Project: Serve Request System (SRS)

CSE 5325 – Fall 2024

Project Management

Module: COCOMO

Deliverable: COCOMO Estimate Report

Version: [1.0] **Date: [04/10/2024]**

Prepared by: Prayag Parikh, 1002073011

TABLE OF CONTENTS

1. Introduction

The aim of this document is to outline the technical specifications and estimations for the development of a comprehensive Service Request System (SRS). This system is intended to streamline the process of managing service requests, facilitating efficient communication between users and service providers, and ensuring timely and accurate handling of service-related tasks. By detailing the Source Lines of Code (SLOC), Scale Drivers, and Cost Drivers, this document provides insights into the project's scope, scale, and resource requirements.

The Service Request System (SRS) is designed to address the growing need for a centralized platform to manage service requests effectively. It encompasses features such as user registration, login, security setup, service category management, provider registration, service request placement, payment handling, reviews, order history, search functionality, map integration, customization, advertisement, final testing, deployment, and documentation.

Through the utilization of the Incremental Model, the project team aims to iteratively develop and refine the SRS, ensuring that each phase of development is meticulously planned and executed. By leveraging the insights derived from the COCOMO estimation, the Project Manager can effectively allocate resources, optimize project schedules, and make informed decisions to enhance the overall quality and efficiency of the final product.

Upon completion of the estimation process, this document will serve as a guiding framework for project planning and execution, providing valuable insights into the optimized allocation of resources, effort, and cost. It aims to empower the project team to navigate each phase of development with precision, ultimately delivering a high-quality Service Request System (SRS) that meets the evolving needs of users and stakeholders, while maximizing profitability and ensuring customer satisfaction.

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: 7000**
Justification: The SLOC value chosen for the Service Request System (SRS) project is estimated to be 7000. This estimation considers the complexity of the system's functionalities, including user registration, authentication, service categories, provider registration, service request placement, payment handling, review system, order history tracking, search functionality, map integration, setup control fields, advertisement opportunities, and more. Each of these features requires significant code implementation to ensure robustness, security, and usability. Additionally, the SRS project involves multiple layers of development, including frontend, backend, and database components, contributing to the overall codebase. But as we are using the trial version of System Star, it bounds us to be in maximum limit of 5000 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: **Low**
Justification: The project team includes developers with varying levels of experience and expertise, including a senior full-stack engineer and a senior testing engineer. Additionally, the project manager brings leadership and guidance based on prior project management experience. With a diverse team and experienced leadership, the project has a precedent for handling similar projects, indicating a low level of precedentedness.	

FLEX | Flexibility	Value Chosen: **High**
Justification: The project involves developing a web platform, mobile applications for Android and iOS, and integration with various external services such as payment partners and GPS mapping services. The requirements may evolve during the project due to changing market demands or technological advancements. Additionally, the team includes developers proficient in multiple technologies, allowing for flexibility in adapting to changing requirements and technologies.	

RESL | Risk Resolution	Value Chosen: **High**
Justification: The project involves integrating multiple technologies and external services, including AWS Lambda for production deployment, AWS Database for data management, and various payment services partners. Additionally, the project requires robust security measures due to handling sensitive user data and financial transactions. The complexity of the architecture and the potential risks associated with integrating multiple systems indicate a high level of architecture and risk resolution.	

TEAM | Team Cohesion	Value Chosen: **Low**
Justification: The project team consists of developers with diverse skills and backgrounds, including full-stack engineers, an Android+iOS developer, and a database engineer. While the team has the necessary expertise to handle the project requirements, there may be challenges in coordinating efforts and maintaining cohesion due to the distributed nature of the team and the varying levels of experience. Effective communication and collaboration tools will be essential to mitigate potential cohesion issues.	

PMAT | Process Maturity	Value Chosen: **Nominal**
Justification: While the company is in a phase of expansion and has established certain procedures and practices, there is still room for improvement in terms of overall process maturity. The project team operates at a nominal level of process maturity, with some repeatable processes in place but opportunities for further refinement and optimization. As the company progresses, there is potential for enhancing process maturity through continuous improvement initiatives and learning from project experiences.	

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: **Very High - 90th percentile**
Justification: With the involvement of an experienced project manager and a senior full-stack engineer, coupled with the diverse expertise of the development team, the analyst capability for the project is considered very high. The team's collective experience in requirements gathering, analysis, and system design will contribute to effective understanding and translation of project objectives into technical specifications.	

APEX | Application Experience	Value Chosen: **Very High – 6 years**
Justification: The development team comprises seasoned professionals with extensive experience in developing similar service request systems, web applications, and mobile applications. The collective experience spans over 6 years, covering full-stack development, Android/iOS development, database management, and API development. This high level of	

application experience will significantly contribute to efficient development, risk mitigation, and adherence to best practices.

PCAP | Programmer Capability	Value Chosen: **Very High – 90th percentile**
Justification: With a diverse team of highly skilled developers, including senior engineers specializing in various areas (full-stack, mobile, database, APIs), the programmer capability for this project is considered very high. Their expertise in multiple programming languages, frameworks, and tools, coupled with strong problem-solving abilities, will result in efficient coding, effective collaboration, and timely resolution of technical challenges.	

PLEX | Platform Experience	Value Chosen: **Very High – 6 years**
Justification: The development team has extensive experience working with the required platforms and technologies, such as AWS Lambda, Android Studio, iOS development, AWS databases, and various APIs. The collective experience spans over 6 years, ensuring a high level of familiarity with the development platforms, which will facilitate smoother integration, deployment, and maintenance processes.	

LTEX | Language and Tool Experience	Value Chosen: **Very High – 6 years**
Justification: The developers are highly proficient in various programming languages and tools required for the project, including Node.js, SQL, Android/iOS development frameworks, APIs, and testing tools like Selenium. With over 6 years of collective experience in these languages and tools, the team is well-equipped to handle development and testing tasks efficiently, leveraging best practices and avoiding common pitfalls.	

PCON | Personnel Continuity	Value Chosen: **High – 6% turnover per year**
Justification: Assuming a stable and experienced team composition with a low turnover rate, the personnel continuity is expected to be high. A turnover rate of 6% per year is considered reasonable for a project of this scale and complexity. This high level of continuity will ensure consistency in the development process, minimize knowledge transfer overhead, and facilitate smooth collaboration among team members.	

TOOL | Use of Software Tools	Value Chosen: **Nominal**
Justification: The project utilizes various tools like version control systems, testing frameworks (e.g., Selenium), deployment platforms (AWS Lambda), and development environments (Android Studio, Xcode). While these tools are integrated to some extent, they may not constitute a fully automated and integrated toolset as described earlier.	

SITE | Multi Site Development	Value Chosen: **Same city or metro area – Wideband electronic communication.**
Justification: Multi Site Development for SRS is chosen as **Same city or Metro area**. The entire project team consisting of project manager, developers and a tester all live in the same metropolitan area. This enables the team to get together to work on the project, rather than working remote. Communication among the team members is extremely important for any project to make sure that the entire team is up to date with what's happening in the project and everyone is on the same page. This in turn enables for easier and hassle-free communication.	

SCED | Development Schedule	Value Chosen: **Nominal – 100% of nominal schedule**
Justification: While the project has a specified deadline, and the schedule was initially considered tight, it may be more reasonable to select the "Nominal" value for the SCED cost driver. This assumes that the schedule is reasonable and aligns with typical project timelines, given the experienced team and careful planning.	

TIME | Execution Time	Value Chosen: **Nominal – <=50% use of available execution time**
Justification: While the system involves various components and features, such as user authentication, service request management, payment processing, rating systems, and map integration, the experienced team and the use of cloud infrastructure (AWS Lambda) should allow for nominal execution time constraints. The team's expertise and the scalable cloud environment will facilitate efficient performance optimization and resource utilization.	

STOR | Main Storage	Value Chosen: **Very High - 85% use of available storage.**
Justification: With the need to store massive volumes of user data, service requests, transactions, and potentially multimedia content (images, videos), the project is expected to have very high storage requirements. The scalable cloud infrastructure (AWS databases and storage) should accommodate these storage needs, potentially utilizing up to 90% of available storage capacity.	

PVOL | Platform Volatility	Value Chosen: **Low. Major change every 12 months, Minor change every 1 month.**
Justification: The project utilizes established platforms and technologies (e.g., AWS, Android, iOS) that are relatively stable and have low volatility in terms of major changes. While minor updates, patches, or new feature releases may occur more frequently (approximately every month), significant platform changes or migrations are expected to be infrequent, occurring no more than once every 12 months.	

RELY | Required Reliability	Value Chosen: **High – Risk to human life**
Justification: As the system handles sensitive user data, financial transactions, and potentially **critical service requests (e.g., emergency services**), a very high level of reliability is required. Any failure or data breach could potentially risk user safety or result in significant financial losses. Robust security measures, fault tolerance, and fail-safe mechanisms must be implemented to ensure the highest level of reliability and data integrity.	

DATA | Database Size	Value Chosen: **Very High - (Database bytes / SLOC) >= 1000.**
Justification: The project involves storing and managing massive volumes of user data, service requests, transactions, and potentially multimedia content (images, videos). The ratio of database bytes to codebase size (Source Lines of Code - SLOC) is anticipated to be greater than or equal to 1000, indicating a very high database size requirement.	

CPLX | Product Complexity	Value Chosen: **Very High**
Justification: The SRS platform involves highly complex control operations, such as user authentication, service request management, payment processing, rating systems, map integration, and real-time communication. The system must adhere to strict security, reliability, performance, and usability constraints, further increasing the overall product complexity. The integration of multiple external services and the need for robust error handling contribute to the very high complexity level.	

RUSE | Required Reusability	Value Chosen: **High - Across program.**
Justification: While the project does not explicitly mention reusability requirements, it is generally beneficial and industry-standard practice to develop code with potential reuse across different components or future projects. Given the modular nature of the SRS platform and the potential for expanding functionality or integrating with other systems, a high level of reusability should be targeted to facilitate future development and maintenance efforts.	

DOCU | Documentation match to life-cycle needs	Value Chosen: **Nominal. Right-sized to life-cycle needs.**
Justification: Documentation for SRS is chosen as Nominal. The project team will ensure that at all stages of the development and testing, the reports are documented, so that it can be revisited and understood in the future by anybody who wishes to reuse some of the features of the project as long as they have the required rights to do so. Code is also documented to make sure that the software is understood on a technical level. Hence the DOCU value chosen is **Nominal.**	

3 Project Final Timeline and Cost Structure

PREVIOUDLY: #Assignment 2 [Cost, Work, and Duration]:

The total cost, work and duration of the project estimated by MPP for the PHMS is tabulated as below:

Work (days)	**64.5 days**
Duration (hrs)	**1,404 hrs**
Development Cost ($)	**$93,240.00**

The split up of Human and Non-Human Resources, Overhead cost of the above is given as follows:

- **Human Resources** [cost based on the Cost Resource Sheet of Microsoft Project Planner]

  Resource salaries cost[64.5 days]: $86,840

- **Non- Human Resources:**

Material name	**Cost (of 65 days)**
AWS Lambda	$800
AWS Database	$2000
Android Studio Professional	$1000
Payment services	$1200
GPS Mapping Services	$400
Version control systems	$400
Selenium & other testing tools	$600
Total Software/Hardware cost	**$6400**

Overhead cost:

Overhead cost is assumed to be 50% of the Resources = 0.5 * (human resources + non-human resources)
= 0.5 * (86,640 + 6,400)
= $46,520
Which gets added in the cost = $86,640 + $6,400 + $46,520 = **$139,560**

Total Profit:

If a profit of 50% is considered for the above cost, then:

Development cost including overhead cost: $139,560.00

Profit: 0.5 * (139,560) = $69,780.00

Total Cost:

Development cost including overhead + Profit = $139,560 + $69,780 = **$209,340.00**

NEW Values obtained from System Star COCOMO:

Efforts	11.3 Person-Months
Duration	9.1 Months
Average staffing: Efforts/ Duration	1.24 Persons
COCOMO Estimated Cost	**$170,500**

- **Important Note:** This cost of $170,500 includes the resources and material cost both because the values in the Costs tab of SoftStar Systems is taken from the Assignment #2 (Cost column) and this Cost column of Microsoft Project Planner has Resources+Material cost both included. Therefore, I'm directly adding Overhead cost below.

Overhead cost:

Overhead cost is assumed to be 50% of the Resources = 0.5 * (COCOMO Estimated Cost)
= 0.5 * (170,500)
= $85,250
Which gets added in the cost = $170,500+ $85,250 = **$255,750**

Total Profit:

If a profit of 50% is considered for the above cost, then:

Development cost including overhead cost: $255,750.00

Profit: 0.5 * (255,750) = $127,875.00

Total Cost:

Development cost including overhead + Profit = $255,750 + $127,875 = **$383,625.00**

4. Conclusion and Recommendations

{A brief recap of the project highlighting if and why are previous and current estimates different (what factors do COCOMO bring that were not considered before). A recommendation on what to do – CONTINUE, CANCEL, SPLIT…}

The below table compares the difference in the cost, work and duration estimates between Microsoft Project Plan estimation and COCOMO II Tool estimation.

Estimation Method	**Factors**	
	Duration	**Cost**
Microsoft Project Plan (MPP)	3.1 months	**$209,340.00**
COCOMO Softstar Estimation	9.1 months	**$383,625.00**

The Service Request System (SRS) project estimation underwent a significant shift when transitioning from the Microsoft Project Plan (MPP) to the COCOMO II Tool estimation. The differences in the cost estimates between the two methods warrant analysis to understand the underlying factors contributing to this variation and provide recommendations for moving forward.

The table comparing the two estimation methods reveals a substantial variance in both the project duration and cost. The COCOMO estimator projected nearly double the cost compared to the MPP estimation. This divergence can be attributed to several factors that were not adequately accounted for in the MPP estimation.

Main reason: One crucial factor is the consideration of Source Lines of Code (SLOC) and the Estimating Model, both of which are integral components of the COCOMO estimation method. These factors influence the size of the project codebase and the estimation methodology used, respectively. Additionally, factors such as CPU execution time, storage space, database size, platform expertise, and other cost drivers were not fully accounted for in the MPP estimation but were meticulously considered in the COCOMO estimation process.

The inclusion of these previously overlooked factors in the COCOMO estimation led to a more comprehensive and accurate assessment of the project's duration and cost. While some of these factors may have inflated the final estimates, they ultimately contribute to a more realistic portrayal of the project's requirements and challenges.

Recommendations:

Based on the analysis of the estimation variances and considering the comprehensive nature of the COCOMO estimation, the following recommendations are proposed:

Proceed with COCOMO Estimates: Given the thorough consideration of scaling and cost drivers in the COCOMO estimation, it is advisable to base project planning and resource allocation on these estimates. Despite the higher projected cost and duration, the COCOMO method provides a more accurate representation of the project's complexity and resource requirements.

Optimize Project Duration: While the COCOMO estimation suggests a longer project duration of 9.1 months, efforts should be made to expedite project delivery. Additional resources may be allocated to the project team to enhance productivity and accelerate

development timelines. Shortening the project duration can enable earlier deployment and realization of benefits from the SRS application.

Continuous Monitoring and Adaptation: Throughout the project lifecycle, it is essential to monitor progress against the COCOMO estimates and adapt strategies as needed. Regular reviews and adjustments can help mitigate risks, address challenges promptly, and ensure alignment with project objectives.

In conclusion, while the COCOMO estimation presents higher cost and duration projections compared to the MPP method, its comprehensive consideration of key factors provides a more reliable basis for project planning. By embracing the COCOMO estimates and implementing proactive measures to optimize project execution, the SRS project can achieve its objectives effectively while minimizing risks and maximizing success.

Appendices

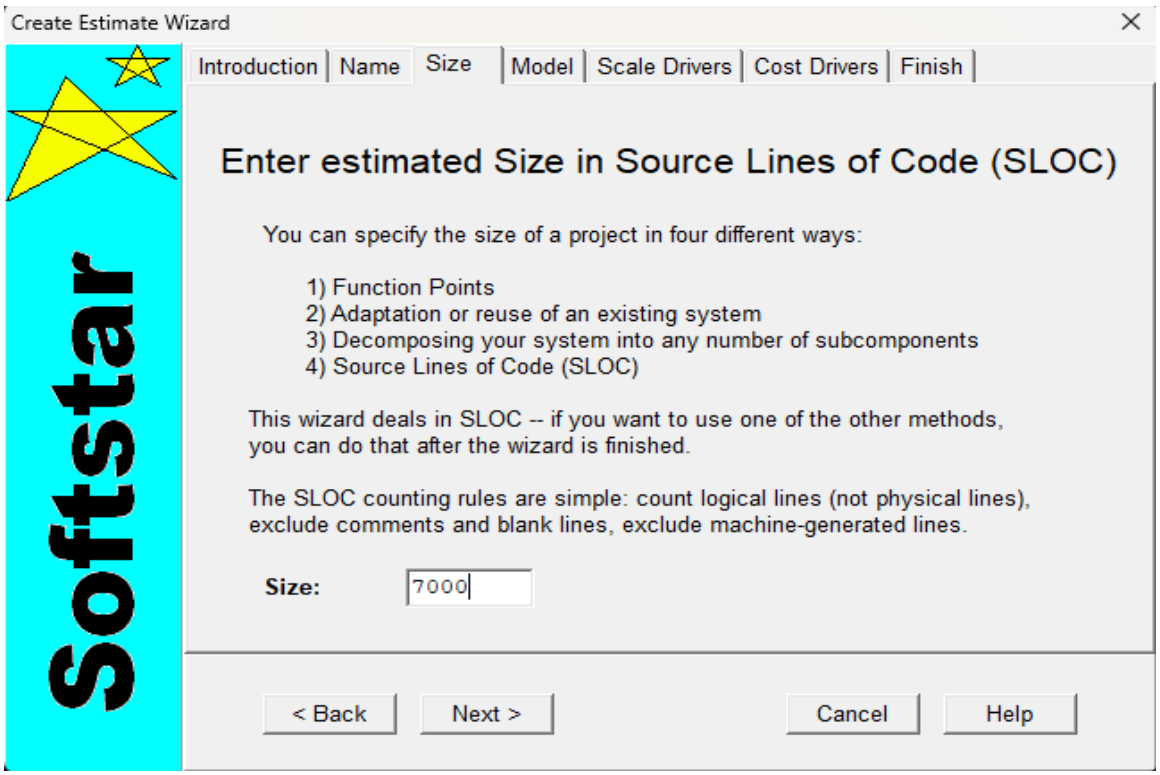
1. https://www.theknowledgeacademy.com/blog/cocomo-model-in-software-engineering/
2. Deeper understanding and SLOC: https://www.rose-hulman.edu/class/cs/csse372/201310/Homework/CII_modelman2000.pdf

Screenshots:

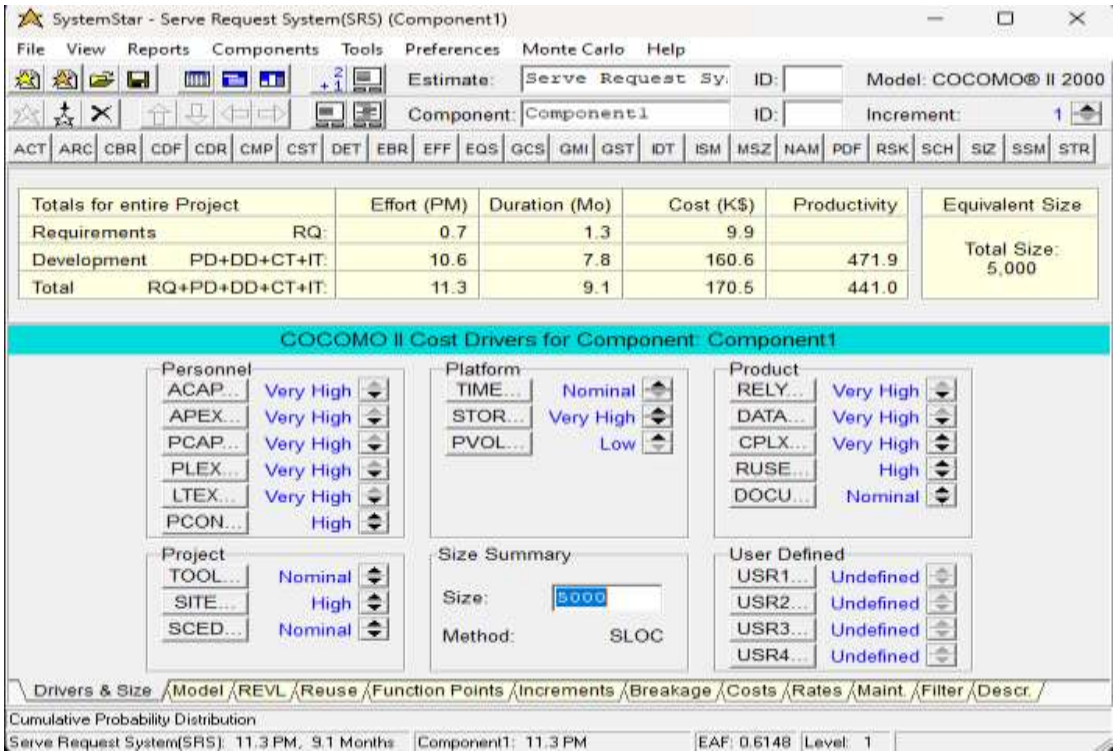
1. **Cost estimation before any overhead cost/profit from MPP:**

	WBS	Task Name	Duration	Start	Finish	Predecs	Resource Names	Work	Cost
1	1	1 Serve Request System (SRS)	64.5 days	Thu 01-02-24	Wed 01-05-24			1,404 hrs	$93,240.00
2	1.1	1.1 Project Planning	10 days	Thu 01-02-24	Wed 14-02-24			120 hrs	$10,650.00
3	1.1.1	1.1.1 Project Vision and Scope	2 days	Thu 01-02-24	Fri 02-02-24		Developer 1 (Senior Full-stack	32 hrs	$2,880.00
4	1.1.2	1.1.2 Requirements Gathering	2 days	Mon 05-02-24	Tue 06-02-24	3	A Project Manager	16 hrs	$1,600.00
5	1.1.3	1.1.3 Project Roadmap	2 days	Wed 07-02-24	Thu 08-02-24	4	A Project Manager	16 hrs	$1,600.00
6	1.1.4	1.1.4 Resouce planning and Hiring	2 days	Fri 09-02-24	Mon 12-02-24	5	A Project Manager	16 hrs	$1,600.00
7	1.1.5	1.1.5 Release Planning	1 day	Tue 13-02-24	Tue 13-02-24	5,6	A Project Manager,MB-Version	8 hrs	$850.00
8	1.1.6	1.1.6 Sprint Planning	1 day	Wed 14-02-24	Wed 14-02-24	5,6,7	A Project Manager,Developer 4	32 hrs	$2,080.00
9	1.2	1.2 Sprint 1: (User Registration, Login, Security Setup)	10 days	Thu 15-02-24	Wed 28-02-24			192 hrs	$12,090.00
10	1.2.1	1.2.1 User Story Analysis and Design	1 day	Thu 15-02-24	Thu 15-02-24		A Project Manager,Developer 1	40 hrs	$2,720.00
11	1.2.2	1.2.2 Development of user registration functionality	1 day	Fri 16-02-24	Fri 16-02-24	10	Developer 2 (Full-stack Engineer),MB-Version Control	16 hrs	$1,250.00
12	1.2.3	1.2.3 Development of user login functionality	1 day	Mon 19-02-24	Mon 19-02-24	11	Developer 2 (Full-stack Engineer),Developer 1 (Senior	16 hrs	$1,090.00
13	1.2.4	1.2.4 Backend database configuration for user management	1 day	Tue 20-02-24	Tue 20-02-24	11,12	Developer 5 (Node & SQL Developer for APIs),Developer	24 hrs	$1,320.00
14	1.2.5	1.2.5 API service code development for	1 day	Wed 21-02-24	Wed 21-02-24	13	Developer 4 (Database	16 hrs	$940.00

2. **From SoftStar Systems [SLOC]:**



3. **Cost Drivers:**



4. **Costs:**