

Guru Gobind Singh Indraprastha University

---

# Software Requirement Specification For Cab Booking System

Submitted To:

Mrs. Nishi Sharma (NIEC)

Submitted by:

Gaurav Deep Mahesh

05715604410

---

## ACKNOWLEDGEMENT

---

A study or a project of this volume can never be an outcome of a single person. We are indebted to our mentor Mrs. Nishi Sharma for being the epitome of guidance during the entire project.

We are also thankful to our parents for their Encouragement. Without their help this project would not have seen the light of day.

We are happy to present a vote of thanks to them for their sincere advice and co-operation that they have lent us unconditionally.

# CAB BOOKING SYSTEM

## Problem Statement:-

---

**Cab booking service is a major transport service provided by the various transport operators in a particular city. Mostly peoples use cab service for their daily transportations need. The company must be a registered and fulfils all the requirements and security standards set by the transport department.** Taxi Booking Software is a web based platform that allows your customers to book their taxi's and executive taxi's (such as Limousines) all online from the comfort of their own home or office. The platform should offer an administration interface where the taxi company can manage the content, and access all bookings and customer information. Usually the platform will include all the required functionality such as hosting, email accounts, updates, a domain name (the web address) and, most importantly, backups!

In a nutshell your Taxi Booking Software should be able to:

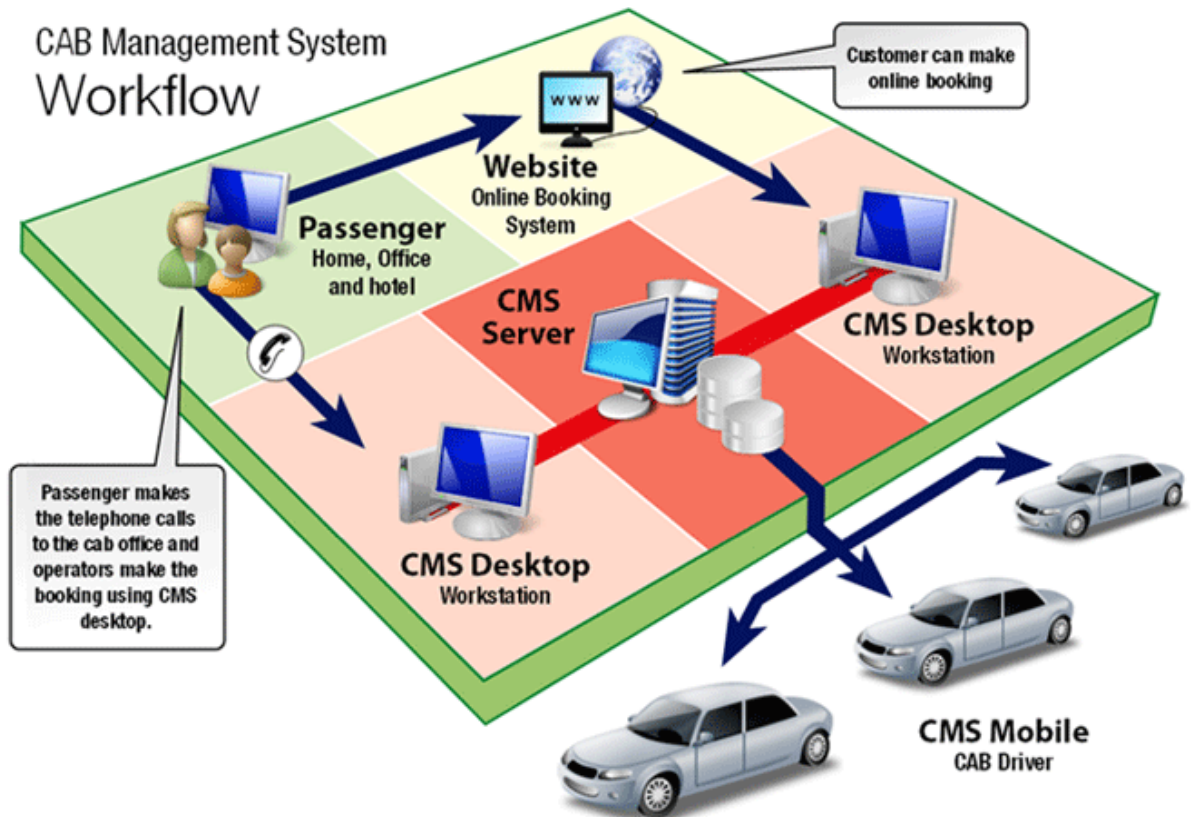
- o Provide the functionality to make your own bookings
- o Give your customers the facility to make payments and deposits online with their credit / debit card.
- o Generate Invoices
- o Update your web site without the need to get a web designer involved.
- o Provide the customer with taxi availability
- o Track your customers.
- o Engage your customers through interaction such as feedback forms

More and more Taxi companies are looking for integrated taxi booking systems as it makes life much easier for a) the customer - this is highly important and in today's internet age people should be able to book taxis online without having to pick up the phone and b) the taxi company - as all their bookings are now managed via an automated system which means they have an electronic record of future and historic bookings. From the historic data the taxi firm will be able to look at booking trends of set periods of time, and future bookings will allow them to budget their staff and taxi routes accordingly.

Taxi Booking Software can be built on top of a great looking web site and third party payment providers can be used to provide secure transactions. One factor that may be a stumbling block for Taxi firms is the cost of the taxi booking software. Some vendors offer a shared pricing model that allows certain taxi firms to share the cost of the system. This type of model is known as multi-vendor. As the taxi booking systems are web based this doesn't cause any issues as the software is located on a central web server and the web site front end can be branded in any way required by the taxi company. In today's technological world can your taxi company afford to be without booking software?

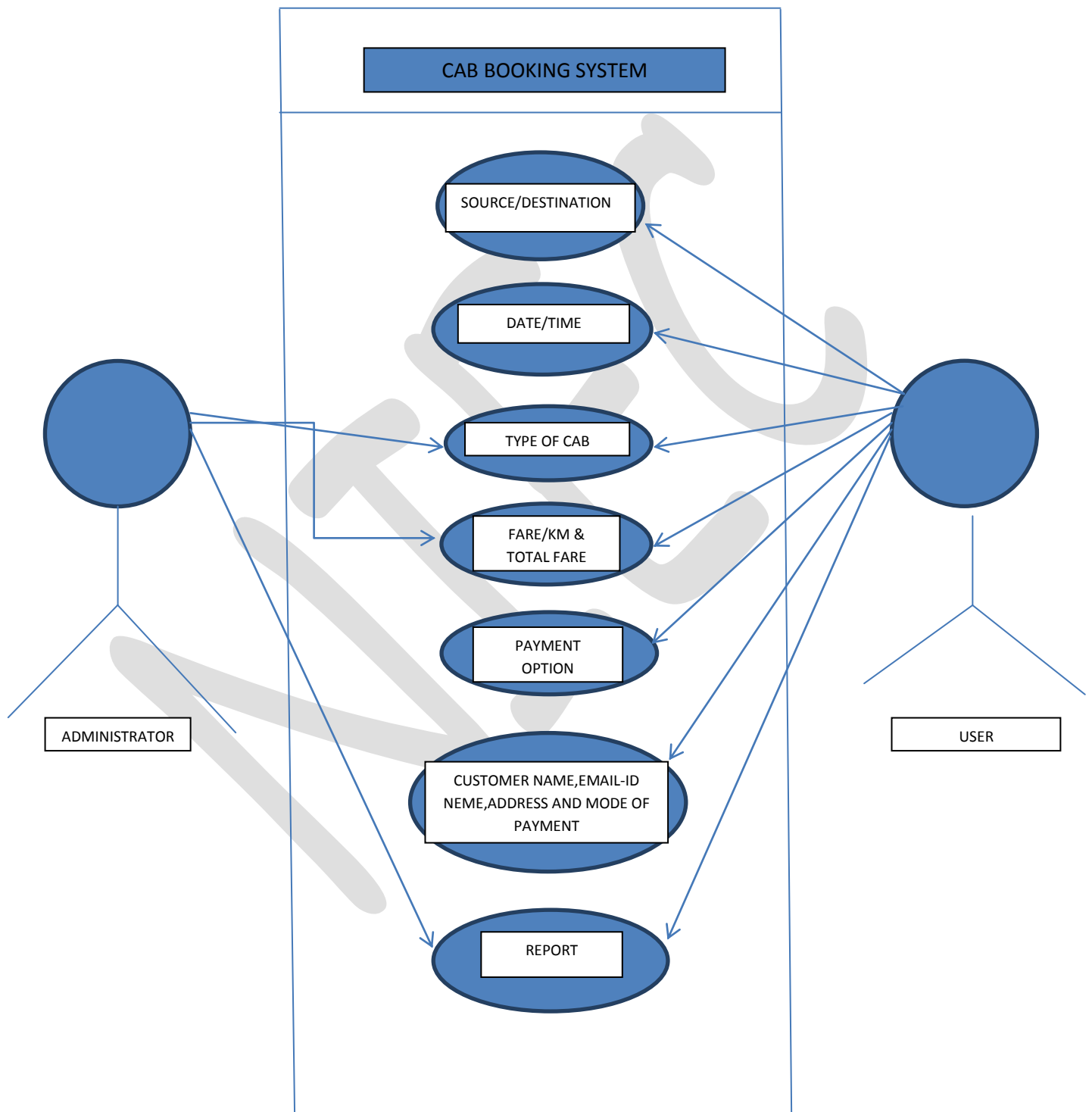
- The fare must be economical so that it must be in reach/budget of every person.
- Cab must be reach on time on the defined destination.
- There must be a large fleet of cabs (AC/Non AC).
- Provide the functionality to make your own bookings
- Update your web site without the need to get a web designer involved.
- Provide the customer with taxi availability.
- Track your customers.
- Engage your customers through interaction such as feedback forms
- Easy payment facility must be provided in cab i.e. by cash or by card.
- Payment bill must be provided by cab driver.
- Driver's identification data must be given i.e. driver's name, id & photograph at the time of booking of cab.
- Estimated time for a particular journey must be provided.
- Details of the route must be provided to the customer. Customers can my take the cab by his/her own route.
- Customer satisfaction is necessary.
- The user interface must be friendly so that the user can easily book a cab in few minutes by doing few clicks.
- Payment modes can be also of prepaid or post-paid.
- If the payment mode is prepaid then the customer have to provide its full name, address, type of card(visa, master, electron-visa etc.), account number, bank name, and branch.
- In both prepaid and post-paid customer's email id must be provided.
- At the time of booking the web page must have the interface for the starting point, destination, type of cab (AC/Non Ac), charge per kilometre, cab driver details, time, payment options, service area etc.
- Night charges are also added to the total fare on the basis of per kilometre.
- In the end the report must be generated which shows the customer name, address, source and destination, total fare, driver details and timings i.e. the invoice must be generated
- If the payment is made at the time of booking i.e. prepaid then it must be shown in the report.
- The information must be provided to the customer on its email id and to driver on its job sheet.

## CAB Management System Workflow



## Use - Case Diagram :-

---



## Use Case Template:-

---

### 1. **Source / Destination** :- Use case specification

1.1 Brief Description :- The main use of this use case is to provide the details about source and the destination of the user of the cab.

#### **Flow of Events :-**

#### 1.2 **Basic Flow** :-

1. User books a cab by providing the details of source and destination.
2. Booking clerk check the database.
3. On successful traveller makes the booking of the cab.

#### **1.3 Pre conditions :-**

The traveller should have a cab for a destination place.

#### **1.4 Post conditions :-**

The database must be modified after the booking transaction takes place.

### **2. Date / Time** :- Use case specification

#### **2.1 Brief Description :-**

The main purpose of this use case is to know details about the number of cabs available at that particular date and time.

#### **Flow of events :-**

#### **2.2 Basic Flow:-**

1. User checks the availability on website.

2. On successful completion of transaction user makes booking.

### **2.3 Pre condition**

The traveller should have a cab booked for destination place.

### **2.4 Post condition**

The booking database must be modified.

## **Questionnaire :-**

---

The aim of the questionnaire is to collect information on the methods and practices for software development. Your input is extremely important to identify gaps and desired data that will help in driving future tools and methods that may better support your activity. An online version of the questionnaire is also published on the web portal.

**The information collected is completely confidential.**

**We shall never share or release your private information to an unaffiliated third party.**

1. Please specify the type of your institution/company:

☐ Research      ☐ Industry      ☐ Higher Education      ☐ Public body  
Other \_\_\_\_\_

2. Please classify the application area of your institution/company:

☐ Education      ☐ Consumer appliances production      ☐ Museum      ☐  
Mechanical Engineering      ☐ Electronic engineering      ☐ Cultural Heritage  
☐ CAD/CAE, Product Modelling      ☐ Gas, Oil      ☐ Transportation, Traffic  
☐ Geo-Informatics      ☐ Simulation  
☐ IT Service Provider      ☐ IT Software Developer      ☐ Web development  
☐ Game development      ☐ Health (Medical), Bioinformatics      ☐  
Management

3. Please specify the user category (or categories) you belong to:

☐ Professional user      ☐ Professional IT developer  
☐ Scientist      ☐ Creator of digital 3D content  
☐ End user      ☐ Publisher/dealer of 3D repositories  
☐ Other \_\_\_\_\_

4. Institution/Company size:

☐ 0-10      ☐ 11 - 20      ☐ 21 - 50      ☐ 50 - 100  
☐ 100 - 250      ☐ 250 - 500      ☐ 500 - 1000      ☐ over 1000



## GENERAL:

- Is the software is web based or standalone software.
- Why do we require the software.
- What is the working of the software.
- How it is different from manual system.
- Is the Short Messaging Services(SMS) is provided.
- How it is useful.
- What interface we are using at frontend and backend.
- Interface is of button based or menu driven.
- What is the compatibility and performance of software.
- How many modules user want.
- What are the modules where we want security.
- What are the number of users.
- Where can I lodge complaint against any type of grievances in the services of cabs, drivers, on this web site and how do I report the problems on this web site and give suggestions?

## NEED SPECIFIC:

How do I use "cab services"?

Rules / Passenger Services

Journey planning

Knowing source/destination codes

## PLEASE NOTE:

The site maintenance activities takes place from 0200hrs to 00350 hrs (Indian Standard Time) daily. During this period information may not be available. Also, note that since the system is under maintenance activities status changes might take a larger time place to book a cab during this period.

**INTERNET BOOKING FACILITY THROUGH CREDIT CARDS:** Internet booking facility is available on the web site <http://www.cabbookings.com.in> is operational, wherein you can get the cab booked. For more on bookings click here [Internet bookings](#).

## Software Requirement Specification :-

---

### **FULL TAXI DISPATCH SYSTEM web-based.**

Will include:-

- Geolocalisation ( Client mode on Dreevo , Iphone, Android, etc...)

Geolocalization services are provided directly by the satellite to the employees of the company which are provided with the GPS enabled cell phones so that they can track, heck and automize the services of the cabs.

- Track via GPRS all the taxis equiped with Dreevo ( at least )

All the cabs are fitted with automated GPRS system and are connected round the clock with the main server for their location information.

- SMS function ( for dispatch )

Now all the enquiries offered on the web site [www.cabbookings.com](http://www.cabbookings.com) are available on your mobile phone through SMS facility. For more information on the mobile service providers and the key words to be used on the mobile, please click here, SMS help .

Please note that we are giving the backend service only for the SMS queries. For more information and help on key words and SMS facility, kindly contact the mobile service provider according to the table.

- Store in a Database all the customers with ID, Address, Telephone, X, Y.

The database is maintained with the customer id, address and telephone numbers of all the customers.

- Search the closest available taxis to a specific address of the a Customer (based on a Google Maps Cartography )

Service provider can search the closest available cab which is nearer to the customers address and the service is based on a technique based on google maps

- Keep track of the Service until it finishes and receive approximate distance and time elapsed.
- Has a dedicated module to introduce new customers to the database, this is done through address search.
- Send the Service via GPRS and receive the information from the taxis via GPRS. ( And SMS to ! )
- Integrated function for payment onboard and invoice system on the web-interface.

- The solution must have an Operator Command Panel to lead the dispatch and make the regulation of the taxi traffic, with CHAT and IM in direct with the drivers on the road ( through the client : Dreevo, Iphone, Android...)

The purpose of this case study is to describe the cab booking system which provides the cab booking details ,billing, and cancellation on various types of booking namely.

- Confirm booking.
- Online booking.
- Phone booking.

- Main heading: Bold

### 1.3 Intended Audience and reading suggestions:

The different types of readers are:

- (a).Customers
- (b).Developers
- (c).Management people

Specifically,

Passengers

#### **1.4 Definitions, Acronyms, and Abbreviation:**

Following abbreviation has been used throughout this document:

CBS: Cab booking system.

PDA: Personal Digital Assistant

IVRS: Voice Response System

- Cab Details
- Booking Form
- Billing
- Cancellation

#### **1.5 Scope**

Passenger Revenue Enhancement.

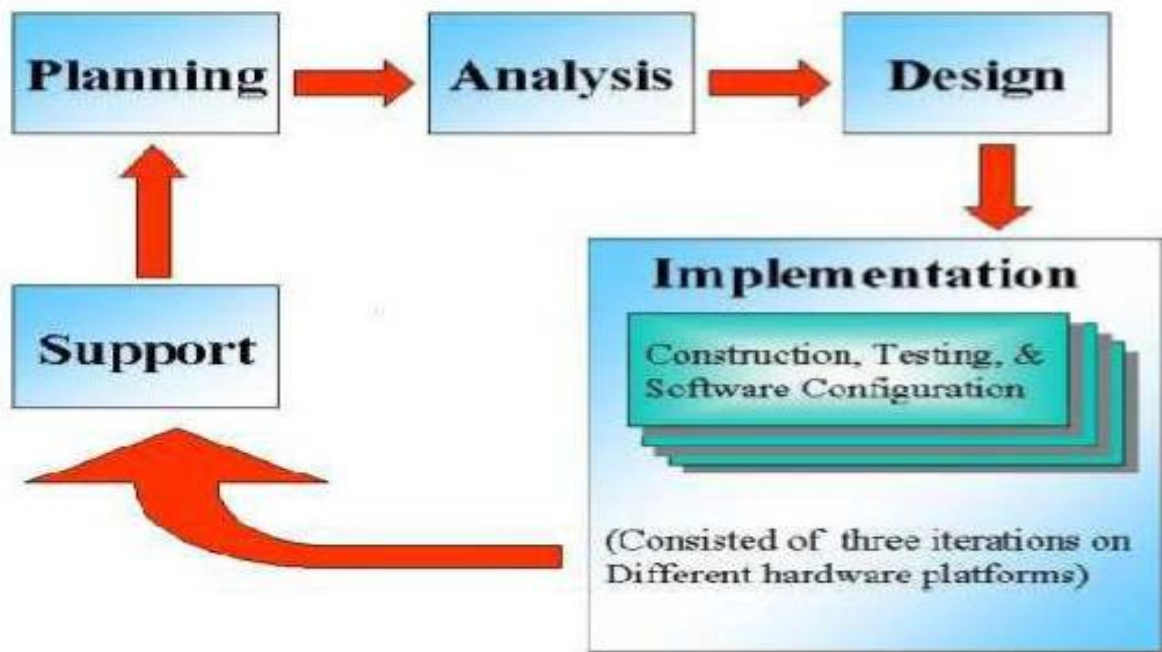
Improved and Optimized service

#### **1.6 Approach**

### **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 conscabts 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 proposals purpose, identify expected benefits, and explain how the proposed system supports one of the organizations 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, onetime 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 (I TPR)



process. The I TPR must be approved by the State CIO before the project can move forward.

## **PICTORIAL REPRESENTATION OF SDLC:-**

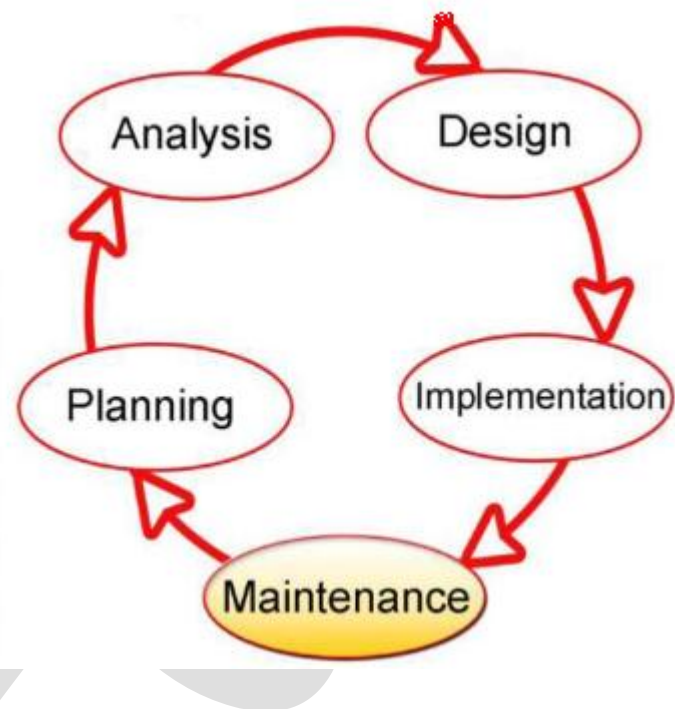
### **Requirements Analysis Phase:**

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 Cabling Plan.

### **Development Phase:**

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.

### **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. Security staff assesses 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 cabing, 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 organizations 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.

### 2. Overall description:

It enables us to maintain the cab details like this.

**1Product Perspective:** heir timing number of seat available, and booking billing and canceling the cab.

#### 2.1.1 User Interface:

Key Board and Mouse.

#### 2.1.2Hardware Interface:

## System requirements

---

#### Minimum system requirements for Cab Management:

- **Processor:** Intel Core 2Duo or AMD equivalent, 2Gz or better
- **RAM:** 2GB
- **HDD:** 160 GB, 7200k spin
- **Operating system:** Windows XP sp3 / Windows Vista Business sp1
- Business broadband connection with static IP (8mb download)
- Online backup
- Logmein, or Windows Remote Desktop Connectivity.

Other suggested components

- 17 inch or larger display
- Secondary hard drive – for additional local backups

### Minimum system requirements for CMS Mobile:

- Operating system Windows Mobile 6
- Touchscreen
- GPS Module
- GPRS Data Plan

### 2.1.3 SOFTWARE INTERFACE:

☐ Front end -> C++

☐ Back end -> MS-Access

### 2.1.4.Communication Interfaces

The website [www.cabbookings.com](http://www.cabbookings.com) offeres CBS enquiries on the internet availability, status , fare,service area etc.

Mobile telephone based SMS inquiry service.

Setting up of voice response system.

### 2.1.4Operating Environment:

The OS types are:

- ❖ Windows NT
- ❖ Windows XP
- ❖ Windows Vista
- ❖ Windows 7

### 2.1.5 Operations:

- Any booking can be done 24 hours.
- One form for single booking.
- Booking is done through pre defined logic.

### 2.2Product Functions:

It tells the short note about the product.

### **2.2.1Cab Details:**

Customers may view the cab timing at a date their name and their type of booking.

### **2.2.2Booking:**

After checking the number of cab available the customers books a cab or number of cabs according to their requiremnets.

### **2.2.3Billing:**

After reserving the required cab, the customer pays the amount in advance (optional).

### **2.2.4Cancellation:**

If the customer want to cancel the cab then 10% of the amount per person is deducted if the booking is cancelled before 30 mins of the service time.

## **2.3 User characteristics:**

Knowledge user

No voice user

Expert user

## **2.4. Conscabts**

- ✓ Less than 1 sec for local transactions.
- ✓ 3 sec for network transaction.
- ✓ Uptime of CBS is 99.5+%.

SOFTWARE CONSCABTS:

Designing->Rational Rose

## **3. SPECIFIC REQUIREMENTS**

### **3.1. EXTERNAL INTERFACES**

Cab Delay Alert Service.

Booking Terminals

Interactive voice Response System.

### **3.2. PERFORMANCE REQUIREMENTS:**

It is available during all 24 hours.

Offered through Air conditioned or non Air conditioned Cabs.

About 1600 cabs run daily.

*Types of concerns and complexities:*

Special 10 % discount is given to corporate customers for their advance monthly bookings.

### **3.3 Software system Attributes:**

- Reliable.
- Available.
- Secure.

### **4. Document Approval**

The bill passed on any proposal related to cab management needs approval of the top level management.

## **5 Testing**

### **5.1 Introduction**

Till now the database design, user interface design and implementation are complete. The system now is tested for its functionality, validity and performance. In order to test the system, a wide variety of tests are conducted to make sure that the system matches the entire identified user requirements and conscabts. This chapter focuses on testing the developed systems using different test strategies in order to verify its correctness and user acceptance.

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error.

The development of software systems involves a series of production activities where opportunities for injection of human fallibility are enormous. Errors may begin to occur at the very inception of the process where the objectives may be enormously or imperfectly specified, as well as in later design and development stages. Because of human inability to perform and communicate with perfection, software development is accompanied by a quality assurance activity.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and coding. And it needs to be done in almost every phase of product development life cycle not just before a product is handed to a customer.

The following are some attributes of a good test:

- A good test has a high probability of finding an error. To achieve this goal the tester must understand the software and attempt to develop a mental picture of how the software may fail. Ideally the classes of failure are probed.
- A good test is not redundant: testing time and resources are limited. There is no point in conducting the test that has the same purpose as another test. Every test should have a different purpose.
- A good test should be best of breed. In a group of tests that have a similar intent time and resource limitations may militate for the execution of only a subset of these tests. In such cases the tester that has the highest likelihood of uncovering a whole class of errors should be used.
- A good test should be neither too simple nor too complex: although it is sometimes possible to combine a series of tests into one test case, the possible side effects associated with this approach may mask errors. In general each test should be executed separately.

## **5.2 Types of testing**

### **5.2.1 White Box Testing**

White box testing, sometimes called glass box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that :

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structure to assure their validity.

White box testing was performed at all levels of development of i-Admit. The coding team took all care to test the code and guarantee that it meets all the specifications as well as logically correct. All loops were tested and all internal data structures evaluated and verified.

### **5.2.2 Black Box Testing**

It focuses on the functional requirements of the software. That is black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white box techniques. Rather it is a complementary approach that is likely to uncover a different class of errors than white box methods. Black box testing attempts to find errors in their following categories

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external data base access
- Performance errors
- Initialization and termination errors.

Unlike white box testing, which is performed early in the testing process, black box testing is to be applied during later stages of testing.

Black box testing was performed with the application code of the software being developed to verify that it is functionally correct and gives appropriate output at different situations of inputs. It was also verified that the software takes a good care of exceptional conditions and errors are handled well and that the software does not crash.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. In many ways, testing is an individualistic process, and the number of different types of tests varies as much as the different development approaches.

### **5.2.3 Unit Testing**

Unit testing focuses verification effort on the smallest unit software design- the module. Using the procedural design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The module interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in algorithmic execution. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing. All independent paths (bases paths) through the control structure are exercised to ensure that all elements in a module have been executed at least once. And finally all error-handling paths are tested.

Application interface of our system was unit tested at all levels of implementation, right from start of code writing, to integrating the code with other modules. Every module was tested fully to check its syntax and logical correctness. Error handling was implemented into relevant modules so that the code doesn't crash on errors.

#### **5.2.4 Integration Testing**

Integration testing is a systematic technique for constructing the program structures, while conducting test to uncover errors associated with interfacing, the objective is to take unit tested modules and build a program structure that has been dictated by design.

User interface of i-Admit was developed in modules. All of them were joined together to make the complete running application. While integrating these modules, integration testing was performed on them to verify that they meet all interfacing requirements and that they pass relevant information among themselves. In the end the complete program structure was tested to ensure interoperability of all the modules.

#### **5.2.5 Validation Testing**

At the culmination of integration testing software is completely assembled as a package: interfacing errors have been uncovered and corrected and a final series of software tests – Validation Testing may begin. Validation can be defined in many ways, but a simple definition is that validation succeeds when software functions in a manner that can be reasonably expected by the customer. Software validation is achieved through a series of Black Box tests that demonstrate conformity with requirements.

#### **5.2.6 Alpha Testing**

It is virtually impossible for a software developer to foresee how the customer will really use a program. When custom software is built for one customer a series of acceptance tests are conducted to enable the customer to validate all requirements.

A customer conducts the alpha test at the developer site. The software is used in a natural setting with the developer "looking over the shoulder" of the user and recording errors and usage problem. Alpha tests are conducted in a controlled environment.

Alpha tests were performed at our development site with the help of our friends, who were called and asked to run the program in the manner they like, without our guidance and errors and usage problems were noted and code was updated to remove all of them.



### **5.2.7 Beta Testing**

The Beta test is conducted at one or more customer sites by the end user of the software. Unlike alpha testing the developer is generally not present. Therefore the beta test is a live application of the software in an environment that cannot be controlled by the developer. The customer records all problems that are encountered during beta testing and reports these to the developer at regular intervals. As a result of problems reported during beta test the software developer makes modification and then prepares for the release of software product to the entire customer base.

Beta testing of our system is not performed as the product is not yet fully developed and has not been installed at the user site as it still is in the development phase. Beta testing will be performed when the software is deployed at the user's site.

### **5.2.8 System Testing**

Ultimately software is incorporated with other system elements (new hardware, information) and a series of system, integration and validation tests are conducted. It is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose all work to verify that all system elements have been properly integrated and perform allocated functions.

System testing of this system was performed at the development lab of this system by integrating the functional systems to imitate the actual work environment. Since no special hardware was needed for this purpose, the testing proceeded and succeeded with no errors.

### **5.2.9 Recovery Testing**

Many computer-based systems must recover from faults and resume processing within a pre-specified time. In some cases a system must be fault tolerant that is processing faults must not cause overall system function to cease. Recovery testing is a system test that forces the software to fail in a variety of ways and verifies that recovery is properly performed. If recovery is automatic, re-initialization, check pointing mechanisms, data recovery, and restart are each evaluated for correctness. If recovery requires human intervention the meantime to repair is evaluated to determine whether it is within acceptable limits.

### **5.2.10 Security Testing**

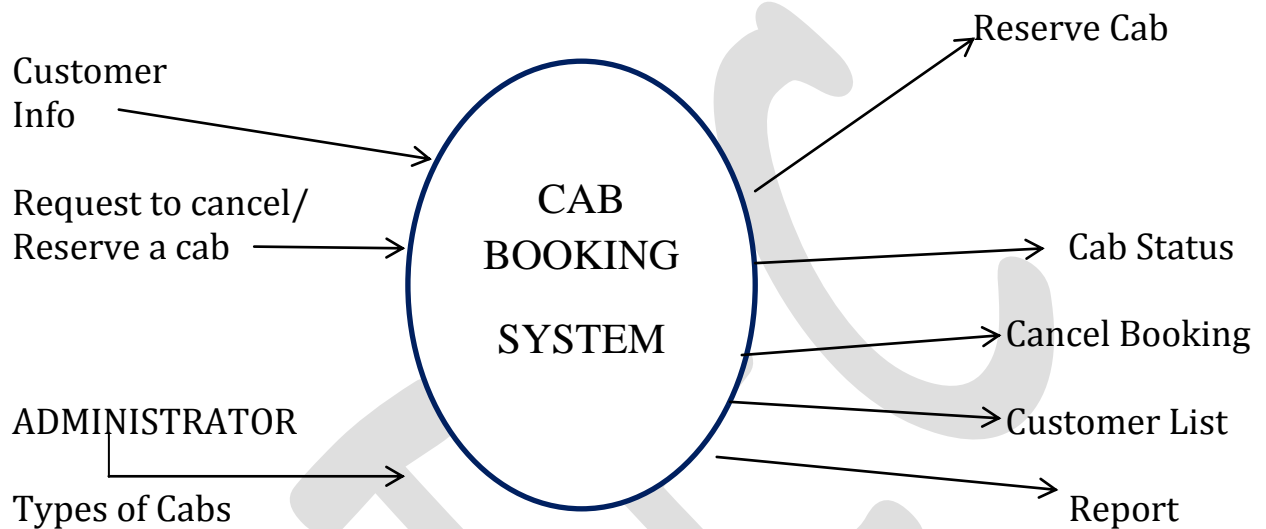
Security Testing attempts to verify protection mechanism built into a system will in fact protect it from improper penetration. Security is provided for each user by giving them login name and password. Security testing was done, as any other anonymous user can't log in with a user password if the user is already logged in.

### **5.2.11 Performance Testing**

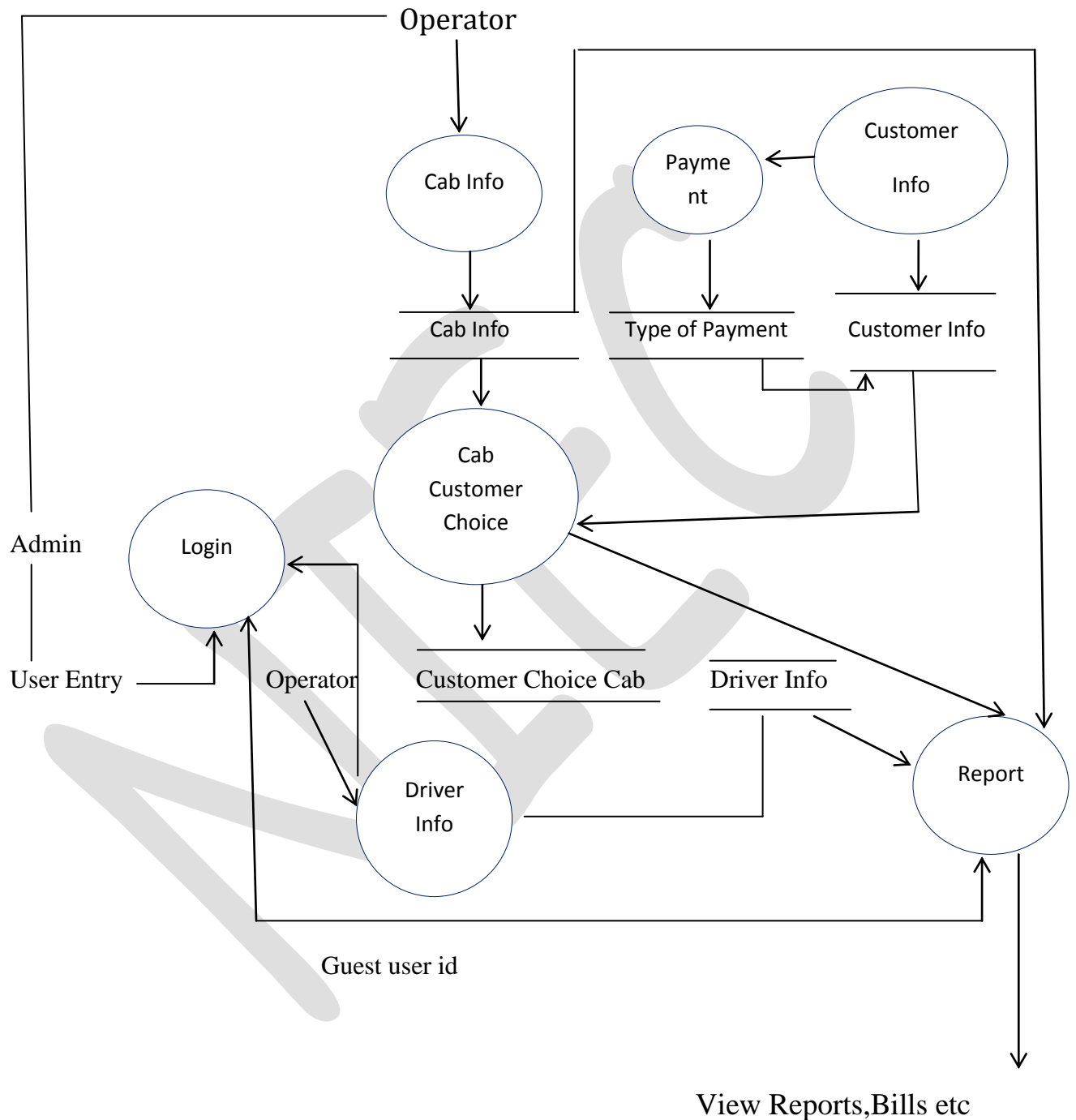
Performance Testing is designed to test run time performance of software within the context of an integrated system. Performance Testing occurs throughout all steps in the testing process. Performance tests are often coupled with stress testing and often require both hardware and software instrumentation. That is it is often necessary to measure resource utilization in an exacting fashion. External instrumentation can monitor execution intervals, log events as they occur, and sample machines take on a regular basis. By instrumenting a system the tester can uncover situations that lead to degradation and possible system failure.

## Context Diagram

---



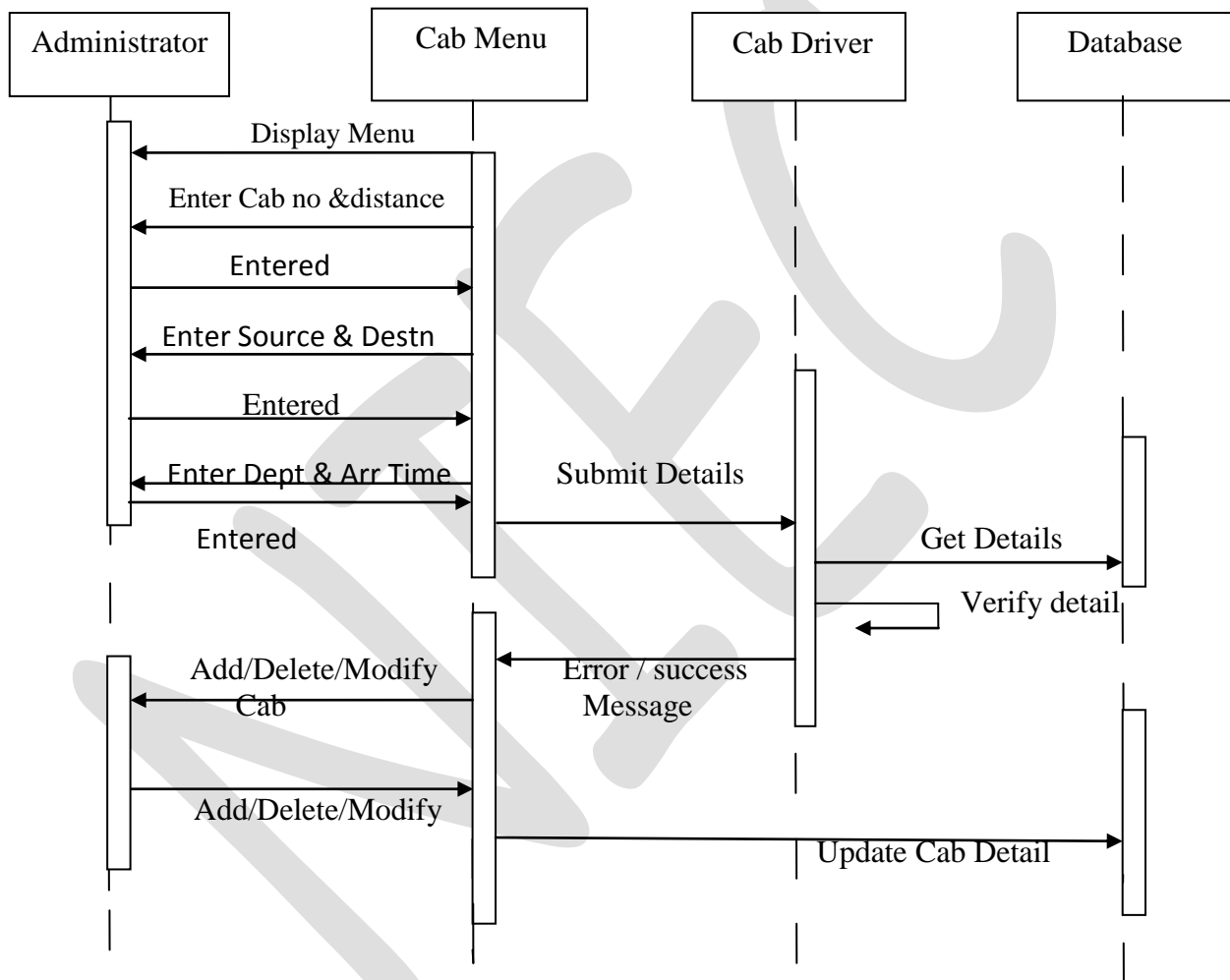
## LEVEL 1 DFD



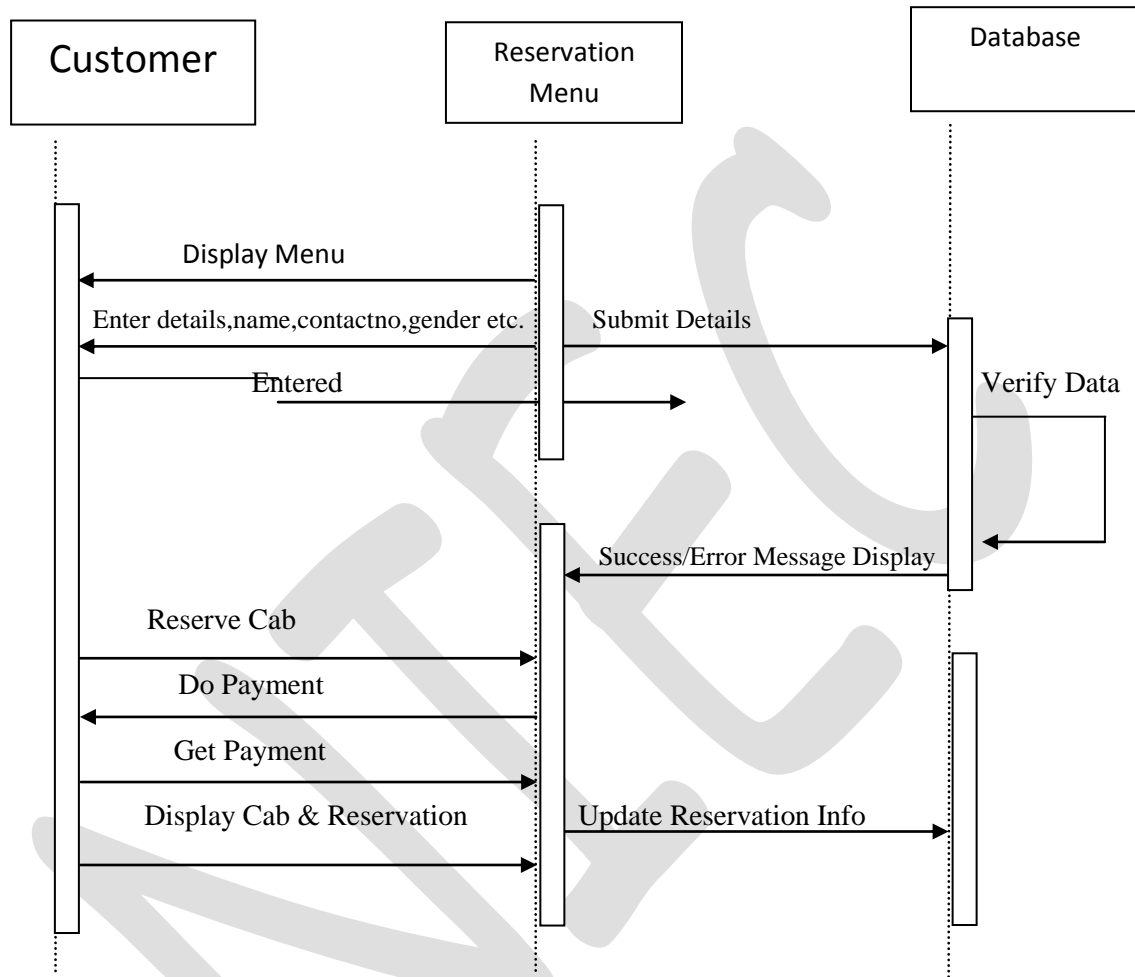
## SEQUENCE DIAGRAM:-

---

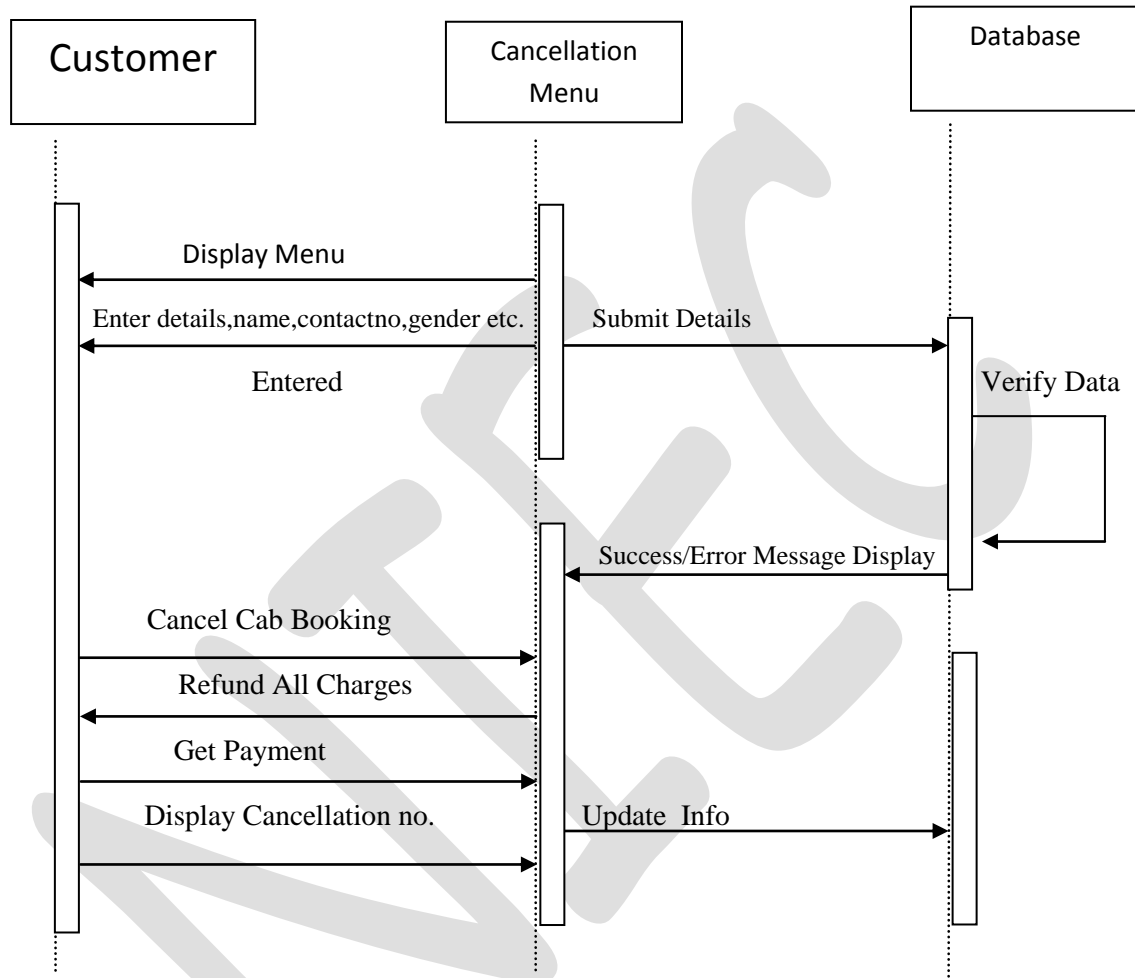
### MAINTAIN CABS INFORMATION



## SEQUENCE DIAGRAM:- Reservation



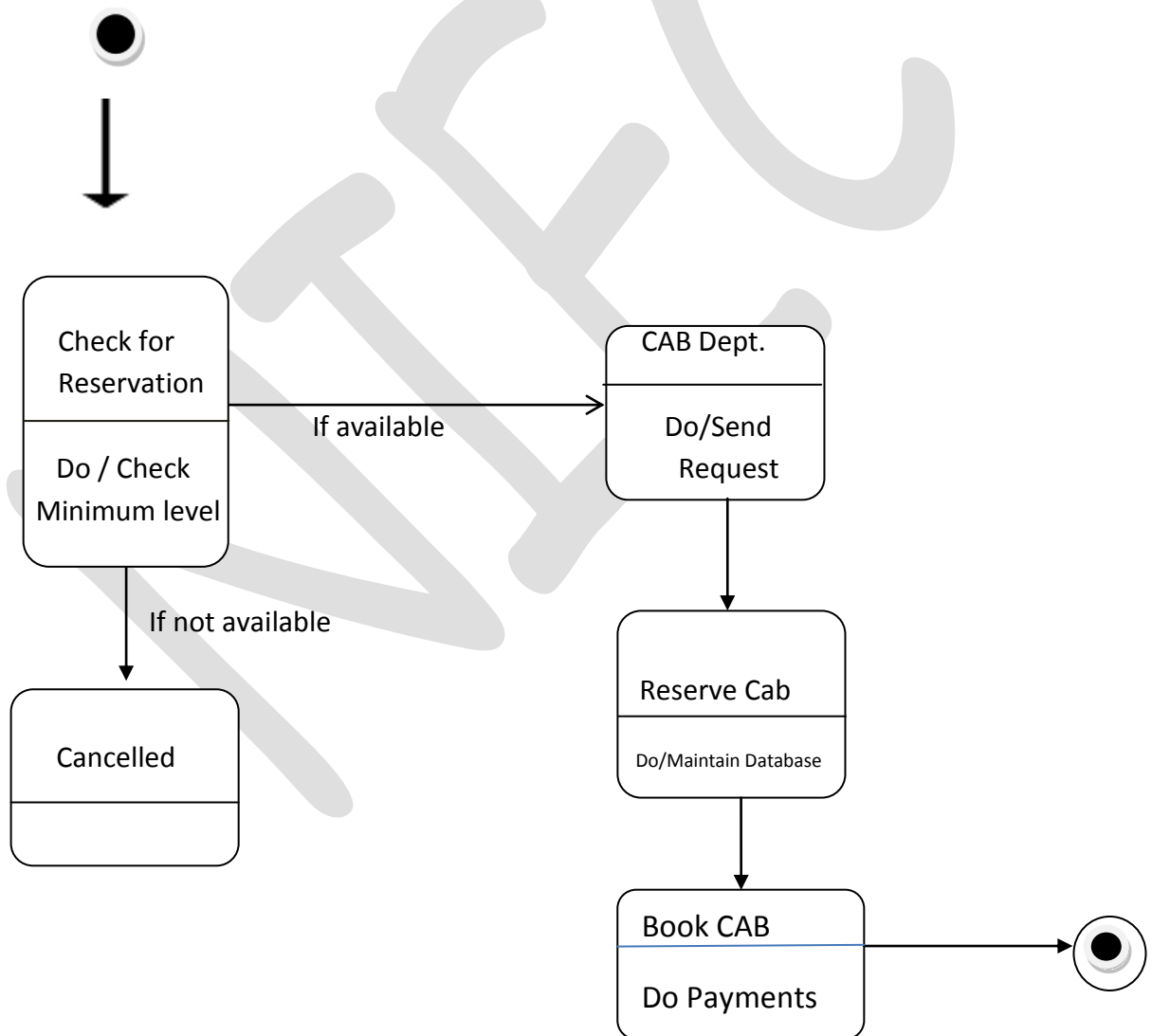
## SEQUENCE DIAGRAM:- Cancellation



## STATE CHART DIAGRAM :

---

### RESERVATION

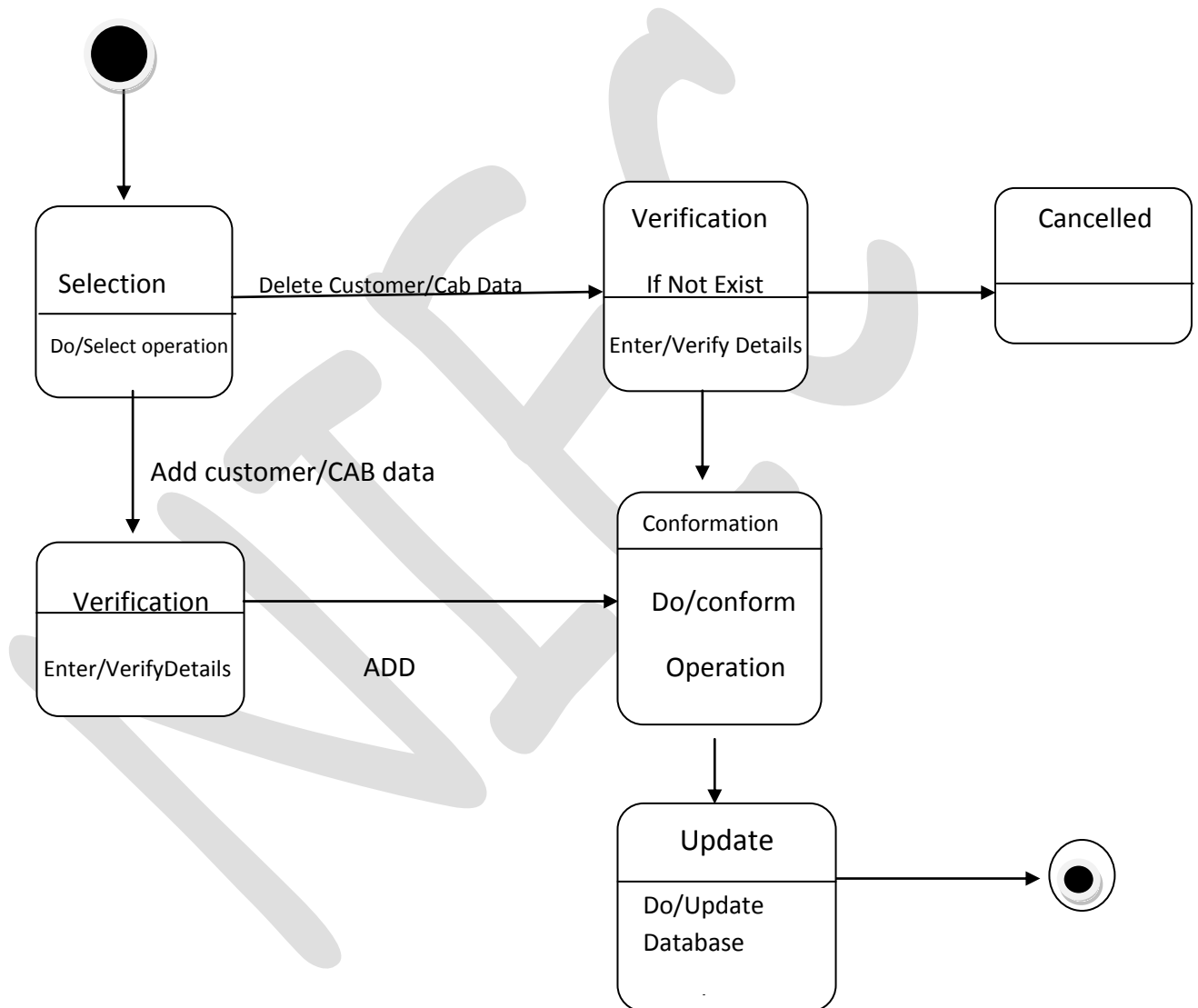




## STATE CHART DIAGRAM :

---

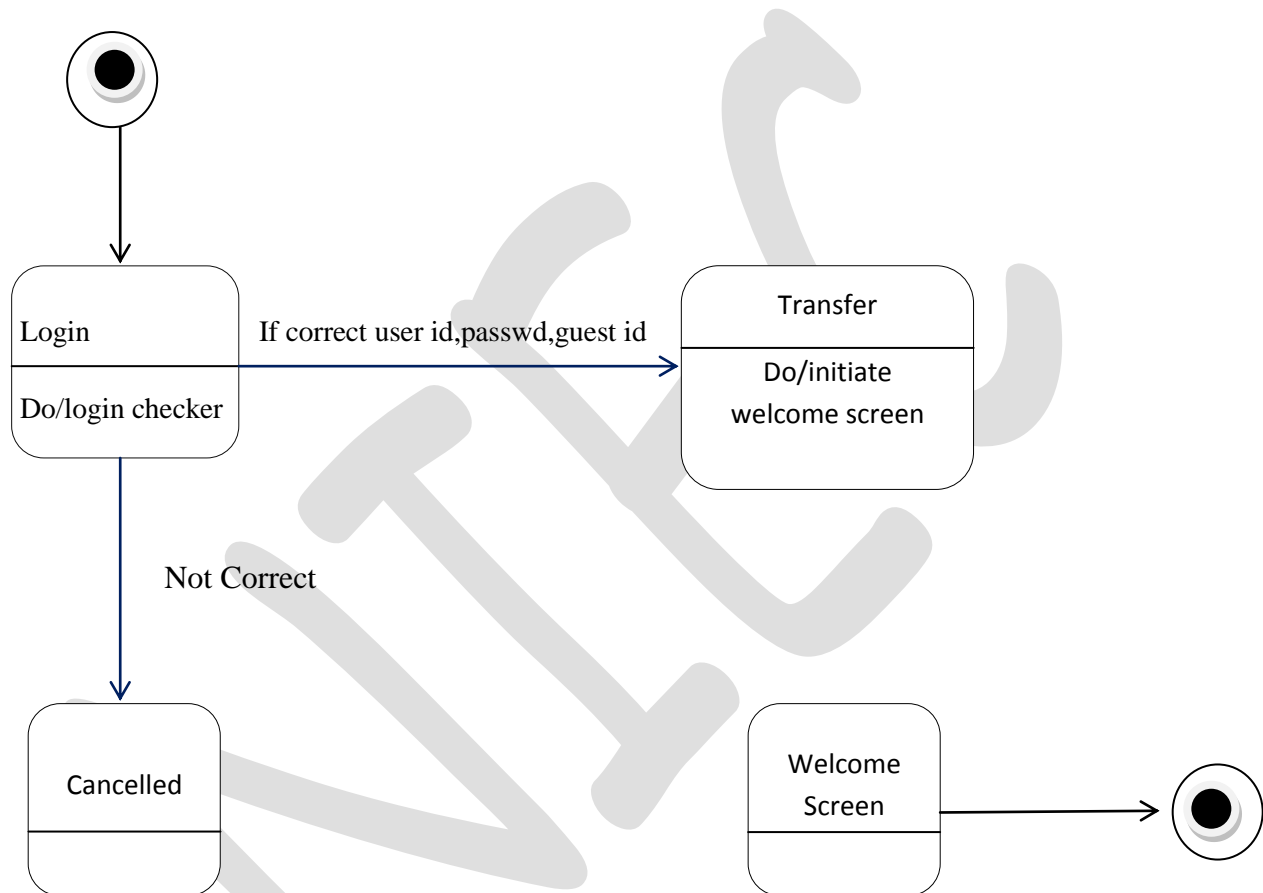
### MAINTAIN CUSTOMER/CAB INFORMATION



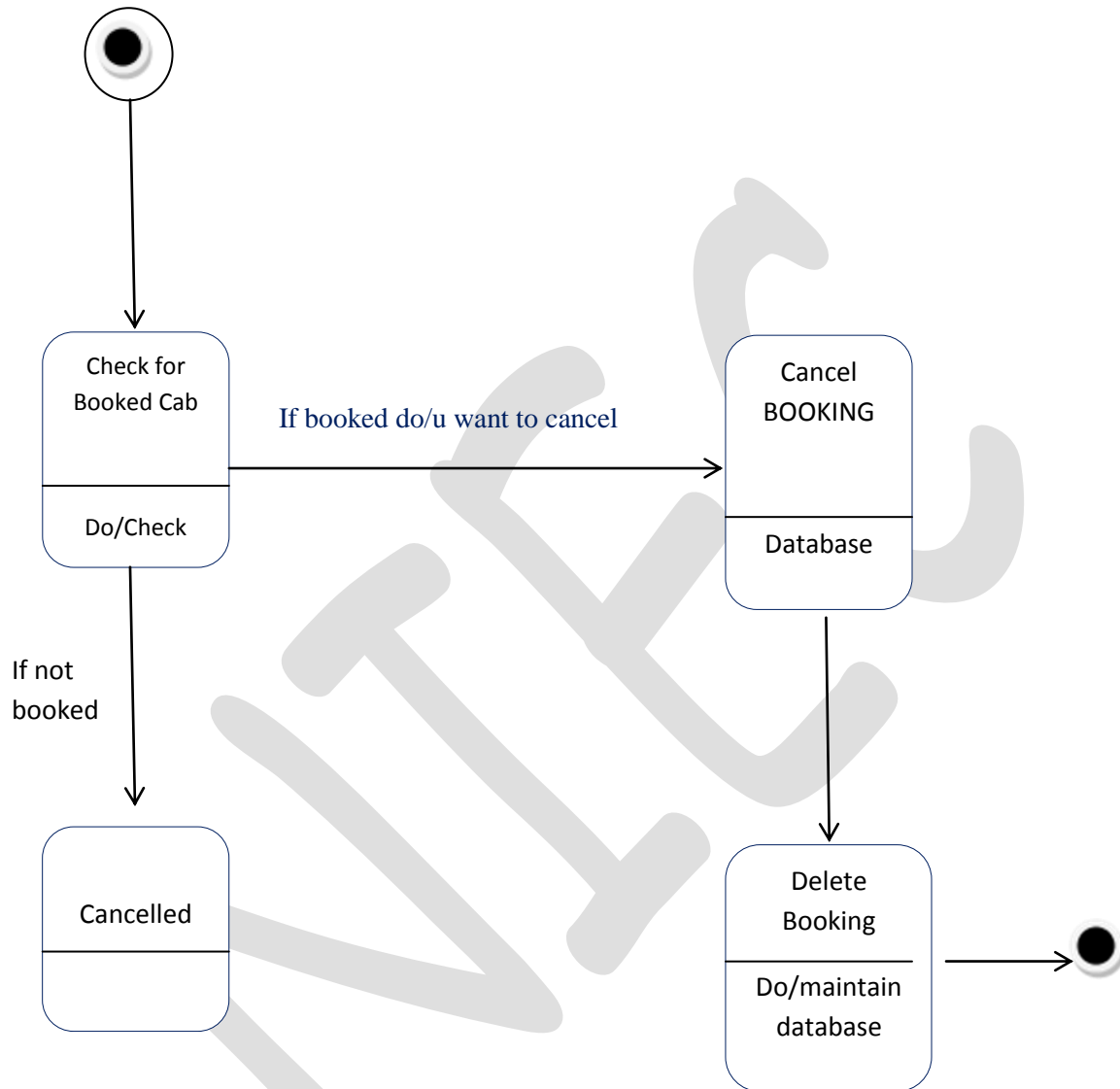
## STATE CHART DIAGRAM :

---

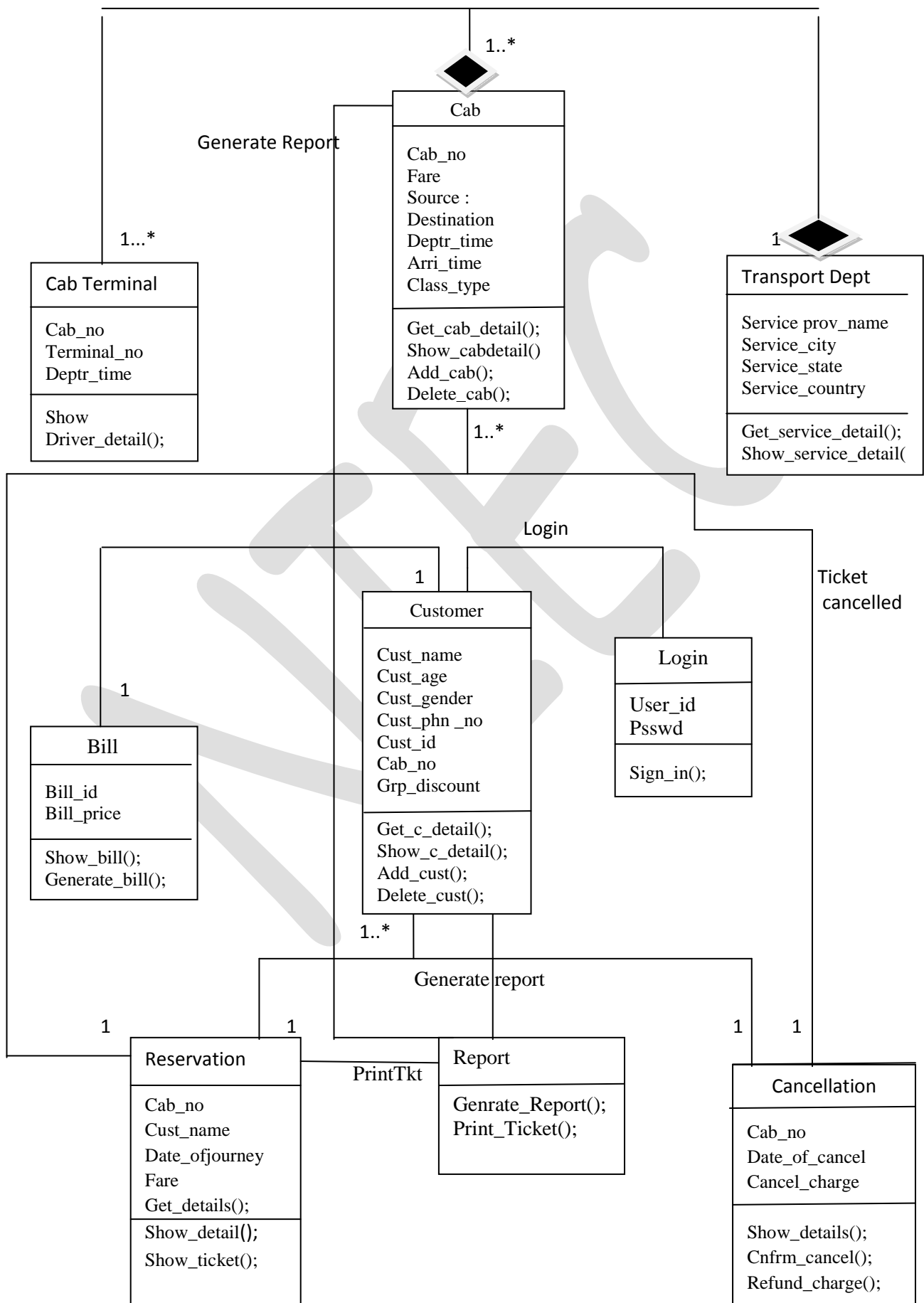
### LOGIN



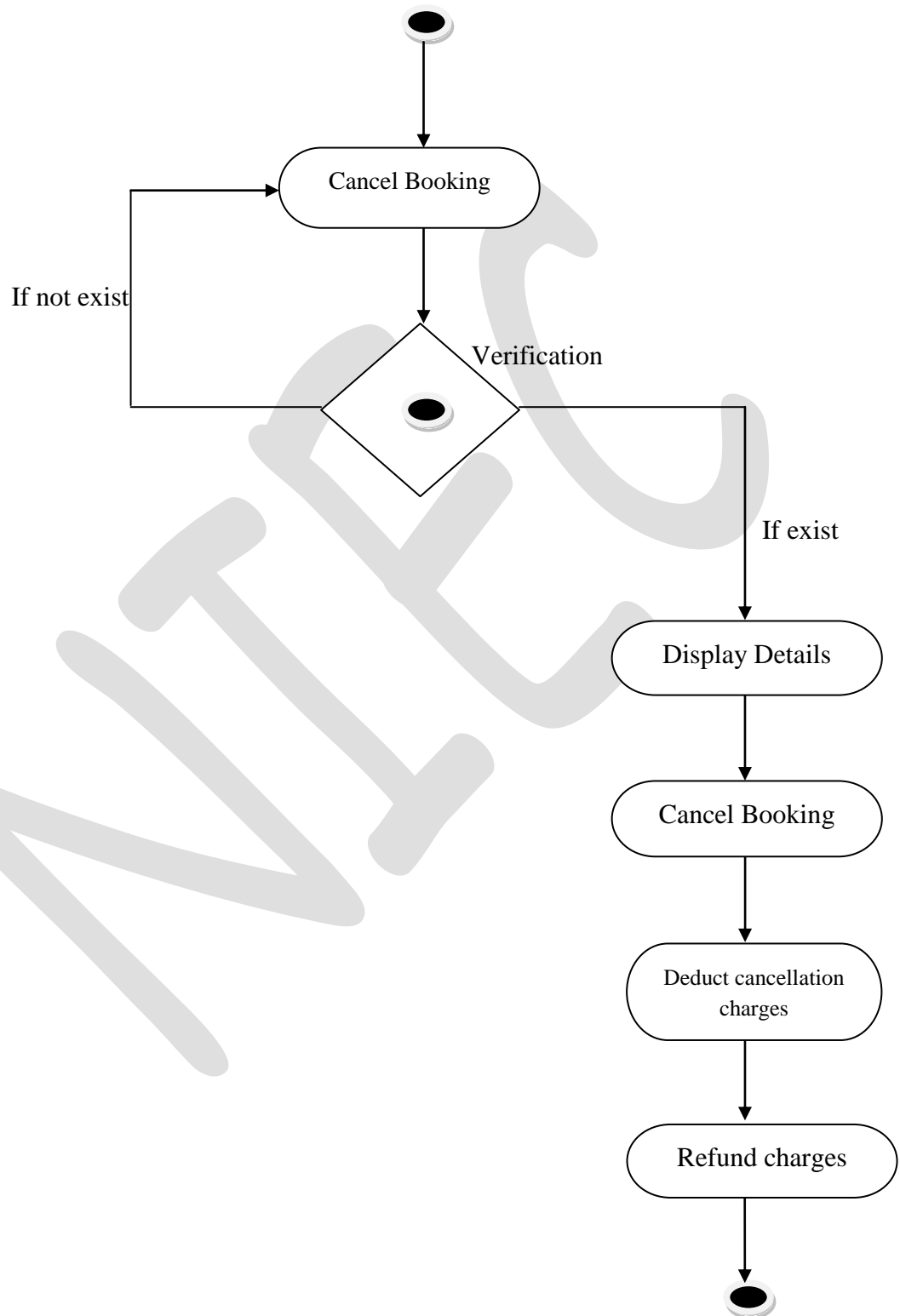
## STATE CHART DIAGRAM\_: CANCELLATION



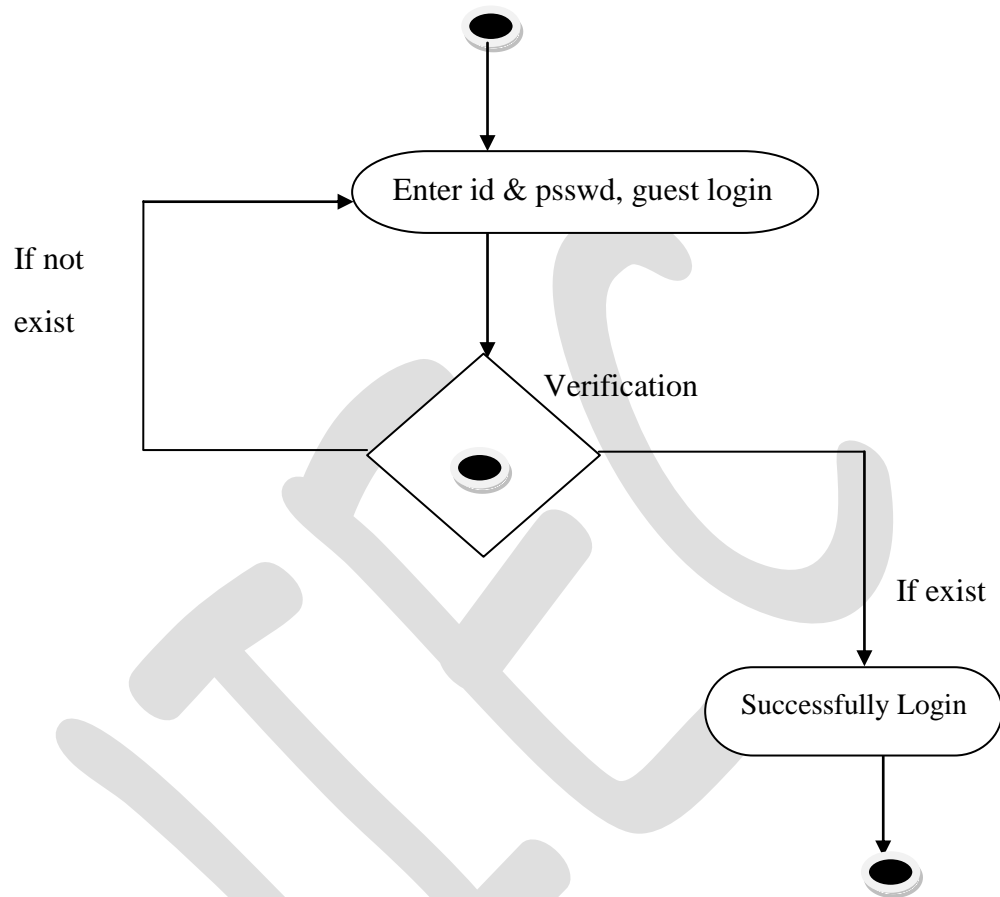
# CLASS DIAGRAM



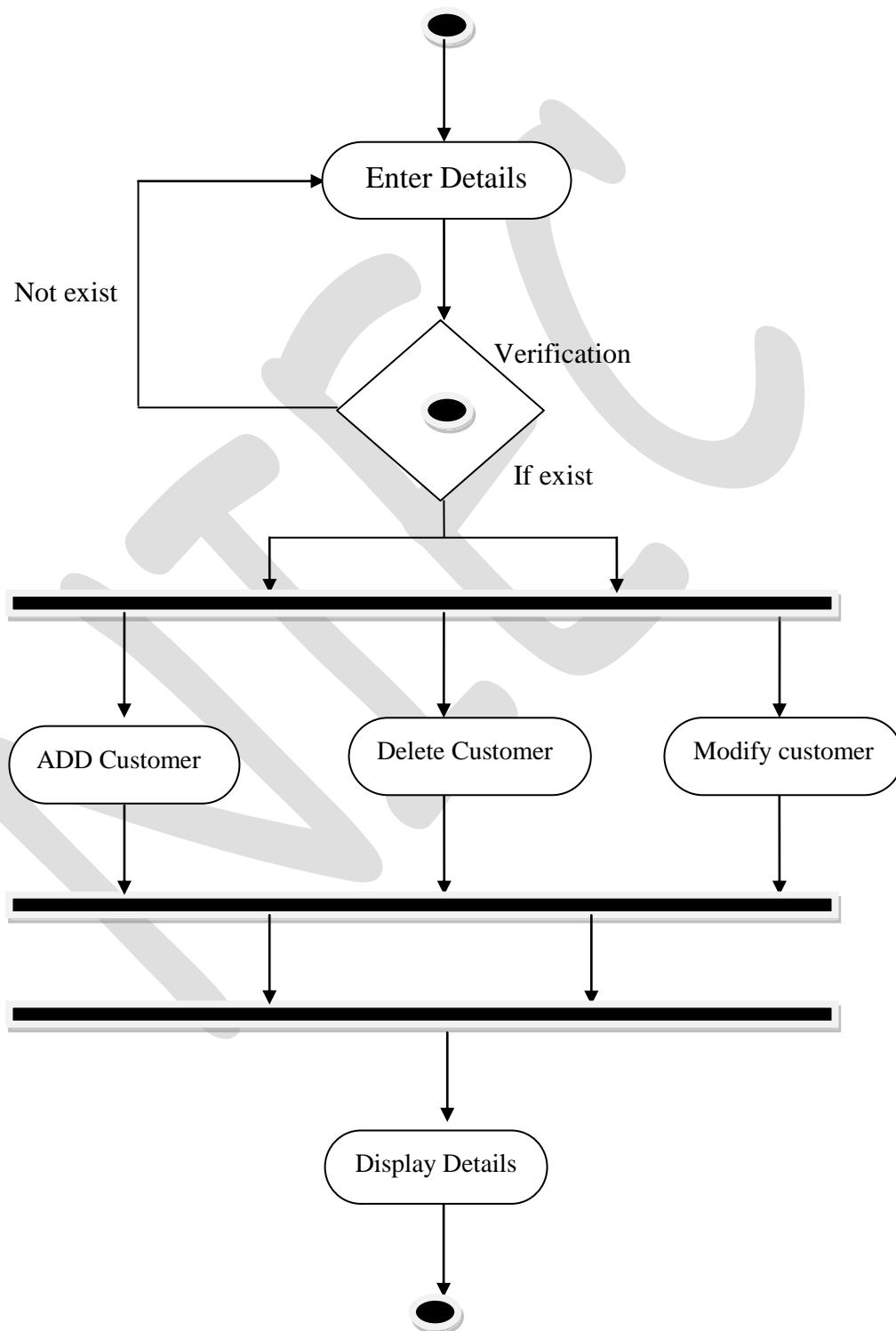
## Activity Diagram : CANCELLATION



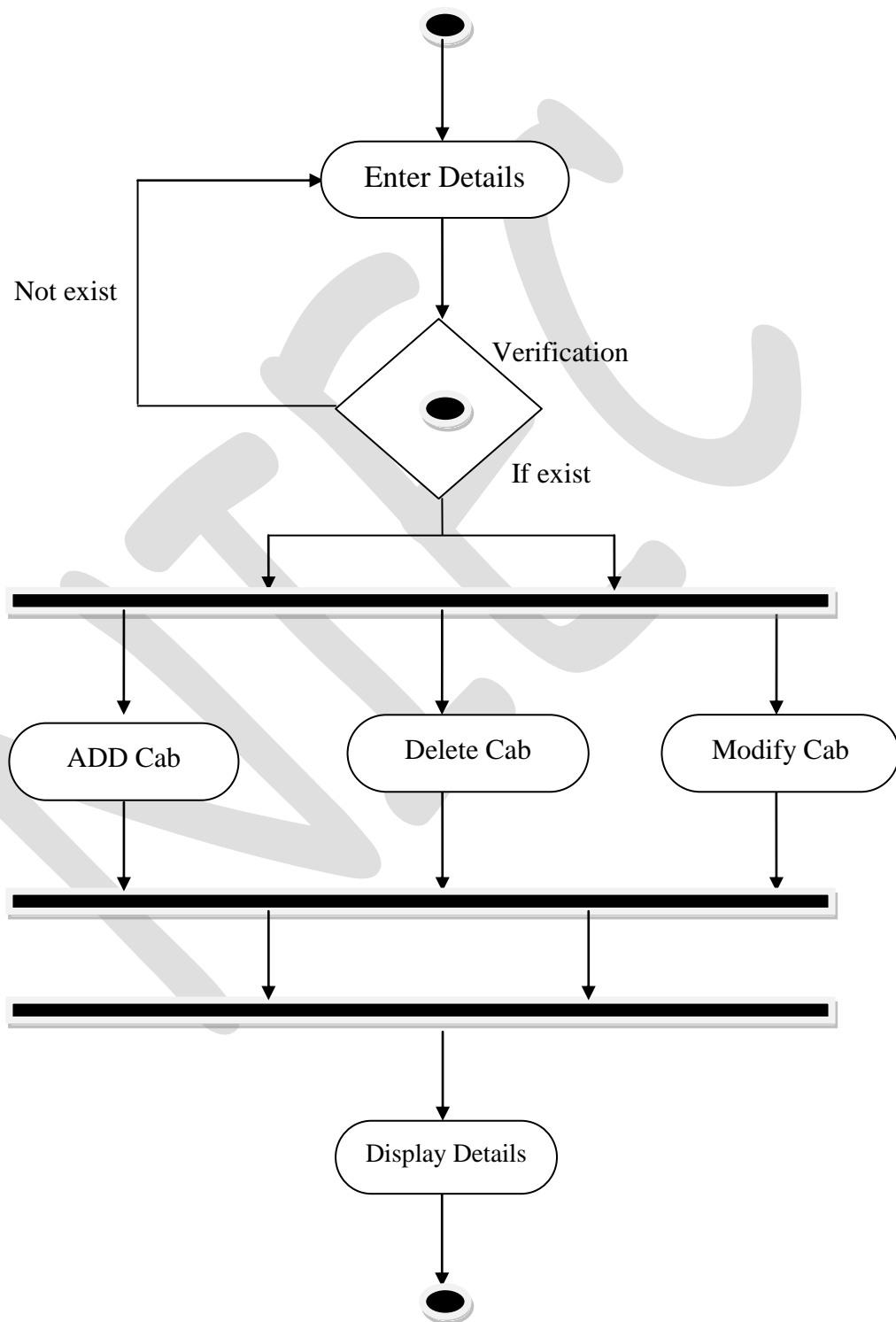
## Activity diagram : LOGIN



## Activity diagram : MAINTAIN CUSTOMER INFORMATION

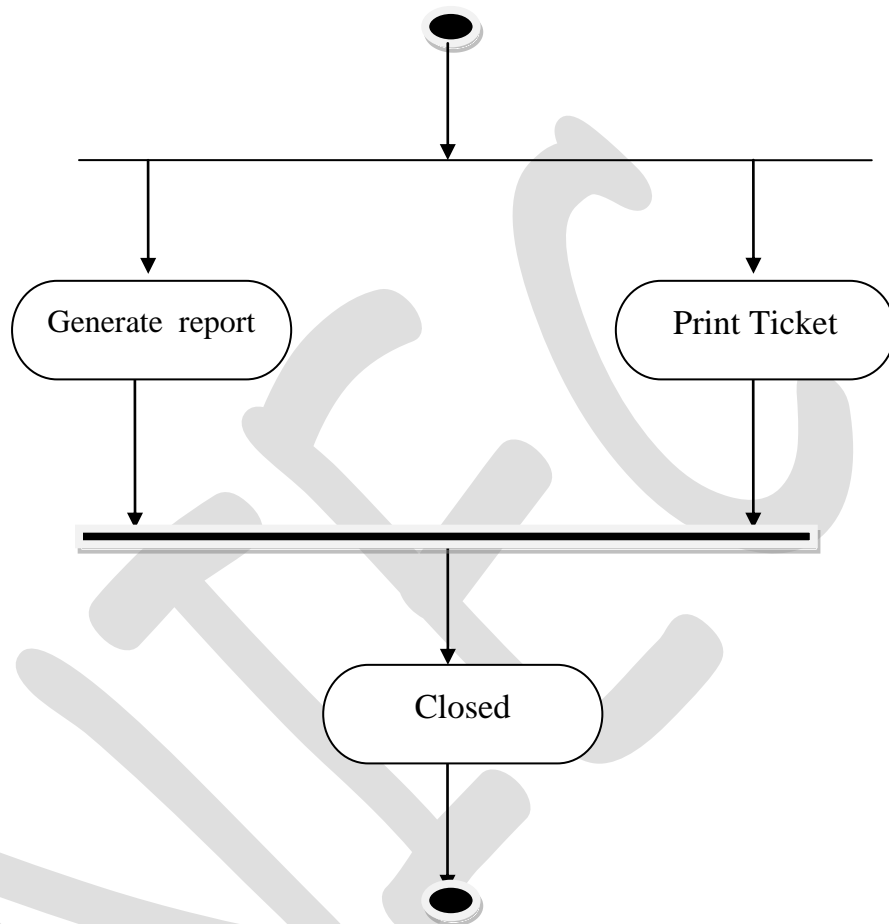


Activity diagram : MAINTAIN CAB  
INFORMATION

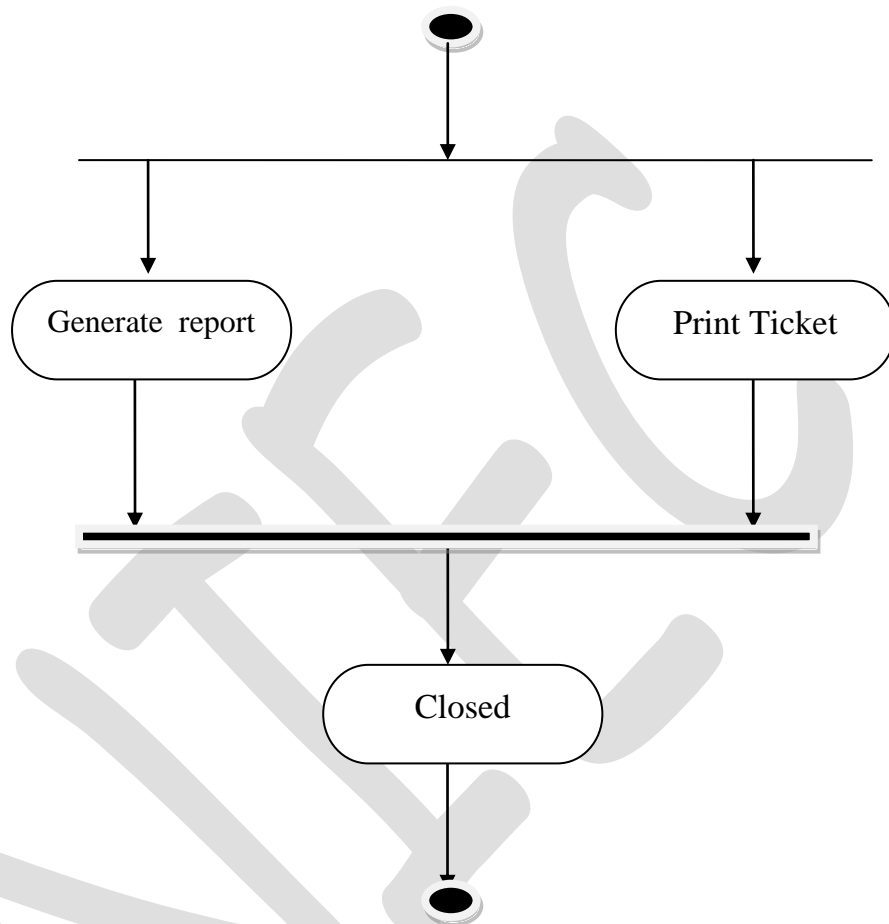




## Activity Diagram : REPORT AND TICKET GENERATION



## Activity Diagram : REPORT AND TICKET GENERATION



## ADVANTAGES OF CAB BOOKING SYSTEM

---

Now one can easily plan the journey comfortably as the process is efficient and fast with being easy to access. Bookings can be made through the cab booking site or by the phone call. This being a big step in terms of improvement in the cab system it is widely accepted across the country.

- \* A route-based booking system that facilitates the issue of journey-cum-booking cab, which can be issued from any station to any station.
- \* Passenger journey to multiple laps of booking can be handled from a single terminal window.
- \* The booking facility is offered round-the-clock (24 hours uninterrupted).
- \* Changes in cab profiles (cab addition, replacement, de-allocation), route structures, etc., can be made effective immediately with the appropriate contingency handling.
- \* Dynamic definition of the advance booking period is possible. This feature facilitates defining different advance booking periods for different cabs.
- \* Any cab running schedule can be accommodated.
- \* Provides on-line aggregation of EIS figures such as revenue, cab utilisation, etc, and presentation of the summarised data in the form of visual analytics from the operational system's information store. The data aggregation is done incrementally, to inflict minimal impact.
- \* Provides automatic database recovery against all kinds of hardware and software failures.
- \* Complete audit trails for transactions and data access.
- \* The application software is parametric, and standard cab business rules are incorporated in the form of data instead of being part of the logic.

## Conclusions

---

Information Technology plays a vital role not only in a particular field, it provides various kinds of solutions and services to the various problems prevailing in many fields. Cabs exploits information technology at the maximum extent. It uses the information technology in an efficient way for providing better passenger services. The online booking system helps to solve the every day problems of the world biggest Indian .

