|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | | | | |
|  | |  | | | |
| Project: | | Dealership Management System (DMS)  CSE 5325 – Fall 2019  Project Management | | | |
| Module: | | COCOMO | | | |
| Deliverable: | | COCOMO Estimate Report | | | |
| Version: | | | 1.0 | Date: | 11/07/2019 |

Prepared by: Goutami Padmanabhan 1001669338

TABLE OF CONTENTS

[1. Introduction 2](#_Toc23340379)

[2. Estimating Factors 3](#_Toc23340380)

[2.1 Source of Lines of Code 3](#_Toc23340381)

[2.2 Scale Drivers 3](#_Toc23340382)

[2.3 Cost Drivers 3](#_Toc23340383)

[3 Project Final Timeline and Cost Structure 4](#_Toc23340384)

[4. Conclusion and Recommendations 5](#_Toc23340385)

[Appendices 6](#_Toc23340386)

# 1. Introduction

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e. **number of Lines of Code**. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort & Schedule.

The purpose of this document is to understand the costs that are to be incurred by the project through various factors and drivers such as Scale drivers and Cost drivers. The criteria considered in the scale drivers and cost drivers help us understand and determine the costs estimated to be spent at a minute level. This document also shows an almost accurate and safe cost estimation for a project. By reading this document or report one might be able to plan the budget for a project in a well-informed manner as it goes in detail about what causes the cost and where and in which phase it occurs.

The previous schedule made in the Microsoft Project Plan was for 3 months and we were able to estimate the cost and project completion in those 3 months. The new schedule has a duration of 5.7 months from the COCOMO tool. The time constraint was not considered for the previous assignment and hence we were able to estimate. But in this assignment, we clearly define the Scale drivers and the Cost drivers for estimation.

I would recommend that we should go ahead with the project and ask the client to divide the project into two phases. The first phase having the important modules to be implemented and the next phase with the less important modules like the Notes module. I believe that going ahead with the project would really give the team with good exposure as well as help us get into the Dealership market and gain more customers.

# 2. Estimating Factors

## 2.1 Source of Lines of Code

The following is the number of lines of code delivered as part of this project, A justification for the total amount of LOC is provided.

|  |  |
| --- | --- |
| **SLOC | Source Lines Of Code** | Value Chosen: 5000 |
| Justification: There are 10 modules such as Welcome page, User Registration, Login, Makers, Models, Customers, Inventory, Sales, Notes and Search that are expected to be built by the entire team. Since each of these modules involve a lot of front-end (UI) as well as backend (database) activities, the number of lines of code for each module will be a minimum of 1000. Assuming that we code the minimum number of lines for each of those 10 modules we get a minimum of (1000\*10) 10000 lines of code. But due to the constraint of the COCOMO tool we consider the maximum lines of code it allows i.e. 5000. This is the main reason we have considered 5000 source lines of code. | |

## 2.2 Scale Drivers

The following is the list of scale drivers, the values applicable to this project and a justification for each value chosen:

|  |  |
| --- | --- |
| **PREC | Precendentedness** | Value Chosen: High - Generally Familiar |
| Justification: When compared to the projects the team has done before, the new project is quite familiar. The team has already worked on a similar project of creating a web application from the scratch. The team has experience of developing web applications similar to the new one. Though the requirements differ from previous projects, the team is confident of the knowledge they have on this area. Hence the team is generally familiar with such a project. | |

|  |  |
| --- | --- |
| **FLEX | Development Flexibility** | Value Chosen: Nominal – Some relaxation |
| Justification: The requirements for this project was given only on a high level. The client expects us to come up with more requirements based on the description they have given. The client is always welcome to new changes suggested in the requirements by the team. The client just wants a web application to improve their business in the dealership market. Hence there is some flexibility and relaxation to the team based on requirements. But the sole purpose of the client cannot be defeated. Adding more functionalities to the existing requirements is welcome. Hence there is only some relaxation. | |

|  |  |
| --- | --- |
| **RESL | Architecture / Risk Resolution** | Value Chosen - High (Generally 75%) |
| Justification: The architecture of the project has already been defined by an experienced Architect in the team. The team has a risk resolution plan and can mitigate or handle risks during this process. The Architect along with the team have designed the project to quite some extent with the requirements that were given. There is still some part of the design that must be done in the design phase. Hence the Risk resolution and the Architecture of the project is done up to 75%. | |

|  |  |
| --- | --- |
| **TEAM | Team Cohesion** | Value Chosen: Very High – Highly cooperative |
| Justification: The relationship of the team with the stakeholders are really good. This was one of the reasons why we got the project in the first place. Their trust on us is high since we have delivered good projects before to a friend of the client. Due to the trust developed, the relationship with the stakeholders are good and highly cooperative. | |

|  |  |
| --- | --- |
| **PREC | Process Maturity** | Value Chosen: Nominal - Repeatable |
| Justification: On the SEI Maturity scale, our organization rates SEI CMM Level 2. Since ours is a growing organization and needs a lot of process improvements and the processes need to be set and standardized throughout the organization, the process maturity level is 2. | |

## 2.3 Cost Drivers

The following is the list of cost drivers, the values applicable to this project and a justification for each value chosen:

|  |  |
| --- | --- |
| **ACAP | Analyst Capability** | Value Chosen: High – 75th percentile |
| Justification: The analyst for this project is highly capable. Team has an experienced analyst who analyses and designs the project and has already done the design up to 75%. Only some part of it is left to complete in the design phase. This is a medium sized project and the analyst has good analytical ability, efficiency, cooperativeness and has good communication skills. | |

|  |  |
| --- | --- |
| **APEX | Application Experience** | Value Chosen: High – 3 years |
| Justification: The team has already worked in a similar kind of project for around 3 years. Although the requirements may differ in a minor way, the team has already delivered the same kind of project once before. Hence the team is highly capable. | |

|  |  |
| --- | --- |
| **PCAP | Programmer Capability** | Value Chosen: Very High – 90th percentile – your best team |
| Justification: The programmers in the project are highly capable. They have very good programming skills and skills that are rightly required for this project. They have complete knowledge on what they work, high ability, efficiency and thorough knowledge of what is expected of the client. They have high communication skills and cooperativeness with the client that is required for good programmers. Since they have already completed successful projects before, the team is confident about this one too. | |

|  |  |
| --- | --- |
| **PLEX | Platform Experience** | Value Chosen: Nominal – 1 year |
| Justification: The team has worked on Java servlets and JSP before for many years. But they have started working on web applications on HTML, CSS only before a year. The hardware requirements are new and hence they have moderate knowledge on the platform. They have worked on Red Hat Linux only for a year. | |

|  |  |
| --- | --- |
| **LTEX | Language and Tool Experience Capability** | Value Chosen: High – 3 years |
| Justification: As discussed before, though the platform is new, the team has good expertise on the language Java and also has experience on various tools such as Junit. They have an upper hand on the language and the tools that are expected to be used. They have been working on this for 3 years. | |

|  |  |
| --- | --- |
| **PCON | Personnel Continuity** | Value Chosen: Nominal – 12% turnover per year |
| Justification: Since this organization is at the growing stage and the process maturity is at level 2, the processes are yet to be standardized and the turnover is relatively moderate. The organization has a nominal turnover of 12% per year over the past 3 years. | |

|  |  |
| --- | --- |
| **TOOL | Use of Software tools** | Value Chosen: Nominal – Basic life-cycle tools, moderately integrated |
| Justification: The software tools the team will be using are Maven, Junit and JaCoCo for testing and other development purposes. These are basic life-cycle tools used in all software projects. They are hence moderately integrated. The team does not use any other tool that incurs some extra cost. The basic tools would be sufficient for this project. | |

|  |  |
| --- | --- |
| **SITE | Multisite Development** | Value Chosen: Extra High – Fully collocated. Interactive multimedia |
| Justification: The team is not split among several sites. The team is in a single place and they are fully collocated. The team members are located at walkable distance from each other. Hence there is no gap in the communication and have interactive multimedia. Any type of communication could be done directly. The team sits together during work. | |

|  |  |
| --- | --- |
| **SCED | Development Schedule** | Value Chosen: Very Low – 75% of the nominal schedule |
| Justification: The schedule has been highly compressed from the nominal schedule. There is 75% acceleration with respect to nominal schedule. This is one of the fastest schedules along with good amount of feasibility. | |

|  |  |
| --- | --- |
| **TIME | Execution Time Constraint** | Value Chosen: Nominal - <=50% use of available execution |
| Justification: The CPU time used by the software developed by our team is normal. Due to our efficient programmers, there has been an attempt to make the CPU execution time nominal. It takes less than 50% of the available execution time. | |

|  |  |
| --- | --- |
| **STOR | Main Storage Constraint** | Value Chosen: Nominal - <=50% use of available storage |
| Justification: The main memory used by the software developed by our team is normal. Due to our efficient programmers, there has been an attempt to make the space storage nominal. It takes less than 50% of the available storage available to the team. | |

|  |  |
| --- | --- |
| **PVOL | Platform Volatility** | Value Chosen: Low – Major change every 12 months; Minor change every month |
| Justification: The platforms such as Red Hat Linux and DBMS such as MySQL has vey less changes. Any of the minor changes in Linux are released every month. The major changes are done every 18 months or 12 months. The platforms do not change frequently and are reliable. | |

|  |  |
| --- | --- |
| **RELY | Required Reliability** | Value Chosen: Low – easily recoverable losses |
| Justification: In case of software failure, the backups are taken at constant intervals by the team. Since this application is not a highly critical application like in case of banks, this application failure may cause very less impact. The losses are easily recoverable from the backups taken from time to time. | |

|  |  |
| --- | --- |
| **DATA | Database Size** | Value Chosen: High – 100 <= (Database bytes/ SLOC) <1000 |
| Justification: Since the number of simultaneous users for this application allowed is 500, the application need large amount of data that must be used to test the software. Different kinds of data have to be given to the application during the testing phase in order to come up with a good product. Also, the use of 500 users simultaneously may involve large amount of data in production. | |

|  |  |
| --- | --- |
| **CPLX | Product Complexity** | Value Chosen: Nominal – Nested code, standard math routines, multiple files |
| Justification: The complexity of software is determined by the number of nested loops in the program or software developed. Since the programmers already have experience with such kind of application, language and tools, they will code in a good way and make the complexity nominal. | |

|  |  |
| --- | --- |
| **RUSE | Required Reusability** | Value Chosen: Nominal – Across project |
| Justification: The software components developed in this project are programmed in such a way that they can be reused in a later project. Module-wise coding methods is used to make the component reusability normal. The components can be reused across the project in various other components. The programmers make an effort to code in the form of functions or methods that can be called later. Hence the reusability factor is nominal. | |

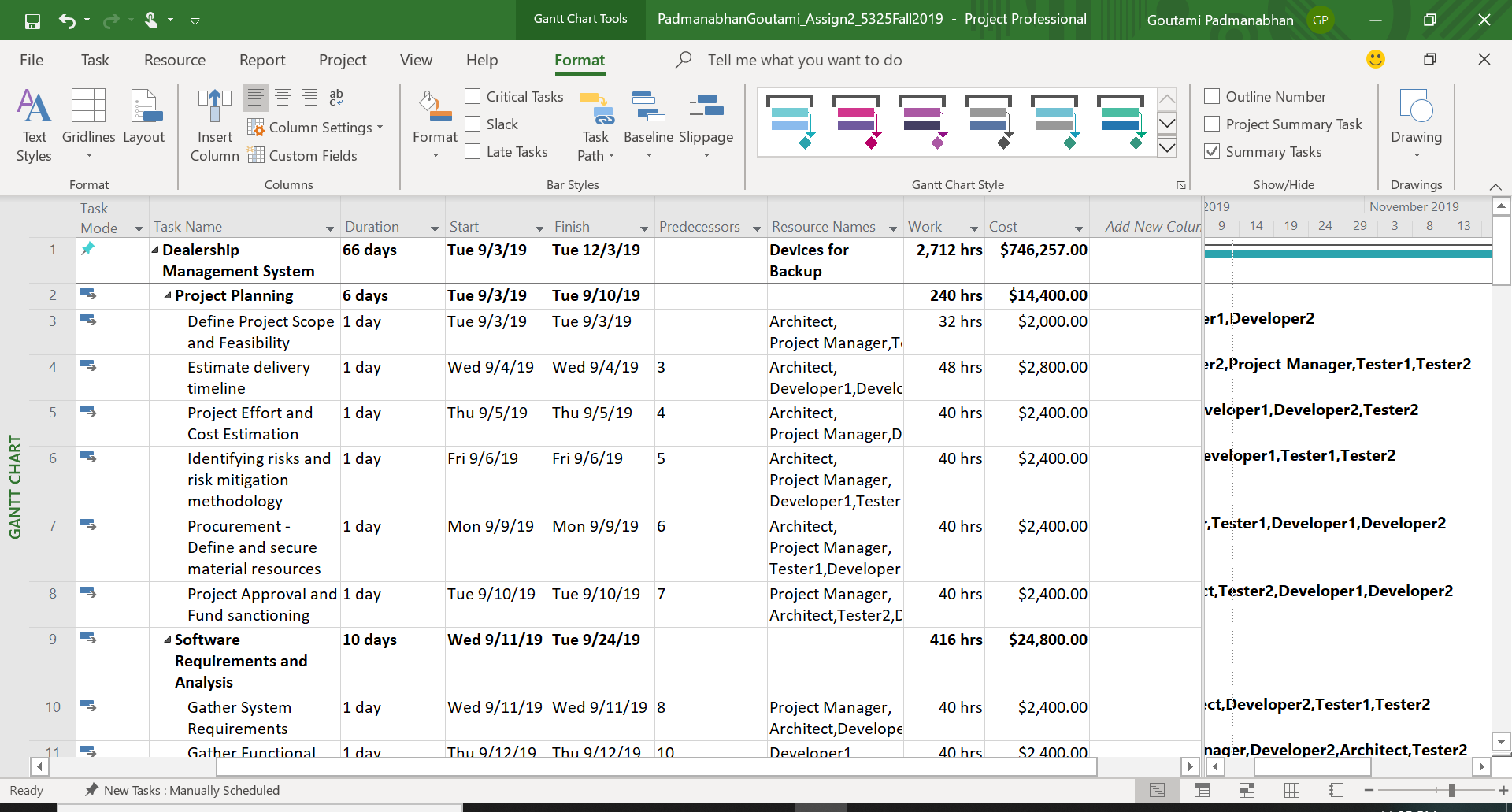
|  |  |
| --- | --- |
| **DOCU | Documentation match to life-cycle needs** | Value Chosen: Nominal – Right-sized to life-cycle needs |
| Justification: The project involves a documenting phase where all the team members are involved in documenting any changes made by each and every member. Code documentation is done to understand the software technically. User guide documentation is performed so that a layman can understand how to use the application. Right amount of documentation is done for this project. | |

# 3 Project Final Timeline and Cost Structure

**Previous Cost, Work and Duration:**

The estimation done previously using Microsoft Project Plan (MPP) is shown below:

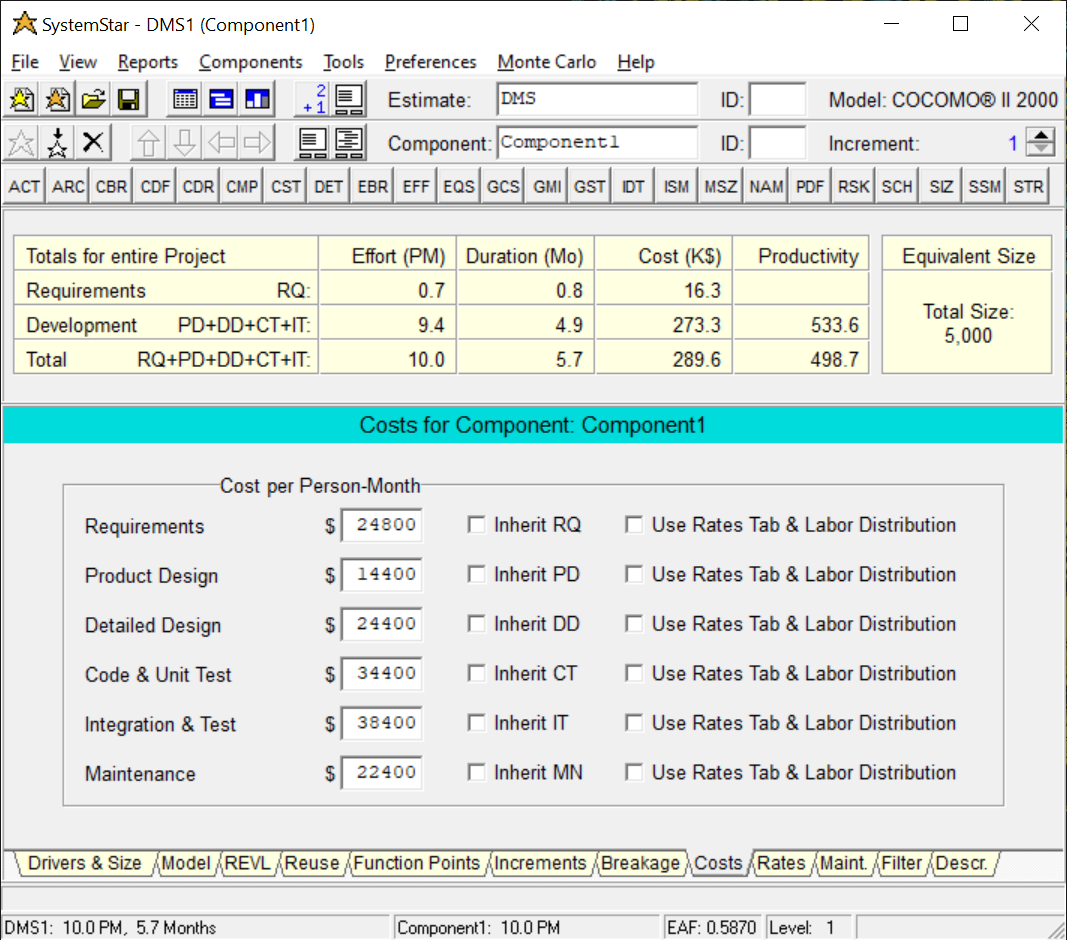
|  |  |  |
| --- | --- | --- |
| **MPP Duration** | **MPP Work** | **MPP cost** |
| 66 days | 2,712 hours | $746,257.00 |



|  |  |
| --- | --- |
| **Type of resource** | **Cost in $** |
| Non-Human resources | $340,457.00 |
| Human resources | $405,800.00 |
| **Total Cost** | **$746,257.00** |

**New Schedule (Duration):**

The duration of the new schedule from the COCOMO estimation tool is as shown in the screenshot below:



The new schedule takes 5.7 months to complete the project. This is high when compared to the previous assignment (Assignment #2 - MPP). The old schedule showed a duration of 3 months (66 days). By considering various scale drivers and cost drivers we have made an accurate estimate of 5.7 months for this project.

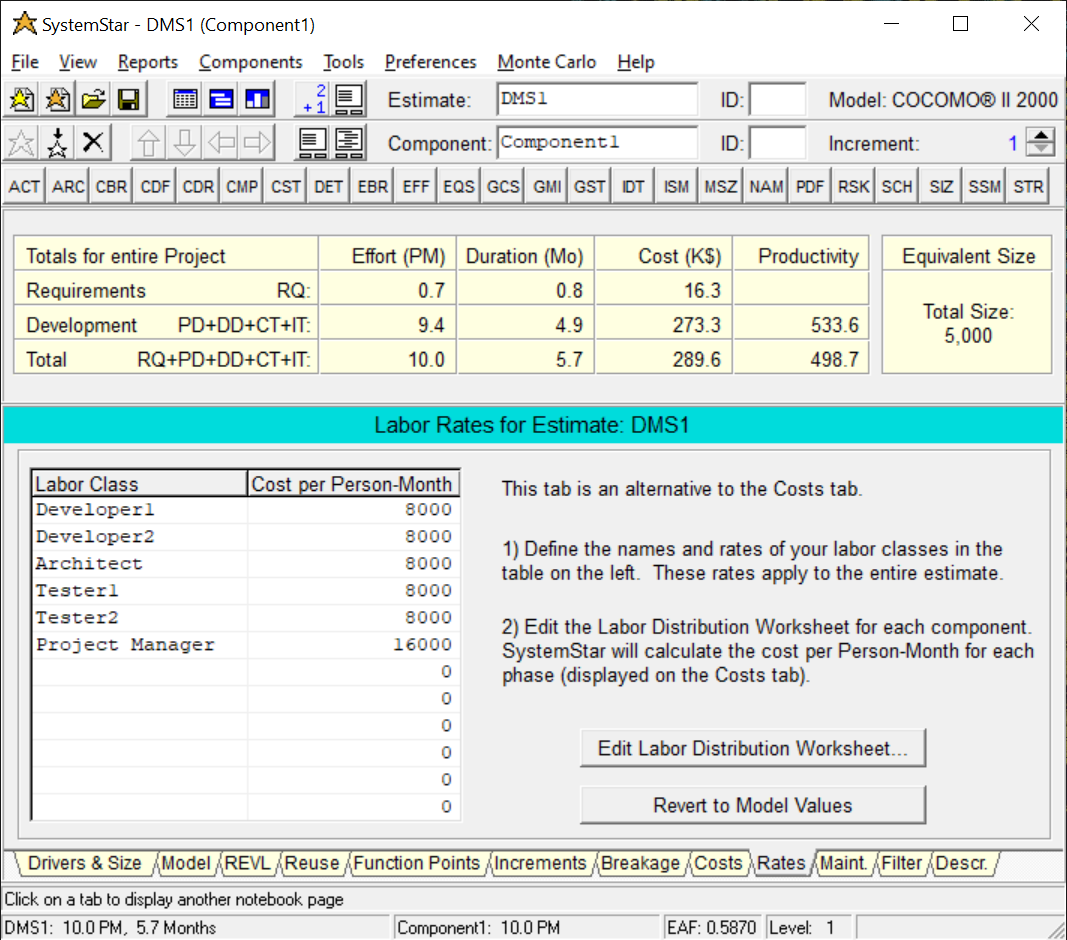
**COCOMO Estimated Costs (Human Resources):**

The costs for human resources are calculated base on the two classification of employees.

1. Project Manager
2. Rest of the employees i.e. Developers, Testers and Architect

The salary given to each classification of the employee is as shown below:

|  |  |
| --- | --- |
| **Role** | **Cost/hour** |
| Project Manager | $100/hr |
| Rest of the employees i.e. Developers, Testers and Architect | $50/hr |



**Non-Human Resources:**

The total cost of non-human resources is estimated up to $310,457.00

|  |  |
| --- | --- |
| **Resource Name** | **Cost/Use** |
| Microsoft Project Professional License | $1,038.00 |
| Electricity | $20,000.00 |
| Health insurance (50% extra surge) | $120,000.00 |
| Application Servers | $12,000.00 |
| Database Servers | $12,000.00 |
| Monitors and Desktops | $7,000.00 |
| Other Devices | $1,000.00 |
| Operating Systems License | $10,000.00 |
| Database License | $10,000.00 |
| Security Licenses | $5,000.00 |
| Testing Tools | $2,000.00 |
| Devices for Backup creation | $10,000.00 |
| Project Reserve and Emergencies | $100,419.00 |

**Profit:**

The project cost estimate without any profit is $289,600.00+$310,257.00=$599,857.00. The profit margin is 50%. The profit margin is taken as minimum 50% because we give advantage to the client. Since we are trying to get hold of the dealership market and its prospective clients, we lure the customers by getting the bare minimum profit margin.

The profit estimate is $299,928.50

**Total Cost:**

The project cost estimate along with the profit margin i.e. the Total cost of the project is $899,785.50

The COCOMO estimation using SoftStar tool has been attached to this document below:



# Conclusion and Recommendations

The difference between Microsoft Project Plan estimation and COCOMO II Tool estimation are listed below:

|  |  |  |
| --- | --- | --- |
| Parameters | Microsoft Project Plan Cost Estimation | COCOMO II tool Cost estimation |
| Duration | 3 months (66 days) | 5.7 months |
| Work | 2,712 hours | 8,208 hours |
| Cost | $746,257.00 | $899,785.50 |

From the above table we infer that COCOMO estimation is larger than Microsoft Project plan estimation. The time constraint was not considered for the previous assignment and hence we were able to estimate. Also, the size, lines of code and complexity of the code was not taken into consideration. But in this assignment, we clearly define the 5 Scale drivers and the 17 Cost drivers for estimation.

The factors that were missed in the Microsoft Project Plan were Number of lines of code. We did not check whether the schedule is compressed from the Nominal schedule. We did not take into account how much CPU time our software will use to run. We also did not consider the complexity of the project, how big the project is, the lines of code are more nested or not. These factors were not taken into consideration and hence a lesser cost estimation was done.

All the above factors that were missed in previous assignment were considered in COCOMO tool and hence we got an accurate estimate. The factors such as Number of lines of code, SCED - Development Schedule, TIME - Execution Time Constraint and CPLX - Product Complexity were now taken into consideration thereby giving us close to accurate estimation.

**Recommendations:**

I would recommend that we should go ahead and continue with the project. We can ask the client to divide the project into two different phases. The first phase can have the main modules such as Welcome Page, User Registration, Login, Makers, Models and Customers. These modules can be implemented first. The second and the last phase may include the less important modules like the Inventory, Sales, Notes and Search module. I believe that going ahead with the project would really give the team with good exposure as well as help us get into the Dealership market and gain more customers.

I will also recommend the board of directors to increase the resources if they want the project to be delivered in 3 months. This project will give a lot of monetary benefit. Money, we get from such a big customer will be highly useful in investing in company’s future projects. The profit margin estimates are a minimum of 50%. Taking up this challenging project would open the doors to core customers in the field of dealership market. Having a trusted and big customer in hand would make us increase our market share. This would be a steppingstone to meeting our business objective of becoming #1 in this segment.

# Appendices

Additional documents, printscreens of COCOMO reports, references.

**References:**

<https://www.geeksforgeeks.org/software-engineering-cocomo-model/>

<http://www.softstarsystems.com/overview.htm>

**Screenshots of various drivers for COCOMO II Estimation tool:**

