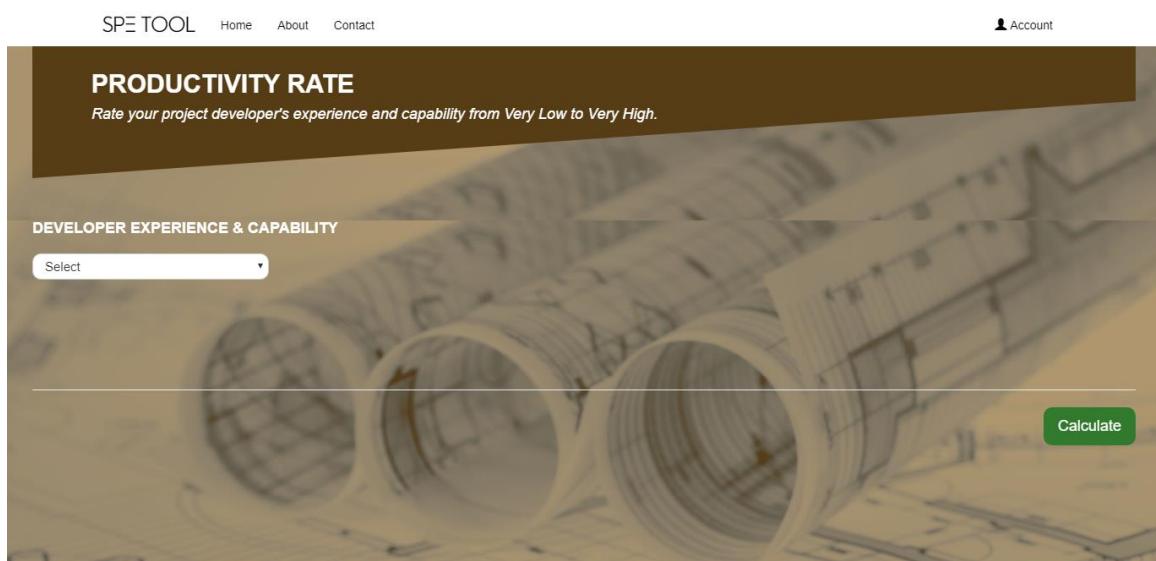
OBJECT TYPE	TOTAL SCREENS/REPORTS	SIMPLE	MEDIUM	DIFFICULT
Screens	<input type="text"/> *Required	<input type="text"/>	<input type="text"/>	<input type="text"/>
Reports	<input type="text"/> *Required	<input type="text"/>	<input type="text"/>	<input type="text"/>
3GL MODULES	<input type="text"/>	REUSE PERCENTAGE	<input type="text"/> *Required	

**CALCULATE**

© 2018 - Software Project Estimation Tool

### o Productivity Rate

(UI ID #20)



SPE TOOL Home About Contact Account

### PRODUCTIVITY RATE

Rate your project developer's experience and capability from Very Low to Very High.

DEVELOPER EXPERIENCE & CAPABILITY

Select

**Calculate**

© 2018 - Software Project Estimation Tool

## o Scale Component

(UI ID #21)

SPE TOOL Home About Contact Account

**SCALE COMPONENT (Application Composition Model)**

*Rate all Scale Factors from Very Low to Very High.*

SCALE FACTORS	WEIGHTS
Precededness	Select
Development/Flexibility	Select
Architecture/Risk Resolution	Select
Team Cohesion	Select
Process Maturity	Select

**CALCULATE**

© 2018 - Software Project Estimation Tool

## o Unadjusted Function Points

(UI ID #22)

SPE TOOL Home About Contact Account

**UNADJUSTED FUNCTION POINT (Early Design Model)**

*Enter the values in given fields according to your project requirements.*

EXTERNAL INPUTS	EXTERNAL OUTPUTS	EXTERNAL INQUIRIES
Low		
Average		
High		

EXTERNAL INTERFACE FILES	INTERNAL LOGICAL FILES
Low	
Average	
High	

**CALCULATE**

© 2018 - Software Project Estimation Tool

## Software Project Estimator

### ○ Value Adjustment Factors (UI ID #23)

SP<sup>E</sup> TOOL Home About Contact Account

**VALUE ADJUSTMENT FACTOR (Early Design Model)**

*Rate the following factors from 1-5, 1 means, the component is not relevant to the system and 5 means, component is essential:*

DATA COMMUNICATION	Select	ONLINE DATA ENTRY	Select	INSTALLATION EASE	Select
DISTRIBUTED DATA PROCESSING	Select	END USER EFFICIENCY	Select	OPERATIONAL EASE	Select
PERFORMANCE	Select	ONLINE UPDATES	Select	MULTIPLE SITES	Select
HEAVILY USED CONFIGURATION	Select	COMPLEX PROCESSING	Select	FACILITATE CHANGE	Select
TRANSACTION RATE	Select	REUSABILITY	Select		

**CALCULATE**

© 2018 - Software Project Estimation Tool

### ○ Language factor (UI ID #24)

SP<sup>E</sup> TOOL Home About Contact Account

**LANGUAGES FACTOR (Early Design Model)**

*Select a Programming Language used in your project.*

PROGRAMMING LANGUAGES	Select
-----------------------	--------

**CALCULATE**

© 2018 - Software Project Estimation Tool

## Software Project Estimator

### ○ Effort Adjustment Factor (Cost Drivers) (UI ID #25)

SPE TOOL    Home    About    Contact     Account

#### EFFORT ADJUSTMENT FACTOR (Early Design Model)

Please rate the following cost drivers according to your project specifications.



Cost Drivers	Weights
RELY	Select
DATA	Select
CPLX	Select

Cost Drivers	Weights
TIME	Select
STOR	Select
VIRT	Select
TURN	Select

Cost Drivers	Weights
ACAP	Select
AEXP	Select
PCAP	Select
VEXP	Select
LEXP	Select

Cost Drivers	Weights
MODP	Select
TOOL	Select
SCED	Select

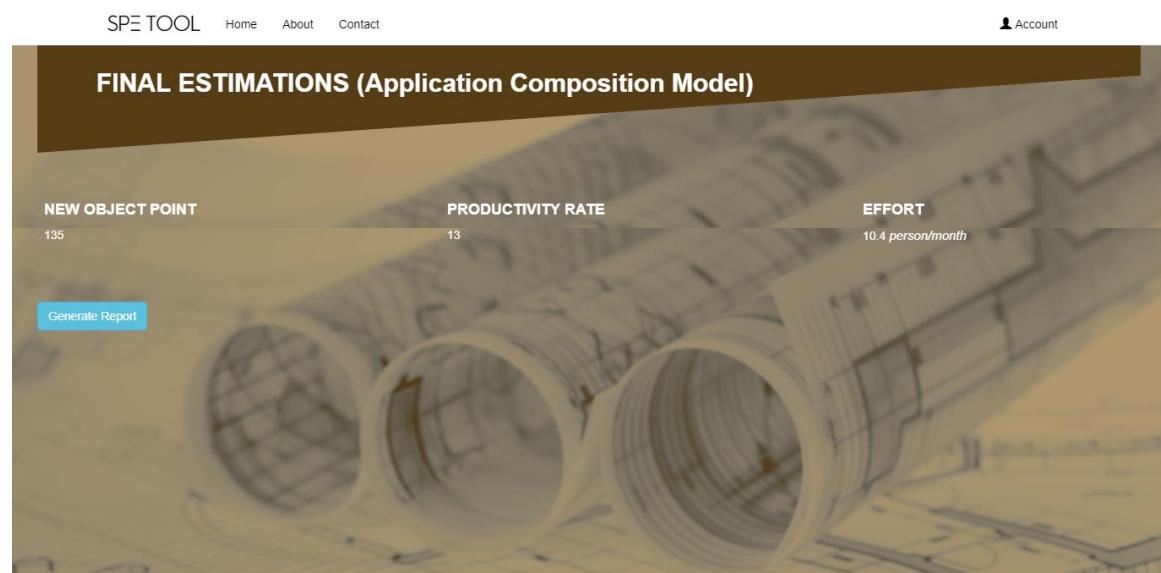
**CALCULATE**

© 2018 - Software Project Estimation Tool

### ○ Result (Application Composition Model) (UI ID #26)

SPE TOOL    Home    About    Contact     Account

#### FINAL ESTIMATIONS (Application Composition Model)



NEW OBJECT POINT	PRODUCTIVITY RATE	EFFORT
135	13	10.4 person/month

Generate Report

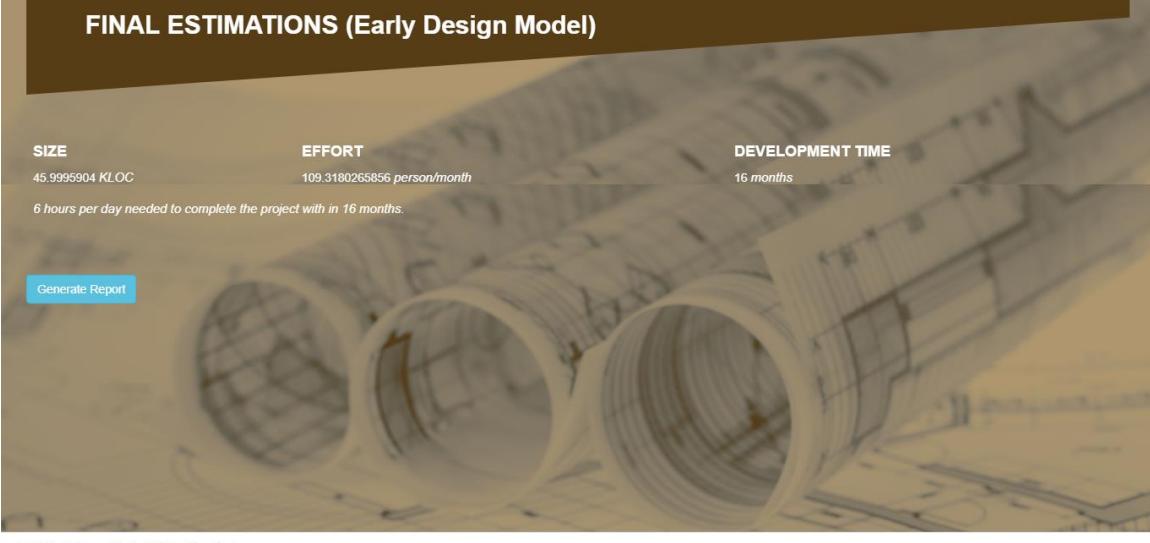
© 2018 - Software Project Estimation Tool

# Software Project Estimator

## ○ Result (Early Design Model) (UI ID #27)

SPE TOOL Home About Contact  Account

### FINAL ESTIMATIONS (Early Design Model)



SIZE	EFFORT	DEVELOPMENT TIME
45.9995904 KLOC	109.3180265856 person/month	16 months

*6 hours per day needed to complete the project with in 16 months.*

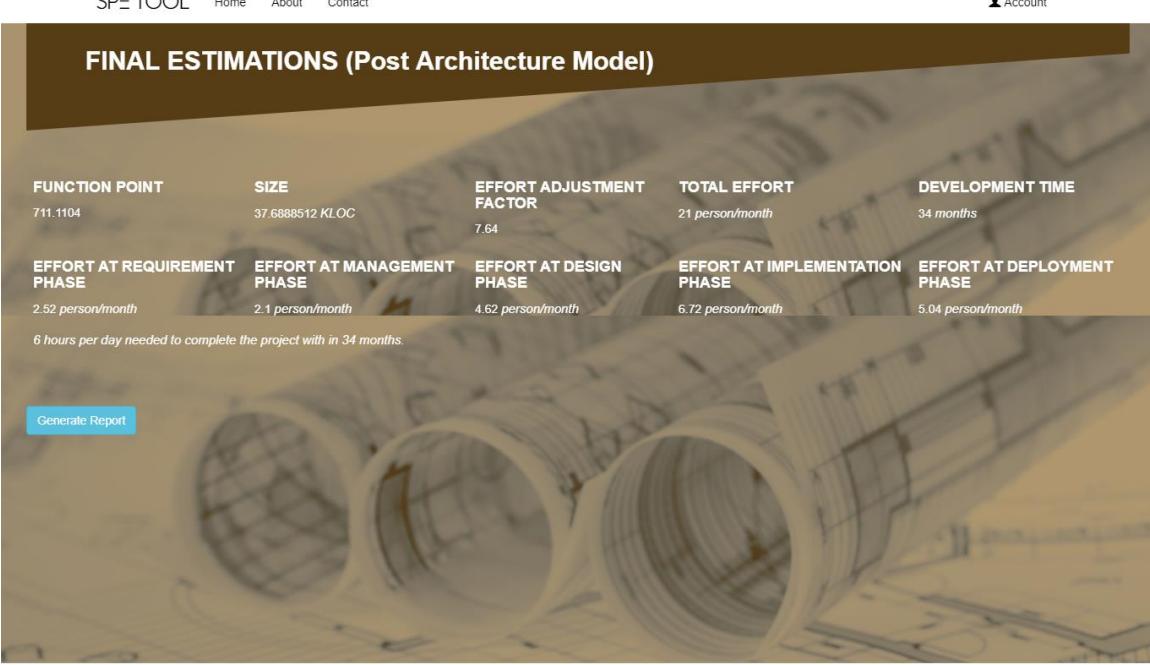
[Generate Report](#)

© 2018 - Software Project Estimation Tool

## ○ Result (Post Architecture Model) (UI ID #28)

SPE TOOL Home About Contact  Account

### FINAL ESTIMATIONS (Post Architecture Model)



FUNCTION POINT	SIZE	EFFORT ADJUSTMENT FACTOR	TOTAL EFFORT	DEVELOPMENT TIME
711.1104	37.6888512 KLOC	7.64	21 person/month	34 months

EFFORT AT REQUIREMENT PHASE	EFFORT AT MANAGEMENT PHASE	EFFORT AT DESIGN PHASE	EFFORT AT IMPLEMENTATION PHASE	EFFORT AT DEPLOYMENT PHASE
2.52 person/month	2.1 person/month	4.62 person/month	6.72 person/month	5.04 person/month

*6 hours per day needed to complete the project with in 34 months.*

[Generate Report](#)

© 2018 - Software Project Estimation Tool

## Software Project Estimator

### o SLIM

(UI ID #29)

SLIM is an automated costing system based on the Rayleigh-Putnam Model. SLIM applies the Putnam software model, linear programming, statistical simulation, and program evaluation and review technique (PERT) techniques to derive software project estimates. SLIM works reasonably well for very large systems, but seriously overestimates the effort for medium and small systems. Some versions of SLIM use a so-called skill factor  $\beta$ , which is a scaling term and is a function of project size. Software Lifecycle Management Model contains three phases:

- 1. Process Productivity
- 2. Size in LOC
- 3. Manpower Equation

You just have to choose a model according to your project requirements. The system will guide you at every step of the Estimation.

Lets Get Start

© 2018 - Software Project Estimation Tool

### o Process Productivity

(UI ID #30)

PROCESS PRODUCTIVITY

Please select your application type below.

APPLICATION TYPE:

Select

Calculate

© 2018 - Software Project Estimation Tool

- **Unadjusted Function Points**  
(UI ID #31)

SPE TOOL    Home    About    Contact     Account

### UNADJUSTED FUNCTION POINT (Post Architecture Model)

*Enter the values in given fields according to your project requirements.*

<b>EXTERNAL INPUTS</b>	<b>EXTERNAL OUTPUTS</b>	<b>EXTERNAL INQUIRIES</b>
<i>Low</i> <input type="text"/>	<i>Low</i> <input type="text"/>	<i>Low</i> <input type="text"/>
<i>Average</i> <input type="text"/>	<i>Average</i> <input type="text"/>	<i>Average</i> <input type="text"/>
<i>High</i> <input type="text"/>	<i>High</i> <input type="text"/>	<i>High</i> <input type="text"/>
<b>EXTERNAL INTERFACE FILES</b>		
<b>INTERNAL LOGICAL FILES</b>		
<i>Low</i> <input type="text"/>	<i>Low</i> <input type="text"/>	<i>Low</i> <input type="text"/>
<i>Average</i> <input type="text"/>	<i>Average</i> <input type="text"/>	<i>Average</i> <input type="text"/>
<i>High</i> <input type="text"/>	<i>High</i> <input type="text"/>	<i>High</i> <input type="text"/>

**CALCULATE**

© 2018 - Software Project Estimation Tool

- **Value Adjustment Factors**  
(UI ID #32)

SPE TOOL    Home    About    Contact     Account

### VALUE ADJUSTMENT FACTOR (Post Architecture Model)

*Rate the following factors from 1-5, 1 means, the component is not relevant to the system and 5 means, component is essential:*

DATA COMMUNICATION	ONLINE DATA ENTRY	INSTALLATION EASE	Select ▾
DISTRIBUTED DATA PROCESSING	END USER EFFICIENCY	OPERATIONAL EASE	Select ▾
PERFORMANCE	ONLINE UPDATES	MULTIPLE SITES	Select ▾
HEAVILY USED CONFIGURATION	COMPLEX PROCESSING	FACILITATE CHANGE	Select ▾
TRANSACTION RATE	REUSABILITY		

**CALCULATE**

© 2018 - Software Project Estimation Tool

## Software Project Estimator

### o Result (SLIM)

(UI ID #33)

The screenshot shows the 'FINAL ESTIMATIONS (SLIM)' page. At the top, there are four summary statistics:

SIZE	PROCESS PRODUCTIVITY	TOTAL EFFORT	EFFORT AT DELIVERY TIME
775680 (KLOC)	9308160	690.745496095722 person/month	271.817332405116 person/month

Below these is a section titled 'Manpower Build Up Equation' with a detailed explanation of Putman's manpower acceleration equation. The equation is:

$$E = 0.399736976907246 \cdot S^{1.23}$$

where  $E$  is effort and  $S$  is size.

At the bottom left, it says '© 2018 - Software Project Estimation Tool'.

### o REPORT

(UI ID #34)

The screenshot shows the 'Post Architecture Model - COCOMO II' report. It includes sections for 'Project Details', 'Unadjusted Function Points', and 'Final Estimations'.

**Project Details:**

- Project Name: Android Task Monitoring
- Process Model: Prototyping

**Unadjusted Function Points:**

Elements	Complexity	Weighting Factors	
	Low	Average	High
External Inputs (EI)	1	1	1
External Outputs (EO)	1	1	1
External Inquiries (EQ)	1	1	1
External Interface files (EIF)	1	1	1
Internal Logical files (ILF)	1	1	1
Total:	96		

**Final Estimations:**

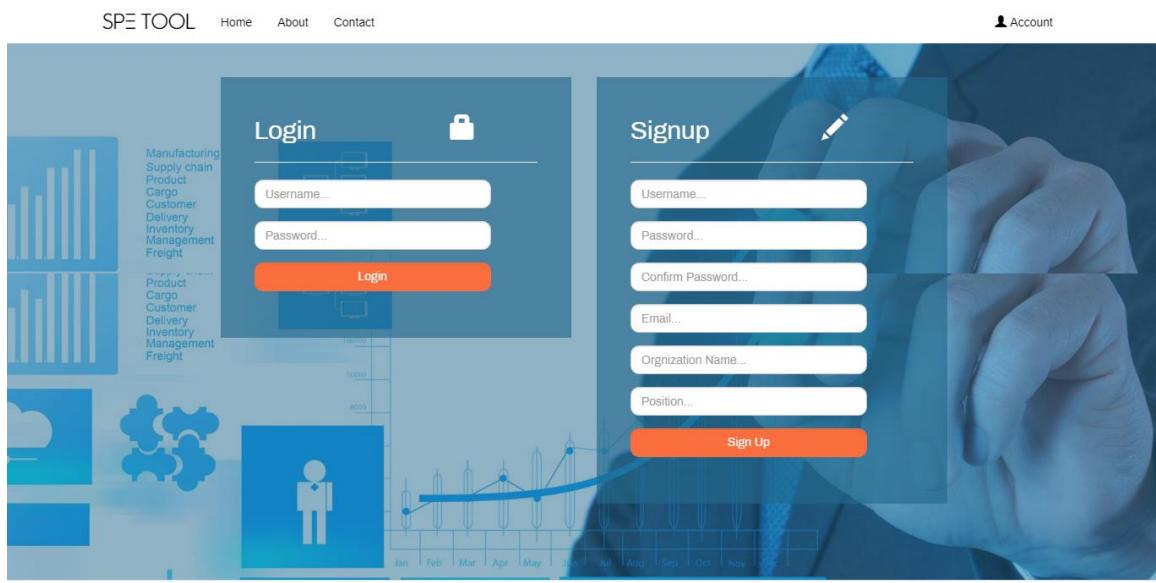
- Size: 7.3574112 (KLOC)
- Effort: 54.07697232 (Person/Month)
- Total Effort: 4 (Person/Month)
- Development Time: 13 (Months)

At the bottom right, there is a button labeled 'Login to save'.

## Software Project Estimator

### o Login / Signup

User need to log in save a report. New user need to be registered. (UI ID #35)



© 2018 - Software Project Estimation Tool

### o Profile

(UI ID #36)

The image shows a user profile page for 'ahmed'. At the top, there's a dark header with the user's name 'ahmed' and a 'Logout' link. Below the header is a sidebar with 'Dashboard' and links for 'Account Overview', 'User Info', and 'Projects'. The main content area has three sections: 'Account Overview' showing 'SKT Organization' and 'CEO Position', 'User Info' showing a table with one row for 'ahmed', and 'Projects' showing a list of five projects: 'Student Management', 'IDM', 'IM', 'LLP', and 'Android Task Monitoring', each with a 'View Details' link. The background is a blue-toned architectural drawing.

© 2018 - Software Project Estimation Tool

## **4.5 Trace-ability Matrix**

**Following columns are involved in the trace-ability matrix.**

**1.**

<b>Feature:</b>	<b>choose best suited estimation technique.</b>
<b>Use Case ID:</b>	UC-2
<b>UI ID:</b>	(UI ID #2)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	UC_Enter_Project_Name
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	UC-1
<b>Dependent Classes:</b>	ProjectNameModel

**2.**

<b>Feature:</b>	<b>Select COCOMOI mode</b>
<b>Use Case ID:</b>	UC-14
<b>UI ID:</b>	(UI ID #4)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	
<b>Dependent Classes:</b>	Nill

**3.**

<b>Feature:</b>	<b>Select COCOMOI stage</b>
<b>Use Case ID:</b>	UC-15
<b>UI ID:</b>	(UI ID #5)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	Nill

**4.**

<b>Feature:</b>	<b>Unadjusted Function points will be calculated here.</b>
<b>Use Case ID:</b>	UC-3, UC-4, UC-5, UC-6, UC-7, UC-8
<b>UI ID:</b>	(UI ID #6) , (UI ID #22), (UI ID #31)
<b>Priority:</b>	High

<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	UnadjustedFunctionPointsModel

5.

<b>Feature:</b>	<b>Value Adjustment Factors will be calculated here.</b>
<b>Use Case ID:</b>	UC-9
<b>UI ID:</b>	(UI ID #7), (UI ID #23), (UI ID #32)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	ValueAdjustmentFactorsModel

6.

<b>Feature:</b>	<b>Cost Factors will be calculated here.</b>
<b>Use Case ID:</b>	UC-10, UC-11, UC-12, UC-13, UC-17
<b>UI ID:</b>	(UI ID #8) (UI ID #25)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	CostDriversModel

7.

<b>Feature:</b>	<b>result for COCOMOI (basic) Model will be calculated here.</b>
<b>Use Case ID:</b>	UC-16, UC-18
<b>UI ID:</b>	(UI ID #11)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	Nill

8.

<b>Feature:</b>	<b>result for COCOMOI (Intermediate) Model will be calculated here.</b>
<b>Use Case ID:</b>	UC-19, UC-21, UC-22
<b>UI ID:</b>	(UI ID #12)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill

<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	Nill

9.

<b>Feature:</b>	<b>result for COCOMOI (Advance) Model will be calculated here.</b>
<b>Use Case ID:</b>	UC-23,UC-29,UC-30
<b>UI ID:</b>	(UI ID #13)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	Nill

10.

<b>Feature:</b>	<b>Cost Drivers by phase will be calculated here.</b>
<b>Use Case ID:</b>	UC-24 UC-25 UC-26 UC-27 Uc-28
<b>UI ID:</b>	(UI ID #9)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	CostDriversByPhaseModel

11.

<b>Feature:</b>	<b>Scale component will be calculated here.</b>
<b>Use Case ID:</b>	UC-31 UC-32 UC-33 UC-34 UC-35 UC-41
<b>UI ID:</b>	(UI ID #21)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	Nill

12.

<b>Feature:</b>	<b>New object point will be calculated here.</b>
<b>Use Case ID:</b>	UC-37, UC-38, UC-39
<b>UI ID:</b>	(UI ID #19)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill

<b>Dependent Classes:</b>	Nill
---------------------------	------

13.

<b>Feature:</b>	Productivity rate will be calculated here.
<b>Use Case ID:</b>	UC-40
<b>UI ID:</b>	(UI ID #20)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	Nill

14.

<b>Feature:</b>	result for COCOMOII (ACM) Model will be calculated here.
<b>Use Case ID:</b>	UC-42
<b>UI ID:</b>	(UI ID #26)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	Nill

15.

<b>Feature:</b>	result for COCOMOII (EDM) Model will be calculated here.
<b>Use Case ID:</b>	UC-44, UC-46
<b>UI ID:</b>	(UI ID #27)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	Nill

16.

<b>Feature:</b>	result for COCOMOII (ACM) Model will be calculated here.
<b>Use Case ID:</b>	UC-42
<b>UI ID:</b>	(UI ID #26)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	Nill

**17.**

<b>Feature:</b>	<b>result for COCOMOII (PAM) Model will be calculated here.</b>
<b>Use Case ID:</b>	UC-48, UC-51, UC-52
<b>UI ID:</b>	(UI ID #28)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	Nill

**18.**

<b>Feature:</b>	<b>Application type will be selected here.</b>
<b>Use Case ID:</b>	UC-53
<b>UI ID:</b>	(UI ID #30)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	Nill

**19.**

<b>Feature:</b>	<b>result for SLIM Model will be calculated here.</b>
<b>Use Case ID:</b>	UC-54, UC-55, UC-56, UC-57, UC-58, UC-59
<b>UI ID:</b>	(UI ID #33)
<b>Priority:</b>	High
<b>Use Case Cross Ref:</b>	Nill
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	Nill
<b>Dependent Classes:</b>	Nill

**20.**

<b>Feature:</b>	<b>User registration</b>
<b>Use Case ID:</b>	UC-61
<b>UI ID:</b>	(UI ID #35)
<b>Priority:</b>	Medium
<b>Use Case Cross Ref:</b>	UC_Login
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	UC-62
<b>Dependent Classes:</b>	UserModel

**21.**

<b>Feature:</b>	<b>Login</b>
<b>Use Case ID:</b>	UC-62
<b>UI ID:</b>	(UI ID #35)

<b>Priority:</b>	Medium
<b>Use Case Cross Ref:</b>	UC_ Registration
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	UC-61
<b>Dependent Classes:</b>	UserModel

22.

<b>Feature:</b>	<b>Profile</b>
<b>Use Case ID:</b>	UC-66, UC-67
<b>UI ID:</b>	(UI ID #36)
<b>Priority:</b>	Medium
<b>Use Case Cross Ref:</b>	UC_ Login
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	UC-61
<b>Dependent Classes:</b>	Nill

23.

<b>Feature:</b>	<b>Report will be generated here.</b>
<b>Use Case ID:</b>	UC-64, UC-65
<b>UI ID:</b>	(UI ID #34)
<b>Priority:</b>	Medium
<b>Use Case Cross Ref:</b>	UC_ Login
<b>DB Table Id:</b>	Nill
<b>Elaborated Use-case ID:</b>	UC-61
<b>Dependent Classes:</b>	Nill

## **Chapter 5: Software Testing**

---

## **5.1. *Introduction:***

This deliverable is based on the IEEE standard of software testing i.e. IEEE SOFTWARE TEST DOCUMENTATION Std 829-1998. This standard describes a set of basic test documents that are associated with the dynamic aspects of software testing (i.e., the execution of procedures and code). The standard defines the purpose, outline, and content of each basic document. While the documents described in the standard focus on dynamic testing, several of them may be applicable to other testing activities (e.g., the test plan and test incident report may be used for design and code reviews). This standard may be applied to commercial, scientific, or military software that runs on any digital computer. Applicability is not restricted by the size, complexity, or criticality of the software. However, the standard does not specify any class of software to which it must be applied. The standard addresses the documentation of both initial development testing and the testing of subsequent software releases. For a particular software release, it may be applied to all phases of testing from module testing through user acceptance. However, since all of the basic test documents may not be useful in each test phase, the particular documents to be used in a phase are not specified. Each organization using the standard will need to specify the classes of software to which it applies and the specific documents required for a particular test phase.

The standard does not call for specific testing methodologies, approaches, techniques, facilities, or tools, and does not specify the documentation of their use. Additional test documentation may be required (e.g., code inspection checklists and reports). The standard also does not imply or impose specific methodologies for documentation control, configuration management, or quality assurance. Additional documentation (e.g., a quality assurance plan) may be needed depending on the particular methodologies used.

Following are standard artifacts, which must be included in this deliverable:

1. Test Plan
2. Test Design Specification
3. Test Case Specification
4. Test Procedure Specification
5. Test Item Transmittal Report
6. Test Log
7. Test Incident Report
8. Test Summary Report

## **5.2. *Test plan***

### **5.2.1. Purpose**

Software Estimation Tool helps in the estimation of size, effort, cost and development time of a project at different phases of design and implementation, that too, beforehand. The estimation will be done using three techniques; COCOMO I, COCOMO II and SLIM. All three of these accommodate different design process models.

### **5.2.2. Outline**

A test plan shall have the following structure:

- a. Test plan identifier
- b. Introduction
- c. Test items

- d. Features to be tested
- e. Features not to be tested
- f. Approach
- g. Item pass/fail criteria
- h. Suspension criteria and resumption requirements
- i. Test deliverables
- j. Testing tasks
- k. Environmental needs
- l. Responsibilities
- m. Staffing and training needs
- n. Schedule
- o. Risks and contingencies
- p. Approvals

The sections shall be ordered in the specified sequence. Additional sections may be included immediately prior to Approvals. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content. The referenced material must be attached to the test plan or available to users of the plan.

Details on the content of each section are contained in the following sub-clauses.

### **5.2.2.1. Test plan identifier**

Master plan for Software Estimation Tool is SPETOOL-01.

### **5.2.2.2. Introduction**

Summarize the software items and software features to be tested. The need for each item and its history may be included. References to the following documents, when they exist, are required in the highest-level test plan:

- a. Project authorization;
- b. Project plan;
- c. Quality assurance plan;
- d. Configuration management plan;
- e. Relevant policies;
- f. Relevant standards.

In multilevel test plans, each lower-level plan must reference the next higher-level plan.

### **5.2.2.3. Test items**

There are following test items which will be tested:

- As a user, logging into the website
- As a user, navigating the website
- As a user, generate estimation report
- As a user, save estimation report
- User can view saved project reports
- Can system predict development size accurately?
- Can system predict development time accurately?
- Can system predict development cost accurately?

#### **5.2.2.4. Features to be tested**

- login page, registration, profile and reports etc.
- As a user, navigating the website
- Can system predict development time accurately?
- User can save/ view project reports
- Can system predict development size accurately?
- Can system predict development cost accurately?

#### **5.2.2.5. Features not to be tested**

- If the system goes down can it come up with in 10 minutes? This feature cannot be tested as we cannot simulate an actual server crash and try and recover it.
- How does the system performance vary during high loads? This non-functional feature of performance under higher loads could not be tested due to lack of time and resources to generate large volumes of incident report requests.

#### **5.2.2.6. Approach**

Testing is the process of executing with intent of finding errors rather than showing correct functionality of the modules. Test will be conducted as per documented test cases. Each member will test each feature and mark each case as Pass/Fail. Each tester will note the actual result and all relevant details. Once all tests will complete, the test manager will review the test report to the team accordingly.

Testing should have following objective

- To reveal design errors
- To reveal logic errors
- To reveal security loopholes
- To reveal operational deficiencies

The test approach describes the scope of system testing, overall strategy to be adopted and the resources to be required.

#### **5.2.2.7. Item pass/fail criteria**

All core functionality of the systems should function as expected and outlined in the individual test cases. There must be no critical defects found and an end user must be able to complete an estimation cycle successfully and generate a report without any errors. 95% of all test cases should pass and no failed cases should be crucial to the end-user's ability to use the application.

#### **5.2.2.8. Suspension criteria and resumption requirements**

In this section we have to specify when to stop testing. If any major functionalities are not functional or system experiences login issues, then testing should suspend. Testing should be paused immediately if either system experiences login issues or failure in any ESTIMATION steps.

#### **5.2.2.9. Test deliverables**

Upon completion, the test run results will be saved in Test Lodge and the test manager should then run a report for all completed tests. List of documents need to be delivered at each phase of testing life cycle. The list of all test artifacts. Following is list of test deliverables:

- a. Test plan;
- b. Test design specifications;
- c. Test case specifications;

- d. Test procedure specifications;
- e. Test item transmittal reports;
- f. Test logs;
- g. Test incident reports;
- h. Test summary reports.

#### **5.2.2.10. Testing tasks**

Specify the list of testing tasks we need to complete in the current project. Test environment should be ready prior to test execution phase.

The following activities must be completed:

- Test plan prepared.
- Functional specifications written and delivered to the testing team
- Environment should be ready for testing (test data, test logins, test payment information, etc.).
- Perform the tests.
- Prepare test summary report.

#### **5.2.2.11. Environmental needs**

##### ***Hardware requirements***

- Laptop, smartphone or any browser supported device.

##### ***Software Requirements***

- Google Chrome, Microsoft Edge or any browser

#### **5.2.2.12 Responsibility**

Test plan should be prepared by test lead. Preparation and execution of tests should be carried out by testers. The test manager is responsible for facilitating the testing project, coordinating availability and schedule of testers and training them as needed. Each tester should understand the expectations on completion date and level of quality. The Test Manager should also communicate any risks to the team.

#### **5.2.2.13 Staffing and training needs**

Testing should be done by three testers. Testers should conduct testing on each system. It is preferred that there will be at least one of the testers assigned to the project for the system/integration and unit testing phases of the project. This will require assignment of a person in the full project from the beginning of the project to participate in reviews, development etc. Two other members will be assigned more into testing. If a separate test person is not available, the test lead will assume this role.

In order to provide complete and proper testing the following areas need to be addressed in terms of training:

- The developers and tester(s) will need to be trained on the basic operations of the Software estimation tool. Prior to final acceptance of the project the operations staff will also require complete training on the Software estimation tool process.
- The project manager will need training in using a Software estimation tool.

#### **5.2.2.14. Schedule**

Complete details on when to start, finish and how much time each task should take place. Testing will take place 4 weeks prior to the launch date. The first round of testing should be completed in 1 week.

### **5.2.2.15. Risks and contingencies**

In case of a wrong budget estimation, the cost may overrun. Contingency Plan establish the scope before beginning the testing tasks and pay attention in the project planning and also track the budget estimates constantly.

There are numerous issues to be addressed before deploying the Software estimation tool process as a defective system will lead to wrong estimation that may cause in overrun budget and cost. The various risks should be considered and the system should be tested properly. The other things which should also be checked are as follows:

- The Estimation Algorithms should be functioning properly and efficiently.
- The server system on which the system deployed must be fast and efficient.
- Unable to acquire the necessary number of skilled personnel as the components become ready to test.
- Unable to acquire some of the necessary hardware and software required for integration and system testing.
- Third party services utilized in the system become unavailable during testing.
- Components are not delivered on time.

### **5.2.2.16 Approvals**

**Name:** Ms. Mahum Adil

**Title:** Project Supervisor

**Signature:** \_\_\_\_\_

**Date:** \_\_\_\_\_

## **5.3. Test Case Specification**

### **5.3.1. Purpose**

To define a test case identified by a test design specification. The purpose of test case specification is to guide the end user about the system usage.

### **5.3.2. Outline**

A test case specification shall have the following structure:

- a. Test case specification identifier
- b. Test items
- c. Input specifications
- d. Output specifications
- e. Environmental needs
- f. Special procedural requirements
- g. Inter case dependencies

**Test Case 1:**

**Software:** Software Project Estimator

**Test ID:** 1

**Test Name:** Enter Project Name

Pre-conditions	User visits website.
Action	<ol style="list-style-type: none"> <li>1. User enters the project name in the provided textbox.</li> <li>2. User clicks on ‘Start Estimation’ button without entering name.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Project name is entered successfully.</li> <li>2. Error message displayed; ‘Required’</li> </ol>
Result	Pass.

**Test Description:** The test verifies that the user enters the project name in the textbox provided.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date:** 23 July, 2018

**Test Case 2:**

**Software:** Software Project Estimator

**Test ID:** 2

**Test Name:** Select Process Model

Pre-conditions	User visits website.
Action	<ol style="list-style-type: none"> <li>1. User selects the process model from the provided dropdown menu.</li> <li>2. User clicks on ‘next’ button without selecting process model.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. User is taken to the estimation technique that best suits the selected process model.</li> <li>2. Error message displayed; ‘Please select an item in the list!’</li> </ol>
Result	Pass.

**Test Description:** The test verifies that the user selects the process model from the dropdown menu and is taken to the estimation technique that best suits it.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Test Case 3:**

**Software:** Software Project Estimator

**Test ID:** 3

**Test Name:** Select Project Mode

Pre-conditions	User selects waterfall model from the dropdown menu. User is directed to COCOMO I.
Action	User selects the project mode from the three modes given i.e., organic mode, semi-detached mode, and embedded mode.
Expected Results	Project mode is selected for further estimation and user is taken to the next step.
Result	Pass.

**Test Description:** The test verifies that the user selects the project mode from the three options and is taken to the next step.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date:** 23 July, 2018

**Test Case 4:**

**Software:** Software Project Estimator

**Test ID:** 4

**Test Name:** Select COCOMO I Stage

Pre-conditions	User is directed to COCOMO I. User selects project mode.
Action	User selects the desired stage for which the estimation needs to be done. These three stages are; Basic, Intermediate, and Advanced.
Expected Results	Desired stage is selected for further estimation and user is taken to the next step.
Result	Pass.

**Test Description:** The test verifies that the user selects the desired stage of estimation from the three options and is taken to the next step.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date:** 23 July, 2018

**Test Case 5:**

**Software:** Software Project Estimator

**Test ID:** 5

**Test Name:** Enter External Inputs

Pre-conditions	User selects project mode. User selects estimation stage.
Action	<ol style="list-style-type: none"> <li>1. User will enter the number (whole numbers) of external input files of low, average and high complexity.</li> <li>2. Enter alphabets in text boxes.</li> <li>3. Enter special characters in text boxes.</li> <li>4. Enter decimal points in text boxes.</li> <li>5. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>6. Leave textbox empty.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Values are entered successfully.</li> <li>2. Error message displayed; ‘Enter a whole no’.</li> <li>3. Error message displayed; ‘Enter a whole no’.</li> <li>4. Error message displayed; ‘Enter a whole no’.</li> <li>5. Error message displayed; ‘Enter a whole no’.</li> <li>6. Error message displayed; ‘Required’.</li> </ol>
Result	Pass.

**Test Description:** The test verifies the entering of external inputs of low, average and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date:** 23 July, 2018

**Test Case 6:**

**Software:** Software Project Estimator

**Test ID:** 6

**Test Name:** Enter External Outputs

Pre-conditions	User selects project mode. User selects estimation stage.
----------------	--

## Software Project Estimator

Action	<ol style="list-style-type: none"> <li>1. User will enter the number (whole numbers) of external input files of low, average and high complexity.</li> <li>2. Enter alphabets in text boxes.</li> <li>3. Enter special characters in text boxes.</li> <li>4. Enter decimal points in text boxes.</li> <li>5. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>6. Leave textbox empty.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Values are entered successfully.</li> <li>2. Error message displayed; ‘Enter a whole no’.</li> <li>3. Error message displayed; ‘Enter a whole no’.</li> <li>4. Error message displayed; ‘Enter a whole no’.</li> <li>5. Error message displayed; ‘Enter a whole no’.</li> <li>6. Error message displayed; ‘Required’.</li> </ol>
Result	Pass.

### Test case

**Test Description:** The test verifies the entering of external outputs of low, average and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date: 23 July, 2018**

### Test Case 7:

**Software:** Software Project Estimator

**Test ID:** 7

**Test Name:** Enter External Inquiries

Pre-conditions	User selects project mode. User selects estimation stage.
Action	<ol style="list-style-type: none"> <li>1. User will enter the number of external inquiries of low, average and high complexity.</li> <li>2. Enter alphabets in text boxes.</li> <li>3. Enter special characters in text boxes.</li> <li>4. Enter decimal points in text boxes.</li> <li>5. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>6. Leave textbox empty.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Values are entered successfully.</li> <li>2. Error message displayed; ‘Enter a whole no’.</li> <li>3. Error message displayed; ‘Enter a whole no’.</li> <li>4. Error message displayed; ‘Enter a whole no’.</li> <li>5. Error message displayed; ‘Enter a whole no’.</li> <li>6. Error message displayed; ‘Required’.</li> </ol>
Result	Pass.

**Test Description:** The test verifies the entering of external inquiries of low, average and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date: 23 July, 2018**

**Test Case 8:**

**Software:** Software Project Estimator

**Test ID:** 8

**Test Name:** Enter External Interface Files

Pre-conditions	User selects project mode. User selects estimation stage.
Action	<ol style="list-style-type: none"> <li>1. User will enter the number of external interface files of low, average and high complexity.</li> <li>2. Enter alphabets in text boxes.</li> <li>3. Enter special characters in text boxes.</li> <li>4. Enter decimal points in text boxes.</li> <li>5. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>6. Leave textbox empty.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Values are entered successfully.</li> <li>2. Error message displayed; ‘Enter a whole no’.</li> <li>3. Error message displayed; ‘Enter a whole no’.</li> <li>4. Error message displayed; ‘Enter a whole no’.</li> <li>5. Error message displayed; ‘Enter a whole no’.</li> <li>6. Error message displayed; ‘Required’.</li> </ol>
Result	Pass.

**Test Description:** The test verifies the entering of external interface files of low, average and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date:** 23 July, 2018

**Test Case 9:**

**Software:** Software Project Estimator

**Test ID:** 9

**Test Name:** Enter Internal Logical Files

Pre-conditions	User selects project mode. User selects estimation stage.
Action	<ol style="list-style-type: none"> <li>1. User will enter the number of internal logical files of low, average and high complexity.</li> <li>2. User will enter the number of external interface files of low, average and high complexity.</li> <li>3. Enter alphabets in text boxes.</li> <li>4. Enter special characters in text boxes.</li> <li>5. Enter decimal points in text boxes.</li> <li>6. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>7. Leave textbox empty.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Values are entered successfully.</li> <li>2. Error message displayed; ‘Enter a whole no’.</li> <li>3. Error message displayed; ‘Enter a whole no’.</li> <li>4. Error message displayed; ‘Enter a whole no’.</li> <li>5. Error message displayed; ‘Enter a whole no’.</li> <li>6. Error message displayed; ‘Required’.</li> </ol>
Result	Pass.

**Test Description:** The test verifies the entering of internal logical files of low, average

## Software Project Estimator

and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date: 23 July, 2018**

### Test Case 10:

**Software:** Software Project Estimator

**Test ID:** 10

**Test Name:** Calculate Unadjusted Function Point

Pre-conditions	User selects project mode. User selects estimation stage. User enters external inputs, external outputs, external interface files, internal logical files, and external inquiries.
Action	User clicks on calculate button.
Expected Results	Unadjusted function point should be calculated according to the following formula: $UFP = \text{Sum} ( EI + EO + EQ + EIF + ILF )$
Result	Expected results are achieved due to accurate values according to the formula.

**Test Description:** The test verifies the accurate calculation of unadjusted function points by taking inputs from the user and calculating the results.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date: 23 July, 2018**

### Test Case 11:

**Software:** Software Project Estimator

**Test ID:** 11

**Test Name:** Calculate Value Adjustment Factor

Pre-conditions	User calculates unadjusted function points.
Action	User rates 14 components on a scale of 1-5. User clicks on calculate button.
Expected Results	Value adjustment factor is calculated according to its formula.
Result	Pass.

**Test Description:** The test verifies the calculation of value adjustment factor by making the user rate fourteen factors required for VAF.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date: 23 July, 2018**

### Test Case 12:

**Software:** Software Project Estimator

**Test ID:** 12

**Test Name:** Calculate Function Points

Pre-conditions	User calculates unadjusted function points. User calculates value adjustment factor.
Action	User clicks on calculate button.
Expected Results	Function Points is calculated according to its formula.
Result	Pass.

**Test Description:** The test verifies the calculation of function points by making the user click on show results button.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date: 23 July, 2018**

### Test Case 13:

**Software:** Software Project Estimator

**Test ID:** 13

**Test Name:** Estimate Effort (Basic COCOMO I)

Pre-conditions	User calculates unadjusted function points. User calculates value adjustment factor.
Action	User clicks on show results button.
Expected Results	Effort is calculated and shown to user using the formula; $a (S)^b * EAF$ where 'a' and 'b' are based on the selected mode i.e., test case 2. S is the size in KLOC found through function points i.e., test case 12. EAF is set as 1 for this stage.
Result	Pass.

**Test Description:** The test verifies the calculation of effort by making the user click on show results button.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date: 23 July, 2018**

### Test Case 14:

**Software:** Software Project Estimator

**Test ID:** 14

**Test Name:** Rate Cost Drivers (Intermediate COCOMO I)

Pre-conditions	User calculates unadjusted function points for intermediate COCOMO I. User calculates value adjustment factor for intermediate COCOMO I.
Action	User rates 15 cost drivers ranging from very low to very high.
Expected Results	All the cost drivers are rated and effort multiplier is determined based on these ratings.
Result	Pass.

**Test Description:** The test verifies the ratings of cost drivers by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date: 23 July, 2018**

### Test Case 15:

**Software:** Software Project Estimator

**Test ID:** 15 **Test Name:** Calculate Effort Adjustment Factor (Intermediate COCOMO I)

Pre-conditions	User rates 15 cost drivers ranging from very low to very high.
Action	User clicks on calculate button.

## Software Project Estimator

Expected Results	Effort adjustment factor is calculated.
Result	Pass.

**Test Description:** The test verifies the calculation of effort adjustment factor by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date:** 23 July, 2018

### Test Case 16:

**Software:** Software Project Estimator

**Test ID:** 16

**Test Name:** Calculate Effort (Intermediate COCOMO I)

Pre-conditions	User rates 15 cost drivers ranging from very low to very high. User calculates effort adjustment factor.
Action	User clicks on show results button.
Expected Results	Effort is calculated and shown to the user.
Result	Pass.

**Test Description:** The test verifies the calculation of effort by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date:** 23 July, 2018

### Test Case 17:

**Software:** Software Project Estimator

**Test ID:** 17

**Test Name:** Calculate Development Time (Intermediate COCOMO I)

Pre-conditions	User rates 15 cost drivers ranging from very low to very high. User calculates effort adjustment factor. User calculate effort.
Action	User clicks on show results button.
Expected Results	Development time is calculated and shown to the user.
Result	Pass.

**Test Description:** The test verifies the calculation of development time by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date:** 23 July, 2018

### Test Case 18:

**Software:** Software Project Estimator

**Test ID:** 18

**Test Name:** Rate Cost Drivers According to 4 Phases (Advanced COCOMO I)

Pre-conditions	User calculates unadjusted function points for advanced COCOMO I. User calculates value adjustment factor for advanced COCOMO I.
Action	User rates 15 cost drivers ranging from very low to very high. These cost drivers are rated at four phases; requirements planning and product design (RPD), detailed design (DD), code and unit test (CUT), and integration and test (IT)

## Software Project Estimator

Expected Results	Cost drivers are rated according to four phases.
Result	Pass.

**Test Description:** The test verifies the rating of cost drivers by user according to four phases.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date:** 23 July, 2018

### Test Case 19:

**Software:** Software Project Estimator

**Test ID:** 19

**Test Name:** Calculate Effort According to 4 Phases (Advanced COCOMO I)

Pre-conditions	User rates 15 cost drivers according to four phases.
Action	User clicks on show results button.
Expected Results	Effort is calculated according to four phases.
Result	Pass.

**Test Description:** The test verifies the calculation of effort by user according to four phases.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date:** 23 July, 2018

### Test Case 20:

**Software:** Software Project Estimator

**Test ID:** 20

**Test Name:** Calculate Development Time According to 4 Phases (Advanced COCOMO I)

Pre-conditions	User rates 15 cost drivers according to four phases.
Action	User clicks on show results button.
Expected Results	Development time is calculated according to four phases.
Result	Pass.

**Test Description:** The test verifies the calculation of development time by user according to four phases.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Qadeer

**Date:** 23 July, 2018

### Test Case 21:

**Software:** Software Project Estimator

**Test ID:** 21

**Test Name:** Enter External Inputs

Pre-conditions	User selects project mode. User selects estimation stage.
----------------	--

## Software Project Estimator

Action	<ol style="list-style-type: none"> <li>1. User will enter the number (whole numbers) of external input files of low, average and high complexity.</li> <li>2. Enter alphabets in text boxes.</li> <li>3. Enter special characters in text boxes.</li> <li>4. Enter decimal points in text boxes.</li> <li>5. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>6. Leave textbox empty.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Values are entered successfully.</li> <li>2. Error message displayed; ‘Enter a whole no’.</li> <li>3. Error message displayed; ‘Enter a whole no’.</li> <li>4. Error message displayed; ‘Enter a whole no’.</li> <li>5. Error message displayed; ‘Enter a whole no’.</li> <li>6. Error message displayed; ‘Required’.</li> </ol>
Result	Pass.

**Test Description:** The test verifies the entering of external inputs of low, average and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date:** 23 July, 2018

### Test Case 22:

**Software:** Software Project Estimator

**Test ID:** 22

**Test Name:** Enter External Outputs

Pre-conditions	User selects project mode. User selects estimation stage.
Action	<ol style="list-style-type: none"> <li>1. User will enter the number (whole numbers) of external input files of low, average and high complexity.</li> <li>2. Enter alphabets in text boxes.</li> <li>3. Enter special characters in text boxes.</li> <li>4. Enter decimal points in text boxes.</li> <li>5. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>6. Leave textbox empty.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Values are entered successfully.</li> <li>2. Error message displayed; ‘Enter a whole no’.</li> <li>3. Error message displayed; ‘Enter a whole no’.</li> <li>4. Error message displayed; ‘Enter a whole no’.</li> <li>5. Error message displayed; ‘Enter a whole no’.</li> <li>6. Error message displayed; ‘Required’.</li> </ol>
Result	Pass.

**Test Description:** The test verifies the entering of external outputs of low, average and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date:** 23 July, 2018

**Test Case 23:**

**Software:** Software Project Estimator

**Test ID:** 23

**Test Name:** Enter External Inquiries

Pre-conditions	User selects project mode. User selects estimation stage.
Action	<ol style="list-style-type: none"> <li>1. User will enter the number of external inquiries of low, average and high complexity.</li> <li>2. Enter alphabets in text boxes.</li> <li>3. Enter special characters in text boxes.</li> <li>4. Enter decimal points in text boxes.</li> <li>5. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>6. Leave textbox empty.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Values are entered successfully.</li> <li>2. Error message displayed; ‘Enter a whole no’.</li> <li>3. Error message displayed; ‘Enter a whole no’.</li> <li>4. Error message displayed; ‘Enter a whole no’.</li> <li>5. Error message displayed; ‘Enter a whole no’.</li> <li>6. Error message displayed; ‘Required’.</li> </ol>
Result	Pass.

**Test Description:** The test verifies the entering of external inquiries of low, average and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date:** 23 July, 2018

**Test Case 24:**

**Software:** Software Project Estimator

**Test ID:** 24

**Test Name:** Enter External Interface Files

Pre-conditions	User selects project mode. User selects estimation stage.
Action	<ol style="list-style-type: none"> <li>1. User will enter the number of external interface files of low, average and high complexity.</li> <li>2. Enter alphabets in text boxes.</li> <li>3. Enter special characters in text boxes.</li> <li>4. Enter decimal points in text boxes.</li> <li>5. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>6. Leave textbox empty.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Values are entered successfully.</li> <li>2. Error message displayed; ‘Enter a whole no’.</li> <li>3. Error message displayed; ‘Enter a whole no’.</li> <li>4. Error message displayed; ‘Enter a whole no’.</li> <li>5. Error message displayed; ‘Enter a whole no’.</li> <li>6. Error message displayed; ‘Required’.</li> </ol>
Result	Pass.

**Test Description:** The test verifies the entering of external interface files of low, average and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date: 23 July, 2018**

**Test Case 25:**

**Software:** Software Project Estimator

**Test ID:** 25

**Test Name:** Enter Internal Logical Files

Pre-conditions	User selects project mode. User selects estimation stage.
Action	<ol style="list-style-type: none"> <li>1. User will enter the number of internal logical files of low, average and high complexity.</li> <li>2. User will enter the number of external interface files of low, average and high complexity.</li> <li>3. Enter alphabets in text boxes.</li> <li>4. Enter special characters in text boxes.</li> <li>5. Enter decimal points in text boxes.</li> <li>6. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>7. Leave textbox empty.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Values are entered successfully.</li> <li>2. Error message displayed; ‘Enter a whole no’.</li> <li>3. Error message displayed; ‘Enter a whole no’.</li> <li>4. Error message displayed; ‘Enter a whole no’.</li> <li>5. Error message displayed; ‘Enter a whole no’.</li> <li>6. Error message displayed; ‘Required’.</li> </ol>
Result	Pass.

**Test Description:** The test verifies the entering of internal logical files of low, average and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date: 23 July, 2018**

**Test Case 26:**

**Software:** Software Project Estimator

**Test ID:** 26

**Test Name:** Calculate Unadjusted Function Point

Pre-conditions	User selects project mode. User selects estimation stage. User enters external inputs, external outputs, external interface files, internal logical files, and external inquiries.
Action	‘User clicks on calculate button.
Expected Results	Unadjusted function point should be calculated according to the following formula: $UFP = \text{Sum} ( EI + EO + EQ + EIF + ILF )$
Result	Expected results are achieved due to accurate values according to the formula.

**Test Description:** The test verifies the accurate calculation of unadjusted function points by taking inputs from the user and calculating the results.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date: 23 July, 2018**

**Test Case 27:**

**Software:** Software Project Estimator

**Test ID:** 27

**Test Name:** Calculate Value Adjustment Factor

Pre-conditions	User calculates unadjusted function points.
Action	User rates 14 components on a scale of 1-5. User clicks on calculate button.
Expected Results	Value adjustment factor is calculated according to its formula.
Result	Pass.

**Test Description:** The test verifies the calculation of value adjustment factor by making the user rate fourteen factors required for VAF.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date: 23 July, 2018**

**Test Case 28:**

**Software:** Software Project Estimator

**Test ID:** 28

**Test Name:** Calculate Function Points

Pre-conditions	User calculates unadjusted function points. User calculates value adjustment factor.
Action	User clicks on show results button.
Expected Results	Value adjustment factor is calculated according to its formula.
Result	Pass.

**Test Description:** The test verifies the calculation of function points by making the user click on show results button.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date: 23 July, 2018**

**Test Case 29:**

**Software:** Software Project Estimator

**Test ID:** 29

**Test Name:** Total Effort (B)

Pre-conditions	User should estimate the size of the project, estimate time which is given by client to complete the project, and know the value of process productivity.
Action	User enters the estimated time to complete the project and press the calculate button.
Expected Results	Show the value of Total Effort (B).
Result	Pass.

**Test Description:** The test verifies that the user calculates the values of size, time and the process productivity and the system will show the value of Total Effort required for the project to the user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date: 23 July, 2018**

**Test Case 30:**

**Software:** Software Project Estimator

**Test ID:** 30

**Test Name:** Calculate Productivity Index (PI)

Pre-conditions	User knows the application type complexity, the tools to complete the projects, the computer availability, and the experience of the development team.
Action	User selects the application type from the drop-down menu and rate the tools, computer availability and the experience of the development team from the drop-down menu.
Expected Results	Show the value of Productivity Index (PI).
Result	Pass.

**Test Description:** The test verifies that the user select the application type and rate the productivity index from the drop down menu.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date: 23 July, 2018**

**Test Case 31:**

**Software:** Software Project Estimator

**Test ID:** 31

**Test Name:** Estimate product productivity using product index

Pre-conditions	User should know the application type and the environment of the project to select the productivity index (PI) high or low values.
Action	User selects the application type, productivity index from the drop-down menu and press the calculate button.
Expected Results	Show the value of product productivity.
Result	Pass.

**Test Description:** The test verifies that the user selects the application type and productivity index from the drop-down menu and the system will show the value of product productivity to the user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date: 23 July, 2018**

**Test Case 32:**

**Software:** Software Project Estimator

**Test ID:** 32

**Test Name:** Calculate E (effort at delivery time)

Pre-conditions	User knows the total effort required to complete the project.
Action	User press the calculate button.
Expected Results	Show the value of Effort at Delivery Time (E).
Result	Pass.

**Test Description:** The test verifies that the user calculates the values of total effort (B) and the system will show the value of Effort at Delivery Time required for the project to the user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date: 23 July, 2018**

### Test Case 33:

**Software:** Software Project Estimator

**Test ID:** 33

**Test Name:** Calculate Manpower Acceleration

Pre-conditions	User know the total effort and estimated time to complete the project.
Action	User press the calculate button.
Expected Results	Show the value of Manpower Acceleration.
Result	Pass.

**Test Description:** The test verifies that the user calculates the values of Total Effort and the Estimated Time and the system will show the value of Manpower Acceleration.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date: 23 July, 2018**

### Test Case 35:

**Software:** Software Project Estimator

**Test ID:** 35

**Test Name:** Select Project Mode

Pre-conditions	User selects incremental from the dropdown menu. User is directed to COCOMO II.
Action	User select the project mode to move further for the selection of COCOMO II models.
Expected Results	Project mode is selected and user is taken to the next step.
Result	Expected results achieved.

**Test Description:** The test verifies that the user selects the project mode and is taken to the next step.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date: 23 July, 2018**

### Test Case 36:

**Software:** Software Project Estimator

**Test ID:** 36

**Test Name:** Select Project Model

Pre-conditions	User selects the Start Estimation button. User is directed to COCOMO II model selection page.
Action	User select the project model from the three given i.e. Application Composition Model, Early Design Model, and Post Architecture Model.
Expected Results	Project model is selected and user is taken to the next step.
Result	Expected results achieved.

## Software Project Estimator

**Test Description:** The test verifies that the user selects the project mode from the three options and is taken to the next step.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date:** 23 July, 2018

### Test Case 37:

**Software:** Software Project Estimator

**Test ID:** 37

**Test Name:** Calculate New Object Point

Pre-conditions	User selects Application Composition Model from the three given models.
Action	<ol style="list-style-type: none"><li>1. User estimate the no of screens.</li><li>2. User estimate the no of reports.</li><li>3. Estimated screens and reports should be equal to the total number of reports.</li><li>4. If any component is acquired mention that.</li><li>5. User click on calculate object point and sum them all.</li><li>6. If any part of system is reuse, assign it a percentage.</li><li>7. User click on calculate new object point.</li></ol>
Expected Results	New object point is calculated according to its weights and formulas.
Result	Expected results achieved.

**Test Description:** The test verifies the calculated of new object point by taking no of screen, reports, component acquired and reuse percentage from user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date:** 23 July, 2018

### Test Case 38:

**Software:** Software Project Estimator

**Test ID:** 38

**Test Name:** Calculate Productivity Rate

Pre-conditions	User calculate new object point.
Action	<ol style="list-style-type: none"><li>1. User estimate the developer skill/experience by assigning different weights according to developer skills.</li><li>2. User select one weight from dropdown list.</li><li>3. User click on calculate productivity rate.</li></ol>
Expected Results	Productivity rate is calculated.
Result	Expected results achieved.

**Test Description:** The test verifies the calculation of productivity rate.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date:** 23 July, 2018

**Test Case 39:**

**Software:** Software Project Estimator

**Test ID:** 39

**Test Name:** Calculate Effort (Application Composition Model)

Pre-conditions	<ol style="list-style-type: none"> <li>1. User calculate new object point.</li> <li>2. User calculate productivity rate.</li> </ol>
Action	User click the calculate button which will get new object point and productivity rate value, which is use in formula along with other constants to calculate effort.
Expected Results	New object points and productivity rate is displayed, and effort is calculated.
Result	Expected results achieved.

**Test Description:** The test verifies the calculation of effort through Application Composition Model.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date:** 23 July, 2018

**Test Case 40:**

**Software:** Software Project Estimator

**Test ID:** 40

**Test Name:** Calculate Language Factor

Pre-conditions	User needs to calculate KLOC.
Action	<ol style="list-style-type: none"> <li>1. User select the language from dropdown list.</li> <li>2. User click calculate language factor button.</li> </ol>
Expected Results	Language factor is shown for the selected language
Result	Expected results achieved.

**Test Description:** The test verifies the calculation of Language Factor.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date:** 23 July, 2018

**Test Case 41:**

**Software:** Software Project Estimator

**Test ID:** 41

**Test Name:** Enter External Inputs

Pre-conditions	User needs to calculate KLOC.
Action	<ol style="list-style-type: none"> <li>1. User will enter the number (whole numbers) of external input files of low, average and high complexity.</li> <li>2. Enter alphabets in text boxes.</li> <li>3. Enter special characters in text boxes.</li> <li>4. Enter decimal points in text boxes.</li> <li>5. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>6. Leave textbox empty.</li> </ol>

## Software Project Estimator

Expected Results	<ul style="list-style-type: none"> <li>7. Values are entered successfully.</li> <li>8. Error message displayed; ‘Enter a whole no’.</li> <li>9. Error message displayed; ‘Enter a whole no’.</li> <li>10. Error message displayed; ‘Enter a whole no’.</li> <li>11. Error message displayed; ‘Enter a whole no’.</li> <li>12. Error message displayed; ‘Required’.</li> </ul>
Result	Expected results achieved.

**Test Description:** The test verifies the entering of external inputs of low, average and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Raiz

**Date: 23 July, 2018**

### Test Case 42:

**Software:** Software Project Estimator

**Test ID:** 42

**Test Name:** Enter External Outputs

Pre-conditions	User needs to calculate KLOC.
Action	<ul style="list-style-type: none"> <li>1. User will enter the number (whole numbers) of external input files of low, average and high complexity.</li> <li>2. Enter alphabets in text boxes.</li> <li>3. Enter special characters in text boxes.</li> <li>4. Enter decimal points in text boxes.</li> <li>5. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>6. Leave textbox empty.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>7. Values are entered successfully.</li> <li>8. Error message displayed; ‘Enter a whole no’.</li> <li>9. Error message displayed; ‘Enter a whole no’.</li> <li>10. Error message displayed; ‘Enter a whole no’.</li> <li>11. Error message displayed; ‘Enter a whole no’.</li> <li>12. Error message displayed; ‘Required’.</li> </ul>
Result	Expected results achieved.

**Test Description:** The test verifies the entering of external outputs of low, average and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Raiz

**Date: 23 July, 2018**

### Test Case 43:

**Software:** Software Project Estimator

**Test ID:** 43

**Test Name:** Enter External Inquiries

Pre-conditions	User needs to calculate KLOC.
----------------	-------------------------------

## Software Project Estimator

Action	<ol style="list-style-type: none"> <li>1. User will enter the number of external inquiries of low, average and high complexity.</li> <li>2. Enter alphabets in text boxes.</li> <li>3. Enter special characters in text boxes.</li> <li>4. Enter decimal points in text boxes.</li> <li>5. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>6. Leave textbox empty.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Values are entered successfully.</li> <li>2. Error message displayed; ‘Enter a whole no’.</li> <li>3. Error message displayed; ‘Enter a whole no’.</li> <li>4. Error message displayed; ‘Enter a whole no’.</li> <li>5. Error message displayed; ‘Enter a whole no’.</li> <li>6. Error message displayed; ‘Required’.</li> </ol>
Result	Expected results achieved.

**Test Description:** The test verifies the entering of external inquiries of low, average and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Raiz

**Date:** 23 July, 2018

### Test Case 44:

**Software:** Software Project Estimator

**Test ID:** 44

**Test Name:** Enter External Interface Files

Pre-conditions	User needs to calculate KLOC.
Action	<ol style="list-style-type: none"> <li>1. User will enter the number of external interface files of low, average and high complexity.</li> <li>2. Enter alphabets in text boxes.</li> <li>3. Enter special characters in text boxes.</li> <li>4. Enter decimal points in text boxes.</li> <li>5. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>6. Leave textbox empty.</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Values are entered successfully.</li> <li>2. Error message displayed; ‘Enter a whole no’.</li> <li>3. Error message displayed; ‘Enter a whole no’.</li> <li>4. Error message displayed; ‘Enter a whole no’.</li> <li>5. Error message displayed; ‘Enter a whole no’.</li> <li>6. Error message displayed; ‘Required’.</li> </ol>
Result	Expected results achieved.

**Test Description:** The test verifies the entering of external interface files of low, average and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date:** 23 July, 2018

**Test Case 45:**

**Software:** Software Project Estimator

**Test ID:** 45

**Test Name:** Enter Internal Logical Files

Pre-conditions	User needs to calculate KLOC.
Action	<ul style="list-style-type: none"> <li>8. User will enter the number of internal logical files of low, average and high complexity.</li> <li>9. User will enter the number of external interface files of low, average and high complexity.</li> <li>10. Enter alphabets in text boxes.</li> <li>11. Enter special characters in text boxes.</li> <li>12. Enter decimal points in text boxes.</li> <li>13. Enter combination of whole numbers with alphabets, special characters, or decimal points.</li> <li>Leave textbox empty.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>1. Values are entered successfully.</li> <li>2. Error message displayed; ‘Enter a whole no’.</li> <li>3. Error message displayed; ‘Enter a whole no’.</li> <li>4. Error message displayed; ‘Enter a whole no’.</li> <li>5. Error message displayed; ‘Enter a whole no’.</li> <li>6. Error message displayed; ‘Required’.</li> </ul>
Result	Expected results achieved.

**Test Description:** The test verifies the entering of internal logical files of low, average and high complexity by user.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date:** 23 July, 2018

**Test Case 46:**

**Software:** Software Project Estimator

**Test ID:** 46

**Test Name:** Calculate Unadjusted Function Point

Pre-conditions	User enters external inputs, external outputs, external interface files, internal logical files, and external inquiries.
Action	<ul style="list-style-type: none"> <li>1. User clicks on calculate button.</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>○ All the entered value will be multiplied with constants.</li> <li>○ Unadjusted function point should be calculated according to the following formula:  <math display="block">UFP = \text{Sum} ( EI + EO + EQ + EIF + ILF )</math> </li> </ul>
Result	Expected results are achieved due to accurate values according to the formula.

**Test Description:** The test verifies the accurate calculation of unadjusted function points by taking inputs from the user and calculating the results.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date:** 23 July, 2018

**Test Case 47:**

**Software:** Software Project Estimator

**Test ID:** 47

**Test Name:** Calculate Value Adjustment Factor

Pre-conditions	User calculates unadjusted function points.
Action	<ul style="list-style-type: none"> <li>○ User rates 14 components on a scale of 1-5.</li> <li>○ User clicks on calculate button.</li> </ul>
Expected Results	Value adjustment factor is calculated according to its formula.
Result	Expected results achieved.

**Test Description:** The test verifies the calculation of value adjustment factor by making the user rate fourteen factors required for VAF.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date:** 23 July, 2018

**Test Case 48:**

**Software:** Software Project Estimator

**Test ID:** 48

**Test Name:** Calculate Function Points

Pre-conditions	<ul style="list-style-type: none"> <li>○ User calculates unadjusted function points.</li> <li>○ User calculates value adjustment factor.</li> </ul>
Action	User clicks on calculate button.
Expected Results	<ul style="list-style-type: none"> <li>○ Unadjusted function point and value adjustment factor value will be get and will be displayed in textboxes.</li> <li>○ Function Point is calculated using unadjusted function point and value adjustment factor.</li> </ul>
Result	Expected results achieved.

**Test Description:** The test verifies the calculation of function points by making the user click on show results button.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date:** 23 July, 2018

**Test Case 49:**

**Software:** Software Project Estimator

**Test ID:** 49

**Test Name:** Calculate KLOC

Pre-conditions	<ul style="list-style-type: none"> <li>○ User calculates language factor.</li> <li>○ User calculates function point.</li> </ul>
Action	User clicks on calculate button.
Expected Results	<ul style="list-style-type: none"> <li>○ Language factor and function point value will be get and will be displayed in textboxes.</li> <li>○ KLOC is calculated using language factor and function point along with other operations and formulas.</li> </ul>
Result	Expected results achieved.

**Test Description:** The test verifies the calculation of KLOC by making the user click on show results button.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

Date: 23 July, 2018

**Test Case 50:****Software:** Software Project Estimator**Test ID:** 5**Test Name:** Enter Product Factor (15 Cost Drivers)

Pre-conditions	User needs to calculate EAF.
Action	User will select weights in terms of low, very low, nominal, high, very high and extra high from dropdown list for cost drivers listed below: <ul style="list-style-type: none"> <li><input type="radio"/> Reliability (RELY)</li> <li><input type="radio"/> Database Size (DATA)</li> <li><input type="radio"/> Complexity (CPLX)</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li><input type="radio"/> Weights will be selected from dropdown list.</li> <li><input type="radio"/> Product factor values are entered.</li> </ul>
Result	Expected results achieved.

**Test Description:** The test verifies the entering of product factor using cost drivers weights in terms of low, very low, nominal, high, very high, and extra high by users.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server**Tested by:** Arslan Ahmed

Date: 23 July, 2018

**Test Case 51:****Software:** Software Project Estimator**Test ID:** 51**Test Name:** Enter Platform Factor (15 Cost Drivers)

Pre-conditions	User needs to calculate EAF.
Action	User will select weights in terms of low, very low, nominal, high, very high and extra high from dropdown list for factors listed below: <ul style="list-style-type: none"> <li><input type="radio"/> Execution Time Constraint (TIME)</li> <li><input type="radio"/> Main Storage Constraint (STOR)</li> <li><input type="radio"/> Virtual Machine Volatility (VIRT)</li> <li><input type="radio"/> Computer Turn Around Time (TURN)</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li><input type="radio"/> Weights will be selected from dropdown list.</li> <li><input type="radio"/> Platform factor values are entered.</li> </ul>
Result	Expected results achieved.

**Test Description:** The test verifies the entering of platform factor using cost drivers weights in terms of low, very low, nominal, high, very high, and extra high by users.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server**Tested by:** Arslan Ahmed

Date: 23 July, 2018

**Test Case 52:**

**Software:** Software Project Estimator

**Test ID:** 52

**Test Name:** Enter Personal Factor (15 Cost Drivers)

Pre-conditions	User needs to calculate EAF.
Action	User will select weights in terms of low, very low, nominal, high, very high and extra high from dropdown list for factors listed below: <ul style="list-style-type: none"> <li><input type="radio"/> Analysts Capability (ACAP)</li> <li><input type="radio"/> Application Experience (AEXP)</li> <li><input type="radio"/> Programmer Capability (PCAP)</li> <li><input type="radio"/> Virtual Machine Experience (VEXP)</li> <li><input type="radio"/> Language Experience (LEXP)</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li><input type="radio"/> Weights will be selected from dropdown list.</li> <li><input type="radio"/> Personal factor values are entered.</li> </ul>
Result	Expected results achieved.

**Test Description:** The test verifies the entering of personal factor using cost drivers weights in terms of low, very low, nominal, high, very high, and extra high by users.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date:** 23 July, 2018

**Test Case 53:**

**Software:** Software Project Estimator

**Test ID:** 53

**Test Name:** Enter Project Factor (15 Cost Drivers)

Pre-conditions	User needs to calculate EAF.
Action	User will select weights in terms of low, very low, nominal, high, very high and extra high from dropdown list for factors listed below: <ul style="list-style-type: none"> <li><input type="radio"/> Modern Programming Practices (MODP)</li> <li><input type="radio"/> Software Tool (TOOL)</li> <li><input type="radio"/> Development Schedule (SCED)</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li><input type="radio"/> Weights will be selected from dropdown list.</li> <li><input type="radio"/> Project factor values are entered.</li> </ul>
Result	Expected results achieved.

**Test Description:** The test verifies the entering of project factor using cost drivers weights in terms of low, very low, nominal, high, very high, and extra high by users.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date:** 23 July, 2018

**Test Case 54:**

**Software:** Software Project Estimator

**Test ID:** 54

**Test Name:** Calculate EAF (Using 15 Cost Drivers)

Pre-conditions	<ul style="list-style-type: none"> <li><input type="radio"/> User enter product factor.</li> <li><input type="radio"/> User enter platform factor.</li> <li><input type="radio"/> User enter personal factor.</li> <li><input type="radio"/> User enter project factor.</li> </ul>
Action	User clicks on calculate button.
Expected Results	EAF will be calculated.
Result	Expected results achieved.

**Test Description:** The test verifies the calculation of EAF by making the user click on show calculate button.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date:** 23 July, 2018

**Test Case 55:**

**Software:** Software Project Estimator

**Test ID:** 55

**Test Name:** Calculate Effort (Early Design Model)

Pre-conditions	<ul style="list-style-type: none"> <li><input type="radio"/> User calculate KLOC.</li> <li><input type="radio"/> User calculate EAF.</li> </ul>
Action	User click the calculate button which will get KLOC and EAF value, which is use in formula along with other constants to calculate effort.
Expected Results	KLOC and EAF is displayed, and effort is calculated.
Result	Expected results achieved.

**Test Description:** The test verifies the calculation of effort through Early Design Model.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date:** 23 July, 2018

**Test Case 56:**

**Software:** Software Project Estimator

**Test ID:** 56

**Test Name:** Calculate Scale Component (b)

Pre-conditions	User calculated KLOC.
----------------	-----------------------

## Software Project Estimator

Action	<p>User will select weights in terms of low, very low, nominal, high, very high and extra high from dropdown list for scale factor listed below:</p> <ul style="list-style-type: none"> <li><input type="radio"/> PREC</li> <li><input type="radio"/> FLEX</li> <li><input type="radio"/> RESL</li> <li><input type="radio"/> TEAM</li> <li><input type="radio"/> PMAT</li> </ul> <p>User will click on calculate SF button and scale factor value will be displayed by taking sum of scale factor.</p> <p>User will then click on calculate b which will calculate scale component value using constant and formula.</p>
Expected Results	Scale component (b) will be calculated.
Result	Expected results achieved.

**Test Description:** The test verifies the calculation of scale component (b) by making the user selecting scale factors.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date: 23 July, 2018**

### Test Case 57:

**Software:** Software Project Estimator

**Test ID:** 57

**Test Name:** Enter Product Factor (17 Cost Drivers)

Pre-conditions	User needs to calculate EAF.
Action	<p>User will select weights in terms of low, very low, nominal, high, very high and extra high from dropdown list for cost drivers listed below:</p> <ul style="list-style-type: none"> <li><input type="radio"/> Reliability (RELY)</li> <li><input type="radio"/> Database Size (DATA)</li> <li><input type="radio"/> Complexity (CPLX)</li> <li><input type="radio"/> Requirement Reusability (RUSE)</li> <li><input type="radio"/> Documentation (DOCU)</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li><input type="radio"/> Weights will be selected from dropdown list.</li> <li><input type="radio"/> Product factor values are entered.</li> </ul>
Result	Expected results achieved.

**Test Description:** The test verifies the entering of product factor using cost driver's weights in terms of low, very low, nominal, high, very high, and extra high by users.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date: 23 July, 2018**

**Test Case 58:**

**Software:** Software Project Estimator

**Test ID:** 58

**Test Name:** Enter Platform Factor (17 Cost Drivers)

Pre-conditions	User needs to calculate EAF.
Action	User will select weights in terms of low, very low, nominal, high, very high and extra high from dropdown list for factors listed below: <ul style="list-style-type: none"> <li><input type="radio"/> Execution Time Constraint (TIME)</li> <li><input type="radio"/> Main Storage Constraint (STOR)</li> <li><input type="radio"/> Virtual Machine Volatility (VIRT)</li> <li><input type="radio"/> Computer Turn Around Time (TURN)</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li><input type="radio"/> Weights will be selected from dropdown list.</li> <li><input type="radio"/> Platform factor values are entered.</li> </ul>
Result	Expected results achieved.

**Test Description:** The test verifies the entering of platform factor using cost drivers weights in terms of low, very low, nominal, high, very high, and extra high by users.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date:** 23 July, 2018

**Test Case 59:**

**Software:** Software Project Estimator

**Test ID:** 59

**Test Name:** Enter Personal Factor (17 Cost Drivers)

Pre-conditions	User needs to calculate EAF.
Action	User will select weights in terms of low, very low, nominal, high, very high and extra high from dropdown list for factors listed below: <ul style="list-style-type: none"> <li><input type="radio"/> Analysts Capability (ACAP)</li> <li><input type="radio"/> Application Experience (AEXP)</li> <li><input type="radio"/> Programmer Capability (PCAP)</li> <li><input type="radio"/> Virtual Machine Experience (VEXP)</li> <li><input type="radio"/> Language Experience (LEXP)</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li><input type="radio"/> Weights will be selected from dropdown list.</li> <li><input type="radio"/> Personal factor values are entered.</li> </ul>
Result	Expected results achieved.

**Test Description:** The test verifies the entering of personal factor using cost drivers weights in terms of low, very low, nominal, high, very high, and extra high by users.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date:** 23 July, 2018

**Test Case 60:**

**Software:** Software Project Estimator

**Test ID:** 60

**Test Name:** Enter Project Factor (17 Cost Drivers)

Pre-conditions	User needs to calculate EAF.
Action	User will select weights in terms of low, very low, nominal, high, very high and extra high from dropdown list for factors listed below: <ul style="list-style-type: none"> <li>○ Modern Programming Practices (MODP)</li> <li>○ Software Tool (TOOL)</li> <li>○ Development Schedule (SCED)</li> </ul>
Expected Results	<ul style="list-style-type: none"> <li>○ Weights will be selected from dropdown list.</li> <li>○ Project factor values are entered.</li> </ul>
Result	Expected results achieved.

**Test Description:** The test verifies the entering of project factor using cost driver's weights in terms of low, very low, nominal, high, very high, and extra high by users.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date:** 23 July, 2018

**Test Case 61:**

**Software:** Software Project Estimator

**Test ID:** 61

**Test Name:** Calculate EAF (Using 17 Cost Drivers)

Pre-conditions	<ul style="list-style-type: none"> <li>○ User enter product factor.</li> <li>○ User enter platform factor.</li> <li>○ User enter personal factor.</li> <li>○ User enter project factor.</li> </ul>
Action	User clicks on calculate button.
Expected Results	<ul style="list-style-type: none"> <li>○ will be calculated.</li> </ul>
Result	Expected results achieved.

**Test Description:** The test verifies the calculation of EAF by making the user click on show calculate button.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Arslan Ahmed

**Date:** 23 July, 2018

**Test Case 62:**

**Software:** Software Project Estimator

**Test ID:** 62

**Test Name:** Calculate Effort (Post Architecture Model)

Pre-conditions	<ul style="list-style-type: none"> <li>○ User calculate KLOC.</li> <li>○ User calculate b.</li> <li>○ User calculate EAF.</li> </ul>
----------------	--

## Software Project Estimator

Action	User click the calculate button which will get KLOC, b and EAF value, which is used in formula along with other constants to calculate effort.
Expected Results	KLOC, b and EAF is displayed, and effort is calculated.
Result	Expected results achieved.

**Test Description:** The test verifies the calculation of effort through Post Architecture Model.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date:** 23 July, 2018

### Test Case 63:

**Software:** Software Project Estimator

**Test ID:** 63

**Test Name:** Login

Pre-conditions	<ul style="list-style-type: none"><li><input type="radio"/> User enter name.</li><li><input type="radio"/> User enter password.</li></ul>
Action	User clicks on login button.
Expected Results	<ul style="list-style-type: none"><li><input type="radio"/> User will be logged in.</li></ul>
Result	Pass

**Test Description:** The test verifies the login.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date:** 23 July, 2018

### Test Case 64:

**Software:** Software Project Estimator

**Test ID:** 64

**Test Name:** Signup

Pre-conditions	<ul style="list-style-type: none"><li><input type="radio"/> User enter name.</li><li><input type="radio"/> User enter password.</li><li><input type="radio"/> User enter email.</li><li><input type="radio"/> User enter organization name.</li><li><input type="radio"/> User enter position.</li></ul>
Action	User clicks on signup button.
Expected Results	<ul style="list-style-type: none"><li><input type="radio"/> User will be registered.</li></ul>
Result	Pass

**Test Description:** The test verifies the signup.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date:** 23 July, 2018

### Test Case 65:

**Software:** Software Project Estimator

**Test ID:** 65

**Test Name:** Generate report

Pre-conditions	<ul style="list-style-type: none"><li><input type="radio"/> User performs all estimation steps.</li></ul>
Action	User clicks on generate report button.

## Software Project Estimator

Expected Results	<input type="radio"/> User will be registered.
Result	Pass

**Test Description:** The test verifies that report is generated at the end of estimation.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date:** 23 July, 2018

### Test Case 66:

**Software:** Software Project Estimator

**Test ID:** 65

**Test Name:** Save report

Pre-conditions	<input type="radio"/> Report is generated
Action	User clicks on generate report button.
Expected Results	<input type="radio"/> User will be registered.
Result	Pass

**Test Description:** The test verifies that user can save report.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

**Date:** 23 July, 2018

### Test Case 67:

**Software:** Software Project Estimator

**Test ID:** 63

**Test Name:** Calculate Cost

Pre-conditions	<input type="radio"/> User enter average personal salary. <input type="radio"/> Effort is estimated. <input type="radio"/> Time is estimated.
Action	User clicks on calculate cost button.
Expected Results	<input type="radio"/> User will be logged in.
Result	Pass

**Test Description:** The test verifies the login.

**Testing Environment:** Asp.net MVC, Visual Studio, SQL Server

**Tested by:** Ahmed Riaz

## 5.4. Test procedure specification

### 5.4.1. Purpose

To specify the steps for executing a set of test cases or, more generally, the steps used to analyze a software item in order to evaluate a set of features.

### 5.4.2 Outline

A test procedure specification shall have the following structure:

- a. Test procedure specification identifier
- b. Purpose
- c. Special requirements
- d. Procedure steps

#### **5.4.2.1. Test procedure specification identifier**

Identifier for this module is **SETOOLTPS-001**.

#### **5.4.2.2. Purpose**

This module describes the testing procedures and the whole testing process. By looking at this module, a person will learn how to execute the testing procedures of a management system. This procedure executes following test cases 1, 02, 06..., 62.

#### **5.4.2.3. Special requirements**

Front-end is being tested manually and Back-end testing would be on a tool (automated). We use testing in different stages like pre-testing, repair and regression testing, future compliance testing, end-to-end testing, re-testing after certification, etc.

#### **5.4.2.4. Procedure steps**

##### **5.4.2.4.1. Log**

Describe any special methods or formats for logging the results of test execution, the incidents observed, and any other events pertinent to the test (see Clauses 9 and 10).

##### **5.4.2.4.2. Set up**

1. Compile the source.
2. Start testing

##### **5.4.2.4.3. Start**

We need a single browser supported machine.

##### **5.4.2.4.4. Proceed**

Run test cases in dependency order: Test cases with no dependencies or solved dependencies first.

##### **5.4.2.4.5. Measure**

Measurement system is based on human observation mostly, so log what we see on the screen, if something goes wrong.

##### **5.4.2.4.6. Shut down**

Close browser/tab

##### **5.4.2.4.7. Restart**

Refresh page.

##### **5.4.2.4.8. Stop**

Close browser/tab

##### **5.4.2.4.9. Wrap up**

Report hours and check that test log is written.

##### **5.4.2.4.10. Contingencies**

---

We need to submit bug reports. Test manager will use that information to create incident reports.

## **5.5. Test item transmittal report**

### **5.5.1. Purpose**

To identify the test items being transmitted for testing. It includes the person responsible for each item, its physical location, and its status. Any variations from the current item requirements and designs are noted in this report.

### **5.5.2. Outline**

A test item transmittal report shall have the following structure:

- a. Transmittal report identifier
- b. Transmitted items
- c. Location
- d. Status
- e. Approvals

#### **5.5.2.1. Transmittal report identifier**

Identifier for this document is **SPETOOLTR-0001**.

#### **5.5.2.2. Transmitted items**

Briefly describe the item to be transmitted here. Include version/revision level. Add references to documentation etc. Identify people responsible for each item.

#### **5.5.2.3. Location**

Identify the location of transmitted items and media to use in transmission.

#### **5.5.2.4. Status**

Status of items being transmitted.

#### **5.5.2.5. Approvals**

**Name:** Ms. Mahum Adil

**Title:** Project Supervisor

**Signature:** \_\_\_\_\_

**Date:** \_\_\_\_\_

## **5.6. Test log**

### **5.6.1. Purpose**

Our purpose is to provide a chronological record of relevant details about the execution of tests.

### **5.6.2. Outline**

A test log shall have the following structure:

- a. Test log identifier;
- b. Description;
- c. Activity and event entries.

#### **5.6.2.1. Test log identifier**

Identifier for this document is **SETOOLTL-00001**

#### **5.6.2.2. Description**

Information that applies to all entries in the log except as specifically noted in a log entry should be included here. The following information should be considered:

- o Identify the items being tested including their version/revision levels. For each of these items, supply a reference to its transmittal report, if it exists.
- o Identify the attributes of the environments in which the testing is conducted. Include facility identification, hardware being used (e.g., amount of memory being used, CPU model number, and number and model of tape drives, and/or mass storage devices), system software used, and resources available (e.g., the amount of memory available).

#### **5.6.2.3. Activity and event entries**

For each event, including the beginning and end of activities, record the occurrence date and time along with the identity of the author. The information in 5.2.3.1 through 5.2.3.5 should be considered:

##### **5.6.2.3.1. Execution description**

Record the identifier of the test procedure being executed and supply a reference to its specification. Record all personnel present during the execution including testers, operators, and observers. Also indicate the function of each individual.

##### **5.6.2.3.2. Procedure results**

For each execution, record the visually observable results (e.g., error messages generated, aborts, and requests for operator action). Also record the location of any output (e.g., reel number). Record the successful or unsuccessful execution of the test.

##### **5.6.2.3.3. Environmental information**

Record any environmental conditions specific to this entry (e.g., hardware substitutions).

##### **5.6.2.3.4. Anomalous events**

Record what happened before and after an unexpected event occurred (e.g., A summary display was requested and the correct screen displayed, but response seemed unusually long. A repetition produced the same prolonged response). Record circumstances surrounding the inability to begin execution of a test procedure or failure to complete a

test procedure (e.g., a power failure or system software problem).

#### **5.6.2.3.5. *Incident report identifiers***

Record the identifier of each test incident report, whenever one is generated.

### **5.7. *Test incident report***

#### **5.7.1. Purpose**

The purpose of the test log is to document the events and results encountered during the test. To provide a sequential record of relevant details about the execution of tests. This log represents the results of testing of whole system.

#### **5.7.2. Outline**

A test incident report shall have the following structure:

- a. Test incident report identifier
- b. Summary
- c. Incident description
- d. Impact

##### **5.7.2.1. Test incident report identifier**

Identifier for Test incident report is **SETOOLTIR-000001**.

##### **5.7.2.2. Summary**

All the items are tested for ‘Software estimation Tool’. The testing of this system is conducted manually. When testing the SLIM, there was an anomalous event occurred. But later that was fixed.

##### **5.7.2.3. Incident description**

Provide a description of the incident. This description should include the following items:

- a. Inputs
- b. Expected results
- c. Actual results
- d. Anomalies
- e. Date and time;
- f. Procedure step;
- g. Environment;
- h. Attempts to repeat;
- i. Testers;
- j. Observers.

Related activities and observations that may help to isolate and correct the cause of the incident should be included (e.g., describe any test case executions that might have a bearing on this particular incident and any variations from the published test procedure).

##### **5.7.2.4. Impact**

If known, indicate what impact this incident will have on test plans, test design specifications, test procedure specifications, or test case specifications.

## **5.8. Test summary report**

### **5.8.1. Purpose**

To summarize the results of the designated testing activities and to provide evaluations based on these results.

### **5.8.2. Outline**

A test summary report shall have the following structure:

- a. Test summary report identifier
- b. Summary
- c. Variances
- d. Comprehensive assessment
- e. Summary of results
- f. Evaluation
- g. Summary of activities
- h. Approvals

The sections shall be ordered in the specified sequence. Additional sections may be included just prior to Approvals. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content. The referenced material must be attached to the test summary report or available to users of the summary report.

Details on the content of each section are contained in the following sub clauses.

#### **5.8.2.1. Test summary report identifier**

Identifier for Test incident report is **SPETOOLTSR-0000001**.

#### **5.8.2.2. Summary**

Summarize the evaluation of the test items. Identify the items tested, indicating their version/revision level. Indicate the environment in which the testing activities took place. For each test item, supply references to the following documents if they exist: test plan, test design specifications, test procedure specifications, test item transmittal reports, test logs, and test incident reports.

#### **5.8.2.3. Variances**

Report any variances of the test items from their design specifications. Indicate any variances from the test plan, test designs, or test procedures. Specify the reason for each variance.

#### **5.8.2.4. Comprehensiveness assessment**

Evaluate the comprehensiveness of the testing process against the comprehensiveness criteria specified in the test plan if the plan exists. Identify features or feature combinations that were not sufficiently tested and explain the reasons.

#### **5.8.2.5. Summary of results**

Summarize the results of testing. Identify all resolved incidents and summarize their resolutions. Identify all unresolved incidents.

#### **5.8.2.6. Evaluation**

Provide an overall evaluation of each test item including its limitations. This evaluation

shall be based upon the test results and the item level pass/fail criteria. An estimate of failure risk may be included.

**5.8.2.7. Summary of activities**

Summarize the major testing activities and events. Summarize resource consumption data, e.g., total staffing level, total machine time, and total elapsed time used for each of the major testing activities.

**5.8.2.8. Approvals**

**Name:** Ms. Mahum Adil

**Title:** Project Supervisor

**Signature:** \_\_\_\_\_

**Date:** \_\_\_\_\_