

TERM PAPER
OF
OBJECT ORIENTED
PROGRAMMING

COURSE CODE: CSE 202

RAILWAY RESERVATION
SYSTEM

CONTENTS

PAGE NO:

1. **INTRODUCTION**
4
2. **PROPOSED SYSTEM**
5
3. **SYSTEM DEVELOPMENT LIFE CYCLE**
6
4. **SOURCE CODE**
18
5. **TESTING**
37
 - **WHITE BOX TESTING**
 - **BLACK BOX TESTING**
6. **CODE COMPLETENESS EVALUATION**
39
7. **DATA FLOW DIAGRAM**
40
8. **ADVANTAGES OF THE SYSTEM**
42
9. **HARDWARE/SOFTWARE REQUIREMENTS**
43

10. **BIBLIOGRAPHY**
45

INTRODUCTION



This project introduces railway reservation system . It explains how reservation is being done in Indian Railways . The step by step procedure is explained . This project is developed in c ++ language . Almost all the header files have been used in this project. Proper comments have been given at desired locations to make the project user friendly. Various functions and structures are used to make a complete use of this language.

Thos project is well versed with the programming . Railway reservation can easily accompanied with the help of this.

PROPOSED SYSTEM

Today one cannot afford to rely on the fallible human beings of be really wants to stand against today's merciless competition where not to wise saying "to err is human" no longer valid, it's outdated to rationalize your mistake. So, to keep pace with time, to bring about the best result without malfunctioning and greater efficiency so to replace the unending heaps of files with a much sophisticated hard disk of the computer. One has to use the data management software. Software has been an ascent in atomization various organizations. Many software products working are now in markets, which have helped in making the organizations work easier and efficiently. Data management initially had to maintain a lot of ledgers and a lot of paper work has to be done but now software product on this organization has made their work faster and easier. Now only this software has to be loaded on the computer and work can be done. This prevents a lot of time and money. The work becomes fully automated and any information regarding the organization can be obtained by clicking the button. Moreover, now it's an age of computers of and automating such an organization gives the better look.

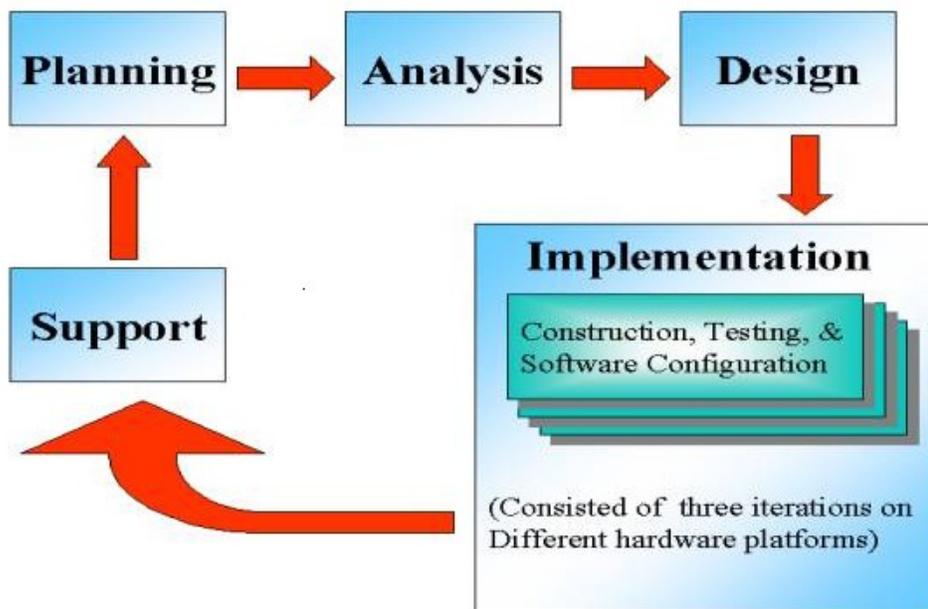
SYSTEM DEVELOPMENT LIFE CYCLE

The systems development life cycle is a project management technique that divides complex projects into smaller, more easily managed segments or phases. Segmenting projects allows managers to verify the successful completion of project phases before allocating resources to subsequent phases.

Software development projects typically include initiation, planning, design, development, testing, implementation, and maintenance phases. However, the phases may be divided differently depending on the organization involved. For example, initial project activities might be designated as request, requirements-definition, and planning phases, or initiation, concept-development, and planning phases. End users of the system

under development should be involved in reviewing the output of each phase to ensure the system is being built to deliver the needed functionality.

The System Development Life Cycle as Used in the Construction of the Server Appliance



PHASES OF SYSTEM DEVELOPMENT LIFE CYCLE



Initiation Phase

The Initiation Phase begins when a business sponsor identifies a need or an opportunity. The purpose of the Initiation Phase is to:

- **Identify and validate an opportunity to improve business accomplishments of the organization or a deficiency related to a business need.**
- **Identify significant assumptions and constraints on solutions to that need.**
- **Recommend the exploration of alternative concepts and methods to satisfy the need including questioning the need for technology, i.e., will a change in the business process offer a solution?**
- **Assure executive business and executive technical sponsorship.**

The Sponsor designates a Project Manager and the business need is documented in a Concept Proposal. The Concept Proposal includes information about the business process and the relationship to the Agency/Organization Infrastructure and the Strategic Plan. A successful Concept Proposal results in a Project Management Charter which outlines the authority of the project manager to begin the project.

Careful oversight is required to ensure projects support strategic business objectives and resources are effectively implemented into an organization's enterprise architecture. The initiation phase begins when an opportunity to add, improve, or correct a system is identified and formally requested through the presentation of a business case. The business case should, at a minimum, describe a proposal's purpose, identify expected benefits, and explain how the proposed system supports one of the organization's business strategies. The business case should also identify alternative solutions and detail as many informational, functional, and network requirements as possible.



System Concept Development Phase

The System Concept Development Phase begins after a business need or opportunity is validated by the Agency/Organization Program Leadership and the Agency/Organization CIO.

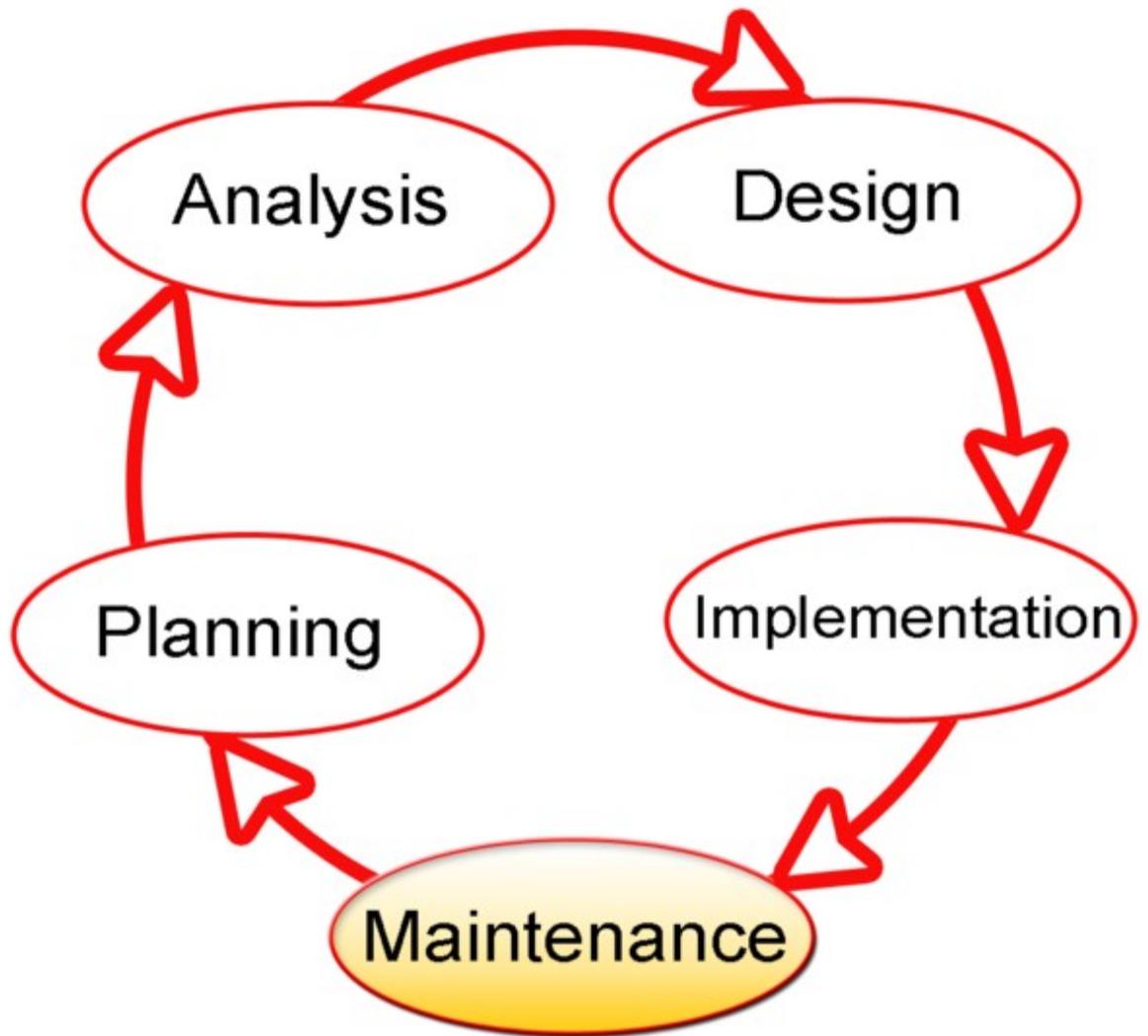
The purpose of the System Concept Development Phase is to:

- **Determine the feasibility and appropriateness of the alternatives.**
- **Identify system interfaces.**
- **Identify basic functional and data requirements to satisfy the business need.**
- **Establish system boundaries, identify goals, objectives, critical success factors, and performance measures.**
- **Evaluate costs and benefits of alternative approaches to satisfy the basic functional requirements**
- **Assess project risks**
- **Identify and initiate risk mitigation actions, and**
- **Develop high-level technical architecture, process models, data models, and a concept of operations.**

This phase explores potential technical solutions within the context of the business need. It may include several trade-off decisions such as the decision to use COTS software products as opposed to developing custom software or reusing software components, or the decision to use an incremental delivery versus a complete, one-time deployment. Construction of executable prototypes is encouraged to evaluate technology to support the business process.

The System Boundary Document serves as an important reference document to support the Information Technology Project Request (ITPR) process. The ITPR must be approved by the State CIO before the project can move forward.

PICTORIAL REPRESENTATION OF SDLC:-



Planning Phase

The planning phase is the most critical step in completing development, acquisition, and maintenance projects. Careful planning, particularly in the early stages of a project, is necessary to coordinate activities and manage project risks effectively. The depth and formality of project plans should be commensurate with the characteristics and risks of a given project.

Project plans refine the information gathered during the initiation phase by further identifying the specific activities and resources required to complete a project. A critical part of a project manager's job is to coordinate discussions between user, audit, security, design, development, and network personnel to identify and document as many functional, security, and network requirements as possible.

During this phase, a plan is developed that documents the approach to be used and includes a discussion of methods, tools, tasks, resources, project schedules, and user input. Personnel assignments, costs, project schedule, and target dates are established. A Project Management Plan is created with components related to acquisition planning, configuration management planning, quality assurance planning, concept of operations, system security, verification and

**validation, and systems engineering
management planning.**



Requirements

Analysis Phase

This phase formally defines the detailed functional user requirements using high-level requirements identified in the Initiation, System Concept, and Planning phases. It also delineates the requirements in terms of data, system performance, security, and maintainability requirements for the system. The requirements are defined in this phase to a level of detail sufficient for systems design to proceed. They need to be measurable, testable, and relate to the business need or opportunity identified in the Initiation Phase. The requirements that will be used to determine acceptance of the system are captured in the Test and Evaluation Master Plan.

The purposes of this phase are to:

- Further define and refine the functional and data requirements and document them in the Requirements Document,**
- Complete business process reengineering of the functions to be supported (i.e., verify what information drives the business process, what information is generated,**

- who generates it, where does the information go, and who processes it),
- **Develop detailed data and process models (system inputs, outputs, and the process.**
 - **Develop the test and evaluation requirements that will be used to determine acceptable system performance.**



Design Phase

The design phase involves converting the informational, functional, and network requirements identified during the initiation and planning phases into unified design specifications that developers use to script programs during the development phase. Program designs are constructed in various ways. Using a top-down approach, designers first identify and link major program components and interfaces, then expand design layouts as they identify and link smaller subsystems and connections. Using a bottom-up approach, designers first identify and link minor program components and interfaces, then expand design layouts as they identify and link larger systems and

connections.

Contemporary design techniques often use prototyping tools that build mock-up designs of items such as application screens, database layouts, and system architectures. End users, designers, developers, database managers, and network administrators should review and refine the prototyped designs in an iterative process until they agree on an acceptable design. Audit, security, and quality assurance personnel should be involved in the review and approval process.

During this phase, the system is designed to satisfy the functional requirements identified in the previous phase. Since problems in the design phase could be very expensive to solve in the later stage of the software development, a variety of elements are considered in the design to mitigate risk. These include:

- Identifying potential risks and defining mitigating design features.**
- Performing a security risk assessment.**
- Developing a conversion plan to migrate current data to the new system.**
- Determining the operating environment.**
- Defining major subsystems and their inputs and outputs.**
- Allocating processes to resources.**
- Preparing detailed logic specifications for each software module.**

The result is a draft System Design Document which captures the preliminary design for the system. Everything requiring user input or approval is documented and reviewed by the user.

Once these documents have been approved by the Agency CIO and Business Sponsor, the final System Design Document is created to serve as the Critical/Detailed Design for the system. This document receives a rigorous review by Agency technical and functional representatives to ensure that it satisfies the business requirements.

Concurrent with the development of the system design, the Agency Project Manager begins development of the Implementation Plan, Operations and Maintenance Manual, and the Training Plan.



Phase

Development

The development phase involves converting design specifications into executable programs. Effective development standards include requirements that programmers and

other project participants discuss design specifications before programming begins. The procedures help ensure programmers clearly understand program designs and functional requirements.

Programmers use various techniques to develop computer programs. The large transaction-oriented programs associated with financial institutions have traditionally been developed using procedural programming techniques. Procedural programming involves the line-by-line scripting of logical instructions that are combined to form a program.

Effective completion of the previous stages is a key factor in the success of the Development phase. The Development phase consists of:

- Translating the detailed requirements and design into system components.**
- Testing individual elements (units) for usability.**
- Preparing for integration and testing of the IT system.**



Integration and Test Phase

Subsystem integration, system, security, and user acceptance testing is conducted during the integration and test phase. The user, with those responsible for quality assurance, validates that the functional requirements, as defined in the functional requirements document, are satisfied by the developed or modified system. OIT Security staff assess the system security and issue a security certification and accreditation prior to installation/implementation. Multiple levels of testing are performed, including:

- **Testing at the development facility by the contractor and possibly supported by end users**
- **Testing as a deployed system with end users working together with contract personnel**
- **Operational testing by the end user alone performing all functions.**

Requirements are traced throughout testing, a final Independent Verification & Validation evaluation is performed and all documentation is reviewed

and accepted prior to acceptance of the system.



Implementation **Phase**

This phase is initiated after the system has been tested and accepted by the user. In this phase, the system is installed to support the intended business functions. System performance is compared to performance objectives established during the planning phase. Implementation includes user notification, user training, installation of hardware, installation of software onto production computers, and integration of the system into daily work processes.

This phase continues until the system is operating in production in accordance with the defined user requirements.



Operations and Maintenance Phase

The system operation is ongoing. The system is monitored for continued performance in accordance with user requirements and needed system modifications are incorporated. Operations continue as long as the system can be effectively adapted to respond to the organization's needs. When modifications or changes are identified, the system may reenter the planning phase. The purpose of this phase is to:

- Operate, maintain, and enhance the system.
- Certify that the system can process sensitive information.
- Conduct periodic assessments of the system to ensure the functional requirements continue to be satisfied.
- Determine when the system needs to be modernized, replaced, or retired.

SOURCE CODE

```
#include<fstream.h>
#include<process.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<dos.h>
void res_menu();
void help();
void colyb()
{
textcolor(YELLOW);
textbackground(BLUE);}
void coldes(int a,int b)
{
textcolor(a);
```

```

textbackground(b);}
void s(int i,int j)
{
    for(int a=i;a>0;a-=j)sound (a);
}
void saund(int j)
{
    j=0;
    for(int i=20;i<20000;i+=1000)
    {sound(i);delay(j);nosound();}
}
class address
{
    protected :
    char hno[10],city[30],post[30],colony[30],pin[10];
    public :

    void address_inp()
    {
        cout<<"House No:";
        gets(hno);
        cout<<"Colony:";
        gets(colony);
        cout<<"Post:";
        gets(post);
        cout<<"City:";
        gets(city);
        cout<<"Pin Code No:";
        gets(pin);
    }
    void address_out()
    {
        cout<< hno<<" , "<<colony<<" , "<<post<<" , "
        <<city<<" - "<<pin;
    }
}

```

```

};

class p_det
{
    protected :
    address add;
    float age;
    char name[40];
    char sex;
    public:

    void p_det_inp()
    {
        colyb();
        cout<<"Enter Name :";
        gets(name);
        cout<<"
Enter address      ";
        add.address_inp();
        cout<<"
Enter age (Years.Months) :
";cin>>age;
        cout<<"
Enter sex group (M / F) :
";cin>>sex;

    }

    void p_det_out()
    {
        colyb();
        cout<<"
PERSONAL DETAILS";
        cout<<"
Name              :
";puts(name);
        cout<<"
Adress            :      ";

```

```

        add.address_out();
        cout<<"
Age (Years.Months) :      ";cout<<age;
        cout<<"
Sex group (M/F)   :      ";cout<<sex;
        delay(500);
    }

void assign_name(p_det p,char str[]);
};
void p_det :: assign_name(p_det p,char str[])
{
    strcpy(str,p.name);
}
class dat{

    int dd,mm,yy;
    public :
    void dat_inp()
    {
        h:
        cout<<"          date :
dd : ";cin>>dd;
        cout<<"          mm : ";cin>>mm;
        if((dd>31) || (mm>12))
        {
            cout<<"the date is invalid";
            goto h;
        }

        cout<<"          yy : ";cin>>yy;

    }
    void dat_out()

```



```
strcpy(from,"LUCKNOW");strcpy(to,"DELHI");strcpy(via,"ÄÄ");
    fare=100;
    }
    else if(strcmp(fcode,"0037")==0)
    {
```

```
strcpy(from,"LUCKNOW");strcpy(to,"CHENNAI");strcpy(via,"D
ELHI");
    fare=700;
    }
    else if(strcmp(fcode,"1070")==0)
    {
```

```
strcpy(from,"DELHI");strcpy(to,"COCHIN");strcpy(via,"ÄÄ");
    fare=730;
    }
    else if(strcmp(fcode,"1007")==0)
    {
```

```
strcpy(from,"DELHI");strcpy(to,"BANGALORE");strcpy(via,"ÄÄ
");
    fare=800;
    }
    else if(strcmp(fcode,"1017")==0)
    {
```

```
strcpy(from,"LUCKNOW");strcpy(to,"BANGALORE");strcpy(via
,"DELHI");
    fare=850;
    }
    else if(strcmp(fcode,"7001")==0 )
    {
```



```

    cout<<"
enter the code ";
    gets(code);
    cout<<"
Checking for record";
    fin.open("oldmas.dat",ios::binary | ios::in);
    fin1.open("newmas.dat",ios::binary | ios::out);
    if(fin==NULL && fin1==NULL)
    {
        cout<<" No records found in Reservation Master";getch();
        goto label;
    }
    else
    {
        fin.read( (char*)&r,sizeof(res) );
        while(!fin.eof() )
        {

            assign_code(r,code1);
            r.p.assign_name(r.p,name1);// cout<<"
name 1 & code1 :
";puts(name1);puts(code1);getch();
            if(strcmpi(name1,name)==0 &&
strcmp(code1,code)==0)
            {
                cout<<"
record found ";
                cout<<"
Deleting.a.a.a.a.";
            }
            else
            {
                fin1.write( (char*)&r ,sizeof(res));
            }
            fin.read( (char*)&r,sizeof(res) );
        }
    }

```



```
}  
  
//          END OF PROGRAMME  
  
//#####  
#####
```

Testing

Software Testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test^[1], with respect to the context in which it is intended to operate. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. It can also be stated as the process of validating and verifying that a software

program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics.

Software Testing, depending on the testing method employed, can be implemented at any time in the development process, however the most test effort is employed after the requirements have been defined and coding process has been completed.

Testing methods

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

Black box testing

Black box testing treats the software as a "black box," without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

Specification-based testing

Specification-based testing aims to test the functionality of software according to the applicable requirements.^[16] Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case.

Specification-based testing is necessary, but it is insufficient to guard against certain risks

Advantages and disadvantages

The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code *must* have bugs. Using the principle, "Ask and you shall receive," black box testers find bugs where programmers don't. *But*, on the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed. That's why there are situations when (1) a black box tester writes many test cases to check something that can be tested by only one test case, and/or (2)

some parts of the back end are not tested at all.

Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.

White box testing

White box testing, by contrast to black box testing, is when the tester has access to the internal data structures and algorithms (and the code that implement these)

Types of white box testing :-

The following types of white box testing exist:

- **api testing - Testing of the application using Public and Private APIs.**
- **code coverage - creating tests to satisfy some criteria of code coverage. For example, the test designer can create tests to cause all statements in the program to be executed at least once.**
- **fault injection methods.**
- **mutation testing methods.**
- **static testing - White box testing includes all static testing.**

Code completeness evaluation

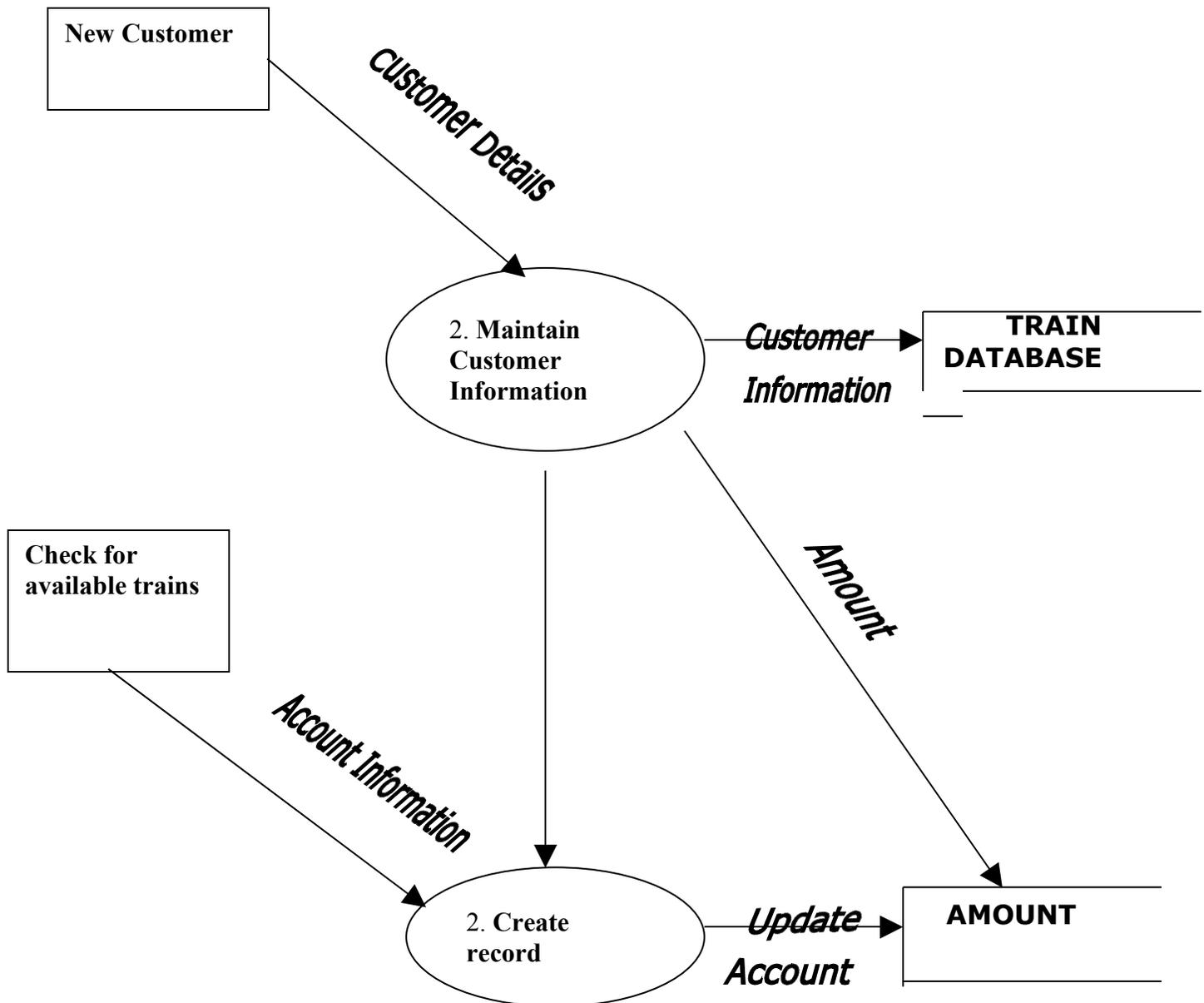
White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.^[19]

Two common forms of code coverage are:

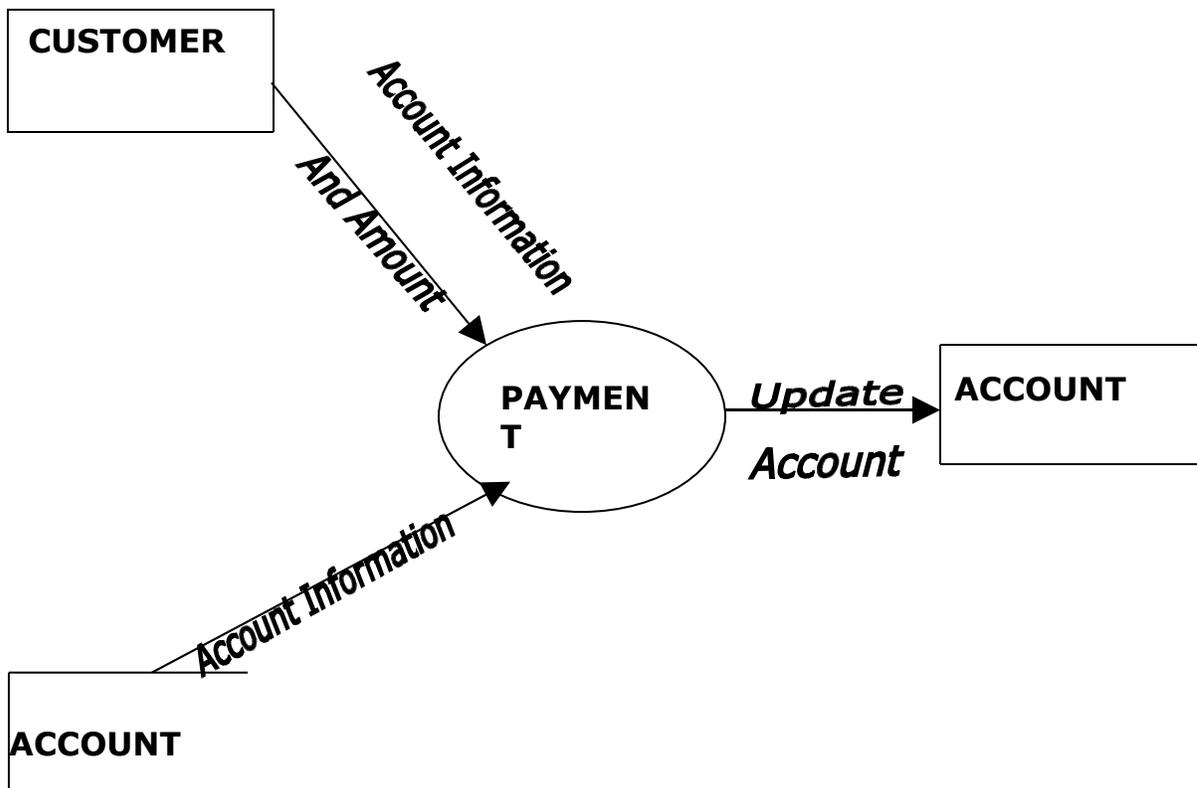
- *function coverage*, which reports on functions executed
- and *statement coverage*, which reports on the number of lines executed to complete the test.

They both return a coverage metric, measured as a percentage

DATA FLOW DIAGRAM:-



DFD FOR PAYMENT:-



ADVANTAGES OF RAILWAY RESERVATION SYSTEM

Now one can easily plan the journey comfortably as the process is efficient and fast with being easy to access. Reservations can be made through the Indian railways site or at the ample reservation centers all over the country. Also now there are authorized agencies which provide reservation facility on behalf of India railways and without waiting in long line one can easily book a ticket. The booking is done through an E-Ticket issue which have a PNR number of which one has to take a print and just have to show at the station.

It not only provide reservation but cancellation can also be done through this system at ease and one can use a credit card to complete the process.

This being a big step in terms of improvement in the railway system it is widely accepted across the country.

HARDWARE REQUIREMENTS:

I. OPERATING SYSTEM :

Any

II. PROCESSOR:

**1. PENTIUM (ANY) OR AMD
ATHALON (3800+- 4200+ DUAL
CORE)**

III. MOTHERBOARD:

**1. 845 OR 915, 995 FOR PENTIUM
OR MSI K9MM-V VIA
K8M800+8237R PLUS CHIPSET
FOR AMD ATHALON**

IV. RAM: 512MB+

V. Hard disk:**Sata 40 gb or above****VI. CD/DVD r/w multi drive combo****VII. FLOPPY DRIVE 1.44 MB****VIII. MONITOR 14.1 or 15 -17 inch****IX. Key board and mouse****X. Printer****SOFTWARE REQUIREMENTS:****I. office XP**

BIBLIOGRAPHY

These are the following links which assist me at each and every step in completing this project , without them i was not be able to finish this important project:-

1. www.scribd.com
2. www.wikipedia.com
3. www.answers.com
4. www.askme.com
5. www.esnips.com

Instead of these useful websites , following books also help me:

**1.OBJECT ORIENTED PROGRAMMING WITH
C++
BY E.BALAGURUSWAMI**

**2.OBJECT ORIENTED PROGRAMMING
BT ROBERT LAFORE**