GOVERNMENT POLYTECHNIC, AHMEDABAD

COMPUTER ENGINEERING DEPARTMENT

Affiliated
To

Gujarat Technological University, Ahmedabad

Micro project Report

D. E. Second Year (Semester–IV)

Introduction to Software Engineering

(4340702)

Government Polytechnic, Ahmedabad
Computer Engineering Department

CERTIFICATE

This is to certify that

Sr. No.	Enrollment No.	Name
1	226170307228	Vatsal Vaghela
2	226170307235	Yadav Prajwal P.
3	226170307233	Vohra Moin
4		

Of **Fourth** semester of Diploma in Computer Engineering of Government Polytechnic, Ahmedabad has completed the Micro-Project satisfactorily in Subject **Introduction to Software Engineering(4340702)** for the academic year **2023-2024** asprescribed in the curriculum.

Lecturer,
Computer Engg. Dept.,
Government Polytechnic, Ahmedabad

HOD
Computer Engg. Dept.,
Government Polytechnic,
Ahmedabad

RUBRICS FOR MICRO-PROJECT ASSESMENT

Parameters	Allocated Marks	High	Medium	Low
Problem Analysis and Solution(R1)	8	Problem is Properly Analyzed and Solved	Problem is Properly Analyzed but Partially Solved	Problem is properly Analyzed but not Solved.
		8 Marks	5 Marks	2 Marks
Viva Voce(R2)	2	Student Answered All The Viva Voce Questions	Student Answered Only A Few Viva Voce Questions	Student Did Not Answer Any Viva Voce Questions
		2 Marks	1 Marks	0 Marks

INDEX

Enrollment Number	Student Name	Marks(R1)	Marks(R2)	Total Marks
226170307228	**Vatsal Vaghela**			
226170307235	**Yadav Prajwal P.**			
226170307233	**Vohra Moin**			

Name and Sign of Faculty:

INDEX

Introduction

Are you looking for an online bus booking system project? We are here to help you. You can contact us. Online Bus Booking System cloud-based online software. This system would help customers to book a seat for their journey, book a bus. This system would also help the owner to manage the coaches, employees, clients, services, etc.

Bus Reservation System will increase the booking process faster, convenient, and comfortable. Customers can book their desired seats. They can check the availability of posts on a specific date. The customer can check availability, book a ticket, or cancel a ticket 24X7. The online system is available to use anytime. The user doesn't require to visit any office.

Abstract

The abstract of a bus reservation system generally involves the development of a web-based application that automates the process of booking bus tickets, making it easier for customers to reserve seats and check availability online. The system aims to streamline the booking process, reduce errors, and improve customer experience by eliminating the need for manual data entry and long queues at the counter. The system can also provide a report for the purpose of managing decisions, making it a valuable tool for the transportation industry. The system can also be accessed from anywhere in the world, making it convenient for customers to reserve tickets through telephone lines or the internet. The system also includes features such as seat selection, price per seat, and bill generation, making it a comprehensive solution for bus reservation needs.

Scope

The scope of a bus reservation system encompasses various aspects related to the functionality and features of the system. Based on the provided sources, the scope of a bus reservation system includes:

Maintenance of Passenger Records: The system involves the maintenance of detailed records of each passenger, including information like schedule, seat availability, price per seat, and bill generation

Online Reservation: Customers can reserve tickets from any part of the world through telephone lines or the internet. They can check availability, reserve selective seats, and make payments online

Seat Reservation and Cancellation: Customers have the ability to reserve seats, check for open positions on specific dates, and cancel tickets 24/7 through the online system without the need to visit a physical office

Advanced Inventory Control: The system allows for advanced inventory control, enabling operators to manage seat inventory efficiently and maximize seat utilization and income per kilometer

Integration Capabilities: The system provides comprehensive and well-documented APIs for integration with other systems, allowing for future-oriented booking and interaction with various platforms

Digital Marketing Tools: A modern bus reservation system should include digital marketing tools to enhance online presence, attract passengers, and boost ridership through social media, email marketing, and loyalty programs

Business Analysis Capabilities: The system should provide data in a structured and accessible format for business intelligence tools, enabling operators to analyze performance, make informed decisions, and plan future improvements

Efficient Information Update: An integrated reservation system is needed to efficiently handle information updates, reservation handling, and provide an easy-to-use platform for both the company and customers

Web-Based Application: The system is a web-based application that allows visitors to check bus availability, buy and pay for bus tickets online, enhancing the efficiency of the reservation process

Need

The need for a bus reservation system arises from the challenges faced by both bus operators and customers in the manual bus reservation process. The manual process is time-consuming, error-prone, and can lead to disputes between customers and operators. The need for an efficient, integrated, and easy-to-use reservation system is evident in the following ways:

Efficient Information Update and Reservation Handling: An integrated bus reservation system can streamline the process of updating information and handling reservations, providing a more efficient and accurate system for both operators and customers.

Seat Availability and Reservation: A bus reservation system can provide real-time information on seat availability, allowing customers to reserve specific seats and check availability on specific dates. This feature also enables customers to cancel tickets 24/7 through the online system without the need to visit a physical office.

Advanced Inventory Control: A bus reservation system can maximize seat utilization and income per kilometer by allowing passengers to select their favorite seat on all stop segments without affecting availability on other segments.

Integration Capabilities: Bus reservation systems can integrate with other systems, such as fleet ERP, registers, and social media, to provide a future-oriented booking experience and prevent complexity and difficulties.

Real-Time Omnichannel Sales Management: A bus reservation system can provide a real-time view of every accessible seat across the entire sales network, enabling operators to implement demand-based yielded pricing and manage sales in real-time.

Self-Service Capabilities: A bus reservation system can provide self-service functions, allowing passengers to manage their travel-related aspects on their own, improving cost efficiency for operators and enhancing customer satisfaction.

Real-Time Passenger Information: Modern bus reservation systems should provide real-time passenger information, such as seat assignments, bus schedules, and delays.

Data Management and Analysis: A bus reservation system can provide data in a structured and accessible format for business intelligence tools, enabling operators to analyze performance, make informed decisions, and plan future improvements.

Online and Offline Payment Options: Bus reservation systems can provide various payment options, such as credit cards, debit cards, internet banking, PayPal-like online wallets, or cash, to make it easier for customers to pay for their tickets.

Agile Model in BUS RESERVATION SYSTEM

To implement an Agile model in a Bus Reservation System, the XP (Extreme Programming) methodology is chosen for the development process. This methodology focuses on flexibility, communication, and customer involvement, which are crucial in agile development. By utilizing XP, the system aims to be more functional and user-friendly, potentially benefiting many bus transport companies in the future. Agile methodologies like XP emphasize iterative development, continuous feedback, and adaptability, which are essential in dynamic projects like a Bus Reservation System.

1. Planning

Gather requirements from stakeholders (bus operators, passengers, etc.)

Define the scope and objectives of the system

Create user stories and prioritize them based on business value

Estimate effort and assign stories to sprints

2. Designing

Design the system architecture and components

Create wireframes and mockups for the user interface

Design the database schema to store bus, passenger and booking information

Identify potential risks and plan for mitigation

3. Coding

Set up the development environment

Implement the core functionality in an iterative manner

Write unit tests to ensure code quality

Refactor code to improve performance and maintainability

4. Testing

Conduct integration testing to verify system components work together

Perform end-to-end testing to validate the system meets requirements

Carry out user acceptance testing with real users

Fix bugs and issues found during testing

5. Deployment

Package the application for deployment

Set up the production environment

Deploy the system to production

Monitor system performance and stability

6. Feedback and Improvement

Gather feedback from users on the deployed system

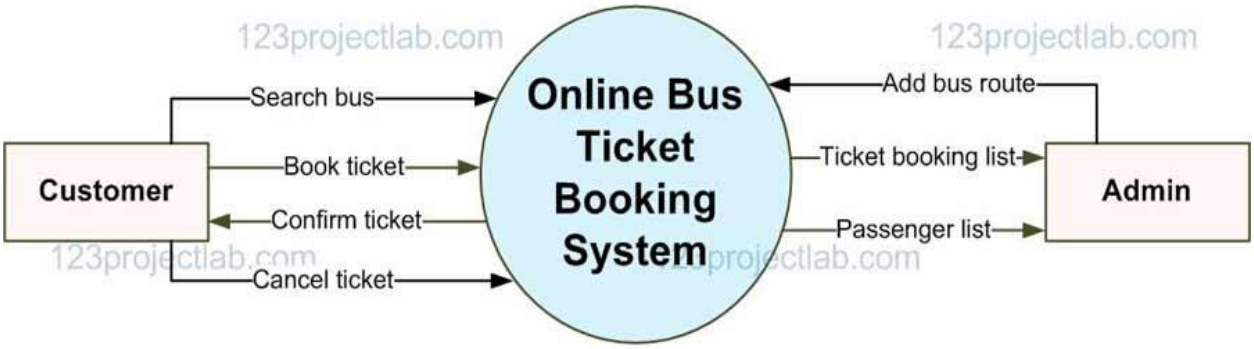
Analyze usage data and identify areas for improvement

Prioritize and plan for improvements in the next iteration

Continuously enhance the system based on user needs

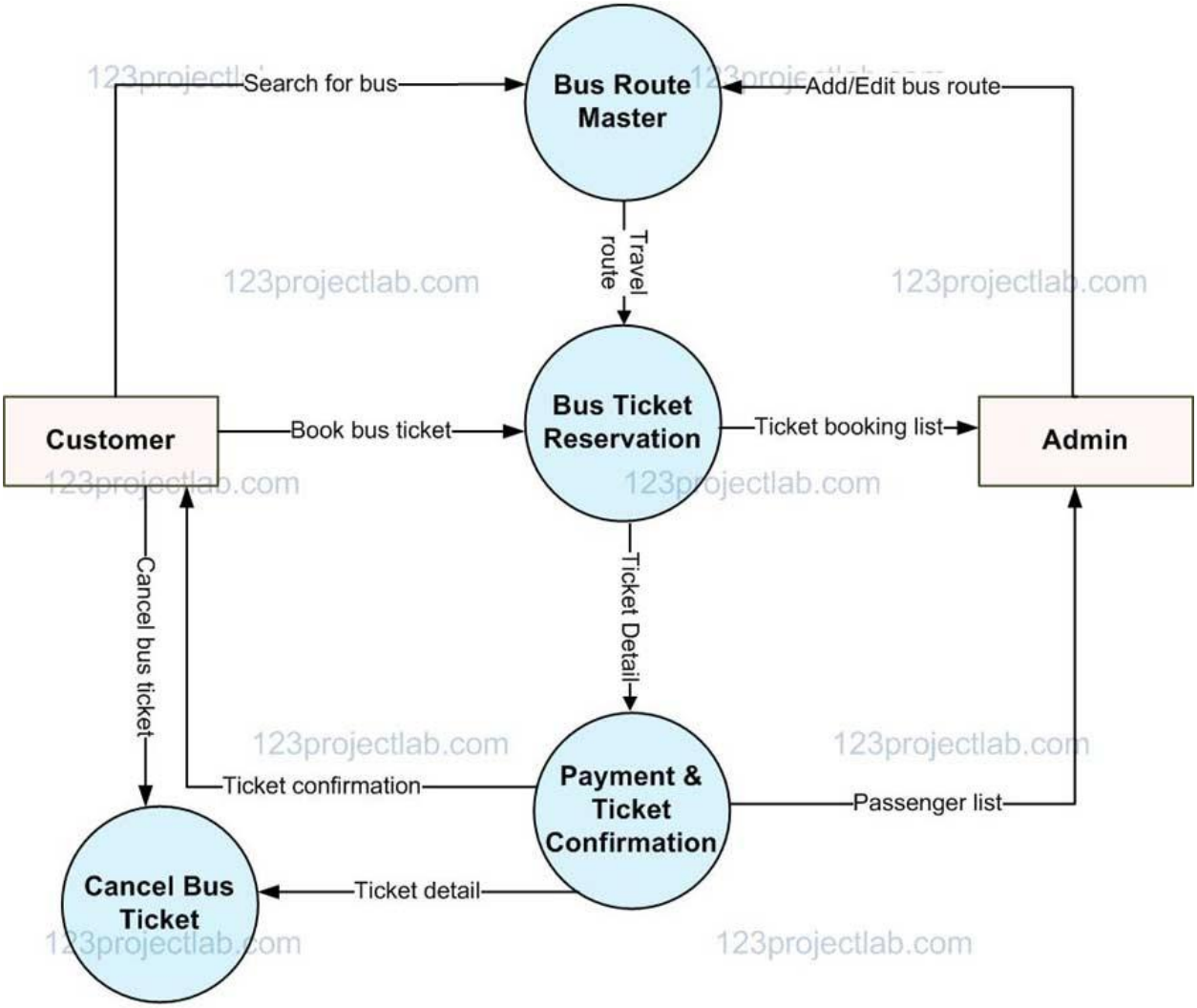
DFD Diagram in BUS RESERVATION SYSTEM

➢ Zero Level:-



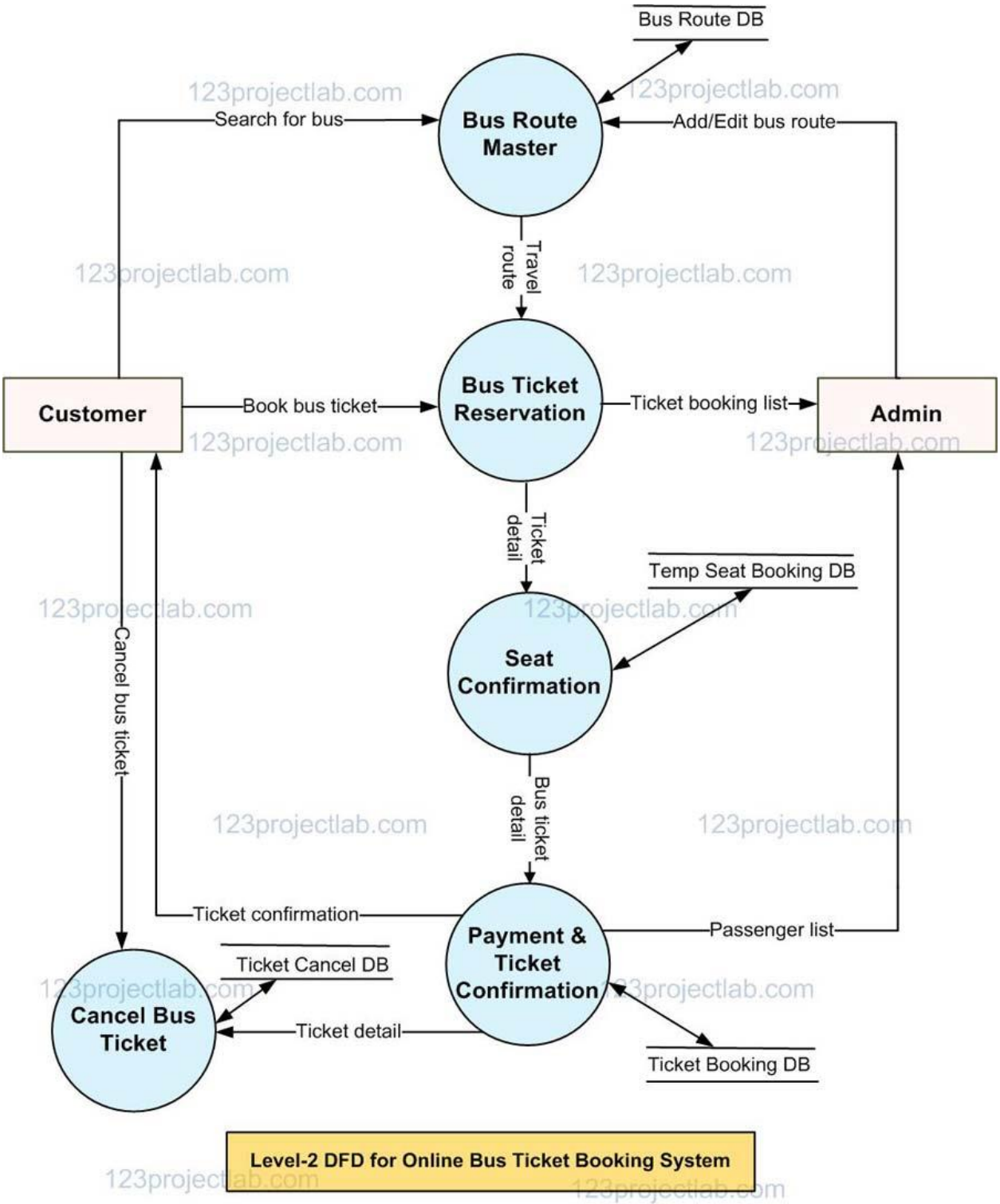
Context Level DFD for Online Bus Ticket Booking System

➢ First Level DFD:-

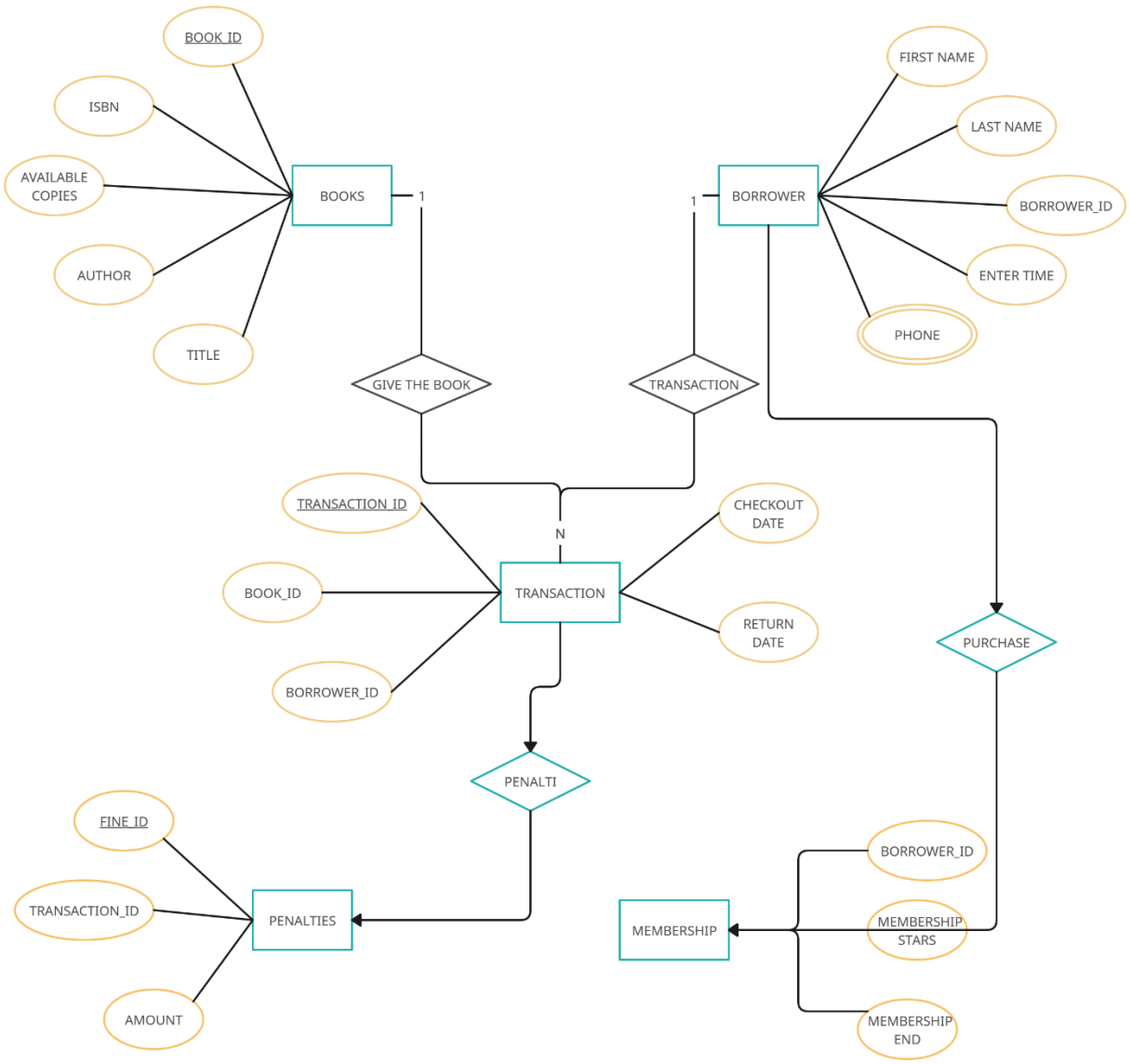


Level-1 DFD for Online Bus Ticket Booking System

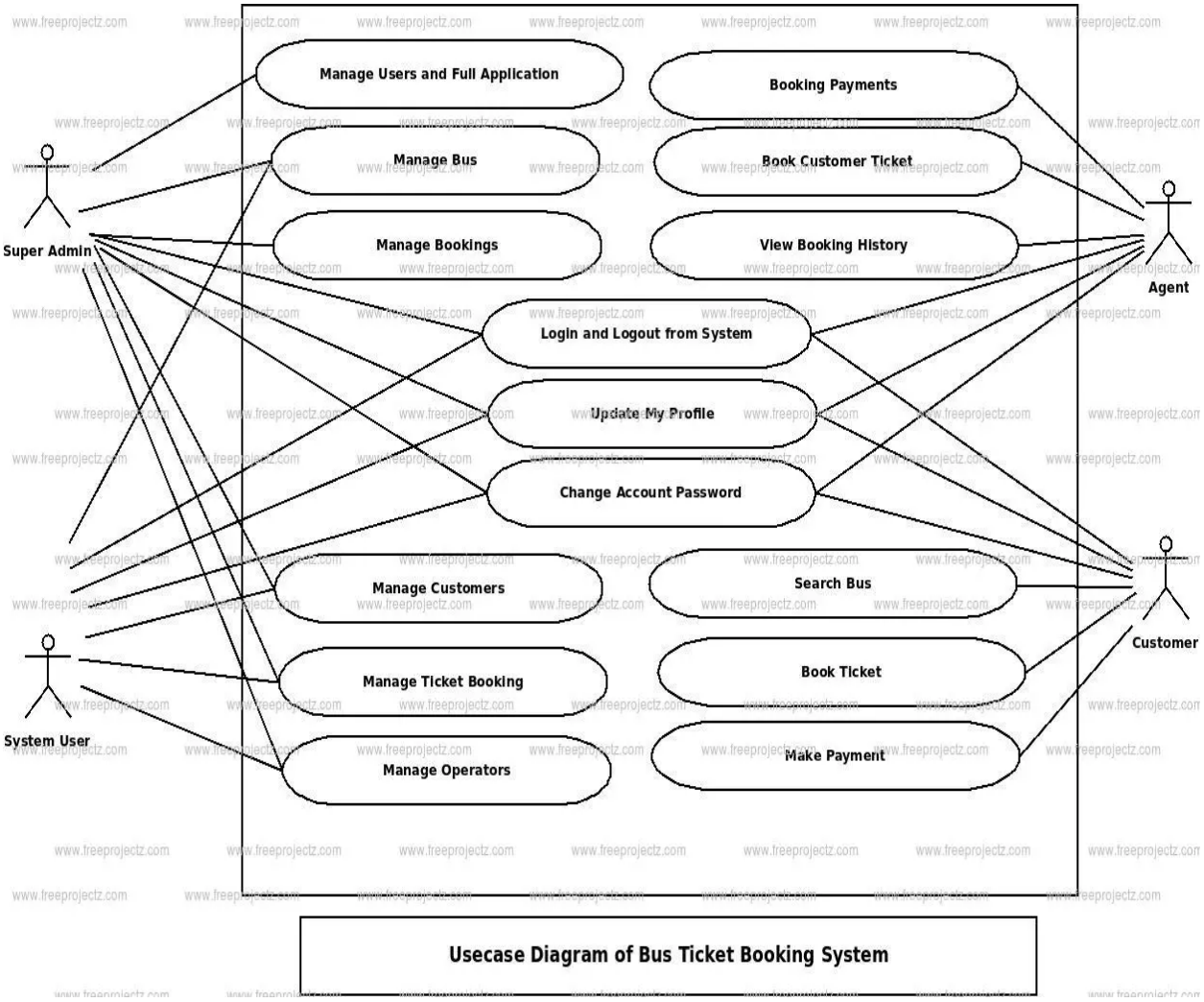
➢ Second Level DFD:-



ER Diagram in BUS RESERVATION SYSTEM

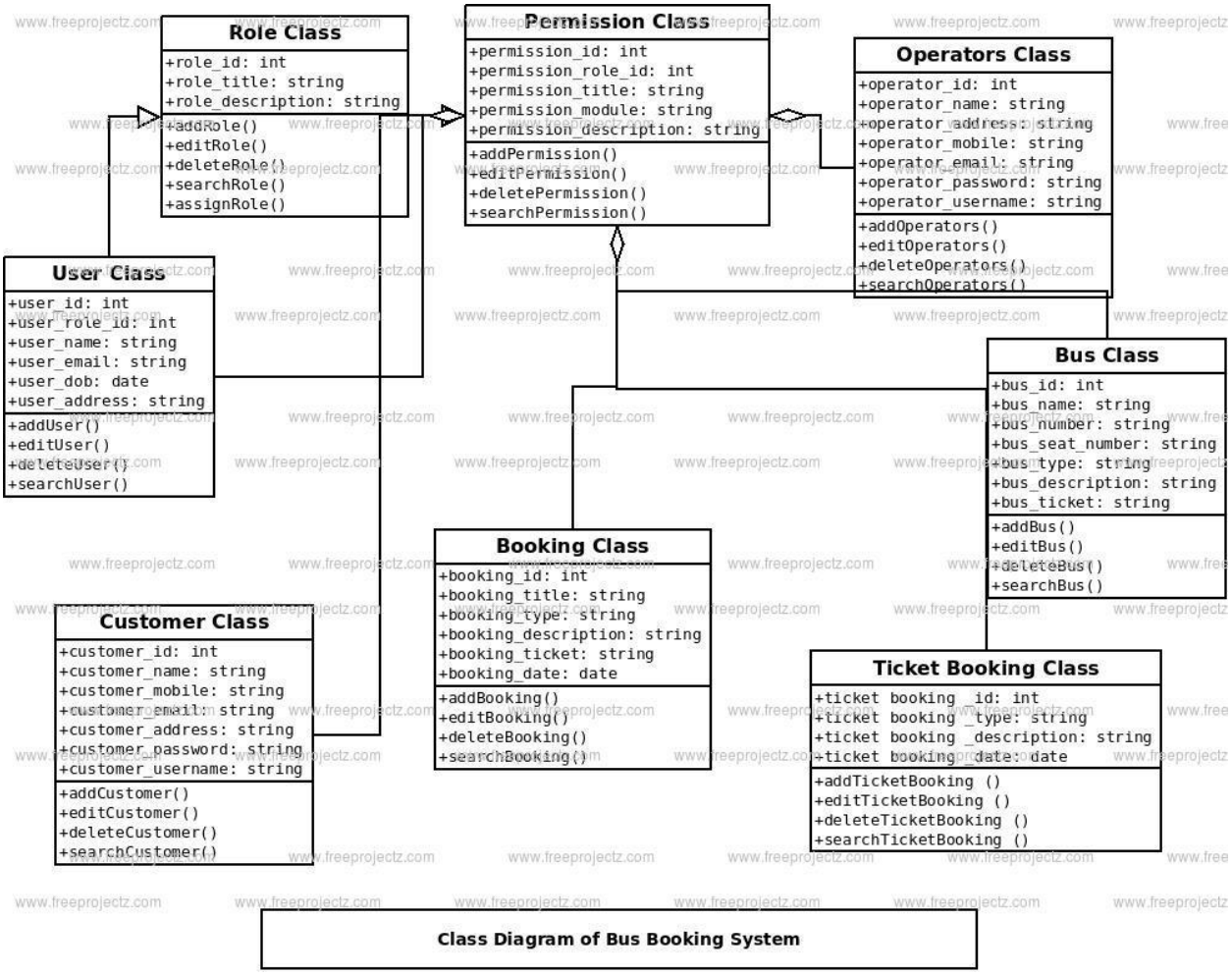


Use Case Diagram in BUS RESERVATION SYSTEM



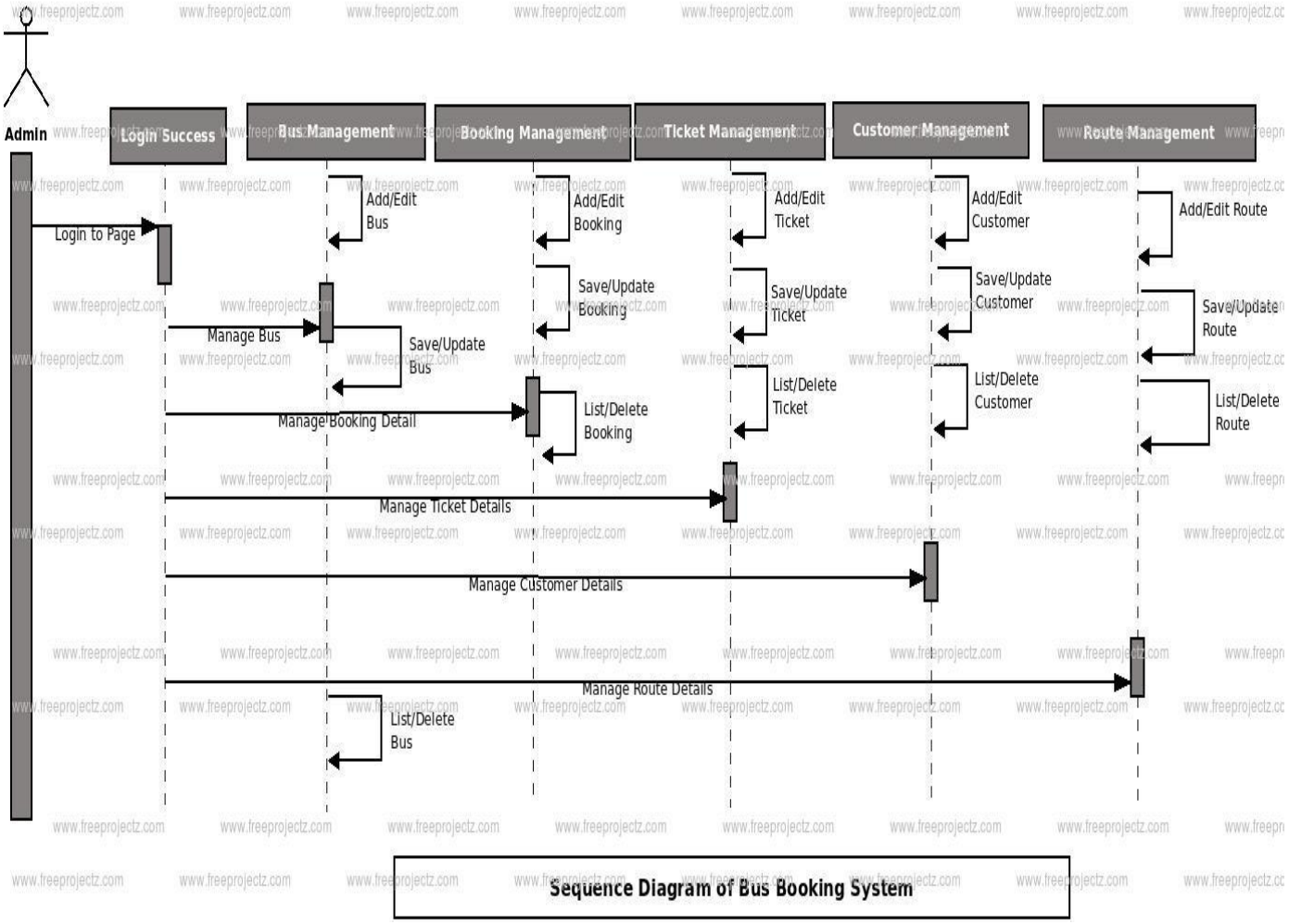
Usecase Diagram of Bus Ticket Booking System

Class Diagram in BUS RESERVATION SYSTEM

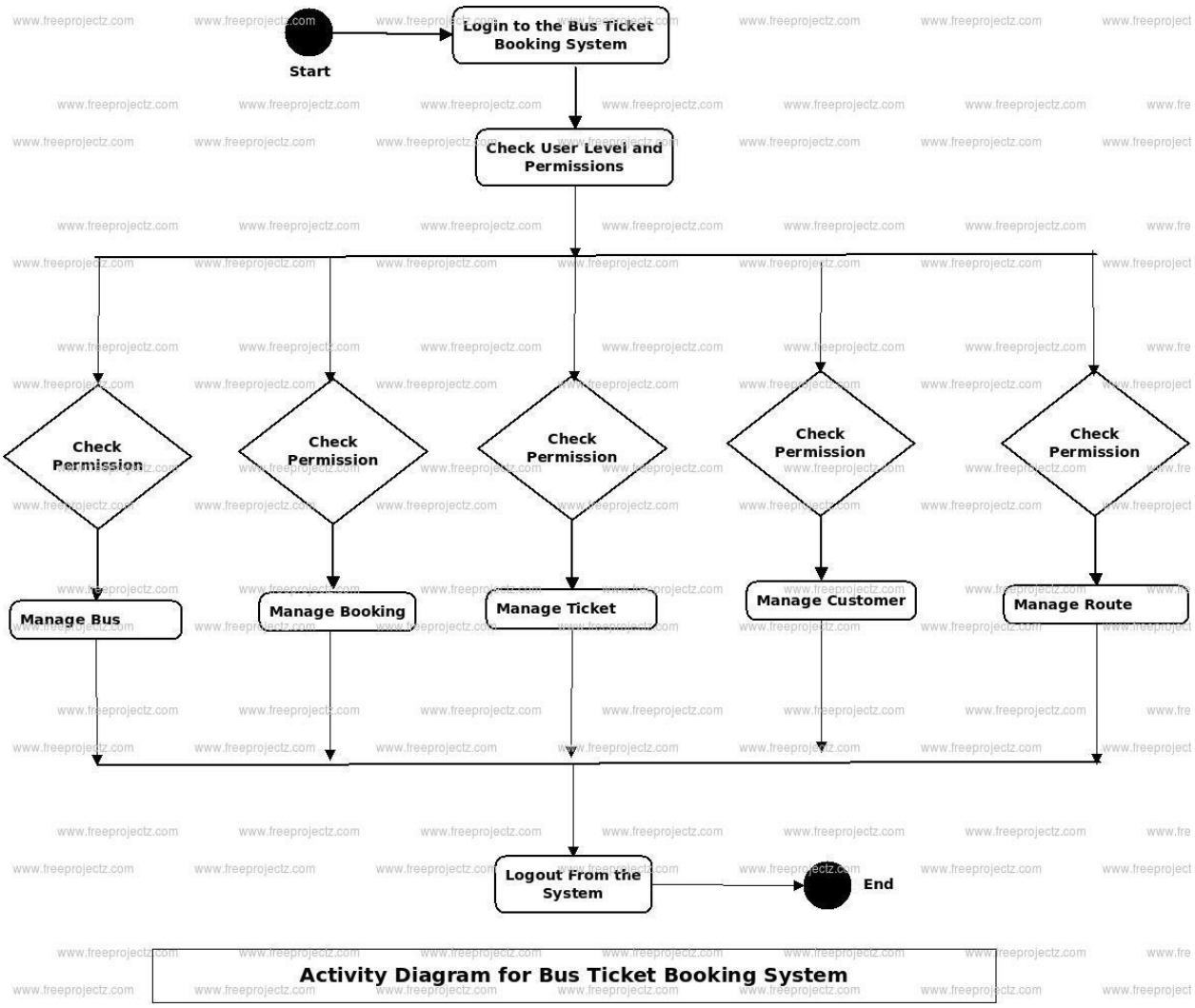


Class Diagram of Bus Booking System

Sequence Diagram in BUS RESERVATION SYSTEM



Activity Diagram in BUS RESERVATION SYSTEM



Activity Diagram for Bus Ticket Booking System

Evaluate size of the project using Function point metric for the BUS RESERVATION SYSTEM

Function Point (FP) analysis is a method to measure the size of a software project based on its functionality from the user's perspective. To evaluate the size of a bus reservation system using Function Point metrics, follow these steps:

Identify and classify functions:

External Inputs (EI): These are user interactions where data is input into the system.

External Outputs (EO): These are user interactions where data is output from the system.

External Inquiries (EQ): These are user interactions where data is both input and output from the system.

Internal Logical Files (ILF): These are logical groups of data maintained within the system.

External Interface Files (EIF): These are logical groups of data used for reference purposes but maintained by another system.

Assign complexity to each function:

Each function (EI, EO, EQ, ILF, EIF) is classified as low, average, or high complexity based on criteria like the number of data elements involved or the complexity of the logic.

Calculate unadjusted function points (UFP):

Use the standard weights to assign points to each function based on its type and complexity.

Example: Bus Reservation System

1. Identify and classify functions

External Inputs (EI):

User login (low)

Create reservation (average)

Update reservation (average)

Cancel reservation (average)

Add bus schedule (high)

Update bus schedule (high)

Remove bus schedule (high)

External Outputs (EO):

Reservation confirmation (average)

Reservation details report (high)

Schedule report (high)

External Inquiries (EQ):

Search bus schedule (average)

Check reservation status (average)

Internal Logical Files (ILF):

User data (average)

Reservation data (high)

Bus schedule data (high)

Payment transactions (high)

External Interface Files (EIF):

External payment gateway (average)

External user verification service (low)

2. Assign complexity to each function

Let's classify the complexity as follows:

Low (EI, EO, EQ): 3, 4, 3

Average (EI, EO, EQ): 4, 5, 4

High (EI, EO, EQ): 6, 7, 6

Low (ILF, EIF): 7, 5

Average (ILF, EIF): 10, 7

High (ILF, EIF): 15, 10

3. Calculate unadjusted function points (UFP)

Let's count the function points:

External Inputs (EI):

User login (low): 1 * 3 = 3

Create reservation (average): 1 * 4 = 4

Update reservation (average): 1 * 4 = 4

Cancel reservation (average): 1 * 4 = 4

Add bus schedule (high): 1 * 6 = 6

Update bus schedule (high): 1 * 6 = 6

Remove bus schedule (high): 1 * 6 = 6

Total for EI = 3 + 4 + 4 + 4 + 6 + 6 + 6 = 33

External Outputs (EO):

Reservation confirmation (average): 1 * 5 = 5

Reservation details report (high): 1 * 7 = 7

Schedule report (high): 1 * 7 = 7

Total for EO = 5 + 7 + 7 = 19

External Inquiries (EQ):

Search bus schedule (average): 1 * 4 = 4

Check reservation status (average): 1 * 4 = 4

Total for EQ = 4 + 4 = 8

Internal Logical Files (ILF):

User data (average): 1 * 10 = 10

Reservation data (high): 1 * 15 = 15

Bus schedule data (high): 1 * 15 = 15

Payment transactions (high): 1 * 15 = 15

Total for ILF = 10 + 15 + 15 + 15 = 55

External Interface Files (EIF):

External payment gateway (average): 1 * 7 = 7

External user verification service (low): 1 * 5 = 5

Total for EIF = 7 + 5 = 12

Unadjusted Function Point Total

Add all the totals:

UFP = 33 (EI) + 19 (EO) + 8 (EQ) + 55 (ILF) + 12 (EIF) = 127

Adjusted Function Points (AFP)

Function Points can be adjusted using a Value Adjustment Factor (VAF), which is determined by evaluating 14 General System Characteristics (GSCs) such as data communications, performance, heavily used configuration, transaction rates, etc.

Assume a typical VAF in many systems ranges from 0.65 to 1.35. Let's assume our system has a VAF of 1.1.

Calculate Adjusted Function Points

AFP = UFP * VAF

AFP = 127 * 1.1 = 139.7d

Estimate cost of the project using COCOMO II approach for BUS RESERVATION SYSTEM

The Constructive Cost Model (COCOMO) is an algorithmic software estimation model developed by Barry W. Boehm. To estimate the cost of a bus reservation system project, we can use either COCOMO I or COCOMO II. Here, I'll illustrate the steps for both:

COCOMO I

COCOMO I has three levels of estimation:

1. **Basic**: Provides a rough estimate.
2. **Intermediate**: Includes additional factors that influence the software development effort.
3. **Detailed**: Provides more precise estimates considering all phases of software development.

Basic COCOMO

The basic COCOMO model is given by: Effort (Person-Months)=a ×(KDSI)b Effort (Person-Months)=a×(KDSI)b

Where:

- KDSIKDSI is the number of delivered source instructions in thousands.
- a a and b b are constants that depend on the project type (organic, semi-detached, embedded).

For a bus reservation system, which we assume to be a semi-detached project:

- a =3.0a=3.0
- b =1.12b=1.12

We first need to estimate the size of the project in KDSI. Based on historical data, we can assume that 1 Function Point (FP) translates to about 100 lines of code. With our previous calculation of 140 Function Points:

KDSI=140×100/1000=14KDSI=140×100/1000=14

Using the formula: Effort=3.0×(14)1.12=3.0×21.47≈64.41 Person-MonthsEffort=3.0×(14)1.12=3.0×21.47≈64.41 Person-Months

Intermediate COCOMO

Intermediate COCOMO adds 15 cost drivers that account for attributes of the project, personnel, and product.

The intermediate effort equation is: Effort=a ×(KDSI)b ×$E\ A\ F$ Effort=a×(KDSI)b×EAF

Where EAF is the Effort Adjustment Factor calculated from the cost drivers. Let's assume an average EAF of 1.1 (typical for many projects).

Effort=3.0×(14)1.12×1.1=3.0×21.47×1.1≈70.85 Person-MonthsEffort=3.0×(14)1.12×1.1=3.0×21.47×1.1≈70.85 Person-Months

Cost Estimation

Assuming an average monthly salary of $8,000 for a developer:
Cost=70.85×8000=$566,800Cost=70.85×8000=$566,800

COCOMO II

COCOMO II is more detailed and includes models for early estimation (Application Composition Model), post-architecture estimation (Post-Architecture Model), and reused software.

We will use the Post-Architecture Model for this estimate.

Steps:

1. **Calculate the size (in KSLOC)**
    - KSLOC=14KSLOC=14
2. **Determine Scale Factors (SF)**
    - SFs influence the exponent of the effort equation and include factors like Precedentedness (PREC), Development Flexibility (FLEX), Architecture/Risk Resolution (RESL), Team Cohesion (TEAM), and Process Maturity (PMAT).

Assume moderate values for SFs:

- PREC: 3.72
- FLEX: 3.04
- RESL: 4.24
- TEAM: 4.12
- PMAT: 4.52

Summing these up and adding 100:
Scale=100+(3.72+3.04+4.24+4.12+4.52)=119.64Scale=100+(3.72+3.04+4.24+4.12+4.52)=119.64

Exponent b b:
b =0.91+0.01×(Scale)=0.91+0.01×19.64=1.1064b=0.91+0.01×(Scale)=0.91+0.01×19.64=1.1064

3. **Effort Multiplier (EM)**
    - Determine values for 17 cost drivers. Assume an overall average EM of 1.15.
4. **Calculate Effort**

Effort (Person-Months)=2.94×(KSLOC)b ×EMEffort (Person-Months)=2.94×(KSLOC)b×EM

Effort=2.94×(14)1.1064×1.15Effort=2.94×(14)1.1064×1.15

Effort=2.94×22.64×1.15≈76.62 Person-MonthsEffort=2.94×22.64×1.15≈76.62 Person-Months

Cost Estimation

Assuming the same average monthly salary: Cost=76.62×8000=$612,960Cost=76.62×8000=$612,960

Conclusion

Using COCOMO I (Intermediate):

- Estimated Effort: 70.85 Person-Months
- Estimated Cost: $566,800

Using COCOMO II:

- Estimated Effort: 76.62 Person-Months
- Estimated Cost: $612,960